TWO SECURE TRANSPORTATION SCHEMES FOR MOBILE AGENTS

Iuon-Chang LIN, Hsia-Hung OU and Min-Shiang HWANG

1. Introduction

Mobile agents have been the focus of a great number of research studies. All the experts in this domain aim to study the relevant technologies and thus enhance business activities.[1,2] In the information era, the Internet is widespread; it is both open and general.

Mobile agent technology has been proposed for use on the Internet. The mobile agent is a software program that acts on behalf of a user or software. It has the following features: (1) it is autonomous; (2) it has one or more objectives; (3) it has a scope of competence; and (4) it may, or may not, collaborate and communicate with other software and users.[3] In order to perform its job, it is able to migrate from a source host to a target host on a network under its own control.[4] However, this may lead to a great deal of security threats and attacks. When a mobile agent moves between a series of hosts, it may happen to encounter either trust-worthy or malicious hosts. A mobile agent must be capable of authenticating legal hosts and other agents to avoid malicious attacks. Ideally, a mobile agent should be versatile, robust, and secure in changing environments. Therefore, the security issue when dealing with mobile agents becomes essential. So far, the research on mobile agent security has been focused on the following topics:[5]

1. Protecting hosts from access by unauthorized parties;
2. Protecting hosts from attacks by malicious agents;
3. Protecting agents from attacks by other agents;
4. Protecting agents from attacks by malicious hosts.

However, only a few research studies have been focused on transportation security, which is in fact a very important topic in mobile agent systems, especially when it comes to business. When a user makes a request for a work to be performed, the request may be tampered with during the transportation, which causes trouble when the user is unwilling to disclose the information as to what agents are to be dispatched and where the destination should be. In this paper, we address the transportation security for mobile agents. When the mobile agent is transported between distributed hosts, there are several security issues that we have to carefully account for: [6,7,8]

- *Confidentiality*: In order to protect the privacy from being violated, all of the transported messages are encrypted. No malicious attacker can wiretap the transported contents.
- *Integrity*: No malicious attacker can modify any message being transferred. If a transported message has been modified, the receiver can easily detect it.
- *Authentication*: The identities of the source hosts and the mobile agents must be identified. Such identification can stop the attacks from malicious users and agents.
- *Non-repudiation*: The system provides the property of non-repudiation. It can prevent the user from denying having sent the request for launching the mobile agent. The property can be applied in many business applications.
- *Audit*: The system should be able to easily launch an audit process in order to find anything exceptional.

In the next section, we shall present a basic framework for the proposed scheme. Then, we shall give the details of the scheme and perform security analysis in Sections 3 and 4, respectively. Finally, we shall give our conclusions in Section 5.

2. A Basic Framework of the Proposed Scheme

This section introduces the basic framework of the proposed protocol. Our method is based on trusted third party and cryptography. The framework is shown in Figure 1.

The framework includes the Source Host (SH), the Trust Server (TS), and the Target Host (TH), whose functions are described as follows:

1. *Source Host*:

   It is a host that owns mobile agents and sends requests for performing jobs to the trust server.

2. *Trust Server*:

   It is a trusted third party. It supports all requests for secure transportation between source hosts and target hosts. When TS receives request for an agent, it verifies the validity of the request and then dispatches the requested agent to a given target host. There are three modules in the trust server:

(a) Agent Directory Database:

This is a database that records the agent functions, source hosts, and historical records. When a target host is forced to accept a visit from a source host, the agent directory database is used to verify the agent.

(b) Agent Control Center:

It supports all control functions of the agents. It searches the applicable data from either the agent directory database or service directory database and controls all the messages transferred via agents.

(c) Service Directory Database:

This is a database that records all the supported target hosts. When an agent is appointed to take a work request, the trust server uses the service directory database to search all the applicable hosts.

3. *Target Host*:

It is a host that an agent is sent to in order for the job to be completed.

