Journal of Information Assurance and Security.
ISSN 1554-1010 Volume 8 (2013) pp. 156-166

Prevention of SMS against Repudiation Attack over the GSM Network

Neetesh Saxena[1], Narendra S. Chaudhari[2]

[1, 2] Department of Computer Science & Engineering
Indian Institute of Technology, Indore, India

neetesh.saxena@gmail.com, nsc183@gmail.com

Abstract: **As Short Message Service (SMS) is now widely used as a business tool, its security has become a major concern for business organizations and customers. However, their security is a critical issue cumbering their applications and development. This paper analyses the most popular digital signature algorithms such as DSA, RSA, ECDSA and a variant of ECDSA. These signature algorithms were implemented in Java with the different key sizes. The experimental comparison results of RSA, DSA and ECDSA digital signature algorithms are presented and analyzed. These experimental results show the effectiveness of each algorithm and to choose the most suitable algorithm for SMS digital signature. The results show that ECDSA is more suitable to generate the signature and RSA is more suitable to verify the signature on mobile devices. In this paper, we also find an attack on a variant of the ECDSA algorithm which seems more secure than the original ECDSA algorithm.**

Keywords: SMS, ECDSA, RSA, DSA, digital signature

I. Introduction

The mobile phone is already an integral part of the lives of more than 1.8 billion people worldwide [1]. With the Internet rapidly developing, SMS with e-commerce plays an important role in business transactions and is conducting business communications and solutions over the networks and through computers and mobiles [2]. These networks may be wireless or wired in nature. Apart from this, digital signatures are important because they provide not only end-to-end message integrity guarantees but also authentication information about the originator of a short message service (SMS). In applications, they are suitable for signing messages in e-commerce, e-voting, and other transactional activities. SMS is a store-and-forward, easy to use, popular, and low cost service [3]. But the problem is that the existing SMS is not free from the eavesdropping, but security is the main concern for any business company such as banks who are providing these mobile banking. Presently there is no such scheme which can give the complete SMS security [4].
The rapid development in mobile communication has transformed SMS as a wider tool for social and business messaging [1]. A number of new information and commercial technologies and applications have been explored in the past and mobile technology is one of them. Mobile applications have been developed and used in different areas. SMS is versatile, its services are growing day by day. With SMS, people can easily share personal and official messages in a fast and cost effective manner [2]. SMS enables the transmission of up to 1120 bits alphanumeric messages between mobile phones and external systems. In 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 [5]. SMS usage is threatened with security concerns [6], 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 contents are stored in the systems of the network operators and can easily be read by their personnel.

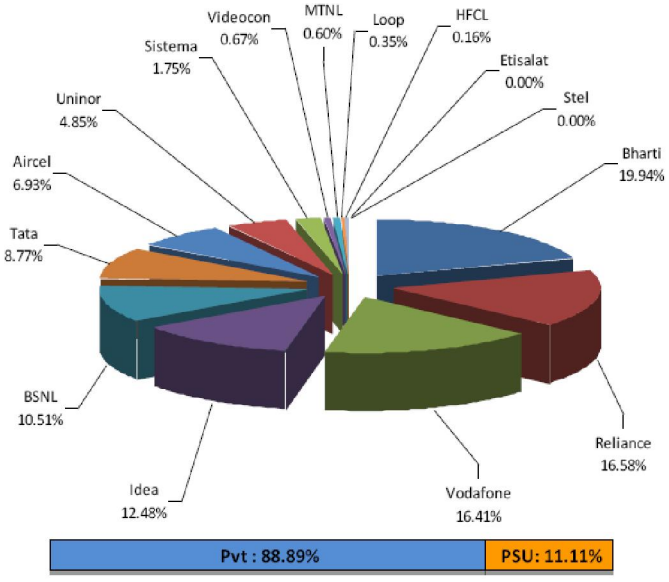

Figure 1. Service Provider wise Market Share (Wireless Segment) as on 31st May, 2012

The BTS act as a transmitter and receiver of the radio signals from mobile phones. The BTS translates the radio signals into digital format and then it transfers the digital signals to the Base Station Controller (BSC). The BSC controls multiple BTSs within a small geographical area. The BSC forwards the received signals to Mobile Switching Centre (MSC) and the MSC interrogates its databases (Home Location Register (HLR) and Visitor Location Register (VLR) for the location information about the destination mobile handset [7]. If the signal originates or terminates at the fixed telephone line network then the signal will be routed from the MSC to the Gateway MSC (GMSC).

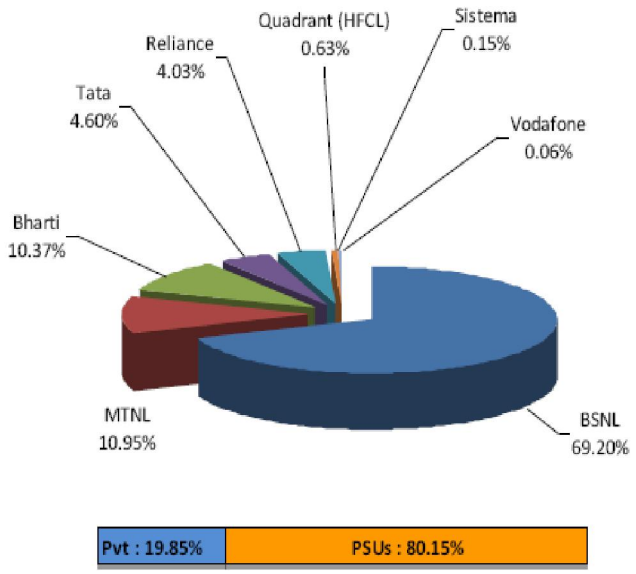

Figure 2. Service Provider wise Market Share (Wireline Segment) as on 31st May 2012

If the received signal is an SMS message 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. It uses an SMS center for its routing operation in one network and can be transmitted into another network through the SMS gateway [8]. The Indian market of the wireless and wireline segments is shared by various service providers where the number of telephone subscribers in India increased to 960.90 Million whereas the total number of wireless subscribers is 929.37 Million at the end of May 2012 [13]. Figure 1 and Figure 2 represent the private and public distribution of wireless and wireline segments of market share respectively.

This paper is divided into eight sections. Second section represents the preliminary information about the Unstructured Supplementary Service Data (USSD) and the SMS. In the section 3, the literature review of the work done in the past is discussed. Section 4 illustrates the architecture of an SMS. Section 5 and section 6 discuss about the data encryption and integrity, and, the digital signature algorithms respectively. Section 7 discusses an attack over the variant of ECDSA algorithm. Finally the conclusion is summarized in Section 8.

