New Authenticated Key Exchange Using Pre-shared Key Model for Ad-hoc Networks

Pin-Chang Su

Department of Information Management, College of Management, National Defense University

ABSTRACT

The paper discusses an implementation of an authenticated key exchange for Ad-hoc networks. Security issues concerning ad-hoc networks are attracting increasing attention. There are various challenges that are faced in the ad-hoc environment. These challenges are due mostly to the resource scarcity in these networks. Generally, public key infrastructures are assumed to be unavailable in ad-hoc networks. The key exchange problem for this type of network has now become important. In particular, we take advantage of elliptic curve cryptography (ECC) of the inherent efficiency and security in this type of wireless networks - without any public key infrastructure - to defend message exchange against the threat of denial of service attacks, the man-in-the-middle attacks and the replay attacks. An efficient scheme based on ECC is proposed for networks in environments without any public key infrastructure.

Keywords: Ad-hoc networks, ID-based cryptosystem, Elliptic Curve cryptography.

適用於 Ad Hoc 網路之快速交換金鑰機制設計

蘇品長

國防大學管理學院資訊管理學系

摘　要

Ad hoc 為一種無線特定的網路結構，強調的是多跳、獨立性及無控制中心等概念，由一組帶有無線收發裝置的移動終端組成的一個多跳臨時性自治系統，同時具備行動通訊和電腦網路的特點，可視為一種特殊類型的行動電腦通訊網路。近幾年的網路國際會議都有 Ad hoc 研討專題。綜觀 Ad hoc 網路所面臨到的挑戰主要為:動態的拓撲結構和組織成員、易受攻擊的無線連結、漫遊在危險的環境。安全問題一直是 Ad hoc 網路環境始終無法克服的障礙。本研究為設計以橢圓曲線系統應用於 Ad hoc 網路成員間身分識別及驗證的快速方法，適用於 Ad hoc 網路內成員間的金鑰交換協定。

關鍵字：Ad hoc 網路、身分識別、橢圓曲線系統

文稿收件日期 97.5.27; 文稿修正後接受日期 97.10.23.
Manuscript received May 27, 2008; revised October 23, 2008.

I. INTRODUCTION

Ad-hoc networks have no fixed infrastructure such as base stations for mobile switching. Nodes within each other's radio range communicate directly via wireless links. When nodes are far apart they rely on other nodes to relay messages. Ad-hoc networks present huge security risks. The major problem in providing security services involves managing cryptographic keys. There are several issues, such as routing, scalability, quality of service and security that must be solved before implementing practical security. Most of the research on ad-hoc networking has been on routing [1-3]. Other issues such as security and network addressing have received considerably less attention [4-7]. In Capkun et al. [8] explained that there are two extreme ways to introduce security in mobile ad hoc networks: (1) through a single authority domain, where certificates are issued in system setup phase or (2) through full self-organization, where security does not rely on any trusted authority, not even in the initialization phase. We assumed that all devices are resource-constrained in energy, band width, computational ability, memory, and possibly long term storage as well. Even if the self-organized protocol can be defined, maintaining and keeping it available to all users in such a dynamic network is not feasible. Recent researches have shown that wireless ad-hoc networks are highly vulnerable to various security threats due to their inherent characteristics [9, 10]. This leaves ad-hoc key management and key distribution as a wide open problem.

The Identity Based cryptosystem was first proposed by Shamir [11] to simplify the conventional public key cryptosystem, and make management easier [12]. Khalili, et al. proposed a protocol for ad hoc network management and authentication based on an ID-based scheme in [13]. The authors did not provide an actual protocol and there were many open questions for implementation. For instance, the authors did not explain how this scheme could be implemented using existing ID-based crypto systems, although this function is crucial for the system [14].

Motivated by the above discussion, we explore the possibility of designing an authenticated key exchange scheme that can be implemented in pre-shared key model. Our approach using ID-Based encryption provides end-to-end authentication and can simultaneously prevent leakage of user's identity and denial of service (DoS) attack. Of course, we will also ensure that our scheme is efficient with respect to other measures, including communication complexity and computational efficiency. Our scheme solves the ad-hoc network security problem and is suitable for application to other wired network structures.

