Proceedings of the 5th WSEAS International Conference on Applied Computer Science, Hangzhou, China, April 16-18, 2006 (pp416-421)

Constructing Secure Mobile-Agent-Based Consumer Electronic Applications

WOEI-JIUNN TSAUR and GIN-WE LIU
Department of Information Management
Da-Yeh University
112 Shan-Jiau Rd., Da-Tsuen, Changhua 515
TAIWAN

Abstract: - This paper proposes two appropriate security schemes for protecting consumer electronic applications in mobile agent based networks. As far as mobile agent security is concerned, we develop a proxy signature scheme for protecting mobile agents against malicious agent hosts using the ECC-based self-certified public key cryptosystem. The proposed proxy signature scheme can protect users' private keys, and provide the fairness of contracts signed by agents. In addition, based on the proposed proxy signature scheme, we further design a proxy authenticated encryption scheme so that the signature of the contracts will satisfy users' constraints, and the non-repudiation of servers can be achieved. Furthermore, we also implement the proposed security schemes to achieve security requirements of confidentiality, integrity, and non-repudiation for protecting Linux-based mobile agents in an electronic auction application. Hence, we affirm that the proposed security schemes are suitable for protecting consumer electronic applications in mobile-agent-based network environments.

Key-Words: - Mobile agent, Elliptic curve cryptosystem (ECC), Proxy signature, Self-certified public key cryptosystems, Consumer electronic applications, Linux

1 Introduction

In recent years, there are many business applications based on mobile agents on the Internet. Those agents of business applications usually provide personalization, automation, intelligence, etc. However, it also results in many security threats such as stealing data from hosts by agents and tampering constraints of agents by hosts. For instance, when a mobile agent carrying a user's private key roams among servers on the Internet, the agent may find a bid satisfies the user's constraints, and then sign the bid [1, 17]. However, users will not wish to equip agents with their private signature keys when the agents may execute on distrusted agent hosts [9, 15, 16]. In this paper, we will develop security schemes based on cryptographic solutions [10, 15] for the prevention of agent tampering.

This paper develops a proxy signature scheme and a proxy authenticated encryption scheme for protecting mobile agents against malicious agent hosts using the ECC-based self-certified public key cryptosystem proposed by Tsaur [8]. The proposed security schemes are constructed using the ECC, and it also integrates the self-certified public key cryptosystem [2, 12-14] to provide higher security strength. The proposed proxy signature scheme can protect users' private keys, and provide the fairness of contracts signed by agents. Furthermore, we employ the proposed proxy signature scheme to further design a proxy authenticated encryption scheme so that the signature of the contracts will satisfy users' constraints, and the non-repudiation of servers can be achieved. In summary, these proposed schemes are able to accomplish the security requirements of confidentiality, integrity, and non-repudiation for protecting mobile agents in electronic business applications. Also, we implement the proposed security schemes for protecting an electronic auction application in Linux-based mobile agent networks.

The rest of this paper is organized as follows. In Section 2, based on the ECC-based self-certified public key cryptosystem, the proxy signature scheme and proxy authenticated encryption scheme are constructed for protecting mobile-agent-based consumer electronic applications. In Section 3, security analyses about attacks on the proposed schemes consolidate the feasibility of the schemes. Section 4 presents the implementation of the proposed schemes on an electronic auction application. Finally, some concluding remarks are given in Section 5.

2 Security Schemes For Mobile Agent Based Consumer Electronic Applications

In this section, several security schemes constructed using the efficient public key cryptosystem [8] are designed for protecting consumer electronic applications in mobile agent based networks.

A. Initialization

The entities in the system are a system authority (*SA*), users (U_i), hosts (H_i), and mobile agents (*MA*) generated by specific users. Assume that *SA* is responsible for user and host registration and key generation. We first define the notations used in the proposed schemes as follows:

