Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 6.199

IJCSMC, Vol. 8, Issue. 5, May 2019, pg.132 – 142

SMS SECURITY SYSTEM USING ENCRYPTION TECHNIQUES

[1]Jitha P V; [2]Unnikrishnan S Kumar
[1]M.Tech scholar, MCET, Thrissur, Kerala
[2]Asst. Prof (CSE Department), MCET, Thrissur
[1] jitha5593@gmail.com; [2] uksknair@gmail.com

ABSTRACT: ***Communication has been a major factor in human's everyday life. The advancement of information technology changed the method of online transactions. No one uses SMSs that frequently to pass messages from one to another because of the arrival of the free data services and cheap data packs from the ISPs. But SMS still plays a very vital role in our day to day lives and its theft is increasing concerns to secure it. No airline or bank or subscription services send their transaction details through any web application but through SMS. SMS plays an important role in online transactions .With the increasing use of SMS for communication and information exchange, care should be taken when sensitive information is transmitted using SMS. Users should be aware that SMS messages might be subject to interception. Solutions such as encrypted SMS should be considered if there is a need to send sensitive information via SMS. The messages are encrypted by strong cryptographic algorithms. The encrypted message received by the customer decrypt the message by authentication. This is only known to user. It uses RSA algorithm for secure keys and AES algorithm to secure message. Authentication is implemented by using pattern lock. This system provide secure and reliable communication environment.***
Keywords: Encryption, cryptography, authentication, SMS, RSA, AES

I. INTRODUCTION

A computer or a cyber-crime [1] refers to a crime or a malicious act of causing harm to an individual or property of an individual by taking the aid of a computer or a network of computers. Life of almost most of the people in this world is dependent on computers and so on the internet. When people share their lives i.e. their personal information on the network or keep it in the storage of a device then there is a vulnerability of theft of information. One such attack is the information leak attack [2]. As the name suggests information leak attack refers to all the attacks related to theft of information from a device or by intercepting the information on a network. Information theft is very dangerous and can cause a lot of damage and disturbance if applied at a sensitive site such as bank transactions. Suppose, a transaction took place between a bank and a customer online, which means crucial data transfer from one node to another. Now, if someone is able to leak this information while it is on its way or afterwards at the client's device then he/she may be

able to know all the credentials of the customer and so will be able to access the bank account of the corresponding customer. Information leak attack can also be used to hamper privacy [3]. Communication can be leaked and all the personal information can thus be known. Moreover, all the browsing details of a person can be leaked.

Android is mobile Operating system based on the Linux kernel and currently developed by Google. Android is popular with technology companies which require ready-made, low cost customizable Operating system for high-tech devices. It is the customizable, easy to use Operating system. Mobile phone users desire more secure and private communication in their daily usage of their mobiles. This is especially important in communications of secret nature such as that in military and governmental communication. On the other hand, securing communication through the popularly used means, namely text messages, can be helpful and useful in many cases. We are going to describe a secured text messages communication environment via SMS. For this purpose, we are going to develop a mobile-based application named Safe Short Message Service (SSMS). It encrypts a text message before sending it and decrypts the message in the receiver's side. In this way, the message is unreadable while transmitted even if it is intercepted while transmitting it over the network. The proposed system can send encrypted messages via SMS and allow users to encrypt/decrypt messages for personal usage without sending them. The latter feature is desirable for those who want to ensure the privacy of their own information. SMS employs symmetric-key encryption. The same secret key is used for both encryption and decryption. Therefore, the secret key must be known by the sender and the receiver of the message. Key distribution remains a problem when using symmetric-key encryption, but we found that it is the best solution when considering time complexity, efficiency, and costs. SMS depends on secret key embedding, where the messages 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. This schema saves time and space as there is no need for a database to store the secret key related to each message.

II. SHORT MESSAGE SERVICE

SMS stands for short message service [4]. Simply put, it is a method of communication that sends text between cell phones, or from a PC or handheld to a cell phone. The "short" part refers to the maximum size of the text messages: 160 characters (letters, numbers or symbols in the Latin alphabet). For other alphabets, such as Chinese, the maximum SMS size is 70 characters.

A. Working of SMS

It is well-known that SMS service is a cell phone feature but indeed, SMS can also work on other computing devices such as PC, Laptop, or Tablet PC as long as they can accept SIM Card. SIM Card is needed because SMS service needs SMS center client which is built-in on the SIM Card.