II. Preliminary

This section describes about the various ways to use m-banking mainly USSD and SMS. We prefer SMS over the USSD because of SMS's popularity and, its store and forward mechanism. A brief description with the difference of both the ways are as follows:

A. Unstructured Supplementary Service Data (USSD)

USSD is a session oriented service protocol. It is used by the GSM cellular network to communicate information between a user and an application specific computer system. It can be used for WAP based applications, menu-based information services, mobile based money services, prepaid call-back service, location based services on the network. These USSD messages are up to 182 alphanumeric characters in length. Unlike SMSs, USSD messages create a real-time based connection during a session. The connection remains open, allowing a bidirectional exchange of data. This makes the message more responsive than services that use SMS [1]. A typical USSD message starts up with a * followed by some digits which shows an action to be performed or are some basic parameters. Each group of numbers is separated by a * and the message is terminated by a #. The USSD gateway can interact with external applications based on the USSD command, which allows access to a number of value added services via USSD [9], [10].

Nowadays a USSD message can work in two different ways: one is USSD1 and the other is USSD2. USSD2 allows messages to be pushed in a mobile phone. It is several times faster than mobile originated SMS (MO-SMS). It doesn't have the mechanism to store and forward message. The USSD gateway supports an open HTTP interface and will also have an interface to the MSC over SS7. The functionality will not change in roaming because USSD messages always routed back to Home Location Register HLR. It uses the MAP protocol to send and receive USSD data from the HLR. USSD1 only allows one-way communication to the network, while USSD2 allows two way communications between the user and the network. In the USSD1, the data is segmented in the same way as in SMS. In USSD2, it held in the same session and allows the conversation between user and service. It is similar to e-mail and instant messaging, e-mail waits for the recipient to read and respond while as instant messaging allows for immediate dialogue. Generally the USSD functionality is implemented in the following two modes: Pull Mode handles Mobile Initiated USSD Requests and Push Mode handles network Initiated USSD Requests [11] [12].

B. Short Message Service (SMS)

Today, the mobile technology made it possible and all are acquainted with SMS. The Short Message Service is one of its superior and well-tried services with a global availability in the GSM networks and at the beginning of 2007, the worldwide number of mobile users reached to 2.83 billion people. The SMS is the most popular data bearer/service within GSM, IS-95, CDMA2000, and other cellular networks. It is a store-and-forward, easy to use, popular, and low cost service. While it is mainly used for the personal communications, it has also been used in applications where the other party is an information system [5]. Public-key algorithms consider mutual authentication and key exchange between two untrusted parties such as two nodes in a wireless sensor network. The transmission of an SMS in GSM network is not secure; therefore it is desirable to secure SMS by additional encryption. The SMS tapping is possible in GSM network at some places. There could be used the encryption for securing of SMS. Encryption is most often realized through some user encryption applications. So there

is a need of comparing differences in the use of symmetric and asymmetric cryptography for SMS transfer securing [6].

Table 1. Evolution of SMS in GSM

Year	Description
1982	European Conference of Postal and Telecommunications Administrations (CEPT) created the Group Special Mobile (GSM) which develop a standard for a mobile telephone system that could be used across Europe [44]
Mid 1984 to Feb. 1985	Development of the SMS service concept and contribution to GSM work by Franco German cooperation [45]
Feb. 1985 to End 1986	Standardization of the SMS point-to-point service concept in GSM [45]
Mid 1987 to End 1990	Technical design of SMS point-to-point including support on the radio interface and in the GSM network [45]
End 1990 to 1996	Technical improvements to SMS person-to-person: e.g. multiple service centers, delivery reports, SMS character sets, SIM management by SMS, concatenated SMS in GSM [45]
1997 to 2005	Technical evolution of SMS features: e.g. SIM toolkit data download, Enhanced Messaging Service, voice mail management, routers, language tables (in SMG/3GPP) [45]
2010	SMS text messaging is the most widely used data application in the world, with over 3.7 billion active users, or 78% of all mobile phone subscribers [46]
2010	The GSM Association estimates that 80% of the global mobile market uses the standard. GSM is used across more than 212 countries and territories [44]

Some of the messages are normally computer generated messages sent over Short Message Peer to Peer (SMPP) protocol. SMS is the text communication service component of mobile communication systems, using standardized communications protocols that allow the exchange of short text messages between mobile phone devices. SMS will play a very vital role in the future business areas whose are popularly known as M-Commerce, mobile banking etc. For this future commerce, SMS could make a mobile device in a business tool as it has the availability and the effectiveness. 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 [7], [14]. Table 1 lists the development of SMS evolution in the GSM network.

The rapid development in mobile communication has transformed SMS as a widespread tool for business and social messaging. As an SMS is now widely use as a business tool, it security has become a major concern for business organizations and customers. There is a need for an end to end SMS Encryption in order to provide a secure medium for communication. Paper [1] introduces our Trusted-SMS system, which allows users to exchange non-repudiable SMS's, digitally signed. SMS messages are used in many different application fields, even in cases where security features, such as authentication and confidentiality between the communicators must be ensured. Unfortunately, the SMS technology does not provide a built-in support for any security feature.

C. USSD Vs SMS

USSD differs from SMS in many ways. Some of these differences are as follows [15]:

1. SMS uses a store and forward technique to deliver text messages while USSD doesn't have the storage capacity.
2. USSD is sent directly from a sender's mobile handset to an application platform handling the USSD service while SMS is sent to first SMSC and then to the handset of the recipient.
3. A real-time session is initiated between the mobile user and the USSD application platform when the service is invoked while SMS uses a store and forward mechanism and there is no real-time connection.
4. A USSD service could be invoked by either the mobile user or the USSD platform but not fit for mobile-to-mobile service while SMS is suitable for this kind of service.
5. An SMS is a one-way information message from one user to another. USSD on the other hand, is an interactive, two-way communication between a user and a service.
6. SMS messages are point-to-point calls (from one user to another user). USSD services are interactive user to service provider calls.
7. USSD is much faster (up to seven times) than SMS.
8. USSD carries longer character messages than SMS (182 compared to 160).
9. In rural area, people prefer to send an SMS than USSD.

In this paper, we focus on the security of SMS especially against the repudiation attack. The repudiation attack can be prevented by imposing the digital signature over the message.

