New Security Concurrent Signature Design

Shu-Hwang Liaw[1], Erl-Huei Lu[1], Henry Ker-Chang Chang[2], Pin-Chang Su[3]

[1] Department of Graduate Institute of Electrical Engineering, Chang Gung University, Taiwan
[2] Department of Graduate Institute of Information Management, Chang Gung University, Taiwan
[3] Department of Information Management, National Defense University, Taiwan
pcspace@gmail.com, {lueh, changher}@mail.cgu.edu.tw, spc.cg@msa.hinet.net

Abstract

The rapid growth of the Internet has accelerated e-commerce application development. Numerous problems involving Internet trade transaction security remain. Therefore, security and transaction fairness establishment in e-commerce applications is a crucial research topic. A fair exchange protocol enables users to securely obtain information from each other. This has become a widely applied research topic in e-commerce. A concurrent signature eliminates the need for a trusted third party (TTP) or substantial mutual communication between two parties, to provide fair exchange protocol with e-commerce ideas. The concurrent signature concept was proposed by Chen et al. in 2004 EUROCRYPT. Some researchers improved the concurrent signature susceptibility to the message substitution attack defect, but there are still identify authentication and information exchange procedure security issues. This paper proposes a concurrent signature scheme based on elliptic curve cryptography (ECC) that uses the same security strength shorter key length, reduces public storage costs and the calculation and management risks. We strengthen the identity authentication mechanisms to prevent the counterfeiting identity attacks and provide a more confidential protocol through a self-certified mechanism.

Keywords: E-Commerce, Fair exchange, Concurrent signature, Elliptic curve cryptography, Self-certified

1 Introduction

Internet technology developments have promoted the rapid growth of e-commerce applications. Transactions through the Internet include the exchange of messages, e-commerce payment agreements, the signing of contracts and electronic e-mail. However, the Internet is a transparent and open environment. It may be attacked by stealing, forging, tampering and counterfeiting identity. Therefore, creating a secure and reliable e-commerce trading environment is very important. A safe and reliable environment means that messages cannot be forged, illegally accessed, tampered with, stolen and eavesdropped upon during transmission online. The other party to the transaction may not be denied proven transaction records. A safe and reliable environment must rely on the digital signature to achieve this purpose.

Fair exchange of digital signature agreements was the subject of a wide-ranging study on e-commerce applications. Bao et al. [1] proposed the use of an off-line trusted third party (TTP) fair trading system. Asokan et al. [2] proposed the signature optimistic fair exchange of digital signatures concept. A large number of fair trade system studies [3-6] in e-commerce applications on have been proposed. Chen et al. [7] insightfully observed that a fair exchange signature scheme is not required for numerous applications because they found a mechanism that enables more natural conflict resolution without the participation of a TTP, namely, concurrent signatures. To enhance the anonymity of the concurrent signature scheme proposed by Chen et al. [7], Susilo et al. [8] proposed perfect concurrent signatures (PCS). Like Wang et al. study [9] on the PCS protocol, because in the initial stages the signer independently generates two keystones, they can freely combine various ambiguous signatures, rather than the originally determined maching signer. Because the original signer problem generates concerns regarding unfairness to the matching signer, perfect cocurrent signature improvements were suggested (iPCS). The difference between PCS and iPCS is that the initial signer and matching signer can establish a single keystone under the iPCS protocol. In the PCS protocol only the initial signer can generate a keystone. Thereafter, Chow and Susilo [10] proposed a PCS protocol based on identity authentication. Asymmetrical signatures [11], three-party concurrent signatures [12], and multi-party concurrent signatures [13] have all been proposed. Previous studies [7-9, 11-12] found a weakness in signature protocols that render them vulnerable to the following attack: Either party can create multiple ambiguous signatures containing differing messages that can also be bound by the same keystone. However,

*Corresponding Author: Shu-Hwang Liaw; E-mail: pcspace@gmail.com
DOI: 10.3966/160792642018051903011

these imposter messages are not the initial signature message that should have been transmitted. Thus, this attack is known as a message substitute attack. The accountability security characteristic was suggested in the literature based on these observations [14]. The accountability characteristics refers to the ability of any third party to confirm the accuracy of the signature through a VERIFY algorithm for the concurrent signature after the keystone is announced, thereby determining the uniqueness of the ambiguous signature. Because the improvements proposed in Li et al. [14] and i2PCS1 [15] did not achieve security on demand, Wang et al. [16] recommended including messages that Alice and Bob were to exchange in the keystone fix to achieve accountability. However, researchers found that this method involved transmitting messages in plaintext format. Both parties know the signature message from the other party before communication generates the risk of message content theft by malicious third parties and is inconsistent with actual e-commerce procedures.

To prevent the above concurrent signature message theft risk, this study uses elliptic curve cryptography (ECC) to hide the contents. This study enhances the identity authentication mechanism and prevents the counterfeiting identity attack by introducing a self-certified scheme. The proposed scheme introduces an elliptic curve digital signature algorithm (ECDSA) into a fully accountable concurrent signature scheme to achieve the level of security advocated in previous literature. Because the elliptic curve digital signature algorithm has a relatively short key length the attacker will face the elliptic curve discrete logarithm problem (ECDLP). Part of the contents of this article were published in conference [17]. For algorithm safety and feasibility validation a more complete discussion is presented here. In Section 2 we review ECC, the self-certified scheme, and concurrent signature. The improved scheme is proposed in Section 3. with an example implemented in Section 4. The security and benefits analyses are conducted in Section 5. Section 6 provides the research conclusion.

2 Literature Review

In this section, we firstly explain the concept of concurrent signature algorithms, secondly review ECC. At last, we descript self-certified scheme.

2.1 Concurrent Signature

Concurrent signature schemes were proposed by Chen et al. [7] which allow both signing parties to produce and exchange ambiguous signatures, with third parties not learning the identity of the original signer until an additional keystone is announced by one of the two parties. Subsequently, the third party can use this information to verify the identity of the ambiguous signature signer.

The concurrent signature scheme is composed of SETUP, ASIGN, AVERIFY and VERIFY algorithms, as described below.

(1) SETUP: Input a security parameter l and randomly produce two large prime numbers, p and q, with $q|(p-1)$, and order as a generator g of q, where $g \in Z_p$. Next, select a hash function $h:(0,1)^* \to Z_p$. Then establish a message space M, keystone space K and keystone fix space F. Here $M = K = \{0,1\}^*$ and $F = Z_q^*$. In addition, $(x_A, y_A = g^{x_A} \bmod p)$ and $(x_B, y_B = g^{x_B} \bmod p)$ are assumed to be Alice and Bob's private and public keys.

