An Efficient and Secure Weighted Threshold Signcryption Scheme

Chien-Hua Tsai

Department of Accounting Information, Chihlee University of Technology, Taiwan
chienhua@mail.chihlee.edu.tw

Abstract

Threshold cryptosystems are well suited for the application of group-oriented collaborations which is of such a nature that specific cooperation should be established in connection with a number of organizational tasks, such as multigroup computation, threshold signature, and electronic information transfer. Over the years many researchers have made salient contributions to the concept of allowing group members to sign or encrypt digital messages on behalf of the entire group. Yet almost all the group-oriented signature or encryption cryptosystems have been developed on the premise that members of the party have the same weights (fixed-size thresholds) and the tasks (shared secrets), cannot be adjusted. This study proposes a new weighted threshold signcryption scheme based on elliptic curve cryptography (ECC) with a dynamic knapsack-type technique and the Chinese remainder theorem (CRT) to customize its weight values of group members in such a way that they are appropriately arranged to perform a more flexible group-oriented signcryption task on behalf of a group.

Keywords: Threshold cryptosystem, Elliptic curve cryptography, Dynamic knapsack cryptosystem, Chinese remainder theorem, Weighted threshold signcryption

1 Introduction

In today's organizations where the use of teams is increasingly widespread, important decision making that was once reserved for a single individual is more often than not now made by teams [19]. In situations that specifically involve both the identification of authorized members and the access privileges of sensitive or critical data group decision-making actually often faces the challenge of information sharing. Therefore, secret sharing or threshold schemes are appropriate to be applied to such applications. The threshold secret sharing schemes were introduced independently by Blakley [4] and Shamir [33] in 1979. Blakley's threshold scheme is based on a hyperplane geometry subspace to create a specific point of intersection as the secret in t-dimensional hyperplanes and to recover it from n distinct hyperplanes (shares). And Shamir's secret sharing scheme is based on a polynomial interpolation method to produce a $(t-1)th$ degree polynomial with the secret and to reconstruct it from any t out of n shares. Unlike Shamir's and Blakely's schemes, Mignotte's and Asmuth-Bloom's approaches [1, 28] suggest another two threshold secret sharing schemes based on the Chinese remainder theorem, and employ a special sequence of coprime positive integers to construct the secret given any of the n choose $t-1$ shares, but will not reveal it less than any choice of $t-1$ of them. In order to design more efficient secret sharing schemes, several well-known advanced strategies based on Blakley's [5, 14], Shamir's [9, 22, 27], and Mignotte's/Asmuth-Bloom's [16, 18, 23] protocols, respectively, have been proposed in the literature.

In addition to delivering high efficiency benefits, security requirements have also become a great concern within a threshold cryptosystem. For the reason, Desmedt and Frankel [10] in 1991, present a (t, n) threshold signature scheme based on the RSA cryptosystem to secure authenticators. Since then, more improvements and variants of the threshold signature schemes have been proposed [6, 15, 26]. Most of the existing threshold signature solutions are based on the equivalent-weight assumption with respect to each group member. However, there are some of the real life applications, that depend on different weight values such as authorizing an e-transfer on financial transactions, making critical decisions in developing and influencing organizational activities, or launching various dedicated weapon systems in legal, ethical and operational norms. The idea of the weighted threshold secret sharing method was offered in the late 1990s by Morillo et al. [31], who proposed that a weighted threshold scheme assigns different weights to involved members, and a subset of the participants is authorized to reconstruct the secret if the total combined weights are greater than or equal to the threshold. After that, numerous other studies contribute to this idea [11-12, 24, 35], which

*Corresponding Author: Chien-Hua Tsai; E-mail: chienhua@mail.chihlee.edu.tw
DOI: 10.3966/160792642019092005018

have brought much-needed attention to weight assignment applications. For example, Li et al. [24] introduce the weighted threshold Mignotte sequence into secret image sharing, where the participants share different shadow images and the secret image can be reconstructed losslessly if and only if the sum of all of the shadow images' weights is no less than the given weight threshold; whereas Dikshit and Singh [11] present a weighted threshold scheme with bitcoin wallets using elliptic curve digital signature algorithm, in which all the players get unequal priorities and the method can provide a higher level of security to bitcoin transactions.

Recently, two articles by Iftene and Grindei [17] and Guo and Chang [13] apply the digital signature technology to this area of the weighted threshold scheme. Iftene and Grindei's method combines the weighted threshold secret sharing scheme with the RSA public-key cryptosystem to provide the weighted threshold decryption as well as generating a digital signature inside its security perimeter. Guo and Chang's algorithm employs a proactive weighted signature skill based on the generalized Chinese remainder theorem, to offer the group members the use of RSA-based signature for secure mission-critical documents. Although the correctness of the weighted threshold signature schemes has been explicitly proven and assured given in these studies, it is generally considered that most problems relevant for the security breaches still remain possible. Additionally, if using a better solution offers further performance improvements for almost all RSA-based weighted threshold protocols, this significantly enhances critical information on weight distribution, security and efficiency in applications of secret sharing schemes. Naturally, it takes the ECC-based [20, 30] weighted threshold signcryption [38] operation from an RSA-based weighted threshold signature perspective into consideration.

This paper concentrates on weighted threshold signature schemes. In order to provide an efficient and secure solution with respect to the secret share dissemination, we apply an ECC-based signcryption technique and also incorporate the dynamic knapsack cryptosystem approach with the Chinese remainder theorem as well as the ECC to construct a secure and fast weighted threshold cryptosystem. In addition, the weight of each group member remains unknown from each other except the designated dealer in our scheme, while two of the existing weighted threshold signature works have publicly disclosed the weights of group members involved. Having designed such features for the weighted threshold signature solution, this proposed protocol has the ability to reduce the consequences of making poor group decisions such as ignoring the lower weight voice, tending to achieve the higher weight towards a decision, or arising from the formation of sub-coalitions and collusion.

The main contributions of this study can be summarized as follows: 1. The mechanism enabling the reliable transmission of critical information performs much better than the existing RSA-based approaches in terms of security properties. 2. This efficient but elegant structure of the fast computation of elliptic curves provides significant performance improvements over the existing similar algorithms in its resource consumption [25, 32]. 3. The rigorous and effective design of the authentication method is able to avoid the collusion of intentional members to make sure that each member of the group has adequate opportunity providing their comments during the group decision-making process. The remainder of the paper is divided into five sections. Section 2 provides background information that links theoretical prerequisites to the study. Section 3 presents the methodology of a new weighted threshold signcryption work based on ECC. Section 4 analyzes the security of the proposed scheme, and Section 5 covers the result of experiments along with computational efficiency and performance evaluation. Finally, conclusions and future research areas appear in Section 6.

2 Preliminary Backgrounds

To begin with, this section presents a quick primer on the concept of dynamic knapsack cryptosystem whose structure is difficult but complex. Then, a basic understanding of the threshold secret sharing scheme will be included in this section. These techniques and methods are appropriately utilized in this paper, and are thoroughly described in Section 3 and Section 4.

