Threshold-Multisignature Schemes where Suspected Forgery Implies Traceability of Adversarial Shareholders

Chuan-Ming Li, Tzonelih Hwang and Narn-Yih Lee

Institute of Information Engineering, National Cheng-Kung University, Tainan, Taiwan, R.O.C.

Abstract. In this paper, we are going to combine the idea of the (t, n) threshold signature scheme with the multisignature scheme and propose a new type of signature scheme, called the (t, n) threshold-multisignature scheme. In the (t, n) threshold-multisignature scheme, at least t shareholders of a specific group have to cooperate to generate a valid group signature and suspected forgery implies traceability of adversarial shareholders. The validity of that signature for the specific group can be verified without knowing the membership of the signers to that group.

1 Introduction

Digital signature is very important in the modern electronic data processing systems. The signer of the conventional digital signature schemes is usually a single user. However, the responsibility of signing messages needs to be shared by a set of signers from time to time. The (t, n) threshold signature schemes [1][2] and the multisignature schemes [5][8][9][10][11] are used to solve these problems.

In the (t, n) threshold signature schemes, t or more shareholders of the group cooperate to generate a valid group signature and the verifier can verify the validity of the group signature without identifying the identities of the signers. However, as being pointed out in [4], if t or more shareholders act in collude (assuming that they do not open their keys in public,) then they can impersonate any other set of shareholders to forge signatures. In this case, the malicious set of signers does not have to take any responsibility for the forged signatures and thus encourages collusion. Consider the setting that assumes there is a log of input messages for signing. If the system recognizes that a certain message was never an input of that log but was signed, then the identities of the signers have to be uncovered. Unfortunately, with the (t, n) threshold signature schemes proposed so far, this problem cannot be solved.

On the other hands, in the multisignature schemes, the signers of a multisignature are identified in the begining and the validity of the multisignature has to be verified together with the identities (or public keys) of the signers. Under this model, it is indeed unnecessary to put a threshold value to restrict the number of signers to generate a valid multisignature. However, consider the other setting where a group of anonymous members would have to generate a

multisignature. The members of this group use pseudonyms as their identities in the public directory. What the verifier concerns most is that a message is signed by at least t members and they indeed come from that group. But the verifier has no way to verify whether a user is in fact a member in that group because of the anonymity of the membership. In this case, the multisignature schemes cannot solve this problem, however, the threshold schemes do.

In reality, there are many applications that simultaneously belong to both settings described above. From the point of view of the signing group, what the group concerns is the *traceability* of the signing set. On the other hand, from the verifier's point of view, whether the signature is indeed from that group and signed by at least t members (not the membership of the members in that group) is concerned most. Both the (t, n) threshold signature schemes and the multisignature schemes cannot independently solve this problem.

In this paper, we are going to combine the idea of (t, n) threshold signature schemes with the multisignature schemes and propose a new type of signature scheme, called the (t, n) threshold-multisignature scheme, to solve this problem. The new schemes, one requires a trusted share distribution center (SDC) to determine the group secret keys and all shareholders' secret shares while the other does not, are based on Harn's (t, n) threshold signature scheme [2] and Yen-Laih's digital signature scheme [7].

The structure of this paper is as follows. In the next section, we propose a (t, n) threshold-multisignature scheme with a trusted SDC and discuss its security. Section 3 proposes the other (t, n) threshold-multisignature scheme without the assistance of a trusted SDC. Finally, we conclude this paper in Section 4.

