An Extended Approach for SMS Security using Authentication Functions

Neetesh Saxena
Deptt. of Computer Sc. & Engg.
Indian Institute of Technology
Indore, India
Neetesh.saxena@gmail.com

Narendra S. Chaudhari
Deptt. of Computer Sc. & Engg.
Indian Institute of Technology
Indore, India
narendra@iiti.ac.in

Gend Lal Prajapati
Deptt. of Computer Engg.
IET-DAVV,
Indore, India
Glprajapati1@gmail.com

***Abstract*- Nowadays, security of SMS is a crucial aspect because it plays an important role in value added services and mobile commerce. Asymmetric algorithm like Diffie-Hellman can be used to encrypt the SMS message in M-commerce or mobile banking system. We use authentication functions to maintain the integrity of data. Password key exchange protocol based on Diffie-Hellman algorithm generates a secret shared key which can be used in message encryption and in MAC function. MAC (message authentication code) or hash functions are used maintain the integrity of message and can be used with the encryption. These functions also act as an error detecting code or checksum. This paper discusses the comparative analysis of both the authentication functions separately for password key exchange protocol by analyzing some of the security issues. The discussion of this paper concludes that MAC functions are more secure than hash function, but having greater complexity and take more to execute. So, it's better to use hash function for maintaining the integrity of message over a network where the transmitted amount of message is very small (SMS). Here, digital signature is generated with RSA to show the functionality of MD5 and SHA1, which prevents SMS from message modification and non-repudiation attack.**

Keywords- GSM; SMS security; authentication function; public key cryptography

I. INTRODUCTION

Asymmetric or public key cryptography technique uses the concept of public-private keys that can be used for transmitting the message over a network channel. The basic aim of public-key cryptography is to provide encryption approach to minimize an attempt to attack on the two functionalities of the symmetric technique: key distribution and digital signature [1]. Diffie and Hellman achieved a solution that addressed both problems. The use of cryptographic hash functions like MD5 or SHA for message authentication has become a standard approach in many Internet applications and protocols [2][3].Though very easy to implement, these mechanisms are usually based on ad hoc techniques that lack a sound security analysis.

II. PRILIMINARIES

Verifying the integrity and authenticity of information is a prime necessity in computer systems and networks [4]. In particular, two parties communicating over an insecure channel require a method by which information sent by one party can be validated as authentic by the other [5]. Before we move into the broader concept let us have a brief review of some basic facts.

A. SMS Architecture and Security in GSM

Short Message Service (SMS) has become an extension of our lives and plays an important role in daily life. SMS is a popular medium for delivering Value Added Services and are suitable for mobile banking, payment reminders, stock and news alerts, railway and flight enquiries etc [12]. These types of messages are normally computer generated messages sent over Short Message Peer to Peer (SMPP) protocol. It is a store-and-forward, easy to use, popular, and low cost service. The existing SMS is not free from the eavesdropping, but security is the main concern for any business company such as banks who will provide these mobile banking. Presently there is no such scheme which can give the complete SMS security [13].

In the GSM, only the airway traffic between the Mobile Station (MS) and the Base Transceiver Station (BTS) is optionally encrypted with a weak and broken stream cipher (A5/1 or A5/2). The authentication is unilateral and also vulnerable [15].There is some security issues related to services of SMS for such M-Commerce or mobile banking. The service includes confidentiality, integrity, non-repudiation and authentication [15]. Understanding the basics of SMS security opens the door to preventing some common security threats in SMS usage [14][16]:

1) Man-in-middle Attack: This is the network that authenticates users. The user does not authenticate network so the attacker can use a false BTS with the same mobile network code as the subscriber's legitimate network to impersonate himself and perform a man-in-the-middle attack.

2) Replay Attack: The attacker can misuse the previously exchanged messages between the subscriber and network in order to perform the replay attacks.

3) Message Disclosure: Since encryption is not applied to short message transmission by default, messages could be intercepted and snooped during transmission. In addition, SMS messages are stored as plain text by the SMSC before they are successfully delivered to the intended recipient. These messages could be viewed by operators in the SMSC who have access to the messaging system.

