A New Authentication Scheme for Protecting Mobile Agent Platforms Using Pairing-Based Cryptosystems

Woei-Jiunn Tsaur, Chien-Hao Ho and Chii-Jyh Guo
Department of Information Management
Da-Yeh University, Changhwa, Taiwan, R.O.C.
E-mail: wjtsaur@mail.dyu.edu.tw

Abstract

This paper presents an authentication scheme for protecting mobile agent platforms against unauthorized mobile agents using the proposed self-certified pairing-based public key cryptosystem. The proposed cryptosystem is constructed using the pairing-based cryptosystems, and is developed by integrating the identity-based public key cryptosystems with the self-certified public key cryptosystems to provide higher security strength. In addition, we employ the integrated cryptosystems to design an authentication scheme so that a mobile agent can register once to a system authority for many services in the mobile agent based networks. As far as agent platform security is concerned, we apply the idea of proxy signature to construct the proposed authentication scheme to protect mobile agent platforms. In summary, this platform protection scheme is based on the proposed cryptosystem to accomplish the requirements of authentication and authorization of mobile agents in a new way by proxy signature.

Key Words: mobile agent, authentication scheme, proxy signature, self-certified public key cryptosystems, pairing-based cryptosystems

1. Introduction

Mobile agents are one of the fastest growing areas of information technology. Currently, mo-bile agents are employed in an increasing wide variety of applications. Mobile agent technology offers a new computing paradigm in which a program can suspend its execution on a host platform, transfer itself to another agent-enabled platform on the network, and resume execution on the new host platform [20]. Therefore, the ability to remote execution across the nodes of a wide area network (WAN) allows deployment of e-commerce services and applications in a more dynamic, flexible, and customizable way. Al-though the mobile agent paradigm extends the capabilities of remote communication and distributed computing, it also raises new security issues [4]. The capability of mobility may lead to a great deal of security threats and attacks. So far, the research on mobile agent security is generally divided into two broad areas [12]: 1. Protecting the platforms against unauthorized and hostile agents under execution, 2. Protecting the agents against tampering attempts by the malicious host platforms. In the past few years, a lot of researchers devoted to solve the latter problem to protect the mobile agents, such as [17]. However, the problem to protect mobile agent platforms against malicious mobile agents is also complicated and cannot be omitted.

As far as server security is concerned, a major problem specific to mobile agents is the protection of the agent platforms running the agents. A hostile agent could destroy the hard drive, steal data, or do all sorts of other undesirable things. In this paper, we consider security schemes based on the cryptographic solution for prevention of the malicious agents. We employ a proxy signature scheme to accomplish authentication for protecting mobile agent platforms. In other words, in this scheme a mobile agent can register once to a system authority for many ser-vices in the mobile agent based networks. Moreover, we propose a secure pairing-based public key cryptosystem by integrating the identity-based and the self-certified public key cryptosystems to provide higher security strength, and we also employ the integrated cryptosystems to design the authentication scheme for protecting mobile agent platforms.

2. Previous Works

2.1 Public Key Cryptosystems

Public key cryptosystems are primary basics for the realization of contemporary encryption or digital signature schemes, where one secret key is used as the decryption key or signature generation key and the corresponding public key is used as the ciphertext generation key or signature verification key. In public key cryptosystems, any public key should be verified before using it for subsequent cryptographic applications. Two most widely adopted approaches for public key verification are named as the certificate-based and identity-based (ID-based) public key cryptosystems [18], respectively. The certificate-based approach requires an extra public key certificate issued by the system authority (SA) after user registration. The ID-based approach regards the user's identity as his/her public key, and hence no extra public key certificate is required. In the certificate-based approach, anyone that wants to use a public key for certain subsequent cryptographic application (e.g., key exchange or signature verification) should independently perform public key verification and subsequent cryptographic application through two separate steps. As to the ID-based approach, it usually requires an interactive identification protocol for authenticating the user's identity (i.e., the public key) before proceeding certain cryptographic application. Although both the certificate-based and the ID-based approaches effectively solve the problem of public key verification for practical usage, they bring out another security leak that SA knows all users' secret keys after user registration. Therefore, SA may have the opportunity to masquerade as any legitimate user by generating a valid public-key/secret-key pair for that user without being detected.