BTS: A base transceiver station (BTS) is a piece of equipment that facilitates wireless communication between user equipment (UE) and a network. UEs are devices like mobile phones (handsets), WLL phones, computers with wireless internet connectivity, WiFi and WiMAX devices and others.

MSC: The mobile switching center (MSC) is the primary service delivery node for GSM/CDMA, responsible for routing voice calls and SMS as well as other services (such as conference calls, FAX and circuit switched data)[5]. The MSC sets up and releases the end-to-end connection, handles mobility and hand-over requirements during the call and takes care of charging and real time pre-paid account monitoring.

SMSC: When SMS is transmitted from a cell phone, the message will be received by mobile carrier's SMS Centre (SMSC), do destination finding, and then send it to destination devices (cell phone). SMSC is SMS service centre which is installed on mobile carrier core networks. Beside as SMS forwarding, SMSC also acts as temporary storage for SMS messages. So, if the destination cell phone is not active, SMS will store the message and then deliver it after the destination cell phone is active. As additional, SMSC also notify the sender whether the SMS delivering is success or not. However SMSC cannot store the SMS message forever since the storage capacity is not unlimited. During the SMS delivering, sender cell phone and SMSC is actively communicating. So, if the non-active destination cell phones become active, SMSC directly notifies the sender cell phone and tell that the SMS delivering is success.

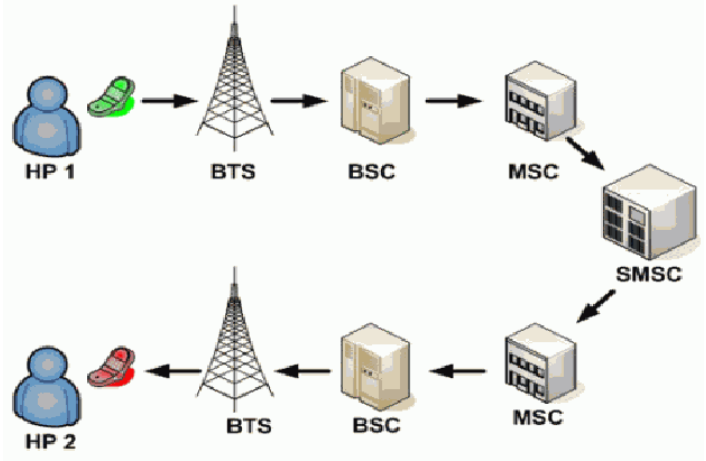

Figure 1: SMS transmission

B. SMS theft attack in android mobile phone devices

Many research works have been surveyed, to find the vulnerabilities in the android operating system, regarding the privacy thefts and the SMS shortcomings in the android APIs. Several in built APIs of the android operating system have been used to leak the messages (SMSs) through a parallel medium which are received by an android device.

Now these inbox messages are very much useful and can contain crucial data which can be misused. Now here are some classes and objects in the android and java API that could be used to leak all the messages very easily to an external file.

III. LITERATURE SURVEY

1. Trusted SMS communication on mobile devices.

In this paper author has introduced the higher growth of the Short Message Service (SMS) use has transformed this service in a widespread tool for social and commerce messaging. However, security concerns have been raised as applications become more critical and complex. Thus, this paper introduces an SMS security framework, which allows programmers and users to exchange confidential, non -reputable and digitally signed text messages. This framework can fit in many development scenarios, such as commercial transactions or bureaucratic delegations. In addition, the proposed framework is highly flexible and e fficient, since programmers can choose among several encryption algorithms according to the computational power and battery usage of each mobile device.

2. SMS encryption for mobile communication.

This paper deals with an SMS encryption for mobile communication. The SMS transmission in GSM network is not secure; therefore it is desirable to secure SMS by additional encryption. In SMS, there are compared differences in the use of symmetric and asymmetric cryptography for SMS transfer securing. In the next part, there is the description of design and implementation of the application for mobile phones, which encrypts and signs SMS using an asymmetric RSA cipher. At the end, there are described attacks on secured SMS and future extension of the application.