(2) ASIGN: Output an ambiguous signature $\sigma = (c, s', s)$ by inputting (y_A, y_B, x_A, s, m) where y_A and y_B are Alice and Bob's public keys and $y_A \neq y_B$ and x_A are Alice's private key. $s \in F$, and $m \in M$ are the signed messages. This algorithm performs the following parameter calculations:

(a) Select a random number $\alpha \in Z_q$.

(b) Calculate $c = h(m, g^{\alpha} y_B^s \bmod q)$.

(c) Calculate $s' = (\alpha - c) \cdot x_A^{-1} \bmod q)$.

(d) Output an anonymous ambiguous signature $\sigma = (c, s', s)$.

(3) AVERIFY: Input (σ, y_A, y_B, m), and verify $c = h(m, g^c, y_A^{s'}, y_B^s)$. If true, then output "accept"; if not, output "reject."

(4) VERIFY: This algorithm inputs (k, S), $k \in K$, $S = (\sigma, y_A, y_B, m)$. The algorithm VERIFY outputs accept if AVERIFY(S) = accept and the keystone k is valid by running a keystone verification algorithm. Otherwise, VERIFY outputs reject.

2.2 Elliptic Curve Cryptography

Miller [18] and Koblitz [19] first suggested using elliptic curves to implement public key crypto systems. A general elliptic curve is of the form, $y^2 + axy + by = x^3 + cx^2 + dx + e$, where a, b, c, d and e are real numbers. A special addition operation is defined over elliptic curves and this can be described algebraically as well as geometrically inclusive of a point ∞ called "point at infinity". If three points (i.e., p, q, and a unique third point) are on a line that intersects an elliptic curve, then the sum equals the point at infinity (∞). If the field K whose characteristic of q is neither two nor three (e.g., $K = F_q$ where q is greater than 3 and a prime), then an elliptic group over the Galois field $E(F_q)$ can be obtained by computing $y^2 = x^3 + ax + b \bmod q$ for $0 \le x \le q$.

The constants for a and b are non-negative integers that are less than the prime number q and satisfy the

condition, i.e., $4a^3 + 27b^2 \bmod q \neq 0$ Let the points $A=(x_1, y_1)$ and $B=(x_2, y_2)$ be in the elliptic group $E(F_q)$. The rules for addition over the elliptic group $E(F_q)$ are:

(1) $P + \infty = \infty + P = P$

(2) If $x_2 = x_1$ and $y_2 = -y_1$, that is $P=(x_1, y_1)$ and $Q=(x_2, y_2)=(x_1, -y_1)=-P$, then $P+Q=\infty$

(3) If $Q \neq P$, then the sum $P+Q=(x_3, y_3)$ is given by:

$x_3 = \lambda^2 - x_1 - x_2 \bmod q$, $y_3 = \lambda(x_1 - x_3)\text{-}y_1 \bmod q$

where $\lambda = (x_2 - y_1)/(x_2, x_1)$ If $x_1 \neq x_2$

or $\lambda = (3x_1^2 + a)/2y_1$ If $x_1 = x_2, y_1 \neq 0$.

To introduce a group operation on the curve with the following properties: we double a point *P*, and it is equivalent to *P* + *P*. We can similarly calculate 3*P* = 2*P* + *P*, and so on. One important property is that it is very difficult to find an integer s in such an equation ***sP*** = *Q*. The merit of ECC is that compared with RSA crypto systems they can provide the same security level with a shorter key length. Because of this mathematical property, ECC is faster and requires less hardware than RSA [20-21].

2.3 Self-Certified

A sophisticated approach first introduced by Girault [22], is called the self-certified public key (SCPK), which can be regarded as an intermediate between the identity-based approaches and the traditional PKI approaches. Using a RSA crypto system a user chooses his or her private key, computes the corresponding public key and gives it to a certificate authority. The certificate authority then computes certificate parameters for the user, which satisfy a computationally unforgeable relationship with the public key and the user's identity. A verifier can compute the public key from the identity and the certificate parameters. In 1997, Saeednia [23] successfully combined those merits with the inherency in both the ID-based and the self-certified systems. However, Wu et al. [24] showed that the original version of Saeednia's scheme is not secure enough against withstanding the impersonation attack. Subsequently, Saeednia [25] further proved that it is possible to make the attack ineffective by taking additional precautions. The latter resulting model presents great loss in the merits compared to the original model and no longer has to meet the primary self-certified notion contribution. Tsaur [26] extended Girault's works to ECC-based crypto systems which are quite suitable for electronic transactions. A main problem in SCPK schemes is that they only provide implicit authentication, i.e., the validity of a SCPK is verified only after a successful communication.

In this section we first present a formal definition for self-certified signature (SCS) schemes. The two main entities involved in the SCS scheme are a certificate authority and a client. We then propose a concrete SCS scheme from pairings. The SCS scheme consists of four randomized algorithms [27]: KEYGENPARAM, EXTRACT, SIGN, and VERIFY. The details are as follows.

KEYGENPARAM. The certificate authority CA chooses a master-key s and computes the corresponding public key P_{CA}. Each client U_A chooses a partial private key S_A and computes the corresponding partial public key Y_A. The actual public key of the user consists of the CA public key, partial public key and identity of the user together with the system parameters.

EXTRACT. CA runs the extract algorithm, which takes as input the system parameters, the master-key s, the partial public key Y_A and an arbitrary $ID_A \in \{0,1\}^*$, the infinite set of all binary strings, and returns the partial private key d_A. The CA sends d_A securely to the client with (P_{CA}, ID_A, Y_A) over a public channel. The actual private key of the client is (S_A, d_A), the actual public key is (P_{CA}, ID_A, Y_A).

SIGN. A client with his actual private key (S_A, d_A) uses the sign algorithm to compute signature σ for any message m.

VERIFY. Any verifier can validate the signature σ by checking the verification equation with respect to the actual public key (P_{CA}, ID_A, Y_A).

These algorithms must satisfy the standard consistency constraint, namely when (S_A, d_A) is the actual private key generated by the algorithm Extract when it is given the actual public key (P_{CA}, ID_A, Y_A), then $\forall m \in (0,1)^* : Verify((P_{CA}, ID_A, Y_A), m, \sigma) = Valid$ where $\sigma = Sign((P_{CA}, ID_A, Y_A), (S_A, d_A), m)$.