2.1 Dynamic Knapsack Cryptosystem

In 1988, Chor and Rivest [7] proposed a public key cryptosystem of high-density dynamic knapsack, which has several advantages, including fast, efficient, and secure calculations that the knapsack system offers. This algorithm has been proven resistant to many classes of attacks, for instance, low density attack which can be done by requesting a direct solution to the knapsack problem. This category contains the following known operations.

- Select two random knapsack vectors $U=(u_1, \ldots, u_n)$ and $V=(v_1, \ldots, v_n)$ where $v_i = u_i - 2^{n-i}, i = 1, \ldots, n$.
- Generate a two-dimensional reversible matrix W whose elements are integers, and is denoted as W^{-1}.
- Calculate the transposed matrix $(G, H)^T = W(U, V)^T$ to get two transformed knapsack vectors $G=(g_1, \ldots, g_n)$ and $H=(h_1, \ldots, h_n)$.
- Choose two arbitrary prime numbers p and q, that satisfy the given conditions $p > 2\max\{\sum_{g_i>0} g_i, -\sum_{g_i<0} g_i\}$ and $q > 2\max\{\sum_{h_i>0} h_i, -\sum_{h_i<0} h_i\}$.
- Use the Chinese remainder theorem to generate

another random vector $A=(a_1, \ldots, a_n)$ where $0 \le a_i \le pq-1$, and compute a_i such that $a_i \equiv g_i \pmod p$ and $a_i \equiv h_i \pmod q$ for $i = 1, \ldots, n$. Also, let A be a public key while p, q and W^{-1} are the private keys.

- For the encryption, let $m = m_1\, m_2 \ldots m_n$ where m_i is an n-bit long binary plaintext, and the ciphertext c is then produced as $a_1m_1 + a_2m_2 + \ldots a_nm_n$.
- In the process of decryption, require to compute $c_p = c \pmod p$ and $c_q = c \pmod q$, where c_p and c_q take the sum of the absolute minimum remainders of modulo the primes p and q.
- Reckon the equation for the secret knapsack vector by $(s_p, s_q)^T = W^{-1}(c_p, c_q)^T$ where s_p and s_q stand for the corresponding message before binary transfer, and thus $m = (s_p - s_q)$ is given in return the original binary plaintext.

The deeper the matrix transformation $(G, H)^T = W(U, V)^T$ uses, the more the relationship between the private vectors of $U = (u_1, \ldots, u_n)$ and $V = (v_1, \ldots, v_n)$ complicates apparatus. Therefore, the improved knapsack calculation has a security intensity of higher resistance of against key recovery attack.

2.2 Threshold Secret Sharing Scheme

A secret sharing scheme is a method used in several cryptographic protocols and distributes a secret among a group of participants, each of whom is allocated a share of the secret. The secret can only be reconstructed from its shares when the shares are combined together, and any participant in the group cannot reveal any partial information on the secret. Such a technique is called a (t, n)-threshold scheme [3]. The essential idea of secret-sharing threshold scheme is as follows.

- Suppose using a (t, n) threshold to share the secret S.
- Pick a large prime number p which is a positive integer in a finite field Z_p^*.
- Let a_0 be the shared secret (i.e., $a_0 = S$) and t be the number of players needed to reconstruct the secret where $0 < t \le n < p$ and $a_0 < p$.
- Choose $t-1$ random integers $a_1, a_2, \ldots, a_{t-1}$ with $a_i < p$, and compute $a_1, a_2, \ldots, a_{t-1} \bmod p$.
- Build the polynomial $f(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{t-1}x^{t-1}$.
- Construct n pieces from the polynomial to obtain $(i, f(i))$ where $i \ne 0$.
- Give each player a different pair of the form $(x, f(x))$ and find the coefficients of the polynomial.
- Reconstruct the secret S (i.e., $f(0)$) from any subset of t of these pairs.

If $t = n$ then all players are required to reconstruct the secret. Now each player has a distinct pair for producing secret shares. Therefore, the secret S can be derived from t of these pairs using the interpolation polynomial in the Lagrange form.

3 The Proposed Weighted Threshold Signcryption Scheme

In this section, we present a new group-oriented weighted threshold signcryption scheme with adjustable values, which combines three cryptographic techniques with the secret sharing method for transmission of digital messages. One is the elliptic curve discrete logarithm problem (ECDLP), another is the dynamic knapsack problem (DKP) and the other is the Chinese remainder theorem (CRT). The proposed mechanism consists of the following seven phases: initial phase, registration phase, encryption phase, authentication phase, participant selection phase, weight distribution phase, signature production phase. There are four kinds of roles in our group-oriented weighted threshold signature scheme, namely a group A of n members, a dedicated administrator GM_A as a dealer, a sender C as a user outside the group A, and a certificate authority CA as a trusted third party, respectively. The operational context diagram of this scheme is shown in Figure 1, and Table 1 lists the symbols and the denotations thereof about the protocol used.

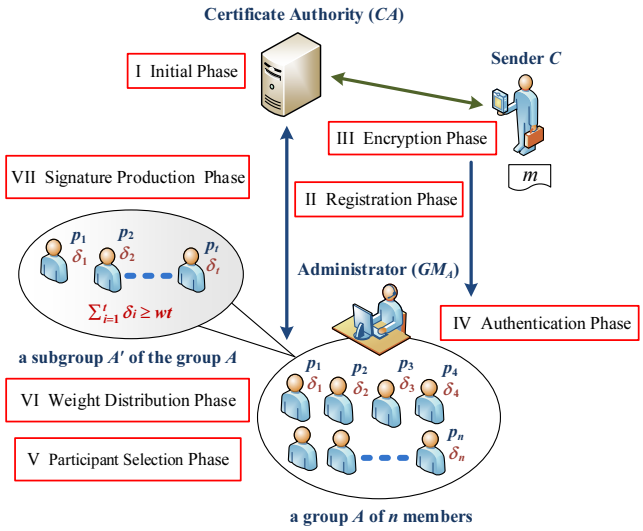

Figure 1. The operational context diagram of the proposed model

Table 1. The system parameters and the meanings

Item	Notation	Description
1	$E(F_q)$	an elliptic curve E over a finite field F_q
2	G	a base point G of an elliptic curve
3	h	an order n of an elliptic curve
4	q	a large prime number q such that $q > 2^{256}$
5	$H(\,)$	a universal one-way hash function
6	Y_i	public keys of a certificate authority CA, an administrator GM_A, a sender C, a dedicated group A, and selected members of a group

Table 1. The system parameters and the meanings (continue)

