International Journal of Innovative Computing, Information and Control
Volume **8**, Number **4**, April **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **2713–2727**

A STUDY ON EFFICIENT GROUP-ORIENTED SIGNATURE SCHEMES FOR REALISTIC APPLICATION ENVIRONMENT

Yu-Fang Chung[1], Tzer-Long Chen[2], Tzer-Shyong Chen[3] and Chih-Sheng Chen[4]

[1]Department of Electrical Engineering
[3]Department of Information Management
[4]Department of Statistics
Tunghai University
No. 181, Sec. 3, Taichung Harbor Road, Taichung 40704, Taiwan
{ yfchung; arden; sschen }@thu.edu.tw

[2]Department of Information Management
National Taiwan University
No. 1, Sec. 4, Roosevelt Road, Taipei 10617, Taiwan
d97725005@ntu.edu.tw

Received October 2010; revised February 2011

Abstract. *Rapid development of data processing systems has made digital signatures an essential application. A digital signature basically associates a signer with the message. Its important characteristics are easy verification, unforgeability and undeniability. However, conventional digital signature schemes generally consider only single signer situations; this is impractical, because the authorized signatory in the business world is generally composed of signatures of several people. Therefore, to enable co-signatories on a document, several group signature schemes are hereby proposed in this paper, including threshold group signature, anonymous ring signature, and group signature that incorporates ring signature technology. Since the aforementioned signature schemes are all based on Elliptic Curve Cryptosystem (ECC), they have short key size, low computation load, and little bandwidth requirement. Therefore, all the above schemes are considerably efficient. Finally, analyses are carried out to prove that the proposed schemes can withstand signature forgery attack and are signer undeniable, and thus meet the security requirements.*

Keywords: Digital signature, Group signature, Threshold group signature, Ring signature, Signcryption, Elliptic curve cryptosystem

1. **Introduction.** Advances in cryptography provide better information security on the Internet. Encryption systems ensure confidentiality of message transmission, while digital-signature technology ensures authenticity and integrity of information. These factors play a significant role in information security. Conventional handwritten signatures are increasingly replaced by digital signatures, which are widely used in the Internet society. The digital signature method was proposed by W. Diffie and M. Hellman in 1976 [1], since then various other digital signature methods have been developed, including RSA, ElGamal and DSS, which form the basis of the methods presented in this study. However, in practice, people have different requirements with regard to digital signatures. Various digital signatures [36,37,39], such as group signatures, ring signatures [2], blind signatures, proxy signatures [3], threshold signatures [4] and signcryption [5], have been designed to meet different needs.

Most of the past research focused on RSA-based or ElGamal-base group signatures. Since these techniques showed the problems of over-computation and application restriction as well as not being able to control dynamic in a threshold scheme members, dynamic member access security could not be easily designed. The proposed group signature, based on ECC, presents the same efficacy and features as previous group signatures. Besides, it integrates the features of ring signatures that it has the advantages on the verification of identity in signature. The integration therefore makes the technique to meet the security requirements of availability, non-repudiation, anonymity, and confidentiality as a whole. Further more, with the short session keys and low computation in the ECC cryptosystem, the proposed technique displays better advantage on performance.

Owing to the fact that electronic signature applications are mostly used for business purposes, legal signatory is generally composed of several signatures. To enable co-signatory, group signature technology is increasingly gaining importance. This paper mainly focuses on the research and development of various types of group signatures, including group threshold signature, ring signature, and group anonymous signcryption based on ring signature.

1.1. **Group-oriented threshold signature.** The concept of group-oriented cryptography, introduced by Desmedt [20] in 1987, is related to secure communication between different groups. Since then, group-oriented schemes have evolved into threshold signature schemes. In relation to the application of the perfect secret sharing scheme [21] developed by Shamir, Harn [22] constructed a (t, n) threshold signature scheme based on Lagrange polynomials. The so-called (t, n) threshold signature scheme is that any t members of a n-member group can represent the whole group and validly sign on behalf of the group, where t represents the threshold and has a value that lies between 1 and n ($1 \leq t \leq n$). Much research has been conducted in this area [23-29] in recent years.

Some schemes have no restrictions with regard to the verifiers. Wang et al. presented a (t, n) threshold signature scheme with (k, l) threshold shared verification [30], requiring a specified verifier. In other words, only the specified verifier can verify the group signature. In his scheme, k members of a specific l-verifier group can verify the group signature, where k is a threshold that lies between 1 and l ($1 \leq k \leq l$). In 2002, Hsu et al. showed that the scheme violated the requirements for (k, l) threshold shared verification [31]. Namely, an attacker could validate the group signature without help from other members of the verifier group. Hsu also stated that the private key of the signer could be easily retrieved from the individual signature. Hsu proposed an improvement to overcome these two security problems. His improvement specifies that a random number selected by a system center (SC) to be included in the solution to solve the aforementioned problems. This additional operation is performed every time an individual signature is generated. Consequently, efficiency is affected. Therefore, a scheme based on ECC [32-35] that provides both security and efficiency is presented in Section 3.1. The additional operation, which involves the SC in the generation of each individual signature, is avoided.

1.2. **Ring signature.** The ring signature scheme [2], developed by R. Rivest et al. in 2001, was created from the concept of "how to leak a secret". Ring signature is an unusual group signature that does not require creating a group. Although the scheme uses an administrator, a signer can create a ring signature through his private key in combination with a randomly chosen portion of the public keys of members. This signature method significantly lowered the complexity of the mutual authentication process by allowing the signer to remain anonymous, and thus protects the privacy of a signer.

Some researchers, who utilized elliptic curve cryptosystem and bilinear pairing cryptosystem like Weil pairing [9] and Tate pairing [10], chose to employ applied hyper-elliptic

curve on ring signature, for instance, the identity-based ring signature scheme [11,38] presented by F. Zhang et al. in 2002, its improvement method [12] presented by C. Y. Lin et al. in the following year, and the identity-based threshold ring signature [13] presented by S. Chow et al.

We briefly mention a number of other signature schemes as follows. J. Yu et al. [14] proposed a ring signature scheme which is secure against the chosen message attack without using the random oracle model. An efficient identity-based ring signature by S. Chow et al. [15] is applicable to different group sizes. J. K. Liu et al. [16] introduced a separable threshold signature scheme whose greatest contribution is that it is applicable in both RSA-based as well as DLP-based public key cryptosystems. In general, the scheme supported the combination of public keys for all trapdoor-one-way type as well as three-move type signature schemes.