3 Concurrent Signature Method Design

This section will present the self-certified mechanism based on the ECDSA concurrent signature algorithm. Compared to the original concurrent signature algorithm the proposed method is more secure and fast. Table 1 presents the symbol definitions used in this algorithm.

In case Alice and Bob are performing an electronic transaction and exchange signatures the signature messages are the content of the transaction, or m_A, m_B.

Table 1. Definitions of notations

Item	Notation	Description
1	$E(F_p)$	An elliptical curve within a finite field F_p.
2	G	Base point of the elliptical curve.
3	n	Order of the base point of the elliptical curve.
4	p	A prime number: $p > 2^{224}$.
5	m_A, m_B	Message of Alice and Bob.
6	PK_A, PK_B	Public key of Alice and Bob.
7	id_A	Identification number of Alice.
8	M_A	Mainfest of Alice.
9	T_A	Signature of id_A.
10	P_A, P_B	Public keys of Alice and Bob.
11	Q_A, Q_B, Q_{KGC}	Private key of Alice, Bob, and KGC.
12	$h()$	Hash function (value transposition).
13	$f_1()$	Function which transforms a message into points on an elliptical curve.
14	$f_2()$	Function which transforms points on an elliptical curve into a message.

3.1 Improved Procedure

Setup phase. The key generation center (KGC) defines the system parameters used to create self-certified public keys for the registering users. The KGC selects the parameters for the elliptic curve domains and these specific items are defined geometrically with the underlying fields. The order *n* is a prime used for the elements of F_p. KGC chooses a secure Elliptic Curve, where *p* is a prime more than 224-bits in length; *n*: a order of *G*, make $n \cdot G = O$, *O* is the infinity point of the elliptic curve, $G \in E(F_p)$; *h*() KGC also chooses a one-way hash function.

The KGC selects a secure value $Q_{KGC} \in [1.n-1]$, compute

$$PK_{KGC} = Q_{KGC} \cdot G \quad \textbf{(1)}$$

The KGC keeps the Q_{KGC} in secret, then publishes the E, G, p, PK_{KGC}, $h()$.

Register phase.

(1) Alice id_A selects a secure value $r_A \in [2, n-2]$, computes

$$T_A = h(r_A \| id_A) \cdot G \quad \textbf{(2)}$$

Alice keeps the r_A secret and then sends the id_A and T_A to KGC.

(2) KGC selects a secure value $l_A \in [2, n-2]$ and calculates a public key PK_A and its manifests M_A for Alice through the following equations

$$PK_A = T_A + [l_A - h(id_A)] \cdot G = (q_{Ax}, q_{Ay}) \quad \textbf{(3)}$$

$$M_A = l_A + Q_{KGC}[q_{Ax} + h(id_A)] \quad \textbf{(4)}$$

KGC sends the PK_A and M_A to Alice.

Alice then derives a secret key Q_A as

$$Q_A = M_A + h(r_A \| id_A) \quad \textbf{(5)}$$

Also verifies the authenticity of PK_A by testing if

$$P_A = Q_A \cdot G = [l_A + Q_{KGC}(q_{Ax} + h(id_A)) + h(r_A \| id_A)] \cdot G \quad \textbf{(6)}$$

Proof:

$\because PK_{KGC} = Q_{KGC} \cdot G$

$\therefore P_A = [l_A + h(r_A \| id_A)] \cdot G + [q_{Ax} + h(id_A)] \cdot PK_{KGC}$

$\therefore P_A = l_A G + h(r_A \| id_A) \cdot G + [q_{Ax} + h(id_A)] \cdot PK_{KGC}$

$\because T_A = h(r_A \| id_A) \cdot G$

$\therefore P_A = l_A \cdot G + T_A + [q_{Ax} + h(id_A)] \cdot PK_{KGC}$

$\because PK_A = T_A + [l_A - h(id_A)] \cdot G$

$\because T_A = PK_A - [l_A - h(id_A)] \cdot G = PK_A - l_A \cdot G + h(id_A) \cdot G$

$\therefore P_A = l_A \cdot G + PK_A - l_A \cdot G + h(id_A) \cdot G + [q_{Ax} + h(id_A)] \cdot PK_{KGC}$

$\therefore P_A = PK_A + h(id_A) \cdot G + [q_{Ax} + h(id_A)] \cdot PK_{KGC}$

Once the information associated with the participant users (z) is stored in the KGC during the registration process, the users can compute Q_z and verify P_z using PK_z and T_z. Users may authenticate each other using (id_z, PK_z, P_z) without KGC

Authentication phase.

After Alice and Bob obtain valid identities from KGC they can authenticate each other by verifying (id_A, PK_A, P_A) and (id_B, PK_B, P_B) before transmitting messages. Bob verifies Alice using the following equation:

$$P'_A = PK_A + h(id_A) \cdot G + [q_{Ax} + h(id_A)] \cdot PK_{KGC} \quad \textbf{(7)}$$

$$P'_A ? = P_A \quad \textbf{(8)}$$

Similarly, Alice may verifies Bob using the following equation:

$$P'_B ? = P_B \quad \textbf{(9)}$$

Messages exchange phase.

(1) Alice transmits a ciphertext message to Bob. Alice uses the $f_1()$ function to encode the desired message as a point on an elliptic curve, ensuring that $J_A = f_1(m_A) = (x_j, y_j)$.

A. Select any integer $j \in Z_n^*$, then calculate the ciphertext (L_1, L_2), where $L_1 = j \cdot G$, $Y = (x_\alpha, y_\alpha) = j \cdot P_B$, $L_2 = (x_1, y_1) = (x_\alpha \cdot x_j \bmod n, y_\alpha \cdot y_j \bmod n)$.

B. Alice transmits the ciphertext (L_1, L_2) to Bob.

C. Bob decrypts the ciphertext as follows:

(a) Calculate $Z = (x_z, y_z) = Q_B \cdot L_1$.

(b) Calculate the plaintext $J_A = (x_l \cdot x_z^{-1} \bmod n,\ y_l \cdot y_z^{-1} \bmod n)$.

D. Bob uses the $f_2()$ function to encode the point on an elliptic curve as a message, ensuring that $m_A = f_2(J_A)$

(2) Bob transmits a ciphertext message to Alice. The calculation procedures are the same as explained previously.

Signature exchange phase.

The initial signer Alice performs the following actions:

(1) Randomly select a keystone $k \in K$ and establish $s_2 = h(k,\ m_A,\ m_B)$.

