DOI: 10.21917/ijct.2011.0063

USING MODIFIED STERN SERIES FOR DIGITAL SIGNATURE AUTHENTICATION IN ELLIPTIC CURVE CRYPTOGRAPHY

Latha Parthiban[1] and Nivetha Shree[2]

[1,2]*Department of Computer Science and Engineering, SSN College of Engineering, Tamil Nadu, India*

E-mail: [1]lathaparthiban@yahoo.com

Abstract

This paper presents the generation of digital signature along with message recovery based on Elliptic Curve Cryptography (ECC) and knapsack algorithm. In digital signature along with message recovery scheme, signature alone is sent and message is recovered from the signature (r, s). ECC provides greater security with less key size, when compared to integer factorization and discrete logarithm system. As the strength of knapsack algorithm depends on the selection of the series, the proposed algorithm uses modified Stern series which not only reduces the time complexity but also provides better security.

Keywords:

Message Recovery, ECC, Knapsack Algorithm, Modified Stern Series

1. INTRODUCTION

Digital signature authenticates the identity of message sender to ensure that the message sent is unchanged. It also ensures message integrity, authentication and confidentiality about the origin of a message. Digital signature along with message recovery schemes reduce transmission costs, because the message is contained in the signature itself and separately message and signature need not be sent again. It is also suitable for key exchange applications, due to the small size of the key. In this paper, we propose a new digital signature scheme along with message recovery using modified Stern series in knapsack algorithm. Exponential series is used in existing method which has higher computational time when compared to modified Stern series.

The proposed method has four levels of authenticated encryption. First level is based on elliptic curve signature, second level is applying knapsack value for the signing message, third level is encrypting using receiver's public key and fourth level is again encrypting using secret key generated based on elliptic curve differ- Hellman algorithm. The proposed method provides high security with reasonable computational cost. It is computationally infeasible for the intruders to find the private key from the publicly known domain parameters due to the difficult in computing Elliptic Curve Discrete Logarithm Problem (ECDLP).

In this paper, section 2 deals with literature survey, section 3 deals with the proposed new digital signature scheme which applies knapsack on ECC with usage of modified Stern series in knapsack algorithm, section 4 deals with experimental results which shows difference in execution time between exponential and Stern series and in section 5 the concluding remarks are provided.

2. LITERATURE REVIEW

In RSA [2], message was given as input to the hash function that produced the desired hash code which was encrypted using sender‘s private key to generate the signature. Message and signature were transmitted to the receiver and the sender's public key was used for decrypting the signature. Signature was accepted as authenticated, if the calculated hash code and the decrypted signature are same. ElGamal [3] proposed digital signature scheme based on the difficulty of computing discrete logarithms over a finite field with a large prime.

National Institute of Standards and Technology [4] published the Digital Signature Algorithm (DSA) in the Digital Signature Standard (DSS) to generate digital signature. It cannot be used for encryption or key exchange, but RSA can be used for both encryption and digital signature. Nyberg and Rueppel's approach [5] was based on the same principles as DSA along with the implementation of message recovery. In Hsu-Wu scheme [6], any cipher text of a signature (sign first, then encrypt) for a message was sent to a specified group of verifiers. It follows the *(t, n)* threshold scheme in order to decrypt the cipher text of signature. In (*t, n*) threshold scheme, any *t* out of *n* verifiers in the group shared the responsibility for message recovery.

ECC, introduced by Koblitz and Miller [11] obtained its high level of security from the concept of the ECDLP. In ECDLP, it is difficult to determine k when Q and P are given which denotes the points in elliptic curve where Q=kP. ECC provides better level of security with less key size and higher computational efficiency. ECC 160 bit provides same level of security when compared to RSA -1024 bit. ECC can be used in resource-constrained device like cell phone, smart cards etc. Chen et al [7] combined one way hash function and the identification scheme by Popescu [10] based on zero-knowledge, digital signature scheme. The design of one-way hash function is given with two characteristics: One is that the output will be of fixed length instead of the various length of input. The other is that the length of message with the signature can be reduced into a shorter digest through the hash function.

