Two Secure Transportation Schemes for Mobile Agent *

Iuon-Chang Lin‡ Hsia-Hung Ou† Min-Shiang Hwang†

Department of Information Management†
Chaoyang University of Technology
168 Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
Email: mshwang@mail.cyut.edu.tw
Fax: 886-4-3742337

Department of Computer Science ‡
and Information Engineering,
National Chung Cheng University,
Chaiyi, Taiwan, R. O. C.

March 7, 2002

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC90-2213-E-324-005.

†Responsible for correspondence: Prof. Min-Shiang Hwang

Two Secure Transportation Schemes for Mobile Agent

Abstract

The mobile agent is a new emerging popular research topic. Nowadays, the mobile agent is widely spread and implemented on the Internet. Application areas of the mobile agent include electronic commerce, electronic marketing, and enterprise information systems, etc. Since all the information about the mobile agent is transported over the Internet, the security policies become very important. However, the transportation security is usually neglected. In this paper, we propose two secure transportation schemes for the mobile agent. They can prevent all possible attacks during the process of transporting agents. Furthermore, users can choose the best transportation scheme according to the system's scale.

Keywords: Mobile agent, privacy, security, secure transportation.

1 Introduction

A great number of research topics have been focused on the mobile agent. All the experts in this domain wish to study the relevant technologies and thus enhance business activities [6, 8]. In the information era, the Internet prevails all over the world; it is both open and general.

The concept of a mobile agent has been offered for the use on the Internet. A mobile agent is software that acts on behalf of a user or another software. It has the following features: (1) it is autonomous; (2) it has one or more goals; (3) it has a scope of competence; and (4) it may, or may not, collaborate and communicate with other software and users [1]. In order to do its job, it is able to transport from a source host to a target host on a network under its own control [7]. However, many security threats and attacks are derivative. When a mobile agent transports between a series of hosts, it may have 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 in the management of mobile agents becomes essential. So far, many researches as to mobile agent security have been focused on the following topics [2]:

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, few researches have been focused on transportation security, which is in fact a very important topic in the mobile agent system, especially when it comes to business. When a user makes a work request, 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 aim at the transportation security for the mobile agent. When the mobile agent is transported between the distributed hosts, there are several secure issues [5, 3, 4] that we must carefully pay attention to:

- Confidentiality: In order to protect the privacy from being violated, all of the transported messages are encrypted. No malicious attacker can wiretap the transportation 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 scare malicious users and agents away from attacking.
- Non-repudiation: The system provides the property of non-repudiation. It can prevent the user from deny 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 trail the audit to find any thing 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 conclusion in Section 5.

2 A Basic Framework of the Proposed Scheme

In this section, a basic framework of our proposed protocol is to be introduced. Our method is based on the 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 following:

1. Source Host:

   A host that owns mobile agents and makes requests for performing jobs to the trust server.

2. Trust Server:

   It is a trusted third party. It supports all of the requests for secure transportation between source hosts and target hosts. When TS receives