A. Perform $\sigma_A \leftarrow ASIGN(P_A,\ P_B,\ Q_A,\ s_2,\ m_A)$

B. Select a random number $\alpha \in Z_n$.

C. Calculate $U_1(x_1,\ y_1) = \alpha G + s_2 P_B$. **(10)**

D. Calculate $c = h(m_A,\ x_1)$. **(11)**

E. Calculate $s_1 = (\alpha - c)Q_A^{-1}$. **(12)**

F. Output $\sigma_A = (c,\ s_1,\ s_2)$ **(13)**

Send an ambiguous signature to the matching signer Bob.

After receiving the ambiguous signature Bob performs the following actions:

(1) Calculate $U_2(x_2,\ y_2) = cG + s_1 P_A + s_2 P_B$. **(14)**

(2) Confirm that the signature is verified using the AVERIFY algorithm: $c = h(m_A,\ x_2)$. **(15)**

If the verification fails, Bob terminates the agreement. If not, Bob performs the following actions:

(1) Randomly select $u \in Z_n$ and calculate $U_3(x_3,\ y_3) = uP_B$ and $u' = x_3$.

(2) Calculate $U_4(x_4,\ y_4) = (Q_B t)P_A$ **(16)**

(3) Calculate $k' = h(x_4,\ m_A,\ m_B)$. **(17)**

(4) Establish $s_1' = s_2 + h(k') \bmod n$. **(18)**

(5) Perform $\sigma_B \leftarrow ASIGN(P_B,\ P_A,\ Q_B,\ s_1',\ m_B)$

A. Select a random number $\beta \in Z_n$.

B. Calculate $U_5(x_5,\ y_5) = \beta G + s_1' P_A,\ c' = h(m_B,\ x_5)$. **(19)**

C. Calculate $s_2' = (\beta - c')Q_B^{-1}$. **(20)**

D. Output $\sigma_B = (c',\ s_1',\ s_2')$. **(21)**

(6) Send a signature $(\sigma_B = (c',\ s_1',\ s_2'),\ P_A,\ P_B,\ u')$ to Alice.

After receiving the signature, Alice performs the following actions:

(1) Calculate $r = Q_i u'$ and $k' = h(r,\ m_A,\ m_B)$.

(2) Test whether $s_1' = s_2 + h(k') \bmod n$ is true. If not, Alice terminates the protocol. **(22)**

(3) Calculate $U_6(x_6,\ y_6) = c'G + s_1' P_A + s_2' P_B$. **(23)**

(4) Confirm that the signature is verified using the AVERIFY algorithm: $c' = h(m_B,\ x_6)$. **(24)**

(5) If AVERIFY verification is successful, then Alice publically releases the keystone $(k,\ k')$, simultaneously binding and bringing signatures $(\sigma_A = (c,\ s_1,\ s_2),\ P_A,\ P_B,\ m_A)$ and $(\sigma_B = (c',\ s_1',\ s_2'),\ P_A,\ P_B,\ m_B)$ into effect.

Public verification phase. After the keystone $(k,\ k')$ is publically released, if $s_2 = h(k,\ m_A,\ m_B)$ and $s_2' = s_2 + h(k') \bmod n$ is true, anyone can confirm that the ambiguous signatures $(\sigma_A = (c,\ s_1,\ s_2))$ and $(\sigma_B = (c',\ s_1',\ s_2'))$ were signed by either Alice or Bob.

3.2 Structure Chart of Proposed Scheme

The structure chart of proposed scheme in this study is shown in Figure 1.

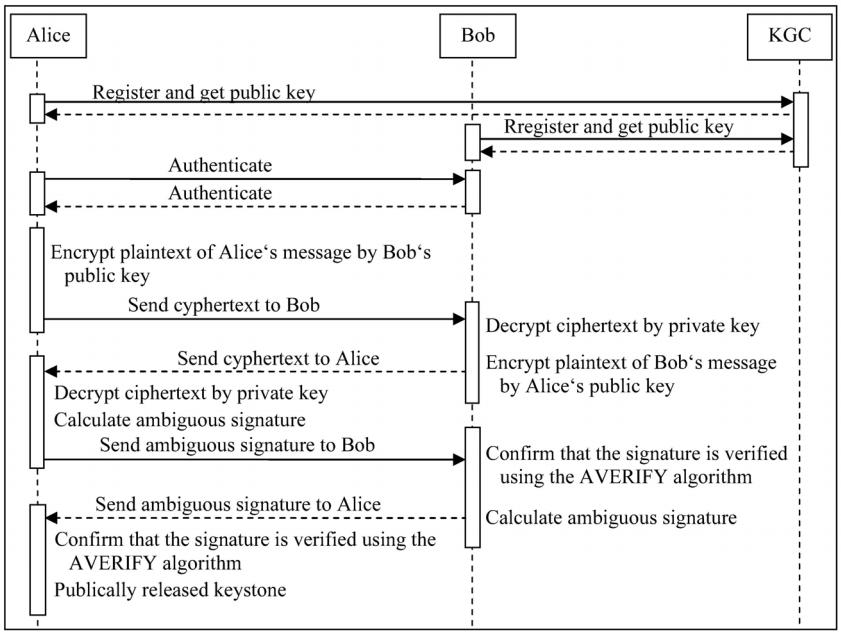

Figure 1. The structure chart of proposed scheme

4 Example of Proposed Scheme

This chapter explains propose scheme using the real ECC number point.

4.1 Initial Phase

The KGC defines the system parameters used to create self-certified public keys for the registered users. The KGC selects the parameters for the elliptic curve domains and these specific items are defined geometrically with the underlying fields. From $y^2 = x^3 + 9x + 6 \bmod 2819$ establish $E_p(a, b)$, and select n = 2801 with maximum order, base point $G(1, 4)$, $n \cdot G = O$ on $E_p(a, b)$; $h()$: KGC also chooses a one-way hash function.

The KGC selects a secure value $Q_{KGC} = 3$, then computes $PK_{KGC} = Q_{KGC} \cdot G = (396, 2483)$.

The KGC keeps the Q_{KGC} in secret, then publishes the $E, G, p, PK_{KGC}, h()$.

4.2 Key Generation Phase

Alice $id_A = 135$ selects a secure value $r_A = 5$, computes.

$T_A = h(r_A \| id_A) \cdot G = 3854795 \cdot G = 619 \cdot G = (599, 373)$

Alice keeps the r_A secret and then sends the id_A and T_A to KGC.

