IPHV8I2018X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

PROPOSED TERMINAL DEVICE FOR END-TO-END SECURE SMS IN CELLULAR NETWORKS

[#1]ZIA ARSHI, M.Tech student,
[#2]D.KUMAR SWAMY, Associate Professor & HOD,
Dept of ECE,
SAHAJA INSTITUTE OF TECHNOLOGY & SCIENCES FOR WOMEN, KARIMNAGAR, TS, INDIA.

ABSTRACT: Nowadays, SMS is very popular mobile service and even poor, illiterate, and rural areas living people use SMS service very efficiently. Although many mobile operators have already started 3G and 4G services but 2G services are still be used by the people in many countries. In 2G (GSM), only encryption provided is between the MS and the BTS, there is no end-to-end encryption available. Sometimes we all need to send some confidential messages to other people containing bank account numbers, passwords, financial details, etc. Normally, a message is sent in plain text only to the recipient and it is not an acceptable standard for transmitting such important and confidential information. Authors propose an end-to-end encryption approach by proposing a terminal for sending/receiving a secure message. An asymmetric key exchange algorithm is used in order to transmit secret shared key securely to the recipient. The proposed approach with terminal device provides authentication, confidentiality, Integrity and nonrepudiation.

Keywords: *SMS, GSM, GSM Terminal, encryption, nonrepudiation.*

I. INTRODUCTION

SMS messages are currently one of the most widespread forms of communication. It is a store-and-forward, easy to use, popular, and low cost service. There are many unusual or strange applications, such as devices which allow the switching on and off of house heating systems using an SMS [1], requests for public transport service in Asia [2], and payment applications which have been widely accepted in Europe and Asia [3], reminder for tuberculosis medication [4] and a general health care reminder system [5], and in selling theatre tickets [6]. SMS enables the transmission of up to 1120 bits alphanumeric messages between mobile phones and external systems. It uses SMS center for its routing operation in one network and can be transmitted into another network through the SMS gateway [7]. SMS usage is threatened with security concerns [8], such as eavesdropping, interception and modification. SMS messages are transmitted as plaintext between the mobile stations and the SMS center using the wireless network. SMS content are stored in the systems of the network operators and can easily be read by their personnel.

The A5 algorithm, which is the GSM standard for encrypting transmitted data, can easily be compromised. SMS tapping from radio broadcast, when SMS is sent or received from a mobile phone to base transceiver station (BTS), is not easy. When a user is roaming, the SMS content passes through different networks and perhaps the Internet that exposes it to various vulnerabilities and attacks. To exploit the popularity of SMS as a serious business bearer protocol, it is necessary to enhance its functionalities to offer the secured transaction capability. Data confidentiality, integrity, authentication, and non-repudiation are the most important security services in the security criteria that should be taken into account in many secure applications. However, such requirements are not provided by the traditional SMS messaging. The SMS GWMS (SMS gateway MSC) is a gateway MSC that can also receive short messages. The gateway MSC is a mobile network's point of contact with other networks. On receiving the short message from the short message center, GMSC uses the SS7 network to interrogate the current position of the mobile station form the HLR, the home location register.

HLR is the main database in a mobile network. It holds information of the subscription profile of the mobile and also about the routing information for the subscriber, i.e. the area (covered by a MSC) where the mobile is currently situated. The GMSC is thus able to pass on the message to the correct MSC. MSC (Mobile Switching Center) is the entity in a GSM network which does the job of switching connections between mobile stations or between mobile stations and the fixed network. A VLR (Visitor Location Register) corresponds to each MSC and contains temporary information about the mobile, information like mobile identification and the cell (or a group of cells) where the mobile is currently situated. Using information from the VLR the MSC is able to switch the information (short message) to the corresponding BSS (Base Station System, BSC + BTSs), which transmits the short message to the mobile. The BSS consists of transceivers, which send and receive information over the air interface, to and from the mobile station. This information is passed over the signaling channels so the mobile can receive messages even if a voice or data call is going on.