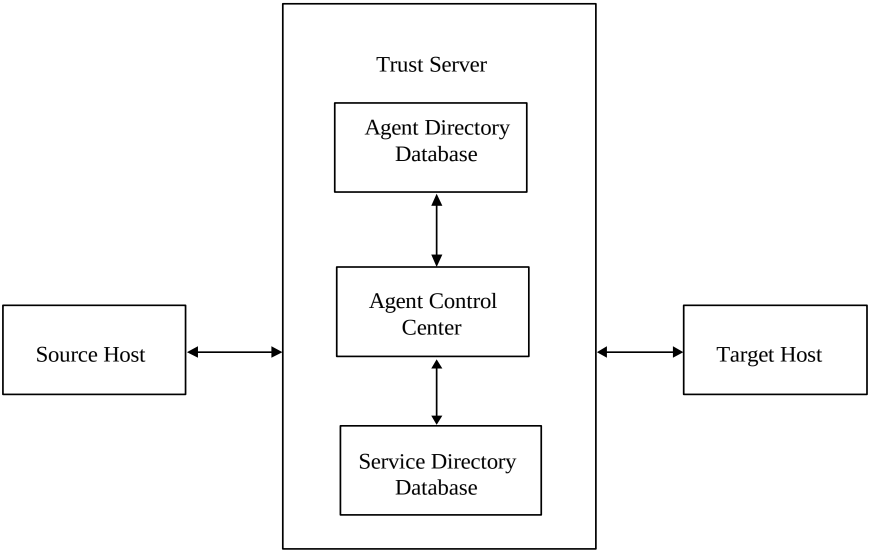

Figure 1: The basic framework

We use the trust server to solve several security problems in mobile agents. Both the source hosts and the target hosts transact through the trust server. All agents must register and leave the records in the agent directory before starting the mobile agent. The agent directory records information such as: which agents belong to which hosts, what the agents' objectives are, and what the agents' source codes or certificates are. Then, all hosts which provide services must register with the service directory. The service directory records all the target host addresses and the services they provide.

3. Two Secure Transportation Schemes for Mobile Agent

In this section, two secure transportation schemes for mobile agents are proposed. The two schemes are designed to set up the transportation protocols for agent delivery between the source and target hosts.

In order to simplify the description of our schemes, we define some notations here.

TS: Trust Server;

SH: Source Host;

TH: Target Host;

A: An agent;

ID_i: The identity of an entity i;

$E_{PK\,i}\,[\ldots]$: An encryption function or a signature verification function using asymmetric crypto-systems, such as RSA, with the entity i's public key being PK_i;

$D_{SK\,i}\,[\ldots]$: A decryption function or digital signature product function using asymmetric crypto-systems, such as RSA, with the entity i's private key being SK_i;

$F_{K\,j}\,[\ldots]$: The encryption function using symmetric crypto-systems, such as DES, with the j-th session key being K_j, which is also used in the decryption function;

Response: A target host's response, Yes or No.

$Noise_n$: A unique serial number.

Scheme One:

In this scheme, we use cryptography techniques to accomplish our goals. Initially, each agent must register with the trust server and send the agent code to the trust server. The trust server will verify the agent to ensure the agent is secure and then store the data in the agent directory of the trust server. The scheme is shown in Figure 2.

The procedure of our first scheme looks as follows:

Step 1. SH sends a request for performing jobs to TS. The request includes TS's ID, SH's ID, the agent's ID, $Noise_1$, session key (K_1), and the signature of these messages. In order to provide confidentiality of communication, the request must be encrypted using TS's public key PK_{TS}. The main purpose of this step is SH to inform TS which agent will be launched.

Step 2. Upon receiving the above messages, TS decrypts them and verifies the signature by using SH's public key PK_{SH}. If the verification result is positive, TS records $Noise_1$ and its corresponding K_1 and locates the agent's function in the Agent Directory Database. Then, TS searches the Service Directory Database for a suitable TH and generates a new session key K_2 with this TH. Next, TS sends TS's ID, SH's ID, agent's ID, $Noise_2$, K_2, and the signature of these messages, which are encrypted with TH's public key PK_{TH}, to TH. Here, TS checks whether the target host provides the requested services.

Step 3. The target host verifies the validity of the received messages and records $Noise_2$ and its corresponding K_2. Then, it sends a reply to the trust server. The messages containing answers are encrypted using a symmetric encryption function F with session key K_2.