In Wu–Lin's [8] approach, both authentication of the public key and verifying the signature can be done in one step with concept of self certified public key cryptosystem. Convertible authenticated encryption [8] based on ECC provides the computational secrecy, i.e., the cipher text is computationally distinguishable with respect to two candidate messages. In knapsack based ECC approach [9], message to be encrypted was converted into ASCII value, and then it is given as input to ECC algorithm. In order to provide better security, knapsack algorithm was applied in encrypted message and then transmitted to receiver. In receiver side, reverse knapsack process and decryption was done to recover the original message. It is combination of encoding and decoding process in knapsack and reverse knapsack algorithm respectively.

3. PROPOSED METHOD

The proposed scheme is divided into three phases: Initial Phase, Signature Generation and Message Recovery.

3.1 INITIAL PHASE

During initial phase, server selects domain parameters and all clients compute the public and private keys. Server's database contains information about clients such as their ID and password. Clients transmit their ID and password to server for authentication. Server checks its database to verify whether the client has given the proper password for the corresponding ID, then the server accepts the client as an authenticated client. Domain parameters for elliptic curve over Fp are p, a, b, Gm and n. p is the prime number defined for finite field Fp. a and b (0,p-1)are the parameters defining the curve $y^2 = x^3 + ax+b$ mod p. Gm is the generator point (x_G, y_G), a point on the elliptic curve chosen for cryptographic operations and 'n' denotes order of the elliptic curve. Standards for Efficient Cryptography (SEC), provides predefined values for domain parameters {112,128,160,192,224,256,384,521 bits} which have a standard curve. In this approach, 128 bit prime field is considered and then corresponding values for p, a, b, n, G are chosen from SEC. After server selects the domain parameters it is sent to all authenticated clients. Clients compute private and public keys with help of domain parameters. Clients selects a random number '*x*' ε [1, p-1] where x is private key, then computes public key using,

$$y = x.G \{G + G + \ldots\ldots + x\text{times}\} \tag{1}$$

Public key of all authenticated clients is sent to server. If any client needs other client's public keys, then request should be sent to server in order to obtain it. For each session communication between any two clients, server generates a pair of private and public key {PR_K, PU_K}.

3.2 SIGNATURE GENERATION

During signature generation phase, if sender wants to sign a message and sent it to the receiver, then first sender has to select a random number k ε (1, p-1) and convert the message M to ASCII form. The sender computes r and s by:

$$r = M + (kG)_x \text{ modn} \tag{2}$$

$$s = \{k - \{\text{HASH(r) x private key of sender}\}\} \text{ modn} \tag{3}$$

In order to find r, k random numbers are considered with G base points. It gives $(kG)_x$ where only *x* coordinate are considered for calculations. Choose another random k if r=0 in Eq. (2). Apply SHA-256 algorithm to r value in order to produce the hash value. The inputs to knapsack algorithm are r and s along with series. In 1970, Merkle and Hellman inverted the knapsack algorithm which is a public key cryptography algorithm. It encodes and decodes the given messages. The series used in knapsack algorithm should be super increasing sequence. A super increasing sequence is one in which the next term of the sequence is greater than the sum of all preceding terms. It is easy to solve a super increasing knapsack by considering the total weight of the knapsack and comparing it with the largest weight in the sequence. If the total weight is less than the largest weight, then it is not in the knapsack. If the total weight is greater than the largest weight, then it is in the knapsack. Subtract the number from the total, and compare with the next highest number. This methodology is continued until the total reaches zero. If the total doesn't reach zero, then there is no solution.

In 1858, Stern defined the Stern sequence as follows:

$$\text{Stern (1)} = 1 \tag{4}$$

$$\begin{aligned} \text{Stern}(n) &= \text{Stern}(n/2)) \qquad - \text{if n is even} \\ \text{Stern}(n) &= \text{Stern}(n-1)/2) + \text{Stern}((n+1)/2) - \text{if n is odd} \end{aligned} \tag{5}$$

In Stern series, each row is created by inserting the sum of pair of consecutive elements into the previous row. Stern showed that gcd{s(n),s(n+1)}=1 and that for every pair of relatively prime positive integers (a,b) there exists a unique n>=1 with s(n)=a and s(n+1)=b. When (a,b)=(0,1), it is easy to see that each row of the diatomic array repeats as the first half of the next row down. Stern series is modified according to the super increasing sequence constraint. The r and s value is converted into binary form which acts as input to knapsack algorithm along with modified Stern series. The output of knapsack algorithm gives encoded r and s value.