1.3. **Group anonymous signcryption.** Signcryption is a kind of public key cryptosystem that can both digitally and simultaneously encrypt and sign a message. Compared with traditional systems like the PGP that signs and encrypts a message in sequential procedures, its ability to simultaneously sign and encrypt makes the signcryption system more secure and more efficient. To be specific, the signcryption system is 50% to 90% more efficient than the traditional ones.

The concept of Signcryption was introduced by Zheng [5] in 1997. Since then, many researchers have addressed and discussed many variations of signcryption schemes [6-8]. Lee and Mao presented a signcryption scheme based on RSA [6] and proposed security proofs in the random oracle model aiming at privacy and unforgeability. Libert and Quisquater presented an ID-based signcryption using bilinear pairing [7]. Additionally, Yum and Lee proposed the new signcryption schemes based on KCDSA [8].

Nevertheless, the above-mentioned schemes were unable to meet the requirement of anonymity for signers. Anonymous signcryption is useful in cases where the identity of a sender must remain secret, yet the message verifiable. Thus, in Section 3.2, an anonymous signcryption scheme based on the elliptic curve cryptosystem is presented; the scheme encompasses all the advantages of a ring signature scheme. The application of elliptic curve cryptosystem improves the performance by increasing the efficiency. As for security, the proposed scheme not only ensures the confidentiality of the signer but also possesses characteristics like unforgeability, anonymity, undeniability, and forward secrecy.

2. Review of Previous Research.

2.1. **Elliptic curve cryptosystem.** Elliptic Curve Cryptosystem is known to provide security equal in level to RSA or DSA in the discrete logarithm problem (DLP); it also has lower computation overhead and smaller key size. Owing to ECC, the proposed scheme attains high security and efficiency. The mathematic background of ECC [17,19] is explained below.

As the name implies, ECC uses elliptic curves. With the variables and coefficients of elliptic curves restricted to elements of a finite field, added efficiency is achievable in the operation of ECC. Two families of elliptic curves, *prime curves* defined over Z_p and *binary curves* constructed over $GF(2^n)$, are used in cryptographic applications. Quoting Fernandes [18], "prime curves are best for software applications because the extended bit-fiddling operations needed by binary curves are not required; and that binary curves are best for hardware application, where it takes remarkably few logic gates to create a powerful, fast cryptosystem".

In this study, the applied elliptic curve over Z_p, defined modulo a prime p, is the set of solutions (x, y) to the equation $E_p(a, b)$: $y^2 = x^3 + ax + b(\text{mod}p)$, where a and $b \in Z_p$ and $4a^3 + 27b^2 \mod p \neq 0$. The condition $4a^3 + 27b^2 \mod p \neq 0$ is necessary to ensure that $x^3 + ax + b(\text{mod}p)$ has no repeated factors; meaning, a finite abelian group can be defined based on the set $E_p(a, b)$. The definition of an elliptic curve also includes a point at infinity denoted as O. This point is the third point of intersection of any straight line with the curve; such a line has points of intersection of the form (x, y), $(x, -y)$ and O. Not every elliptic curve over Z_p can be applied in cryptographic applications. Figure 1 (taken from [19]) shows an example of the elliptic curve which is defined by the equation, $y^2 = x^3 + x + 1$ (mod 23).

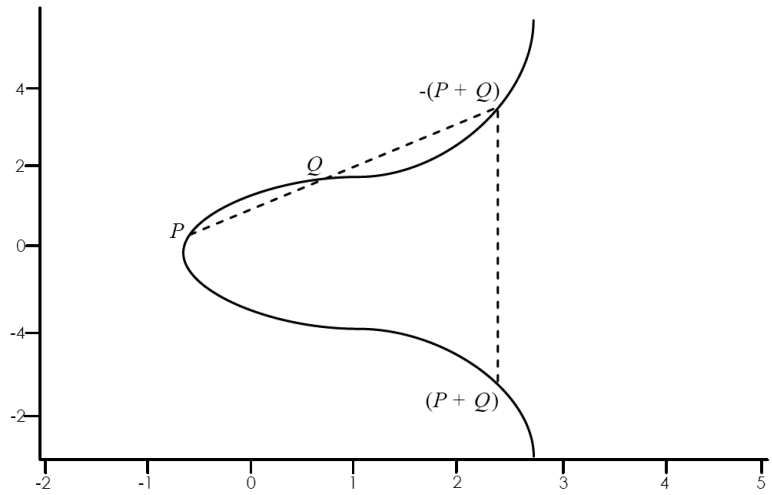

FIGURE 1. Example of elliptic curve in case of $y^2 = x^3 + x + 1$ (mod 23)

The example depicted in Figure 1 has $a = 1$ and $b = 1$, so that $4a^3 + 27b^2 \mod 23 \equiv 8 \mod 23 \neq 0$. Thus, the elliptic group $E_{23}(1, 1)$ consists of the points shown in Table 1, extracted from [19].

TABLE 1. Points over the elliptic curve $E_{23}(1, 1)$

(0, 1)	(6, 4)	(12, 19)	(0, 22)	(6, 19)	(13, 7)	(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)	(3, 10)	(9, 7)	(17, 20)	(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)	(5, 4)	(11, 20)	(19, 5)	(5, 19)	(12, 4)	(19, 18)

Addition operation has been used over $E_p(a, b)$. For all points P and $Q \in E_p(a, b)$, the rules for addition over $E_p(a, b)$ are defined as follows.

(1) $P + O = P$, where O serves as the additive identity.
(2) If $P = (x_p, y_p)$, then $P + (x_p, -y_p) = O$. The point $(x_p, -y_p)$ is the negative of P, denoted as $-P$. For example, in $E_{23}(1, 1)$, for $P = (6, 4)$, $-P = (6, –4)$ is received. Since –4 mod 23 ≡ 19, $–P = (6, 19)$, which is also in $E_{23}(1, 1)$.
(3) If $P = (x_p, y_p)$ and $Q = (x_q, y_q)$ with $P \neq -Q$, then $R = P + Q = (x_r, y_r)$ is in $E_{23}(1, 1)$ and is determined by the following rules.

$$\begin{cases} x_r = (\lambda^2 - x_p - x_q) \bmod p \\ y_r = (\lambda^2(x_p - x_r) - y_p) \bmod p \end{cases}, \text{ where } \lambda = \begin{cases} \left(\dfrac{y_q - y_p}{x_q - x_p}\right) \bmod p, & \text{if } P \neq Q \\ \left(\dfrac{3x_p^2 + a}{2y_p}\right) \bmod p, & \text{if } P = Q \end{cases}$$

(4) Multiplication by an integer is defined by repeated addition; for example, $2P = P+P$.