Step 4. The trust server receives the answers from the target host. According to $Noise_2$, TS can find the corresponding session key K_2 to decrypt the message. If the reply is "Yes", the trust server will send the agent to the target host. The agent is stored in the agent directory of the trust server at the time when the agent registers. If the reply is "No", then the transportation process is stopped.

Step 5. The trust server notifies the source host which target hosts the agent will be sent to. These messages are encrypted with the session key K_1. Therefore, the content cannot leak out when it is passed over the Internet, and SH can be sure that the messages are sent from TS.

In our first scheme, we use both symmetric cryptography and public key cryptography to achieve data protection. The public key cryptography is used only in the first two transactions to achieve confidentiality and integrity. Since the performance of symmetric cryptography is better than that of public key cryptography, we use symmetric cryptography instead of public key cryptography to achieve the same goals. In this schema, all of the transaction messages must go through the trust server. It has the advantage that the trust server can record all the messages for trail audit if any disagreement occurs in the transaction. However, the shortcoming is that there is a heavy load at the trust server. Therefore, we propose another scheme. In the second scheme, the agents do not need to be stored in the agent directory. When an agent registers, the trust server sends a certificate (*Cert*) to the source host. The certificate is composed of $D_{SK\,TS}(H(Agent) \oplus ID_{SH})$. The certificate can then be used to verify that the agent is a legitimate agent.

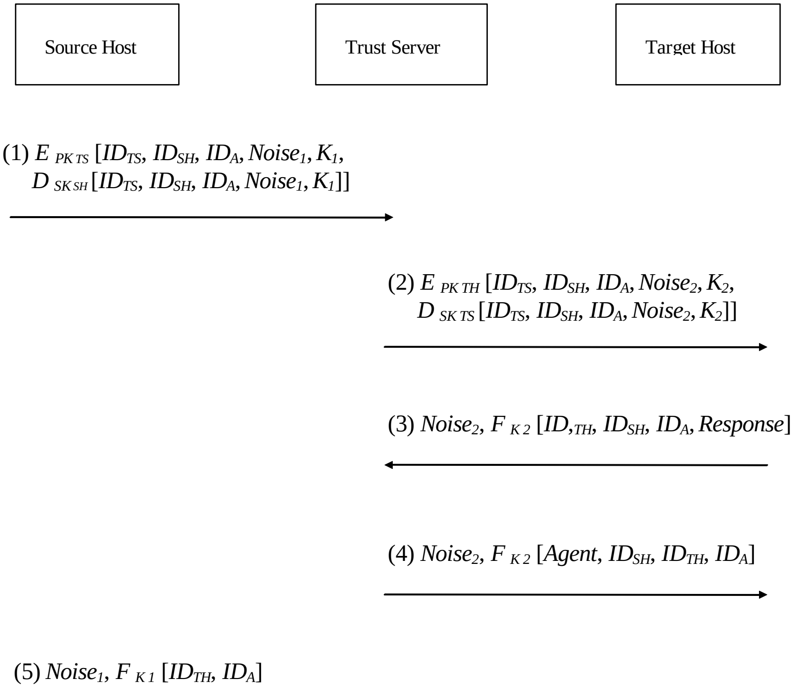
Source Host
Trust Server
Target Host
(1) $E_{PK\,TS}\,[ID_{TS}, ID_{SH}, ID_A, Noise_1, K_1,$
$D_{SK\,SH}\,[ID_{TS}, ID_{SH}, ID_A, Noise_1, K_1]]$
(2) $E_{PK\,TH}\,[ID_{TS}, ID_{SH}, ID_A, Noise_2, K_2,$
$D_{SK\,TS}\,[ID_{TS}, ID_{SH}, ID_A, Noise_2, K_2]]$
(3) $Noise_2, F_{K\,2}\,[ID,_{TH}, ID_{SH}, ID_A, Response]$
(4) $Noise_2, F_{K\,2}\,[Agent, ID_{SH}, ID_{TH}, ID_A]$
(5) $Noise_1, F_{K\,1}\,[ID_{TH}, ID_A]$

Figure 2: The first secure transportation scheme for mobile agents

Scheme Two (Based on certificate):

We use the above mentioned certificate to improve our first scheme. The procedure of our second scheme is described below:

Step 1. This step is the same as the first step in our previous scheme. The main purpose of this step is SH to inform TS which agent will be launched.