In 1991, Girault [9] proposed a self-certified public key cryptosystem, which is intermediary between certification-based and identity-based ones, to resolve the problem of public key verification. A self-certified public key system has three features. First, the secret key could be determined by the user himself/herself or together by the user and SA, and does not be known to SA. Second, the user can use his/her own secret key to verify the authenticity of the self-certified public key issued by SA, and hence 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 earns more efficiency in saving the communicational cost and the computational effort as compared to that of the certificate-based and the ID-based approaches. Recently, Saeednia [15] successfully amalgamated the merits inherent in both the ID-based and the self-certified systems, and proposed an ID-based self-certified public key system that can be applied to the realization of key exchange protocols. However, Wu et al. [19] and Kim et al. [11] showed that the original version of Saeednia's ID-based self-certified public key system is not secure enough to withstand the impersonate attack. They also proposed an improvement to overcome the flaw inherent in the original version, respectively.

2.2 Bilinear Pairings

The use of the Weil pairing and Tate pairing in cryptography goes back to the results of MOV attack [13] and Frey-Rück attack (FR attack) [6]. However, these first applications were destructive such as using pairings to transform the ECDLP into a discrete logarithm problem in the multiplicative group of a finite field.

More recently it has been noticed that pairings can be used to build cryptosystems with certain functionality [2, 3]. The foundational paper is [10] proposed by Antoine Joux, in this paper he proposed a one-round protocol for tripartite Diffie-Hellman. In fact, earlier work suggesting the use of pairings in cryptography was done by Sakai *et al.* in 2000 [16]. In particular, the paper of Sakai *et al.* suggests that pairings could be used to enable identity-based cryptography.

The most impressive application of pairings to cryptography is the identity-based encryption scheme of Boneh and Franklin [2]. This system elegantly solves the long-standing open problem of providing secure and efficient identity-based encryption [18]. So far, there are many kinds of ID-based cryptosystems based on bilinear pairings, but ID-based public key cryptosystems have some drawbacks as we have described in section 2.1. These problems still exist in such pairing-based cryptosystems. Therefore, in this paper we will design a self-certified pairing-based cryptosystems to achieve higher security level. We will develop our new secure public key cryptosystems based on bilinear pairings to construct our related security schemes for mobile agent based networks.

In the following, we briefly describe the basic definition and properties of the Weil pairings and the CDH (Computational Diffie-Hellman) assumption.

[The Weil Pairings]

Let G1 be a cyclic additive group generated by P, whose order is a prime q, and G2 be a cyclic multiplicative group of the same order q. We assume that the discrete logarithm problems (DLP) in both G1 and G2 are hard. Let e: G1 × G1 $\rightarrow$ G2 be a pairing which satisfies the following conditions [1, 5 , 8, 10]:

1. *Bilinear*:

$e(P_1+P_2, Q) = e(P_1, Q)e(P_2, Q)$,

$e(Q, P_1+P_2,)= e(Q, P_1)e(Q, P_2)$, and

$e(aP, bQ) = e(P, Q)^{ab}$

2. *Non-degenerate*:

There exists $P \in G_1$ and $Q \in G_1$ such that $e(P, Q) \neq 1$;

There exists $P \in G_1$ and $Q \in O$ such that $e(P, Q) = 1$, (O is a point at infinity)

3. *Computability*:

There is an efficient algorithm to compute $e(P, Q)$ for all $P, Q \in G_1$.

Note that the Weil and Tate pairings associated with supersingular elliptic curves or abelian varieties can be modified to create such bilinear maps.

Definition 1

The Computational Diffie-Hellman (CDH) problem for a bilinear pairing $e: G_1 \times G_1 \rightarrow G_2$ is defined as follows:

Given P, aP, bP $\in G_1$*, compute abP* $\in G_1$*, where a, b are randomly chosen from* Z_q^**. An algorithm is said to solve the CDH problem.*

It is widely believed that if the group G_1 comes from an appropriately chosen elliptic curve and if q (the size of G_1) is of the order of 2^{160}, then these problems are computationally infeasible [2]. In this paper, we will construct our security schemes based on the CHD assumption. The identity-based approaches can reduce the computation cost greatly, and the bilinear pairings can be used as a building block of the identity-based schemes [2]. Thus, we use the bilinear pairings on elliptic curves to construct our related security schemes. Furthermore, the self-certified cryptographic schemes may further provide the higher security level [9, 14, 15]. Therefore, in the paper we will design a public key cryptosystem based on the pairing-based cryptosystems [2, 3] and equipped with identity-based [18] and self-certified public key cryptosystems to construct mobile agent related security scheme.