978-1-4577-2119-9/12/$26.00 ©2011 IEEE

4) Spamming: While using SMS as a legitimate marketing channel, many people have had the inconvenience of receiving SMS spam. The availability of bulk SMS broadcasting utilities makes it easy for virtually everyone to send out mass SMS messages.

5) Denial of Service (DoS) Attacks: DoS attacks are made possible by sending repeated messages to a target mobile phone, making the victim's mobile phone inaccessible.

6) SMS Phone Crashes: Some vulnerable mobile phones may crash if they receive a particular type of malformed short message. Once a malformed message is received, the infected phone becomes inoperable.

7) SMS Viruses: There have been no reports of viruses being attached to short messages, but as mobile phones are getting more powerful and programmable; the potential of viruses being spread through SMS is becoming greater.

8) SMS Phishing: SMS phishing is a combination of SMS and phishing. Similar to an Internet phishing attack using email, attackers are attempting to fool mobile phone users with bogus text messages.

B. Diffie-Hellman Key Exchange

Diffie and Hellman proposed and developed a public key cryptography algorithm known as Diffie-Hellman key exchange [6]. This technique allows two users to exchange a secure key over the network that can be used for the encryption of message. A brief overview of this algorithm is given here:

Global Elements

q: prime number;

n: a primitive root of p and n < q;

'User A' Key Generation

select private key a : a < q

calculate public key x: $x = n^a \bmod q$;

'User B' Key Generation

select private key b : b < q;

calculate public key y : $y = n^b \bmod q$;

Generation of Secret Key by User A

$k = y^a \bmod q$;

Generation of Secret Key by User B

$k = x^b \bmod q$;

The password key exchange protocol based on public encryption may encounter the man-in-the-middle attack because it mainly uses the Diffie-Hellman Key Exchange protocol [21]. In the original description, the Diffie-Hellman exchange by itself does not provide authentication of the communicating parties and is thus vulnerable to a man-in-the-middle attack. Thus, there is a need to manage the password key exchange protocol encryption based on authentication functions like MAC and Hash function.

C. Authentication Functions

To maintain the security essentials a message must be filtered by authentication and digital signature approaches. An authentication and digital signature mechanism has two parts: first part provides some functions for authentication (that produces an authenticator) by the sender and the second part enables receiver to verify the authenticity of the message [7][8]. These functions are:

1. Message encryption
2. Message authentication code (MAC)
3. Hash function

1. **Message encryption:** The encrypted form of message i.e. the cipher text works as its authenticator.
2. **Message authentication code (MAC):** Message is encrypted by a public function (a MAC function) and a secret key that generates a fix length code.
3. **Hash function:** Message is encrypted by a public function (form of a hash function) that converts a message of any length into the fixed length code.

Assume that:
User A
Private key: a
Public key: x

User B
Private key: b
Public key: y

(i) Encryption: confidentiality, authentication and signature
$A \rightarrow B: E_y[E_a(M)]$

(ii) MAC Encryption: confidentiality and authentication
$A \rightarrow B: E_{k2}[M] \,||\, C_{k1}[E_{k2}(M)]$
Where k1 and k2 are the two shared secret keys.

(iii) Hash Function Encryption: confidentiality and authentication
$A \rightarrow B: E_k [M \,||\, H(M) \,||\, N]$
Where k is the shared secret key and N is the shared secret value between A and B.

III. PASSWORD KEY EXCHANGE PROTOCOL BASED ON PUBLIC ENCRYPTION

Let us assume that there are two parties A and B respectively, who want to communicate with each other. They can share only the passwords of each other i.e. a calculated public key. Various steps of communication are as follows [17]:

1. Both User A and B know their private keys (a random number) 'a' and 'b' respectively.
2. User A calculates its password (public key) 'x' as $[n^a \bmod q]$.
3. User B calculates its password (public key) 'y' as $[n^b \bmod q]$.
4. User A sends its ID_A and password 'x' to User B.
5. User B sends its ID_B and password 'y' to User A.
6. User A generates a shared secret key 'k' by password of User B 'y' and its private key 'a'.
   $k = y^a \bmod q = [n^b \bmod q]^a \bmod q = n^{ab} \bmod q$
7. User B generates a shared secret key 'k' by password of User A 'x' and its private key 'b'.
   $k = x^b \bmod q = [n^a \bmod q]^b \bmod q = n^{ab} \bmod q$