Step 2. This step also coincides with the corresponding step from the first scheme. TS checks whether the target host provides the requested services.

Step 3. The target host verifies the validity of the received messages. Then, it sends a reply to the trust server. If the answer is "Yes", it appends $Noise_3$ and its corresponding session key K_3, which are encrypted and signed with PK_{SH} and SK_{TH}, respectively. Then TH encrypts the messages with the session key K_2 and sends $Noise_2$ and the encrypted messages to the TS.

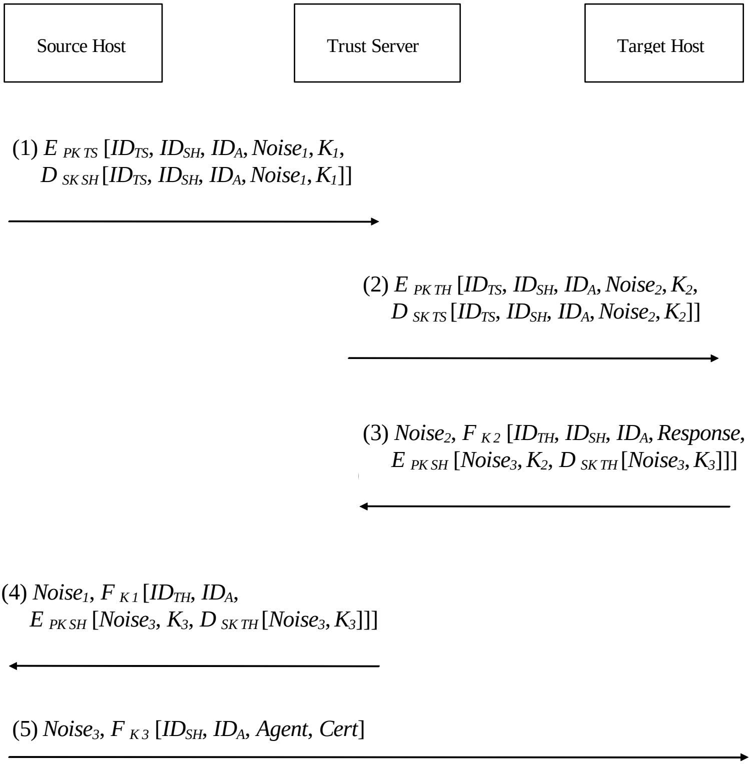
Source Host
Trust Server
Target Host
(1) $E_{PK\,TS}\,[ID_{TS}, ID_{SH}, ID_A, Noise_1, K_1, D_{SK\,SH}\,[ID_{TS}, ID_{SH}, ID_A, Noise_1, K_1]]$
(2) $E_{PK\,TH}\,[ID_{TS}, ID_{SH}, ID_A, Noise_2, K_2, D_{SK\,TS}\,[ID_{TS}, ID_{SH}, ID_A, Noise_2, K_2]]$
(3) $Noise_2, F_{K\,2}\,[ID_{TH}, ID_{SH}, ID_A, Response, E_{PK\,SH}\,[Noise_3, K_2, D_{SK\,TH}\,[Noise_3, K_3]]]$
(4) $Noise_1, F_{K\,1}\,[ID_{TH}, ID_A, E_{PK\,SH}\,[Noise_3, K_3, D_{SK\,TH}\,[Noise_3, K_3]]]$
(5) $Noise_3, F_{K\,3}\,[ID_{SH}, ID_A, Agent, Cert]$

Figure 3: The second secure transportation scheme for mobile agents

Step 4. The trust server receives the messages containing the answers from the target host. According to $Noise_2$, TS can find the corresponding session key K_2 to decrypt the messages. If the reply is "Yes", then the TS notifies the SH which TH the agent will be sent to. These messages are then encrypted with the session key K_1, which includes ID_{TH}, ID_A and $E_{PK\,SH}\,[Noise_3, K_3, D_{SK\,TH}\,[Noise_3, K_3]]$. If the reply is “No”, then the transaction is stopped. The objective is to let the SH know where the agent will be delivered and which session key will be used to protect the confidentiality in the next step.