The encoded r and s value is double encrypted with first layer of encryption using receiver's public key and second layer of encryption using server's public key which ensures confidentiality. Elliptic curve cryptography algorithm is used for encryption process. The steps involved in ECC encryption algorithm are: first transform the input into points. In Koblitz's method [11] for encoding input to points, select parameter 'k' where k is a random number. Then for each input 'm', compute x=mk+1and solve the corresponding y value ($y^2=x^3+ax+b$). In second step, select a random number that lies in the range of 0 to n-1. ECC algorithm transforms the input to points. In third step, compute the ciphertext as {kG, P_{ml}+k × public key of receiver} where P_{ml} denotes (*x*,y) coordinates which is output of Koblitz's method. Now the double encrypted signature is sent to server.

3.3 MESSAGE RECOVERY

When the signature reaches server, it is decrypted using server's private key. Elliptic curve cryptography algorithm is used for decryption process. The steps involved in ECC decryption algorithm are: first step, manipulate the first point (kG) with private key of receiver where kG be the first point and P_{ml} + kP_B be the second point. Second step involves computation of P_{ml} using,

$$P_{ml} + kP_B - n_B kG\, P_{ml} + k(n_B G) - n_B kG \tag{6}$$

Third step involves decoding of point back to input using Koblitz's method where for each point (*x*, *y*), compute m= (*x*-1)/k for decoding the point (*x*, *y*) to back the symbol '*m'*.

Then encrypt the signature using the secret key shared between server and receiver. The secret key is generated based on elliptic curve diffe-Hellman. In the elliptic curve Diffe-Hellman (ECDH) key agreement, the two communicating parties agree beforehand to use the same curve parameters and base point G. They generate their private keys Sa and Ca, respectively, and the corresponding public keys Sb = Sa × G and Cb = Ca × G. Both the client and server exchange their public keys, and each multiplies its private key with the other party's public key to derive a common shared secret key Sb × Ca = Ka = Sa × Cb. An attacker cannot determine the secret key. The encrypted signature is sent to receiver.

The receiver first decrypts using secret key generated with help of elliptic curve diffie-Hellman algorithm. Then it again decrypt using receiver's private key to obtain R and S value. Reverse knapsack algorithm is used in order to obtain the signature (r, s) from R and S value using formula R-n^m in an iterative fashion. If R- n^m > 0, then assign binary bit 1 at the m^{th} position. The current value is R=R-n^m. If value is negative then assign binary bit 0 and R value remains same. Subtract n^{m-1} from the current R. Depending upon whether it is +ve or –ve; assign 1 or 0 at the relevant bit position. Then continue this process until the series becomes null. This is will recover the binary bit pattern of r and s value. Then convert binary form of r and s to integer form. Receiver recovers message M using

$$M = r - \left\{sG + H(r)\{\text{public key of sender}\}_x\right\} \bmod n \qquad (7)$$

Proof

M= r-{sG+H(r){publickey of sender}$_x$} mod n

= M+$(kG)_x$-{(k-H(r)×sender'sprivatekey}G+H(r){public key of sender}$_x$ mod n}

= M + $(kG)_x$ – $(kG)_x$ + H(r)× sender's privatekey×G-H(r) public key of sender mod n

⇨ Message

3.4 SECURITY ANALYSIS

There are number of attacks against the proposed scheme and the two basic attacks against public-key digital signature schemes are key-only attacks and message attacks.

In key-only attacks, an intruder knows only the signer's public key. He attempts to derive the sender's private key from known domain parameters (*E*, *p*,*n* and *G* point, public key of sender). This attack is not possible in our proposed scheme. An intruder cannot derive y= *x*. G from known domain parameter, because obtaining sender's private key is difficult due to ECDLP.

In message attacks, intruder attempts to forge a digital signature to impersonate as sender, which is not possible in the proposed scheme as he has to know the knapsack series which is unique for every user along with sender's private key.

3.5 EXPERIMENTAL RESULTS

Once the elliptical curve is known, select a base point G which has (*x*, y) coordinates that satisfy the equation $y^2 = x^3 + ax + b$. The method of encryption and decryption with a single character message after applying knapsack series is illustrated in Appendix. As knapsack algorithm uses exponential series it may slow down the systems performance.

In the proposed system, use of modified Stern series in knapsack algorithm instead of exponential series reduces the time complexity effectively. Stern series provides better security when compared to exponential series because if any one value in series is known to intruders successive terms can be easily found which is not possible in Stern series. There is also reduction in time complexity while using Stern series. The computational time of different authenticated encryption schemes for signature generation phase and signature verification phase are shown in the Table.1 and Table.2 respectively.