3. Authentication Scheme for Protecting Mobile Agent Platforms

Based on the proxy signature , the proposed scheme can be used for protecting platforms against unauthorized mobile agents. In the following, we will describe the proposed scheme in details.

3.1 Initialization

The entities in the system are a system authority (SA), users (Ui), host platforms (Hi), and a mobile agent (MA) generated by a specific user. We assume that the system authority SA is responsible for a key generation center and a registration center. First, We define notations used in the proposed schemes as follows:

E(F_{3^m}): a supersingular elliptic curve E: y2 = x3 – x + 1 (mod 3^m), where the characteristic is 3, and the security multiplier is 6.

G: an additive group of the elliptic curve E whose order is a large prime q. We also write G* $\equiv$ G – {O}, and O is the point at infinity.

B: a base point of G whose order is q.

V: a multiplicative group of order q on the elliptic curve E.

ê: a bilinear pairing map where $\hat{e}: G \times G \rightarrow V$.

H1: a one-way hash function denoted by $H_1: \{0, 1\}^* \rightarrow G^*$, which means that the input is a string {0, 1}* and the output is a point G*.

H2: a one-way hash function, where $H_2: \{0, 1\}^* \rightarrow Z_q^*$.

H3: a one-way hash function $H_3: V \rightarrow \{0, 1\}^n$, where n $\in$ N denotes the size of message.

3.2 The Proposed Public Key Cryptosystems

The algorithms of the proposed public key cryptosystems are divided into two phases: system setup and key generation.

[System Setup]

SA creates a system public key and some public parameters in this phase, and then *SA* releases these parameters. *SA* randomly chooses a number $S_{SA} \in Z_q^*$ and keeps it secret. Then *SA* computes the system public key $P_{SA} = S_{SA} \cdot B$. Therefore, the public parameters in the system are < *E*, *q*, *G*, *V*, *ê*, *B*, P_{SA}, H_1, H_2, H_3 >, and S_{SA} is *SA*'s private key.

[Key Generation]

User U_i and host platform H_i perform the following steps to register to *SA*, and obtain the corresponding public key, respectively. They also compute their private keys in this phase.

Step 1. U_i and H_i choose a random number $k_i \in Z_q^*$, respectively. Then they compute $K_i = k_i \cdot B$, and transmit their own K_i and identity $ID_i \in \{0, 1\}^*$ to the *SA*.

Step 2. After receiving ID_i and K_i, *SA* calculates $I_i = H_1(ID_i) \in G^*$, and randomly chooses an integer $x_i \in Z_q^*$ to compute $Q_i = x_i \cdot B$. Then *SA* generates each participant's public key $P_i = K_i + Q_i$ and the witness of the public key $W_i = S_{SA}(P_i + I_i) + x_i \cdot P_{SA}$.

Finally, SA sends $\{P_i, W_i\}$ to the participant.

Step 3. Upon receiving $\{P_i, W_i\}$, the participant calculates his/her own private key $S_i = W_i + k_i \cdot P_{SA}$, and he/she can verify the public key by performing the following formula:

$$e(S_i, B) = e(2P_i, P_{SA}) \cdot e(I_i, P_{SA}) \qquad (1)$$

If the result is correct, then the participant's private key is S_i; otherwise, it means that the public key P_i is altered in the transmission.

Theorem 3.1 User U_i and host platform H_i can utilize the formula (1) to verify his/her public key P_i by himself/herself.

Proof:

$$\begin{aligned} e(S_i, B) &= e(W_i + k_i P_{SA}, B) \\ &= e(s_{SA}(P_i + I_i) + x_i P_{SA} + k_i P_{SA}, B) \\ &= e(s_{SA}P_i, B)\ e(s_{SA}I_i, B)\ e((x_i + k_i)P_{SA}, B) \\ &= e(P_i, P_{SA})\ e(I_i, P_{SA})\ e((x_i + k_i)B, P_{SA}) \\ &= e(P_i + I_i, P_{SA})\ e(Q_i + K_i, P_{SA}) \\ &= e(P_i + I_i + Q_i + K_i, P_{SA}) \\ &= e(P_i + I_i + P_i, P_{SA}) \\ &= e(2P_i + I_i, P_{SA}) \\ &= e(2P_i, P_{SA})\ e(I_i, P_{SA}) \end{aligned}$$

Q.E.D.