KGC selects a secure value $l_A = 7$ and calculates a public key PK_A and its mainfest M_A for Alice through the following equations.

$$PK_A = T_A + [l_A - h(id_A)] \cdot G = 619G + [7 - 981]$$
$$= 2446G = (q_{Ax}, q_{Ay}) = (1199, 498)$$
$$M_A = l_A + Q_{KGC}[q_{Ax} + h(id_A)]$$
$$= 7 + 3[1199 + 981] = 6547$$

KGC sends the PK_A and M_A to Alice.

Alice then derives a secret key Q_A as. $Q_A = M_A + h(r_A \| id_A) = 6547 + 3854795 = 3861342$ also verifies the authenticity of PK_A by testing if

$$P_A = Q_A \cdot G = 1564G$$
$$= [l_A + Q_{KGC}(q_{Ax} + h(id_A))] \cdot G + h(r_A \| id_A) \cdot G$$
$$= [7 + 3(1199 + 981)]G + 619G$$
$$= 1564G$$
$$= (2103, 2119)$$

4.3 Authentication Phase

After Alice and Bob obtain valid identities from KGC they can authenticate each other by verifying (id_A, PK_A, P_A) and (id_B, PK_B, P_B) before transmitting messages. Bob verifies Alice using the following equation:

$$P_A' = PK_A + h(id_A) \cdot G + [q_{Ax} + h(id_A)] \cdot PK_{KGC}$$
$$= 2446G + 981G + (1199 + 981)3G$$
$$= 1564G$$
$$P_A' \stackrel{?}{=} P_A$$

Similarly, Alice may verifies Bob using following equation:

$$P_B' \stackrel{?}{=} P_B$$

4.4 Messages Exchange Phase

Alice transmits a ciphertext message to Bob.

$m_A = 8, m_B = 9$.

(1) Alice uses the $f_1()$ function to encode the desired message as a point on an elliptic curve, ensuring that $J_A = f_1(m_A) = (x_j, y_j) = (13, 43)$.

(2) Select any integer $j \in Z_n^*$, then calculate the ciphertext (L_1, L_2), where

$L_1 = j \cdot G = (1572, 1670)$

$Y = (x_\alpha, y_\alpha) = j \cdot P_B = 55G = (1273, 1796)$,

$L_2 = (x_l, y_l) = (x_\alpha \cdot x_j \bmod n, y_\alpha \cdot y_j \bmod n)$

= (2544, 1601).

Alice transmits the ciphertext (L_1, L_2) to Bob.

Bob decrypts the ciphertext as follows:

(1) Calculate $Z = (x_z, y_z) = n_B \cdot L_1 = 5 \cdot 11G = (1273, 1796)$.

(2) Calculate the plaintext

$J_A = (x_l \cdot x_z^{-1} \bmod n, y_l \cdot y_z^{-1} \bmod n)$

$= (2544 \cdot 1273^{-1} \bmod n, 1601 \cdot 1796^{-1} \bmod n)$

= (13,43).

Bob uses the $f_2()$ function to encode the point on an elliptic curve as a message, ensuring that $m_A = f_2(J_A) = 8$.

Bob transmits a ciphertext message to Alice. The calculation procedures are the same as explained previously.

4.5 Signature Sign and Verify Phase

The initial signer Alice performs the following actions:

Randomly select a keystone $k = 7$ and establish $s_2 = h(k, m_A, m_B) = h(7, 15, 17) = 413768$.

Perform $\sigma_A \leftarrow ASIGN(P_A, P_B, n_A, s_2, m_A)$.

(1) Select a random number $\alpha = 9$.

(2) Calculate $U_1(x_1, y_1) = \alpha G + s_2 P_B = 9G + 2194G = (179, 77)$.

(3) Calculate $c = h(m_A, x_1) = h(15, 179) = 21563$.

(4) Calculate $s_1 = (\alpha - c)n_A^{-1} = (9 - 21563)3^{-1} = 288$.

(5) Output $\sigma_A = (c, s_1, s_2) = (21563, 288, 413768)$.

Send an ambiguous signature to the matching signer Bob. After receiving the ambiguous signature, Bob performs the following actions:

(1) Calculate $U_2(x_2, y_2) = cG + s_1P_A + s_2P_B$.

$= cG + (\alpha - c)Q_A^{-1} \cdot Q_A \cdot G + s_2P_B$

$= \alpha G + s_2P_B = (179, 77)$

(2) Confirm that the signature is verified using the AVERIFY algorithm: $c = h(m_A, x_2)$. Owing to $U_2(x_2, y_2) = \alpha G + s_2P_B = U_1(x_1, y_1)$, $c = h(m_A, x_2) = h(m_A, x_1)$.

The verification successes, Bob performs the following actions:

(1) Randomly select $u = 37$ and calculate $U_3(x_3, y_3) = uP_B = 185G = (79, 2677)$ and $u' = x_3$

(2) Calculate $U_4(x_4, y_4) = (Q_At)P_A = 555G = (1351, 2797)$.

(3) Calculate $k' = h(x_4, m_A, m_B) = h(1351, 15, 17)$.

(4) Establish $s_1' = s_2 + h(k') \bmod n$.

(5) Perform $\sigma_B \leftarrow ASIGN(P_B, P_A, Q_B, s_1', m_B)$.

A. Select a random number $\beta = 29$.

B. Calculate $U_5(x_5, y_5) = \beta G + s_1'P_A$, $c' = h(m_B, x_5)$.

C. Calculate $s_2' = (\beta - c')Q_B^{-1}$.

D. Output $\sigma_B = (c', s_1', s_2')$.

(6) Send a signature $\sigma_B = ((c', s_1', s_2'), P_A, P_B, u')$ to Alice.

After receiving the signature, Alice performs the following actions:

(1) Calculate $r = Q_iu'$ and $k' = h(r, m_A, m_B) = h(1351, 15, 17)$.

(2) Test whether $s_1' = s_2 + h(k') \bmod n$ is true. If not, Alice terminates the protocol.

(3) Calculate $U_6(x_6, y_6) = c'G + s_1'P_A + s_2'P_B$

$= c'G + s_1'P_A + (\beta - c')Q_B^{-1} \cdot Q_B \cdot G$

$= \beta G + s_1'P_A$

(4) Confirm that the signature is verified using the AVERIFY algorithm: $c' = h(m_B, x_6)$. Owing to $U_6(x_6, y_6) = \beta G + s_1'P_A = U_5(x_5, y_5)$ $c = h(m_B, x_6) = h(m_B, x_5)$.

