Self-Certified Key Exchange Scheme Based on Hybrid Mode Problems

Pin-Chang Su[1*] and Chien-Hua Tsai[2]

[1]*Department of Information Management ,College of Management, National Defense University*
[2]*Department of Accounting Information, Chihlee Institute of Technology*

ABSTRACT

In this paper, an enhanced key exchange scheme based on one-round and self-certified protocol has been proposed to establish more secure channel on which we are able to exchange session keys between two participants at a time providing the provably-secure scheme for two-party authenticated key exchange. This paper provides four notable advantages; (1) the scheme provides a strong one-round and self-certified type of protocol, which users can trust completely; (2) the scheme offers higher speeds and smaller certificate sizes than other existing public key schemes; (3) both distributing a session key and verifying the validity of public key can be concurrently achieved in a logically single step; (4) since the proposed methods are combined with the ID-based, linearly shift knapsack(LSK) and elliptic curve cryptography(ECC) public key cryptosystem, they can demonstrate the feasibility of constructing a fast and extremely secure user identification system.

Keywords: Key exchange, One-round, Self-certified, ECC.

運用混合式計算難題設計具自我證認的金鑰交換機制

蘇品長[1*] 蔡建華[2]

[1]國防大學管理學院資訊管理學系
[2]致理技術學院會計資訊學系

摘 要

雖然已經有許多學者提出各式的公開密鑰交換演算法，但實際上當所運用的演算方法有安全的顧慮時，如何強化安全性是公開密鑰系統必須面臨的問題。本研究為設計以橢圓曲線密碼系統及線性位移背包原理，應用於網路成員間身分識別及自我驗證的快速方法，適用於網路內成員間的金鑰交換協定。本文具有四個顯著的優點(1)具自我驗證機制，用戶端能完全地信任(2)提供更快的交換速度和更小的簽證值(3)可同步完成會議金鑰分配及公鑰持有者身分確認(4)植基於身分基底、線性轉移背包、橢圓曲線密碼學等理論，設計一個更安全且可行的金鑰交換系統。

關鍵字：金鑰交換，單回合，自我認證，橢圓曲線密碼系統

文稿收件日期 98.04.10; 文稿修正後接受日期 99.2.2;*通訊作者
Manuscript received April 10, 2009; revised February 2, 2010;* Corresponding author

I. INTRODUCTION

The concept of public-key cryptography was introduced by Whitfield Diffie and Martin Hellman [1] in 1976. Since then, several public-key cryptographic algorithms have been proposed continually. In public key infrastructure, a certificate authority (CA) is needed to issue digital certificates for users. A certificate binds an entity's identity information with the corresponding public key. It has some well-know and bothersome side-effects such as the need for cross-domain trust and certificate management and certificate revocation, which requires a large amount of storage and computing [2]. Cryptographic techniques enable many seemingly impossible problems to be solved. One such problem is the construction of secure identification schemes. In order to avoid the problem and the cost of distributing the public keys, Shamir [3] firstly introduced the concept of ID-based public key cryptosystem in 1984, which allows a user to use his identity information such as name, Email address, IP address or telephone number, *et al.* as his own public key. It means that there is no need for a user to keep a public key directory or obtain other users' certificates before communication. The first ID-based signature scheme was proposed by Shamir, but the size of generated signature is quite large, which has 2048 bits when one utilizes a 1024-bit RSA modulus. In 1988, Guillou and Quisquater [4] improved Shamir's scheme and shortened the signature size to 1184 bits when one uses 1024-bit RSA modulus and 160-bit hash function. However, the size of signatures generated by the scheme [4] is still too large to be applied widely in practice, especially in environments with stringent bandwidth constraints.

In 1991, Girault [5] first proposed a self-certified public key system to resolve the problem of public key verification. A self-certified public key system has three features: First, the secret key can be determined by the user himself/herself or together by the user and CA, and does not be known to CA. Second, the user can use his/her own secret key to verify the authenticity of the self-certified public key issued by CA, and thus no extra certificate is required. Third, the task of public key verification can be further accomplished with subsequent cryptographic application (e.g., key distribution or signature scheme) in a logically single step. Therefore, public key verification of the self-certified approach provides more efficient in saving the communication cost as well as the computation effort compared to that of the certificate-based and the ID-based approaches by storage-wasting and time-consuming drawbacks.

Most existing cryptosystem designs incorporate just one cryptographic assumption, such as factoring (FC), discrete logarithm (DL), and elliptic curve discrete logarithm (ECDL) problems. These assumptions appear to be secure today, but it is possible for those efficient algorithms to be developed in the future to break one or more of these assumptions. Several cryptographic systems try to consolidate their security in solving multiple hard problems simultaneously based on FC and DL [6, 7, 8, 9, 10]. Enhancing security is the major objective for public-key cryptosystems built on multiple assumptions. Unlike the FC and DL, the general subset sum (decision) problem is known as a special case of the knapsack problem, even thought it has been proven to be NP-complete [11]. The set of all decision problems whose solutions can be verified and broken [11, 12, 13] quickly, if every subset sum problem uses the same cryptographic techniques. Some researchers believe that the broken knapsack cryptosystems to be cracked because either their construction cannot completely disguise the easy knapsack or their densities are too low. In 1989, Laih *et al.* [14] published a new cryptosystem to improve the Merkle-Hellman scheme, specifically, by transforming a superincreasing sequence into a "high density" knapsack sequence and using a linearly shift method to ameliorate the cryptosystem security, showing that the enciphering keys obtained from this algorithm have a very high probability of falling into the worst knapsack with NP-completeness category. Therefore, Shamir's approach [12] and the low density attack cannot apply the theory in this system. Some researchers think that a polynomial algorithm will be eventually invented to solve FC and DL, while the subset sum problem will remain with the NP-complete class of problems.