2 (t,n) Threshold-Multisignature Scheme with A Trusted SDC

We assume that there is a trusted SDC for determining the group secret parameters and all shareholders' secret shares. Let A ($|A| = n$) be the set of all shareholders, B be any subset in A of size t ($|B| = t$). The new scheme is described as follows:

Part 1: Group Secret Key and Secret Shares Generation Phase

First of all, the trusted SDC selects :

- a collision free one-way hash function H [12];
- a prime modulous p, where $2^{511} < p < 2^{512}$;
- a prime q, where q is a divisor of $p - 1$ and $2^{159} < q < 2^{160}$;
- a number α, where $\alpha \equiv h^{(p-1)/q} \bmod p$, h is a random integer with $1 \leq h \leq p - 1$ such that $\alpha > 1$; (α is a generator of order q in GF(p).)
- a polynomial $f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1} \bmod q$, such that each a_i, for $i = 0, \ldots, t - 1$, is a random integer with $0 < a_i < q$.

$\{H, p, q, \alpha\}$ are the group public parameters, and the polynomial $f(x)$ must be kept secret. It is noted here that if α is a generator of order q in GF(p), then we have $\alpha^r \bmod p = \alpha^{r \bmod q} \bmod p$, for any non-negative integer r [6].

The SDC determines the group secret key as $f(0)$ and computes the group public key, y, as

$$y = \alpha^{f(0)} \quad \bmod p.$$

Then, the SDC computes the secret share u_i for each shareholder i, $i \in A$, as

$$u_i = g_i + f(x_i) \quad \bmod q,$$

where x_i is the public value associated with each shareholder i, and g_i is a random integer with $0 < g_i < q$. The SDC also needs to compute public keys, y_i, z_i, as

$$\begin{aligned} y_i &= \alpha^{u_i} \quad \bmod p, \quad \left(= \alpha^{g_i + f(x_i) \bmod q} \quad \bmod p\right) \\ z_i &= \alpha^{g_i} \quad \bmod p, \end{aligned} \tag{1}$$

for each shareholder i, $i \in A$.

Part 2: Partial Signature Generation and Verification Phase

To create the group signature for the message m, each shareholder i, $i \in B$, will randomly select an integer k_i, $k_i \in [1, q-1]$, and compute a public value, r_i, as

$$r_i = \alpha^{k_i} \quad \bmod p.$$

Then each shareholder i, $i \in B$, makes r_i publicly available through a broadcast channel. Once all r_i, $i \in B$, are available, each shareholder i in B computes the product R and a hash value E as

$$\begin{aligned} R &= \prod_{i \in B} r_i \quad \bmod p, \left(= \alpha^{\sum_{i \in B} k_i \bmod q} \quad \bmod p\right) \\ E &= H(m, R) \quad \bmod q. \end{aligned} \tag{2}$$

Then, shareholder i uses his secret keys, u_i and k_i, to calculate the partial signature s_i as

$$s_i = u_i \cdot \left(\prod_{\substack{j \in B \\ j \neq i}} \frac{0 - x_j}{x_i - x_j} \right) + k_i \cdot E \quad \bmod q. \tag{3}$$

Each shareholder i in B sends the values, $\{m, s_i\}$, to a designated combiner DC who takes the responsibility to collect and verify each partial signature and then produces a group signature. We should note that there is no secret information associated with the DC.

DC can verify the partial signature $\{m, r_i, s_i\}$ by the following:

$$\alpha^{s_i} \stackrel{?}{\equiv} y_i^{\left(\prod_{\substack{j\in B \\ j\neq i}} \frac{0-x_j}{x_i-x_j}\right) \bmod q} \cdot r_i^E \mod p.$$

If the above equation holds, then the partial signature $\{m, r_i, s_i\}$ for shareholder i is valid.

Part 3: Group Signature Generation and Verification Phase

Once these t partial signatures are verified, the DC can compute an S by

$$S = \sum_{i\in B} s_i \mod q.$$

$\{B, R, S\}$ is the group signature of m.

To verify the validity of the group signature $\{m, B, R, S, \}$, the verifier needs to compute a verification value T and the hash value E as

$$T = \prod_{i\in B} z_i^{\left(\prod_{\substack{j\in B \\ j\neq i}} \frac{0-x_j}{x_i-x_j}\right)} \mod p, \tag{4}$$
$$E = H(m, R) \mod q.$$

Then, the verifier uses the group public key y to check

$$\alpha^S \stackrel{?}{\equiv} y \cdot T \cdot R^E \mod p. \tag{5}$$

If the Eq. (5) holds, the group signature $\{m, B, R, S\}$ is valid.

