Mobile Agent Protection for M-Commerce

Sheng-Uei Guan
Brunel University, UK

INTRODUCTION

The introduction of the mobile Internet is probably one of the most significant revolutions of the 20th century. With a simple click, one can connect to almost every corner of the world thousands of kilometers away. This presents a great opportunity for m-commerce. Despite its many advantages over traditional commerce, m-commerce has not taken off successfully. One of the major hindrances is security. The focus of this article is secure transport of mobile agents. A mobile agent is useful for handheld devices like a palmtop or PDA. Such m-commerce devices usually have limited computing power. It would be useful if the users of such devices could send an intelligent, mobile agent to remote machines to carry out complex tasks like product brokering, bargain hunting, or information collection.

When it comes to online transactions, security becomes the primary concern. The Internet was developed without too much security in mind. Information flows from hubs to hubs before it reaches its destination. By simply tapping into wires or hubs, one can easily monitor all traffic transmitted. For example, when Alice uses her VISA credit card to purchase an album from Virtual CD Mall, the information about her card may be stolen if it is not carefully protected. This information may be used maliciously to make other online transactions, thus causing damage to both the card holder and the credit card company.

Besides concerns on security, current m-commerce lacks the intelligence to locate the correct piece of information. The Internet is like the world's most complete library collections unsorted by any means. To make things worse, there is no competent librarian that can help readers locate the book wanted. Existing popular search engines are attempts to provide librarian assistance. However, as the collection of information is huge, none of the librarians are competent enough at the moment.

An intelligent agent is one solution to providing intelligence in m-commerce. But having an agent that is intelligent is insufficient. There are certain tasks that are unrealistic for agents to perform locally, especially those that require a large amount of information. Therefore, it is important to equip intelligent agents with roaming capability.

Unfortunately, with the introduction of roaming capability, more security issues arise. As the agent needs to move among external hosts to perform its tasks, the agent itself becomes a target of attack. The data collected by agents may be modified, the credit carried by agents may be stolen, and the mission statement on the agent may be changed. As a result, transport security is an immediate concern to agent roaming. The SAFE (*s*ecure roaming *a*gent *f*or *e*-commerce) transport protocol is designed to provide a secure roaming mechanism for intelligent agents. Here, both general and roaming-related security concerns are addressed carefully. Furthermore, several protocols are designed to address different requirements. An m-commerce application can choose the protocol that is most suitable based on its need.

BACKGROUND

There has been a lot of research done on the area of intelligent agents. Some literature (Guilfoyle, 1994; Johansen, Marzullo, & Lauvset, 1999) only propose certain features of intelligent agents, some attempt to define a complete agent architecture. Unfortunately, there is no standardization in the various proposals, resulting in vastly different agent systems. Efforts are made to standardize some aspects of agent systems so that different systems can inter-operate with each other. Knowledge representation and exchange is one of the aspects of agent systems for which KQML (Knowledge Query and Manipulation Language; Finin, 1993) is one of the most widely accepted standards. Developed as part of the *Knowledge Sharing Effort,* KQML is designed as a high-level language for runtime exchange of information between heterogeneous systems. Unfortunately, KQML is designed with little security considerations because no security mechanism is built to address common security concerns, not to mention specific security concerns introduced by mobile agents. Agent systems using KQML will have to implement security mechanisms on top of KQML to protect themselves.

While KQML acts as a sufficient standard for agent representation, it does not touch upon the security aspects of agents. In an attempt to equip KQML with built-in security mechanisms, Secret Agent is proposed by Thirunavukkarasu, Finin and Mayfield (1995).

Another prominent transportable agent system is Agent TCL developed at Dartmouth College (Gray, 1997; Kotz et al., 1997). Agent TCL addresses most areas of agent transport by providing a complete suite of solutions. It is probably one of the most complete agent systems under research. Its security mechanism aims at protecting resources and

the agent itself. In terms of agent protection, the author acknowledges that "it is clear that it is impossible to protect an agent from the machine on which the agent is executing … it is equally clear that it is impossible to protect an agent from a resource that willfully provides false information" (Gray, 1997). As a result, the author "seeks to implement a verification mechanism so that each machine can check whether an agent was modified unexpectedly after it left the home machine" (Gray, 1997). The other areas of security, like non-repudiation, verification, and identification, are not carefully addressed.

Compared with the various agent systems discussed above, SAFE is designed to address the special needs of m-commerce. The other mobile agent systems are either too general or too specific to a particular application. By designing SAFE with m-commerce application concerns in mind, the architecture will be suitable for m-commerce applications. The most important concern is security, as discussed in previous sections. Due to the nature of m-commerce, security becomes a prerequisite for any successful m-commerce application. Other concerns are mobility, efficiency, and interoperability. In addition, the design allows certain flexibility to cater to different application needs.

MAIN FOCUS OF THE ARTICLE

As a prerequisite, each SAFE entity must carry a digital certificate issued by SAFE Certificate Authority, or SCA. The certificate itself is used to establish the identity of a SAFE entity. Because the private key to the certificate has signing capability, this allows the certificate owner to authenticate itself to the SAFE community. An assumption is made that the agent private key can be protected by function hiding (Thomas, 1998). Other techniques were also discussed in the literature (Bem, 2000; Westhoff, 2000), but will not be elaborated in this article.

From the host's viewpoint, an agent is a piece of foreign code that executes locally. In order to prevent a malicious agent from abusing the host resources, the host should monitor the agent's usage of resources (e.g., computing resources, network resources). The agent receptionist will act as the middleman to facilitate and monitor agent communication with the external party.

General Message Format

In SAFE, agent transport is achieved via a series of message exchanges. The format of a general message is as follows:

SAFE Message = Message Content + Timestamp + Sequence Number + MD(Message Content + Timestamp + Sequence Number) + Signature(MD)

The main body of a SAFE message comprises message content, a timestamp, and a sequence number. The message content is defined by individual messages. Here MD stands for the Message Digest function. The first MD is the function applied to Message Content, Timestamp, and Sequence Number to generate a message digest. The second MD in the equation is the application of digital signature to the message digest generated. A timestamp contains the issue and expiry time of the message.

To prevent replay attack, message exchanges between entities during agent transport are labeled according to each transport session. A running sequence number is included in the message body whenever a new message is exchanged.

In order to protect the integrity of the main message body, a message digest is appended to the main message. The formula of the message digest is as follows:

Message Digest = MD5(SHA(message_body) + message_body)

Here SHA (Secure Hash Algorithm) stands for a set of related cryptographic hash functions. The most commonly used function, SHA-1, is employed in a large variety of security applications. The message digest alone is not sufficient to protect the integrity of a SAFE message. A malicious hacker can modify the message body, and recalculate the value of message digest using the same formula and produce a seemingly valid message digest. To ensure the authenticity of the message, a digital signature on the message digest is generated for each SAFE message. In addition to ensuring message integrity, the signature serves as a proof for non-repudiation as well.

If the message content is sensitive, it can be encrypted using a symmetric key algorithm (e.g., Triple DES). The secret key used for encryption will have to be decided at a higher level.

To cater for different application concerns, three transport protocols are proposed: supervised agent transport, unsupervised agent transport, and bootstrap agent transport. These three protocols will be discussed in the following sections in detail.

Supervised Agent Transport

Supervised agent transport is designed for applications that require close supervision of agents. Under this protocol, an agent must request a roaming permit from its owner or butler before roaming. The owner has the option to deny the roaming request and prevent its agent from roaming to undesirable hosts. Without the agent owner playing an active role in the transport protocol, it is difficult to have tight control over agent roaming.

The procedure for supervised agent transport is shown in Figure 1.