Transferable e-cheques using Forward-Secure Multi-signature Scheme

N.R.Sunitha, B.B.Amberker and Prashant Koulgi

Abstract

With the modern world going on-line for all businesses, we need to transact with various business organizations all over the world using different modes of payment obtained through various Financial Institutions. We have considered the following bank transaction. A person X having an account in bank C issued a cheque for certain amount in favor of a person Y having an account in bank B. But, Y wants to issue a cheque for the same amount favoring Z. In normal course, Y need to deposit the cheque in bank B, wait for clearance and then issue a cheque in favor of Z. This consumes time as bank B sends the cheque to bank C for clearance.

In this paper, we propose a scheme in which a cheque is transferable. That is, the cheque in favor of Y can be reissued for the same amount to Z without presenting in the bank. Person Z can deposit the cheque directly in his bank, thus saving time and work load on banks. Further, the scheme can be used to transfer the cheque to any number of persons and only the last person deposits in the bank. In view of this, e-cheque can be used as e-cash to a limited extent. Our scheme is based on multi-signatures. We have augmented the multi-signature scheme to provide forward security. This guarantees the security of cheques signed in the past even if the signer's secret key is exposed today.

Keywords : Digital Signature, ElGamal Signatures, Serial multi-signature, Parallel multi-signature, Forward-Security.

I Introduction

E-cheques are a mode of electronic payments. This technology was developed couple of years ago and has been promoted by many of the financial institutions. E-cheques work the same way as paper cheques and are a legally binding promise to pay. The payer/account holder writes an e-cheque using a computer or other type of electronic device and transmits the e-cheque to the payee electronically. However, e-cheques are affixed with digital signatures. Cryptographic signatures on every e-cheque can be verified at all points.

The payer writes an e-cheque by structuring an electronic document with the information legally required to be in a cheque and digitally signs it. The payee receives the e-cheque over email or web, verifies the payer's digital signature, writes out a deposit and digitally signs it. The payee's bank verifies the payer's and payee's digital signatures and forwards the cheque for clearing and settlement. The payer's bank verifies the payer's digital signature and debits the payer's account.

The standard notion of digital signature [5, 2, 3] security is extremely vulnerable to leakage of the secret key which over the lifetime of the scheme may be quite a realistic threat. Indeed if the secret key is compromised any message can be forged. Forward-secure signature schemes, first proposed by Anderson in [2] and formalised by Bellare and Miner in [3] are intended to address the above limitation. A forward-secure digital signature scheme [1, 3, 6] is a method for creating digital signatures signed with secret keys changing with time periods, all of which can nevertheless be verified by the verifier using the same public key. An adversary with access to this public key and the secret key of some time period, will be unable to forge signatures for an earlier time period.

When a signature depends on more than one signer we call it a multi-signature. A multi-signature scheme [8, 9, 10]enables a group of signers to produce a compact, joint signature on a common document. As many applications require multiple signers to sign the same document, we propose to apply the concept of forward-security to multi-signatures. Using Forward-secure multi-signatures all signers of the document can guarantee the security of document signed in the past even if their secret key is exposed today. An adversary will not be able to forge such a multi-signature unless the secret key of all the signers are compromised in the same time period, which is practically not possible.

N.R.Sunitha is with department of Computer Science & Engg., Siddaganga Institute of Technology, Tumkur-572103, India (email:nrsunitha@gmail.com).

B.B.Amberker is with department of Computer Science & Engg., National Institute of Technology, Warnagal, India (email:bba@nitw.ac.in).

Prashant Koulgi is with department of Computer Science & Engg., Siddaganga Institute of Technology, Tumkur-572103, India (email:prashantkoulgi@yahoo.com).