III. Related Work

Many authors have used different encryption techniques to provide confidentiality of the transmitted messages. Some of these works are presented in this section. In a study by Mary Agoyi and Devrim Seral [1] large key size algorithms are not suitable for SMS encryption due to small memory and low computational power of mobile phones. Elliptic curve's ability of providing high security with smaller key size makes it very useful in resource-limited device such as mobile phones. This has put Elliptic curve at an advantage

over the RSA and ELGamal in SMS encryption. In the work of Alfredo De Santis, Aniello Castiglione and Umberto Ferraro Petrillo [2] the results seem to show that RSA and DSA cryptosystems perform generally better than ECDSA, except when using very large keys. Nassim Khozooyi, Maryam Tahajod and Peyman khozooyi [14] are discussed the security of the mobile network protocol along with information security for governmental transactions. A new public key-based solution for secure SMS messaging (SSMS) is introduced by M. Toorani and A. Beheshti Shirazi [5]. It efficiently combines encryption and digital signature and uses public keys for a secure key establishment to be used for encrypting the short messages via a symmetric encryption. In a study of [16] the application for securing of SMS has been designed and implemented, which prevents tapping and also substituting. For securing, it has been chosen the asymmetric cipher RSA. Brutal force decryption of RSA cipher with a length of 1,024 bit keys has not been successfully implemented yet. In the paper of C. Narendiran, S. Albert Rabara and N. Rajendran [7] an end-to-end security framework using PKI for mobile banking is proposed. The security framework solution allows us to provide strong customer authentication and non-repudiation by employing public-key cryptography for customer certificates and digital signatures. In the paper of Mohsen Toorani, Ali Asghar and Beheshti Shirazi [17], the security of the GSM network is evaluated, and a complete and brief review of its security problems is presented. Next the technical paper [18] describes the NextGen Short Message Gateway (NSMG) Architecture, which can support SMS over cellular, non-cellular generic IP networks and internetworking between the different messaging methods used in different networks.
The [19] proposes an ECC-based PKI that overcomes all the limitations of the mobile phone's small screen, low computing power, small storage capacity etc. In 1994, Lim and Lee [20] proposed a more flexible pre-computation method LLECC used in wireless network environments for speeding up the computation of exponentiation which is also used for speeding up the scalar multiplication of elliptic curves. However, the less storage is equipped with the computing devices, the less efficient it is and for this reason, [21] proposes a more efficient algorithm than LLECC_s. The [22] presented a practical implementation of the ECC over the field GF (p) and obtained timing results of 46ms and 92ms for the ECC-160 signature generation and verification on a 32-bit ARM processor, respectively. Hu Junru in [23] proposed an improved algorithm of ECDS which reduces the computational cost while keeping the same security as original ECDSA and is suitable for the users who have limited compute capacity in different cases. The ECDSA is also used in VANETs and wireless sensor networks. A variation of elliptic curve digital signature algorithm (ECDSA) is used in combination with the identity-based (ID-based) signature where current position information about a vehicle is utilized as the ID of the corresponding vehicle [48]. Similarly, in [49], the proposed scheme exploits an Elliptic Curve Digital Signature Algorithm (ECDSA) signature to authenticate all broadcast messages.

IV. SMS Architecture

This section discusses about the implementation of secure SMS exchange by using binary SMS messages rather than traditional textual messages.

1 byte	8 bytes	1 byte	8 bytes	122 bytes
UDP Header	UDP sender and receiver port	Msg Type	Timestamp	Data

Figure 3. SMS payload

Each SMS message can hold a maximum of 140 bytes (equivalent to the 160 7-bit characters used for text messages) [3]. This total 140 bytes are partitioned as shown in Figure 3. The first two fields represent the User Data Header (UDH), an extension to the GSM specifications that deliver the message to a specific application listening on a specific port of destination. Next, the subsequent 9 bytes are used to specify the message type (1 byte) and the timestamp (8 bytes). The Message Type field indicates the Encrypted approach used to process the current SMS and the key length used by that cipher. The Timestamp field stores the time when the SMS has been sent. Finally, data field is used by the chosen cryptosystem to carry the contents of SMS with public-keys and signatures [2]. One SMS message can contain at most 140 bytes (1120 bits) of data, so one SMS message can contain up to:
1. 160 characters if 7-bit character encoding is used. (Encoding Latin characters like English alphabets.)
2. 70 characters if 16-bit Unicode UCS2 (2-byte Universal Character Set) character encoding is used. (SMS text messages containing non-Latin characters like Chinese characters should use 16-bit character encoding).
There are two ways of SMS transmission: one is Mobile Terminated SMS and another is Mobile Originated SMS.

A. SMS Architecture: Mobile-Terminated

Step 1. The short message is first delivered from the message sender GSM MS to a Short Message Service Center (SMS-C).
Step 2. The SMS-C is connected to the GSM network through a GSM-MSC and SMS-GMSC.
Step 3. Following the GSM roaming protocol, the SMS-GMSC locates the current MSC of the message receiver and forwards the message to the MSC.
Step 4. The MSC broadcasts the message to BSS, and BTS page the destination MS.
Step 5. The MS used for short message services must contain special software to enable the messages to be decoded and stored.
The logical message path is SMS-C -> GMSC -> terminating MSC -> MS.
Short messages can be stored either in the SIM or in the memory of ME for display on the standard screen of the MS.

B. SMS Architecture: Mobile-Originated

Step 1. An MS may send or reply a short message by delivering to a short message service Inter-working MSC

(IWMSC) and then to the SMS-C.
Step 2. The recipient of the short message can be an MS, a fax machine, or a PC connected to the Internet.
The logical message path is MS -> IWMSC -> SMS-C
As SMS is a store-and-forwarded service, Short message cannot be sent directly from the sender to the receipt without passing through the SMS-C.
Three types of short messages:
1. User-Specific messages are displayed to the users.
2. ME-Specific messages are processed by the ME instead of showing to the users.
3. SIM-Specific messages are processed on the SIM card.

V. Data Encryption and Integrity

Both symmetric and asymmetric cryptography can be used to encrypt the message. But the main problem with symmetric key system is its secret key; anyone who knows the secret key can decrypt the message very easily. So to prevent it, one answer is asymmetric encryption. In this paper, an asymmetric algorithm like RSA is implemented with PKCS1 padding scheme and OAEP padding scheme with MD5 and SHA1 message digests based on a hash function.