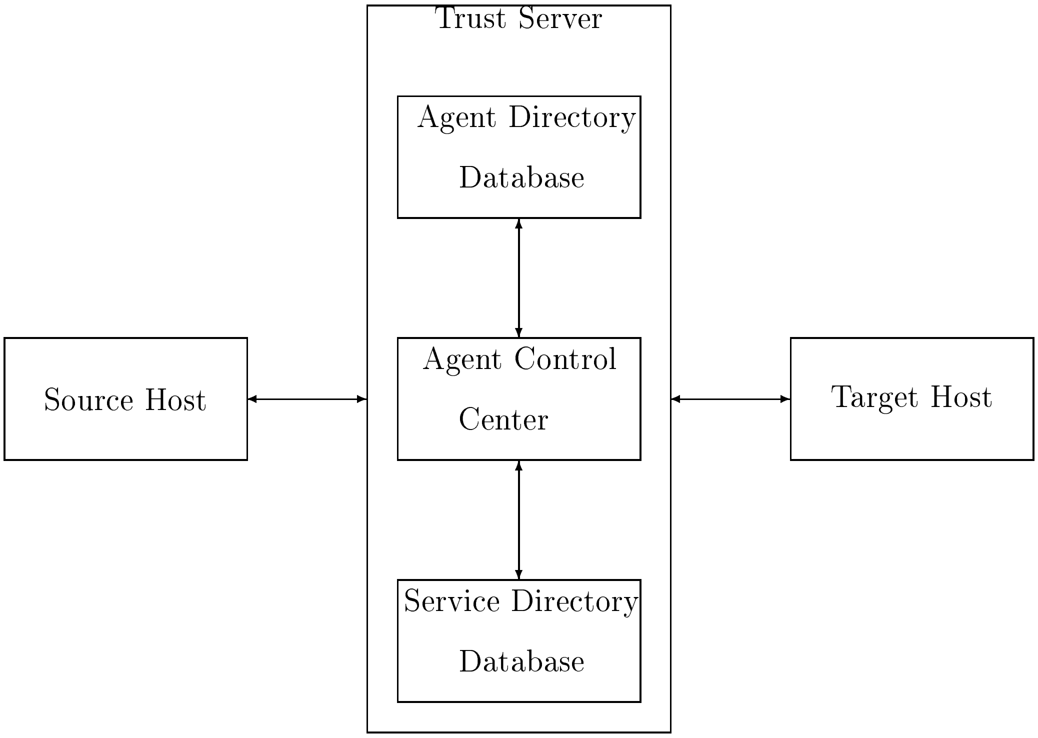

Figure 1: Our basic framework

the request for an agent, it verifies the validity of the request and then dispatches the requested agent to a certain target host. There are three modules in the trust server as follows:

(a) Agent Directory Database:

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

(c) Service Directory Database:

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:

A host that an agent is sent to in order for the jobs to be done.

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 the agents must register and leave the records in the agent directory before starting the mobile agent. The agent directory records the information as to which agents belongs 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, we shall propose two secure transportation schemes for the mobile agent. 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 as follows.

TS: Trust Server;

SH: Source Host;

TH: Target Host;

A: An agent;

ID_i: The identity of an entity i;

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

$D_{SK_i}[\cdots]$: An decryption function or digital signature product function using asymmetric cryptosystems, such as RSA, with the entity i's private key being SK_i;

$F_{K_j}[\cdots]$: The encryption function using symmetric cryptosystems, such as DES, with the jth 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 procedures of our first scheme are described 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 for confidential communication, the request must be encrypted using TS's public key PK_{TS}. The main purpose of this step is for SH to inform TS which agent will be launched.

Step 2. Upon receiving the above messages, TS decrypts them and verify 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 by way of the Agent Directory Database. Then, TS searches the Service Directory Database for a suitable TH and generates

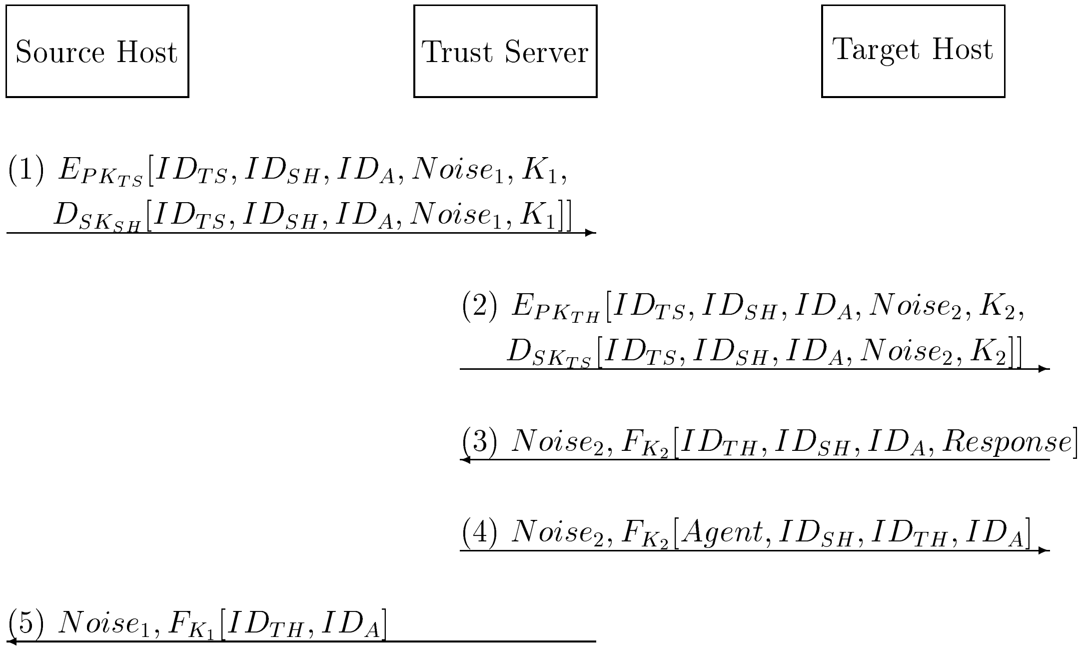

Figure 2: The first secure transportation scheme for the mobile agent

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 with the target host whether it provides the requested services.