Table.1. Computational Time of Different Authenticated Encryption Schemes for signature generation phase

Schemes	**Signature Generation Phase (milliseconds)**	
Chen et al [7]	$2T_{ECmul} + T_{ECadd} + T_{mul} + T_h$	91.825776
Hsu-Wu [6]	$3T_{exp} + T_{mul}$	7.947362
Wu-Lin[8]	$5T_h + 2T_{inv} + 3T_{mul} + T_{ECadd} + 4T_{ECmul}$	194.79795
Knapsack based ECC[1]	$T_{ECmul} + T_{mul} + T_h + T_{KV}$	48.658936

Table.2. Computational Time of Different Authenticated Encryption Schemes for signature verification phase

Schemes	**Signature Verification Phase (milliseconds)**	
Chen et al [7]	$3T_{ECmul} + 2_{TECadd} + T_h$	134.771934
Hsu-Wu [6]	$3T_{exp} + (2t + 1)T_{mul+ (t-1)}T_{inv}$	12.474339
Wu-Lin[8]	$5T_h + T_{mul} + 2T_{ECadd} + 5T_{ECmul}$	230.974826
Knapsack based ECC[1]	$T_{ECmul} + T_{ECadd} + T_h + T_{inKV}$	47.356199

T_{ECmul} is used to indicate the time for multiplying a number by a point on the elliptic curve. T_{ECadd} is the time for the adding one point to another on the elliptic curve. T_{mul} is the time for multiplication. T_h is the time for executing hash function. $T_{\exp}$ is the time for exponentiation with mod*P*. T_{inv} is the time for inversion mod*P*. T_{KV} is the time for knapsack value generation. T_{inKV} is the time for inverse knapsack value generation.

For comparison between existing and proposed algorithm, message of size 128 bytes is considered. The analysis of computational time needed for signing and verifying is shown in the Fig.1 and Fig.2 respectively based on the message description given in Table.3. In exponential series, n can be random number that is less than p bits which used in modulo arithmetic or n=pk where k is random integer.

Table.3. Description of messages and its representation

Description	**Representation**
!@#$%^&*()	A
Password SSN college	B
Password pin number 0234	C
Mobile sms banking application	D
Knapsack based ECC using Stern series for digital signature authentication	E

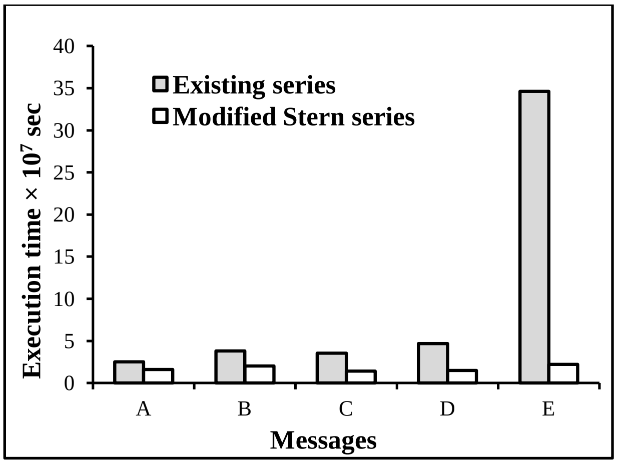

Fig.1. Comparing computational time of exponential and modified Stern series for signature generation

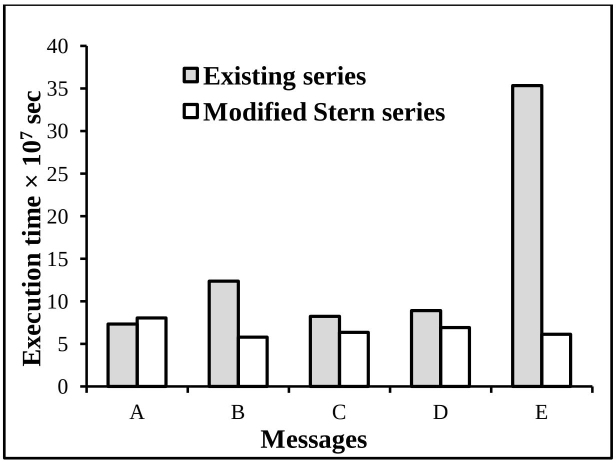

Fig.2. Comparing computational time of exponential and modified Stern series for message recovery