RSA Algorithm

RSA is considered secure with respect to its factorization problem. In other words, the difficulty of factoring large numbers is the basis of the security of RSA and over 1000 bits long numbers are used [24]. The RSA algorithm can be summarized as:
Select random prime numbers p and q, and check that $p \neq q$
Compute modulus n = p*q
Compute phi, ø = (p - 1)*(q - 1)
Select public exponent e, $1 < e < ø$ such that gcd(e, ø) = 1
Compute private exponent d = $e^{-1} \bmod ø$
Public key pair is {n, e}, private key pair is {n, d}
Encryption: c = $m^e \bmod n$,
Decryption: m = $c^d \bmod n$
The RSA algorithm is implemented here with PKCS1 and OAEP padding schemes. A brief detail of these padding schemes is as follows:

1) PKCS1 padding [25]

PKCS1 is a part of the family of standards called Public-Key Cryptography Standards (PKCS) which was published by the RSA Laboratories. This scheme provides the basic definitions and recommendations to implement the RSA algorithm as a use for public-key cryptography. This standard defines the mathematical definitions and properties that RSA keys (public and private) must retain. The RSA key pair is based on a modulus n, which is a product of two distinct large prime numbers, say p and q in such a way that n = p*q. The RSA public and private keys are represented as the tuples {n, e} and {n, d} respectively, where the integer e is the public exponent and d is the private exponent.

2) Optimal Asymmetric Encryption Padding (OAEP) [26]

OAEP is a padding scheme which is often used together with the RSA algorithm to encrypt the data. It is a form of a Feistel network as shown in Figure 4, which uses a pair of random oracles G and H to process the plaintext prior to asymmetric encryption. This scheme is also proved secure against chosen cipher text attack. To recover m, you must recover the entire X and the entire Y; X is required to recover r from Y, and r is required to recover m from X. Since any bit of a cryptographic hash completely changes the result, the entire X, and the entire Y must both be completely recovered. OAEP satisfies the following two goals:
1. Add an element of randomness which can be used to convert a deterministic encryption scheme (e.g., RSA algorithm) into a probabilistic scheme.
2. Prevent partial decryption of cipher text (or other information leakage) by ensuring that an adversary cannot recover any portion of the plaintext without being able to invert the trapdoor one-way permutation.

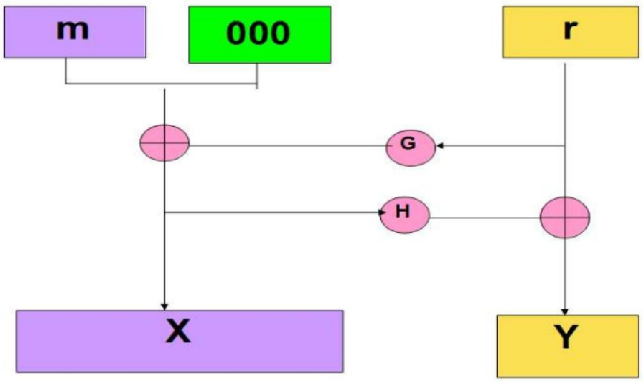

Figure 4. OAEP Padding Scheme

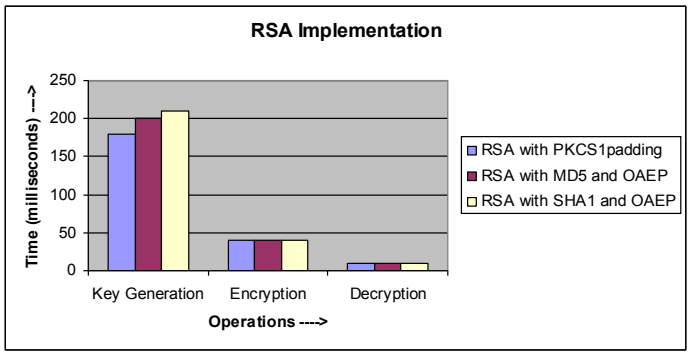

Figure 5. RSA Key Generation (1024 bits) with Encryption and Decryption

To provide integrity, the hash code should be sufficiently small enough to be manageable in further manipulations and large enough to prevent an attacker from randomly finding a block of message that generates the same hash code [27].

VI. Focus on the Digital Signature Algorithms

Digital signatures rely on the encryption process to ensure the authentication [28]. Digital signatures can provide the assurance of the evidence for provenance and identity approval by a signatory. Some common reasons are to for applying a digital signature to communications including

Authentication, Integrity, and Non-repudiation [27]. In this section, we will discuss some popular digital signature algorithms like RSA, DSA and ECDSA.

A. *RSA Digital Signature Algorithm [24] ,[29]*

In the RSA digital signature process, the private key is used to cipher the message digest and that ciphered message digest becomes the digital signature. The original message m is never signed directly, instead it is usually hashed with the hash function and that message digest is signed. To verify the contents of digitally signed message, the receiver generates a new message digest from the received message. The recipient decrypts the original message digest with the sender's public key and compares the decrypted digest with the newly generated digest. If the two digests are equal then the integrity of the message is verified. The identity of the sender is also confirmed because the public key can decrypt only that message which was previously encrypted with the corresponding private key. The RSA digital signature generation and verification is done as follows:

Select random prime numbers p and q, and check that $p \neq q$
Compute modulus $n = p*q$
Compute phi, $ø = (p - 1)*(q - 1)$
Select public exponent e, $1 < e < ø$ such that $gcd(e, ø) = 1$
Compute private exponent $d = e^{-1} \bmod ø$
Public key pair is $\{n, e\}$, private key pair is $\{n, d\}$
Digital signature: $s = H(m)^d \bmod n$, where H is a publicly known hash function.
Verification: $m' = s^e \bmod n$,
If $m' = H(m)$ then the signature is verified.

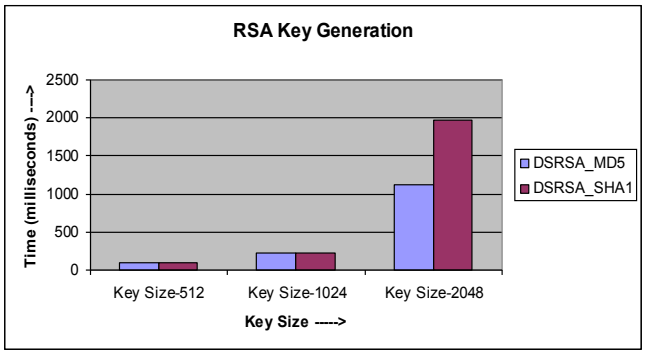

Figure 6. Digital Signature RSA with Key Generation

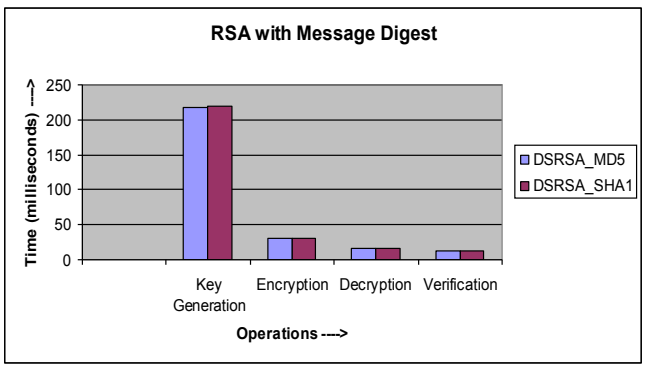

Figure 7. Digital Signature RSA with Message Digests

Signature Generation:
First pass the message through the hash function to create the message digest: $m' = H(m)$, then compute $s = m'^e \bmod n$, where n is the modulus and d is the private key. The end result is s which is the signature.