Cheques once issued to a customer must be deposited in a bank for further processing. Generally there is no provision for a cheque to be transferred among customers i.e if a customer has a cheque in hand for a specified amount, he cannot give the same cheque to another customer for the same specified amount. He must first deposit the cheque in his bank and issue another cheque to the customer. We have come up with a proposal through e-cheques, to provide an option for a customer to transfer a cheque to another customer without depositing it in the bank . Each customer receiving the cheque is convinced that the cheque is from the intended sender. Only the last receiver of the e-cheque deposits it in the bank. All previous customers transfer the e-cheque off-line. This reduces the work load on bank to clear the e-cheques. Also e-cheques can be used like hard cash to some extent. We use the concept of Serial Forward-Secure multi-signatures which ensure forward-security of the document and allow signers to sign the same document serially .

In the section II we discuss ElGamal-like Signature Scheme [7], which is the basic signature scheme that we have considered and in the section III we make this scheme Forward-secure. In section IV we apply this forward-secure scheme for a group of signers who need to sign the same document. Here we discuss Forward-secure serial multi-signatures which ensure forward-security of the document and allow signers to sign the same document serially. In section V we explain the model to use the concept of forward-secure serial multi-signature to transfer e-cheques among customers. In section VI we give the security analysis of our scheme by considering the possible attacks against the multi-signature scheme and also discuss the forward-security of our scheme. Lastly in section VII we conclude.

II ElGamal-like Signature Scheme:

Recall that, the signature for the message m in the basic ElGamal scheme [5] with the secret key s and public key β is (y_1, y_2) where

$$y_1 = \alpha^k \mod p \tag{1}$$

where k is a random number chosen such that $0 < k < p-1$ and $gcd(k, p-1) = 1$. α is the randomly chosen generator of the multiplicative group Z_p^*.

$$y_2 = (H(m) - sy_1)k^{-1} \mod (p-1) \tag{2}$$

where H is a collision-resistant hash function [4].
The verification equation is given by

$$\alpha^{H(m)} = \beta^{y_1} y_1^{y_2} \mod p \tag{3}$$

In saying that our forward-secure scheme is based on a basic signature scheme, we mean that, given a message and the secret key of a time period, the signing algorithm is the same as in the basic signature scheme. But this approach does not work directly with basic ElGamal Signature scheme [5] because a verification equation satisfying forward security [3] was difficult to obtain. Therefore, we have made minor changes in the computation of signature generation. We have changed the computation of y_2 wherein product component of secret key s and y_1 is replaced by their corresponding sum (see equations (2) and (4)).

$$y_2 = (H(m) - (s + y_1))k^{-1} \mod (p-1) \tag{4}$$

The verification equation gets changed as follows:

$$\alpha^{H(m)} = \beta \;\; \alpha^{y_1} \;\; y_1^{y_2} \mod p. \tag{5}$$

We call this new signature scheme as ElGamal-like signature scheme [7].

Recall that the claim of security of the standard ElGamal signature scheme is based on the difficulty of computing discrete logarithms. The same security guarantee is obtained in the ElGamal-like Signature Scheme.

III Forward Secure ElGamal-like Signature Scheme

To specify a forward-secure signature scheme, we need to (i) give a rule for updating the secret key (ii) specify the public key and (iii) specify the signing and the verification algorithms.

Here are the details.

1. **Secret Key Updation**: Let p be a large prime. Choose α to be a large prime $< p$ such that

   $$\gcd(\alpha, p) = 1, \gcd(\alpha, \phi(p)) = 1, \gcd(\alpha, \phi^2(p)) = 1, \\ \ldots, \gcd(\alpha, \phi^{T-1}(p)) = 1$$

   where $\phi(p)$ is the totient function and $\phi^{T-i}(p) = \phi(\phi^{T-i-1}(p))$ for $1 \leq i \leq T-1$ with $\phi^0(p) = p$. The base secret key a_0 (this is the initialisation for the secret key updation) is chosen randomly in the range $1 < a_0 < p-1$.
   The secret key a_i in any time period i is derived as a function of a_{i-1}, the secret key in the time period $i-1$, as follows:

   $$a_i = \alpha^{a_{i-1} \bmod \phi^{T-i+1}(p)} \mod \phi^{T-i}(p) \tag{6}$$

   for $1 \leq i < T$. Once the new secret key a_i is generated for time period i, the previous secret key a_{i-1} is deleted. Thus an attacker breaking in period i will get a_i but cannot compute $a_0, \ldots, a_{i-1}$, because of difficulty of computing discrete logarithms.