7	d_i	private keys of a certificate authority *CA*, an administrator GM_A, a sender *C*, a dedicated group *A*, and selected members of a group
8	ID_i	digital identities for relevant entities
9	l_i, r_i, τ_i, k_i	random integer values
10	α, β	large prime numbers
11	u_i, v_i,	confusion values
12	σ_i	digital credentials issued by GM_A for group members
13	ca_i	identity certificates for an administrator GM_A and a sender *C*
14	ρ_i, θ_i	weight-encoded valves
15	δ_i	weighted values for each individual participant
16	*wt*	a globally weighted threshold value
17	(*r*, *s*)	a weighted threshold signature on behalf of the group

3.1 Initial Phase

First, a secure elliptic curve $E(F_q)$ is defined over a finite field F_q, where q is a large prime number such that the number equals approximately 256 bits, i.e., a 256-bit key in ECC is considered to be as secured as 3072-bit key in RSA [2]. Next, an order h will be given, together with the base point G on the elliptic curve $E(F_q)$, and the proper choice satisfies $h \cdot G = O$, where O is the point at infinity. Then, the *CA* chooses a one-way hash function $H()$. After that, to generate a public-private key pair (Y_{CA}, d_{CA}), the *CA* uses a secret number of d_{CA} as the private key, and the associated public key is the point $Y_{CA} = d_{CA} \cdot G$. Finally, the global system parameters are publicly known to all parties including $E(F_q)$, G, h, Y_{CA} and $H()$.

3.2 Registration Phase

The members p_i, $i = 1, \ldots, n$, of the group *A* (e.g., the administrator GM_A) and external users (e.g., a sender *C*) all need to register the knowledge of their identities and relationships with the *CA* as legitimate participants before the cryptographic or sharing process. Since this way all applications are dealing with the same rules for all users, we hereby describe the GM_A's activity applied to make the process easy to understand. For the registration phase, the following steps are performed.

- First, GM_A registers his/her identity information, e.g., the account ID_{GM_A}, to *CA*, and creates the individual public-private key pair (Y_{GM_A}, d_{GM_A}) which satisfies equation (1).

$$Y_{GM_A} = d_{GM_A} \cdot G \tag{1}$$

- Next, *CA* takes ID_{GM_A} and the public key Y_{GM_A} to compute the association value e_{GM_A} according to equation (2) for GM_A.

$$e_{GM_A} = H(ID_{GM_A} \| Y_{GM_A}) \tag{2}$$

- Then, *CA* randomly selects an integer $l_{GM_A} \in Z_q^*$ as a secret authentication argument to calculate the GM_A's respective point, W_{GM_A}, such that $W_{GM_A} = (x_{GM_A}, y_{GM_A})$, by equation (3).

$$W_{GM_A} = l_{GM_A} \cdot G = (x_{GM_A}, y_{GM_A}) \tag{3}$$

- Last, *CA* issues the certificate of identity, ca_{GM_A}, to GM_A using equation (4).

$$ca_{GM_A} = l_{GM_A} \cdot (e_{GM_A} + x_{GM_A} \cdot d_{CA})^{-1} \tag{4}$$

3.3 Encryption Phase

When a user outside the group *A*, i.e., the sender *C*, tends to make communication among members p_i of the group, they need to agree on many different parameters and then exchange a series of information about encryption. This process is performed to provide high security in the encrypted message, *m*, as follows.

- First of all, *C* registers his/her identity information to *CA*, and creates the individual public-private key pair (Y_C, d_C) in the same style as equation (1).
- Second, *C* takes a random integer r_C chosen from Z_q^* and converts it to a nonce point on an elliptic curve using equation (5). Doing this as a one-time token value can effectively prevent replay attacks.

$$R_C = r_C \cdot G = (x_{R_C}, y_{R_C}) \tag{5}$$

- And then, to achieve high security level, the *x*-coordinate of a point on an elliptic curve must not equal zero, where the condition needs to be met. This ensures that $E(F_q)$ is non-singular and has no repeated roots. That is to say, while x_{R_C} equals zero, *C* will repeat the preceding equation again. Thus, *C* combines the hash value $H(m)$ with GM_A's public key Y_{GM_A} to encrypt *m* by performing the addition of two points (m, $H(m)$) and ($r_C \cdot Y_{GM_A}$) on $E(F_q)$. According to Menezes-Vanstone primes [21], there is no need for mapping a point on $E(F_q)$ and this form of (m, $H(m)$) can significantly improve the performance of the scalar multiplication operation. Therefore, it proceeds by equation (6).

$$M = (m, H(m)) + r_C \cdot Y_{GM_A} \tag{6}$$

- After that, *C* produces a digital signature s_C on the message from equation (7), by applying his/her private key d_C and r_C. The digital signature is used to verify whether the original message sent by the sender is valid- made by the owner of the corresponding private key, and it can assist in a non-

repudiation argument. Note that equation (6) and equation (7) simultaneously fulfill the signcryption functions of both a secure encryption and a digital signature.

$$s_C = r_C^{-1} \cdot (H(m) + d_C \cdot x_{R_C}) \quad (7)$$

- Finally, C transmits the set of messages, (M, s_C, R_C, W_C, e_C, ca_C), through a secure channel to GM_A.

3.4 Authentication Phase

After completing the encryption process, C is able to effectively convey an encoded message to the group members p_i. The authentication procedure between C and GM_A is presented as below.

- Upon receiving the message set, GM_A uses his/her private key d_{GM_A}, the one-time pad point R_C and the CA's public key Y_{CA} to verify that the authenticity of C's identity, and that the hash embedded in the signature matches the encrypted message M by checking equations (8) and (9).

$$M - d_{GM_A} \cdot R_C = (m, H(m)) \quad \textbf{(8)}$$

$$I_C = ca_C(e_C \cdot G + x_C \cdot Y_{CA}) \quad \textbf{(9)}$$

- If $I_C = W_C$, GM_A is ensured that this message set is coming from C. To assert the authenticity of the message set that is tied to C, GM_A feeds the hash digest $H(m)$ and C's public key Y_C into the verification equation (10) to check the signature validity. If there is a signature point Γ on $E(F_q)$ when $x_{R_C} = x_\Gamma$ in the x-axis, GM_A confirms that this message is indeed originated with the signed sender C and is not altered along the way.

$$\Gamma = (H(m) \cdot s_C^{-1} \cdot G + s_C^{-1} \cdot x_{R_C} \cdot Y_C) = (x_\Gamma, y_\Gamma) \quad \textbf{(10)}$$

3.5 Participant Selection Phase

As each member in the group A finishes the registration with CA, GM_A is capable of building a work team of the participants to share a secret link through the following measures, and some of the group members p_i can accordingly accomplish the specific tasks that have been agreed upon.

- GM_A sets a weighted (wt, t, n)-threshold value for the group to establish a secret according to the two equations (11) and (12). Let $f(x)$ and $g(x)$ be polynomials over the field F_q, whose two leading terms a_0 and b_0 are the secret parameters, for all arguments x, where t is a non-negative integer, $a_0 = f(0)$, $b_0 = g(0)$, and $a_1,...,a_{t-1},b_1,...,b_{t-1} \in Z_q^*$.