Signature Verification:
Since the signatures are always verified with public key the public key must be obtained before the signature can be verified. Signature verification can be done as: $m' = s^d \bmod n$, where *m'* is the *H(m)*. If the verification fails then the signature is not authentic.

B. *DSA Algorithm [30]*

In this subsection we discuss about the DSA algorithm which is also used to create digital signatures.

DSA Parameters
A DSA digital signature is computed using a set of domain parameters as a private key x, a per-message secret number k, the message to be signed m, and a hash function H which generates hash code *H(M)*. A digital signature is verified using the same domain parameters with a public key y that is mathematically associated with the private key x. These parameters are defined as follows:

p: a prime modulus, where $2^{L-1} < p < 2^L$, and L is the bit length of p

q: a prime divisor of (p – 1), where $2^{N-1} < q < 2^N$, and N is the bit length of q

g: a generator of the subgroup of order q mod p, such that $1 < g < p$

x: the private key which is a randomly or pseudorandom generated integer, such that $0 < x < q$, i.e., x is in the range $[1, q–1]$

y: the public key, where $y = g^x \bmod p$

k: a secret number that is unique to each message which is a randomly or pseudorandom generated integer, such that $0 < k < q$, i.e., k is in the range $[1, q–1]$

DSA Signature Generation
Let N be the bit length of q. Let *Min(N, M)* denote the minimum of the positive integers N and M, where M is the bit length of the hash function output. The signature of a message m consists of the pair of numbers r and s that is computed according to the following equations:

$r = (g^k \bmod p) \bmod q$

z = the leftmost *Min(N, M)* bits of *H(m)*

$s = (k^{-1}(z + xr)) \bmod q$

When computing s, the string z obtained from *H(m)* shall be converted to an integer and r is computed based on k, p, q and g. The DSA uses *SHA1* as the hash function which generates 160 bits of message digest and that works as z. The values of r and s shall be checked to determine if either $r = 0$

or $s = 0$, a new value of k shall be generated, and the signature shall be recalculated. It is extremely unlikely that $r = 0$ or $s = 0$ if signatures are generated properly. The signature *(r, s)* may be transmitted along with the message to the verifier.

DSA Signature Verification

Signature verification is performed using the signatory's public key. Prior to verifying the signature of a signed message, the domain parameters, and the signatory's public key and identity shall be made available to the verifier. Let m', r', and s' be the received versions of m, r, and s, respectively. The signature verification process is as follows:

1. The verifier shall check that $0 < r' < q$ and $0 < s' < q$; if either condition is violated, the signature shall be rejected as invalid.
2. If the two conditions in step 1 are satisfied, the verifier computes the following:

$$w = (s')^{-1} \bmod q$$

z = the leftmost *Min(N, M)* bits of *H(m')*

$$u_1 = (zw) \bmod q$$

$$u_2 = ((r')w) \bmod q$$

$$v = (((g)^{u_1} (y)^{u_2} \bmod p) \bmod q$$

3. If $v = r'$, then the signature is verified. For a proof that $v = r'$ when $m' = m$, $r' = r$, and $s' = s$. If the $v \neq r'$ then the signature is not authentic.

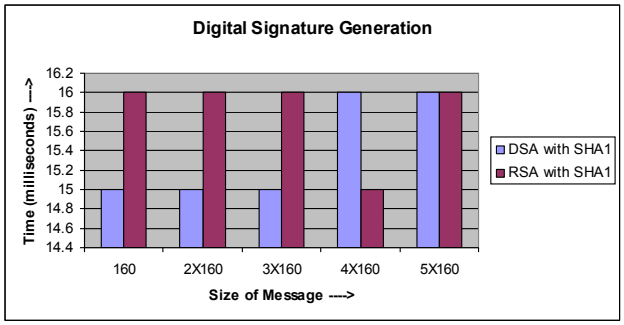

Figure 8. Digital Signature DSA and RSA Generation

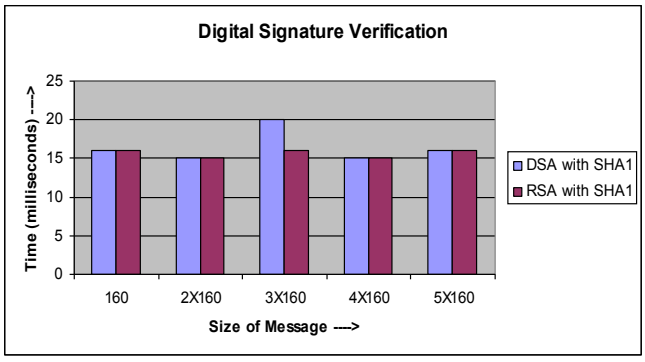

Figure 9. Digital Signature DSA and RSA verification

C. *ECDSA Algorithm*

The mathematical basis for the security of elliptic curve cryptosystems is the computational intractability of ECDLP and smaller parameters can be used in ECC than with DL systems with equivalent levels of security. Select a rational point G on $E(GF(p))$, called base point, find n which is a prime number $E(GF(2^n))$ where 2^n for binary) satisfies the formula $n*G = O$, and select a one-way secure Hash function H(m) such as *SHA1*. For each system user, there is a private key d, calculate the public key $P = d*G$. If User A wants to sign on the message m, the scheme can be described as:

1. User A selects an integer k randomly, $0 < k < n$, calculate $k*G = (x, y)$, $r = x \bmod n$; if $r = 0$, return to (1).
2. Calculate $e = H(m)$, $s = k^{-1} (e + r * d) \bmod n$, if $s = 0$, return to (1).
3. Take *(r, s, e)* as the digital signature of message m by user A.

The verification of digital signature:

1. Calculate $e_1 = H(m_1)$, $u = s^{-1} * e_1 \bmod n$ and $v = s^{-1} * r \bmod n$
2. Calculate

$$X = u * G + v * P = s^{-1} (e_1 * G + r * d * G) = s^{-1}(e + r * d) G = k * G = (x_1, y_1)$$

3. If $X = 0$, this signature is refused; else calculate $r_1 = x_1 \bmod n$; if $r = r_1$, the User B accepts this signature.