It should be obvious that the group size, $(|A| = n)$, of the new scheme can be increased by assigning the new public keys, x_j, y_j and z_j, and the secret share, u_j, to a new shareholder j without affecting the secret shares of the others.

Theorem 1 : If $\alpha^S \equiv y \cdot T \cdot R^E \mod p$, then $\{B, R, S, \}$ is the group signature of m.

$<$ *proof* $>$ With the knowledge of the hash value E and the secret keys, u_i and k_i, shareholder i can generate his partial signature $\{m, r_i, s_i\}$ for message m to satisfy

$$\alpha^{s_i} \equiv \alpha^{u_i \cdot \left(\prod_{\substack{j\in B \\ j\neq i}} \frac{0-x_j}{x_i-x_j}\right) + k_i \cdot E \bmod q} \mod p$$
$$\equiv y_i^{\left(\prod_{\substack{j\in B \\ j\neq i}} \frac{0-x_j}{x_i-x_j}\right) \bmod q} \cdot r_i^E \mod p. \tag{6}$$

By mulitiplying α^{s_i} for all $i \in B$, we have

$$\prod_{i\in B} \alpha^{s_i} \equiv \prod_{i\in B} y_i^{\left(\prod_{\substack{j\in B \\ j\neq i}} \frac{0-x_j}{x_i-x_j}\right) \bmod q} \cdot r_i^E \mod p. \tag{7}$$

According to Eq. (1), Eq. (2) and Eq. (4), we can rewrite the right-hand side of Eq. (7) as

$$\prod_{i\in B} y_i^{\left(\prod_{j\in B, j\neq i} \frac{0-x_j}{x_i-x_j}\right) \bmod q} \cdot r_i^E \mod p$$

$$\equiv \alpha^{\sum_{i\in B}\left(u_i \cdot \prod_{j\in B, j\neq i} \frac{0-x_j}{x_i-x_j}\right) \bmod q} \cdot \prod_{i\in B} r_i^E \mod p$$

$$\equiv \alpha^{\sum_{i\in B}\left(g_i \cdot \prod_{j\in B, j\neq i} \frac{0-x_j}{x_i-x_j} + f(x_i)\cdot \prod_{j\in B, j\neq i} \frac{0-x_j}{x_i-x_j}\right) \bmod q} \cdot R^E \mod p$$

$$\equiv \alpha^{\sum_{i\in B}\left(g_i \cdot \prod_{j\in B, j\neq i} \frac{0-x_j}{x_i-x_j}\right) \bmod q} \cdot \alpha^{\sum_{i\in B}\left(f(x_i)\cdot \prod_{j\in B, j\neq i} \frac{0-x_j}{x_i-x_j}\right) \bmod q} \cdot R^E \mod p$$

$$\equiv T \cdot \alpha^{\sum_{i\in B}\left(f(x_i)\cdot \prod_{j\in B, j\neq i} \frac{0-x_j}{x_i-x_j}\right) \bmod q} \cdot R^E \mod p. \quad (8)$$

With the knowledge of t pairs of $(x_i, f(x_i))$, the unique $(t-1)$-th degree polynomial, $f(x)$, can be determined as [3]

$$f(x) = \sum_{i\in B}\left(f(x_i)\cdot \prod_{j\in B, j\neq i} \frac{x-x_j}{x_i-x_j}\right) \quad mod\ q. \quad (9)$$

Thus, the Eq. (8) can be further derived as

$$T \cdot \alpha^{\sum_{i\in B}\left(f(x_i)\cdot \prod_{j\in B, j\neq i} \frac{0-x_j}{x_i-x_j}\right) \bmod q} \cdot R^E \mod p$$
$$\equiv T \cdot \alpha^{f(0)} \cdot R^E \mod p$$
$$\equiv y \cdot T \cdot R^E \mod p.$$