When U_i and H_i receive $\{P_i, W_i\}$, they can perform the equation $e(S_i, B) = e(2P_i, P_{SA}) \cdot e(I_i, P_{SA})$ to verify the public key and the witness. Because of using self-certified public key cryptosystems, we do not need the certificate in our schemes. Moreover, we can attach the user's identity ID_i and the public key P_i to the mobile agent. Generally, the size of a certificate is much larger than an identity or a public key. Thus, our scheme can be better than certificate-based schemes and also be efficiently used for securing the mobile agents.

3.3 The Proposed Authentication Scheme

In this paper, we apply the concept of proxy signature to construct the proposed scheme, which possesses the merits that users can register once in the system authority SA for obtaining many services in different servers. We will give further details about how to achieve this.

Suppose there are n hosts (platforms) $H = \{H_1, H_2, H_3, \dots, H_n\}$, and a system authority SA in the mobile agent based networks. The SA is responsible for authenticating the users and hosts upon their registrations, and generating the corresponding public and private keys. Furthermore, the SA is also responsible for issuing the corresponding proxy keys to users. Thus, users can use the proxy keys to sign his/her request carried by the mobile agents. Figure 1 illustrates the authentication model for protecting the mobile agents.

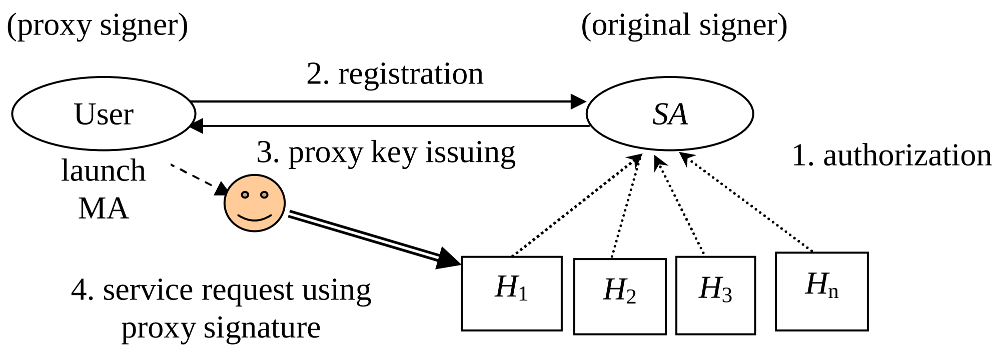

Figure 1. An authentication scheme for protecting mobile agent platforms.

The proposed scheme contains three phases: authorization phase, registration phase, and request and verification phase. We describe them in the following.

[Notations]

m_w: a warrant for a user's mobile agent, which specifies the user's identity (ID_U), mobile agent's identity, user's public key, the valid terms, routing list and the subset of servers that are authorized to the user.

Req: the login request of a user.

[Authorization Phase]

In this phase, the hosts authorize the SA to authenticate the users' mobile agents and issue the corresponding warrants and proxy keys to users, upon users' registration.

[Registration Phase]

In this phase, a user can register only once to SA for all services granted from multiple hosts. When a user U_i wants to apply for some services of the domain by using the mobile agents, he/she registers himself/herself to SA. If the user authentication succeeds, the SA prepares a warrant

m_w to the him/her. Then SA do the following steps:

Step 1. *SA* chooses a random number $r \in Z_q^*$, and computes $R' = r.\ B$ and $R = \hat{e}(P_{SA}, R')$.

Step 2. *SA* computes $M = \mathrm{H}_1(m_w) \in G^*$

Step 3. *SA* uses his/her own private key S_{SA} to calculate

$$w' = r.\ P_{SA} + S_{SA}.\ M.$$

Step 4. *SA* sends (R, w', m_w) to the user U_i, where w' should be secretly sent in a secure manner.