3. SMS Encryption using 3D-AES Block Cipher on Android Message Application.

SMS messages are one of the popular ways of communication. Sending any message is very easy. We can send and receive our confidential data at the time of transmission. During transmission of message through SMS is very difficult to protect it and also it is widely used in mobile banking. Security is the important thing in this but SMS does not provide a secure medium. SMS transmission through GSM network is also not secure, so there is need to secure SMS by providing encryption process. Encryption is important during transmission of SMS. There are so many type of encryption algorithms like AES, DES, RC4 are available. In this entire algorithm AES is most widely suitable algorithm. We develop an application which is based on Android platform which allows the user to encrypt the messages before it is transmitted over the network. The 3D-AES block cipher symmetric cryptography algorithm is used for secured transmission of message. In this system 3D AES block cipher symmetric algorithm is used for providing a secured medium by providing encryption. If message size is more than 256 bits then it required more time and size for sending that message**.**

4. SMS-A secure SMS messaging protocol for the m-payment systems.

In this paper the GSM network with the greatest worldwide number of user that provide security. The short message service (SMS) is one of its superior and well-tried services with a global availability in the GSM networks. The main contribution of this paper is to introduce a new secure application layer protocol, called SSMS, to e fficiently embedded the desired security attributes in the SMS messages to be used as a secure bearer in the m-payment systems. SSMS efficiently embeds the confidentiality, integrity, authentication, and non -repudiation in the SMS messages. It also provides an elliptic curve-based public key solution that uses public keys for the secret key establishment of a symmetric encryption and the attributes of public verification and forward secrecy. It efficiently makes the SMS messaging suitable for the m-payment applications where the security is the great concern.

5. **Building secure user-to user messaging in mobile telecommunication networks.**

In this paper author explained that Short Message Service (SMS) and Multimedia Message Service (MMS) are popularly used and will be more popular in the future. However, the security of SMS (Short message service) and MMS (Multimedia message service) messages is still a problem. There is no end -to-end security (including integrity, confidentiality, authentication, and non - repudiation) in these services. This hinders service providers to provide some services that require communication of high -level security. There have been some solutions proposed for this

6. Mobile sms banking security using elliptic curve Cryptosystem.

In this paper Mobile devices have many differences in their capabilities, computational powers and security requirements. Mobile devices can be used as the enabling technology for accessing Internet based services, as well as for personal communication needs in networking environments. Mobile services are spread throughout the wireless network and are one of the crucial components needed for various applications and services. However, the security of mobile communication has topped the list of concerns for mobile phone users. Confidentiality, Authentication, Integrity and Non-repudiation are required security services for mobile communication. Currently available network security mechanisms are inadequate; hence there is a greater demand to provide a more flexible, reconfigurable, and scalable security mechanism. This project provides effective security solution using Public key cryptography. The implementation of this project is divided into two parts first, design of API for ECC (Elliptic Curve Cryptography) which generates shared secret key required for secure communication and secondly, a web service is created which distributes this key to validate mobile user.

7. Code Tracker

SMS authorization codes [20] play an important role in the application ecosystem, as a number of transactions (e.g., personal identification and online banking) require users to provide a code for authorization purposes. However,

authorization codes in SMS messages can be stolen and forwarded by attackers, which introduces serious security concerns. CodeTracker [20], a lightweight approach to track and protect SMS authorization codes. Specifically, leverage the taint tracking technique to mark the authorization code with taint tags at the origin of the incoming SMS messages (taint sources), and then, propagate the tags in the system. To this end, modify the related array structure, array operations, string operations, IPC mechanism, and file operations for secondary storage of SMS authorization codes to ensure that the taint tags cannot be removed. When the authorization code is sent out via either SMS messages or network connections (taint sinks), we extract the taint tag of the data and enforce pre-defined security policies to prevent the code from being leaked. It is a prototype on Android's ART virtual machine and used 1, 218 SMS-stealing Android malware samples to evaluate the system. The evaluation results show that CodeTracker can effectively track and protect SMS authorization codes with a small performance overhead.

IV. PROPOSED SYSTEM

This is an advanced Encryption and decryption System targeting the SMS for Android Users both go and from. The User can send an Encrypted message while he can decrypt an encrypted message. The System makes use of the SMS that you see in the inbox, but this system filters out the one which are encrypted and shows it in their Personal Inbox in the Application. The Shared private Key is already defined in the application and one has not to insert anything but the lock pattern generates the key, which is by default encrypted in the message. So whenever the user is sending a message he should know the receiver's key also appended to the message so that while the receiver logins to the system the message is already decrypted if he is the desired recipient. The key is Auto generated and cannot be changed but for the users ease the system allows the user to save the recipient's key in a separate column. The Lock pattern is necessary. This System makes use of AES Encryption Algorithm [8], [9] to encrypt and decrypt the messages and RSA algorithm to encrypt and decrypt keys[11].