Addition in ECC is the counterpart of modular multiplication in RSA, and multiplication in ECC is the counterpart of modular exponentiation in RSA. A difficult computational problem is the key to a secure cryptographic system using elliptic curves over Z_p. Consider the equation $Q = kP$, where Q and $P \in E_p(a, b)$ and $k < p$. Although it is relatively easy to calculate Q given k and P, it is extremely hard to determine k given Q and P. This is called the elliptic curve discrete logarithm problem (ECDLP).

Given an example taken from [19], suppose $E_{23}(9, 17)$ is the elliptic curve defined by $y^2 = x^3 + 9x + 17 \pmod{23}$. Find the discrete logarithm k of $Q = (4, 5)$ to the base $P = (16, 5)$. One solution is the brute-force method, in which multiples of P is computed until Q is found. The elliptic group $E_{23}(9, 17)$ consists of the points shown in Table 2.

TABLE 2. Points over the elliptic curve $E_{23}(9, 17)$

$P = (16, 5)$	$2P = (20, 20)$	$3P = (14, 14)$
$4P = (19, 20)$	$5P = (13, 10)$	$6P = (7, 3)$
$7P = (8, 7)$	$8P = (12, 17)$	$9P = (4, 5)$

Note that $9P = (4, 5) = Q$, that is, the discrete logarithm k of Q to the base P is 9. However, in practice, the brute force method is quite infeasible as p and k are so large that the method would not be practical.

As it seems, the efficiency of an ECC depends on how fast $Q = kP$ can be calculated for some numbers k and a point P on the curve. The addition of elliptic curve points only requires few modular calculations. As shown in [19], the prime p in ECC can be of a much smaller value than the corresponding numbers in the other types of systems, achieving an advantage with efficiency over integer factorization and discrete logarithm systems.

2.2. **Ring signature scheme.** The original ring signature [2] makes signers sign documents anonymously to protect the identity of the signer. The concept of ring signature scheme is similar to that of Fuzzy Theory. In a ring signature scheme, a signer can dynamically choose members and the number of members according to the situation, and then uses the public key of other members and the secret key of the signer to generate a ring signature for a particular message. Such a system does not require a manager to handle affairs. A verifier can only determine the group that the signer belongs to but not the identity of the signer. The above-mentioned property is also the major difference between ring signature and group signature.

2.2.1. *Ring signature algorithm and ring signature verification algorithm.* Suppose that the said scheme is based on Public Key Infrastructure (PKI). All users hold a public key P_k, and the corresponding secret key S_k is registered with PKI certificate authority.

For the operation of ring signature algorithm, suppose the number of members of the signer group (also called a ring) is r. The signer uses the chosen members' public keys $P_1, \ldots, P_r$ and the individual secret key S_s to sign the message m by generating a ring signature σ for the said message.

For the operation of ring verification algorithm, the verifier inputs (m, σ), and then judges whether the signature is real or fraud according to the output, True or False.

2.2.2. *Definition of combination function.* Combination function, which is used not only to improve performance but also to enhance security, must possess properties like one-wayness, single input inversion, and multiple inputs in solvability. Firstly, an invertible

combination function z, such as Equation (1) [2] below, is defined. It can be verified that z is a combination function for any index value s using Equation (2).

$$z = C_{k,v}(y_1, \cdots, y_n) = E_k(y_n \oplus E_k(y_{n-1} \oplus E_k(\cdots \oplus E_k(y_1 \oplus v) \cdots))) \tag{1}$$

$$y_s = E_k^{-1}(y_{s+1} \oplus \cdots E_k^{-1}(y_n \oplus E_k^{-1}(z)) \oplus (\cdots \oplus E_k(y_1 \oplus v) \cdots)) \tag{2}$$

The combination function can be employed to output a unique value y_s as the public key of the signer s, where the inputs involve an initialization value v, a symmetric key k, and a series of variables of length l.

2.2.3. *Ring signature generation stage.* Suppose signer s employs the secret key S_s and the public keys $P_1, P_2, \ldots, P_r$ of all group members. Given a message m, the generation of ring signature is as follows.

Step 1 Calculate corresponding symmetric key $k = H(m)$ or $k = H(m, P_1, P_2, \ldots, P_r)$;
Step 2 Randomly select an initialization value v from $\{0, 1\}^b$;
Step 3 Randomly select x_i $(1 < i < r$, where $r \neq s)$ from $\{0, 1\}^b$ and calculate y_i as follows;

$$y_i = g_i(x_i), \text{ where } \begin{cases} x_i = q_i n_i + r_i, \quad 0 \le r_i \le n_i \\ g_i(x_i) = \begin{cases} q_i n_i + f_i(r_i) & \text{if } (q_i + 1)n_i \le 2^b \\ x_i & \text{else} \end{cases} \end{cases}$$

Step 4 Calculate y_s using $C_{k,v}(y_1, y_2, \ldots, y_r) = v$;
Step 5 Calculate the invertible function $x_s = g_s^{-1}(y_s)$ of $y_s = g_s(x_s)$;
Step 6 Output the ring signature $(P_1, P_2, \ldots, P_r; v; x_1, x_2, \ldots, x_r)$ for m as shown in Figure 2.

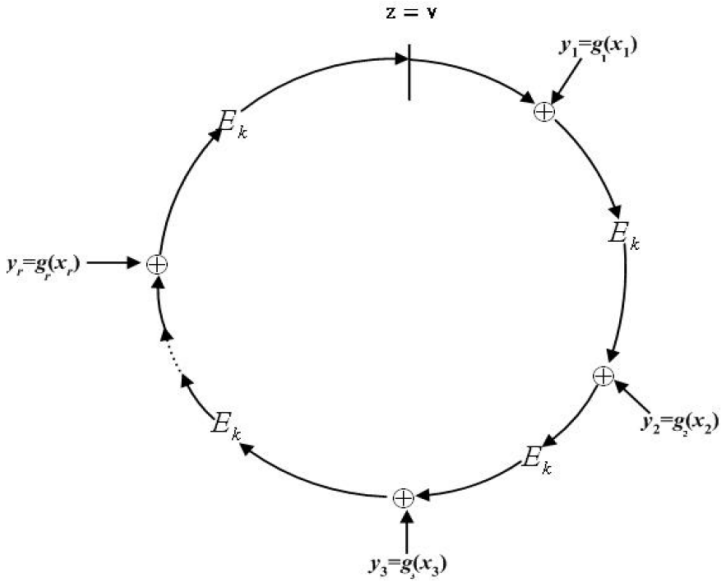

FIGURE 2. Ring signature

2.2.4. *Ring signature verification stage.* Upon receiving the ring signature $(P_1, P_2, \ldots, P_r; v; x_1, x_2, \ldots, x_r)$ for m, the verifier verifies it as follows.