Step 5. Upon receiving the data, U_i computes

$$w = w' + S_U$$

Step 6. U_i computes $T_U = \hat{e}(2P_U, P_{SA}).\ \hat{e}(I_U, P_{SA})$ and verifies this proxy key by checking whether the formula (2) holds.

$$\hat{e}(w, B) = \hat{e}(M, P_{SA}).\ T_U.\ R \qquad (2).$$

[Request and Verification Phase]

After acquiring the proxy key, the user U_i can use the proxy key to sign login request as follows. Then, he/she attaches the signed request to the mobile agent.

Step 1. The user U_i prepares the request *req* that contains the identity of hosts and the current timestamp.

Step 2. The user U_i randomly chooses $c \in Z_q^*$, and then he/she computes $C = \hat{e}(P_{SA}, c.\ B)$ and $a = \mathrm{H}_2(C||req)$

Step 3. Then U_i calculates $T = c.\ P_{SA} + w.\ a$. Thus, the login request message is $\{R, T, C, ID_U, m_w, req\}$, and then U_i attaches the login request message to the mobile agent and lunches it.

Step 4. Upon receiving the request message, the host platform H_i checks the validity of the timestamp and the warrant, and then verifies the following verification equation, where the user's public key P_U and identity ID_U are specified in the warrant.

$$\hat{e}(T, B) = \hat{e}(M, P_{SA})^a.\ T_U{}^a.\ R^a.\ C \qquad (3)$$

If the host's identity is included in the warrant m_w, the timestamp is valid, and the verification of formula (3) succeeds, then the host accepts the request.

Theorem 3.2 User U_i can verify the authorized proxy key by checking whether formula (2) holds.

Proof:

$$\begin{aligned}
&\hat{e}(w, B)\\
&= \hat{e}(w' + S_U, B)\\
&= \hat{e}(rP_{SA} + S_{SA} M + S_U, B)\\
&= \hat{e}(P_{SA}, R')\ \hat{e}(M, P_{SA})\ \hat{e}(S_U, B)\\
&= \hat{e}(P_{SA}, R')\ \hat{e}(M, P_{SA}).\ T_U\\
&= \hat{e}(M, P_{SA}).\ T_U.\ R
\end{aligned}$$

Q.E.D.

According to Theorem 3.2, the user U_i can only register once at the *SA* for all services by using the proxy key, where the proxy key is authorized by the host platforms of the domain.

Theorem 3.3 The host platform can verify the authenticity of the mobile agent by checking the formula (3) whether holds.

Proof:

$$\begin{aligned}
&\hat{e}(T, B) = \hat{e}(c.\ P_{SA} + w.\ a, B)\\
&= \hat{e}(P_{SA}, c.\ B)\ \hat{e}(w, B)^a\\
&= \hat{e}(P_{SA}, c.\ B)\ \hat{e}(M, P_{SA}).\ T_U.\ R\\
&= \hat{e}(M, P_{SA})^a.\ T_U{}^a.\ R^a.\ C
\end{aligned}$$

Q.E.D.

According to Theorem 3.3, users can utilize the proxy keys to sign their requests, and then attach to the mobile agents to provide a secure authentication in the mobile agent based networks.

4. Security Analyses

The security of the proposed schemes is primarily relied on the difficulties of solving the computational Diffie-Hellman problem (CDHP), elliptic curve discrete logarithm problem (ECDLP), and one-way hash function (OWHF). The security analyses of the proposed security schemes for mobile agents are discussed in the following.

4.1 Security Consideration for Pairings

In most applications of the bilinear pairings to cryptography, we consider that the appropriate elliptic curve E over F_q is defined as follows:

Order and finite field:

The number of points is divisible by some prime l, and the finite field F_{q^k} is defined by k is the smallest integer such that $l|(q^k-1)$. Moreover, it is necessary that l has at least 160 bits for security, and for efficiency it is desired that l and q not be too large [8]. It is also necessary that q^k has at least 1000 bits for security, and not too big for efficiency.

The appropriate elliptic curves:

There are three cases elliptic curves most relevant for cryptography in pairing-based cryptosystems [7, 8]:

1. Supersingular elliptic curves such as $y^2 = x^3 + 1$ over certain prime fields F_p where p has 512 bits (in this case $k = 2$).
2. Supersingular elliptic curves of the form $y^2 + y = x^3 + x + b$, where $b \in \{0, 1\}$ over F_2 considered as a group over $F_2{}^m$ where m is prime of size around 250 bits (in this case that $k = 4$).
3. Supersingular elliptic curves of the form $y^2 = x^3 - x \pm 1$ over F_3 considered as a group over $F_3{}^m$ where m is prime of size around 110 bits (in this case $k = 6$).