Since S can be expressed as

$$S = \sum_{i\in B} s_i \quad mod\ q,$$

we can rewrite the left-hand side of Eq. (7) as

$$\prod_{i\in B} \alpha^{s_i} \equiv \alpha^{\sum_{i\in B} s_i \bmod q} \mod p$$
$$\equiv \alpha^S \mod p. \quad Q.E.D.$$

Security Discussion:

According to the Theorem 1, any subset B of t shareholders can generate a valid group signature for the set A, however, less than t shareholders cannot. The group signature can also be verified easily by any verifier. Here, we will discuss several possible attacks. None of these attacks can successfully break our new scheme.

1. Can one retrieve the group secret key $f(0)$ and the secret share u_i, $i \in A$, from the group public key y and y_i, $i \in A$?

   Obviously, this difficulty is as same as solving the discrete logarithm problem.
2. Can one retrieve the secret share u_i from the Eq. (3) ?

   For a given message and the corresponding signature pair, Eq. (3) involves two unknown parameters, u_i and k_i. For a signature pair of another message m', the unknown parameter is also increased by one. Thus, the number of unknown parameters is always larger than the number of available equations. This attack cannot work successfully.
3. Can one retrieve the group secret key $f(0)$ from the following?

$$S \equiv \sum_{i \in B} \left(u_i \cdot \prod_{\substack{j \in B \\ j \neq i}} \frac{0 - x_j}{x_i - x_j} + k_i \cdot E \right) \mod q$$

$$\equiv f(0) + \sum_{i \in B} \left(g_i \cdot \prod_{\substack{j \in B \\ j \neq i}} \frac{0 - x_j}{x_i - x_j} \right) + \sum_{i \in B} k_i \cdot E \mod q \qquad (10)$$

   For a given message and the corresponding group signature pair, the Eq. (10) involves three unknown parameters, $f(0)$, $\sum_{i \in B} \left(g_i \cdot \prod_{\substack{j \in B \\ j \neq i}} \frac{0 - x_j}{x_i - x_j} \right)$, $\sum_{i \in B} k_i$, for any shareholder i. It is noted that the unknown parameters are increased along with the increment of the corresponding group signatures. On the other hand, if t or more shareholders act in collude, there are still two unknown parameters, $f(0)$, $\sum_{i \in B} \left(g_i \cdot \prod_{\substack{j \in B \\ j \neq i}} \frac{0 - x_j}{x_i - x_j} \right)$, involved in Eq. (10). For any increment of subset B, the unknown parameter is still increased by one. Therefore, the number of unknown parameters is always larger than the number of available equations. This attack cannot work successfully.
4. Can one impersonate a shareholder i, $i \in B$?

   A forger may try to impersonate a shareholder i, $i \in B$, by randomly selecting an integer $k_i' \in [1, q-1]$ and broadcasting $r_i' \equiv \alpha^{k_i'} \bmod p$. Since the productive value, $R' \equiv \left(\prod_{\substack{j \in B \\ j \neq i}} r_j \right) r_i' \bmod p$, is determined by all t members and the hash value, E', is obtained by $H(m, R')$, without knowing the secret share, u_i, it is difficult to generate a valid value s' to satisfy the verification equation as

$$\alpha^{s_i'} \equiv y_i^{\left(\prod_{\substack{j \in B \\ j \neq i}} \frac{0 - x_j}{x_i - x_j} \right) \bmod q} \cdot r_i'^{E'} \mod p.$$

5. Can one forge a signature $\{m, B, R, S\}$ based on the Eq. (5) ?

   A forger may randomly select an integer R, and then computes the hash value E such that $E = H(m, R) \bmod q$. Obviously, to compute the integer S is equivalent to solving the discrete logarithm problem. On the other hand, the forger can randomly select E and S first, and then try to dertermine

a value R', that satisfy the Eq. (5) and the equation $E = H(m, R')$ simultaneously. However, according to the propertyof the H function, it is quite impossible. Thus, this attack will not be successful.