For this approach, it is difficult for an attacker to guess the passwords of both User A and User B. This protocol depends upon the effectiveness of computing discrete logarithms and attacker is forced to determine the discrete logarithm to generate the key. The security of this protocol depends on the fact, that, it is relatively easy to calculate exponential modulo of a prime number while it is very difficult to calculate the discrete logarithms and for a large number it is almost infeasible to calculate discrete logarithm.

IV. PASSWORD KEY EXCHANGE PROTOCOL BASED ON MAC ENCRYPTION

Once again, let us assume that there are two shared keys k_1 and k_2 respectively that are used by both users. Various steps of communication followed in MAC encryption are as follows [17]:

1. User A and User B know their private keys (a random number) 'a' and 'b' respectively.
2. User A calculates its password (public key) 'x' as [n^a mod q] and encrypted message digest code C_k(password) using a MAC function/algorithm.
3. User B calculates its password (public key) 'y' as [n^b mod q] and C_k(password) using a MAC function/ algorithm.
4. User A sends its ID_A and E_{k2} [(password) || C_{k1}(password)] to User B.
5. User B sends its ID_B and E_{k2} [(password) || C_{k1}(password)] to User A.
6. User A generates a shared secret key 'k' by the password of User B and its private key 'a'. { Decrypt the message digest code as D_{k2} [(password) || C_{k1}(password)] to get the password and use k1 shared key to calculate C_{k1}(password) and match with the actual C_{k1}(password) send by User B to detect the error}
   $k = y^a \bmod q = [n^b \bmod q]^a \bmod q = n^{ab} \bmod q$
7. User B generates a shared secret key 'k' by the password of User A and its private key 'b'.{[use D_{k2} [(password) || C_{k1}(password)] to get the password and use k_1 shared key to calculate C_{k1}(password) and match with the actual C_{k1}(password) send by User A to detect the error}
   $k = x^b \bmod q = [n^a \bmod q]^b \bmod q = n^{ab} \bmod q$

But MAC does not provide digital signature because it uses shared keys.

V. PASSWORD KEY EXCHANGE PROTOCOL BASED ON HASH FUNCTION ENCRYPTION

This protocol uses a different approach known as hash function based approach. The basic aim of hash function is to provide authentication by generating a fixed length hash code value (like fingerprints). It is easy to calculate H(M) for a given message M [2]. Three main properties of any hash function are as follows [17]:

1. One way property: For a given value h, it is computationally infeasible to find M such that H(M) = h.

2. Weak collision resistance: For a given block M, it is computationally infeasible to find N ≠ M such that H(N) = H(M).

3. Strong collision resistance: It is computationally infeasible to find any pair (M, N) such that H(M) = H(N).

The difference of this protocol with password key exchange public encryption protocol is that, here User A and User B send message digest fixed length code H(password) instead of only password. Assume that there is a shared key k_3 used by both users. Various steps of communication by hash function encryption are as follows:

1. User A and User B know their private keys (a random number) 'a' and 'b' respectively.
2. User A calculates its password (public key) 'x' as [n^a mod q] and H(password) using a hash function.
3. User B calculates its password (public key) 'y' as [n^b mod q] and H(password) using a hash function.
4. User A sends its ID_A, E_{k3} [(password) || E_{kRa} [H(password)]] to User B where ID_A is the identification of user A and KR_a is the private key of user A in public cryptography.
5. User B sends its ID_B, E_{k3} [(password) || E_{kRb} [H(password)]] to User A where ID_B is the identification of user B and KR_b is the private key of user B in public cryptography.
6. User A generates a secret key 'k' by the password of User B and its private key 'a'. {Use D_{k3} [(password) || E_{kRa} [H(password)]] to get the password and use public key of User A 'kU_a' to decrypt the H(password). Now calculate H(password) and match with the decrypted H(password) send by User B to detect the error.}
   $k = y^a \bmod q = [n^b \bmod q]^a \bmod q = n^{ab} \bmod q$
7. User B generates a secret key 'k' by the password of User A and its private key 'b'. {Use D_{k3}[(password) || E_{kRb} [H(password)]] to get the password and use public key of User B 'kU_b' to decrypt the H(password). Now calculate H(password) and match with the decrypted H(password) send by User A to detect the error}
   $k = x^b \bmod q = [n^a \bmod q]^b \bmod q = n^{ab} \bmod q$