- *p:* a field size, where *p* is typically either an odd prime or a power of 2 in general applications, and its length is about 160 bits.
- an elliptic curve *E* over F_p:
  E: $y^2 = x^3 + ax + b$, where the two field elements $a, b \in F_p$ and $4a^3+27b^2 \neq 0$ (mod *p*), and all the points (*x*, *y*), $x \in F_p, y \in F_p$, on *E* form the set of $E(F_p)$ containing a point *O* called the point at infinity[6].
- *B*: a base point of order *n* over $E(F_p)$, where *n* is a large prime (160 bits) and the number of F_p-rational points on *E* denoted by # $E(F_p)$, is divisible by *n*.
- s_{SA}: SA's private key, where $s_{SA} \in [2,\ n-2]$.
- P_{SA}: SA's public key, where $P_{SA} = s_{SA} \cdot B$ ("$\cdot$" means the multiplication of a number and an elliptic curve point.).
- *h*(.): a one-way hash function that accepts a variable length input and produces a fixed length output value *j*, where $j \in [2,\ n-2]$ and its length is 160 bits. The one-way hash function *h*(.) should satisfy the properties [3] that given *h(x)*, it is computationally infeasible to find $x' \neq x$ such that $h(x') = h(x)$, meanwhile, $h(x') \neq h(x)$ if and only if $x' \neq x$.
- P_x: the x-coordinate of point *P*.

After that, SA publishes *E, B, p, n,* P_{SA} and *h*, while keeping s_{SA} secret. Based on the efficient public key cryptosystem proposed by Tsaur [8], U_i and H_i then obtain their public-key/private-key pair P_u/s_u and P_h/s_h, respectively.

B. The Proposed Proxy Signature Scheme

The proposed proxy signature scheme can hide users' private keys and provide the fairness of contracts. In the following, we first define the notations used in the proposed schemes, and then show the steps in the commission phase, the transaction phase and the verification phase, respectively.

[Notations]

U_i: a user who generates the mobile agent (*MA*).
H_i: a host who executes the mobile agent in the agent platform.
M: the content of the contract, where $M = \{0,1\}^*$ denotes that the message space is $\{0,1\}^*$, and its size is *n*.
N: constraints of a user, including description of goods, maximum price, delivery data, issuing time of constraint, etc. The requirements with constraints are assigned to *MA* by user U_i.
w: the delegation warrant for *MA*, which is empowered by U_i, including agent name, routing lists, user identity ID_i, user's public key P_i, allotted time and other public information.
||: a symbol denoting concatenation.

[Commission Phase]: Preparing the Agent

Step 1. U_i establishes his/her requirements with constraints *N* to limit the power of hosts. U_i then computes
$$d = h(N) \quad \text{and} \quad D = d \cdot B = (D_x, D_y)$$
Step 2. U_i computes $c = h(w \| D_x)$.
Step 3. U_i uses his/her private key s_u to calculate $r = d - c \cdot s_u \pmod n$.
Step 4. U_i assigns the requirement lists (*c*, *r*, *w*) to *MA*, and launches *MA* to search a specific product through networks.

[Transaction Phase]: Executing the Agent

When host H_i generates a bid that conforms to U_i's requirements, *MA* and H_i will negotiate to sign a contract. They take the following steps to perform the transaction:

Step 1.When H_i receives the requirement list (*c*, *r*, *w*), it can obtain U_i's I_u and public key P_u from the warrant *w*, and then calculate $V_u = P_u + h(I_u) \cdot B + [(P_{ux} + h(I_u)) \bmod n] \cdot P_{SA}$
Step 2. H_i computes $D' = r \cdot B + c \cdot V_u$ (2)
Step 3. H_i verifies whether the equation h($w \| D'_x$) = *c* holds. If the result is correct, (*c*, *r*, *w*) is a valid requirement list, and H_i then goes to Step 4; otherwise, H_i rejects the transaction.
Step 4. H_i chooses a random integer $t \in [2,\ n-2]$, and computes $E = [(t \cdot r) \bmod n] \cdot B = (E_x, E_y)$, where *E* is a message denoted the bidding

information and dealings information.

Step 5. *MA* negotiates with H_i, and makes a dealings contract *M*. Then H_i computes $g = h(M \| E_x)$.

Step 6. H_i uses its private key s_h to compute $y = t \cdot r - g \cdot s_h \pmod{n}$.

Step 7. H_i attaches to *MA* the requirement list, bidding information, and the contract, ((*c*, *r*, *w*), (*g*, *y*, *M*)), and then *MA* comes back to U_i.

[Verification Phase]: Verifying the Signature

When U_i receives the message from *MA*, he/she can verify the validity of his/her purchase as follows:

Step 1. A verifier or U_i can take H_i's I_h and P_h from the received message, ((*c*, *r*, *w*), (*g*, *y*, *M*)). The verifier computes $V_h = P_h + h(I_h) \cdot B + [(P_{hx} + h(I_h)) \bmod n] \cdot P_{SA}$ to check the validity of signature.