2. **Public Key Generation:** In Bellare-Miner scheme, the public key is obtained by updating the base secret key $T+1$ times. However, we obtain the public key by executing the Secret Key Updation Algorithm T times as follows :

$$\beta = \alpha^{a_{T-1}} \mod p = a_T \mod p \tag{7}$$

3. **Signature Generation:** The signature generated in any time period i is $\langle y_{1,i}, y_{2,i}, y_{3,i} \rangle$. Additionally we have $y_{3,i}$. This parameter helps in obtaining the verification equation.
The computation of $y_{1,i}$ is

$$y_{1,i} = \alpha^k \mod p \tag{8}$$

where k is a random number chosen such that $0 < k < p$ and $gcd(k, (p-1)) = 1$.
The computation of $y_{2,i}$ is

$$y_{2,i} = (H(m+i) - (a_i + y_{1,i}))k^{-1} \mod (p-1) \tag{9}$$

where H is a collision-resistant hash function. While hashing, i is added to m to indicate the time period in which the message is signed.
The computation of $y_{3,i}$ is

$$y_{3,i} = \alpha^{a_i - \mathcal{A}(\alpha, T-i-1, a_i)} \mod p \tag{10}$$

where, by the notation $\mathcal{A}(\alpha, u, v) = \alpha^{\cdot^{\cdot^{\alpha^v}}}$ we mean that there are u number of α 's in the tower and the topmost α is raised to v, i.e in the above equation there are $(T-i-1)$ number of α's in the tower and the topmost α is raised to a_i.

Notice that the public key β can also be given in terms of a_i as,

$$\beta = \mathcal{A}(\alpha, T-i, a_i) \bmod p, \tag{11}$$

This relation gets employed in the verification of validity of the signature.

4. **Verification:** As for verification, a claimed signature $\langle y_{1,i}, y_{2,i}, y_{3,i} \rangle$ for the message m in time period i is accepted if

$$\alpha^{H(m+i)} = \beta \ \ y_{1,i}^{y_{2,i}} \ \ \alpha^{y_{1,i}} \ \ y_{3,i} \bmod p \tag{12}$$

else rejected.

We use the Forward-secure ElGamal-like signature scheme discussed in the previous section to design Forward-Secure Serial Multi-signature scheme [10].

IV The Forward-secure Serial Multi-signature Scheme

The Forward-secure Serial Multi-signature Scheme ensures forward-security of the document and allows multiple signers to sign the same document serially i.e. one after the other. Here signing order need not be predetermined. During this process each signer verifies the signature of his/her predecessor's and then signs the document by creating a partial multi-signature. The signature generated by the last signer will be the multi-signature which can be verified by any verifier with a single public key.

A Partial multi-signature generation and verification:

Any signer $U_j (2 \leq j \leq n)$ computes $(y_j, y'_{j,i}, y''_{j,i})$ as follows,

$$y_j = \alpha^{k_j} \mod p \tag{13}$$

where k_j is a random number chosen such that $0 < k_j < p-1$ and $gcd(k_j, p-1) = 1$.

$$y'_{j,i} = (H(m+i) - (a_{j,i} + y_j))k_j^{-1} \mod (p-1) \tag{14}$$

where H is a collision-resistant hash function.

