A Secure Off-line Electronic Payment System Based on Bilinear Pairings and Signcryption

Debasis Giri and Arpita Mazumdar

Abstract—In this paper, we have designed an off-line e-cash payment system based on bilinear pairings for low-value transaction. We use the concept of proxy signcryption for communication among the entities. In our system, the token is issued & authenticated by Bank. Customer delegates the signing capability to Merchant. Bank verifies the original signer (customer) and proxy signer (Merchant) and ensures the originality of the transaction. Unlike the existing e-payment system question of double spending of e-cash arises because each transaction is made uniquely identifiable. Hence, no separate protocol is needed to check double spending. The proposed scheme provides anonymity, authenticity, confidentiality and fairness.

Index Terms—Off-line; Proxy Signature; Security; Electronic Payment; Bilinear pairings.

I. INTRODUCTION

The growth of the Internet, in the last years, has created an electronic market place for goods and services. Information or more generally intangible goods is gradually becoming important. Electronic payment system plays a crucial role, acts as a backbone of this virtual market place. Hence, the need for more efficient electronic payments has become an essential fact. The aim of this paper is to present a digital e-cash system which is devoid of the problems inherent in such systems. The problems encountered in such systems are as follows:-

1: Problem due to PC hard drive crash:-
As digital cash is often stored in user's hard drive, there would have no way to imburse since the bank does not link the money to the user.

2: Bank has to maintain a large database for storing the coins series number to keep track of double spent coins.

3: Problem of security:-
Often in digital cash system customer account number, password are send without encryption via e-mail. There is a security breach due to Man-in- middle attack.

In this work, we present an off-line electronic payment system in which the payment instrument is token (which is an electronic wallet, it could be stored in a hard drive of a personal computer) containing electronic cash (E-cash). In off-line transaction, merchant submits several verification tasks to Issuing bank for verification after a certain period of time (for example, at the end of every day). The proposed scheme maintains a backup of updated tokens (payment instrument) in a Bank server which eliminates problem 1.

Unlike the existing e-payment system question of double spending of e-cash arises because each transaction are made unique having unique transaction id. Therefore no need for additional storage for spent coins. All transaction detail are signcrypted and hence secure. Hence, eliminates problem 2 and 3 respectively.

To transfer these long messages [4] by merchant to Issuing bank, a random number generator generating a random number of arbitrary length is designed and is used to encrypt long confidential message. The random number can perform some logical operations (for an example, XOR) with the transmitted bits of long message. Many authors proposed schemes [9, 10, 11, 12, 13] for electronic payment systems.

II. PROPOSED SCHEME

An e-cash system is a set of entities with their interactions, exchanging e-cash and goods. Our system has three entities:

- Customer(C): purchases goods or services from the merchant using the e-cash.
- Merchant(M): sells goods or services to the customer, and deposits the e-cash to the bank.
- Bank(B): issues the e-cash and maintains the bank account for customers and merchants.

There are also three protocols in the system: withdrawal, payment and deposit.

The customer withdraws token from the bank and pays token to the merchant. The token is readable only. The merchant get token from customer and deposits it in the bank. The bank manages customer accounts, issues and updates the token. No separate protocol is required to trace a dishonest customer. The following Figure-1 depicts the token life cycle in the proposed system. In a transaction, the following events take place :-

At first, setup is constructed by a central authority.

1 : (C→B) The customer requests Bank for the withdrawal of electronic coin (token) from his account.

2 : (B→ C) Bank issues signed token to C and securely send to C and decreases appropriate amount from the customer's account.

3 : (C→ M) C inserts token, chooses an item from M's home page and sends signcrypted order information (OI) to M. C also sends hash form of token information duly signed which later helps B to verify C. Customer is the original signer, he delegates the signing capability to merchant and the bank is the verifier.

4 : (M→ C) M sends hash value of SEQNO. of token duly signed as acknowledgement to C. M also delivers products to customer. Transfer of products may be immediate (for intangible goods) or delayed (for tangible goods).

5 : (M→B) Merchant appends price details, own account number (MAC) to OI and forward the signcrypted

Manuscript received on January, 2013.