Moreover, it is often the case that curves over fields of characteristic 3 are used to achieve the best possible ratio between security level and space requirements for supersingular curves [1]. Thus, in our proposed PKC we employ the supersingular elliptic curves over fields of characteristic 3 to construct related security schemes

4.2 Security of the Proposed PKC

In the proposed PKC, the security keys include *SA*'s private key, users' master keys, and users' derived private keys. The security analyses of the proposed PKC are discussed as follows:

Theorem 4.1. *Revealing SA's private key and any registering users' master key is infeasible.*

Proof:

Attackers can obtain the parameters P_i and I_i through a public channel. Since $2P_i + I_i$ (=aB) and P_{SA} (= $s_{SA}\,B$), attackers cannot get S_i (= $a\,s_{SA}\,B$) due to the difficulty of solving the CDHP. Furthermore, since $W_i = s_{SA}\,(P_i + I_i) + x_i \cdot P_{SA}$ is cooperatively generated by user U_i and the authority, attackers cannot get x_i owing to the difficulty of solving the ECDLP. Attackers cannot also get s_{SA} due to the difficulty of solving the ECDLP. Therefore, attackers cannot get *SA*'s private key and any registering user's master key.

Theorem 4.2. *Revealing any registering user's derived private key is infeasible.*

Proof:

Without knowing k_i, *SA* cannot obtain the user's private key by using the equation $S_i = W_i + k_i \cdot P_{SA}$. It is obvious that if *SA* tries to find k_i satisfying $K_i = k_i \cdot B$, then the security is based on the intractability of solving the ECDLP.

Theorem 4.3. *The attacker cannot generate a valid PK.*

Proof:

Because the user and the authority cooperatively generate the user's public key, the user cannot generate the public key by himself/herself. Furthermore, according to the equation for generating the witness $W_i = s_{SA}\,(P_i + I_i) + x_i P_{SA}$, the user cannot forge a guarantee to validate the public key by himself/herself without knowing x_i and s_{SA}. Hence the generation of a valid public key is secure.

Theorem 4.4. *SA's dishonesty is detectable.*

Proof:

If *SA* wants to masquerade U_i, it must generate a key pair (P_i', S_i'). However, this will lead to the fact that there are two public keys P_i and P_i' in the public key directory for an identical user, thus, the dishonesty of *SA* is detectable. Because the generation of key pairs is based on the proposed self-certified public key cryptosystems as presented in subsection 2.1, the verification of $\hat{e}(S_i, B) = \hat{e}(2P_i, P_{SA}) \cdot \hat{e}(I_i, P_{SA})$ cannot succeed. Hence, *SA* cannot impersonate any legal user.

4.3 Security of the Proposed Authentication Scheme

[Secure Proxy Signature Scheme]

Here, we will discuss the security of the proxy signature as follows:

Theorem 4.5 *Impersonation of the original signer by malicious attacker is unsuccessful.*

Proof:

a. It is computationally infeasible for an attacker to obtain SA's private key s_{SA} from the SA's proxy key w'. Although an attacker can get M and P_{SA}, without knowing r, he/she cannot derive s_{SA} from the equation $w' = r \cdot P_{SA} + s_{SA} \cdot M$.

b. An attacker can get SA's public key P_{SA} from the public channel, but he/she cannot derive s_{SA} from P_{SA}. It is protected by the CDH assumption that we have mentioned in the security analysis of proposed PKC.

Theorem 4.6 *Acquiring the SA's random integer r is infeasible.*

Proof:

a. Because it is computationally infeasible for an attacker to obtain SA's private key s_{SA} from the proxy delegation (R, w', m_w). Thus, without knowing s_{SA}, the attacker cannot derive X from the equation $R = d \cdot P_{SA} - c \cdot S_a$. Moreover, based on the OWHF assumption, it is hard to compute X from d.

b. An attacker can obtain $D = D' = e(R, B) \cdot (T_a)^c$ from the requirement list, and then he/she may try to derive d from the equation $D = d \cdot B$. But it is protected by the ECDLP.