3.5.1 Application in mobile banking:

The algorithm proposed is implemented in mobile banking. In traditional system to obtain mobile banking service, customers have to go to the bank to submit the registration form by giving their mobile number, account number and transaction details. Each customer is then given a 4-digit number (ATM pin number) for authentication through postal communication which may not be confidential. After an analysis of the traditional banking services [12], a system that would provide better security is proposed. This system is called the ECC banking module which receives the text messages from the clients/banks and processes them and sends the output back to the banks/users as and when required. This ECC banking module provides secure data encryption and decryption using public key cryptography. Instead of obtaining the PIN number/password through post, with help of ECC banking module, it is possible to securely transmit it as shown in Fig.3. A digital signature using ECC technology used in each module provides message authentication, message integrity and non-repudiation. Mobile SMS banking in India does not provide these advantages. Mobile banking in android operating system has been implemented successfully with ECC banking module.

3.5.1.1 Implementation of Mobile Banking in Android:

Bank server has the ATM pin number for all customers along with corresponding customer's Unique Identification Number (UID). UID is a recently finalized initiative by the Government of India to create and manage a centralized identification system for all the adult citizens and residents of India, for a variety of identification purposes. Bank server will sent an alert to customer that he/she would receive pin number within few minutes. When the bank sent the pin number, at that time if the mobile is possessed by the third party other than the desired customer, pin number is known to third party. In order to avoid that, an alert is sent to customer along with request of customer UID.

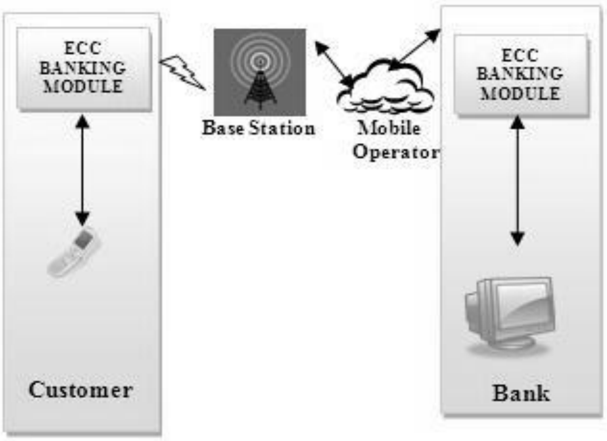

Fig.3. Mobile Banking in Android

After the customer receive that alert, bank will transmit the corresponding UID of the customer. Bank server would check whether the customer has sent the correct UID and then bank generate the public key based on ECC which is sent to customer. When the customer receives the public key from bank, they generate their public key and transmit to bank. Once the bank receive the public key of customer, they generate signature with pin number as input. Then the signature is encoded using knapsack algorithm. Double encryption is then performed on the output of the knapsack algorithm. First layer of encryption is done using customer's public key and second layer of encryption is done using secret key based on elliptic curve diffe-Hellman. The double layer of encrypted signature (pin number) is then sent to the customer securely.

ECC banking module has to be embedded in bank server and in the customer handset. When the customer receives the signature, ECC banking module would perform double decryption using secret key and private key. Then the decrypted signature is given to reverse knapsack algorithm in order to decode it. After decoding, PIN number is recovered from the signature with highest level of security and is displayed in the customer's handset.

4. CONCLUSION

In this paper, generation of digital signature based on ECC using modified Stern series has been proposed which provides confidentiality, authentication and non repudiation. The time complexity has been reduced drastically while using modified

Stern series in knapsack algorithm, when compared to the existing exponential series [1]. In mobile banking, secure transmission of ATM pin number with ECC banking module embedded in bank server and in customer handset has been implemented in Android.

APPENDIX

Consider the elliptic curve

$$y^2 \bmod 487 = (x^3 - 5x + 25) \bmod 487.$$

The base point G is selected as (0, 5) which has the smallest x, y co-ordinates that satisfies the elliptic curve. Let P_m be another affine point (can be G itself) choosen as 1,316. Generate the secret integer k, and private key nB of recipient B, assumed as 225, and 277 respectively.

Let the plaintext be “S”, with ASCII value 83. Therefore,

$P_B = nBG = 277(0, 5) = (260, 48)$

$P_{m1} = 83(1, 316) = (475, 199)$

$kP_B = 225(260, 48) = (212, 151)$

$P_{m1} + kP_B = (475, 199) + (212, 151) = (51, 58)$

$kG = 225(0, 5) = (99, 253).$

Encrypted version of the message is: {(99, 253),(51, 58)}, where $x_1 = 99$, $y_1 = 253$, $x_2 = 51$, and $y_2 = 58$.