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} (\text{mod } q) \quad \textbf{(11)}$$

$$g(x) = b_0 + b_1 x + b_2 x^2 + \ldots + b_{t-1} x^{t-1} (\text{mod } q) \quad \textbf{(12)}$$

- To generate a public/private key pair for the group A, GM_A adds the secret values of $f(0)$ and $g(0)$ together to get a composite with equation (13) as the private key d_A, and the corresponding public key Y_A is denoted by equation (14).

$$d_A = f(0) + g(0) \quad \textbf{(13)}$$

$$Y_A = d_A \cdot G = (f(0) + g(0)) \cdot G \quad \textbf{(14)}$$

- Each p_i of A independently picks a random number $k_i \in Z_q^*$ and converts this value to the elliptic-curve form by equation (15). All these K_i are sent to GM_A as arranged. In the meanwhile, GM_A chooses arbitrary integers $\tau_i \in Z_q^*$ for p_i and takes the input arguments (τ_i, K_i) to produce an individual hash value for each member using equation (16). Subsequently, the hash values σ_i as digital identities or credentials are sent back to the group members when they have been correctly computed. By evaluating the corresponding hash value, it is led to determine whether a particular member is legal, whereas GM_A can also intervene on a member's withdrawal in the group after modifying the identity attribute of p_i to specify the status-participation relationship. For example, GM_A simply resets $\sigma_i = H(1)$ and the credential state is currently unassigned to the member.

$$K_i = k_i \cdot G \quad \textbf{(15)}$$

$$\sigma_i = H(\tau_i \| K_i) \quad \textbf{(16)}$$

- Once selected members of the group are assembled, GM_A creates a respective public-private key pair (Y_i, d_i) for all participants, represented by (Y_i, d_i) in equations (17) and (18), according to the same way of calculating the group key pair (Y_A, d_A). In addition, GM_A makes public to the participants after the relevant parameters are well established, such as $E(F_q)$, G, $H()$, Y_A and Y_i.

$$d_i = f(x_i) + g(x_i) \quad \textbf{(17)}$$

$$Y_i = d_i \cdot G = (f(x_i) + g(x_i)) \cdot G \quad \textbf{(18)}$$

3.6 Weight Distribution Phase

For a given message, GM_A casually distributes a share of the secret among the selection of participants. Each participant $p_i \in A$ receives shares corresponding to different weight coefficients wt_1, wt_2, …, wt_n, and any portion of the threshold combined together such that $\sum_{p_i \in A} wt_i \geq wt$ can open the shared secret by fulfilling particular conditions. The calculation procedure is explained in the following steps.

- To start with, GM_A determines a concatenated data set $B = (\delta_1, \ldots, \delta_i, \ldots, \delta_n)$ with weight values, where

$\delta_1 = \max(\delta_i)$ and B is a vector space with an ordered basis such that the ordered n-tuple is a sequence of n positive integers. According to the DKP, the knapsack packages the specified items, i.e., weight values, to produce a variety of combinations. Also, GM_A picks two prime numbers w and γ at random, and gives γ in the range $\sum_{i=1}^{n} \delta_i < \gamma < 2\delta_i$. Then an encrypted set of weight values $\rho_1, \rho_2, \ldots, \rho_n$ is collected as an n-dimensional vector $D = (\rho_1, \rho_2, \ldots, \rho_n)$, and the corresponding components of the vector D, are chosen by equation (19).

$$\rho_i = w \times \delta_i \pmod{\gamma},\ i = 1, \ldots, n \tag{19}$$

- Then GM_A converts this positive integer ρ_i to a binary string and separates two asymmetric partitions from the decimal number in binary form, which contains the sequence e_i in the upper j bits and the sequence v_i in the lower $n–j$ bits as described by equation (20). Additionally, to avoid the carry propagation that converts overflows in its binary representation into an error return decimal value, e_i and v_i need to meet the condition of $(e_i + z \times v_i) < 2^j - 1$, where z is a non-negative integer.

$$j \gg n–j,\ \rho_i = e_i \times 2^{n-j} + v_i \tag{20}$$

- Next, GM_A sets up a confusion value u_i which makes the relationship between two parts consisting of the lower $n–j$ bits and the upper j bits as chaotic as necessary, i.e., the substitution-permutation technique, from equation (21).

$$u_i = e_i + z \times v_i,\ i = 1, \ldots, n \tag{21}$$

- Afterwards in order to apply the CRT, two positive integers, α and β, that are relatively prime, are chosen randomly such that $\alpha > \sum_{i=1}^{n} u_i$ and $\beta > \sum_{i=1}^{n} v_i$. The Chinese remainder theorem says every pair of congruence relations for an unknown integer θ_i as it is a weight-encoded valve in our case, where $0 \le \theta_i \le \alpha\beta–1$, $i = 1, \ldots, n$, of the form in equation (22), gives a unique solution with coprime moduli.

$$\theta_i = u_i \pmod{\alpha} = v_i \pmod{\beta},\ i = 1, \ldots, n \tag{22}$$

- In the end, GM_A distributes the secret vector $A = (\theta_1, \theta_2, \ldots, \theta_n)$ among the selection of participants, and each of them has a portion of the threshold based on different weights of share allocation.

3.7 Signature Production Phase

While accomplishing different weighting the allocation of share to the selected participants p_i, for instance, t subjects in this set A', GM_A prepares for giving a share of the secret to the players. Before doing so, there are specific conditions such as the verification of the respective player signs and the inspection of the total weights, which fulfill the given participants in the subgroup A'. The signature production process takes place in several steps and involves an individual signature and a group signature.

- Each player of the subgroup participants randomly chooses an integer k_i to convert it into an elliptic curve point (ω_i, φ_i) from equation (23). Also, each participant p_i in A' creates an individual signature (r_i, s_i) on m, and the signature is computed by a set of parameters $(m, \omega_i, d_i, c_i, k_i)$ where d_i is the private key of the respective member and c_i is the secret value of the individual player, through equations (24) to (26). The digital signature will be used to authenticate the identity of the player and the integrity of the message to be verified at a later time.

$$K_i = k_i \cdot G = (\omega_i, \varphi_i) \text{ for } i = 1, \ldots, t \tag{23}$$

$$r_i = H(H(m) \| \omega_i) \tag{24}$$

$$c_i = \frac{\prod_{j=1, j \ne i}^{t} -\omega_i}{\omega_i - \omega_j} \tag{25}$$

$$s_i = (r_i \cdot d_i \cdot c_i + k_i) = (r_i \cdot (f(x_i) + g(x_i)) \cdot c_i + k_i) \tag{26}$$

- All of the 3-tuples (r_i, s_i, θ_i) containing an individual signature and an encrypted weight thereof, are transmitted back to GM_A to check if the corresponding values are valid. Then, GM_A can use the signatory's public key Y_i to verify the validity of this signature if equation (27) holds true at the specific values.