Step 1 Calculate $y_i = g_i(x_i)$, where $i = 1, 2, \ldots, r$;
Step 2 Calculate the corresponding symmetric key $k = H(m)$;

Step 3 Take v and y_i $(i = 1, 2, \ldots, r)$ into the ring signature verification equation below.

$$C_{k,v}(y_1, y_2, \cdots, y_r) \stackrel{?}{=} v$$

If the equation is satisfied, the signature is taken as an authentic signature; otherwise, the signature is rejected.

3. **Proposed Schemes.** In Section 3.1, a new scheme related to threshold signature based on ECC is introduced. The new scheme achieves the security requirements of signature, unforgeability and undeniability; it is also highly efficient. In Section 3.2, an anonymous signcryption scheme that is also based on the ECC is presented; it integrates the advantages of a ring signature scheme, which is another group-oriented signature scheme. As to security, this proposed scheme not only meet confidentiality requirement but also possesses characteristics like unforgeability, anonymity, undeniability and forward secrecy.

The proposed technique presents the following advantages.

(1) Performance advantage: Since previous group signature schemes were on RSA or ElGamal cryptosystem, the overall computing strength of the proposed method has been enhanced by ECC. As ECC presents short session keys and low computation, the proposed technique shows superior performance to the previous ones.
(2) The network application environment: Past techniques could not be widely applied because of the burden of computation or the restrictions on application environment. The fixed machine or the limited computation could be simply developed for single technique or application. The proposed technique extends the range of applications and can be flexibly applied in various network environments such as electronic commerce or electronic voting.
(3) Dynamic access for group signature: Previous group signature used to apply threshold signature for a given group. The dynamic control on members was rather difficult and the security was threatened. The proposed new technique could achieve the function of group signature as well as enhance the dynamic access control on members.
(4) Anonymity and untraceability feature: The identities of group members can never be traced under our proposed scheme. The privacy of individual is maintained and anonymity and untraceability features are achieved.

3.1. **The group-oriented threshold signature scheme.** The proposed scheme requires a system center (SC) to generate the necessary parameters of the system and the users. Let a group of n signers be represented by $G_s = \{u_1, u_2, \ldots, u_n\}$, an association of any t members of which $(1 \leq t \leq n)$ can validly sign a message for the whole group; Let a group of l verifiers be represented by $G_v = \{u_{v1}, u_{v2}, \ldots, u_{vl}\}$, an association of any k members of which $(1 \leq k \leq l)$ can validate the received group signature on behalf of the verifier group. Then, these t signers jointly elect a clerk (CLK) among themselves to validate all individual signatures and combine t valid individual signatures into a group signature. The proposed scheme contains the following three phases.

3.1.1. *Parameter generation phase.* The SC is responsible for generating the required parameters of the system and the keys of the users. The generation phase involves the following.

(1) a field size p, which is a large odd prime;
(2) two field elements a and $b \in F_p$, which define the elliptic curve equation E over F_p, (i.e., $y^2 = x^3 + ax + b(\mathrm{mod}p)$ where $p > 3$ and $4a^3 + 27b^2 \neq 0(\mathrm{mod}p)$);
(3) a finite point $G = (x_g, y_g)$ whose order is a large prime number, where $G \neq O$ (O denotes an infinite point);

(4) the order of $G = q$;
(5) a one-way hash function h;
(6) two secret polynomials $f_s(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_1x + a_0 \mod q$ and $f_v(x) = c_{k-1}x^{k-1} + c_{k-2}x^{k-2} + \cdots + c_1x + c_0 \mod q$, where $a_i, c_j \in [1, q-1]$ for $i = 0, 1, 2, \ldots, t-1$ and $j = 0, 1, 2, \ldots, k-1$;
(7) a group private key $f_s(0) = a_0$ and a group public key $Y_s = f_s(0)G$ for G_s; and a group private key $f_v(0) = c_0$ and a group public key $Y_v = f_v(0)G$ for G_v;
(8) an individual private key $f_s(x_i)$ and a public key $y_i = f_s(x_i)G$ for each signer u_i in G_s, where $i = 1, 2, \ldots, n$ and x_i is the public value associated with each signer u_i. Assume that $x_1, x_2, \ldots, x_n$ are distinct.
(9) an individual private key $f_v(x_{vi})$ and a public key $y_{vi} = f_v(x_{vi})G$ for each verifier u_{vi} in G_v, where $i = 1, 2, \ldots, l$.

Then, the SC declares the system parameters p, E, G, q, h, y_i (for $i = 1, 2, \ldots, n$), y_{vi} (for $i = 1, 2, \ldots, l$), Y_s and Y_v public.

3.1.2. *Individual signature generation and verification phase.* Consider an arbitary association of t signers $\{u_{s1}, u_{s2}, \ldots, u_{st}\}$. To validly a sign message m, each signer u_{si} $(i = 1, 2, \ldots, t)$ generates an individual signature, as follows.

Step 1: Randomly select an integer $b_{si} \in [1, q-1]$, compute $B_{si} = b_{si}G$, and send B_{si} to the associates via a broadcast channel;
Step 2: Combine all received B_{si} $(i = 1, 2, \ldots, t)$ to obtain B as follows.

$$B = \sum_{i=1}^{t} B_{si} = (x_b, y_b)$$

Step 3: Compute the commitment value r_{si} using the private key $f_s(x_{si})$, the group public key Y_v of G_v, and the random integer b_{si}; then send r_{si} to the associates via a secure channel;

$$r_{si} = \left(b_{si} + f_s(x_{si}) \prod_{j=1, j\neq i}^{t} \frac{0 - x_{sj}}{x_{si} - x_{sj}} \right) Y_v$$

Step 4: Derive the common session key r of G_s and of G_v using the received r_{si} $(i = 1, 2, \ldots, t)$ to generate all individual signatures s_i and send s_i to the CLK, where

$$r = \sum_{i=1}^{t} r_{si} = (x_r, y_r)$$

$$s_i = x_b h(m) f_s(x_{si}) \prod_{j=1, j\neq i}^{t} \frac{0 - x_{sj}}{x_{si} - x_{sj}} - x_r b_{si} \bmod q,$$

Upon receiving all individual signatures for the message m, the CLK must validate each signature S_i using the following signature verification equation;

$$x_b h(m) \prod_{j=1, j\neq i}^{t} \frac{0 - x_{sj}}{x_{si} - x_{sj}} y_{si} \stackrel{?}{=} s_i G + x_r B_{si}$$

If the equation holds, the individual signature s_i is validated.