Step 2. The verifier computes

$$D' = r \cdot B + c \cdot V_u = (D'_x, D'_y) \text{, and}$$

$$E' = y \cdot B + g \cdot V_h = (E'_x, E'_y) \tag{3}$$

Step 3. The verifier checks whether the equation $h(w \| D'_x) = c$ holds. If it is correct, H_i is a valid proxy signer; otherwise, the verifier rejects the message ((*c*, *r*, *w*), (*g*, *y*, *M*)).

Step 4. The verifier checks whether the equation $h(M \| E'_x) = g$ holds. If it is true, (*g*, *y*, *M*) is a valid proxy signature subject to the constraints of the requirements.

C. The Proposed Proxy Authenticated Encryption Scheme

The proposed proxy signature scheme can hide users' private keys, but the transmitted messages are plaintexts on the networks. Therefore, due to the requirement of confidentiality, we also construct a proxy authenticated encryption scheme for mobile-agent-based networks using the above proposed proxy signature scheme. We first define the notations used in the scheme, and then show the steps in each phase as follows.

[Notations]

V : a third-party authority or any unspecified verifier
$\oplus$: bitwise *exclusive-or* operator.

[Proxy Authorization]

In this phase, host H_i uses a proxy key to sign the contract subject to the constraints of the user. It is exactly the same as Step 1 to Step 5 of the transaction phase in the proposed proxy signature scheme.

In order to generate a proxy key S_P, H_i uses its private key s_h to generate $S_P = r + g \cdot s_h \pmod{n}$, where $g = h(M \| E_x)$ is a hashed bidding information.

[Proxy Authenticated Encryption]

In this phase, host H_i can proxy-authenticated-encrypt the contract *M*, and create a valid authenticated encryption message to the user. In addition, the contract contains a specific redundancy.

Step 1. The host H_i obtains U_i's ID_u and public key P_u.

Step 2. H_i computes

$$V_u = P_u + h(I_u) \cdot B + [(P_{ux} + h(I_u)) \bmod n] \cdot P_{SA}$$

Step 3. H_i randomly chooses two integers *t*, $u \in [2,\ n-2]$, and then calculates the following equations:

$$\mathrm{T} = t \cdot B = (T_x, T_y)$$

$$r = M \oplus h(T_x)$$

$$s = t + h(r) \cdot S_p \pmod{n}$$

$$C_1 = u \cdot B$$

$$\mathrm{U} = u \cdot V_u = (U_x, U_y)$$

$$C_2 = r \oplus h(U_x)$$

Step 4. H_i attaches the authenticated encryption message $\{C_1, C_2, s\}$ and the hashed bidding information *g* to the mobile agent, and then send back to the user. Notice that the message (or contract) is not a plaintext.

[Proxy Authenticated Decryption]

Step 1. Upon receiving $\{C_1, C_2, s\}$, U_i uses his/her private key s_u to recover *r* as follows:

$$r = C_2 \oplus h(X(\mathrm{s_u} \cdot C_1)) \tag{4}$$

Step 2. U_i further computes

$$V_h = P_h + h(I_h) \cdot B + [(P_{hx} + h(I_h)) \bmod n] \cdot P_{SA}$$

and

$$S_p \cdot B = D - c \cdot V_u + g \cdot V_h = V_P \tag{5}$$

Then he/she performs the following equation to recover the contract *M* from (*r*, *s*).

$$M = r \oplus h(X(s \cdot B - h(r) \cdot V_p)) \tag{6}$$

Step 3. U_i verifies the authenticated encryption message for *M* by checking whether the validity of attached redundancy holds. If the redundancy is valid, then $\{C_1, C_2, s\}$ is a valid authenticated encryption message.

[Signature Verification]

In this phase, a third-party authority *V* can arbitrate the dispute between users (customers) and hosts (servers). Thus, the authenticated encryption message has no need to be secret. Furthermore, U_i can release

(r, s) to V to verify the validity of the signature.

The third party V, who receives (r, s), can recover M by performing formula (6), and check the attached redundancy to verify the signature.

3 Security Analysis

The security of the proposed schemes is primarily relied on the difficulties of solving elliptic curve discrete logarithm problem (ECDLP) [5-7, 11] and one-way hash function (OWHF). The security analyses of the proposed security schemes for mobile agent based consumer electronic applications are discussed in the following.