Recently, it has been noticed that pairings can be used to build cryptosystems with certain functions [15]. The most impressive application of pairings to cryptography is the ID-based encryption scheme proposed by Boneh and Franklin [16]. So far there are many ID-based cryptographic schemes having been proposed using pairings. However, evaluating pairings is a more complex and costly operation compared to scalar multiplications of the ECC [17]. For a given security, therefore a scalar multiplication is faster than a pairing computation. Furthermore, the security of the pairing-based cryptosystems (PBC) relies mainly on the difficulty of solving the bilinear Diffie–Hellman problem. However, the hardness of solving the elliptic curve discrete logarithm problem is stronger than the one of solving the BDH problem [16]. Thus, we can say that ECC is better than PBC in performance and security considerations. Although there are many papers discussing the complexity of pairings and how to speed up the pairing computation [15], the computation of the pairing is still moretime-consuming [18]. For this reason, we construct our cryptographic schemes in this paper by employing ECC instead of PBC.

II. BACKGROUND THEORIES

In the section we will review the Diffie-Hellman problems over elliptic curve cryptosystem defined in the prime order group G. We also review Jeong et al.'s one-round protocol for authenticated key exchange method [19], which will be recommended as our proposed algorithm in Section 3.

2.1 Elliptic curve cryptography

Miller [20] and Koblitz [21] 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., $4a^3 + 27b^2 \mod q \neq 0$. 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 $\lambda = (x_2 - y_1)/(x_2, x_1) \quad If\ x_1 \neq x_2.$

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

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 One-round protocols for authenticated key exchange

Also, we review Jeong et al.'s method 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 $sid_i = h(g^{\alpha_i} \,\|\, \tau_i \,\|\, g^{\alpha_j} \,\|\, \tau_j)$.

III. OUR PROPOSED SCHEME

According to Miller [20] and Koblitz [21] schemes, we designed an efficient ID-based key exchange scheme based on ECC is proposed in this section. This protocol is suitable for wireless mobile ad-hoc network. Similar to other ID-Based cryptosystems, a trusted key generation center (KGC) is needed in this protocol for verifying user identity numbers and generating the corresponding private keys. The protocol consists of three phases to establish secured key exchange mechanism, as the follows.

3.1 Setup phase

The KGC selects the elliptic curve domain parameters are comprised of [22]:

- The field order q 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 prime.

The system parameter n is made public and p_1, p_2, p_3, p_4 are all discarded. In addition, the KGC also chooses one-way hash functions $\{H_1(), H_2(), H_3()\}$, convert

function $f()$ and computes public key Q_{KGC}, such that $Q_{KGC} = d_{KGC}P$, where d_{KGC} is the KGC's secret key.

U_i takes his identification number id_i and secret key d_i to the KGC to obtain the signature s_i of id_i. If the KGC confirms the correctness and the relationship between U_i and id_i, he then selects a random number k_i and calculates (Q_i, X_i, s_i) using

$$Q_i = d_i P, \quad (1)$$

$$X_i = k_i P = (x_i, y_i), \quad (2)$$

$$e_i = H_1(Q_i, id_i), \quad (3)$$

$$r_i = f(x_i), \quad (4)$$

$$s_i = k_i(e_i + d_{KGC} r_i)^{-1}. \quad (5)$$

Registration must be in person or using some form of secure authenticated communication. Moreover, each participant reliably knows a public key of KGC. When all the users have registered and got his (s_i, X_i, Q_i), the KGC does no need to exist in ad-hoc network any more.

3.2 Computation of the wrapper key phase

Assume U_i and U_j are the two users who want to communicate each other secretly. U_i sends (id_i, s_i, X_i, Q_i) to U_j. Similarly, U_j sends (id_j, s_j, X_j, Q_j) to U_i. Before generating the wrapper key, U_i and U_j need to verify whether (id_i, s_i, X_i, Q_i) and (id_j, s_j, X_j, Q_j) are sent from U_i and U_j, respectively, by checking

$$e_j = H_1(Q_j, id_j), \quad (6)$$

$$r_j = f(x_j), \quad (7)$$

$$\hat{X}_j = s_j(e_j P + r_j Q_{KGC}) \quad (8)$$

$$\hat{X}_j \stackrel{?}{=} X_j \quad (9)$$

U_i and U_j compute the wrapper key K_{ij} respectively, as follows

$$K_{ij} = d_i Q_j = d_j Q_i. \quad (10)$$

3.3 Authentication phase

U_i and U_j share a previously known wrapper key K_{ij}. At the stage, we use a challenge-response-type protocol here. The procedure can be described as follows.

Step 1: U_i randomly picks up a $t_i \in Z_q$, and calculates

$$T_i = t_i P, \quad (11)$$

$$V_i = K_{ij} + T_i \quad (12)$$

and then sends (id_i, V_i) to U_j.