Debasis Giri, Department of Computer Science and Engineering, Haldia Institute of Technology, Haldia-721657, India.

Arpita Mazumdar, Department of Master of Computer Applications, St. Mary's Technical Campus Kolkata, Kolkata-700 126, India.

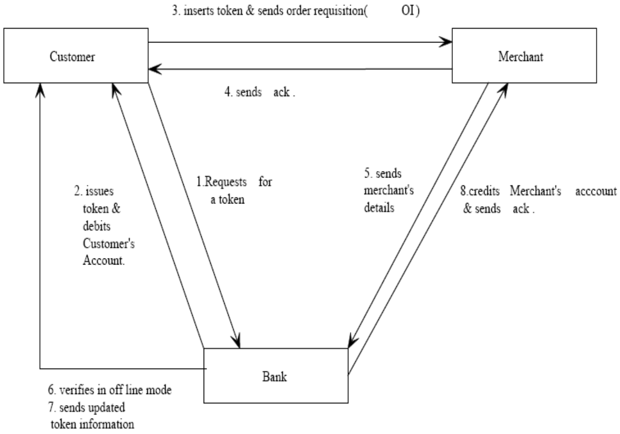
3. inserts token & sends order requisition(OI)
Customer
Merchant
4. sends ack .
2. issues token & debits Customer's Account.
1.Requests for a token
5. sends merchant's details
8.credits Merchant's acccount & sends ack .
Bank
6. verifies in off line mode
7. sends updated token information

Figure 1: Block diagram of proposed scheme

(with proxy signature) modified OI to B along with the signed hashed token.

6 : (Verification by B) Bank verifies proxy signer merchant as well as original signer customer and ensure that OI is genuinely placed by the customer. Bank retrieves customer token details by matching (TOKEN _ID||SEQNO.). Also performs hashing and check whether sent hashed token value is same to the hash value of the stored token

7 : (B $\rightarrow$ C) After verification Bank sends update information to C via e-mail. Updated token information is kept in a server. Valid customer after a certain period of time can download the updated token details. Token value, SEQNO., TS are updated.

8: (B $\rightarrow$ M) Bank credits the account of merchant and sends acknowledgement.

[**Note:** X $\rightarrow$ Y means X sends some information to Y.]

In our system, the format of the token information (TI) is shown below :-

A/C NO	TOKENID	SEQNO	EXPIRES	TS	VALUE	PROPS

A/CNO :- Account number of customer
TOKENID :- Unique identification number of a token
SEQNO. :- Unique sequence number for each transaction
EXPIRES :- Lifespan of a token.
TS :- Time stamp of a transaction.
VALUE :- Monetary value PROPS-other information.

Moreover the format of Order Information(OI) is shown below

TOKENID	ITEMCODE	SEQNO

where ITEMCODE - unique code of an item for sale

III. DESCRIPTION

In this section, we describe a secure and efficient off line electronic payment system based on bilinear pairings and proxy signature schemes. Here we use the proxy signature scheme of Fangguo Zhang and Kwangjo Kim [3] with additional feature of signcryption.

Proxy signatures are very useful tools when one needs to delegate his/her signing capability to other party. In our system, Customer delegates his signature capability to merchant. The Bank verifies the original signer, the customer and the proxy signer, the merchant and get assure that the item order requisition is genuinely placed by the valid customer. In this section, we present an ID-based proxy signature from bilinear pairings which is based on Hess 's [5, 3] ID based signature and propose an ID based signcryption .

In the following, we describe the Setup phase of the proposed scheme.

Setup phase: ID based public key setting involves a key generating center (KGC) and users. G_1 is a cyclic additive group of prime order q. Let P be a generator of G_1. G_2 is a cyclic multiplicative group of same order of G_1. bilinear pairing is given by e : $G_1XG_1 \rightarrow G_2$. Cryptographic one way hash functions are as follows:

H_1 : $\{0,1\} \rightarrow G_1{}^*$;
H_2 : $G_2{}^* \rightarrow \{0,1\}$;
H_3 : $\{0,1\}^* \text{ x } G_2 \rightarrow Z_q$;
H_4 : $\{0,1\}^* \rightarrow Z_q$;
h : $\{0,1\}^* \rightarrow \{0,1\}^n$; which takes a message of arbitrary bit length to a fixed length n.