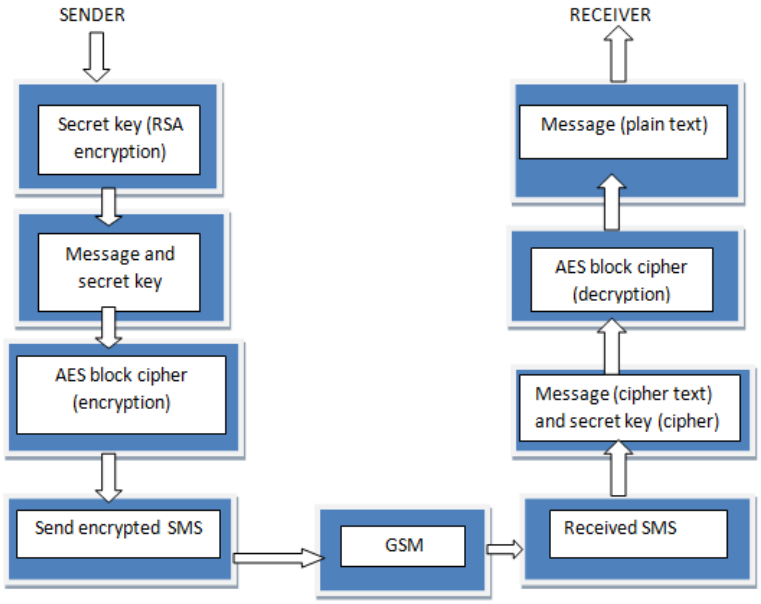

Figure 2: System architecture

1. **Key generation** The user has to set a lock pattern to generate the unique key.
2. **Sender:** The user can send messages which will be encrypted once he sends it, here the user should add mobile number and the public RSA key.
3. **Encryption:** Encryption is a translation of data into a secret code. Encryption is the most effective way to achieve data security. To read an encrypted file, you must have access to a secret key or password that enables you to decrypt it. Unencrypted data is called plain text ; encrypted data is referred to as cipher text.
4. **Decryption:** It is the reverse process of encryption which finally gives user back the original plain text. In the process of decryption, the decryption algorithm uses a private key (in public-key encryption infrastructure) or a secret key (in private-key encryption infrastructure) translating the data from cipher text into human readable plain text.

A. IMPLEMENTATION

1. Key Generation

Public class **SecureRandom** extends Random. This class provides a cryptographically strong random number generator (RNG). A cryptographically strong random number minimally complies with the statistical random number generator tests specified in FIPS 140-2, Security Requirements for Cryptographic Modules[12],[13]. Additionally, SecureRandom must produce non-deterministic output. Therefore any seed material passed to a SecureRandom object must be unpredictable, and all SecureRandom output sequences must be cryptographically strong, as described in RFC 1750: Randomness Recommendations for Security.

Many SecureRandom implementations are in the form of a pseudo-random number generator (PRNG), which means they use a deterministic algorithm to produce a pseudo-random sequence from a true random seed. Other implementations may produce true random numbers, and yet others may use a combination of both techniques.

2. Padding scheme

If for a block cipher you're not going to use a Cipher transformation that includes a padding scheme, you need to have the number of bytes in the plaintext be an integral multiple of the block size of the cipher.So either pad out your plaintext to a multiple of 16 bytes (which is the AES block size), or specify a padding scheme when you create your objects. Unless you have a good reason not to, use a padding scheme that's already part of the JCE implementation.

3. Encoding Of Keys

Public class **PKCS8EncodedKeySpec** extends EncodedKeySpec. This class represents the ASN.1 encoding of a private key, encoded according to the ASN.1 type PrivateKeyInfo.

The PrivateKeyInfo syntax is defined in the PKCS#8 standard as follows:

PrivateKeyInfo ::= SEQUENCE {

version Version,

privateKeyAlgorithm PrivateKeyAlgorithmIdentifier,

privateKey PrivateKey,

attributes [0] IMPLICIT Attributes OPTIONAL }

Version ::= INTEGER

PrivateKeyAlgorithmIdentifier ::= AlgorithmIdentifier

PrivateKey ::= OCTET STRING

Attributes ::= SET OF Attribute

Public class **X509EncodedKeySpec** extends EncodedKeySpec. This class represents the ASN.1 encoding of a public key, encoded according to the ASN.1 type SubjectPublicKeyInfo. The SubjectPublicKeyInfo syntax is defined in the X.509 standard as follows:

SubjectPublicKeyInfo ::= SEQUENCE {

algorithm AlgorithmIdentifier,

subjectPublicKey BIT STRING }

X509EncodedKeySpec this class is designed to convert between the SubjectPublicKeyInfo ASN.1 struct that is in the X.509 standard and Java public key formats.