IPHV8I2018X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

II. EXISTING CELLULAR ARCHITECTURE

This section starts with the basic terminology of GSM network. Fig. 1 represents the basic architecture of GSM technology. The Base Transceiver Station (BTS) translates the radio signals into digital format and transfers it to the Base Station Controller (BSC). BSC controls multiple BTSs within a small geographical area. The BSC forwards the received signals to Mobile Switching Centre (MSC). MSC interrogates its databases (Home and Visitor Location Register (HLR and VLR)) for the location information about the destination mobile handset. If the signal originates or terminates at the fixed telephone line network then the signal will be routed from the MSC to the SMS Gateway MSC (SMS GMSC). If the received signal is an SMS then the message would be stored in the Short Message Service Centre (SMSC) and the message will wait to be delivered. Even after the SMS is delivered, the message content still maintains in the SMSC persistence database. If the signal needs to be redirected internationally then the signal will be routed via the International Switching Centre (ISC) to another country. The maintenance is controlled by the Operation and Management Centre (OMC). The Equipment Identity Register (EIR) and Authentication Register (AUC) databases are used for equipment verifications and user authentication. The security mechanisms (for voice and data communication) of GSM are implemented in three different system elements; the Subscriber Identity Module (SIM), the GSM handset or MS, and the GSM network. Main focus of this paper is to provide end-to-end security of message during communication. During this process, authors also focus on its key management schemes and encryption approach used. The implications of doing end-to-end encryption are to provide encryption security between sender and receiver. Currently, there is no such kind of complete security solution exists; only the airway security provided is between the MS and the BTS. The message goes as plaintext from BTS to SMSC in GSM network which can result in message content disclosure by operator and, various threats and attacks by intruders on transmitted data from the MS.

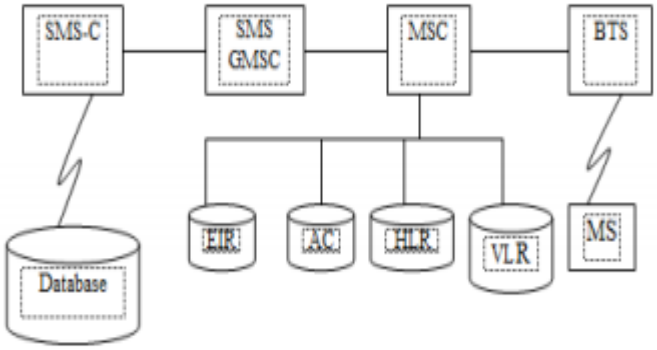

Fig. 1. GSM Architecture

III. PROPOSED SECURE APPROACH

Many security attacks exist on the SMS like man-inmiddle, reply, non-repudiation, and message disclosure. The proposed approach provides authentication, confidentiality, integrity and non-repudiation to the transmitted message. We recommend the encryption algorithms to be stored onto the SIM. Adding extra security means increasing more cost and for this reason authors also propose to include one more service as 'Secure Message' in the menu of mobile software developed by various mobile companies as shown in Fig. 2. The mobile operators can add some extra charge to send these SMS(s) to their customers. First, user as well as network authentication (mutual) takes place similar to described in [22], unlike in the existing GSM network where only unidirectional authentication is provided. Whenever a user wants to send a Secure Message to the other user, first the key management algorithm execute which generates a secret shared key and then encryption of the message takes place with symmetric cryptography.

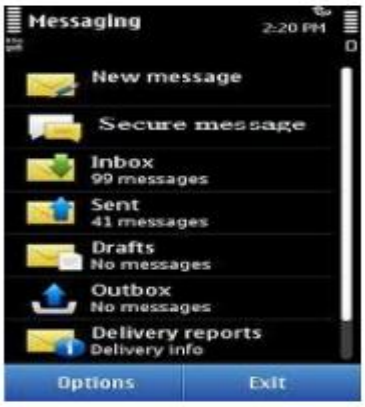

Fig. 2. Secure Message added in Menu

A. Key Management Process

Key management is handled with Diffie-Hellman key and Elliptic Curve Diffie-Hellman key exchange algorithms. Diffie-Hellman Key Exchange: It is assumed that the global public elements, a prime no 'q' and one of its primitive root 'a' (where a< q and x2 < q. Next, both MS calculate public keys y1 and y2 respectively: q mod a y 1 x =1 , and q moda y x2 .=2 Both MS exchange their public keys y1 and y2 to each other as shown in Fig. 3, but exchange of data in an insecure medium is a challenge. Now, the secret shared key 'k' can be generated at MS1 and MS2 as K q mod y x1 2= , and K q mod y x2 1= . This secret key 'k' is used to encrypt and decrypt message between Sender MS and receiver MS. But this approach is infected with man-in-middle attack.