Step 2: Upon receiving the request, U_j randomly chooses a picks up $t_j \in Z_q$, and calculates

$$T_j = t_j P, \quad (13)$$

$$V_j = K_{ij} + T_j \quad (14)$$

Meanwhile, U_j decrypts T_i with the wrapper key as $\hat{T}_i = V_i - K_{ij}$. Considering T_i could have been modified or replaced by a third party attacker, we mark the decrypted T_i as $\hat{T}_i$. With $\hat{T}_i$ obtained, U_j is able to get the Diffie-Hellman key as $W_j = t_j \hat{T}_i$. Besides the session key, U_j also need to get $Auth(B)$ and $Auth(A)^*$ for authentication purpose, where

$$Auth(B) = H_2(id_i, id_j, W_j), \quad (15)$$

$$Auth(A)^* = H_2(id_i, id_j, Z_j), \quad (16)$$

$$where Z_j = W_j + K_{ij}. \quad (17)$$

Then U_j sends $(id_j, V_j, Auth(B))$ to U_i.

Step 3: After receiving the request, U_i virtually has got all he needs to generate the session key. He then first checks whether the $Auth(B)$ from U_j matches $Auth(B)^*$ by calculating

$$\hat{T}_j = V_j - K_{ij} \quad (18)$$

$$\hat{W}_j = t_i \hat{T}_j, \quad (19)$$

$$Auth(B)^* = H_2(id_i, id_j, \hat{W}_j). \quad (20)$$

If so, he continues to calculate the session key and $Auth(A)$ as

$$Z_i = \hat{W}_j + K_{ij}, \quad (21)$$

$$Auth(A) = H_2(id_i, id_j, Z_i) \quad (22)$$

Otherwise, U_j cannot be authenticated, and handshake fails. At last, U_i returns U_j the $Auth(A)$ for authenticating himself. If $Auth(A)$ matches $Auth(A)^*$, the whole process is done.

3.4 Computation of the session key phase

U_i and U_j compute session keys SK_i, SK_j respectively, as follows:

$$SK_i = H_3(id_i, id_j, T_i, \hat{T}_j, \hat{W}_j) \quad (23)$$

$$SK_j = H_3(id_i, id_j, \hat{T}_i, T_j, W_j). \quad (24)$$

Note that achieving explicit authentication is not possible using a one-round in the setting we consider, since the message of either party can always be replayed. Of course, it is well known how to convert any protocol providing implicit authentication to one providing explicit authentication, but only at the expense of additional rounds of communication.

IV. EVALUATION OF OUR SCHEME

This subsection discusses the security of the proposed schemes. 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 [19]. We consider the following scenarios as interrelated types of queries in this scheme.

- The query *Initiate*(*i* ,*j*) is used to "prompt" U_i to initiate execution of the protocol with U_j. This query will result in U_i sending a message, which is given to the adversary.
- A query *Send*(*i, k*(*M*)) is used to send a message *M* to instance Π_i^k (represents the k-th instance of player U_i), this models active attacks by the adversary). When Π_i^k receives *M*, it responds according to the key-exchange protocol.
- A query *Execute*(*i, j*) represents a passive eavesdropping by the adversary on a protocol execution by parties U_i and U_j. In response to this query, U_i and U_j execute the protocol without any interference from the adversary, and the adversary is given the resulting transcript of the execution.
- A query *Reveal*(*i, k*) models known key attacks in the real system. The adversary is given the session key sk_i^k for the specified instance.
- A query *Corrupt*(*i*) models exposure of the long-term key held by U_i. The adversary is assumed to be able to obtain long-term player keys, but cannot control the behavior of these players directly.
- A query *Test*(*i, k*) is used to define the advantage of an adversary. When an adversary *Adv* asks a Test query to an instance sk_i^k, a coin b is flipped. If b is 1, the session key sk_i^k is then returned. Otherwise, a random session key is returned. The adversary is allowed to make a single Test query to a fresh instance at any time during the experiment.

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 protocols. In follow, we will analyze the security protocol for ECDLP and the key exchang model.

4.1 Security proof of protocol

We first analyze our one-round protocol, O*R*, which provides key independence and whose security is based on the computational Diffie-Hellman (CDH) assumption in the random oracle model. In Table 4 we compare our scheme to the scheme of Jeong et al. [19] which achieves the same level of security in the same number of rounds. Our scheme is more efficient than the scheme of Jeong et al. and has other advantages as well: our protocol is simpler and is also symmetric with respect to the two parties. The following theorem states the security achieved by this protocol.