Step 3. Target host verifies the validity of the received messages and records $Noise_2$ and its corresponding K_2. Then, it replies to the trust server with the answer. The answered messages are encrypted by using symmetric encryption function F with session key K_2.

Step 4. Trust server receives the answered messages 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 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 be leaked out when it is passed over the Internet, and SH can be sure 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. S-ince 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 purposes. 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 a second 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 $D_{SK_{TS}}(H(Agent) \bigoplus ID_{SH})$. The certificate can then be used to verify that the agent is a legal agent.

Scheme Two (Based on certificate):

We use the certificate to improve our first scheme. The procedures of our scheme 2 are described as follows:

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

Step 2. This step is also the same as the second step from the first scheme. TS checks with the target host whether it provides the requested services.

Step 3. Target host verifies the validity of the received messages. Then, it replies

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) $Noise_2, F_{K_2}[ID_{TH}, ID_{SH}, ID_A, Response, E_{PK_{SH}}[Noise_3, K_3, D_{SK_{TH}}[Noise_3, K_3]]]$

(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 the mobile agent

the trust server with the answer. 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.

Step 4. Trust server receives the answered messages 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 the 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 legality, 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 step 1 with 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 the 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 in process. 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 deliver 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 little agents, the first scheme is the best choice. Otherwise, the second scheme can support the largest 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. The detailed security analysis is discussed in the next section.

4 Security Analysis

We have described our proposed transportation schemes for the mobile agent in Section 3. In this section, the security of the proposed schemes is to be examined.

1. Preventing the confidential information from leaking out: All the transported messages are encrypted in our schemes. Hence, without the decryption key, it is of no use for any malicious attacker to wiretap the transported contents. The 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.

2. Attaining integrity and authentication: In our schemes, we use asymmetric cryptography (i.e., RSA) 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 come by 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 rear steps, we use symmetric cryptography to encrypt the transported message. The 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 in the trust server. The trust server has to manage its

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

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

4. 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 the 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.

5. 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 legality 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 procedure. 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 suspect. These policies can ensure the security of the hosts.

5 Conclusions and Future Work

In this paper, we have proposed two secure transportation schemes for the mobile agent. In our first scheme, we use symmetric cryptography and asymmetric cryptography as well to accomplish our goal. It is very efficient, but the trust server has to bear a heavy load. In our 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 little agents, the first scheme is a better choice. Otherwise, the second scheme will be superior. Furthermore, according to the security analysis that we have presented, we have found our protocols helpful to mobile agent communication and agent code delivery. However, there are more security problems we should take into account than there were in our discussions. In the future, we will continue our efforts in this domain, especially in the mobile agent technique applied in electronic commerce and enterprise information management.

References

[1] M.S. Greenberg, J.C. Byington, and D.G. Harper, "Mobile agents and security," *IEEE Communications Magazine*, vol. 36, pp. 76–85, July 1995.

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

[3] 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*, vol. 11, no. 2, pp. 1–8, 2000.

[4] Min-Shiang Hwang and Chii-Hwa Lee, "Secure access schemes in mobile database systems," *European Transactions on Telecommunications*, vol. 12, no. 4, pp. 303–310, 2001.

[5] Min-Shiang Hwang and W. P. Yang, "Conference key distribution protocols for digital mobile communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 416–420, Feb. 1995.

[6] P. Jorge, L.M. Silva, and J.G. Silva, "Security mechanisms for using mobile agents in electronic commerce," in *The 18th IEEE symposium on Reliable Distributed Systems*, pp. 378–383, 1999.

[7] Ahmed Karmouch, "Guest editorial mobile software agents for telecommunications," *IEEE Communications Magazine*, July 1998.

[8] P. Maes, R. Guttman, and A. Moukas, "Agents that buy and sell," *Communications of the ACM*, vol. 42, pp. 81–91, March 1999.