Theorem 3.1. *If the individual signature was generated by a valid signer, the signature verification equation holds.*

Proof:

$$
\begin{aligned}
& s_i = x_b h(m) f_s(x_{si}) \prod_{j=1, j\neq i}^{t} \tfrac{0-x_{sj}}{x_{si}-x_{sj}} - x_r b_{si} \bmod q \\
\Leftrightarrow\ & s_i G = x_b h(m) f_s(x_{si}) \prod_{j=1, j\neq i}^{t} \tfrac{0-x_{sj}}{x_{si}-x_{sj}} G - x_r b_{si} G \\
\Leftrightarrow\ & s_i G = x_b h(m) \prod_{j=1, j\neq i}^{t} \tfrac{0-x_{sj}}{x_{si}-x_{sj}} y_{si} - x_r B_{si} \\
\Leftrightarrow\ & x_b h(m) \prod_{j=1, j\neq i}^{t} \tfrac{0-x_{sj}}{x_{si}-x_{sj}} y_{si} = s_i G + x_r B_{si}
\end{aligned}
$$

3.1.3. *Group signature generation and verification phase.* If all of the t individual signatures are validated, the CLK computes the group signature s for the message m and sends it to the verifier group G_v as follows.

$$s = \sum_{i=1}^{t} s_i \bmod q$$

The CLK must declare B as public so that when the verification group G_v verifies the received group signature s, any k verifiers can verify it on behalf of verifier group. Each verifier u_{vi} $(i = 1, 2, \ldots, k)$ computes a commitment value r_{vi} using the private key $f_v(x_{vi})$, the public parameter B, and the group public key Y_s of G_s; he or she then sends r_{vi} to the associates via a secure channel, where

$$r_{vi} = f_v(x_{vi}) \prod_{j=1, j\neq i}^{k} \frac{0 - x_{vj}}{x_{vi} - x_{vj}} (B + Y_s)$$

To validate the group signature for message m, each associated verifier computes r after receiving all r_{vi} $(i = 1, 2, \ldots, k)$ as follows.

$$r = \sum_{i=1}^{k} r_{vi} = (x_r, y_r)$$

We note that this r value is equal to the r value computed by the signer group in Step 4 of the preceding subsection.

If the following verification equation holds, the group signature for message m is validated.

$$x_b h(m) Y_s \stackrel{?}{=} sG + x_r B$$

Theorem 3.2. *If the group signature indeed results from the valid signer group, the signature verification equation holds.*

Proof:

$$
\begin{aligned}
& \text{Since } s_i G = x_b h(m) f_s(x_{si}) \prod_{j=1, j\neq i}^{t} \frac{0 - x_{sj}}{x_{si} - x_{sj}} G - x_r b_{si} G \\
& \sum_{i=1}^{t} s_i G = \sum_{i=1}^{t} \left(x_b h(m) f_s(x_{si}) \prod_{j=1, j\neq i}^{t} \frac{0 - x_{sj}}{x_{si} - x_{sj}} G \right) - \sum_{i=1}^{t} (x_r b_{si} G) \text{ We have} \\
\Leftrightarrow\ & sG = x_b h(m) Y_s - x_r B \\
\Leftrightarrow\ & x_b h(m) Y_s = sG + x_r B
\end{aligned}
$$

3.2. **Group anonymous signcryption.** Elliptic curve cryptosystem has the advantages of high security, low computation load, and small bandwidth requirement, while ring signature protects the signer with its anonymity feature. Integrating elliptic curve cryptosystem and ring signature herein, the result is a system with highly secure and efficient anonymous signcryption scheme. The process comprises four steps, namely system construction, generation of signcryption text, verification of signcryption text, and conversion of signcryption text to standard signature.

3.2.1. *System construction.* Let q denote a large prime number, E denote an elliptic curve, P denote a base point on the elliptic curve E with order q, and H denote a one-way hash function, where q, E, P and H are public parameters, and Z_q is a finite field with q elements.

Let a group member set be $A = \{U_1, U_2, \ldots, U_n\}$ under the ECC, the private keys of Q_1, Q_2, $\ldots$, Q_n are d_1, d_2, $\ldots$, d_n respectively. The corresponding public keys Q_1, Q_2, $\ldots$, Q_n satisfy $Q_i = d_i P$, where $i = 1, 2, \ldots, n$. The private and public keys of verifier U_v are d_v and $Q_v = d_v P$, respectively.

3.2.2. *Generation of signcryption text.* Let a member U_i in A send the signcryption text of a message m to the verifier U_v. U_i executes the process of generating signcryption text as follows.

Step 1: Randomly select $k \in_R [1, q-1]$ and $r \in_R [1, q-1]$;
Step 2: Calculate $(x_i, y_i) = T_i = kP$, $(x_r, y_r) = R = rP$, and $(x_e, y_e) = T_e = rQ_v$;
Step 3: Select $s_t \in_R [1, q-1]$, where $t = i+1, i+2, \ldots, n, 1, \ldots, i-1$;
Step 4: Calculate sequentially $c_t = H(m||x_{t-1})$ and $(x_t, y_t) = T_t = s_t P + c_t Q_t$, where $t = i+1, i+2, \ldots, n, 1, \ldots, i-1$, and use $t-1 = n$ when $t = 1$;
Step 5: Calculate $c_i = H(m||x_{i-1})$ and $s_i = k - d_i c_i \pmod q$;
Step 6: Encrypt the message m to $m' = E_{x_e}(m)$ using the secret symmetric key x_e;
Step 7: Send the signcryption text $\sigma = (m', c_1, s_1, s_2, \ldots, s_n, R)$ to the verifier U_v.

3.2.3. *Verification of signcryption text.* Upon receiving the signcryption text $\sigma = (m', c_1, s_1, s_2, \ldots, s_n, R)$, the verifier U_v performs the following steps to verify.

Step 1: Let $(x_r, y_r) = R$, calculate $(x_d, y_d) = d_v R$ and $m'' = E^{-1}_{x_d}(m')$;
Step 2: For $t = 1, 2, \ldots, n-1$, calculate $(x_t, y_t) = T_t = s_t P + c_t Q_t$ and $c_{t+1} = H(m''||x_t)$;
Step 3: Calculate $(x_n, y_n) = T_n = s_n P + c_n Q_n$ and $c'_1 = H(m''||x_n)$;
Step 4: If $c'_1 = c_1$, confirm that $\sigma = (m', c_1, s_1, s_2, \ldots, s_n, R)$ is a valid anonymous signcryption text from the group $A = \{U_1, U_2, \ldots, U_n\}$; otherwise, reject the signcryption text.