(5) If AVERIFY verification is successful, then Alice publically releases the keystone (k, k'), simultaneously binding and bringing signatures $(\sigma_A = (c, s_1, s_2), P_A, P_B, m_A)$ and $(\sigma_B = (c', s_1', s_2'), P_A, P_B, m_B)$ into effect.

4.6 Publically Verify Phase

After the keystone (k, k') is publically released, if $s_2 = h(k, m_A, m_B)$ and $s_1' = s_2 + h(k') \bmod n$ is true, anyone can confirm that the ambiguous signatures $(\sigma_A = (c, s_1, s_2))$ and $(\sigma_B = (c', s_1', s_2'))$ were signed by either Alice or Bob.

5 Security and Benefits Analyses

The security encryption mechanism proposed in this study is based primarily on the ECDLP, asymmetric encryption methods, concurrent signature scheme and one-way hash function to achieve the information security management requirements. The proposed mechanism satisfies the correctness, ambiguity, unforgeability, fairness, accountability and self-certified approach concurrent signature security requirements recommended by Chen et al. [7] and Wang et al. [16]. Security and benefits analysis are explored in the following section.

5.1 Security Analysis

Correctness. According to Chen et al. [7], if the scheme passes through the ASIGN algorithm and AVERIFY $(\sigma_A, y_A, y_B, m_A) = \text{accept}$, and AVERIFY $(\sigma_B, y_A, y_B, m_B) = \text{accept}$ is true, both the message and ambiguous signature are correct.

Owing to

$U_2(x_2, y_2) = cG + s_1P_A + s_2P_B$

$= cG + (\alpha - c)n_A^{-1} \cdot n_A \cdot G + s_2P_B = \alpha G + s_2P_B.$

Because (10) $U_1(x_1, y_1) = \alpha G + s_2P_B$ and (11) $c = h(m_A, x_1)$, AVERIFY $(\sigma_A, y_A, y_B, m_A) = \text{accept}$, this study satisfies the correctness requirement.

Ambiguity. For ambiguous signatures under the concurrent signature scheme third parties are unable to know the original signer of an ambiguous signature until the keystone is released by one of the two parties. Subsequently, third parties can use the released information to verify the signer of the ambiguous signature.

For the ambiguous signatures of (13) $\sigma_A = (c, s_1, s_2)$ and (21) $\sigma_B = (c', s_1', s_2')$ in this study, because the messages from the two transaction parties m_A and m_B are processed through the one-way no collision hash function calculation in (11) $c = h(m_A, x_1)$ and (24) $c' = h(m_B, x_6)$, third parties cannot determine the identity of the signer; thus, signature ambiguity is achieved.

Unforgeability. Unforgeability refers to the mechanism that precludes data content from being maliciously tampered with or altered by third parties during data transmission and ensures data integrity and accuracy at the receiving end. Regardless of the number of packets intercepted during a transmission, the original ciphertext or plaintext data are protected from deciphering or recovery, thereby preventing tampering and forgery.

If Bob seeks to produce another ambiguous signature for his message $\bar{m}_B$, and the forged signature $\bar{\sigma}_B$ is bound and comes into effect with σ_A after the keystone is released, Bob will fail. Because the messages m_A and m_B to be exchanged by Alice and Bob are already bound to the keystone fix, the correctness of the forged signature $(\bar{\sigma}_B, \bar{m}_B)$ signed by Bob is subject to VERIFY algorithm verification by

anyone following keystone release (k, k').

Fairness. Concurrent signature fairness refers to the ability of anyone to confirm that the signatures were signed by Alice and Bob following protocol completion.

This study assumes that Bob is the deceiving party. Because the messages from both parties have already been bound in the keystone fix, the false signature signed by Bob is subject to VERIFY algorithm verification by anyone following keystone release (k, k').

Accountability. Accountability requires the signers to convince themselves and any third party that the messages signed in the ambiguous signature are the unique set generated in one protocol run. Any signer could not generate an ambiguous signature for any messages other then the one he sent to other signers in his ambiguous signature, which could satisfy the VERIFY and AVERIFY algorithm.

In this protocol we bind the messages from Alice and Bob to the keystone. If anyone cheats using a fake message it cannot pass the VERIFY and AVERIFY algorithm.

Self-certified approach. Our scheme introduces a self-certified approach, known as a public key authentication cryptosystem and makes it resistant against the foregoing attacks. In this mechanism each peer obtains a valid certificate along with corresponding identity information and holds one session key from the participants. The session key ensures that peers can communicate in spite of any possible attacks, even if the KGC does not collude. Messages sent between the communication peers are self-certified, and hence, the certificates can be used to verify the identities if applicable. The proposed measure supports off-line identity assurance in addition to on-line identity verification. Each peer can rely on KGC's public key to reliably verify the authenticity of each participant's identity. This method is effective in avoiding untrustworthy KGC misappropriating the user's secret key.

We say that a self-certified scheme is presently not protected against the adaptive chosen message attack if no polynomial bounded adversary A has a non-negligible advantage against the challenger in the following game. The challenger takes a security parameter r_A' and runs the setup algorithm. It gives the adversary the resulting system parameters and a public key PK_A from the certificate authority. If an attacker attempts to carry out an attack by revealing the private key from the public key from PK_A, he or she can then play the role of PK_A to forge. In that case, the attacker must solve the ECDLP given by PK_A. Table 2 presents a summary comparison of various security mechanisms for concurrent signature schemes.

Table 2. Comparison of security mechanisms

Algorithm	Concurrent Signature [7]	Improved i2PCS1 [16]	Proposed Scheme
Correctness	V	V	V
Ambiguity	V	V	V
Unforgeability	V	V	V
Fairness	V	V	V
Accountability	X	V	V
Self-certified approach	X	X	V

5.2 Benefit Analysis

The notations and computational relationships were defined prior to these analyses (Table 3). The modulus additions and modulus subtractions were excluded because of short computational durations. Regarding the computational amounts discussed in prior literature [28-30], Table 4 presents an analytical comparison of improved accountability for the concurrent signature. To ensure rigorous security the message exchange phase involves encoding and decoding by the ECC. Thus, increased time is required to add the overall computations. The other algorithm phases are ECC multiplication replaces the modular exponentiation, so the proposed scheme is better than improved concurrent signature accountability [16].