$$y''_{j,i} = \alpha^{a_{j,i} - \mathcal{A}(\alpha, T-i-1, a_{j,i})} \mod p \tag{15}$$

and signs the message m by creating the partial multi-signature, $\langle(((\sigma_{j,1}, \sigma_{j,2}, \sigma_{j,3}), m)\rangle$ where

$$\begin{aligned}
\sigma_{j,1} &= \sigma_{j-1,1} . y_j^{y'_{j,i}} \\
&= y_1^{y'_{1,i}} \ldots y_j^{y'_{j,i}} \\
&= \alpha^{k_1(H(m+i)-(a_{1,i}+y_1))k_1^{-1}} \ldots \\
&\quad \alpha^{k_j(H(m+i)-(a_{j,i}+y_j))k_j^{-1}} \\
&= \alpha^{jH(m+i)-(a_{1,i}+\ldots+a_{j,i})-(y_1+\ldots+y_j)}
\end{aligned}$$

$$\begin{aligned}
\sigma_{j,2} &= \sigma_{j-1,2} . \alpha^{y_j} \\
&= \alpha^{y_1} \ldots \alpha^{y_j} \\
&= \alpha^{y_1+\ldots+y_j}
\end{aligned}$$

$$\begin{aligned}
\sigma_{j,3} &= \sigma_{j-1,3} . y''_{j,i} \\
&= y''_{1,i} \cdots y''_{j,i} \\
&= \alpha^{a_{1,i}+\ldots+} . \\
&\quad \alpha^{a_{n,i}-(\mathcal{A}(\alpha,T-i-1,a_{1,i})+\ldots\mathcal{A}(\alpha,T-i-1,a_{j,i}))}
\end{aligned}$$

This partial multi-signature is sent to the next signer U_{j+1}.

Any partial multi-signature received by a signer $U_j (2 \leq j \leq n)$ is verified using the following equation:

$$\alpha^{(j-1)H(m+i)} = \beta_{1\ldots(j-1)}.\sigma_{(j-1),1}.\sigma_{(j-1),2}.\sigma_{(j-1),3} \tag{16}$$

where the public key $\beta_{1\ldots(j-1)}$ is computed as the product of public keys of previous signers using the following equation:

$$\beta_{1\ldots(j-1)} = \beta_1.\beta_2 \ldots \beta_{j-1} \tag{17}$$

where β_1 is the public key of the initiator, β_2 is the public key of the second signer and so on.

The partial multi-signature generated by the last signer is the Forward-secure Serial Multi-signature of n signers which can be verified by any external verifier. The verification equation for the external verifier is

$$\alpha^{nH(m+i)} = \beta_{1\ldots n}.\sigma_{n,1}.\sigma_{n,2}.\sigma_{n,3} \tag{18}$$

Since

$$\begin{aligned} RHS &= \alpha^{\mathcal{A}(\alpha,T-i-1,a_{1,i})} \ldots \alpha^{\mathcal{A}(\alpha,T-i-1,a_{n,i})}. \\ &\quad \alpha^{nH(m+i)-(a_{1,i}+\ldots+a_{n,i})-(y_1+\ldots+y_n)}. \\ &\quad \alpha^{y_1+\ldots+y_n}.\alpha^{a_{1,i}+\ldots+a_{n,i}}. \\ &\quad \alpha^{-(\mathcal{A}(\alpha,T-i-1,a_{1,i})+\ldots\mathcal{A}(\alpha,T-i-1,a_{n,i}))} \\ &= \alpha^{nH(m+i)} \\ &= LHS \end{aligned}$$

a multi-signature of a group of n honest signers will therefore be accepted.

B Signature generated by initiator U_1

The signature generated by the initiator U_1 for the message m in Forward-secure ElGamal-like scheme with the secret key $a_{1,i}$ in time period i is $(y_1, y'_{1,i}, y"_{1,i})$ where

$$y_1 = \alpha^{k_1} \mod p \tag{19}$$

where k_1 is a random number chosen such that $0 < k_1 < p-1$ and $gcd(k_1, p-1) = 1$.

$$y'_{1,i} = (H(m+i) - (a_{1,i} + y_1))k_1^{-1} \mod (p-1) \tag{20}$$

where H is a collision-resistant hash function.