Theorem 4.7 *Acquiring a user's random integer t is infeasible.*

Proof:

a. An attacker can obtain (g, Y, M) from the public channel, but it is computationally infeasible to derive t from the equation $Y = t \cdot P_{SA} - g \cdot S_b$ because of without knowing S_b.

b. An attacker can derive $E = E' = e(Y, B) \cdot (T_b)^g$ from the message (g, Y, M), but he/she still cannot derive t from $E = t \cdot B$. It is based on the difficulty of solving the ECDLP.

Theorem 4.8 *Forging a valid proxy delegation* (c, R, w) *cannot succeed.*

Proof:

Consider the scenario that an attacker attempts to forge a requirement list (c', R', w'). The attacker can create a fake warrant w', and then he/she selects a random number d' to compute $D' = d' \cdot B$ and $c' = H_2(w' \| D')$.

Finally, he/she may attempt to compute R' which satisfies $R' = d' \cdot P_{SA} - c' \cdot S_a$. However, the attacker cannot get S_a, thus he/she still cannot find out R' to satisfy the equation.

Theorem 4.9 *Forging a valid proxy signature* (g, Y, M) *cannot succeed.*

Proof:

If an attacker wants to forge a contract (g', Y', M'), then he/she needs to choose a random number t', and computes $E' = t' \cdot B$ and $g' = H_2(M' \| E')$. However, because he/she cannot obtain the host's private key S_b, he/she cannot find out Y' to satisfy the equation $Y = t \cdot P_{SA} - g \cdot S_b$.

[Secure Authentication Scheme for Protecting Agent Platforms]

The proposed authentication scheme is based on the proxy signature approaches. In addition, any secure proxy signature can be adopted in the implementation of the proposed authentication scheme. The scheme is based on the difficulty of solving ECDL, BDH, and OWHF assumptions.

1. *Secrecy*

It is computationally infeasible for an attacker to derive the user's private key S_U from the corresponding public key P_U. Also Revealing *SA*'s private key will not succeed. Moreover, the host's do not maintain any database of user's keys. Thus, the private keys in our scheme can be kept secret.

2. *Unforgability*

Assume that an intruder wants to forge a legal user to login with the host platform. For remote access, the intruder can previously intercept a login request {R, T, C, ID_U, m_w, req}, but the key point is that the intruder cannot obtain the user's and *SA*'s private keys. Thus he/she cannot forge a fake R', T', C', and m_w'. Moreover, the proposed scheme utilizes the warrant and timestamp to verify the validity of the proxy signature, thus an intruder cannot forge a legal user.

3. *Replay resistance*

To resist the replay attack, our scheme uses the concept of timestamp. When the intruder replays the previously intercepted login messages and wants to masquerade as a legal user. The intruder will fail the test in the authentication phase. Therefore, the proposed scheme is secure to against the replaying attacks.

5. Conclusions

This paper discusses about protecting platforms against hostile mobile agents. In order to protect the security of transactions in e-commerce, we propose an appropriate public key cryptosystem (PKC) for mobile agent based networks. The proposed PKC is constructed using the pairing-based cryptosystems, and is developed by integrating the self-certified public key cryptosystems with the ID-based public key cryptosystems. In addition, we further employ the integrated cryptosystems to design an authentication scheme for the mobile agent based networks.

The proposed protection scheme for agent platforms in the paper has the following advantages:

1. When verifying the validity of public key, it does not need to spend extra time to verify the signature in the digital certificate.
2. It can be concurrently fulfilled to verify both a signature and the valid public key.
3. Since the proposed methods are combined with the ID-based public key cryptosystem, they can reduce the computation cost greatly.
4. The proposed schemes arise the security level of pairing-based cryptosystems to overcome the drawbacks of the ID-based schemes.
5. A user can register once to a system authority for many services.
6. The load of registration can be delegated to one trusted third party.

7. The host platforms do not maintain any database of users' keys.
8. Only one verification operation is needed when verifying both the validity of the user's identity and the validity of the login request.

In summary, the proposed PKC can also reduce the key size, computing time, and transmission cost, so it is quite suitable for the mobile agent environments. Furthermore, according to the security analyses that have been presented in section 4, the proposed schemes are useful for mobile agent platforms protection and mobile agents authentication.

References

[1] Paulo S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems," *Advances in Cryptology – CRYPTO 2002*, Lecture Notes in Computer Science, Vol. 2442, Springer-Verlag, pp. 354-368, 2002.