Key-exchange protocols are among the most basic and widely used cryptographic protocols [15, 16, 17]. Such protocols are used to derive a common session key between two (or more) parties; this session key may then be used to communicate securely over an insecure public network. Thus, securing key-exchange protocols is as basic building blocks for constructing safe, complex, higher-level protocols. For the reason, the

computational efficiency, communication efficiency, and round complexity of key-exchange protocols are very important and have been receiving a lot of attention. A simple example is this; considering traditional Diffie-Hellman key exchange protocol does not provide any authentication. It is described as a two-round protocol in which Alice first sends g^a and Bob then replies with g^b. In this particular case Alice and Bob can send their messages simultaneously, thereby "collapsing" this protocol to a single round. However, this situation will become more complicated when authentication is required. For instance, authenticated Diffie-Hellman key exchange typically involves one party signing messages sent by the other party; this may be viewed as a type of "challenge-response" mechanism. When this is done, it is no longer possible to collapse the protocol to a single round.

The motivation of our paper is based on the three points: (1) The Diffie-Hellman related assumptions have played an important role in designing various key exchange protocols. Apart from the existing Diffie-Hellman assumptions, is it possible to propose new Diffie-Hellman assumption that will be built upon to design new self-certified key exchange schemes? (2) Knapsack cryptosystems had ever received a great deal of attention in the community of cryptography and computational complexity in 1970s' and 1980s'. The basic idea of the scheme is in transforming hard or unfeasible subset sum problems into easy subset sum problems, and the subset sum problem has been proven to be NP-complete. Most of the existing Knapsack cryptosystems were broken. An interesting question is: Has the Knapsack problem already been falling from designing optimistic cryptographic protocols? (3) ECC have already been combined with the Diffie-Hellman and thus created a number of Diffie-Hellman assumptions, that have been used to design key exchange protocols. Is it possible to design a new one-round and self-certified key exchange scheme of Diffie-Hellman assumptions?

Inspired by the above motivation, we explore the possibility of designing a scheme for a key exchange which can be implemented in only a single round of authentication. Our approach using ID-based encryption provides end-to-end authentication and can simultaneously prevent leakage of user's identity. Of course, we will also ensure that our scheme is efficient with respect to other measures, including communication complexity and computational efficiency. This study presents a new LSK-type public key exchange based on a novel application of the ECDL. Also, by choosing appropriate domain parameters, the ratio of their size in bits can be controlled and the vulnerable lattices are completely disguised between numbers of elements in the ECC. The rest of this paper is organized as follows. In the next section we will give a brief introduction to some mathematical theory related to the following schemes. Section 3 then describes a public key exchange system based on these hybrid mode assumptions. Section 4 demonstrates that the proposed schemes satisfy the security conditions. Conclusions are finally drawn in Section 5.

II. BACKGROUND THEORIES

In the section we review the Diffie-Hellman problems over elliptic curve cryptosystem defined in the prime order group *G*, one-round protocols for authenticated key exchange and model of self-certified scheme**.** We also review Laih's linearly shift knapsack sum system and Jeong et al.'s one-round protocol for authenticated key exchange method [18], which will be recommended as our proposed algorithm in Section 3.

2.1 Elliptic curve cryptography

Miller [19] and Koblitz [20] first suggested the use of elliptic curves implementing public key cryptosystems. 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 $y^2 = x^3 + ax + b \bmod q \quad for\, 0 \le x \le q$.

The contents of 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:

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

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

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 *3P* = *2P* + *P*, and so on. One important property is that it is very difficult to find an integer *s* in such an equation *sP* = *Q*.

2.2 Linearly shift knapsack cryptosystem

We review Laih's method to obtain a high density knapsack and linearly shift knapsack algorithms described as follows [14]:

- High density knapsack algorithm

Step 1: Let $\bar{a} = (a_1, a_2, ..., a_l)$ be a superincreasing sequence, select two integers *w*, *m* satisfying *gcd*(*w*, *m*) = 1, where $m > \sum_{i=1}^{l} a_i$.

Step 2: Calculate the original enciphering keys $b_i \equiv a_i \times w \bmod m$ for all *i*.

Step 3: Compute and public the high density enciphering keys, $\bar{b'} = (b_1', b_2', ..., b_l')$, where $b_i' = b_i \bmod w$, then $b_i' < w$ for all *i*.

Step 4: Calculate $\bar{c} = (c_1, c_2, ..., c_l)$, where $c_i = \left\lfloor b_i / w \right\rfloor$, then $0 \le c_i \le v$, and compute the deciphering keys $a_i' = a_i - c_i$, where $v = \left\lfloor m / w \right\rfloor$ (where $\lfloor x \rfloor$ is a floor function, representing the largest integer value smaller than *x*).

It is obvious that the original Merkle-Hellman enciphering keys distributed in $[1, m]$ have been reduced to b_i' distributed in $[1, m]$, but it is with the same security. The density can be controlled by properly choosing *w* which is much less than *m*. The system parameters (*w*, *m*, $\bar{a}$) are made secret and $(\bar{a'}, \bar{b}, \bar{c}, v)$ are all discarded. The density of Laih's algorithm is higher than 0.94 in comparison with 0.5 obtain by the original Merkle-Hellman scheme.

- Linearly shift knapsack algorithm

Step 5: As high density knapsack algorithm, calculate a high density knapsack sequence $\bar{b'} = (b_1', b_2', ..., b_l')$.