Fig. 3. Public keys Exchange

IPHV8I2018X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .II
ISSN Online: 2319 – 9253
Print: 2319 – 9245

As a solution to this, each party signs its own DiffieHellman (DH) value to prevent man-in-middle attack (and the peer's DH value as a freshness guarantee against replay attack). This process can be found in Fig. 4. MS2 concatenates the pair (y1, y2), signs them using digital signature of MS2, and then encrypts them with 'k' and then sends the cipher text along with y2 to MS1. MS1 decrypts and verifies MS2's signature. Similarly, MS1 concatenates the pair (y2, y1), signs them using the digital signature of MS1, and then encrypts them with 'k' and then sends the cipher text to MS2. MS2 decrypts and verifies MS1's signature. MS1 and MS2 are now mutually authenticated and have a shared secret key 'k'. Here, we are restricted to discuss the different approaches of digital signatures and their key management and this discussion is out of the scope of this paper. But, this approach is also infected with Identity Misbinding Attack. Let's consider a situation like in Fig. 5 and assume an attacker as 'A'. Here A doesn't know 'k' but MS2 considers anything sent by MS1 as coming from attacker 'A'. One solution to this problem may be like in Fig. 6. Here, digital signature is created using (y1, y2 and MS identity). Digital signature is generated with MS's private key and verified with MS's public key. Attacker 'A' can generate neither SIG (y1, y2, MS1) nor SIG (y2, y1, MS2) which can only be generated by MS1 and MS2 respectively. But one possibility may be to replace the value of y1 by attacker 'A'. This process can generate the integrity issues. To maintain the integrity in the communication between MS1 and MS2, a hash function is used to create message digest of the sending data y1 and (y1, y2) as H(y1) and H(y1, y2) respectively. MD5 or SHA1 can be used as an efficient hash function. This process as shown in Fig. 7 and is secured enough for key management.

Fig. 4. Added Signature to DH

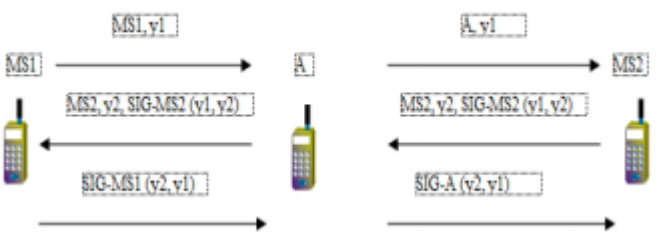

Fig. 5. Identity Misbinding Attack by Attacker 'A'

Fig. 6. Solution to Identity Misbinding Attack

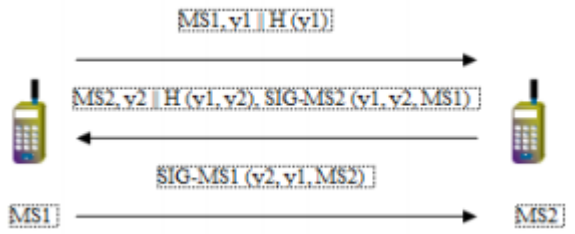

Fig. 7. Secure Data Exchange in DH with Integrity

Elliptic Curve Diffie–Hellman Key Exchange: Another approach for key exchange can be done using Elliptic Curve Diffie–Hellman protocol (ECDH) that allows two parties, each having an elliptic curve public-private key pair, to establish a shared secret over an insecure channel and this shared secret may be directly used as a key. Each party must have a key pair suitable for elliptic curve cryptography, consisting of private key 'd' which is a randomly selected integer in the interval [1, n-1] and a public key 'Q' where Q = d*G. Now, both MS will generate a new secret private keys from their Kc (as Kc is stored in SIM and was used in authentication process) and let's consider them as d1 and d2 respectively where d1

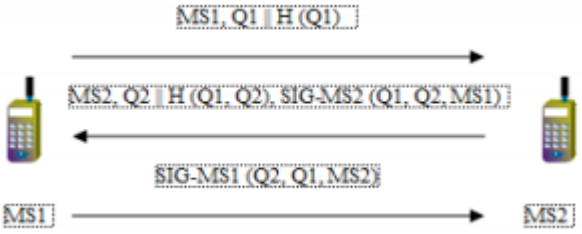

Fig. 8. Secure Data Exchange in ECDH with Integrity

No party other than MS1 can determine MS1's private key, unless that party can solve the elliptic curve Discrete Logarithm problem and MS2's private key is similarly secure. No party other than MS1 or MS2 can compute the shared secret, unless that party can solve the Elliptic Curve DiffieHellman problem.