Step 5. SH’s ID, A’s ID, agent, and certificate are encrypted with the session key K_3. Then SH sends $Noise_3$ and the encrypted messages to the TH.

Most of the procedures of protocol 2 are the same as those of protocol 1. Furthermore, if a target host wants to verify an agent’s legitimacy, it proceeds as follows:

1. Hash the agent code and then take it along with the source host’s ID to perform XOR.
2. Decrypt the certificate using TS’s public key.
3. Compare results of step 1 and step 2, and the target host will verify the correctness.

Comparison

The first scheme is a general method, and the focus is on the trust server. All messages between the source host and the target host must go through the trust server where the agent’s code is stored. The trust server is not only a third party. It also transmits the messages. So, the trust server is important, and it takes a heavy load. For this reason, we have proposed a second scheme to reduce the load on the trust server. In the second scheme, the role of the trust server is to be a successful transactor. There are both authentication and target search. To promote transaction, the source host directly delivers the agent’s code to the target host. That can reduce the load on the trust server.

The point here is that the two schemes we propose in the same area are brought out to offer choices for different situations with different requirements. For local networks or a few agents, the first scheme is the better choice. Otherwise, the second scheme can support the heavy load in a large-scale network.

Both of the two proposed schemes can achieve our objective of offering secure transportation for the mobile agent delivered between distributed hosts. Security issues are discussed in detail in the next section.

4. Security Analysis

In Section 3 we have proposed and discussed two transportation schemes for mobile agents. Here, we intend to examine the security issues related to the proposed schemes.

Preventing the confidential information from leaking out

All the transported messages are encrypted in the proposed schemes. Hence, without the decryption key, it will not be possible for any malicious attacker to wiretap the transported contents. Any confidential information, such as what agent will be dispatched or where the agent works, will not leak out. The confidentiality is not a problem at all.

Attaining integrity and authentication

In the proposed schemes, asymmetric cryptography (i.e., RSA) is used to produce a signature of the transported message in the preceding steps. It is a powerful tool to authenticate the sender of the message and to ensure the integrity of the transported message. Because only the owner knows the private key, no attacker can acquire the correct signature. If a malicious attacker wants to forge a transported message or modify the content of the message, the receiver can check it out. In the preceding steps, we use symmetric cryptography to encrypt the transported message. Authentication and integrity can both be attained, because only the valid sender knows the session key. If the received message can be decrypted and turned back to be the meaningful message by using the same session key, the receiver can ensure the validity of the received message. Furthermore, the agent code is previously stored on the trust server. The trust server has to manage its authentication.

In the second scheme, the agent authentication is done through a certificate. The certificate is issued by the trust server and signed with the trust server’s private key. Therefore, integrity and authentication can be guaranteed.

Resisting the replay attack

To resist the replay attack, the *Noise* and session key for a certain point of time are different from those for the next moment in our schemes. When an attacker replays previously intercepted message, the attack will not work because the receiver can detect that the *Noise* and session key were used before. Therefore, the proposed schemes are secure against the replay attack.

Providing the property of Non-repudiation

Non-repudiation is an important property when the mobile agent is used in business applications. In order to ensure this property, we use digital signature to achieve the objective. In the digital signature scheme, only the owner knows the private key and thus can produce a correct signature. Therefore, the user cannot deny sending the request for launching the mobile agent. Furthermore, all of the transferred messages must go through the trust server so that the trust server can record their message contexts to provide non-repudiation.

Ensuring host's security

In the first scheme, the agent is verified and encrypted by the trust server before arriving at the target host. In the second scheme, the target host can check its legitimacy by verifying the certificate. Furthermore, the agent transfer process is encrypted using the session key. No malicious attackers can attack the agent during the transferring process. Therefore, the host does not have to worry about any attack. On the other hand, the trust server can trail the audit to find anything suspicious. These policies can ensure the security of the hosts.