$$K_i = s_i \cdot G - r_i \cdot c_i \cdot Y_i \tag{27}$$

- When the specific individual signatures of the subgroup participants have been fulfilled, GM_A generates a group signature (r, s) for the collaborating individuals. First, GM_A deciphers the weight-encoded vector $A' = (\theta_1, \theta_2, \ldots, \theta_t)$ assigned to each participant, to derive the sum of all the weights δ_i for the selected party A', and the total weight ε is obtained by equation (28). Next, GM_A compares the two weight values to determine if the sum of the weights of participants involved, ε, is greater than or equal to the fixed weight, wt. Then, if the condition is satisfied, GM_A uses equations (29) and (30) to compute the sum of the signature pairs (r_i, s_i) for the selection of t participants. Last, GM_A accomplishes a weighted threshold signature (r, s) on behalf of the subgroup A' on the message m.

$$\varepsilon = \sum_{i=1}^{t} \delta_i \tag{28}$$

$$r = \sum_{i=1}^{t} r_i \tag{29}$$

$$s = \sum_{i=1}^{t} s_i \tag{30}$$

3.8 Correctness of the Proposed Scheme

The correctness of the proposed scheme can be verified by examining if the form of equation is $K_i = s_i \cdot G - r_i \cdot c_i \cdot Y_i$, and determining if the relation of $\sum_{i=1}^{t} \delta_i \geq wt$ holds. While these statements and equations can stand true for some fixed values of, s_i, r_i, c_i, Y_i, δ_i, in order to define the weighted threshold signcryption algorithm as a general secret sharing scheme, we must prove the properties of equivalent expressions. Namely, GM_A with each signatory's public key Y_i can verify the correctness of an individual signature (r_i, s_i) and an encrypted weight value c_i thereof, to achieve the weighted threshold scheme. Getting to the proof we can formalize it as follows:

Theorem 1. Let (r_i, s_i) be a participant's signature of the group, and let c_i be the encrypted weight value of the participant. Then $K_i = s_i \cdot G - r_i \cdot c_i \cdot Y_i$ is correct.

Proof. To perform the theorem, we use a direct proof [29]. Because equations (18), (24), (25) and (26) are defined for particular values of the individual signature, we are able to write:

$$s_i \cdot G - r_i \cdot c_i \cdot Y_i = (r_i \cdot d_i \cdot c_i + k_i) \cdot G - H(H(m) \| \omega_i) \cdot \frac{\prod_{j=1, j\neq i}^{t} -\omega_i}{\omega_i - \omega_j} \cdot (f(x_i) + g(x_i)) \cdot G.$$

Expanding the form, we get:

$$k_i \cdot G + H(H(m) \| \omega_i) \cdot (f(x_i) + g(x_i)) \cdot \frac{\prod_{j=1, j\neq i}^{t} -\omega_i}{\omega_i - \omega_j} \cdot G - H(H(m) \| \omega_i) \cdot \frac{\prod_{j=1, j\neq i}^{t} -\omega_i}{\omega_i - \omega_j} \cdot (f(x_i) + g(x_i)) \cdot G.$$

Now, we can simplify the expression as $k_i \cdot G$, by eliminating a positive and a negative of the same term. By equation (15), we have $K_i = k_i \cdot G$. Thus, the signature (r_i, s_i) is verified correctly. This proves that the theorem stands in this case.

Theorem 2. Let wt be a globally fixed threshold value with the weights corresponding to each participant δ_1, δ_2, …, δ_t, and the global threshold of wt is defined by $\sum_{i=1}^{t} \delta_i \geq wt$. Then the sum of the weights of participants involved is greater than or equal to the globally fixed threshold.

Proof. As in the preceding section, GM_A deciphers the weight-encoded vector (θ_1, θ_2, …, θ_t) as a set of (δ_1, δ_2, …, δ_t), and the sum of associated weights δ_i involved has to satisfy $\sum_{i=1}^{t} \delta_i \geq wt$. We will prove by induction that [34], for all $n \in Z_q^*$, the following holds: $P(n) \rightarrow$ The sum of any set of n weights is greater than or equal to wt.

Base case. Since the sum of an 1-weight set ≥ wt, and the statement $P(n)$ is true for $n = 1$.

Induction step.

- Suppose $P(k)$ is true, i.e., that the sum of any k-weight set ≥ wt.
- We seek to show that $P(k + 1)$ is true as well, i.e., any $(k+1)$-weight set ≥ wt.
- Let A be a set of with $(k+1)$ weights.
- Let a be a weight of A, and let $A' = A - \{a\}$ (so that is A' set with k weights).
- The sets can be written as the form $A = A' \cup \{a\}$. Since A' has k weights, the induction hypothesis can be applied to this set and we get that the sum of the set of k weights is greater than or equal to wt. Hence the total number of weights of A is $(k+1)$.
- Since A is an arbitrary $(k + 1)$-weight set, we have proved that the sum of any $(k + 1)$-weight set ≥ wt. Thus $P(k+1)$ is true, completing the induction step.

Conclusion. By the principle of induction, $P(n)$ is true for all $n \in Z_q^*$.

Therefore, the sum of any set of n weights is greater than or equal to wt. This shows $\sum_{i=1}^{t} \delta_i \geq wt$ and completes the proof of the theorem.

4 Security Analysis of the Weighted Threshold Signcryption Scheme

The security of the proposed algorithm is based upon the difficulties of solving the ECDLP, the DKP, and the CRT; thus, it satisfies the essential security requirements such as confidentiality, authentication, anonymity, and unforegeability as formalized specifications from the existing threshold-related signature works [6, 10, 13, 17]. In addition, our scheme provides the threshold signature model with the critical properties of elasticity of weights assignment and anti-collusion attack, to increase security measures. We inspect the characteristics of the proposed solution in terms of the security goals and needs as follows.

4.1 Confidentiality

Confidentiality prevents unauthorized disclosure or use of sensitive digital information, ensuring that only those individuals or entities who have a legitimate access to the contents. In this study, all messages (such as the message m, the weight values δ_i or the signatures (r_i, s_i)) are encrypted by the addition operation of on a randomly chosen elliptic curve, and passed through a series of permutation processes. If any adversary intercepts the transmission of the enciphered message as (r_i, s_i, θ_i, c_i, Y_i, K_i, G), the interceptor is unable to decrypt the session data without knowing anything about implementations of the underlying ECDLP. For example, the point of K_i on $E(F_q)$, which depends

parametrically on k_i (an arbitrary integer) and G (a base point), can be hard to deal with by other means. Also, he/she cannot find the private key d_i from $Y_i = d_i \cdot G$ as given by equation (18), to obtain one of the signature values s_i. Accordingly, the signature method satisfies the confidentiality requirement.