6. Can t or more shareholders act in collude to reconstruct the secret polynomial $f(x)$?

   According to the Eq. (9), the secret polynomial $f(x)$ can be reconstructed with the knowledge of any t secret values $f(x_i)$, $i \in A$. Anyone who knows the polynomial $f(x)$ can impersonate any shareholder to sign messages without holding the responsibility to the signatures. Nevertheless, the secret share $u_i, i \in A$, in our new threshold signature scheme contains the integer g_i which is known only by the trusted SDC and has to be removed first before reconstructing the polynomial $f(x)$. A malicious shareholder i may try to retrieve the integer g_i from the public key z_i. However, the difficulty is as same as solving the discrete logarithm problem. Thus, any t or more shareholders cannot conspire to reconstruct the polynomial $f(x)$ by providing their own secret shares.

3 (t,n) Threshold-Multisignature Scheme without A Trusted SDC

In this section, we will propose another (t, n) threshold-multisignature scheme which does not need the assistance of a trusted SDC. Since there is no trusted SDC, each shareholder has to act as a SDC to generate his secret key and distribute the corresponding secret shares to other shareholders. The public parameters, $\{H, p, q, \alpha\}$, should be agreed by all shareholders in advance.

Part 1: Group Public Key and Secret Shares Generation Phase

Each shareholder i, $i \in A$, randomly selects a $(t-1)$-th degree polynomial, $f_i(x)$, and an integer x_i, where $x_i \in [1, q-1]$. Then he computes a corresponding public key, y_i, as

$$y_i = \alpha^{f_i(0) \mod q} \mod p.$$

$\{x_i, y_i\}$ are the public keys of the shareholder i, $i \in A$, and the polynomial $f_i(x)$ is his secret parameter. The group public key y can be determined by all shareholders as

$$y = \prod_{i \in A} y_i \mod p. \left(= \alpha^{\sum_{i \in A} f_i(0) \mod q} \mod p\right)$$

Since there is no trusted SDC, each shareholder i must act as a SDC to generate and distribute following values to the shareholder j, $j \in A, j \neq i$, as :

$$u_{ij} = g_{ij} + f_i(x_j) \mod q,$$

$$y_{ij} = \alpha^{u_{ij}} \mod p, \left(= \alpha^{g_{ij} + f_i(x_j) \mod q} \mod p,\right)$$

$$z_{ij} = \alpha^{g_{ij}} \mod p$$

where x_j is the public key of shareholder j, and g_{ij} is a random integer with $0 < g_{ij} < q$. The value of u_{ij} is the secret share generated by shareholder i for shareholder j, and both y_{ij} and z_{ij} are shareholder j's public values.

Part 2: Partial Signature Generation and Verification Phase

Similar to the (t, n) threshold-multisignature scheme proposed in the previous section, each shareholder i, $i \in B$, randomly selects an integer k_i, $k_i \in [1, q-1]$, and compute a public value, r_i, as

$$r_i = \alpha^{k_i} \mod p.$$

Then each shareholder i, $i \in B$, makes r_i publicly available through a broadcast channel. Once all r_i, $i \in B$, are available, each shareholder i in B computes the product R and a hash value E as

$$R = \prod_{i \in B} r_i \mod p, \left(= \alpha^{\sum_{i \in B} k_i \bmod q} \mod p\right)$$
$$E = H(m, R) \mod q.$$

Then, shareholder i uses his secret keys, $f_i(0)$, k_i and $u_{ji}, j \in A, j \notin B$, to calculate the partial signature s_i as

$$s_i = f_i(0) + \sum_{\substack{j \in A \\ j \notin B}} \left(u_{ji} \cdot \prod_{\substack{e \in B \\ e \neq i}} \frac{0 - x_e}{x_i - x_e} \right) + k_i \cdot E \mod q.$$