B. Symmetric Encryption Approach and Experimental Setup

J2ME the WMA (Wireless Messaging API) provides tools for sending and receiving SMS messages. Our solution based on symmetric cryptography (DES, Triple DES, and AES) is simulated with Java MIDlet, which is an application written in Java for the Micro Edition platform. The application can send and receive SMS messages in binary format using the WMA. Since the J2ME environment does not contain cryptographic algorithms, we use the

Lightweight API from the Legion of the Bouncy Castle. Symmetric algorithms DES, TripleDES with 2 keys, TripleDES with 3 keys, and AES have implemented. The standard key size for DES, TripleDES with 2 keys, TripleDES with 3 keys and AES are 64 bits (out of which 56 bits are used), 112 bits, 168 bits and 128 bits respectively. Fig. 9 and Fig. 10 show the results observed for encryption and decryption with DES, TripleDES with 2 keys, and TripleDES with 3 keys, and AES. The results conclude that out of these algorithms AES takes almost minimum time to encrypt and decrypt the SMS with various sizes where one SMS size is 160 characters. As the input of 160 characters each, the algorithms DES, AES, TripleDES2K and TripleDES3K generates 143, 80, 160 and 168 characters cipher respectively. These results can be found in table I. This shows that AES is the best option for this purpose. The results are calculated on 30 times repeat the execution of each of these algorithms. We have also calculated the range of confidence interval (CI), considering it 95% for each algorithm with 160 characters as input because the reported margin of error is typically about twice the standard deviation – the radius of a 95% confidence interval [18] similar to described in [23].

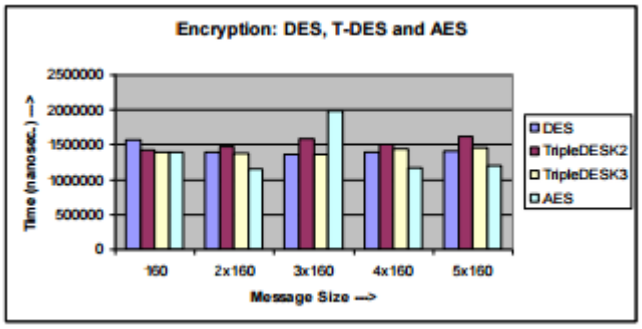

Fig. 9. Encryption using DES, T-DES, and AES

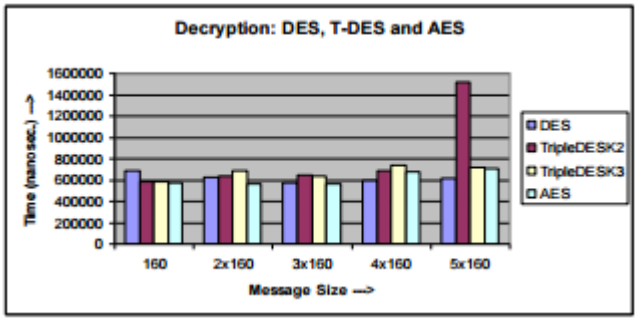

Fig. 10. Decryption using DES, T-DES, and AES

TABLE 1. SMS SIZE PAIR (ORIGINAL, CIPHER) IN VARIOUS ALGORITHMS

DES	AES	TripleDES2K	TripleDES3K
160, 143	160, 80	160, 160	160, 168

Table II and table III represents the results of confidence interval for both encryption and decryption of the message (SMS) for 160, 160x2, 160x3, 160x4, and 160x5 characters in length for DES, TripleDES2K, tripleDES3K and AES algorithms. Here, confidence interval is measured in nanoseconds. We use t-distribution to calculate all these parameters. In this whole process, the SMS size from 160 characters to 800 characters is evaluated where more than 160 characters in a SMS needs to be break and concatenated with another SMS. A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data points are spread out over a large range of values. Thus, AES is strict to its output range and is considered best among them.