3.2.4. *Conversion of signcryption text to standard signature.* Upon receiving signcryption text $\sigma = (m', c_1, s_1, s_2, \ldots, s_n, R)$, the verifier U_v applies the verification process in Section 3.2.3 to confirm the validity of signcryption text σ. Thus, m'' denotes the signed message from a group, and $\sigma' = (m'', c_1, s_1, s_2, \ldots, s_n)$ denotes the standard ring signature converted from σ, which is an ECC-based ring signature. Only verifier U_v can perform the signature conversion process. Any third party can verify the validity of the converted signature.

4. Analysis of Security.

4.1. **Analysis of group-oriented threshold signature.** The security of the proposed scheme in Section 3.1 is based on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP). Analyses of possible attacks and the security of the proposed scheme are presented below.

4.1.1. *Plaintext attack.* The plaintext attacks can be launched in different ways, such as by deriving individual private keys $f_s(x_{si})$ and $f_v(x_{vi})$ using the individual public keys $y_{si} = f_s(x_{si})G$ and $y_{vi} = f_v(x_{vi})G$, or by deriving group private keys $f_s(0)$ and $f_v(0)$, using the group public keys $Y_s = f_s(0)G$ and $Y_v = f_v(0)G$. An attacker may attempt to derive a signer's private key $f_s(x_{si})$ or the verifier's private key $f_v(x_{vi})$ from the commitment value $r_{si} = \left(b_{si} + f_s(x_{si}) \prod_{j=1,j\neq i}^{t} \frac{0 - x_{sj}}{x_{si} - x_{sj}}\right) Y_v$ or $r_{vi} = f_v(x_{vi}) \prod_{j=1,j\neq i}^{k} \frac{0 - x_{vj}}{x_{vi} - x_{vj}} (B + Y_s)$. Such attacks are infeasible against the ECDLP.

4.1.2. *Forgery attack.* Assume that an attacker intends to forge the individual signature S_i of the signer u_{si}. Firstly, the attacker may randomly select an integer b_{si} and compute the corresponding $B_{si} = b_{si}G$; then he forges a commitment value r_{si}. Since he doesn't hold the signer u_{si}'s private key $f_s(x_{si})$, he fails to generate a valid signature s_i to satisfy the following verification equation:

$$x_b h(m) \prod_{j=1,j\neq i}^{t} \frac{0 - x_{sj}}{x_{si} - x_{sj}} y_{si} \stackrel{?}{=} s_i G + x_r B_{si}.$$

Alternatively, the attacker may intend to forge a group signature s for any arbitrary message m to satisfy the following verification equation of group signature:

$$x_b h(m) Y_s \stackrel{?}{=} sG + x_r B$$

First, he may randomly select an integer x_r, a false signature$s$ and a point $B = (x_b, x_b)$ to compute a value Q, where Q should satisfy both $Q = h(m)$ and the verification equation of group signature. Nevertheless, due to the onewayness property of hash function and the difficulty of solving the ECDLP, the possibility of successful forging of a valid Q is extremely small. An attacker may randomly select two integers x_r and $h(m)$ and a point $B = (x_b, x_b)$ and derive a value s that satisfy the verification equation of group signature. However, the difficulty of solving the ECDLP makes it infeasible.

4.1.3. *Equation attack.* Assume that an attacker intends to derive the signer u_{si}'s private key $f_s(x_{si})$ and secret parameter b_{si} from the generating equation of individual signature s_i below:

$$s_i = x_b h(m) f_s(x_{si}) \prod_{j=1,j\neq i}^{t} \frac{0 - x_{sj}}{x_{si} - x_{sj}} - x_r b_{si} \bmod q$$

Since both $f_s(x_{si})$ and b_{si} in the equation are parameters unknown to the attacker, such attack is thus in feasible. The attacker may try to derive $f_s(x_{si})$ from a set of generating equations with a set of plaintexts m' and the corresponding individual signature si'. However, each such equation contains a unknown parameter b'_{si}. Consequently the number of unknown secret parameters is always greater than the number of equations which makes the equation attack infeasible. Also, the attacker may derive both the group private key $f_s(0)$ and the secret parameter $\sum_{i=1}^{t} b_{si}$ by combining t individual signatures into one equation as follows:

$$\sum_{i=1}^{t} s_i \bmod q = \left(x_b h(m) f_s(0) - x_r \sum_{i=1}^{t} b_{si}\right) \bmod q$$

Since this equation contains two unknown $f_s(0)$ and $\sum_{i=1}^{t} b_{si}$, the attacker cannot successfully break the equation.

4.1.4. *Conspiracy attack.* Assume that there are $t-1$ members who conspire to derive the private key of some signer and the group private key of the signer group G_s. These $t-1$ conspiring members have to first reconstruct the polynomial function $f_s(x)$, then compute their target's private key $f_s(x_{si})$ via his public value x_{si} in the signer group or derive the group private key $f_s(0)$ of G_s. The reconstruction of the polynomial function $f_s(x)$ requires at least t members' private keys $f_s(x_i)$. Therefore, though these $t-1$ conspiring members share their private keys $f_s(x_i)$ with each other, they still cannot successfully reconstruct the polynomial function $f_s(x)$, or derive their target's and the group's private keys. The same arguments also apply to the verifier group G_v; any $k-1$ or fewer conspiring members will fail to derive the others' private keys.

4.2. **Analysis of group anonymous signcryption.** This proposed method combines the ECC-based system, ring signature and symmetric encryption, attaining properties such as confidentiality, unforgeability, anonymity, undeniability and forward secrecy.

4.2.1. *Confidentiality.* Message m is sent in ciphertext form so that only those with a secret symmetric key can decrypt it. As to the session key, it is encrypted using the public key of the verifier before it is sent to the verifier. So far an ECC-based public key infrastructure remains secure, thus only verifier U_v can decrypt the message m from the ciphertext.

4.2.2. *Unforgeability.* We first argue that an ECC-based ring signature is unforgeable in the random oracle model. It the follows that our proposed ECC-based signcryption is also unfogeable.