5. Conclusions and Future Work

In this paper, two secure transportation schemes for mobile agents have been proposed. In the first of the proposed schemes, we use both symmetric and asymmetric cryptography to accomplish our goal. It is very efficient, but the trust server has to bear a heavy load. In the second scheme, we use the certificate technique to accomplish the same goal. It reduces the load on the trust server but is less efficient. The tradeoff should be made according to the system's requirements. For small networks or few agents, the first scheme is a better choice. Otherwise, the second scheme will be superior. Furthermore, according to the security analysis that has been presented, we have found our protocols useful in mobile agent communication and agent code delivery. However, there are more security problems that have to be addressed. In the future, we will continue the efforts in this domain, especially in electronic commerce and enterprise information management.

Acknowledgment. The authors wish to thank the anonymous referees for their suggestions to improve this paper. This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC90-2213-E-324-005.

Notes:

[1] P. Jorge, L.M. Silva, and J.G. Silva, "Security mechanisms for using mobile agents in electronic commerce," in *Proceedings of the 18th IEEE symposium on Reliable Distributed Systems* (1999): 378-383.

[2] P. Maes, R. Guttman, and A. Moukas, "Agents that buy and sell," *Communications of the ACM* 42 (March 1999): 81-91.

[3] M.S. Greenberg, J.C. Byington, and D.G. Harper, "Mobile agents and security," *IEEE Communications Magazine* 36 (July 1995): 76-85.

[4] Ahmed Karmouch, "Guest editorial: Mobile software agents for telecommunications," *IEEE Communications Magazine* (July 1998).

[5] F. Hohl, "A model of attacks malicious hosts against mobile agents," in *Proceedings of the 4th Workshop on Mobile Object Systems: Secure Internet Mobile Computations* (1998).

[6] Min-Shiang Hwang and W.P. Yang, "Conference key distribution protocols for digital mobile communication systems," *IEEE Journal on Selected Areas in Communications* 13 (February 1995): 416-420.

[7] M.S. Hwang, I.C. Lin, and Eric J.L. Lu, "A secure nonrepudiable threshold proxy signature scheme with known signers," *International Journal of Informatica* 11, 2 (2000): 1-8.

[8] Min-Shiang Hwang and Chii-Hwa Lee, "Secure access schemes in mobile database systems," *European Transactions on Telecommunications* 12, 4 (2001): 303-310.

I.-C. LIN received a B.Sc. degree in Computer and Information Sciences from the Tung Hai University, Taichung, Taiwan, Republic of China, in 1998; a M.Sc.degree in Information Management from the Chaoyang University of Technology, Taiwan, in 2000. He is currently pursuing his Ph.D. in Computer Science and Information Engineering from the National Chung Cheng University. His current research interests include electronic commerce, information security, cryptography, and mobile communications.

H.-H. OU received his B.Sc. and M.Sc. degrees in Information Management from the Chaoyang University of Technology, Taiwan, Republic of China, in 1999 and 2001 respectively. His current research interests include mobile agent, information security, and cryptography.

M.-S. HWANG received a B.Sc. in Electronic Engineering from the National Taipei Institute of Technology, Taipei, Taiwan, Republic of China, in 1980; a M.Sc. degree in Industrial Engineering from the National Tsing Hua University, Taiwan, in 1988; and a Ph.D. in Computer and Information Science from the National Chiao Tung University, Taiwan, in 1995. He has also studied Applied Mathematics at the National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in the field "Electronic Engineer" in 1988. He has also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a project leader for research in computer security at TL in July 1990. He obtained the 1997, 1998, and 1999 Distinguished Research Awards of the National Science Council of the Republic of China. He is currently a professor and chairman of the Department of Information Management, Chaoyang University of Technology, Taiwan, ROC. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include database and data security, cryptography, image compression, and mobile communications. For correspondence: Prof. Min-Shiang Hwang, Department of Information Management, Chaoyang University of Technology, 168 Gifeng E. Rd., Wufeng, Taichung County, Taiwan 413, R.O.C., Fax: 886-4-3742337. E-mail: mshwang@mail.cyut.edu.tw.