There exists a central authority (CA) who publishes the sustem wide public parameters params ={ $G_{1,}$ $G_{2,}$ e, q, P, P_{pub}, $H_{1,}$ $H_{2,}$ $H_{3,}$ $H_{4,}$h} where P_{pub} =sP, s is random secret chosen from Z_q . s is the master key of CA. Let ID be an identity of user. Public key of the user is Q_{ID}= H_1(ID) and private key is d_{ID}=s. Q_{ID}. The key pairs of Bank, customer and merchant are (Q_{IDb},d_{IDb}) , (Q_{IDc},d_{IDc}) , (Q_{IDm},d_{IDm}) , respectively.

Let E_K be a symmetric encryption algorithm charaterised by the key k and D_K be the corresponding decryption algorithm.

Extraction Phase: The customer and the merchant submits their identities ID_c and ID_m respectively information to CA. CA computes the public key Q_{IDc}=H1($_{IDc}$) and private key d_{IDc}= s. Q_{IDc} for the customer, and public key Q_{IDm}=H1($_{IDm}$) and private key d_{IDm}= s. Q_{IDm} for the merchant.

Generation of proxy key: To delegate the signing capability, the customer make a signed warrant message where m_w= $ID_c||Id_m||exp||TS$, where exp is the lifespan of warrant message and TS is the creation time. Customer computers

r_A= $e(P,P)^i$ where i ε_R $Z_q{}^*$;
$C_A=H_4(m_w||r_A)$;
$U_A=C_A\, d_{IDc} + iP$;

And sends (m_w,C_A,U_A) to merchant at the beginning of each transaction session. Merchant verifies the validity of the signature on m_w. He computes $r_A=e(U_A,P)\ e(Q_{IDc},P_{pub})^{-C_A}$ and accepts this signature if and only if $C_A=H_4(m_w||r_A)$. If the signature is valid M computes the proxy key $S_p=C_Ad_{IDm}+U_{A.}$

Steps in the Figure -1 are elaborately described below:

Step 1 : The user C sends a request for token indicating the amount and his account number with the bank B (token issuing authority). These information are sent securely by signcrypting the message , say req= (A/cno., Amt)

1. x ε_R Z_q *
2. U=xP
3. $K=H_2(e(P_{pub},Q_{IDb})^x)$
4. $c=E_k(req)$
5. r= $e(U,P_{pub})$
6. s_i= xP_{pub} + r. d_{IDc}
7. $g_i=e(P,s_i)$
8. $v=H_3(req||c, g_i)$

C sends (U,c,r,v) to bank. After receiving it, bank verifies the authenticity of the received message. Bank performs the following:

1.$K' = H_2(e(U,d_{IDb}))$
2.$req= D_k (c)$
3.$g_i ' = e(U, P_{pub}) e(P_{pub},Q_{IDc})^r$
4.$v' = H_3(req\|c, g_i')$

Bank accepts the request if and only if v= v'.

[Correctness of K' = K]

$K' = H_2(e(U,d_{IDb}))$
$=H_2(e(xP,s.Q_{IDb}))$
$=H_2(e(sP,Q_{IDb})^x)$
$=H_2(e(P_{pub},Q_{IDb})^x)$
$= K$

[Correctness of $v_1 = v_1$ ']

$gi ' = e(U,P_{pub}) e(P_{\ pub},Q_{IDc})^r$
$=e(U,P_{pub}) e(sP,Q_{IDc})^r$
$=e(xP,P_{pub}) e(P,s.Q_{IDc})^r$
$=e(P,x. P_{pub}) e(P,d_{IDc})^r$
$=e(P,x.P_{\ pub} + r.d_{IDc})$
$=e(P,s_i)$
$=g_i'.$

Hence $v= H_3(TI \| c_1, g_1) = H_3(TI \| c_1, g_1')= v'$

Step 2 : If C is a valid account holder then B issues a token (TI) and sends it to c securely by the following way:

1. $y \ \varepsilon_R \ Z_q$ *
2 $U_1=yP$
3 $K_1=H_2(e(P_{pub},Q_{IDc})^y)$
4 $c_1=E_{k1}(req)$
5 $r_1= e(U_1,P_{pub})$
6 $sign= yP_{pub} + r_1. d_{IDb}$
7 $g_1=e(P,sign)$
8 $v_1=H_3(TI\|c_1, g_1)$

B sends (U_1,c_1,r_1,v_1) to C. Verification of signcrypted message. After receiving it, C verifies the authenticity of the received message.