4.2 Authentication

Authentication is the process of determining whether an individual or entity's identity is valid or not to the system. If an antagonist pretends to be C to manipulate the sensitive data, he/she needs to forge the message (R_C, s_C) from equations (5) and (7), to masquerade as C's identity. When the malicious user intends to identify the critical parameters obtained, this leads to some intractable problems involving the relevant encrypted parameters, of a random integer r_C, of C's private key d_C or the x-coordinate of a point on $E(F_q)$ by exploiting released public data. Besides the ECDLP, if the antagonist impersonates C's identity that sends the fake message to GM_A that purports to come from C, but instead includes the antagonist's public key, GM_A can then detect whether the personator he/she is talking to is genuine by requesting the mutual authentication, i.e., by checking against $I_C = W_C$, as described for equation (9). Hence, the proposed model ensures the authentication property.

4.3 Anonymity

Anonymity means that an agent who performs a certain action is not letting the details of that action to a set of all possible subjects. In this design, all of the legal participants create their own digital signatures (r_i, s_i) through rigorous mechanisms, and the members are unable to derive other members' signatures from their message digests (e.g., r_i) along with the private keys d_i except for the participant himself/herself and GM_A. To put it another way, the identity of the participants in the group remains anonymous to each other throughout. Therefore, the signature solution offers the anonymity as one of the security features.

4.4 Unforgeability

Unforgeability refers that no one is able to produce a valid digital signature on any arbitrary message other than the legitimate signer. In the proposed approach, if an opponent *Opp* can forge a digital signature, he/she has to possess the knowledge to generate a message-signature tuple of (M, s_{Opp}, R_{Opp} W_{Opp}, e_{Opp}, ca_{Opp}) such that anyone could verify that the message indeed originated from the opponent *Opp*. That is, each member of A can verify that the signature s_{Opp} by checking to see if they match the relative reference $\Gamma = (H(m) \cdot s_{Opp}^{-1} \cdot G + s_{Opp}^{-1} \cdot x_{R_{Opp}} \cdot Y_{Opp} = (x_\Gamma, y_\Gamma)$ in equation (10) both belongs to the illegitimate user and is invalid for the associated message. Likewise, GM_A is always able to distinguish the forged signature from a properly generated signature and to determine the real identity of the signatory. Thus, the signature algorithm fulfills the characteristic of unforgeability.

4.5 Threshold Characteristic

Threshold characteristic indicates that a threshold method is used to distribute a secret d among n members such that any party of size k or more can construct the secret d but smaller parties cannot. In the proposed suggestion, each player in A is given a positive integer weighted state δ_i which is associated with distinct encoding patterns θ_i, and the shared secret can be reconstructed when the sum of the weights of the players involved meets an established threshold wt. Furthermore, GM_A is able to elastically assign or eliminate the units the weight to group members through the specific security governance as the DKP and CRT mechanisms. In such a way, the elasticity feature of this model enables the leader of the group to identify and apply the appropriate strategy to dynamically adjust the security responsibilities as the individual members' needs, in order to reduce the biases arising from group decision-making such as high group cohesiveness, insulation from outside experts, and flawed procedures for handling tasks, as well as to satisfy their performance goals.

4.6 Anti-collusion Attack

A collision attack denotes that a number of users may collude with one another in order to obtain the access permission beyond their privileges or perform the intended action against the referenced object. In the work, the participants bear different weights with unequal privileges for the threshold secret sharing, and the weights and the shared secret are processed by adequate dual-protection technologies. One is applying the combination of two functions $f(x)$ and $g(x)$ with their respective domains, which give quite different results even they have the same rule, to evaluate the secret parameters (e.g., the leading terms a_0 and b_0). The other is adopting the interpolation polynomial in the Lagrange form as given by equation (25), which admits a unique solution from any designated points, to recognize the secret value c_i associated to the participant's representation (e.g., ω_i). If some participants try to collude together in order to increase their weights by sharing their secrets like up to more than a coalition of size t, there is no way to measure the leading coefficients $f(0)$ and $g(0)$ at the same time, to say nothing of deriving the parameter c_i in the structure. Consequently, the proposed program has the anti-collusion ability to prevent illegal coalitions within the group from this kind of collusion attacks, where multiple members collude by recovering the threshold group signature (r, s) of their individual acquired weights.

If we reinvestigate the similarity models of the

existing weighted threshold signature methods, it is evident that the current method provides for effective countermeasures in security considerations. Conversely, Iftene and Grindei's algorithm [17] combines the generalized Mignotte sequences whose modules are not necessarily pairwise coprime to choose the weighted threshold elements with the RSA cryptosystem, and its security of the signature verification for the group members cannot be guaranteed. Likewise, Guo and Chang's approach [13] exploits the cryptographic techniques of extended Asmuth-Bloom sequences based on the RSA cryptosystem without applying anonymous mechanism onto the signatures, and might cause the participants in the group to be subject to the conspiracy issues. Table 2 presents a comparison between our approach and the two existing proposals along with the corresponding weighted threshold signature techniques. Symbol "√" indicates that the algorithm satisfies the security feature, and symbol " Δ " refers that the model partially supports the security capability.

Table 2. Comparative analysis of different existing weighted threshold signature algorithms based on their security properties

Algorithms Security attributes	Iftene & Grindei's scheme [17]	Guo & Chang's scheme [13]	The proposed scheme
Confidentiality	Δ	√	√
Authentication	√	√	√
Anonymity	Δ	Δ	√
Unforgeability	Δ	√	√
Threshold characteristic	√	√	√
Anti-collusion attack	Δ	√	√

5 Performance Evaluation of the Proposed Model

Having described the effectiveness of the proposed countermeasures in security requirements, we next evaluate the performance of the proposed algorithm in terms of the execution time, and show that it brings great efficiency compared with the existing works with respect to the application of weighted threshold signature methods. We examine the theoretical framework of these various solutions for solving the techniques of cryptology related to computation costs incurred by each task in accordance with the concept of modular arithmetic operations [8, 36-37]. The notation of modular multiplications has been widely used in many public-key cryptosystems for evaluating the complexity in terms of time and resources needed, and the main operations shown in Table 3 include scalar (or point) multiplication, point addition, modular exponentiation, and modular inversion.

Table 3. The modular mathematical notation

Symbol	Description	Operation cost
T_{ECMUL}	the time for the multiplicative operation of an elliptic curve point	$\approx 29T_{MUL}$
T_{ECADD}	the time for the addition operation of two points on an elliptic curve	$\approx 5T_{MUL}$
T_{ECh}	the time for the hash operation of an elliptic curve point	$\approx 23T_{MUL}$
T_{INVS}	the time for the modular multiplicative inverse operation	$\approx 240T_{MUL}$
T_{EXP}	the time for the modular exponential operation	$\approx 240T_{MUL}$
T_{ADD}	the time for the modular addition operation	The time complexity for T_{ADD} is negligible
T_h	the time for the conventional hash operation	The time complexity for T_h is negligible

Note. Modular multiplication is a fundamental operation in many popular public-key cryptosystems. It converts various operations units to the time complexity in terms of T_{MUL}.