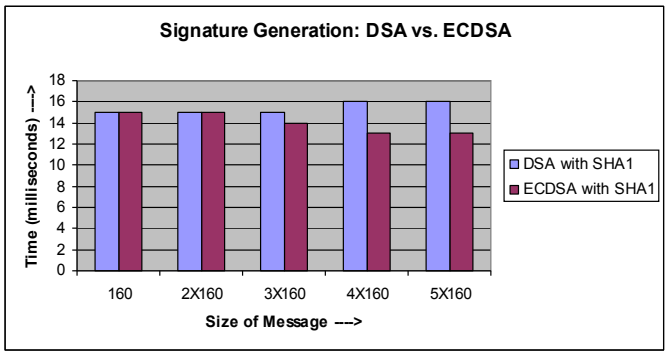

Figure 10. Digital Signature DSA and ECDSA Generation

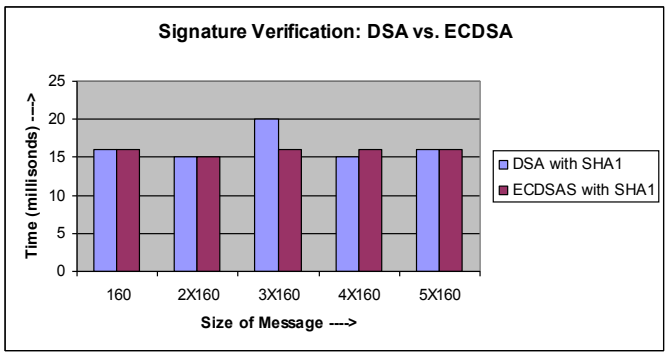

Figure 11. Digital Signature DSA and ECDSA verification

The security of such schemes like DSA and ECDSA relies on the hardness of the discrete logarithm problem, either in the multiplicative group of a prime field or in a subgroup of points of an elliptic curve over a finite field. These results were obtained on computer machine with a configuration pentium4 processor, 160 GB hard disk, 1 GB RAM and Windows7 operating system. The platform used is Java (JDK 1.6) and Java API for this work. Here, three observations have taken for each encryption and decryption process with

various data sizes and finally the average of three is considered.

Figure 5 shows the RSA key generation with 1024 bits. Apart from this it also covers the encryption and decryption process with different padding schemes like PKCS5 padding and OAEP padding. For maintaining the integrity of the message, the message digest algorithms like MD5 and SHA1 are also implemented. Out of these scenarios, RSA with SHA1 and OAEP padding provides the best security, although its key generation takes more time than the other scenarios but the encryption and decryption take almost same time in all scenarios. Figure 6 and Figure 7 shows the results for RSA as digital signature algorithm. Figure 6 is having the result of total execution time to generate keys of size 512-bits, 1024-bits and 2048-bits while Figure 7 demonstrates the result of signature generation with the encryption and signature verification with decryption where the 1024-bits key is used with two message digest algorithms MD5 and SHA1. Out of these SHA1 provides more security as it has a more complex structure than MD5. Figure 8 and Figure 9 explains the results of RSA and DSA signature generation and signature verification respectively. This shows that signature generation takes less time in DSA but it takes more time to verify the signature. Figure 10 and Figure 11 shows the comparison between DSA and ECDSA. The results show that ECDSA is better than DSA in signature generation and verification.

D. A Variant of ECDSA Approach [47]

The security objective of ECDSA is unforgeable against a chosen message attack, and it has been proven secure by Brown [31] under the assumption that the underlying group is a generic group and the hash function employed is collision resistant. The possible attacks on ECDSA can be based on ECDLP and the hash function employed. Some of the attacks and their solutions have summarized in some algorithms [32], [33], [34], [35], [36], [37], [38], [39], [40], and [41]. As a security aspect, a variant of ECDSA was proposed. Let us select a rational point *G* on *E(GF(p))*, called base point, find n which is a prime number *E(GF(2^n))* where 2^n for binary) satisfies the formula *n*G = O*, and select a one-way secure Hash function *H(m)* such as *SHA1*. For each system user, there is a private key *d*, calculate the public key *P = d*G*. If User A wants to sign on the message m, the scheme can be described as:

1. User A selects two integer k randomly, $0 < k_1, k_2 < n$, Calculate $k_1 * G = (x_1, y_1)$, $k_2 * G = (x_2, y_2)$, $r_1 = x_1 \bmod n$, $r_2 = x_2 \bmod n$; if $r_1 = r_2 = 0$, return (1).
2. Next, Calculate $e = H(m); s = k_1^{-1}(e * k_2 + (r_1 + r_2) * d) \bmod n$, if *s = 0*, return to (1).
3. Take (r_1, r_2, s, k_2, e) as digital signature of message m by user A.

The verification of digital signature:

1. Calculate $e_1 = H(m_1)$, $u = s^{-1} * e_1 * k_2 \bmod n$ and $v = s^{-1} * (r_1 + r_2) \bmod n$
2. Calculate $X(x_3, y_3) = u * G + v * P =$

$$s^{-1}(e_1 * k_2 * G + (r_1 + r_2) * d * G) = s^{-1}(e_1 * k_2 + (r_1 + r_2) * d) * G = k_1 * G$$

3. If *X = 0*, this signature is refused; else calculates $r_3 = x_3 \bmod n$; if $r_3 = x_3$, the User B accepts this signature.

Security analysis

Primarily we consider that an adversary can determine an integer k randomly, and use it to recover the private key of user d. Now, suppose that the same per-message secrets k_1 and k_2 are used to generate the ECDSA signatures (r, s_1) and (r, s_2) on two different messages m_1 and m_2. Then

$$s_1 = k_1^{-1}(e_1 * k_2 + (r_1 + r_2) * d) \bmod n$$

$$s_2 = k_1^{-1}(e_2 * k_2 + (r_1 + r_2) * d) \bmod n$$

where e_1 and e_2 are the message digest of some cryptographic algorithms like *SHA1*. Thus,

$e_1 = \text{SHA}(m_1)$, and $e_2 = \text{SHA}(m_2)$; Then

$$k_1 * s_1 = (e_1 * k_2 + (r_1 + r_2) * d) \bmod n$$

and,

$$k_1 * s_2 = (e_2 * k_2 + (r_1 + r_2) * d) \bmod n$$

Thus, $k_1 * (s_1 - s_2) = ((e_1 - e_2) * k_2) \bmod n$

$$k_1 = ((s_1 - s_2)^{-1} * (e_1 - e_2) * k_2) \bmod n$$

An adversary can't determine the secret key k_1 because k_2 is unknown to the adversary.