Note, here we are not providing confidentiality for the data to be transmitted, instead we are assuming the data must be transmit from the authentic source. Thus we have encrypted the data/message with the private keys of users as {E_{kRa} and E_{kRb}}. If we want to provide confidentiality for the data which is to be transmitted then we must encrypt the data with the public keys of users as {E_{kUa} and E_{kUb}} respectively for user A and user B while the encrypted message can be decrypted by their private keys as {D_{kRa} and D_{kRb}}.

VI. COMPARISON

Both MAC and hash values are used for maintain the integrity of message transmitted from sender to receiver over a network. It can also work as an error detecting code or checksum. The difference between a one-way hash and a MAC (Message authentication code), is that the hash verifies the uniqueness of a message and the MAC is usually an encrypted hash, also used to verify the uniqueness of a message, but which only can be verified if you know the secret key [9]. We can analyze the security of both functions by two major cryptographic terms: Brute force attack and, cryptanalysis. MAC uses a secret key to generate a MAC code but hash function doesn't use any key. In the brute force attack, all the possible combination of the key are applied. Suppose, if the key is of r-bits size then the brute force attack is 2^r. In MAC, even if an attacker gets the MAC value, he can't get the original message unless he knows the secret key. Thus we can say that, the MAC encryption is more secure than the hash encryption.

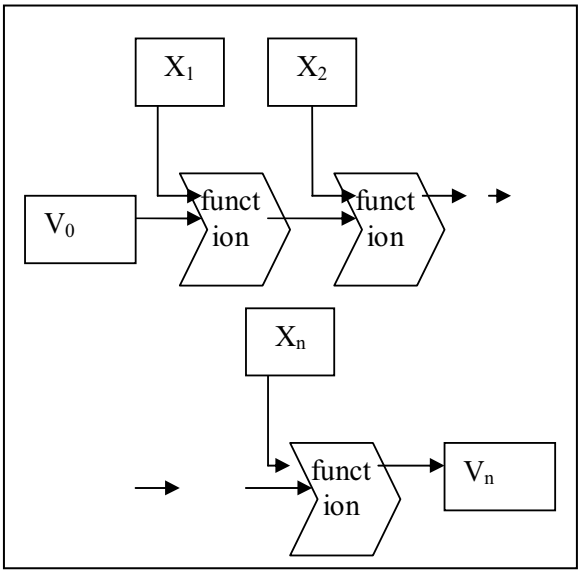

Figure 1. A simple hash function

Here, in this MAC function, AES is used as the encryption algorithm and the encryption is done by a secret key 'K'. As there is a vast variety in the structure of MAC function, so it is very difficult to apply cryptanalysis on it [4]. Further, far less work has been done on developing cryptanalysis of MAC.

Finally, we can conclude that hash is used to ensure message integrity while MAC is used for both integrity and authentic. Most MAC algorithms are a combination of hash function with a secret key, which is called as encryption of hash value with a key. This is the reasons of providing authentication for hash functions by {E_{kRa} and E_{kRb}} in the