Step 6: Choose a random binary sequence $\bar{t} = (t_1, t_2, ..., t_l)$, and an integer *k* with $0 < k < \min \cdot (b_i')$ *for* $t_i = 1$. Then b_i' are linearly shifted by performing $e_i = b_i' - kt_i$ and $\bar{e} = (e_1, e_2, ..., e_l)$, are published as the public enciphering keys.

The deciphering keys for intended receiver are $(k, w, m, \bar{a})$, where $(\bar{t}, \bar{b'})$ can be discarded. After that, $\bar{e} = (e_1, e_2, ..., e_l)$, are published as his public keys.

If the receiver receives $s = \sum_{i=1}^{l} b_i' x_i$, where $\bar{x} = (x_1, x_2, ..., x_l)$, is the message, he/she can decipher *s* properly just by following the normal decryption procedure [12]. However, the receiver will receive $S' = \sum_{i=1}^{l} e_i x_i$ rather than *s*. From the above, i.e., *step* 6, we obtain

$$s \times w^{-1} \equiv (\sum_{i=1}^{l} b_i' x_i) \times w^{-1} \bmod m$$
$$\equiv \sum_{i=1}^{l} (e_i + kt_i) x_i \times w^{-1} \bmod m$$
$$\equiv s' \times w^{-1} + kw^{-1} \times \sum_{i=1}^{l} t_i x_i \bmod m$$

Thus, the receiver can guess the correct $s \times w^{-1} \bmod m$ at most $y + 1 \le l + 1$ times. If the system is one-to-one, the rightness of guessing can be easily verified through normal enciphering procedures. According to Shamir's theorem [12], a random modular knapsack system with *l* generators and modular *m* is likely to be one-to-one when $l < (\log_2 m) < 2$, otherwise it is non one-to-one. That is, from this equation if *m* is chosen larger than 2^{2l}, then the system is likely to be one-to-one.

2.3 One-round protocols for authenticated key exchange

Also, we review Jeong et al.'s method [18] to obtain a one-round protocol for authenticated key exchange described as follows:

Let k be a security parameter, and G be a group of prime order $q(where|q|=k)$ with generator g. Moreover, letting h be a hash function such that $f:\{0,1\}^* \to \{0,1\}^k$, we assume that each user U_i has a public-/private-key pair $(y_i = g^{x_i}, x_i)$, and the public keys of all parties are known to all other parties in the network. Notice, however, that the standard definition of security does not include the possibility of "malicious insiders"; thus, in particular, we assume that all public-/private-keys are honestly generated. The protocol is described from the perspective on U_i, and U_j behaves analogously as its partner, i.e., the protocol is symmetric.

- Setup

Assume U_i wants to establish a session key with U_j and $U_i < U_j$.

- One-round authentication

U_i computes $k_{i,j} = y_j^{x_i}$ that it will use the value as a key for a message authentication code. Next, U_i chooses a random number $\alpha_i \in Z_q$, and computes $\tau_i \leftarrow Mac_{k_{i,j}}(i \| j \| g^{\alpha_i})$, and then sends $h(g^{\alpha_i} \| \tau_i)$ to the other party.

- Computation of the session key

U_i verifies the tag of the received message by using $k_{i,j}$. If verification procedure fails, no session key is computed. Otherwise, U_i computes a session key $sk_i = (g^{\alpha_j})^{\alpha_i}$ and the session identifier is $sid_i = h(g^{\alpha_i} \| \tau_i \| g^{\alpha_j} \| \tau_j)$.

2.4 Model of self-certified signatures

A sophisticated approach, first introduced by Girault [21], is called self-certified public key (SCPK), which can be regarded as intermediate between the identity-based approaches and the traditional PKI approaches. 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. Then we propose a concrete SCS scheme from pairings. The SCS scheme consists of four randomized algorithms [22]: 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 partial private key s_A and computes the corresponding partial public key Y_A. The actual public key of the user consists of public key of CA, partial public key and identity of the user together with 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 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

$\sigma = Sign(P_{CA}, ID_A, Y_A), (s_A, d_A), m)$.

III. OUR PROPOSED SCHEME

A secured key exchange protocol is proposed in this section based on hybrid mode algorithms. As with similar to other cryptosystems, there exists trusted key generation center (KGC) for generating cryptographically secure keys. The KGC can be closed down its service or off-line after all users have registered on a system. It consists of four phases to establish secured key exchange mechanism between two users. The four phases include initialization, registration, verification and key exchange between two users respectively.

3.1 System setup

The KGC selects the parameters of elliptic curve domains and these specific items are defined geometrically with the underlying fields [14].

- The field order q which is used for the elements of F_q.
- Two coefficients $a, b \in F_q$ that define the equation of the elliptic curve E over F_q (i.e., $y^2 = x^3 + ax + b$ in the case of a prime field).
- The order n of P, where $n = 4p_1 \times p_2 + 1$ and $p_1 = 2p_3 + 1$, $p_2 = 2p_4 + 1$ and p_1, p_2, p_3, p_4 are all large primes.

The system parameter n is made in public and p_1, p_2, p_3, p_4 can be discarded. In addition, the KGC also chooses a converting function $h(\ )$ and computes public key Q_{KGC}, such that

$$Q_{KGC} = d_{KGC}P, \quad (1)$$

where d_{KGC} is the KGC's secret key.

After that, KGC publishes E, P, n, Q_{KGC} and $h(\ )$.