4. Initializing a Cipher

Initializing a Cipher is done by calling its init() method. The init() method takes two parameters:

- Encryption / decryption cipher operation mode.
- Encryption / decryption key.

Cipher.init(Cipher.ENCRYPT_MODE,key);

Cipher.init(Cipher.DECRYPT_MODE,key);

5. Encrypting and Decrypting Data

In order encrypt or decrypt data with a Cipher instance you call one of these two methods:

- update()
- doFinal()

There are several overridden versions of both update() and doFinal() which takes different parameters. I will cover the most commonly used versions here.If you have to encrypt or decrypt a single block of data, just call the doFinal() with the data to encrypt or decrypt. Here is an encryption example:

byte[] plain Text=”abcdefghijklmnopqrstuvwxyz”.getBytes(“UTF-8”);

byte[] cipherText=cipher.doFinal(plain Text);

The code actually looks pretty much the same in case of decrypting data. Just keep in mind that theCipher instance must be initialized into decryption mode. Here is how decrypting a single block of cipher text looks:

byte[] cipherText=cipher.doFinal(plain Text);

B. SMS Broadcasting

In Android, when receiving a text message, the system sends the message from the RIL (Radio Layer Interface) layer to the framework layer. The framework layer then packs the text message into an SMSPDU and sends a broadcast indicating the receiving of an SMS message. All apps with the RECEIVE_SMS permission will receive the broadcast along with the SMS message if they have registered the SMS_RECEIVED_ACTION action. Before Android version 4.4, SMS broadcasting was ordered, and apps with higher priority (declared by apps in the manifest file) could access SMS messages first and then discard the messages, which make apps with low priority unreachable to the SMS messages. This mechanism has been abused by malware to intercept SMS messages. In addition, if a malicious app has the permissions (READ_SMS or WRITE_SMS) to directly operate on the SMS database, it could monitor the database continuously. Once an SMS authorization code is received, it could steal the code and then delete it. Starting with Android version 4.4, the SMS system has been changed. When the system receives a text message, the framework layer encapsulates the text message into an SMS PDU and sends it with two types of broadcasting. One type is ordered broadcasting, i.e., SMS_DELIVER_ACTION, in which only the default SMS app can receive it. In other words, only the default SMS app has the permission to delete and insert the text messages to the SMS database. The other type is unordered broadcasting, i.e., SMS_RECEIVED_ACTION, in which the broadcasting cannot be interrupted, and all apps can receive SMS messages by registering the

broadcasting. Due to this difference, malicious apps cannot intercept and delete the received SMS messages, but they still can steal and forward the SMS messages to remote servers [18].

Broadcast receiver is another basic component of Android application. It is use to receive the broadcast announcements and react according to the arise situation. Normally system makes broadcast announcements, such as time zone has changed, Picture has been captured, language preferences have changed, or battery power is low. Same as system code, application can also initiate broadcast and broadcast receivers (contained by applications) may react to it, if needed so. Suppose, browser generate broadcast announcement that the requested download has been complete and it's now available for other applications to use. Application can have more than one broadcast receivers to receive multiple broadcast announcements simultaneously.

All broadcast receivers extend the Broadcast Receiver base class. Unlike service component the broadcast receiver doesn't possess any sort of user interface rather it may start a new activity in response to the message it receives or it may alert the user through Notification Manager. Notification Manager allows application to notify users about events that take place. The notification can be inform of vibration, flashing LEDs of mobile device, playing specific sound against specific event, or could be an icon (persistently displayed into the status bar).

V. CONCLUSION

SMS theft is a cyber-crime and can lead to serious consequences. Although, people now-a-days don’t use SMS for general communication yet all the important transactional and verification messages are communicated through SMS only. Many people have worked to eradicate it, but the open source nature of android operating system has always have been a challenge. The application of SMS Encryption of AES block cipher and RSA algorithm on android application has been designed and implemented. The application is running in the mobile phone and does not require any additional encryption devices. The result showed that suitable and easy to implement in mobile device for the proposed scheme. With the increasing use of SMS for communication and information exchange, care should be taken when sensitive information is transmitted using SMS. Users should be aware that SMS messages might be subject to interception. Solutions such as encrypted SMS should be considered if there is a need to send sensitive information via SMS.