TABLE II. CONFIDENCE INTERVAL FOR SMS ENCRYPTION

Para-meters	CI-160 char	CI-160x2 char	CI-160x3 char	CI-160x4 char	CI-160x5 char
DES	615156 to 2525227	444939 to 2337297	397177 to 2334366	418794 to 2385118	430900 to 2396306
T-DESK2	423715 to 2437266	384923 to 2548397	480270 to 2696873	448570 to 2578392	533472 to 2694613
T-DESK3	294704 to 2301896	337878 to 2407877	332541 to 2398940	346447 to 2530239	371517 to 2538104
AES	775058 to 2017472	515608 to 1802398	14850 to 4573891	518174 to 1809940	549579 to 1850251

TABLE III. CONFIDENCE INTERVAL FOR SMS DECRYPTION

Para-meters	CI-160 char	CI-160x2 char	CI-160x3 char	CI-160x4 char	CI-160x5 char
DES	629011 to 745390	494369 to 750842	521460 to 617265	539561 to 658254	562070 to 666845
T-DESK2	520563 to 650413	546387 to 716769	582147 to 719667	623106 to 751108	128407 to 3157173
T-DESK3	520276 to 645226	610201 to 772473	577035 to 688952	625512 to 851052	637007 to 804383
AES	548127 to 602172	526055 to 603419	512970 to 619404	667586 to 704277	678119 to 732740

C. Discussion

Since, we propose Diffie-Hellman and Elliptic Curve Diffie-Hellman protocol for key exchange, so it's necessary to focus on their security aspects. Authors claim that the proposed algorithm for key exchange is secure because the public key exchange also provides integrity and non-repudiation by including a hash function and digital signature respectively. ECDH is based on elliptic curve, thus it is more secure as every multiplication is done by multiple additions and it is infeasible to construct the original one by any reversible process. It has proved that out of the implemented algorithms DES, TripleDES with 2 keys, TripleDES with 3 keys and AES, the AES algorithm is best suitable for this application. Various attacks have been found on DES and Triple DES including full attack. But no full attack has been found on AES. This algorithm can be used for encryption and decryption process in transmitting secure message.

IV. PROPOSED GSM TERMINAL FOR SECURE SMS

In this section, we propose a terminal through which a secure SMS can be sent or received. Our proposed Terminal for sending/receiving SMS in GSM network is similar to the M20 terminal [19]. We have incorporated the security aspects as well in the proposed terminal and modified it in such a way that the produced overheads can be minimized. The proposed terminal provides authentication, confidentiality, integrity and non-repudiation services to the transmitted message. For the real GSM network, we propose the cryptographic algorithms should be implemented on the SIM card itself at the time of manufacturing. In fact, we can have a separate SIM card for any security related communication and transaction as the Koreans (in Korea) have separate SIM cards for financial transactions. We consider our proposed approach in this paper as a part of GSM terminal which provides authentication, confidentiality and integrity services to the end user. The nonrepudiation service can be provided by (DSA/ECDSA) or (DSA/Variant of ECDSA) digital signature algorithms. These algorithms can be directly stored onto the SIM (for without the use of GSM terminal) or on the terminal device. The digital signature is imposed over the encrypted message; the details of DSA/ECDSA/Variant of ECDSA can be referred in [20]. A proposed GSM terminal can provide the services to both SMS modes: SMS Deliver (Mobile Terminated) as well as SMS Submit (Mobile Originator).

A. Design of Proposed Terminal

This subsection describes the architecture of the proposed GSM terminal as shown in Fig. 11 which provides the authentication, confidentiality, integrity and non-repudiation services in order to secure the transmitted SMS in the network. The maximum data can be occupied in 140 Octets i.e. 1120 bits which means 160 English characters can be written in a single SMS as English characters are encoded with 7-bits encoding scheme (ASCII code, 160*7=1120 bits). In this proposed terminal we have mainly included a bit to check whether encryption is ON/OFF, one bit is to set the ciphering algorithm AES/MAES, one bit is used for the algorithm to maintain data integrity SHA1/HMAC, and one bit is used to set the digital signature algorithm DSA/ECDSA or DSA/Variant ECDSA. Various parameters of the proposed GSM Terminal in both the modes can be understood as follows:

B. Hardware Requirements and Setup

This subsection discusses about the hardware requirements and setup phase for the proposed GSM terminal. The following are the hardware requirements to simulate and test our proposed approach in GSM environment: (1) Mobile Phone, (2) A Terminal (proposed in this paper), (3) Two SIM Cards (One for Mobile Phone and the other for proposed Terminal), (4) GSM Antenna, (5) A power cable for proposed Terminal, (6) RS-232 cable, (7) A PC running on Windows Terminal or Hyper Terminal. Now, here various steps to setup the hardware are stated below: (1) The first thing is to make ready mobile phone with a SIM Card, (2) Terminal Setup Preparation: Run Connect the proposed◊Windows Terminal or Hyper terminal Insert SIM into the◊Terminal to COM1 or COM2 of PC In Windows Terminal,◊proposed Terminal and turn it ON select [Communication] from [Setting] and set the proposed Terminal to the parameters as: Baud Rate: 19200 bps, Data Bits: 8, Stop Bits: 1, Parity: None, Flow Control: Hardware, Reset the proposed Terminal◊Connector: COM1 or COM2 to factory default using AT&F and hence configure the proposed Terminal for SMS using the various AT commands.

V. CONCLUSION

We conclude that the proposed approach based on ECDH is suitable for key exchange while transmitting the SMS from one mobile to another. Symmetric algorithms are faster than asymmetric algorithms, thus we implemented DES, TripleDES with 2 keys, TripleDES with 3 keys and AES. Out of these algorithms, AES is the best algorithm to provide ciphering to the SMS during transmission. Authors also proposed a GSM terminal device for providing authentication, confidentiality, integrity, and non-repudiation services with SMS.

REFERENCES

[1] Tele-Log, 2009. [Online]. http://www.tele-log.com/domotica-e.html.

[2] T. C. Lim, H. K. Garg, “Designing SMS applications for public transport service system in Singapore,” 8th Intern. Conf. on Communication Systems, vol. 2, pp. 706– 710, 2002.

[3] M. R. Hashemi, E. Soroush, “A secure m-payment protocol for mobile devices,” Canadian Conference on Electrical and Computer Engineering, pp. 294–297, 2006.

[4] D. Green, “South Africa: novel approach to improving adherence to TB treatment,” Essential Drugs Monitor, No 33, 72 page, 2003.

[5] S. Treweek, “Joining the mobile revolution,” Scandinavian Journal of Primary Health Care, 2003.

[6] Show tickets sold on mobile phone in Singapore, 2003. [Online]. http://www.m-travel.com/news/2003/07/show-ticketsso.html.

[7] G. Peersman, S. Cvetkovic, “The Global System for Mobile Communications Short Message Service,” IEEE Personal Communications, pp. 15-25, 2000.

[8] D. Lisonek and M. Drahansky, “SMS encryption for mobile communication,” International Conference on Security Technology Hainan Island, pp. 198 – 201, 2008.

[9] S. Doyle, "Using short message service as a marketing tool," Journal of Database Marketing. vol. 8, pp. 273-277, 2001.
[10] M. Toorani and A. B. Shirazi, "SSMS-A secure SMS messaging protocol for the m-payment systems," 13th IEEE Symposium on Computers and Communications, Marrakech, pp. 700-705, 2008.
[11] S. Zhao, A. Aggarwal, S. Liu, "Building secure user-to-user messaging in mobile telecommunication networks," Proceedings of Wireless Telecommunications Symposium, pp. 151-157, 2008.
[12] H. Harb, H. Farahat, M. Ezz, "SecureSMSPay: secure SMS mobile payment model," 2nd International Conference on Anti-counterfeiting, Security and Identification, pp. 11-17, 2008.
[13] 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, pp. 110 – 115, 2008.
[14] 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. Cape Town, pp. 5-8, 2009.
[15] Ruangchaijatupon, P. Krishnamurthy, "Encryption and Power Consumption in Wireless LANs-N," The Third IEEE Workshop on Wireless LANs, Newton, Massachusetts, pp. 27-28, 2001.
[16] Hardjono, Security In Wireless LANS And MANS, Artech House Publishers, 2005.
[17] G. A. Safdar, C. McGrath, M. McLoone, "Limitations of Existing Wireless Networks Authentication and Key Management Techniques for MANETs," IEEE International Symposium on Computer Networks. (ISCN'06), pp. 101-105, 2006.
[18] [Online]. http://www.stat.yale.edu/Courses/1997-98/101/confint.htm
[19] [Online]. http://www.gsmfavorites.com/documents/sms/packetformat/
[20] N. Saxena, N. S. Chaudhari, "EasySMS: A Protocol for End-to-end Secure Transmission of SMS," IEEE Transactions on IFS, Vol. 9, No. 7, pp. 1157-1168, 2014.