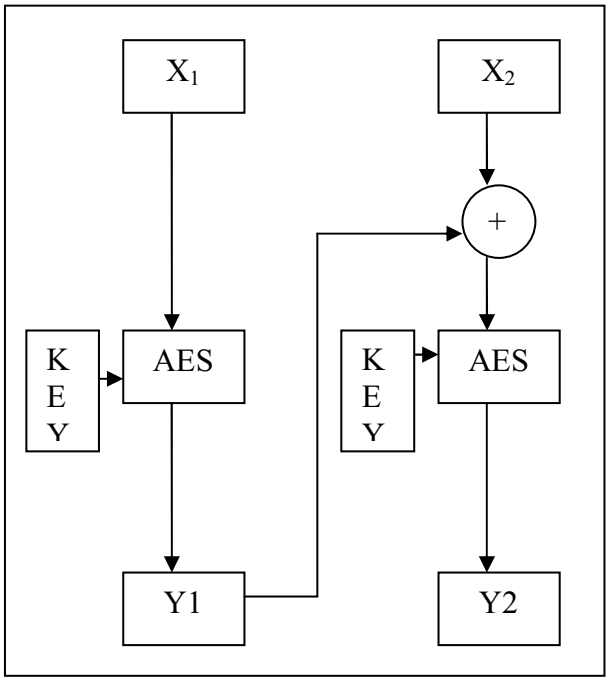

Figure 2. A simple MAC function

section V and now these hash functions can be used for small data transmission in the network.

VII. IMPLEMENTATION

Hash Functions take a block of data as input, and produce a hash or message digest as output which is a fixed length code. The usual intent is that the hash can act as a signature for the original data, without revealing its contents.

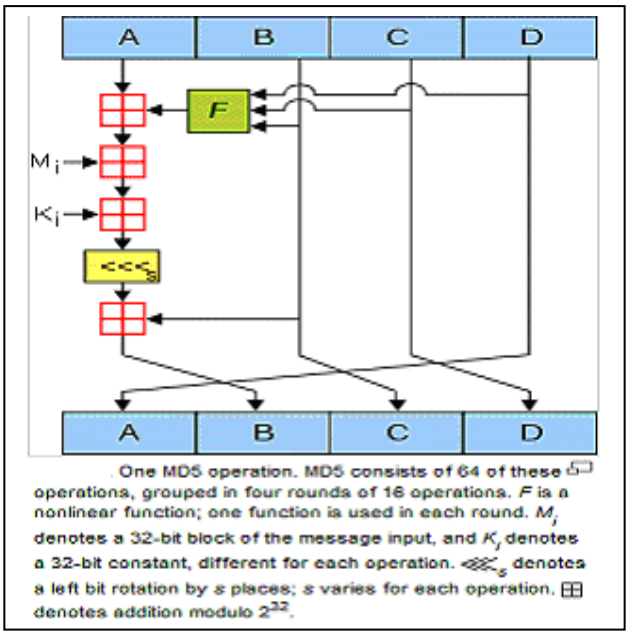

Figure 3. Basic MD5 Structure [19]

Therefore, it's important that the hash function be irreversible (one way hash function), not only should it be nearly impossible to retrieve the original data, it must also be unfeasible to construct a data block that matches some given hash value. Given the exact same input twice, the hash function should always produce the same output and even a single bit changed in the input, though, should produce a different hash value.

The hash value should be small enough to be manageable in further manipulations, yet large enough to prevent an attacker from randomly finding a block of data that produces the same hash code. These hash functions can't be used directly for encryption, but are very useful for authentication. Sender and recipient of some data message share a secret value and running it through a hash function, a signature is generated in the form of the hash value. The data message is transmitted along with the signature. The recipient combines the received message with the secret, generates a hash value or hash code, and checks to make sure it's identical to the signature. The message's authenticity is thus verified.

Here, two hash functions MD5 and SHA1 are implemented by JDK 1.6 with RSA digital signature scheme to provide authentication and avoid non-repudiation attack on SMS. Java is always considered as a good programming language and most of the mobile applications are build in Java. So, we have choosen Java as an implementation language for this work. Results of this implementation are placed in table1 and table2.