[Proposed Proxy Signature Scheme]

We will discuss the security of the proposed proxy signature scheme as follows:

1. *Revealing the user‘s (original signer's) private key*

   a. It is computationally infeasible for an attacker to obtain user's private key s_u from the user's requirement list (c, r, w). Although an attacker can get c and r, he/she cannot derive s_u from the equation $r = d - c \cdot s_u \pmod n$ without knowing d.

   b. An attacker can get user's public key P_u from the public channel, but he/she cannot derive s_u from P_u. It is protected by the ECDL assumption that we have mentioned in the security analysis of proposed PKC.

2. *Revealing the host‘s (proxy signer's) private key*

   a. An attacker may acquire the contract (g, y, M). However, without knowing t, he/she cannot derive the host's private key s_h from the equation $y = t \cdot r - g \cdot s_h \pmod n$.

   b. Because of the ECDL assumption, it is computaionally infeasible for an attacker to derive the host's private key s_h from the corresponding public key P_h.

3. *Acquiring the user's constraints N (or d)*

   a. Because it is computationally infeasible for an attacker to obtain user's private key s_u from the user's requirement list (c, r, w). Thus, the attacker cannot derive N from the equation $r = d - c \cdot s_u \pmod n$ without knowing s_u and d. Moreover, based on the OWHF assumption, it is hard to compute N from d.

   b. An attacker can obtain $D = D' = r \cdot B + c \cdot V_u$ from the requirement list, and then he/she may try to derive d from the equation $D = d \cdot B$. However, it will face the intractability of solving the ECDLP.

4. *Acquiring the host's random integer t*

   a. An attacker can obtain (g, y, M) from the public channel, but it is computationally infeasible to derive t from the equation $y = t \cdot r - g \cdot s_h \pmod n$ because of without knowing s_h.

   b. An attacker can derive $E = E' = y \cdot B + g \cdot V_h$ from the message (g, y, M), but he/she still cannot derive t from $E = [(t \cdot r) \bmod n] \cdot B$, because it will also face the intractability of solving the ECDLP.

5. *Forging a valid requirement list* (c, r, w)

   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 $c' = h(w'||D'_x)$. Finally, he/she may attempt to compute r' which satisfies $r' = d' - c' \cdot s_u \pmod n$. However, the attacker cannot get s_u, thus he/she still cannot find out r' to satisfy the equation.

6. *Forging a valid contract* (g, y, M)

   If an attacker wants to forge a contract (g', y', M'), then he/she needs to choose a random number t', and computes $E' = [(t' \cdot r) \bmod n] \cdot B$ and $g' = h(M'||E'_x)$. However, because he/she cannot obtain the host's private key s_h, he/she cannot find out y' to satisfy the equation $y = t \cdot r - g \cdot s_h \pmod n$.

[Proposed Authenticated Encryption Scheme]

A secure authenticated encryption scheme should satisfy the security requirements of confidentiality, unforgeability, and non-repudiation [4]. According to the three security requirements, we can analysis the proposed proxy authenticated encryption scheme as follows:

1. *Confidentiality of the signed messages*

   If an attacker tries to recover the signed message M from the authenticated encryption message $\{C_1, C_2, s\}$, then he/she should recover r first, and he/she will encounter the intractability of solving the OWHF assumption and ECDL problems.

2. *Confidentiality of the signer's private key*

   An attacker cannot derive the signer's private key form either the authenticated encryption message $\{C_1, C_2, s\}$ or the signature (r, s) which has been released, because he/she will face the difficulty of solving the OWHF assumption and ECDL problems.

3. *Unforgeability of the authenticated encryption*

messages

Consider the scenario that an attacker attempts to masquerade in creating a false message *M'* to create a valid authenticated encryption message for the specified verifier. He/she should choose the random numbers *t'* and *u'*, and then compute C_1', C_2', and *r'*. If the attacker can find an *s'* which satisfies the equation $M = r \oplus h(X(s \cdot B - h(r) \cdot V_p))$, then he/she will succeed in forging attack. However, he/she will encounter the intractability of solving the ECDL and OWHF assumptions.

4. *Unforgeability of the signatures*

Without knowing the signer's private key, an attacker cannot succeed in forging a valid signature (*r*, *s*), because it is also protected by the ECDL and OWHF assumptions.

5. *The property of non-repudiation*

In the proposed authenticated encryption scheme, the receiver can release the recovered signature to a third party, and thus the signer cannot deny sending the signature. Moreover, because of the unforgeability, the signer cannot deny creating the authenticated encryption message.