Each shareholder i in B sends the values $\{m, s_i\}$ to the designated combiner, DC. As similar to the description in the previous section, the DC firstly computes the values of R and E from the broadcast channel, and then he uses shareholder i's public keys x_i, y_i and y_{ji}, for $j \in A, j \notin B$ to verify the validity of the partial signature as

$$\alpha^{s_i} \stackrel{?}{\equiv} \left(y_i \cdot \left(\prod_{\substack{j \in A \\ j \notin B}} y_{ji} \right)^{\prod_{\substack{e \in B \\ e \neq i}} \frac{0 - x_e}{x_i - x_e} \bmod q} \right) \cdot r_i^E \mod p.$$

If the above equation holds, then the partial signature $\{m, r_i, s_i\}$ is valid.

Part 3: Group Signature Generation and Verification Phase

Once all these t partial signatures are verified by the DC, the DC can generate the group signature for the message m as $\{m, B, R, S\}$, where

$$S = \sum_{i \in B} s_i \qquad \bmod q.$$

To verify the validity of the group signature $\{m, B, R, S\}$, the verifier has to compute the veirfication value T and the hash value E as

$$T = \prod_{i \in B} \left(\left(\prod_{\substack{j \in A \\ j \notin B}} z_{ji} \right)^{\prod_{\substack{e \in B \\ e \neq i}} \frac{0 - x_e}{x_i - x_e}} \right) \qquad \bmod p,$$

$$E = H(m, R) \qquad \bmod q.$$

Then, the verifier uses the group public key y to check

$$\alpha^S \stackrel{?}{\equiv} y \cdot T \cdot R^E \qquad \bmod p.$$

If the above equation holds, the group signature $\{m, B, R, S\}$ is valid.

Theorem 2 : If $\alpha^S \equiv y \cdot T \cdot R^E \mod p$, then the group signature $\{B, R, S\}$ is valid.

$<$ *proof* $>$ This proof is similar to the proof of Theorem 1 and is omitted here.

Because the shareholders in this section do not truste each others, how can shareholder i makes sure that the secret share, u_{ji}, given by shareholder j is correctly derived from the secret polynomial $f_j(x)$. This problem may occur in some situations. For example, a dishonest shareholder tries to cheat the shareholders by giving them fake secret shares. The communication errors (i.e., channel noise) can also result in erroneous secret shares. Here, we would like to show that our new scheme can prevent this kind of errors.

Theorem 3 : The erroneous secret share can be detected by any shareholder.

$<$ *proof* $>$ Let us first examine if the fake secret shares is caused by the communication noise. We consider that shareholder i receives a fake secret share, u'_{ji}, from shareholder j, and the corresponding public key is $y_{ji} = \alpha^{u_{ji}} \bmod p$. Obviously, this fake secret share can be easily detected by shareholder i as

$$y_{ji} \stackrel{?}{\equiv} \alpha^{u'_{ji}} \qquad \bmod p.$$

On the other hand, if a dishonest shareholder j picks up a fake secret share, $u'_{ji} = f'_j(x_i) + g_{ji}$, and publishes the corresponding public keys as

$$y_{ji} = \alpha^{u'_{ji}} \qquad \bmod p,$$
$$z_{ji} = \alpha^{g_{ji}} \qquad \bmod p.$$

According to the Eq. (9), a unique $(t-1)$-th degree polynomial, $f_j(x)$, can be reconstructed with the knowledge of t pairs of $(x_i, f_j(x_i))$. Thus, each shareholder

i can verify his public keys y_{ji} and z_{ji}, which are distributed by shareholder j, by combining with any other t-1 shareholders' public keys to compute

$$\prod_{i\in B} y_{ji}^{\left(\prod_{e\in B, e\neq i} \frac{0-x_e}{x_i-x_e}\right) \bmod q} \bmod p \stackrel{?}{\equiv} y_j \cdot \prod_{i\in B} z_{ji}^{\left(\prod_{e\in B, e\neq i} \frac{0-x_e}{x_i-x_e}\right) \bmod q} \bmod p. \tag{13}$$

If the Eq. (13) holds false, it must exist some fake public keys in the subset B. Then, shareholder i knows that shareholder j might be dishonest.