(1)Unforgeability of an ECC-based ring signature

In a random oracle model, consider the ring signature algorithm SIG of the proposed method along with the one-way hash function H as an oracle. Supposing that an algorithm A applies the public keys $Q_1, Q_2, \ldots, Q_n$ as inputs with no knowledge of a valid private key, it then makes requests to SIG and H using a polynomial sequence. That is, A might be able to forge the ring signature for a message m with non-negligible probability. Also, consider an algorithm B, which employs a random point Q over the elliptic curve E as input and calculates s with non-negligible probability satisfying $Q = sP$, attempting to solve the ECDLP.

Assume that the algorithm B can perform a black-box interview with algorithm A and has total control over the requests from the algorithm A. B demands that A makes its request to H by following the direction of the ring built for the forged signature on message m; otherwise, the probability of the forged signature passing verification is negligible [2]. Assume that A sends a request to H following a clockwise or anti-clockwise direction. After A making a polynomial sequence of requests (testing several messages m in the process), B can guess, with non-negligible probability, that A forged the signature on message m. However, B can neither guess which requests were proposed by A in the latest forged signature nor determine the order of requests on the ring. For the other m_j, algorithm B can easily imitate SIG to produce a signature; vector $(c, s_1, s_2, \ldots, s_r)$ is output as the ring signature; the order of the random responses are simultaneously mixed up to enable the signatures of these messages to pass verification. Since B randomly selects the value following the ring structure to generate the signature for m_j, A cannot propose requests that prevent B from selecting a value that A can guess in advance.

Algorithm B randomly selects an insertion point for Q following the direction of the ring, and uses the insertion point to fill the gap between the input and the output values of two continuous hash operations in generating the final forged signature. This approach also forces A to provide the corresponding s, which satisfies to $Q = sP$, thus the gap is

sealed during the signature forgery process. Since only B knows the random value Q, A does not recognize this "trap" and refuses to provide the forged signature.

The main difficulty is that A can determine the inverse using a one-way hash function and can seal the ring using the SIG algorithm by following the direction that is the easiest to compute. This difficulty can be overcome by noting that a gap always exists between the two H values in any signatures forged by A. Irrespective of the order followed by A when sending requests to B, B can still respond to these requests. Additionally, B can answer the second of two adjoining requests based on the input and the output of the previous similar requests. Under this method, B needs only perform an addition operation on the two ends to obtain the desired value of Q, which also forces A to compute s such that it satisfies $Q = sP$ in the final forgery process to seal the gap.

B cannot determine which request was applied by A to the final forgery of signature, and can only make guesses. However, B can only attempt two guesses. The probability of success is $1/2^T$, where T denotes the total number of requests made by A. Consequently, B can compute, with non-negligible probability, the corresponding s that satisfies $Q = sP$ and thus successfully solve the ECDLP.

Above arguments imply that if A can successfully propose forged signature to B with a non-negligible probability, B has a non-negligible probability of solving the ECDLP. This contradicts our current knowledge of ECDLP. Therefore, the ECC-based ring signature cannot be forged.

(2) Unforgeability of an anonymous ECC-based signcryption

If a user U successfully forges an anonymous signcryption text, then he will be able to convert the signcryption to a ring signature to create a forged ECC-based ring signature. This contradicts the results of Section 4.2.2 (1); that is, anonymous signcryption is unforgeable.

4.2.3. *Anonymity.* The main difference between the proposed scheme and other signcryption schemes lies in anonymity of the signer. Upon receiving the signcryption information, a verifier can authenticate the validity of the signcryption information, but cannot identify the signer. As to the anonymity between the signer and third party, after the verified signcryption information has been converted to a ring signature, a third party can only check which group the signature is produced, and whether the signature is issued by a valid member of that group; the third party cannot determine the identity of the signer. In other words, neither the verifier nor the third party can identify a signer using the signcryption information.

4.2.4. *Undeniability.* When a conflict arises, the verifier can convert the signcryption text to a standard ring signature. Any third party can validate this ring signature and confirm the source of the signature. Although the identity of the signer cannot be determined, the group to which the signer belongs can be identified. The signcryption could neither be forged by the verifier, nor be generated by a non-member. Therefore, undeniability of signature can be established since the group members cannot deny the signature.

4.2.5. *Forward secrecy.* With regard to the forward secrecy security feature provided by signcryption, an attacker cannot use the signer's long-term private key, a previously transmitted signcryption text σ, or related public information to carry out the calculations to obtain the plaintext intended for the verifier. In the proposed scheme, because the corresponding key x_e of the final encrypted plaintext is generated with a random number r, a different decryption key is generated every time. Thus, even if the long-term private key d_i of a signer U_i is revealed, previous documents remain secure, achieving forward secrecy.

5. **Conclusions.** RSA-based or ElGamal-based group signature schemes utilized the mathematical differently in solving discrete logarithm problem to satisfy the security requirement. In practical applications, they required large module and heavy computation load. Consequently, these schemes sometimes failed to work smoothly, especially in environments without excellent computing ability. Our proposed group signature scheme is a ECC-based scheme which also incorporates the features of ring signature. The basic requirements of availability, non-repudiation, anonymity, and confidentiality are proved to be satisfied by our group signature scheme. The integration of ring signature therefore is the key reason that the proposed technique presents the efficacy and features of group signature as well as the advantages of short session keys and low computation, reduces operated resources, and enhances system performance. It further reveals advantages on business applications, such as trade transactions among banks or electronic commerce applications which require more efficient information security protection.

The proposed group-oriented threshold signature scheme can specify a verifier group. That is, only a specific verifier group can verify the group signature. Moreover, the integration of the ECC into the system makes the cryptosystem more secure and efficient. The other group signature related anonymous ECC-based signcryption has anonymity as its main property. The anonymity factor through the integration of the attributes of ring signatures–protecting the privacy of signers by signing anonymously is achieved. Furthermore, this proposed scheme also achieves the requirements of confidentiality, unforgeability, undeniability, and forward secrecy.

Acknowledgment. The authors would like to thank the National Science Council, Taiwan, for financially supporting this research under Contract No. NSC 100-2410-H-029 -007.

REFERENCES

[1] W. Diffie and M. Hellman, New directions in cryptography, *IEEE Trans. on Information Theory*, vol.IT-22, no.6, pp.644-654, 1976.

[2] R. Rivest, A. Shamir and Y. Tauman, How to leak a secret, *Proc. of Asiacrypt, LNCS*, vol.2248, pp.552-565, 2001.

[3] M. Mambo, K. Usuda and E. Okamoto, Proxy signature: Delegation of the power to sign messages, *IEICE Trans. on Fundamentals*, vol.E79-A, no.9, pp.1338-1353, 1996.

[4] Y. Desmedt and Y. Frankel, Shared generation of authenticators and signatures, *Proc. of Conference on Advances in Cryptology-Crypto'91, LNCS*, vol.576, pp.457-469, 1992.