[2] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," *Advances in Cryptology – CRYPTO 2001*, Lecture Notes in Computer Science, Vol. 2139, Springer-Verlag, pp. 213-229, 2001.

[3] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," *Advances in Cryptology – ASIACRYPT 2001*, Lecture Notes in Computer Science, Vol. 2248, Springer-Verlag, pp. 514-532, 2001.

[4] D. M. Chess, "Security Issues in Mobile Code Systems," *Mobile Agents and Security*, Lecture Notes in Computer Science, Vol. 1419, Springer-Verlag, pp. 1-14, 1998.

[5] G. Frey, M. Müller, and H. G. Rück, "The Tate Pairing and the Discrete Logarithm Applied to Elliptic Curve Cryptosystems," *IEEE Transactions on Information Theory*, Vol. 45 No. 5, pp. 1717-1719, 1999.

[6] G. Frey and H. G. Rück, "A Remark Concerning m-Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves," *Mathematics of Computation*, Vol. 62, No. 206, pp. 865-874, 1994.

[7] S. D. Galbraith, "Supersingular Curves in Cryptography," *Advances in Cryptology – ASIACRYPT 2001*, Lecture Notes in Computer Science, Vol. 2248, Springer-Verlag, pp. 495-513, 2001.

[8] S. D. Galbraith, K. Harrison, and D. Soldera, "Implementing the Tate pairing," *Algorithmic Number Theory Symposium, ANTS-V*, Lecture Notes in Computer Science, Vol. 2369, Springer-Verlag, pp. 324-337, 2002.

[9] M. Girault, "Self-Certified Public Keys", *Advances in Cryptology – EUROCRYPT '91*, Springer-Verlag, pp. 491-497, 1991.

[10] A. Joux, "A One-Round Protocol for Tripartite Diffie-Hellman," *Algorithm Number Theory Symposium, ANTS-IV*, Lecture Notes in Computer Science, Vol. 1838, Springer-Verlag, pp. 385-394, 2000.

[11] Kim, S., Oh, S., Park, S. and Won, D., "On saeednia's key-exchange protocols," *KICS* (*Korean Institute of Communication Sciences*) *Conference*, Vol. 17, No. 2, pp.1001-1004, 1998.

[12] P. Kotzanikolaou, M. Burmester, and V. Chrissikopoulos, "Secure Transactions with Mobile Agents in Hostile Environments," *Proceedings of the Fifth Australasian Conference on Information Security and Privacy, ACISP 2000*, Lecture Notes in Computer Science, Vol. 1841, Springer-Verlag, pp. 289-297, 2000.

[13] A. J. Menezes, T. Okamoto, and S. Vanstone, "Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field," *IEEE Transactions on Information Theory*, Vol. 39, pp. 1639-1646, 1993.

[14] H. Petersen and P. Horster, "Self-Certified Keys: Concepts and Applications", *Proceedings of Communications and Multimedia Security '97*, Chapman & Hall, pp. 102-116, 1997.

[15] S. Saeednia, "Identity-Based and Self-Certified Key Exchange Protocols," *Proceedings of the Second Australasian Conference on Information Security and Privacy, ACISP '97*, Lecture Notes in Computer Science, Springer-Verlag, pp. 303-313, 1997.

[16] R. Sakai, K. Ohgishi, and M. Kasahara, "Cryptosystems Based on Pairing," *Proceedings of Symposium on Cryptography and Information Security*, *SCIS 2000*, 2000.

[17] T. Sander and C. F. Tschudin, "Protecting Mobile Agents Against Malicious Hosts," *Mobile Agents and Security*, Lecture Notes in Computer Science, Vol. 1419, Springer-Verlag, pp. 44-60, 1998.

[18] A. Shamir, "Identity Based on Cryptosystems and Signature Schemes," *Advances in Cryptology – CRYPTO '84*, Lecture Notes in Computer Science, Vol. 196, Springer-Verlag, pp. 47-53, 1984.

[19] Wu, T.C., Chang, Y.S. and Lin, T.Y., "Improvement of Saeednia's self-certified key exchange protocols," *Electronics Letters*, Vol. 34, No. 11, 1998, pp. 1094-1095.

[20] X. Yi and C. K. Siew, "Secure Agent-Mediated Online Auction Framework," *International Journal of Information Technology*, Vol. 7, No. 1, 2001.