Table 3. Inter-relations of time complexity computations

Notation	Definition	Inter-relation
T_{MUL}	The time for the modular multiplication	$= T_{MUL}$
T_{EXP}	The time for the modular exponentiation	$\approx 240T_{MUL}$
T_{ADD}	The time for the modular addition	(negligible)
T_{INVS}	The time for the modular inverse multiplication	$\approx 240T_{MUL}$
T_{ECMUL}	The time for the ECC multiplication with ECMUL	$\approx 29T_{MUL}$
T_{ECADD}	The time for the ECC addition with ECADD	$\approx 5T_{MUL}$
t_h	The time for the hash function	$\approx 0.4T_{MUL}$

6 Conclusions

Past proposed concurrent signature schemes were all vulnerable to the message substitute attack. This research applied the accountable concurrent signature scheme to fix this weakness and proposed a new concurrent signature scheme that prevents such attacks. This study also found that the modified accountable concurrent signature procedures were inconsistent with actual e-commerce procedures and transmitted messages in plaintext format, incurring a message theft risk. Therefore, this study proposed a concurrent signature scheme based on ECC to achieve the same security strength advantages using a shorter key length than the previous scheme. This proposed scheme also

Table 4. Comparison of time complexity

Algorithm	Concurrent Signature [6]		Improved of i2PCS1 [15]		Proposed Scheme	
Items of comparison	Time complexity	Estimation	Time complexity	Estimation	Time complexity	Estimation
Setup	$2T_{\text{EXP}}$	$\approx 480T_{MUL}$	$2T_{\text{EXP}}$	$\approx 480T_{MUL}$	T_{ECMUL}	$\approx 58T_{MUL}$
Register	None	None	None	None	$8t_h + 2T_{MUL}$ $+6T_{ECMUL}$ $+8T_{ADD}$ $+2T_{ECADD}$	$\approx 189.2T_{MUL}$
Authentication	None	None	None	None	$4t_h$ $+4T_{ECMUL}$ $+4T_{ADD}$	$\approx 117.6T_{MUL}$
Message exchange	None	None	None	None	$4t_h$ $+3T_{ECMUL}$ $+2T_{INVS}$	$\approx 571T_{MUL}$
Signature exchange	$9t_h + 2T_{MUL}$ $+2T_{INVS}$ $+4T_{EXP}$ $+3T_{ADD}$	$\approx 485.6T_{MUL}$	$9t_h + 2T_{MUL}$ $+2T_{INVS}$ $+4T_{EXP}$ $+3T_{ADD}$	$\approx 485.6T_{MUL}$	$9t_h + 2T_{MUL}$ $+2T_{INVS}$ $+12T_{EXP}$ $+10T_{ADD}$	$\approx 835.6T_{MUL}$
Publically verify	$2t_h + T_{ADD}$	$\approx 0.8T_{MUL}$	$2t_h + T_{ADD}$	$\approx 0.8T_{MUL}$	$2t_h + T_{ADD}$	$\approx 0.8T_{MUL}$
Total	$\approx 966.4T_{MUL}$		$\approx 966.4T_{MUL}$		$\approx 1772.2T_{MUL}$	

reduces the computational cost, key management and storage space and prevents the risk of transmission theft, while permiting smart design agreements. The reference self-certification method strengthens the identity authentication mechanism to prevent identity spoofing attacks.

References

[1] F. Bao, R. H. Deng, W. Mao, *Efficient and Practical Fair Exchange Protocols with Off-line TTP, IEEE Symposium on Security and Privacy*, Oakland, CA, 1998, pp. 77-85.

[2] N. Asokan, V. Shoup, M. Waidner, Optimistic Fair Exchange of Digital Signatures, in: K. Nyberg (Ed.), *Advances in Cryptology-EUROCRYPT'98, EUROCRYPT 1998, Lecture Notes in Computer Science*, Vol. 1403, Springer, 1998, pp. 591-606.

[3] N. Asokan, V. Shoup, M. Waidner, *Optimistic Fair Exchange of Digital Signatures, IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 4, pp. 593-610, April, 2000.

[4] R. Indrajit, R. Indrakshi, Fair Exchange in E-commerce, *ACM SIGecom Exchanges*, Vol. 3, No. 2, pp. 9-17, May, 2002.

[5] D. Boneh, C. Gentry, B. Lynn, H. Shacham, Aggregate and Varifiably Encrypted Signatures from Bilinear Maps, in: E. Biham (Ed.), *Advances in Cryptology - EUROCRYPT'03, EUROCRYPT 2003, Lecture Notes in Computer Science*, Vol. 2656, Springer, 2003, pp. 416-432.

[6] J. A. Garay, M. Jakobsson, P. MacKenzie, Abuse-free Optimistic Contract Signing, in: M. Wiener (Ed.), *Advances in Cryptology-CRYPTO'99, CRYPTO 1999, Lecture Notes in Computer Science*, Vol. 1666, Springer, 1999, pp. 449-466.

[7] L. Chen, C. Kudla, K. G. Paterson, Concurrent Signatures, in: C. Cachin, J. L. Camenisch (Eds.), *Advances in Cryptology - Eurocrypt '04, EUROCRYPT 2004, Lecture Notes in Computer Science*, Vol. 3027, Springer, 2004, pp. 287-305.

[8] W. Susilo, Y. Mu, F. Zhang, Perfect Concurrent Signature Schemes, in: J. Lopez, S. Qing, E. Okamoto (Eds.), *Information and Communications Security, ICICS 2004, Lecture Notes in Computer Science*, Vol. 3269, Springer, 2004, pp. 14-26.

[9] G. Wang, F. Bao, J. Zhou, The Fairness of Perfect Concurrent Signatures, in: P. Ning, S. Qing, N. Li (Eds.), *Information and Communications Security, ICICS 2006, Lecture Notes in Computer Science*, Vol. 4307, Springer, 2006, pp. 435-451.

[10] S. S. M. Chow, W. Susilo, Generic Construction of (identity-based) Perfect Concurrent Signatures, in: S. Qing, W. Mao, J. López, G. Wang (Eds.), *Information and Communications Security, ICICS 2005, Lecture Notes in Computer Science*, Vol. 3783, Springer, 2005, pp. 194-206.

[11] K. Nguyen, Asymmetric Concurrent Signatures, in: S. Qing, W. Mao, J. López, G. Wang (Eds.), *Information and Communications Security, ICICS 2005, Lecture Notes in Computer Science*, Vol. 3783, Springer, 2005, pp. 181-193.