4 Implementation

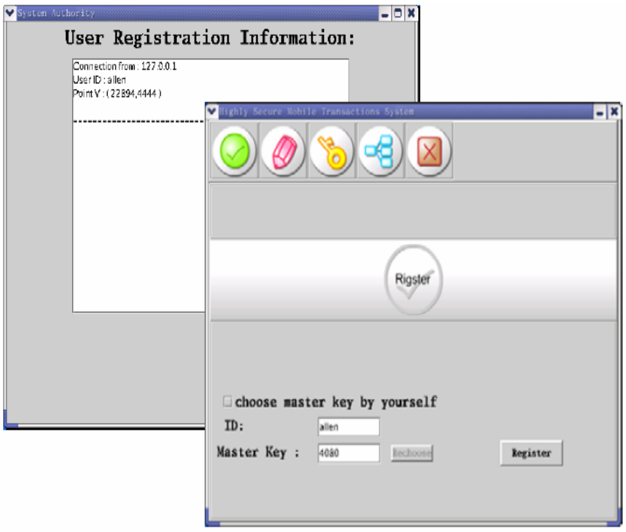
In this section, we will present the implementation of the proposed security schemes for protecting Linux-based mobile agent networks in an electronic auction application. In this paper, we simulate all of the security schemes by developing an electronic auction system using Java, where users are bidders and agent hosts are Linux-based auction servers.

In Fig. 1, it shows a user registers to SA in the key generation phase. In Fig. 2, we simulate the proxy signature scheme such that a user can launch the mobile agent to sign contract with the host. For simulating the proxy authenticated encryption scheme, we also do the same action as the proxy signature scheme, as shown in Fig. 3. However, the difference between the proxy signature scheme and the proxy authenticated encryption scheme is that the proxy authenticated encryption scheme can both encrypt and sign a message in a logically single step.

5 Conclusion

This paper discusses about the security of mobile agents in electronic business applications. We focus on the cryptographic solutions for the confidentiality, integrity, and non-repudiation in mobile agent based consumer electronic applications. In order to protect the security of e-commerce transactions, we propose several appropriate security schemes for the mobile-agent-based networks. The proposed schemes are constructed based on the elliptic curve cryptosystem based self-certified public key cryptosystem, and therefore they do not need to spend extra time to verify the signature in the digital certificate as used in the certificate-based cryptosystem, when checking the validity of public key. Also, the security requirement of signers' non-repudiation is achieved actually.

Furthermore, we also implement all of the proposed security schemes to demonstrate our security schemes can practically carry out the security requirements of confidentiality, integrity, and non-repudiation for protecting mobile agent based consumer electronic applications.



User Registration Information:

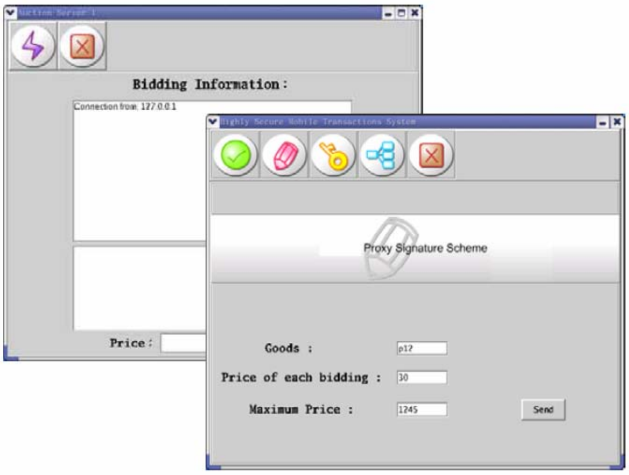
Connection from : 127.0.0.1
User ID : allen
Point V : (22894,4444)

□ choose master key by yourself

ID: allen

Master Key : 4080

Fig. 1. Key generation



Bidding Information :

Connection from: 127.0.0.1

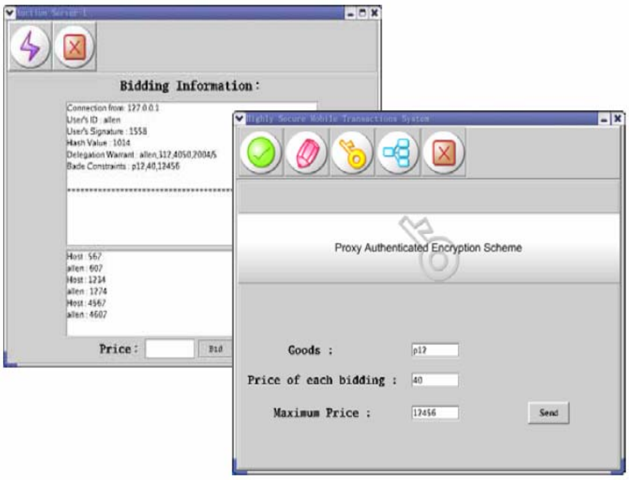
Price :