Suppose that a user U_i wants to register with KGC. The procedure for user registration is stated as below:

Step 1. U_i takes the identification number id_i and randomly choose a master key $d_i \in [2,\ n-2]$ in order to obtain the signature V_i of id_i. Compute

$$V_i = h(d_i \| id_i)P, \quad (2)$$

then submit (id_i, V_i) to KGC.

Step 2. KGC selects a random number $k_i \in [2, n-2]$ and calculates a public key Q_i and its witness z_i for U_i through the following equations

$$Q_i = V_i + (k_i - h(id_i))P = (q_{ix}q_{iy}), \quad (3)$$

$$z_i = k_i + d_{KGC}(q_{ix} + h(id_i)), \quad (4)$$

where k_i is a secret pick of KGC's secret information.

Step3. U_i then derives a secret key s_i as

$$s_i = z_i + h(d_i \| id_i) (\mathrm{mod} n). \quad (5)$$

also verifies the authenticity of Q_i by testing if

$$S_i = s_i P = Q_i + h(id_i)P + (q_{ix} + h(id_i))\, Q_{KGC}. \quad (6)$$

Registration procedure must be in person or using the way to authenticate communication in some secure form. Each participant reliably knows a public key of the KGC, and only the KGC knows the corresponding private key. As Laih's scheme [14], U_i selects parameters $(w_i \| m_i \| k_i \| \overline{t_i} \| \overline{a_i})$ as his/her secret, then calculates his/her public key $EID(id_i) = \overline{p_i} = (p_{i_1} \| ... \| p_{i_l})$. If the center confirms the correctness of the relationship between U_i and id_i, he/she then calculates $\overline{c_i}$ using

$$\overline{c_i} = w_i \cdot \overline{p_i} = (c_{i_1} \| ... \| c_{i_l}) \quad (7)$$

Moreover, each participant reliably knows a public key of KGC. Once all the users have registered and got his (z_i, Q_i), the KGC does have no need to exist in network any more. In the following, we show that the secret key s_i (derived by U_i) and the public key Q_i (issued by KGC) satisfy Eq. (6).

Proof. Eq. (4) is substituted for Eq. (5), then we obtain

$$s_i = k_i + d_{KGC}(q_{ix} + h(id_i) + h(d_i \| id_i)). \quad (8)$$

Both sides of Eq. (8) multiplied by P yield that

$$\begin{aligned} s_i P &= [(k_i + d_{KGC}(q_{ix} + h(id_i)))]P + h(d_i \| id_i)P \\ &= [(k_i + h(d_i \| id_i)]P + (q_{ix} + h(id_i))Q_{KGC} \\ &= k_i P + V_i + (q_{ix} + h(id_i))Q_{KGC} \\ &= Q_i + h(id_i)P_i + (q_{ix} + h(id_i))Q_{KGC} \end{aligned}$$

This implies Eq. (6).

3.2 One-round authentication

Assuming U_i and U_j are the two users, they want to communicate with each other secretly. First, U_i selects a random binary vector $\overline{x_i} = (x_{i_1} \| ... \| x_{i_l})$ as respective parameters. We can compute the certificate from T_i. In terms of the certificate, the signature T_i is obtained as

$$T_i = (\sum_{k=1}^{l} x_{i_k} c_{j_k}) \cdot P \quad (9)$$

Second, U_i sends (id_i, S_i, Q_i, T_i) to U_j. Similarly, U_i selects a random binary vector $\overline{x_j} = (x_{j_1} \| ... \| x_{j_l})$ as respective parameters. Then the public keys are yielded as follows.

$$T_j = (\sum_{k=1}^{l} x_{j_k} c_{i_k}) \cdot P \quad (10)$$

Thus, the wrapped message (id_j, S_j, Q_j, T_j) is sent to U_i in the same way.

3.3 Computation of the session key phase

Before generating the wrapper key, U_i and U_j need to certify that (id_i, S_i, Q_i) and (id_j, S_j, Q_j) are safe and to be sent right from their own entities by checking

$$\hat{S}_j = Q_j + h(id_j)P + (q_{jx} + h(id_j))\, Q_{KGC}, \quad (11)$$

$$\hat{S}_j \stackrel{?}{=} S_j. \quad (12)$$

and

$$\hat{S}_i = Q_i + h(id_i)P + (q_{ix} + h(id_i))\, Q_{KGC}, \quad (13)$$

$$\hat{S}_i \stackrel{?}{=} S_i. \quad (14)$$

U_i, U_j therefore compute session keys thereof SK_i and SK_j each. We have

$$SK_i = d_i Q_j + (\sum_{k=1}^{l} x_{i_k} \cdot c_{j_k}) \cdot T_j \quad (15)$$

$$SK_j = d_j Q_i + (\sum_{k=1}^{l} x_{j_k} \cdot c_{i_k}) \cdot T_i \quad (16)$$

and summarize this result in the subsequent expression since SK_i and SK_j are the same.

$$SK_i = SK_j = d_i \cdot d_j P + (\sum_{k=1}^{l} x_{i_k} \cdot c_{j_k}) \cdot (\sum_{k=1}^{l} x_{j_k} \cdot c_{i_k}) \cdot P \quad (17)$$

IV. EVALUATION OF OUR SCHEME

This subsection discusses the security and performance of the proposed scheme. A secure key agreement protocol should be able to withstand the passive and active attacks and has a number of desirable security attributes such as session key, private key, impersonation and man-in-the-middle attack. The security of the entire proposed system must be measured by means of two different perspectives, i.e., pubic key cryptosystems (PKC) and protocol. We define security based on the capabilities of an adversary. Also, we allow the adversary to potentially control all communication in the network via access to a set of oracles, and the oracles answer back to the adversary. The oracle queries the model and attacks that an adversary may use in the real system [18]. We consider the following scenarios as interrelated types of queries in this scheme.