Q.E.D.

The security analysis of this new scheme is almost the same as the previous one proposed in Section 2 (thus is omitted here.) However, this new scheme does not need the assistance of a trusted SDC.

5 Conclusions

In this paper, we have proposed two (t, n) threshold-multisignature schemes, one requires the assistance of a trusted SDC to determine the group secret keys and all shareholders' secret shares and the other does not. In the new schemes, the designated combiner DC, in fact, can be eliminated and instead the verifier of the group signature takes the responsibility to collect and verify each partial signature and the group signature. The features of our new schemes can be summarized as follows:

1. The group signature can only be generated by t or more shareholders.
2. The partial signatures generated by the shareholders can be verified by a designated combiner, DC, (or by the verifier) before they be combined into a group signature.
3. The validity of the group signature can be verified without knowing the membership of the signers to the group.
4. Since the system secrets cannot be retrieved by malicious shareholders, the signing set of a group signature cannot be impersonted by any other set of shareholders and the suspected forgery can be traced and the faulty shareholders can be caught in our new schemes.
5. The group size, $|A|$, can be dynamically increased without affecting the secret shares of the others.

Acknowledgement The authors wish to thank the referees of this paper for their useful comments that make this paper more readable.

References

1. Y. Desmedt and Y. Frankel: "Shared Generation of Authenticators and Signatures", Advances in Cryptology - Crypto '91, Proceedings, pp.457-469, Springer Verlag, 1991.
2. L. Harn: "(t,n) Threshold Signature and Digital Multisignature", Workshop on Cryptography & Data Security, Proceedings, pp.61-73, June 7-9, Chung Cheng Institute of Technology, R.O.C., 1993.
3. A. Shamir: "How to Share A Secret", Commun. ACM, 22:612-613, 1979.
4. C. Li, T. Hwang and N. Lee: "Remark on the Threshold RSA Signature Scheme", Advances in Cryptology - Crypto '93, Proceedings, 1993.
5. K. Itakura and K. Nakamura: "A Public-Key Cryptosystem Suitable for Digital Multisignatures", NEC Research and Develop., 71, pp.1-8, 1983.
6. "The Digital Signature Standard", Commun. ACM, 35, No.7, pp.36-40, 1992.
7. S. M. Yen and C. S. Laih: "New Digital Signature Scheme Based on Discrete Logarithm", Electrionics Letters, Vol. 29, (12), pp.1120-1121, 1993.
8. T. Hardjono and Y. Zheng: "A Practical Digital Multisignature Scheme Based on Discrete Logarithms", Advances in Cryptology - AUSCRYPTO'92, pp. 16-21, 1992.
9. L. Harn and T. Kiesler: "New Scheme for Digital Multisignatures", Electronic Letters, Vol. 25, No. 15, pp. 1002-1003, 1989.
10. K. Ohta and T. Okamoto: "A Digital Multisignature Scheme Based on the Fiat-Shamir Scheme", Advances in Cryptoloy - ASIACRYPT'91, pp. 75-79, 1991.
11. T. Okamoto: "A Digital Multisignature Scheme Using Bijective Public-Key Cryptosystem", ACM Transactions on Computer Systems, Vol. 6, No. 8, pp. 432-441, 1988.
12. Y. Zheng, T. Matsumoto and H. Imai: "Structural Properties of One - Way Hash Functions", Advances in Cryptology - Crypto '90, Proceedings, pp. 285-302, Springer Verlag, 1990.