VII. An Attack on Variant of ECDSA Algorithm

From the security analysis of the variant ECDSA it seems that this algorithm is safe and secure. However, this variant algorithm is not safe and secure as an attacker can modify the message contents without knowing the private key of the actual sender. We assume that an attacker knows the global parameter *G* as well as captures s from the signature information. Now, in between the attacker capture the sent packet and calculate: $s' = s * G$,

$$\{s' = k_1^{-1}(e * k_2 + (r_1 + r_2) * d) * G \bmod n =$$

$$k_1^{-1}(e * k_2 * G + (r_1 + r_2) * d * G) \bmod n =$$

$$k_1^{-1}(e * k_2 * G + (r_1 + r_2) * P) \bmod n\};$$

It means anyone who knows group *G* and public key *P*, can apply signature without knowing the private key *d*. The attacker can change the value of hash code *e'* and calculate the value of *s'* as:

$s' = k_1^{-1}(e' * k_2 * G + (r_1 + r_2) * P) \bmod n$; because an

attacker knows all these parameters. The attacker forwards the packet (r_1, r_2, s', k_2, e') to the recipient.

This signature can be verified as:

1. Calculate $e_1 = H(m_1)$,

2. Calculate $u = s'^{-1} * e_1 * k_2 \mod n$, and

$v = s'^{-1} (r_1 + r_2) \mod n$

3. Calculate $X(x_3, y_3) = u * G + v * P =$

$s'^{-1} (e_1 * k_2 * G + (r_1 + r_2) * P) = k_1 * G$

$\{ k_1 = s'^{-1} (e_1 * k_2 * G + (r_1 + r_2) * P)$

$k_1 = G^{-1} * s^{-1} (e_1 * k_2 * G + (r_1 + r_2) * P)$

Now,

$k_1 * G = G * G^{-1} * s^{-1} (e_1 * k_2 * G + (r_1 + r_2) * P)$

$k_1 * G = s^{-1} (e_1 * k_2 * G + (r_1 + r_2) * P) = X(x_3, y_3) \}$

4. If $X = 0$, this signature is refused; else calculate $r_3 = x_3 \mod n$; if $r_3 = r_1$, the User B accepts this signature.

Thus, an attacker can get verify the digital signature and can misuse it for some malicious purpose.

VIII. Conclusion

The approach to prevent the SMS from repudiation attack has been designed and implemented. This approach includes the encryption of message asymmetrically using RSA algorithm and then apply a digital signature over the encrypted message. Various digital signature algorithms RSA, DSA and ECDSA were presented along with a variant of ECDSA algorithm. Although, DSA and ECDSA are popular digital signature algorithms, but for the quantum computer environment, these algorithms must be strong enough in order to break or prove vulnerable. There is a need to analyze the existing algorithms and find a better algorithm for digital signature on some harder problems. The security of such schemes relies on the hardness of the discrete logarithm problem, either in the multiplicative group of a prime field or in a subgroup of points of an elliptic curve over a finite field. We found an attack on a variant of the ECDSA algorithm which seems to be stronger than the existing ECDSA by choosing the random number k twice.

Acknowledgment

This work is supported by Tata Consultancy Services Limited (TCS), India.

References

[1] Mary Agoyi, Devrim Seral. “SMS Security: An Asymmetric Encryption Approach”, *Sixth International Conference on Wireless and Mobile Communications*, pp. 448-452, 2010.

[2] Alfredo De Santis, Aniello Castiglione and Umberto Ferraro Petrillo. “An Extensible Framework for Efficient Secure SMS”, *International Conference on Complex, Intelligent and Software Intensive Systems*, pp. 843-850, 2010.

[3] Salman Firdaus bin Haji Sidek. “The Development of the Short Messaging Service (SMS) Application for the School Usage”, *2010 International Symposium in Information Technology (ITSim)*, pp. 1382-1386, 2010.

[4] Neetesh Saxena, 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, pp. 126-133, April 2011.

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

[6] Lisonek and M. Drahansky. “SMS Encryption for Mobile Communication”, *International Conference Security Technology, SECTECH ’08*, pp. 198–201, 2008.

[7] A. Narendiran, S. Albert Rabara, N. Rajendran. “Performance Evaluation on End-to-End Security Architecture for Mobile Banking System”, *1st IFIP Wireless Days, WD’08*, pp. 1-5, 2008.

[8] J. Brown, B. Shipman, and R. S. Vetter. “SMS: The short message service”, *Computer*, Vol. 40, No. 12, pp. 106-110, 2006.

[9] J. Sanganagouda. “USSD - A Potential Communication Technology that can Ouster SMS Dependency”, *International Journal of Research and Reviews in Computer Science (IJRRCS)*, Vol. 2, No. 2, April 2011.

[10] J. Hellström. “The Innovative Use of Mobile Applications in East Africa”, *SIDA 2010*, Available: http://upgraid.files.wordpress.com/ 2010/06/sr2010-12_sida_hellstrom.pdf

[11] Myriadgroup. “USSD Browsing,” *Myriadgroup*, July 2011. http://www.myriadgroup.com/Mobile-Operators/SelfCare-Services.aspx

[12] Google Groups. “Development of the Interactive USSD Services for Mobile Users using Mobicents Platform”, *MobiCents*, May 2010, www.groups.google.com/group/mobicentspublic/web/interactiveussd-services-for-mibile-users?pli=1

[13] Telecom Regulatory Authority of India. “Highlights on Telecom Subscription”, Data as on 31st May 2012, Press Release No. 143/2012 (4th July 2012).

[14] Nassim Khozooyi, Maryam Tahajod, Peyman khozooyi. “Security in Mobile Governmental Transactions”, *2009 Second International Conference on Computer and Electrical Engineering*, pp. 168-172, 2009,.

[15] MobiCent. “USSD”, July 2011, MobiCent. http:// www.mobicents.org/ incubator/ussd/intro.html.

[16] Lisonek and M. Drahansky. “SMS Encryption for Mobile Communication”, *International Conference on Security Technology, SECTECH ’08*, pp. 198–201, Dec. 2008.

[17] Mohsen Toorani, Ali Asghar Beheshti Shirazi. “Solutions to the GSM Security Weaknesses”, *Second International Conference on Next*

Generation Mobile Applications, Services, and Technologies, pp. 576-581, 2008.

[18] Muhammad Saleem, Kyung-Goo Doh. "Generic Information System Using SMS Gateway", *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, pp. 861-866, 2009.

[19] Sangram Ray, G. P. Biswas. "An ECC based Public Key Infrastructure usable for Mobile Applications", *International Conference on Computational Science, Engineering and Information Technology, CCSEIT-12*, pp. 562-568, 2012.

[20] H. Lim, P. J. Lee. "More flexible exponentiation with precomputation", *Advances in Cryptology, Crypto-94*, Springer-Verlag, Berlin, pp. 95–107, 1994.