Apply knapsack algorithm using a_i vector

$a_i = 1, 5, 25, 125, 625, 3125, 15625$ (where n = 5)

$x_{99} = 99$ −! 1100011 {binary value of 99}.

Therefore, $S[x_1] = \sum_{i=1}^{m} a_i x_i$

[99] = 1 + 5 + 0 + 0 + 0 + 3125 + 15625 = 18756.

Similarly, S[253] = 82031

S[51] = 3756

S[58] = 656.

Therefore the message transmitted is (18756, 82031), (3756, 656). The recovery bit pattern for x_{99} is

$18756 - 15625 = 3131 \rightarrow 1$

$3131 - 3125 = 6 \rightarrow 1$

$6 - 625 = -\text{ve} \rightarrow 0$

$6 - 125 = -\text{ve} \rightarrow 0$

$6 - 25 = -\text{ve} \rightarrow 0$

$6 - 5 = 1 \rightarrow 1$

$1 = 1 \rightarrow 1.$

Hence, x_{99} = 1100011 (bottom up). Similarly other coordinates are recovered by applying reverse knapsack algorithm. Hence the encrypted version is recovered as,

{(99, 253), (51, 58)}.

From this P_m should be retrieved, using B’s private key nB.

277(99, 253) = (212, 151)

$P_{m1} = (51, 58) - (212, 151) = (475, 199).$

Apply discrete logarithm in order to get the ASCII value of “S”.

S(1, 316) = (475, 199).

Therefore, S = 83 and the character “S” is retrieved. Similarly all other characters are found.

REFERENCES

[1] R. Rajaram Ramasamy and M. Amutha Prabakar, “Digital Signature Scheme with Message Recovery Using Knapsack-based ECC”, *International Journal of Network security*, Vol. 12, No. 1, pp. 15– 20, 2011.

[2] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public key cryptosystems”, *Communications of the ACM*, Vol. 21, pp. 120-126, 1978.

[3] T. ElGamal, “A public-key cryptosystem and a signature scheme based on discrete logarithms”, *IEEE Transactions on Information Theory*, Vol. IT-31, pp. 469-472, 1985.

[4] National Institute of Standards and Technology (NIST), “The digital signature standard proposed by NIST”, *Communications of the ACM*, Vol. 35, No. 7, pp. 34-40, 1992.

[5] K. Nyberg and R. A. Rueppel, “A new signature scheme based on the DSA giving message recovery”, *ACM Computer and Communications Security*, Vol. 1, pp. 58-61, 1993.

[6] C.L. Hsu and T. C. Wu, “Authenticated encryption scheme with (t, n) shared verification", *IEEE Proceedings - Computers and Digital Techniques*, Vol. 145, No. 2, pp. 117-120, 1998.

[7] T. S. Chen, G. S. Huang, T. P. Liu, and Y. F. Chung, “Digital signature scheme resulted from identification protocol for elliptic curve cryptosystem”, *Proceedings of IEEE TENCON'02 Region 10 Conference on Computers, Communications, Control and Power Engineering*, Vol. 1, pp. 192-195, 2002.

[8] T. S. Wu and H. Y. Lin, “ECC based convertible authenticated encryption scheme using self-certified public key systems”, *International Journal of Algebra*, Vol. 2, No. 3, pp. 109-117, 2008.

[9] R. Rajaram Ramasamy, M. A. Prabakar, M. I. Devi, and M. Suguna, “Knapsack based ECC encryption and decryption”, *International Journal of Network Security*, Vol. 9, No. 3, pp. 218-226, 2009.

[10] C. Popescu, “An Identification Scheme Based on the Elliptic Curve DiscreteLogarithm Problem,” *Proceedings 4th International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region*, Vol. 2, pp. 624-625, 2000.

[11] N. Koblitz, “Elliptic curve cryptosystems”, *Mathematics of Computation*, Vol. 48, pp. 203-209, 1987.

[12] Ranbir Soram, “Mobile SMS Banking Security Using Elliptic Curve CryptoSystem” *International Journal of Computer Science and Network Security*, Vol .9, No.6, pp. 30-38, 2009.