Goods : p12

Price of each bidding : 30

Maximum Price : 1245

Fig. 2. Proxy signature scheme


Auction Server 1
Bidding Information :
Highly Secure Mobile Transactions System
Proxy Authenticated Encryption Scheme
Bid
Send

Connection from: 127.0.0.1
User's ID : allen
User's Signature : 1558
Hash Value : 1014
Delegation Warrant : allen,112,4050,2004/5
Bade Constraints : p12,40,12456

Host : 567
allen : 607
Host : 1234
allen : 1274
Host : 4567
allen : 4607

Price :

Goods : p12
Price of each bidding : 40
Maximum Price : 12456

Fig. 3. Proxy authenticated encryption scheme

References:

[1] D. Chess, B. Grosof, C. Harrison, D. Levine, C. Parris, and G. Tsudik, Itinerant Agents for Mobile Computing, *IEEE Personal Communications Magazine*, Vol.2, No.5, 1995, pp. 34-49.

[2] M. Girault, Self-Certified Public Keys, *Advances in Cryptology: Proceedings of Eurocrypt'91*, *LNCS 547*, Springer-Verlag, 1992, pp. 490-497.

[3] L. Harn, New Digital Signature Scheme Based on Discrete Logarithm, *Electronics Letters*, Vol.30, No.5, 1994, pp. 396-398.

[4] W. H. He and T. C. Wu, Cryptanalysis and Improvement of Petersen-Michels Signcryption Scheme, *IEE Proceedings – Computer and Digital Techniques*, Vol.146, No.2, 1999, pp.123-124.

[5] A. Jurisic and A. Menezes, Elliptic Curves and Cryptography, *Dr. Dobb's Journal*, 1997, pp. 26-35.

[6] N. Koblitz, Elliptic Curve Cryptosystems, *Mathematics of Computation*, Vol.48, No.17, 1987, pp. 203-209.

[7] N. Koblitz, A. Menezes and S. Vanstone, The State of Elliptic Curve Cryptograph, *Designs, Codes and Cryptography*, Vol.19, 2000, pp. 173-193.

[8] W. J. Tsaur, Several Security Schemes Constructed Using ECC-Based Self-Certified Public Key Cryptosystems, *Applied Mathematics and Computation*, Vol.168, No.1, 2005, pp. 447-464.

[9] P. Maes, R. Guttman, and A. Moukas, Agents That Buy and Sell, *Communications of the ACM*, Vol.42, 1999, pp. 81-91.

[10] M. Mambo, K. Usuda, and E. Okamoto, Proxy Signatures: Delegation of the Power to Sign Messages, *IEICE Transactions on Fundamentals*, Vol.E79-A, No.9, 1996, pp. 1338-1354.

[11] A. J. Menezes and S. A. Vanstone, Elliptic Curve Cryptosystem and Their Implementation, *Journal of Cryptology*, Vol.6, No.4, 1993, pp. 209-224.

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

[13] S. Saeednia, Identity-Based and Self-Certified Key Exchange Protocols," *Proceedings of Second Australasian Conference on Information Security and Privacy (ACISP'97)*, LNCS Vol.1270, Springer-Verlag, 1997, pp. 303-313.

[14] S. Saeednia, A Note on Girault's Self-Certified Model, *Information Processing Letters*, Vol.86, 2003, pp. 323-327.

[15] T. Sander and C. F. Tschudin, Protecting Mobile Agents against Malicious Hosts, *Mobile Agents and Security*, LNCS Vol.1419, Springer-Verlag, 1998, pp. 44-60.

[16] H. Takeda, K. Iino, and T. Nishida, Agent Organization and Communication with Multiple Ontologies, *International Journal of Cooperative Information Systems*, Vol.4, No.4, 1995, pp.312-337.

[17] J. E. White, Mobile Agents Make a Network an Open Platform for Third-Party Developers, *IEEE Computer*, Vol.27, No.11, 1994, pp. 89-90.