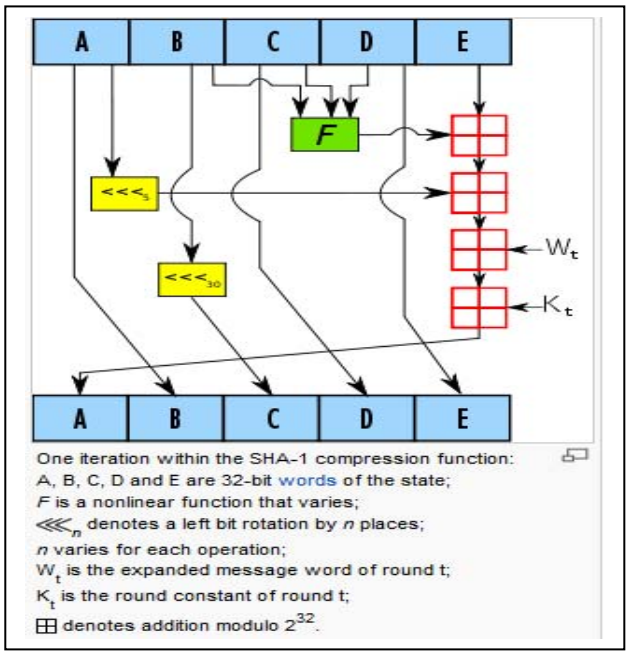

Figure 4. Basic SHA1 Structure [20]

Table1 shows the key generation process for the RSA digital signature with MD5 and with SHA1. Figure3 and figure4 shows the basic functional structure of MD5 and SHA1 respectively and it can be shown from these figures that SHA1 is more complex than MD5.

TABLE I. DIGITAL SIGNATURE RSA WITH KEY GENERATION

Key Generation (Size in bits)	RSA_MD5 (in milliseconds)	RSA_SHA1 (in milliseconds)
512	93	94
1024	218	219
2048	1123	1966

TABLE II. DIGITAL SIGNATURE RSA WITH MESSAGE DIGEST

100 characters	RSA_MD5 (in milliseconds)	RSA_SHA1 (in milliseconds)
Key Generation (1024 bits)	218	219
Encryption	31	31
Decryption	16	16
Verification	13	13

Basically, 512-bit keys no longer provide sufficient security for anything more than very short-term security needs. RSA Laboratories currently recommends key sizes of 1024 bits for corporate use and 2048 bits for extremely valuable keys like the root key pair used by a certifying authority [18]. So we consider the key size of 1024 bits as standard key size.

Table-I results 93, 218, 1123 milliseconds and 94, 219, 1966 milliseconds for RSA with MD5 and RSA with SHA1 respectively for key size 512, 1024 and 2048 bits in length. Table-II shows the key generation time, encryption time, decryption time, and verification time for RSA with MD5 and RSA with SHA1 respectively while the size of message is 100 characters. RSA with MD5 takes 218, 31, 16 and 13 milliseconds for key generation time, encryption time, decryption time and verification time respectively while RSA with SHA1 takes 219, 31, 16 and 13 milliseconds for key generation time, encryption time, decryption time and verification time respectively. It shows that for both the cases encryption, decryption and verification take same time so we can go with one which has more complex structure, result in the more difficulty to cryptanalysis by attackers.

VIII. CONCLUSION AND FUTURE WORK

Finally, it can be concluded that MAC and hash functions are more secure than the encryption based authentication function. In this paper, the comparative study states that the MAC functions are more secure than hash functions because it is more difficult to apply brute force attack on MAC functions than hash functions and also difficult to generalize the cryptanalysis of MAC functions as it has large variety of its structure but, MAC function takes more time than hash function [15]. So, a hash function is sufficient enough to maintain the integrity of message for data transmission like SMS transmission where a small amount of data is need to be transferred. It'll reduce complexity of system duing the transmission of message over a network.

Results in both the tables state that generation of key in MD5 message digest takes almost same time (with very little difference) as SHA1 but the structure of SHA1 is more complex and secure than MD5. So, for a small data like SMS transmission, SHA1 message digest (160 bits) is better than MD5 message digest (128 bits). Apart form this, various attacks have been found on MD5 in the research of previous years. A lot of work has to be done on MAC. So, it is a research - oriented area related to the cryptanalysis of MAC function's internal structure and its various varieties. Finally the paper proposed an encryption approach based on Diffie-Hellman with hash function and after that a comparative study shows that for an SMS, SHA1 is better than MD5 for a digital signature scheme.