1 $K_1 = H_2(e(U_1,d_{IDc}))$
2 $TI= D_{k1} (c_1)$
3 $g_1 ' = e(U_1, P_{pub}) e(P_{pub},Q_{IDb})^{r1}$
4 $v_1' = H_3(TI\|c_1, g_1')$

Customer accepts the request if and only if v1= v1'.

Step 3 : After verification customer (C) signs the token and sends OI securely to merchant. Merchant checks the authenticity of the customer and retrieves the value of OI. Generation of the above information by C described below.

1. $z \ \varepsilon_R \ Z_q$ *
2. $U_2=zP$
3. $K_2=H_2(e(P_{pub},Q_{IDm})^z)$
4. $c_2=E_{k2}(OI)$
5. $r_2= e(U_2,P_{pub})$
6. $sign_1= zP_{pub} + r_2. d_{IDc}$
7. $g_2=e(P,sign_1)$
8. $v_2=H_3(OI\|c_2, g_2)$

C sends signcrypted OI (U_2 ,c_2,r_2,v_2) to M. Moreover C signs token information and send it (m, r_4,U_4,v_4) securely to m in the following way, where m=h(TI).

1. Choose $w \ \varepsilon_R \ Z_q$ *
2. $U_4=wP$
3. $r_4= e(U_4,P_{pub})$
4. $sign_3= w \ P_{pub} + r_4. d_{IDc}$
5. $g_4=e(P,sign_1)$
6. $v_4=H_3(m, g_4)$

Merchant verifies the signature of C by calculating

1. $g_4' = e(U_4,P_{pub})e (p_{\ pub},Q_{IDc})^r{}_4$
2. $v_4'=H_3(m,g_4')$

Merchant accepts and authenticates the customer if $v_4=v'_4$

Step 4: After verification, as an acknowledgement, M returned the hash value of SEQNO. Of token signed by himself and delivers products as per the transaction agreement. Merchant verifies that OI has come from customer as follows:

1. $K_2 = H_2(e(U_2,d_{IDm}))$
2. $OI= D_{k2} (c_2)$
3. $g_2 ' = e(U_2, P_{pub}) e(P_{pub},Q_{IDc})^{r2}$
4. $v_2' = H_3(OI\|c_2, g_2')$

If $v_2=v_2$' then customer is authentic.

After successful verification, M sends the signature of SEQNO. To C in the following manner:

1. Choose $t \ \varepsilon_R \ Z_q$ *
2. $U_3=wP$
3. $r_3= e(U_3,P_{pub})$
4. $sign_2= t. P_{pub} + r_3. d_{IDm}$
5. $g_3=e(P,sign_2)$
6. $v_3=H_3(msg, g_3)$, where msg= SEQNO.

Merchant sends (msg,r_3,U_3,v_3) to c as acknowledgement. C verifies merchant by

1. $g_3' = e(U_3,P_{pub})e (p_{\ pub},Q_{IDm})^r{}_3$
2. $v_3'=H_3(msg ,g_3')$

Step 5 : At the end of the day, Merchant forwards signed token (sent by customer) to B. also M appends its own account number (M_{AC}), price with OI, i.e., MOI=(OI| M_{AC}||price) and sends a bulk of information of transactions with the bank. Merchant makes a proxy signature on the individual transaction information on behalf of his customer and send it securely as follows:

1. choose $j \ \varepsilon_R \ Z_q$ *
2. $U_5= j.P$
3. $K_c=H_2(e(P_{pub},Q_{IDb})^j)$
4. $m_c=E_{KC}(MOI)$
5. $C_p=H_4(m_c\|r_p)$, where $r_p=e(P,P)^K{}_p$ and $K_p \ \varepsilon \ Z_q$ *
6. $U_p=C_pS_p+K_pP$

M sends proxy signature tuple <m_c, C_p,U_p, m_w,r_A,U_5 > to the bank.