4.1 Security aspect – PKC

In general, the security of a cryptosystem is evaluated by the amount of time needed to break it. “Breaking a cryptosystem” means finding the private key used to encrypt a message. The method used to break a cryptosystem is called the “attacking method”. Normally, the time needed to break a practical cryptosystem is never actually obtained, because a cryptosystem that can be broken in a reasonable amount of time would not be considered for practical use. Instead, the amount of time needed to break a cryptosystem is a theoretical estimate of the average time needed to break a cryptosystem by a given attacking method. If there are multiple attacking methods, the time required by the most efficient method would be taken as the time needed to break the cryptosystem. Like any other cryptographic schemes, the security of our knapsack-type elliptic curve cryptosystem is evaluated in terms of its security.

4.1.1 Security under LSK problems

Although most of the knapsack cryptosystems have been broken, the general knapsack (decision) problem is a proven NP-complete, that is unlike integer factorization and discrete logarithm. It should be cautioned that, first, the NP-completeness is based on the worst-case analysis. Second, NP-completeness is a characteristic of the general problem, not of a particular instance. In this regard,

further analysis requires advanced average-case complexity and instance complexity analysis, which is still an emerging research area. Now, we use the following theorem to prove that a random sequence has a small possibility to be an image of another random sequence.

Theorem: Assume that $\bar{e} = (e_1,....,e_l)$ are uniformly distributed, independent random variables in [1, *m*]. The probability *P* that $\bar{e}$ be the image of a sequence *H*, under a modular transformation $m > \sum_{i=1}^{l} a_i$, satisfies

$$P < \left(\sum_{i=1}^{l} e_i \right) / l! < lm / l! .$$

Proof: It can be shown that if we randomly choose *l* integer h_i from [1, *m*], the probability *P* satisfying $\sum_{i=1}^{l} h_i < m$ is $P < 1/l!$.

Since there are at most $\sum_{i=1}^{l} e_i$ minima divide to $\sum_{i=1}^{l} e_i$ interval for the function $g_i(t) = e_i t - s_i m$, where $s_i = \lfloor e_i t / m \rfloor$. The probability, *P*, of success is $P < \left(\sum_{i=1}^{l} e_i \right) / l! < lm / l!$.

According to Theorem, we know that the probability for enciphering keys e_i generated by Laih's algorithm [14] being the image of a superincreasing sequence is less than 2^{-4536} and being the image of random sequence is less than 10^{-95} when $l = 100$ and $m = 2^{200}$. In other words, $\bar{e}$ have very large probability falling into the worst case of the knapsack problem.

The dilemma for the classic knapsack cryptosystem designer is that the trapdoor is easily discovered if the knapsack density is high. In other words, when someone adopts a high density and a difficult-to-discover trapdoor to invent a knapsack cryptosystem for exploiting the difficulty of the knapsack problem, the constructed system will be better than systems based on integer factorization and the discrete logarithm.

4.1.2 Security under ECDLP

This study proposes a knapsack-type key exchange protocol that fully exploits the difficulty of the ECDL, with a high density and a difficult-to-discover trapdoor.

Theorem: Given an elliptic curve *E* defined over F_q, a point $P \in E(F_q)$ of order *n,* and a point $Q \in E(F_q)$, determine the integer *l*, $0 \le l \le n$ - 1, such that $Q = lP$, provided that such an integer exists.

Proof: The Pohlig–Hellman algorithm [23] reduces the determination of *l* to the determination of *l* modulo each of the prime factors of *n*. Hence, in order to achieve the maximum possible security level, *n* should be a prime. The best algorithm known to date for ECDLP is the Pollard $\rho-$ method, as modified by Gallant, Lambert and Vanstone, and Wiener and Zuccherato, which takes about $\sqrt{\pi n}/2$ steps, where a *step* here is an elliptic curve addition. Van Oorschot and Wiener showed how the Pollard $\rho-$ method can be parallelized so that if *r* processors are used, then the expected number of steps by each processor before a single discrete logarithm is obtained is $\sqrt{\pi n}/2r$ [24]. In 1993, Menezes, Okamoto and Vanstone (MOV) [25] showed how the ECDLP can be reduced to the DLP in extension fields of F_q, where the index-calculus methods can be applied. However, this *MOV* reduction algorithm is only efficient for a very special class of curves known as supersingular curves**.** Supersingular curves are specifically prohibited in all standards of elliptic curve systems such as IEEE P1363, ANSI X9.62, and ANSI X9.63.

The *E* is taken over a finite field *K*. Then, *E*(*K*) is finite and, by Hasse's theorem [23], its cardinality is bounded by $q+1-2\sqrt{q} \le |E(K)| \le q+1+2\sqrt{q}$.. The cryptosystem security is based on the difficulties of ECDLP and knapsack problem. The ECDL is analogous to the traditional discrete logarithm problem in the multiplicative group of a finite field. As a discrete log becomes easier, it needs longer bit-lengths which are required to keep the method safe. Discrete logs in ordinary number groups are now pretty easy to solve as compared with those are in elliptic curve groups. The discrete log problem for ordinary groups has been improving steadily due to the discovery of successive refinements in the Number Field Sieve (NFS) techniques. According as Odlyzko [26] pointed out in the paper, the solution of the discrete logarithm requires $O(\exp(\text{const.}(\log q \log\log q)^{1/2}))$ integer multiplication. The advantage of ECC derives from the existence of sub-exponential algorithms of complexity $O(\exp(\text{const.}(\log q)^{1/3}(\log\log q)^{2/3}))$,

that solve the DL over F_q. In this scheme, an adversary tries to reveal the message by the public key from any user. First, any adversary must solve the ECDL problem given by (id_i, S_i, Q_i, T_i) to determine $(\sum_{k=1}^{l} x_{i_k} c_{j_k})$. Second, the adversary must solve the NPC problem to determine the message from T_i. As such, given that $\overline{c_i}$ is publicly known information, and the private key cannot be practically derived from $\overline{x_i}$.