[12] W. Susilo, Y. Mu, Tripartite Concurrent Signatures, in: R. Sasaki, S. Qing, E. Okamoto, H. Yoshiura (Eds.), *Security and Privacy in the Age of Ubiquitous Computing, SEC 2005, IFIP Advances in Information and Communication Technology*, Vol. 181, Springer, 2005, pp. 425-441.

[13] D. Tonien, W. Susilo, R. Safavi-Naini, Multi-party Concurrent Signatures, in: S. K. Katsikas, J. López, M. Backes, S. Gritzalis, B. Preneel (Eds.), *Information Security, ISC 2006, Lecture Notes in Computer Science*, Vol. 4176,

Springer, 2006, pp. 131-145.
[14] Y. Li, D. He, X. Lu, Accountability of Perfect Concurrent Signature, *International Conference on Computer and Electrical Engineering*, Phuket, Thailand, 2008, pp. 773-777.
[15] Y. Zhang, X. Wang, Message Substitute Attack on Concurrent Signatures Protocol and its Improvement, *International Symposium on Electronic Commerce and Security*, Guangzhou, China, 2008, pp. 497-501.
[16] Z.-H. Wang, Z.-Q. Chen, J.-J. Wu, Using and Improvement of Concurrent Signature for Fair Exchange, *Communications of the CCISA*, Vol. 16, No. 3, pp. 60-71, July, 2010.
[17] P.-C. Su, C.-F. Chang, C.-L. Chang, Design of a Self-Certified and Security Concurrent Signature Method, *Advanced Management Information Technologies and Applications Conference (AMITAC2014)*, Hsinchu, Taiwan, 2014, pp. 275-280.
[18] V. S. Miller, Use of Elliptic Curves in Cryptography, in: H. C. Williams (Ed.), *Advances in Cryptology - CRYPTO'85 Proceedings, CRYPTO 1985, Lecture Notes in Computer Science*, Vol. 218, Springer, 1986, pp. 417-426.
[19] N. Koblitz, Elliptic Curve Cryptosystems, *Mathematics of Computation*, Vol. 48, No. 177, pp. 203-209, January, 1987.
[20] V. Gupta, D. Stebila, S. Fung, S. C. Shantz, N. Gura, H. Eberle, Speeding up Secure Web Transactions Using Elliptic Curve Cryptography, *11th Network and Distributed System Security Symposium*, San Diego, CA, 2004, pp. 231-239.
[21] R. Shanmugalakshmi, M. Prabu, Research Issues on Elliptic Curve Cryptography and Its Applications, *International Journal of Computer Science and Network Security*, Vol. 9, No. 6, pp. 19-22, June, 2009.
[22] M. Girault, Self-certified Public Keys, in: D. W. Davies (Ed.), *Advances in Cryptology- EUROCRYPT'91, EUROCRYPT 1991, Lecture Notes in Computer Science*, Vol. 547, Springer, 1991, pp. 490-497.
[23] S. Saeednia, Identity-based and Self-certified Key Exchange Protocols, *Second Australasian Conference on Information Security and Privacy*, Sydney, Australia, 1997, pp. 303-313.
[24] T.-C. Wu, Y.-S. Chang, T.-Y. Lin, Improvement of Saeednia's Self-certified Key Exchange Protocols, *Electronics Letters*, Vol. 34, No. 11, pp. 1094-1095, May, 1998.
[25] S. Saeednia, A Note on Girault's Self-certified Model, *Information Processing Letters*, Vol. 86, No. 6, pp. 323-327, June, 2003.
[26] W.-J. Tsaur, Several Security Schemes Constructed Using ECC-based Self-certified Public key Cryptosystems, *Applied Mathematics and Computation*, Vol. 168, No. 1, pp. 447-464, September, 2005.
[27] Z. Shao, Self-certified Signature Scheme from Pairings, *Journal of Systems and Software*, Vol. 80, No. 3, pp. 388-395, March, 2007.
[28] W.-H. Kuo, P.-C. Su, J.-L. Liu, A Secure Hybrid P2P Network Mobile Communication Device Design, *Applied Mechanics and Materials*, Vol. 479-480, pp. 968-972, December, 2013.
[29] P.-C. Su, Enhanced Short Signature Scheme with Hybrid Problems, *Computers and Electrical Engineering*, Vol. 37, No. 2, pp. 174-179, March, 2011.
[30] R.-C. Wang, W.-S. Juang, C.-L. Lei, A Web Metering Scheme for Fair Advertisement Transactions, *International Journal of Security and Its Applications*, Vol. 2, No. 4, pp. 49-56, October, 2008.

Biographies

Shu-Hwang Liaw received his B.Sc. from the Chinese Military Academy in 1990, the M.B.A. degrees in Information Management from National Defense Management College, Taiwan in 1996. He currently works at the coding and information security laboratory and is a Ph.D. candidate in the Department of electronic engineering, Chang Gung University, Taiwan. His research interests include error-control coding, cryptography and information security.

Erl-Huei Lu was born in Taiwan, Republic of China, on October 11, 1950. He received the B.S. and M.S. degrees in electrical engineering from Chung Cheng Institute of Technology, Taiwan, in 1974 and 1980, respectively, and the Ph.D. degree in electrical engineering from National Cheng Kung University, Taiwan, in 1988. Lu is a professor in the Department of Electrical Engineering, Chang Gung University, Taiwan. His research interests include error-control coding, network security, and systolic architectures.

Henry Ker-Chang Chang received his B.Sc. from the Chinese Military Academy in 1975, the M.B.A. from TamKang University, Taipei, Taiwan, in 1979, and Ph.D. from National Chiao Tung University, Hsinchu, Taiwan, in 1989. He jointed the Graduate School of Resources Management, National Defence Management College, Taipei, Taiwan, as Associate Professor in 1990. Until 1999, he transferred to Chang Gung University, Taiwan. He is now the Professor in Department of Information Management. His research interests are in the areas of Information Security, Applied Cryptography, Electronic Commerce (EC) and Management Information Systems (MIS).

Pin-Chang Su received his B.S. and M.S. degrees in Information Management from National Defense Management College, Taiwan in 1990, 1996 respectively, and his Ph.D. degree in Electrical Engineering from Chang Gung University, Taiwan in 2007. Currently, he is a professor in the Information Management Department at National Defense Management College, National Defense University. His research mainly focuses on algorithms design in error-control coding, cryptography and information security, and e-commerce systems design.