Step 6: Verification by B After receiving the signature, bank verifies proxy
signer merchant as well as original signer customer and ensure that OI
genuinely placed by the customer.

Bank computes $r'_p = e(U_p, P)(e(Q_{IDc}+Q_{IDm}, P_{pub})^{H4(mw\|rA)}.r_A)^{-Cp}$ and
accepts the signature if and only if $C_p = H_4(m_c\|r_p)$. Also decrypts the
value of MOI as follows:

1. Calculate $K_c = H_2(e(U_{5,} d_{IDb}))$
2. MOI=D_{Kc} (m_c)

Bank retrieves customer token details by matching (TOKEN ID||SEQNO.).
Also performs hashing and check whether sent hashed token value sent by
customer is same to the hash value of the stored token.

Correctness of [$r'_p = r_p$]

$$
\begin{aligned}
r'_p &= e(U_p, P) (e (Q_{IDc} + Q_{IDm}, P_{pub})^{H4(mw\|rA)}.r_A)^{-Cp} \\
&= e(U_p, P) (e (Q_{IDc} + Q_{IDm}, P_{pub})^{C}{}_{A}.r_A)^{-Cp} \\
&= e(U_p, P) (e (C_{A.}(Q_{IDc} + Q_{IDm}), sP).r_A)^{-Cp} \\
&= e(U_p, P) (e (C_A(d_{IDc} + d_{IDm}), P).r_A)^{-Cp} \\
&= e(Up, P) (e (S_p -iP,P).r_A)^{-Cp} \\
&= e(U_p, P) (e (S_p, P) e (-iP, P).r_A)^{-Cp} \\
&= e(C_p.S_p + K_pP, P) (e (S_p, P) e (P, P)^{-i}.r_A)^{-Cp} \\
&= e(C_p.S_p + K_pP, P) (e (S_p, P) r_A{}^{-1} .r_A)^{-Cp} \\
&= e(C_p.S_p, P) e(K_pP, P) (e (S_p, P))^{-Cp} \\
&= e(Sp ,P)^{Cp} e(K_pP, P) (e (S_p, P))^{-Cp} \\
&= e(K_pP, P) \\
&= e(P, P)^{K}{}_{p} \\
&= r_p
\end{aligned}
$$

Step 7: After verification Bank sends update information to C via e-mail. Updated token information is kept in a server. Valid customer after a certain period of time can download the updated token details. Token value, SEQNO., TS are updated.

Step 8: Bank credits the account of merchant and sends acknowledgement.

IV. PSEUDO RANDOM NUMBER GENERATOR

The basic idea is to use CA (Cellular Automata) as a pseudo-random number generator (PRNG). The generated sequence is combined using XOR operation with the plain text. The result of that operation formed the cipher text. The secret key is the initial state of CA. The leftmost CA cells hold 128 bits which is actual the seed value (which is the secret key) of the generator. In the decryption process some pseudo-random sequence needed to be regenerated using the secret key and then combined with the cipher text. A CA is a computing model of complex System using simple rule. This can be viewed as a simple model of a spatially extended decentralized system made up of a number of individual components, called as cell. Each cell can be one or several final state. Cells are affected by neighbors' to the simple rule. Cellular Automata are highly parallel and discrete dynamical systems, whose behavior is completely specified in terms of a local relation. In Figure 2, we have used both linear and non linear rules of CA to achieve good randomness and cost effectiveness. For initial CA we used Rule 90,150 since it is shown in [7] that this is a necessary condition for the LHCA (linear hybrid cellular automata) to have a maximum length cycle. The complete details of LHCA can be found in [6, 7]. The corresponding combinational logic of rules 90 and 150 for CA can be expressed as follows:

Rule 90:- $t_i=t_{i-1} \oplus t_{i+1}$, where ⊕ denotes XOR operatopr.