$$y''_{1,i} = \alpha^{a_{1,i} - \mathcal{A}(\alpha,T-i-1,a_{1,i})} \mod p \tag{21}$$

The partial signature $\langle((\sigma_{1,1}, \sigma_{1,2}, \sigma_{1,3}), m)\rangle$ is generated by the initiator and sent to the second signer U_2 where

$$\begin{aligned} \sigma_{1,1} &= y_1^{y'_{1,i}} \\ &= \alpha^{k_1(H(m+i)-(a_{1,i}+y_1))k_1^{-1}} \\ &= \alpha^{H(m+i)-(a_{1,i}+y_1)} \end{aligned}$$

$$\sigma_{1,2} = \alpha^{y_1}$$

$$\begin{aligned} \sigma_{1,3} &= y''_{j,i} \\ &= \alpha^{a_{1,i}-(\mathcal{A}(\alpha,T-i-1,a_{1,i}))} \end{aligned}$$

C Initial Signature Verification and partial multi-signature generation:

The second signer U_2 verifies the signature received by the initiator U_1 using the following equation :

$$\alpha^{H(m+i)} = \beta_1 \;\; \sigma_{1,1} \;\; \sigma_{1,2} \;\; \sigma_{1,3} \mod p \tag{22}$$

where β_1 is the public key of the initiator. If the initiator's signature is verified, then the second signer U_2 computes $(y_2, y'_{2,i}, y''_{2,i})$ as follows,

$$y_2 = \alpha^{k_2} \mod p \tag{23}$$

where k_2 is a random number chosen such that $0 < k_2 < p-1$ and $gcd(k_2, p-1) = 1$.

$$y'_{2,i} = (H(m+i) - (a_{2,i} + y_2))k_2^{-1} \mod (p-1) \tag{24}$$

where H is a collision-resistant hash function.

$$y''_{2,i} = \alpha^{a_{2,i} - \mathcal{A}(\alpha,T-i-1,a_{2,i})} \mod p \tag{25}$$

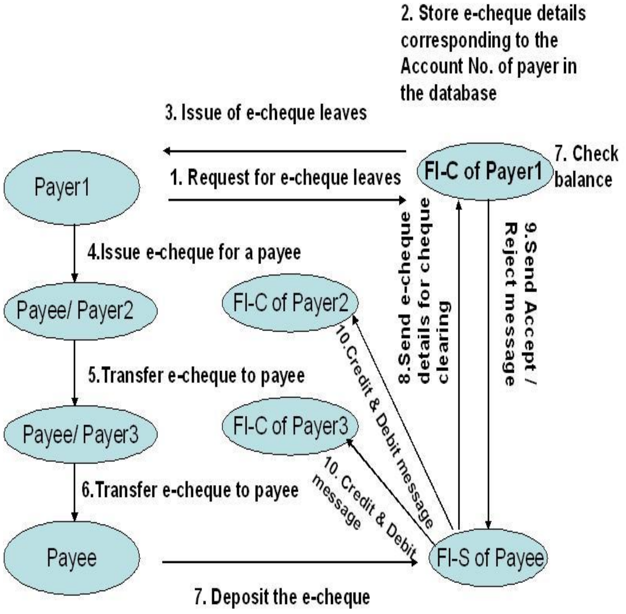
and signs the message m by creating the partial multi-signature, $\langle((\sigma_{2,1}, \sigma_{2,2}, \sigma_{2,3}), m)\rangle$ This partial multi-signature is sent to the third signer U_3.

In the same way the remaining signers verify the partial multi-signature received and generate a new partial multi-signature. The partial multi-signature generated by the last signer is the Forward-secure Serial Multi-signature of n signers which can be verified by any external verifier using a public key which is the product of public keys of all signers.