[21] Woei-Jiunn Tsaur, Chih-Ho Chou. "Efficient algorithms for speeding up the computations of elliptic curve cryptosystems", *Elsevier Applied Mathematics and Computation*, 168, pp. 1045–1064, 2005.

[22] M. Aydos, T. Yanik and C. K. KOG. "High-speed implementation of an ECC-based wireless authentication protocol on an ARM microprocessor", *IEE Proc Commun.*, Vol. 148, No. 5, pp. 273-279, Oct 2001.

[23] Hu Junru. "The Improved Elliptic Curve Digital Signature Algorithm", *2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, pp. 257-259, 2011.

[24] Pekka Riikonen. "RSA Algorithm," 2002, http://iki.fi/priikone/docs/ rsa.pdf

[25] PKCS1, http://en.wikipedia.org/wiki/PKCS1

[26] OAEP, http://en.wikipedia.org/wiki/Optimal_ asymmetric _ encryption_padding

[27] W. Stallings. "Cryptography and network security", *Prentice Hall*, New Jersey, United State, 2006.

[28] Yu Lei, Deren Chen and Zhongding Jiang, "Generating digital signatures on mobile devices,"18th *International Conference on in Advanced Information Networking and Applications, AINA 2004*, Vol. 2, pp. 532 – 535, 2004.

[29] Digital Signatures, http://technet.microsoft.com/en-us/library/ cc962021.aspx

[30] Federal Information Processing Standards Publication, Digital Signature Standard (DSS), FIPS PUB 186-3, *Information Technology Laboratory, NIST*, June 2009.

[31] A. Brown. "The exact security of ECDSA", *Technical report CORR 2000-54*, Dept. of C&O, University of Waterloo, 2000.

[32] S. Pohlig and M. Hellman. "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", *IEEE Transactions on Information Theory*, 24, pp. 106-110, 1978.

[33] J. Pollard. "Monte Carlo methods for index computation mod p", *Mathematics of Computation*, 32, pp. 918-924, 1978.

[34] A. Frey and H. R¨uck. "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves", *Mathematics of Computation*, 62, pp. 865-874, 1994.

[35] A. Menezes. "Elliptic Curve Public Key Cryptosystems", *Kluwer Academic Publishers*, Boston, 1993.

[36] Menezes, T. Okamoto and S. Vanstone. "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Transactions on Information Theory*, 39, 1993, pp. 1639-1646.

[37] T. Satoh and K. Araki. "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves", *Commentarii Mathematici Universitatis Sancti Pauli*, 47, pp. 81-92, 1998.

[38] Semaev. "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p," *Mathematics of Computation*, 67, pp. 353-356, 1998.

[39] N. Smart. "The discrete logarithm problem on elliptic curves of trace one", *Journal of Cryptology*, 12, pp. 193-196, 1999.

[40] R. Gallant, R. Lambert and S. Vanstone. "Improving the parallelized Pollard lambda search on binary anomalous curves", *Mathematics of Computation,* 69, pp. 1699—1705, 1998.

[41] M. Wiener and R. Zuccherato. "Faster attacks on elliptic curve cryptosystems", *Selected Areas in Cryptography, Lecture Notes in Computer Science*, Springer-Verlag, 1556, pp. 190-200, 1999.

[42] Scott a. Vanstone. "Compressed ECDSA Signatures", *United States Patent Application Publication*, App. No. 11/939, 022, Pub. No. US 2010/0023775 A1, Pub. date 28 Jan 2010.

[43] A. D. Sutter, J. Deschamps, J. L. Imana. "Efficient Elliptic Curve Point Multiplication Using Digit-Serial Binary Field Operations", *IEEE Transaction on Industrial Electronics*, Vol. 60, Issue 1, pp. 217-225, 2013.

[44] GSM, http://en.wikipedia.org/wiki/Gsm

[45] GSM History, www.gsm-history.org

[46] Time to Confirm some Mobile User Numbers: SMS, MMS, Mobile Internet, M-News, Jan 2011, http://communities-dominate.blogs.com/brands/ 2011/01/time-to-confirm-some-mobile-user-numbers-sms-mms-mobile-internet-m-news.html

[47] Neetesh Saxena, Narendra S. Chaudhari. "Secure Encryption with Digital Signature Approach for Short Message Service", *2012 World Congress on Information and Communication Technologies, Trivandrum, Kerala*, India, pp. 803-806, 30 Oct – 2 Nov 2012.

[48] Subir Biswas, Jelena Mi˘si´c. "A Cross-layer Approach to Privacy-preserving Authentication in WAVE-enabled VANETs", *IEEE Transactions on Vehicular Technology*, Vol. PP , No. 99, pp. 1-12, 2013.

[49] Liu Yongsheng, Li Jie, M. Guizani. "PKC Based Broadcast Authentication using Signature Amortization for WSNs", *IEEE Transactions on Wireless Communications*, Vol. 11 , No. 6, pp. 2106-2115, June 2012.

Author Biographies

Neetesh Saxena received B. Tech degree from Uttar Pradesh Technical University (UPTU) Lucknow, India, and M. Tech degree from Guru Gobind Singh Indraprastha University (GGSIPU) Delhi, India. He has been worked with IMS Engineering College Ghaziabad, UP, India for six years and currently working as a research scholar in department of Computer Science and Engineering at Indian Institute of Technology (IIT) Indore, India. His current research interest includes Cryptography and Network Security, Network Protocols, Wireless Networks, and Mobile Computing. He is a member of several reputed professional bodies including IEEE, ACM, CSI, and CSTA.

Narendra S. Chaudhari has completed his undergraduate, graduate, and doctoral studies at Indian Institute of Technology (IIT), Mumbai, India. He is currently with the Department of Computer Science & Engineering, Indian Institute of Technology Indore, India as a Professor. Earlier, since 2001 to July 2009, he was with the School of Computer Engineering, Nanyang Technological University (NTU) Singapore. He has been invited as a keynote speaker in many conferences. He has been referee and reviewer for a number of premier conferences and journals including IEEE Transactions, Neurocomputing, etc. He has more than 240 publications in top quality international conferences and journals. His current research work focus is on network protocols and security, algorithms, game AI, grammar learning, and novel neural network models like binary neural nets and bidirectional nets. His research interests include parallel computing, optimization algorithms, and theoretical computer science. He is Fellow of the Institution of Engineers, India (IE-India), as well as Fellow of the Institution of Electronics and Telecommunication Engineers (IETE) (India), Senior member of Computer Society of India, Senior Member of IEEE, USA, Member of Indian Mathematical Society (IMS), Member of Cryptology Research Society of India (CRSI), and many other professional societies.