Rule 150:- $t_i = t_{i-1}\oplus t_i \oplus t_{i+1}$

For next upper CA and lower CA cells we have chosen Rules (51,60,195,204) and Rules (90,102,153,165) respectively since it is proved in [8] that this combination of rules forms reachability tree which ensures a group CA. The complete details of non linear CA can be found in [8]. These two 128 random bits are XORed with the seed value, the result is the seed for the next iteration (next clock pulse). The 128 random bits from the two parallel CA cells are appended and form 256 random bits. These 256 random bits are concatenated in each iteration with the previous result and thus form a large random number.

Note: Merchant and Bank can share a secret key which is used as seed value in CA. At the end of the day, merchant can send the confidential message to Bank using XOR operation of message and random bits generated by CA.

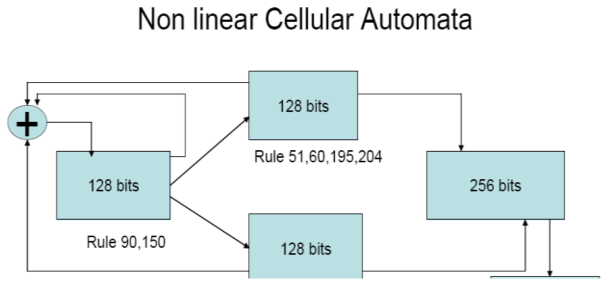

Figure 2: seudo Random number generator

V. SECURITY ANALYSIS

The proposed scheme is based on a cryptographic primitive called ``proxy signature'' using bilinear pairing in which message encryption and digital signature or proxy signature are simultaneously performed efficiently. A secure signcryption scheme should satisfy the following properties.

Unforgeability: It is computationally infeasible for an adaptive attacker to masquerade as the signcrypter is creating a signcrypted text.

Confidentiality: It is computationally infeasible for an adaptive attacker to find out any secret information from a signcrypted text.

Non repudiation: It is computationally feasible for a judge to settle a dispute between the signcrypter and the recipient in an event where the signcrypter denies the fact that he is the sender of the signcrypted text to the recipient.

Our proposed scheme possess all the three properties:

1. Message, cipher and signature are tightly coupled in verification phase. Therefore forgery attack is likely not to occur.
2. Confidentiality: Message is encrypted with asymmetric key and to find out any secret information from signcrypted text is computationally infeasible.
3. Non repudiation: As the signcrypter signs with its private key, therefore non repudiation exists.

Suppose an adversary tries to forge the signature of the sender. It chooses a secret key K' and encrypts its own message m', $c'=E_{K'}(m')$ and tries to forge a signcypted text by sending (U'=U, c', r'=r, v'=v). The receiver verifies by recalculating K'. For signature verification, receiver calculates g_i' and computes $v_{ver}=H_3(m'\|c', g_i')$. Now values of v_{ver} and v' are same is computationally infeasible. Value v calculated by the original signer does the valid verification. Message m, its cipher c, g_i (which is based on sender's signature) are tightly coupled for verification. Therefore, the attack is computationally infeasible.

VI. COMPARISON

In Table 1, we compare our scheme with previously published scheme namely Oros et al [2] scheme. It shows that our scheme is more secure than that of Oros et al scheme [2].

Table 1: Comparison between Oros et al and ours

SCHEME	REGISTRATION	WITHDR-AWAL	PAYMENT	DEPOSIT
Oros et al [2]	Every customer has to obtain certificates from Central Authority	ECC based blind signature	Coin is hashed and signed	Bank validates the signature on the coin. Detection of double spending needs a database search
Our	No such certificate required. Customers submit their IDs and private , public keys are computed by Central Authority	Signcryption using bilinear pairing	Coin is signcrypted	Bank verifies the original signer (C) and the proxy signer (M) and validity of token. Token can't be double spent and be updated.