V Our Model

We assume that the customers issue/transfer e-cheques only if sufficient balance exists in their account. As seen in figure 1, the payer1 requests his bank (FI-C - as this financial institution clears the e-cheque later) for e-cheque leaves. On storing the e-cheque details like e-cheque no, security parameters and so on, the FI-C sends the e-cheque leaves to payer1. The payer1 enters the e-cheque amount, details of the payee and signs it using the forward secure ElGamal-like signature as discussed in section 3 and generates the partial multi-signature as discussed in section 4.2. Payer1 is the initiator. This e-cheque is sent to the payee.

Figure 1: Transfer of e-cheques among multiple customers


2. Store e-cheque details corresponding to the Account No. of payer in the database
3. Issue of e-cheque leaves
Payer1
1. Request for e-cheque leaves
FI-C of Payer1
7. Check balance
4.Issue e-cheque for a payee
8.Send e-cheque details for cheque clearing
9.Send Accept / Reject message
Payee/ Payer2
FI-C of Payer2
10.Credit & Debit message
5.Transfer e-cheque to payee
Payee/ Payer3
FI-C of Payer3
10. Credit & Debit message
6.Transfer e-cheque to payee
Payee
FI-S of Payee
7. Deposit the e-cheque

The payee verifies the multi-signature using equation (22). If it is verified the payee can deposit in his bank (FI-S - as this financial institution submits the e-cheque to FI-C) or can transfer the same e-cheque to another payee. If he is transferring the e-cheque, he becomes payer2. Payer2 must sign as done by payer1 using the forward secure ElGamal-like signature and generate partial multi-signature as discussed in section 4.1. The payee can verify the partial multi-signature and either submit it in his FI-S or transfer it to another payee. This can continue for any number of customers. Once the payee verifies the partial multi-signature, he is convinced that he has received from the intended sender. The e-cheque is given a validity period before which it has to be submitted in a bank. When the last person deposits the e-cheque in his FI-S, the e-cheque is cleared by the FI-C of the first payer. Thus the transfer of e-cheques is done off-line. Also, the e-cheque needs to be cleared only once. The FI-S of the last customer just sends a message to all FI-C of signers of e-cheque to credit and debit the e-cheque amount. This information is required to keep track of all the transactions of a customer.

VI Security Analysis:

In this section we analyze the possible attacks against our forward-secure multi-signature scheme:

A Attacks aiming to get private keys

1. Recover secret key from public key : The public key β for the Forward-Secure Parallel/Serail Multi-signature is computed as the product of public keys of individual signers:

$$\begin{aligned}\beta &= \beta_1 \ldots \beta_n \bmod p\\ &= \alpha^{\mathcal{A}(\alpha,T-i-1,a_{1,i})} \ldots \alpha^{\mathcal{A}(\alpha,T-i-1,a_{n,i})} \bmod p\end{aligned}$$

Recovering $a_{j,i}$ from β_j is equivalent to solving discrete log problem which is computationally not possible.
2. Determining secret key from a set of signatures: There are n equations of the form (18), but $(n+1)$ unknowns (since each signature uses different secret k_j). The system of equations cannot be solved and the private key $a_{j,i}$ is secure.
3.Recovering k_j and then determine $a_{j,i}$: If an adversary is able to get k_j, he can determine $a_{j,i}$. But recovering k_j from y_j is equivalent to solving discrete log problem.
4. When the private keys of one or more users are lost and if the intruder holds this secret information and intend to get private keys of other users, he must break the security as mentioned above(1,2&3).

B Attacks for Forging Multi-signatures

1. The Substitution Attack: This attack is prevented by the use of one-way hash functions (see equation 9).
2. Any signer $U_j (2 \leq j \leq n)$ may want to forge a multi-signature for a message m and then declare that m is signed by $U_1, \ldots, U_{j-1}$ and U_j itself. By this j signer is making all the previous $j-1$ signers responsible for the forged message. This is once again prevented by the use of one-way hash functions. $\langle((\sigma_{j,1}, \sigma_{j,2}, \sigma_{j,3}), m)\rangle$

C Forward security of the proposed scheme

For the sake of completeness we repeat this section [7]. Here we prove that, given a secret key a_i of some time period i an adversary cannot find the secret key for some period $j < i$. We show that in equation (11), finding secret key using public key (as public key is obtained by updating the base secret key T times) is equivalent to solving discrete log problem.