Although the above mentioned two techniques have not the exactly same steps as the new proposed weighted threshold signcryption approach, we still establish the baseline whenever possible to compare the outcomes of different stages on the same measure. Table 4 summarizes the computational costs of each step involved in these weighted threshold signature models. Compared to other two related algorithms for performing cryptographic operations along with the signature or encryption function, we observe that the proposed scheme takes much less T_{MUL} time for the parameter generation, signature (encryption) generation and verification, and weighted threshold signature (encryption) stages where in our case they are - initial and registration - encryption and authentication - participant selection, weight distribution, and signature production phases. For example, our method consumes $(2t+255)T_{MUL}$ time in handling the weighted threshold signature (encryption) generation and verification process, whereas the two approaches spend significantly more time for this purpose as $(244t+243)T_{MUL}$ and $(t+2168)T_{MUL}$ time respectively. The modulus and exponent operations increase the computation time per task in RSA-based cryptosystems, since they are more expensive to deal with factoring large prime numbers to create the encrypted message. Unlike the two existing works depending on the RSA standards, the proposed ECC-based solution takes tremendously low computation cost for both the encryption and authentication phases.

Table 4. Performance comparison between the proposed scheme and the existing algorithms

Stage \ Cost \ Method		Iftene & Grindei's scheme [17]		Guo & Chang's scheme [13]		The proposed scheme	
		Computational cost	Rough estimation	Computational cost	Rough estimation	Computational cost	Rough estimation
Parameter Generation	Initial / Registration	$1T_{MUL}+1T_{EXP}$	241 T_{MUL}	$1T_{EXP}+(3t+2)T_{MUL}+2tT_{ADD}$	$(3t+242)$ T_{MUL}	$4T_{ECMUL}+2T_{MUL}+1T_{ADD}+1T_{INVS}+1T_{ECh}$	381 T_{MUL}
Individual Signature (Encryption) Generation and Verification	Encryption / Authentication	$1T_{MUL}+3T_{EXP}$	721 T_{MUL}	$7T_{EXP}+2T_{INVS}+(2n+t+2)T_{MUL}+nT_{ADD}+1T_h$	$(2n+t+2162)$ T_{MUL}	$8T_{ECMUL}+2T_{MUL}+4T_{ECADD}+1T_{ADD}+3T_{INVS}+2T_{ECh}$	1020 T_{MUL}
Weighted Threshold Signature (Encryption) Generation and Verification	Participant Selection / Weight Distribution / Signature Production	$(t+2)T_{EXP}+(4t+2)T_{MUL}+1T_{ADD}$	$(244t+243)$ T_{MUL}	$8T_{EXP}+1T_{INVS}+(t+8)T_{MUL}$	$(t+2168)$ T_{MUL}	$6T_{ECMUL}+2tT_{MUL}+1T_{ECADD}+(2n+3t)T_{ADD}+2T_{ECh}$	$(2t+225)$ T_{MUL}

Annex 1: Iftene and Grindei only propose an elegant encryption-decryption-signature function for the CRT-based weighted threshold secret sharing method, while Guo and Chang apply a signature idea to the GCRT-based weighted threshold secret sharing solution. The two studies mainly introduce the application of RSA cryptosystems, and we first give a signcryption function for the weighted threshold secret sharing scheme based on the ECC algorithm.

Annex 2: In a (t, n)-threshold secret sharing scheme, the parameter n refers to all members in a group and t is the participants in the group.

6 Conclusion

This paper presents a new group-oriented weighted threshold signcryption scheme based on the three hard problems — the elliptic curve discrete logarithm problem (ECDLP), the dynamic knapsack problem (DKP) and the Chinese remainder theorem (CRT). To improve the security of digital messages signed on behalf of the group members, the signcryption technique is properly incorporated into the weighted threshold protocol. Simultaneously the ECC-based cryptosystem makes the weighted threshold secret sharing process more efficient in the underlying field operations. Apart from the two primary benefits of security and efficiency improvements, the rigorous authentication process embedded in the proposed scheme can effectively prevent potential group members from the colluding agreements.

We give the correctness proof of the proposed algorithm, that the group signature is indeed created by the sum of the weights of participants involved and the total is greater than or equal to a globally fixed threshold value, as well as the verification of the group signature. Through the security analysis, the study satisfies the security requirements for a weighted threshold secret sharing cryptosystem. In addition, we have evaluated the time complexity to demonstrate the efficiency gain, and the results show that the model is able to achieve lower consumption with less computational cost and communication overhead when compared to two other existing algorithms.

To the best of our knowledge, the mechanism described in this paper is the first weighted threshold signcryption using ECC-based cryptographic primitives. Providing efficient and secure solutions in such a malicious cyber activity environment requires the abilities of the possession special features accordingly. We are convinced that the current scheme provides significant ameliorations with the characteristics for the application of weighted threshold secret sharing cryptosystems. Threshold-related cryptosystems are still worth exploring new developments in various research topics. In our future work, for example, we will try to construct a non-centralized model, which will appropriately facilitate the distribution of weights between the leader and the individual members while generating a global threshold for the group. We will also devote more attention to processing multiple digital messages received from various group situations in the area of threshold cryptography.

References

[1] C. Asmuth, J. Bloom, A Modular Approach to Key

Safeguarding, *IEEE Transactions on Information Theory*, Vol. 29, No. 2, pp. 208-210, March, 1983.

[2] E. Barker, *Recommendation for Key Management — part 1: General*, Special Publication (NIST SP)-800-57, Revision 4, January, 2016.

[3] A. Beimel, Secret-sharing Schemes: A Survey, *Coding and Cryptology, Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, Vol. 6639, 2011, pp. 11-46.

[4] G. R. Blakley, Safeguarding Cryptographic Keys, *Proceedings of the 1979 AFIPS National Computer Conference*, Monval, NJ, USA, Vol. 48, 1979, pp. 313-317.

[5] I. N. Bozkurt, K. Kaya, A. A. Selcuk, A. M. Guloglu, Threshold Cryptography Based on Blakley Secret Sharing, *Proceedings of Information Security and Cryptology Conference with International Participation — ISCTURKEY 2008*, Ankara, Turkey, 2008, pp. 183-186.

[6] T. Y. Chang, C. C. Yang, M. S. Hwang, Threshold Untraceable Signature for Group Communications, *IEE Proceedings — Communications*, IEEE, Vol. 151, No. 2, pp. 179-184, April, 2004.

[7] B. Chor, R. L. Rivest, A Knapsack-type Public Key Cryptosystem Based on Arithmetic in Finite Field, *IEEE Transactions on Information Theory*, Vol. 34, No. 5, pp. 901-909, September, 1988.

[8] J. P. David, K. Kalach, N. Tittley, Hardware Complexity of Modular Multiplication and Exponentiation, *IEEE Transactions on Computers*, Vol. 56, No. 10, pp. 1308-1319, 2007.