Theorem: Assuming G satisfies the CDH assumption [19], our method, O*R*, is a KI-secure key-exchange protocol when H is modeled as a random oracle. Concretely, if G is generated by Δ which is (t, ε)-secure with respect to the CDH problem, then

$$\mathrm{Adv}_{OR}^{\mathrm{KI}}(\mathrm{k}, \mathrm{t}, \mathrm{q}_{\mathrm{re}}, \mathrm{q}_{\mathrm{h}}) \le q_h N^2 \cdot \varepsilon + \frac{q_s^2}{2^k}, \quad (25)$$

where t is the maximum experiment time including the adversary's execution time, and the adversary makes q_{re} Reveal queries and q_h hash queries. Here, N is an upper bound on the number of parties, and q_s is an upper bound on the number of instances initiated in the experiment.

Proof: Consider an adversary *Adv* attacking

OR in the sense of key independence. Informally, and using the fact that we work in the random oracle model, there are only two ways an adversary can get information about a particular session key

$$SK_i^k = H(id_i.id_j, T_i^k, \hat{T}_j^k, \hat{W}_j^k),$$

and $SK_j^k = H(id_i, .id_j, \hat{T}_i^k, T_j^k, W_j^k)$,

for a fresh instance: either the adversary queries the random oracle on the value $(id_i.id_j, T_i^k, T_j^k, W_j^k)$ at some value during the experiment. The latter case happens with probability upper-bounded by $q_s^2 / 2^k$, while the former case allows us to solve the CDH problem with probability related to that of the adversary's success probability. We now proceed with a more formal proof [19].

Let *coll* denote the event that a value of SK repeats at some point during the experiment, and let query be the event that, for some $i, j \in [N]$, the adversary at some point makes an oracle query

$SK_i^k = H(id_i.id_j, T_i^k, \hat{T}_j^k, \hat{W}_j^k)$,

$SK_j^k = H(id_i, .id_j, \hat{T}_i^k, T_j^k, W_j^k)$. Observe that $\Pr_{Adv}[b' = b \mid \overline{query} \wedge \overline{coll}] = \frac{1}{2}$, we may write

$$\begin{aligned}
&\Pr_{Adv}[b' = b] - \frac{1}{2} \\
&= \Pr_{Adv}[b' = b \wedge coll] \\
&\quad + \Pr_{Adv}[b' = b \wedge query \wedge \overline{coll}] \\
&\quad + \Pr_{Adv}[b' = b \wedge \overline{query} \wedge \overline{coll}] - \frac{1}{2} \\
&= \Pr_{Adv}[b' = b \mid coll] \cdot \Pr_{Adv}[coll] \\
&\quad + \Pr_{Adv}[b' = b \mid query \wedge \overline{coll}] \cdot \Pr_{Adv}[query \wedge \overline{coll}] \\
&\quad + \frac{1}{2} \cdot \Pr_{Adv}[\overline{query} \wedge \overline{coll}] - \frac{1}{2} \\
&= \Pr_{Adv}[b' = b \mid coll] \cdot \Pr_{Adv}[coll] \\
&\quad + \Pr_{Adv}[b' = b \mid query \wedge \overline{coll}] \cdot \Pr_{Adv}[query \wedge \overline{coll}] \\
&\quad + \frac{1}{2} \cdot (1 - \Pr_{Adv}[coll] - \Pr_{Adv}[query \wedge \overline{coll}] - \frac{1}{2} \\
&\le \frac{1}{2} \cdot (\Pr_{Adv}[query \wedge \overline{coll}] + \Pr_{Adv}[coll]). \qquad (26)
\end{aligned}$$

Now, as noted previously, $\Pr_{Adv}[coll]$ is bounded from above by $q_s^2 / 2^k$ by a "birthday problem" calculation, since for this event to occur two random nonces of length k generated by some players in separate instances must repeat.

Consider the following algorithm F which attempts to solve the CDH problem using Adv as a subroutine. F is given, an instance of the CDH problem. F randomly selects two parties and uses U_i and U_j as their public keys. F simulates the random oracle H, and tries to find SK_{ij} from the hash queries made by Adv.