[5] Y. Zheng, Digital signcryption or how to achieve cost (Signature & Encryption)<< Cost(Signature) + Cost(Encryption), *Proc. of Conference on Crypto'97, LNCS*, vol.1294, pp.165-179, 1997.

[6] J. M. Lee and W. Mao, Two birds one stone: Signcryption using RSA, *Proc. of CT-RSA '03, LNCS*, vol.2612, pp.211-225, 1998.

[7] B. Libert and J. J. Quisquater, New identity based signcryption schemes from pairings, *ITW*, 2003.

[8] D. H. Yum and P. J. Lee, New signcryption schemes based on KCDSA, *Proc. of ICISC '01, LNCS*, vol.2288, pp.205-317, 2002.

[9] P. S. Barreto, H. Y. Kim and M. Scott, Efficient algorithms for pairing-based cryptosystems, *Proc. of Conference on Advances in Cryptology-Crypto'02, LNCS*, vol.2442, pp.354-368, 2002.

[10] S. D. Galbraith, K. Harrison and D. Soldera, Implementing the tate pairing, *ANTS, INCS*, vol.2369, pp.324-337, 2002.

[11] F. Zhang and K. Kim, ID-based blind signature and ring signature from pairings, *Advances in Cryptology-Asiacrypt'02, LNCS*, vol.2501, pp.533-547, 2002.

[12] C.-Y. Lin and T.-C. Wu, An identity-based ring signature scheme from bilinear pairings, *Cryptology ePrint Archive*, 2003.

[13] S. Chow, L. Hui and S. Yiu, Identity based threshold ring signature, *Information Security and Cryptology, LNCS*, vol.3506, pp.218-232, 2004.

[14] J. Xu, Z. Zhang and D. Feng, A ring signature scheme using bilinear pairings, *The 5th International Workshop on Information Security Applications, LNCS*, vol.3325, pp.163-172, 2004.

[15] S. Chow, S. Yiu and L. Hui, Efficient identity based ring signature, *Proc. of the 3rd International Conference on Cryptography and Network Security, LNCS*, vol.3531, pp.499-512, 2005.
[16] J. Liu, V. Wei and D. Wong, A separable threshold ring signature scheme, *Proc. of the 6th International Conference on Information Security and Cryptology, LNCS*, Seoul, Korea, vol.2971, pp.352-369, 2003.
[17] A. Cilardo, L. Coppolino, N. Mazzocca and L. Romano, Elliptic curve cryptography engineering, *Proc. of the IEEE*, vol.94, no.2, pp.395-406, 2006.
[18] A. Fernandes, Elliptic curve cryptography, *Dr. Dobb's Journal*, 1999.
[19] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 3rd Edition, Prentice Hall, 2003.
[20] Y. Desmedt, Society and group oriented cryptography: A new concept, *Proc. of Crypto on Advances in Cryptology*, pp.120-127, 1987.
[21] A. Shamir, How to share a secret,*Commun. ACM*, vol.22, pp.612-613, 1979.
[22] L. Harn, Group-oriented (t, n) threshold signature and digital multisignature, *Proc. of IEEE Conference on Comput. Digit. Tech*, vol.141, no.5, pp.307-313, 1994.
[23] C.-C. Chang, J.-J. Leu, P.-C. Hwang and W.-B. Lee, A scheme for obtaining a message from the digital multisignature, *International Workshop on Practice and Theory Public Key Cryptography*, pp.154-163, 1998.
[24] L. Harn, Digital signature with (t, n) shared verification based on discrete logarithms, *Electron. Lett.*, vol.29, no.24, pp.2049-2095, 1993.
[25] P. Hoster, M. Michels and H. Peterson, Comment: Digital signature with (t, n) shared verification based on discrete logarithms, *Electron. Lett.*, vol.31, no.14, pp.1137, 1995.
[26] S.-J. Hwang, C.-C. Chang and W.-P. Yang, Authenticated encryption schemes with message linkage, *Inf. Process. Lett.*, vol.58, pp.189-194, 1996.
[27] W.-B. Lee and C.-C. Chang, Comment: Digital signature with (t, n) shared verification based on discrete logarithms, *Electron. Lett.*, vol.31, no.3, pp.176-177, 1995.
[28] W.-B. Lee and C.-C. Chang, Authenticated encryption scheme without using a one-way function, *Electron. Lett.*, vol.31, no.19, pp.1656-1657, 1995.
[29] C.-M. Li, T. Hwang and N.-Y. Lee, Threshold multisignature scheme where suspected forgery implies tractability of adversarial shareholders, *Proc. of Eurocrypt '94 on Advances in Cryptology*, pp.194-203, 1995.
[30] C.-T. Wang, C.-C. Chang and C.-H. Lin, Generalization of threshold signature and authenticated encryption for group communications, *IEICE Trans. Fundamentals*, vol.E83-A, no.6, pp.1228-1237, 2000.
[31] C.-L. Hsu, T.-S. Wu and T.-C. Wu, Improvements of generalization of threshold signature and authenticated encryption for group communications, *Inf. Process. Lett.*, vol.81, pp.41-45, 2002.
[32] V. S. Miller, Uses of elliptic curves in cryptography, *Proc. of Conference on Advances in Cryptology-CRYPTO, LNCS*, New York, no.218, pp.417-426, 1985.
[33] N. Koblitz, Elliptic curve cryptosystems, *Mathematics of Computation*, vol.48, pp.203-209, 1987.
[34] A. J. Menezes, T. Okamoto and S. A. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Trans. on Information Theory*, vol.39, pp.1639-1646, 1993.
[35] J. S. Brickell and K. S. McCurely, ECC: Do we need to count? *Advances in Cryptology-ASIACRYPT, LNCS*, no.1716, pp.122-134, 1999.
[36] Y.-F. Chung and K.-H. Huang, Chameleon signature with conditional open verification, *International Journal of Innovative Computing, Information and Control*, vol.5, no.9, pp.2829-2836, 2009.
[37] J.-H. Yang and C.-C. Chang, An efficient fair electronic payment system based upon non-signature authenticated encryption scheme, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3861-3874, 2009.
[38] Y.-M. Tseng, T.-Y. Wu and J.-D. Wu, An efficient and provably secure id-based signature scheme with batch verifications, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3911-3922, 2009.
[39] M.-S. Hwang, S.-F. Tzeng and S.-F. Chiou, A non-repudiable multi-proxy multi-signature scheme, *ICIC Express Letters*, vol.3, no.3(A), pp.259-264, 2009.