[9] E. Dawson, D. Donovan, The Breadth of Shamir's Secret-Sharing Scheme, *Computers & Security*, Vol. 13, No. 1, pp. 69-78, February, 1994.

[10] Y. Desmedt, Y. Frankel, Shared Generation of Authenticators and Signatures, *Advances in Cryptology — CRYPTO '91, Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, Vol. 576, 1991, pp. 457-469.

[11] P. Dikshit, K. Singh, Weighted Threshold ECDSA for Securing Bitcoin Wallet, *ACCENTS Transactions on Information Security*, Vol. 2, No. 6, pp. 43-51, April, 2017.

[12] C. C. Dragan, F. L. Tiplea, Distributive Weighted Threshold Secret Sharing Schemes, *Information Sciences*, Vol. 339, pp. 85-97, April, 2016.

[13] C. Guo, C. C. Chang, Proactive Weighted Threshold Signature Based on Generalized Chinese Remainder Theorem, *Journal of Electronic Science and Technology*, Vol. 10, No. 3, pp. 250-255, September, 2012.

[14] X. Hei, X. Du, B. Song, Two Matrices for Blakley's Secret Sharing Scheme, *Proceedings of the 2012 IEEE International Conference on Communications (ICC)*, Ottawa, ON, Canada, 2012, pp. 810-814.

[15] C. L. Hsu, T. S. Wu, T. C. Wu, Improvements of Generalization of Threshold Signature and Authenticated Encryption for Group Communications, *Information Processing Letters*, Vol. 81, No. 1, pp. 41-45, January, 2002.

[16] S. Iftene, General Secret Sharing Based on the Chinese Remainder Theorem with Applications in e-Voting, *Electronic Notes in Theoretical Computer Science*, Vol. 186, pp. 67-84, July, 2007.

[17] S. Iftene, M. Grindei, Weighted Threshold RSA Based on the Chinese Remainder Theorem, *Proceedings of the 9th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, Timisoara, Romania, 2007, pp. 175-181.

[18] K. Kaya, A. A. Selcuk, Threshold Cryptography Based on Asmuth-Bloom Secret Sharing, *Information Sciences*, Vol. 177, No. 19, pp. 4148-4160, October, 2007.

[19] B. L. Kirkman, Why Teams Often Make Riskier Decisions than Individuals (and What You Can Do about It), *Enterprise Risk Management Initiative*, Raleigh, NC, USA, 2017.

[20] N. Koblitz, Elliptic Curve Cryptosystems, *Mathematics of Computation*, Vol. 48, No. 177, pp. 203-209, January, 1987.

[21] N. Koblitz, A. Menezes, S. Vanstone, The State of Elliptic Curve Cryptography, *Designs, Codes and Cryptography*, Vol. 19, No. 2-3, pp. 173-193, March, 2000.

[22] J. Kurihara, S. Kiyomoto, K. Fukushima, T. Tanaka, A New (k, n)-Threshold Secret Sharing Scheme and Its Extension, *Information Security, Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, Vol. 5222, 2008, pp. 455-470.

[23] H. Lein, F. Miao, C. C. Chang, Verifiable Secret Sharing Based on the Chinese Remainder Theorem, *Security and Communication Networks*, Vol. 7, No. 6, pp. 950-957, June, 2014.

[24] M. Li, S. Ma, C. Guo, A Novel Weighted Threshold Secret Image Sharing Scheme, *Security and Communication Network*, Vol. 8, No. 17, pp. 3083-3097, November, 2015.

[25] D. Mahto, D. K. Yadav, RSA and ECC: A Comparative Analysis, *International Journal of Applied Engineering Research*, Vol. 12, No. 19, pp. 9053-9061, October, 2017.

[26] G. Mante, S. D. Joshi, Discrete Logarithm Based (t, n) Threshold Group Signature Scheme, *International Journal of Computer Applications*, Vol. 21, No. 2, pp. 23-27, May, 2011.

[27] R. J. McEliece, D. V. Sarwate, On Sharing Secrets and reed-Solomon Codes, *Communications of the ACM*, Vol. 24, No. 9, pp. 583-584, September, 1981.

[28] M. Mignotte, How to Share a Secret, *EUROCRYPT 1982, Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, Vol. 149, 1982, pp. 371-375.

[29] V. Mihova, J. Ninova, Direct and Indirect Methods of Proof, The Lehmus-Steiner Theorem, https://arxiv.org/pdf/1410.7526.pdf.

[30] V. S. Miller, Use of Elliptic Curves in Cryptography, *Advances in Cryptology — CRYPTO '85, Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, Vol. 218, 1986, pp. 417-426.

[31] P. Morillo, C. Padro, G. Saez, J. L. Villar, Weighted Threshold Secret Sharing Schemes, *Information Processing Letters*, Vol. 70, No. 5, pp. 211-216, June, 1999.

[32] S. Nigam, K. N. Hande, Survey on "Security Architecture Based on ECC (Elliptic Curve Cryptography) in Network", *International Journal of Computer Science and Mobile Applications*, Vol. 3, No. 1, pp. 17-23, January, 2015.

[33] A. Shamir, How to Share a Secret, *Communications of the ACM*, Vol. 22, No. 11, pp. 612-613, November, 1979.

[34] A. Stefanowicz, Proofs and Mathematical Reasoning, https://

www.birmingham.ac.uk/Documents/college-eps/college/stem/Student-Summer-Education-Internships/Proof-and-Reasoning.pdf.

[35] P. C. Su, C. H. Yang, J. H. Zeng, C. W. Yeh, C. S. Kao, A Design of Weighted Threshold Mechanism for Proxy Blind Signature, *Proceedings of the 2017 Annual Conference on National Defense Management Academic and Symposium*, Taipei, Taiwan, 2017, Vol. 1, pp. 158-174.

[36] N. Tahat, E. E. Abdallah, A New Signing Algorithm Based on Elliptic Curve Discrete Logarithms and Quadratic Residue Problems, *Italian Journal of Pure and Applied Mathematics*, Vol. 32, pp. 125-132, August, 2014.

[37] C. H. Tsai, P. C. Su, Multi-document Threshold Signcryption Scheme, *Security and Communication Network*, Vol. 8, No. 13, pp. 2244-2256, October, 2015.

[38] Y. Zheng, Digital Signcryption or How to Achieve Cost (signature & encryption) <<cost (signature) + cost (encryption), *Advances in Cryptology — CRYPTO'97, Lecture Notes in Computer Science*, Vol. 1294, 1997, pp. 165-179.

Biography

Chien-Hua Tsai is currently an Associate Professor in the Department of Accounting Information at Chihlee University of Technology, Taiwan. He received his Ph.D. degree in Electrical Engineering and Computer Science from Case Western Reserve University, Ohio, USA in 2000. His research interests include Information System Security, Secure Communication Protocols, Public Key Cryptosystems and Electronic Transaction Security.