Let P_1 be the discrete log problem where given α and B we want to find A in

$$B \equiv \alpha^A \bmod \phi^{T-j-1}(p) \tag{26}$$

This problem is believed to be computationally hard.

Let P_2 be a problem where given α and a_i we need to find the secret key a_j in

$$a_i \equiv \mathcal{A}(\alpha, i-j, a_j) \bmod \phi^{T-i}(p) \tag{27}$$

We claim that if P_1 is hard, then P_2 is also hard. Thus, if we can find a_j from a_i in (36) we can find A from B in (35). We prove this by contradiction.

Proof: Let us assume that P_2 is not hard. We will show that P_1 is also not hard. Set

$$a_i \equiv \mathcal{A}(\alpha, i-j-1, \beta) \bmod \phi^{T-i}(p) \tag{28}$$

then a_j obtained from solving P_2 satisfies

$$\alpha^{a_j} \equiv \beta \bmod \phi^{T-j-1}(p) \tag{29}$$

By setting $a = a_j$ we have obtained a solution in P_1 which is a contradiction.

VII Conclusion

Many applications require multiple signers to sign the same document. A multi-signature scheme enables a group of signers to produce a compact, joint signature on a common document. We have come up with a scheme to provide an option for a customer to transfer a cheque to another customer without depositing it in the bank. Each customer receiving the e-cheque is convinced that the cheque is from the intended sender. Only the last receiver of the e-cheque deposits it in the bank. All previous customers transfer the e-cheque off-line. This reduces the work load on bank to clear the e-cheques. We use the concept of Serial Forward-Secure multi-signatures. These schemes ensure forward-security of the messages and the signatures can be verified using a single public key though multiple signers are involved.

References

[1] Abdalla,M., Reyzin,L.: A New Forward-Secure Digital Signature Scheme. In: ASIACRYPT 2000, LNCS, Vol.1976, pp. 116-129. Springer-Verlag, (2000).

[2] Anderson,R.: Invited Lecture, Fourth Annual Conference on Computer and Communications Security, ACM, (1997).

[3] Bellare,M., Miner,S.: A Forward-Secure Digital Signature Scheme. In: Wiener, M. (eds.): Advances in Cryptology-Crypto 99 proceedings, LNCS, Vol.1666, Springer-Verlag, (1999).

[4] Damgard, I.: Collision-free hash functions and public key signature schemes. In: EUROCRYPT 87, LNCS, Vol.304, pp. 203216, Springer- Verlag, (1987).

[5] Taher ElGamal: A Public Cryptosystem and a Signature Scheme based on Discrete Logarithms, IEEE transactions on Information Theory, Vol. IT-31, No.4, (1985).

[6] Malkin Tal, Miccianco Daniele, Miner, S.: Efficient Generic Forward Secure Signatures with an unbounded number of time periods. In. Proceedings of EuroCrypt 2002, LNCS, Vol. 2332, pp 400-417, Springer-Verlag, (2002).

[7] B.B.Amberker, Prashant Koulgi, N.R.Sunitha : Forward-Security for ElGamal-like Signature Scheme. In : Proceedings of 6th Annual Security Conference, Las Vegas, April 2007.

[8] Boyd.C : Digital Multi-signatures. In : Cryptography and Coding, Oxford University Press,pp 241-246, 1989.

[9] Micali S., K.Ohta and L.Reyzin., Accountable Subgroup Multi-signatures, In : ACM Conference on Computer and Communications Security, pp 245-254 (2001).

[10] Shiuh-Pyng Shieh, Chern-Tang Lin, Wei-Bon Yang, and Hung-Min Sun, Digital Multi-signature schemes for Authenticating Delegates in Mobile Code Systems, IEEE transactions on Vehicular Tech., Vol 49, No.4, July 2000.