The probability that F returns the correct answer is at least $\Pr_{Adv}[query \wedge \overline{coll}] / N^2 q_h$, since the simulation is perfect until the point, if any, that query occurs. Furthermore, since the running time of F is essentially the same as the running time of Adv we must have $\Pr_{Adv}[query \wedge \overline{coll}] \le q_h N^2 \varepsilon$. Plugging this into Eq.(26) gives the result of the theorem.

4.2 Security consideration of protocol

This study proposes a pre-shared key authentication for key exchange in Ad-hoc networks protocol that fully exploits the difficulty of the ECDLP, with a difficult-to-discover trapdoor. The E is taken over a finite field K $\geqq F_q$. 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 ECDLP 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 are required to keep the methods safe. Discrete logs in ordinary number groups, like Z_p^*, are now much easier to solve than those in elliptic curve groups. According to Reference [24], the solution of the discrete logarithm requires

$O(\exp(cons\,t.(\log q \log\log q)^{1/2}))$ integer multiplication. The advantage of ECC derives from the existence of sub-exponential algorithms of complexity $O(\exp(const.(\log q)^{1/3} (\log\log q)^{2/3}))$ exist that solve the DLP over F_q [25].

4.2.1 The wrapper key attack

An adversary attempts an attack by revealing the wrapper key from $U_i's$ public key. To derive (d_i, k_i) from (id_i, s_i, X_i, Q_i), the adversary tries to reveal the message from $U_i's$ public key. He must solve the elliptic curve discrete logarithm problem.

4.2.2 The private key attack

If an attacker attempts an attack by revealing the private key d_{KGC} from the public key of the KGC, then he can obtain k_i by computing $s_i = k_i(e_i + d_{KGC} r_i)^{-1}$; thus he can play the role of U_i to forge. However, the attacker must solve the elliptic curve discrete logarithm problem given by Q_{KGC} to determine d_{KGC}.

4.2.3 The man-in-the-middle attack to wrapper key

When U_i sending (id_i, s_i, X_i, Q_i) to U_j, an adversary can intercept the datum from the public channel; then plays the role of U_i to cheat U_j or other users. The cheat does not pass the verification of $\hat{X}_j \stackrel{?}{=} X_j$ because both the identification information id_j are inputs of the one-way function $H_1()$ and used in the operation $\hat{X}_j = s_j(e_j P + r_j Q_{KGC})$.

4.2.4 The man-in-the-middle attack to session key

Based on the Diffie-Hellman key agreement protocol, our method uses a wrapper key as authentication information, encryption seed, and a factor for derivation of the final session key. With wrapper key protection, it prevents the key exchange process from the man-in-the-middle attack, to which Diffie-Hellman protocol alone is vulnerable.

4.2.5 The DoS attack

In proposed key exchange, an initiator can confirm that the intended recipient of the shared key actually processes the shared key by verifying second message, but the responder cannot. Our method can realize initiator's key confirmation. DoS attack can be prevented because of wrapper key properties, that is, an attacker cannot guess valid wrapper key which is need to initiate key exchange.

4.3 Performance analyses

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 - the elliptic curve discrete logarithm problem (ECDLP) - 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 [26] 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_{EXP} +1 T_{ADD} + 2 T_{MUL}+ 2 hash	299 T_{MUL}+ 2 hash
One-round authentication	Computation of the wrapper key	2 T_{EXP} + 1 hash	480 T_{MUL} + 1 hash	3 T_{EC_MUL} + 1 T_{EC_ADD} + 2 hash	87.12 T_{MUL}+ 2 hash
	Authentication			3 T_{EC_MUL} + 5 T_{EC_ADD} + 4 hash	87.6 T_{MUL}+ 4 hash
Computation of the session key		1 T_{EXP} + 1 hash	240 T_{MUL} + 1 hash	1 hash	1 hash

Remark: The security analysis of Jeong et al. on their cryptosystem [19] is too simple. The security of the cryptosystem against the man-in-the-middle attacks, the malicious KGC attacks, the replay attack, and the denial-of-service attacks needs to be carefully investigated.

V. CONCLUSIONS

Until now, public key cryptosystems maintained high security for many communication mechanisms. However, in general, no public key infrastructure is available for ad-hoc networks such that public key exchanges in the network have become a very important problem. This paper discussed a new scheme for identifying a user joining an ad-hoc network based on elliptic curve cryptography. The proposed scheme has three notable advantages; (1) the construction of a fast and extremely secure user identification system; (2) the scheme does not need an on-line KGC to share our key; (3) the scheme verifies two participants in pre-shared key model and uses smaller certificate sizes than other existing public key schemes.