VII. CONCLUSION

In our proposed scheme, we have used the technique of ID-based proxy signcryption from bilinear pairings which can save communication bandwidth compared with traditional schemes as pairing-based schemes feature a relatively small signature overhead. Moreover, to send the long confidential message to bank by merchant, we have designed a PRNG using CA. We have also described the security analysis of our proposed scheme.

REFERENCES

[1] Kevin Cattell and Shujlan Zhang, ``Minimal cost one-dimensional linear hybrid cellular automata of degree through 500". Journal of Electronic Tsting: Theory and Applications, 6:255-258, January 1995.

[2] H. Oros, C. Popescu Horea Oros, Constantin Popescu Department of Mathematics and Computer Science, University of Oradea Str. Universitatii 1, Oradea, Romania, ``A Secure and Efficient Off-line Electronic Payment System forWireless Networks".Int. J. of omputers, Communications & Control, Vol. V , No. 4, pp. 551-557, 2010

[3] Fangguo Zhang and Kwangjo Kim International Research center for Information Security (IRIS) Information and Communications University(ICU), 58-4 Hwaam-dong Yusong-ku, Taejon, 305-732 KOREA, ``Efficient ID-Based Blind Signature and Proxy Signature from Bilinear Pairings"

[4] Debasis Giri, Prithayan Barua, P. D. Srivastava and Biswapati Jana ``A Cryptosystem for Encryption and Decryption of Long Confidential Messages",ISA 2010,CCIS 76, PP 86-96,2010

[5] Florian Hess,``Efficient Identity Based Signature Schemes Based on Pairings",LNCS, Volume 2595/2003, 310-324,2003.

[6] K. Cattell and J.C. Muzio, "Synthesis of one-dimensional linear hybrid cellular automata," Submitted to IEEE Transactions on Computer-Aided Design, April 1994.

[7] M. Serra, T. Slater, J.C. Muzio, and D.M. Miller, "The analysis of one-dimensional linear cellular automata and their aliasing properties," IEEE Transactions on Computer-AidedDesign, vol. 9, pp. 767-778, July 1990.

[8] Sukanta Das , "Theory and Applications of Nonlinear Cellular Automata In VLSI Design", 2006

[9] Yuliang Zheng, ``Digital Signcryption or how to achieve ost(Signature \& Encryption)$ <<$ Cost (Signature) + Cost(Encryption)"- Monash University, Melbourne, Australia, CRYPTO '97 Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology,(LNCS 1294, Springer-Verlag, 1997), pp. 165-179.

[10] R. Sai Anand and CE Veni Madhavan, ``Online Transferable Ecash Payment System", INDOCRYPT 2000, LNCS 1977, pp. 93-103, 2000.

[11] Steve Glassman,Mark Manasse,Martín Abadi,Paul Gauthier and Patrick Sobalvarro, ``Millicent Protocol for inexpensive E.commerce".World Wide Web Journal, 4th WWW Conference Proceedings, p 603-618, December 1995.

[12] W H He and TC- Wu, ``Cryptanalysis and improvement of Peterson-Michels signcryption scheme",IEE Proc.-Comput. Digit. Tech., Vol. 146, No. 2, 1999.

[13] Min-Shiang Hwang and Pei-Chen Sung ``A Study of Micro-payment Based on One-Way Hash Chain", International Journal of Network Security, Vol.2, No.2, PP.81–90, Mar. 2006.

Debasis Giri is presently Professor in the Department of Computer Science and Engineering, Haldia Institute of Technology, Haldia-721657, India. He received his Ph. D degree from Indian Institute of Technology, Kharagpur 721 302, India 1n 2009. He did his masters (M. Tech and M. Sc) both from IIT, Kharagpur in 2001 and 1998 respectively. He has published more than 25 technical papers in the referred journals/conferences. His current research interests include cryptography, E-commerce security, Security in Wireless Sensor Networks and Security in VANET.

Arpita Mazumdar is presently Assistant Professor in the Department of Master of Computer Application of St. Mary's Technical Campus, Kolkata. She did her masters (M.Tech and MCA) degrees from Haldia Institute of Technology, Haldia-721657, India under West Bengal University of Technology in 2011 and Indira Gandhi National Open University in 2009 respectively. Her current research interests include Information security and E-commerce security.