4.2 Security aspect –Protocol

This subsection discusses the security issues regarding the proposed scheme. To analyze the security of the above method, we adopt the security measure and apply to those attack models used in [27, 28, 29]. Assumed that an adversary has total control over the communication channels, then he or she can mount parallel attacks with the previous session keys. The resulting scheme, i.e., a key exchange technique, is secure if the following requirements are satisfied.

- If both participants honestly execute the protocol, then the session key is $SK_{ij} = SK_i = SK_j$.
- No one can calculate the session key SK_{ij} except participants U_i and U_j.
- The session key is indistinguishable from a truly random number.

4.2.1 Security under PKC public key attack

The intruder can widely obtain z_i from wiretapping, off-line dictionary attacks or exchaustive attacks. Then, the adversary tries to derive the private key (k_i, d_{KGC}) from the public key issued by the KGC. To decipher, whose factorization leads to $z_i = k_i + d_{KGC}(q_{ix} + h(id_i))$., and the difficulty in mathematics traps the attacker into solving the ECDL problem given by z_i to determine (k_i, d_{KGC}).

4.2.2 Security under known session key attack

An adversary attempts to launch an attack by revealing the session key from any user's public key. Assumed that, the user has incoming message with U_i's public key to disclose the session key. To derive $(\sum_{k=1}^{l} x_{i_k} c_{j_k})$ from (id_i, S_i, Q_i, T_i), the adversary bears a strong resemblance to the above scenario. That is, in order to determine $\overline{x_i}$, he or she must solve the LSK problem and the ECDL problem given by (3), (4) and (5).

4.2.3 Security under impersonation attack

An impersonation-attack characteristic is that any attacker can, without stealing the identities, easily masquerade as a legitimate user at any time. Notably, the U_i selects a binary vector $\overline{x_i}$ such that the message $s_i = z_i + h(d_i \| id_i)$ can be used to compute the certificate information of U_i's by $S_i = s_i P = Q_i + h(id_i)P + (q_{ix} + h(id_i))\, Q_{KGC}$. Accordingly, an adversary can play the role of U_i to forge (id_i, S_i, Q_i, T_i). However, before the attacker chooses the binary vector $\overline{x_i}$, to obtain the session key the verifier is required, namely $\hat{S}_i = Q_i + h(id_i)P + (q_{ix} + h(id_i))\, Q_{KGC}$ and $\hat{S}_i \stackrel{?}{=} S_i$. As mentioned above, the attacker must again solve the ECDL and LSK problems.

4.2.4 Security under man-in-the-middle attack

When U_i sends (id_i, S_i, Q_i, T_i) to U_j, an adversary can intercept the datum from the public channels, and then play the role of U_i to cheat U_j or other users by (id_i, S_i, Q_i, T_i). The attacker does not pass the verification of $\hat{S}_i = Q_i + h(id_i)P + (q_{ix} + h(id_i))\, Q_{KGC}$ and $\hat{S}_i \stackrel{?}{=} S_i$, since the identification information S_i are considered as the inputs of the one-way function $h(\ )$, which are used in the operation z_i later. Nevertheless, we know that obtaining (k_i, d_{KGC}) from z_i is equal of computing the ECDL and LSK assumptions.

4.2.5 Security under replay attack

Like the man-in-the-middle attack or bucket-brigade attack, the purpose of a reply attack is also to pretend to be as a proper user. When U_i sending (id_i, S_i, Q_i, T_i) to U_j, an adversary is able to intercept the datum from the common

channels, and then act the role of U_i to cheat U_j or other users using (id_i, S_i, Q_i, T_i). Yet, the cheat does not succeed in verifying the (14) expression since the identification information id_j is considered as the inputs of the one-way function $h(\ )$, which are used in the operation $\hat{S}_i = Q_i + h(id_i)P + (q_{ix} + h(id_i))\,Q_{KGC}$ to get the session key.

4.2.6 Security consideration of the malicious KGC attack

An intruder might try to impersonate *KGC* by determining a relationship from the public message for (z_i, s_i). We say that a self-certified scheme is presently counterfeited against 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 the security parameters (k_i', d_{KGC}') and runs the generate algorithm. It gives the adversary the resulting system parameters and a public key Q_{KGC} of the KGC. If an attacker attempts to carry out an attack by revealing the private key (k_i', d_{KGC}') from the public key of the (z_i, s_i), then he or she can play the role of (id_i, KGC) to forge. In case of that, the attacker must solve the ECDL problem given by (z_i, s_i) to determine (k_i', d_{KGC}').

4.2.7 Security consideration of the conspiracy attack

An adversary U_i attempts to launch an attack by revealing the secret key (k_i, d_{KGC}) from any user's public key $(z_i, Q_i\ id_i)$. As for the polynomial $z_i = k_i + d_{KGC}(q_{ix} + h(id_i))$, since the secret key (k_i, d_{KGC}) is a secret pick of KGC, U_i is unable to calculate the secret key (k_i, d_{KGC}) due to infinite solutions deriving from the public key $(z_i, Q_i\ id_i)$. Please refer to Step 2 of Section 3.1, which pertains to the description. Suppose U_i and U_j try to collude to attach the KGC's secret key (k_i, k_j, d_{KGC}). For the secret key that comes from both $(z_i, Q_i\ id_i)$ and $(z_j, Q_j\ id_j)$; likewise the item is unsuccessfully reckoned. However, the adversary must solve the Linear Equations Problems.