REFERENCES

[1] Johnson, D. B., Maltz, D. A., and Hu, Y. C., "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks(DSR)," IETF MANET Working Group Internet-Draft, available at http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt, July 19, 2004.

[2] Jacquet, P., Muhlethaler, P., Clausen, T. A., Laouiti, A. Q., and Viennot, L., "Optimized Link State Routing Protocol for Ad Hoc Networks," Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century, pp. 62-68, 2001.

[3] Milanovic, N., Malek, M., Davidson, A., and Milutinovic, V., "Routing and Security in Mobile Ad Hoc Networks," IEEE Computer, Vol. 37, No. 2, pp. 61-65, 2004.

[4] Zhou, L., and Haas, Z. J., "Securing Ad Hoc Networks," IEEE Network Journal, 1999, Vol. 13, No. 6, pp. 24-30, 1999.

[5] Weimerskirch, A., and Westhoff, D., "Identity Certified Authentication for Ad-hoc Networks," 2003 ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '03), October 31, 2003.

[6] Chen, C. W., and 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, 2008.

[7] 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, 2007.

[8] Capkun, S., Buttyan, L.,and Hubaux, J. P., "Self-Organized Public-Key Management for Mobile Ad Hoc Networks," IEEE Transactions on Mobile Computing, Vol. 2, No. 1, pp. 52-64, 2003.

[9] Kong, J., Zerfos, P., Luo, H., Lu, S.,and Zhang, L., "Providing Robust and Ubiquitous Security Support for Mobile Ad-Noc Networks," Proceedings of the IEEE 9th International

Conference on Network Protocols (ICNP'01), IEEE Computer Society, pp. 251-260, 2001.

[10] Deng, H., Mukherjee, A., and Agrawal, D. P., "Threshold and Identity-based Key Management and Authentication for Wireless Ad Hoc Networks," Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), IEEE Computer Society, pp. 107-111, 2004.

[11] Shamir, A., "Identity-based cryptosystems and signature schemes," Advances in Cryptology – Crypto '84, Lecture Notes in Computer Science 196, Springer-Verlag, pp. 47-53, 1985.

[12] Bohio, M., and Miri, A., "An Authenticated Broadcasting Scheme for Wireless Ad Hoc Network," Proceedings of the Second Annual Conference on Communication Networks and Services Research (CNSR'04), IEEE Computer Society, pp. 69-74, 2004.

[13] Khalili, A., Katz, J., and Arbaugh, W., "Toward Secure Key Distribution in Truly Ad-Hoc Networks," 2003 Symposium on Applications and the Internet Workshops (SAINT 2003), IEEE Computer Society, pp. 342-346, 2003.

[14] Hoeper, K., and Gong, G., "Models of Authentications in Ad Hoc Networks and Their Related Network Properties," CACR, available at, 2004.

[15] Barreto, P., Kim, H., Lynn, B., and Scott, M., "Efficient algorithms for pairing-based cryptosystems," Advances in Cryptology, Proceedings of Crypto_2002LNCS, Springer-Verlag, New York, pp. 354-368, 2002.

[16] Boneh, D., and Franklin, M., "Identity-based encryption from the Weil pairing", Advances in Cryptology," Proceedings of Crypto_2001LNCS, Springer-Verlag, New York, pp. 213-229, 2001.

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

[18] Tsaur, W.J., "Several security schemes constructed using ECC-based self-certified public key cryptosystems," Applied Mathematics and Computation, Vol.168, pp. 447-464, 2005.

[19] 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.

[20] Miller, V.,"Use of Elliptic curves in Cryptography," Advances in Cryptology-CRYPTO'85 Proceedings, Springer-Verlag, pp. 417-426, 1986.

[21] Kobiltz, N., "Elliptic Curve Cryptosystems," Mathematics of Comutation, pp. 203-209, 1987.

[22] Hankerson, D., Menezes, A., and Vanstone, S., "Guide to Elliptic Curve Cryptography," Springer-Verlage, New York, Inc. 2004.

[23] Boneh, D., Lynn, B., and Shacham, H., "Short signatures from the Weil pairing," Advances in Cryptology-Asiacrypt 2001, Springer-Verlag, pp. 514-532, 2001.

[24] 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, No. 1, pp. 106-110, 1978.

[25] Odlyzko, A. M., "Discrete logarithms in the finite fields and their cryptographic signature," Advances in Cryptology CRYPTO'90, Springer-Verlag, pp. 224-316, 1991.

[26] 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.