ACKNOWLEDGMENT

Authors want to thank IIT Indore for the support in this research work.

REFERENCES

[1] Steven M. Bellovin, Michael Merritt, "Augmented Encrypted Key Exchange: A Password-Based Protocol Secure against Dictionary Attacks and Password File Compromise", 1993.

[2] Xun Yi and Kwok Yan Lam, "Hash function based on block cipher", IEE 1997 Electronics Letters Online No: I9971336.

[3] Luo Zhong, Zhao Zhongining, Zhu Chongguang, "The Unfavourable Effects of Hash Coding on CMAC Convergence and Compensatory Measure", Instilutc of Rcmotc Scnsing Applications, CAS Dept Image Processing, P 0. Box 9718 Beijing china.

[4] H. E. Michail, A. P. Kakarountas, G. Selimis, C. E. Goutis, "Throughput Optimization of the Cipher Message Authentication Code", VLSI Design Laboratory, Dpt. of Electrical & Computer Engineering, University of Patras, Greece.

[5] C. J. Mitchell, "Truncation attacks on MACs", IEE 2003 Electronics Letters Online No: 20030921.

[6] Long-Jun Zhang, Ji-Wu Huang, "Identity verification mechanism for E-commerce", Zhongshan Daxue Xuebao, 2003, pp. 20.

[7] Chao Li, Yi-Xian Yang, Xin-Xin Niu, "Biometric-based personal dentity-authentication system and security analysis", Journal of China Universities of Posts and Telecommunications, 2006, vol. 13, no 4, pp. 43-47.

[8] Marsh, “Identity and Authentication in the E-economy”, Information Security Technical Report, September, 2002, no 3, pp. 12-19.

[9] Soh Ben, Joy Aaron, “A hybrid authentication mechanism for e-commerce systems”, Proceedings of the IASTED International Conference on Internet and Multimedia Systems andApplications, 2003, pp. 637-641.

[10] Zarooni Abdulkarim AI, Mabrouk Ali, AI-Qayedi Ali, Adi Wael, “Secure combined web/mobile E-commerce transactions”, Proceedings of the Second IASTED International Conference on Communications, Internet, and Information Technology, 2003, pp. 238-241.

[11] M. Toorani and A. Beheshti Shirazi, “SSMS - A secure SMS messaging protocol for the m-payment systems”, in Computers and Communications, IEEE Symposium on, July 2008 IEEE, pp. 700–705.

[12] C. Narendiran, S. Albert Rabara, N. Rajendran, “Performance Evaluation on End-to-End Security Architecture for Mobile Banking System”, 978-1-4244-2829-8/08/$25.00, 2008 IEEE.

[13] M. A. Hossain, S. Jahan, M. M. Hussain, M.R. Amin, and S. H. 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 IEEE, pp. 235-240.

[14] Mohsen Toorani, Ali Asghar Beheshti Shirazi, “Solutions to the GSM Security Weaknesses”, the Second International Conference on Next Generation Mobile Applications, Services, and Technologies, 2008 IEEE, pp. 576-581.

[15] W. Stallings, “Cryptography and network security”, Prentice Hall, 2006, New Jersey, United State.

[16] Neetesh Saxena and Ashish payal, “Enhancing Security System of Short Message Service for M-Commerce in GSM”, International Journal of Computer Science & Engineering Technology (IJCSET), ISSN: 2229-3345 Vol. 2 No. 4, April 2011, pp. 126-133.

[17] Neetesh Saxena, Narendra S. Chaudhari, “An Approach for SMS Security using Authentication Functions”, International Journal of Computer Application (IJCA), IJCA Special Issue on Communication Security comnetcs(1), ISBN: 973-93-80864-65-2, published by Foundation of Computer Science, New York, USA, March 2012, pp. 6-8.

[18] Web link: http://www.rsa.com/rsalabs/node.asp?id=2218

[19] Web link: http://en.wikipedia.org/wiki/MD5

[20] Web link: http://en.wikipedia.org/wiki/SHA-1

[21] Web link: http://www.rsa.com/rsalabs/node.asp?id=2248