4.3 Performance analysis and comparisons

ECC delivers the highest strength per bit of any known public-key system because of the difficulty of the hard problem upon which it is based. This greater difficulty of the hard problem - ECDL - means that smaller key sizes yield equivalent levels of security. Table 1 compares the key sizes needed for equivalent strength security in ECC with RSA and DSA. In order to present a contrast aimed at the performance, the scheme by Chen [30] and the proposed scheme are illustrated in tables. Table 2 is the definitions of the given notations, and Table 3 shows the relationships of the various operations. Then, the required time complexities in the different phases are estimated as Table 4, so that the efficiency in executing can be specifically analyzed.

Table 1. Key Size Equivalent Strength Comparison

Time to break in MIPS yeas	RSA/DSA key size	ECC key sizes	RSA/ECC key size ratio
10^4	512	106	5:1
10^8	768	132	6:1
10^{11}	1024	160	7:1
10^{20}	2,048	210	10:1
10^{78}	21,000	600	35:1

Table 2. Definitions of Notions

Notations	Definitions
T_{MUL}	The time for the modular multiplication
T_{EXP}	The time for the modular exponentiation
T_{ADD}	The time for the modular addition
T_{EC_MUL}	The time for the multiplication of a number and an elliptic curve point
T_{EC_ADD}	The time for the addition of two points in an elliptic curve

Table 3. Relationships of Various Operations

$T_{EXP} \approx 240T_{MUL}$
$T_{EC\_MUL} \approx 29T_{MUL}$
$T_{EC\_ADD} \approx 0.12T_{MUL}$
T_{ADD} is negligible

Table 4. Time Complexity and Estimation of Authenticated Key Exchange

Items	Scheme by Jeong et al.		Scheme by us	
	Time Complexity	Roughly Estimation	Time Complexity	Roughly Estimation
Setup phase	1 T_{EXP}	240 T_{MUL}	2 T_{EC_MUL} + 1 T_{EC_ADD} + 4 T_{ADD} + 2 T_{MUL} + 3 hash	58.12 T_{MUL} + 3 hash
One-round authentication	2 T_{EXP} + 1 hash	480 T_{MUL} + 1 hash	2 T_{EC_MUL} + 2n T_{ADD}	58 T_{MUL}
Computation of the session key	1 T_{EXP} + 1 hash	240 T_{MUL} + 1 hash	4 T_{EC_MUL} + 4 T_{EC_ADD} + n+2 T_{ADD} + 2hash	116.12 T_{MUL} + 2 hash

Remark: The security analysis of Jeong et al. on their cryptosystem [18] is too simple. The security of the cryptosystem against the man-in-the-middle attacks, the malicious KGC attacks and the replay attacks needs to be carefully investigated.

Due to the underlying of the ECC group structure and the knapsack-based cryptosystem, it is not to difficult to come up with the proposed scheme that can bound the worst time by $O(n^2)$ in encryption/decryption[28]. In Table 5, we compare our scheme with Jeong's one-round method. At the same cost, having concatenated LSK and ECC hard problems, the presented scheme has solid structure and will hopelessly leave the eavesdropper baffled.

Table 5. Comparison between the proposed scheme and Jeong et al one-round cryptosystem

	Algorithm	Method	Cost	Design
The proposal	LSK & ECC	Probabilistic	$O(n^2)$	One-round & Self-Certified & Hybrid Mode
Jeong et al's	DLP	Deterministic	$O(n^2)$	One-round

V. CONCLUSIONS

As mentioned in detail above, we have actually discussed and analyzed our methods. A key exchange design has been proposed based on hybrid mode problems to enhance security, while maintaining the implementation efficiency. However, it should be cautioned that, there is one possibility to break our proposed system. That is, the knapsack problem may be linearly shifted to solve integer factorization and discrete logarithm problems, while the discrete logarithm problem on such elliptic curve groups will stand as a difficult problem in the underlying finite field. As we have seen, the proposed method is robust enough against all known attacks. In addition, a possible goal for future research seems to focus our efforts on developing ID-based crypto-sign schemes that allow system users to receive crypto-sign messages from senders who do not depend on the same authority. Another interestingly open question is

the possibility of equivalence between the security and the efficiency problem. In other words, if someone designs such a cryptosystem that fully exploits the difficulty of the knapsack problem and offers attractively high speed, then it appears to make it suitable for use in low-power mobile communication environments. This paper provides four notable advantages; (1) the scheme provides a strong one-round and self-certified type of protocol, which users can trust completely; (2) the scheme offers higher speeds and smaller certificate sizes than other existing public key schemes; (3) both distributing a session key and verifying the validity of public key can be concurrently achieved in a logically single step; (4) since the proposed methods are combined with the ID-based, linearly shift knapsack(LSK) and elliptic curve cryptography(ECC) public key cryptosystem, they can demonstrate the feasibility of constructing a fast and extremely secure user identification system.

REFERENCES

[1] Diffie, W., and Hellman, M. E., "New directions in Cryptography," *IEEE Transactions on Information Theory*, Vol. 22, Issue 6, pp. 644-654, 1976.