SMS is a simple, straightforward and easy to use. Where access control plays an important role. Thus, our application can be used to authenticate the sender o f message. The most important is the security of the encrypted data against various attacks. Hence this application can be used for secure transfer of data without any corrupted data segment.

REFERENCES

[1] 1S. Doyle, "Using short message service as a marketing tool", Journal of Database Marketing, vol. 8, no3, 2001, pp. 273-277.
[2] H. Harb, H. Farahat, M. Ezz, "Secure SMSPay: secure SMS mobile payment model", 2nd International Conference on Anti-counterfeiting, Security and Identification, ASID. Guiyang, China, 2008, pp. 11- 17.
[3] R. Soram, "Mobile sms banking security using elliptic curve cryptosystem", International Journal of Computer Science and Network Security, vol. 9, no. 6, pp. 30-38.
[4] M. A. Hossain, S. Jahan, M. M. Hussain, M.R. Amin, and S.H. S Newaz, "A proposal for enhancing the security system of short message services in GSM", 2nd International Conference on Anti-counterfeiting, Security and Identification, ASID, Guiyang, China, 2008, pp. 235- 240.
[5] P. H. Kuaté, J. L. Lo and J. Bishop, "Secure asynchronous communication for mobile devices", Proceedings of the Warm Up Workshop for ACM/IEEE ICSE 2010,Cape Town, South Africa, 2009, pp. 5 – 8.
[6] J. J. Garza-Saldana and A. Diaz-Perez, "State of security for SMS on mobile devices", Proceedings of the Electronics, Robotics and Automotive Mechanics Conference, 2008, pp. 110 – 115.
[7] S. Ariffin, R. Mahmod, A. Jaafar and M.R.K. Ariffin, "Byte Permutations in Block Cipher Based on Immune Systems", International Conference on Software Technology and Engineering, 3rd (ICSTE 2011). ASME Press, New York, NY., 2011.
[8] NIST, "Fips197: Advanced Encryption Standard (AES)", FIPS PUB 197 Federal Information Processing Standard Publication 197, Technical report, National Institute of Standards and Technology, 2001.
[9] J. Daemen, V. Rijmen, V., "The Design of Rijndael, AES - The Advanced Encryption Standard", Springer-Verlag, 2002.
[10] S. Ariffin, R. Mahmod, A. Jaafar and M.R.K. Ariffin, "An immune system-inspired byte permutation function to improve confusion performance of round transformation in symmetric encryption scheme", Computer Science and Applications, Lecture Notes in Electrical Engineering, Springer, 2012.
[11] S. Ariffin, R. Mahmod, A. Jaafar and M.R.K. Ariffin, "Symmetric Encryption Algorithm Inspired by Randomness and Non-linearity of Immune Systems", International Journal of Natural Computing Research, IGI Global Publishing, 2012.
[12] D. Lisonek and M. Drahansky, "SMS encryption for mobile communication", International Conference on Security Technology, Hainan Island, 2008, pp 198 – 201.
[13] S. Redl, M. W. Oliphant, M. K. Weber, and M. K. Weber, "An Introduction to GSM", 1st ed. Norwood, MA, USA: Artech House, Inc., 1995.
[14]"Short Message Service Security on Febuary 2008", available http://www.infosec.gov.hk/english/technical/files/short.pdf dated on August 2013.
[15] B. Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C", John Wiley &Sons, Inc., New York, NY, USA, 2nd edition, 1995.
[16] W. Stallings, "Cryptography and network security", Prentice Hall, New Jersey, United State, 2006.
[17] D. Lisonek and M. Drahansky, "SMS encryption for mobile communication", International Conference on Security Technology, Hainan Island, 2008, pp 198 – 201.
[18] J. P. Albuja and E. V. Carrera, "Trusted SMS communication on mobile devices", 11th Brazilian Workshop on Real-Time and Embedded Systems, Pernambuco, Brazil, 2009, pp.165- 170.
[19] M. Toorani and A.A.B. Shirazi, "SSMS-A secure SMS messaging protocol for the m-payment systems", Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC'08), Marrakech,
[20] "CodeTracker: A Lightweight Approach to Track and Protect Authorization Codes in SMS Messages" IEEE, vol 6,2018.