[2] Shi, W., Zhou, Y., "An Improvement of Sui et al's Second ID-based Key Issuing Protocol" , *The 3rd International Conference on Innovative Computing Information and Control (ICICIC'08)*, 18-20 June 2008, pp. 114-117, 2008.

[3] Shamir, A., "Identity-based cryptosystems and signature schemes", *in Proc. Crypto'84*, Santa Barbara, CA, Aug. 1984, pp. 47-53, 1984.

[4] Guillou, L.C., Quisquater, J.J., "A paradoxical identity-based signature scheme resulting from zero-knowledge", *in Proc. Crypto'88*, Santa Barbara, CA, Aug. 1988, pp. 216-231, 1988.

[5] Girault, M., "Self-certified public keys", *Proceedings of the Eurocrypt'91*, pp. 491-497, 1991.

[6] Harn, L., "Public-key cryptosystem design based on factoring and discrete logarithms," *IEE Proc. Comput. Digit. Tech.*, Vol. 141, No. 3, pp. 193-195, 1994.

[7] Lee, N. Y., and Hwang, T., "Modified Harn signature scheme based on factoring and discrete logarithms," IEE Proc. Comput. Digit. Tech., Vol. 143, No. 3, pp. 196-198, 1996.

[8] Lee, N. Y., "Security of Shao's signature schemes based on factoring and discrete logarithms," IEE Proc. Comput. Digit. Tech., Vol. 146, No. 2, pp. 119-121, 1999.

[9] He, W.H., "Digital signature scheme based on factoring and discrete logarithms," *Electronics letters*, Vol. 37, No. 4, February pp. 220-222, 2001.

[10] Su, P. C., Lu, E. H., and Chang, H. K. C., "Cryptographic Identification of Users Based on Inter-mixed Approach," *Journal of Chung Cheng Institute of Technology*, Vol. 33, No. 1, pp.77 - 86, 2004.

[11] Michael, R., and David, S., Computers and Intractability: A Guide to the Theory of NP-Completeness. *W. H. Freeman & Co., Freeman*, New York, 1979.

[12] Shamir, A., "A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem," *IEEE Transactions on Information Theory*, Vol. IT-30, pp. 699-704, 1984.

[13] Huber, K., "Specialised attack on CHOR-RIVEST public key cryptosystem, " *Electronics Letters*, Vol. 27, Issue 23, pp. 2130-2131, 1991.

[14] Laih, C. S., Lee, J. Y., Harn, L., and Su, Y. K., "Linearly Shift Knapsack Public-Key Cryptosystem," *IEEE Journal on Selected Areas in Communications*, Vol. 7, No. 4, pp. 534-539, 1989.

[15] Smart, N., "An identity based authenticated key agreement protocol based on the Weil pairing, " *Electronics Letters,* Vol. 38, No. 13, pp. 630–632, 2002.

[16] Chen, C. W., Lai, C. L.,"An Improved Efficient Performance Design with Multiple Channels and Bandwidth Allocation Strategy for Mobile Ad-Hoc Networks," *Journal of Pervasive Computing and Communications (JPCC)*, Vol. 3, Issue 4, Jul. 2008.

[17] Chou, H. Z., Wang, S. C., Chen, I. Y., Yuan, S. Y., and Kuo, S. Y., "Randomized and Distributed Methods for Reliable Peer-to-Peer Data Communications in Wireless Ad Hoc Networks," *IET Communications*, Vol. 1, No. 5, pp. 915-923, Oct. 2007.

[18] Jeong, I. R., Katz, J., and Lee, D. H., " One-Round Protocols for Two-Party Authenticated Key Exchange, " *Applied Cryptography and Network Security*, ACNS, pp. 220-232, 2004.

[19] Miller, V., "Use of Elliptic curves in Cryptography," Advances in Cryptology-CRYPTO'85 Proceedings, Springer-Verlag, pp.417-426, 1986.
[20] Koblitz, N., "Elliptic Curves Cryptosystems," *Mathematics of computation*, Vol. 48, No. 177, pp. 203-209. 1987.
[21] Girault, M., "Self-certified public keys", *Proceedings of the Eurocrypt'91*, pp. 491-497, 1991.
[22] Shao, Z., "Self-certified signature scheme from pairings", *Journal of Systems and Software*, Vol. 80, No. 3, pp. 388-395, 2007.
[23] Pohlig, G., and Hellman, M., "An improved algorithm for computing logarithms over GF(p) and its cryptographic significance", *IEEE Transactions on Information Theory*, Vol. 24, Issue 1, pp. 106-110, 1978.
[24] Koblitz, N., Menezes, A., and Vanstone, S., "The state of elliptic curve cryptography", *Designs, Codes and Cryptography*, Vol. 19, No. 2-3, pp.173-193, 2000.
[25] Menezes, A., Okamoto, T., and Vanstone, S., "Reducing elliptic curve logarithms to in a finite field", *IEEE Transactions on Information Theory*, Vol. 39, No. 5, pp.1639-1646, 1993.
[26] Odlyzko, A.M., "Discrete logarithms in the finite fields and their cryptographic signature," *Advances in Cryptology CRYPTO'90*, Springer, 1991.
[27] Canetti, R., and Krawczyk, H., "Analysis of key-exchange protocols and their use for building secure channels," *Advances in Cryptology EUROCRYPTO'01*, LNCS 2045, pp. 453-474, 2001.

.

[30] Chen., T. S. Liu, T. P. and Chung, Y. F., "A Proxy-Protected Proxy Signature Scheme Based on Elliptic Curve Cryptosystem," *proceedings of IEEE TENCON'02*, pp 184-187, 1997.