Mobile Agent Authentication and Authorization in E-Commerce

Sheng-Uei Guan
National University of Singapore, Singapore

INTRODUCTION

With the increasing worldwide usage of the Internet, electronic commerce (e-commerce) has been catching on fast in a lot of businesses. As e-commerce booms, there comes a demand for a better system to manage and carry out transactions. This has led to the development of agent-based e-commerce. In this new approach, agents are employed on behalf of users to carry out various e-commerce activities.

Although the tradeoff of employing mobile agents is still a contentious topic (Milojicic, 1999), using mobile agents in e-commerce attracts much research effort, as it may improve the potential of their applications in e-commerce. One advantage of using agents is that communication cost can be reduced. Agents traveling and transferring only the necessary information save the bandwidth and reduce the chances of network clogging. Also, users can let their agents travel asynchronously to their destinations and collect information or execute other applications while they can disconnect from the network (Wong, 1999).

Although agent-based technology offers such advantages, the major factor that is holding people back from employing agents is still the security issues involved. On the one hand, hosts cannot trust incoming agents belonging to unknown owners, because malicious agents may launch attacks on the hosts and other agents. On the other hand, agents may also have concerns on the reliability of hosts and will be reluctant to expose their secrets to distrustful hosts.

To build bilateral trust in an e-commerce environment, the authorization and authentication schemes for mobile agents should be well designed. Authentication checks the credentials of an agent before processing the agent's requests. If the agent is found to be suspicious, the host may decide to deny its service requests. Authorization refers to the permissions granted for the agent to access whichever resource it requested.

In our previous work, we have proposed a SAFER (Secure Agent Fabrication, Evolution & Roaming) architecture (Zhu, 2000), which aims to construct an open, dynamic and evolutionary agent system for e-commerce. We have already elaborated agent fabrication, evolution, and roaming in Guan (1999, 2001, 2002), Wang (2001), and Zhu (2001). This article gives an overview of the authentication and authorization issues on the basis of the SAFER architecture.

BACKGROUND

Many intelligent agent-based systems have been designed to support various aspects of e-commerce applications in recent years, for example: Kasbah (Chavez, 1998), Minnesota AGent Marketplace Architecture (MAGMA) (Tsvetovatyy, 1997), and MAgNet (Dasgupta, 1999). Unfortunately, most current agent-based systems such as Kasbah and MAGMA are serving only stationary agents. Although MAgNet employs mobile agents, it does not consider security issues in its architecture.

D'Agents (Gray, 1998) is a mobile agent system, which employs the PKI for authentication purposes, and uses the RSA (Rivest, Shamir, & Adleman, 1978) public key cryptography (Rivest et al., 1978) to generate the public-private key pair. After the identity of an agent is determined, the system decides what access rights to assign to the agent and sets up the appropriate execution environment for the agent.

IBM Aglets (Lange, 1998; Ono, 2002) are Java-based mobile agents. Each aglet has a globally unique name and a travel itinerary (wherein various places are defined as context in IBM Aglets). The context owner is responsible for keeping the underlying operating system secure, mainly protecting it from malicious aglets. Therefore, he or she will authenticate the aglet and restrict the aglet under the context's security policy.

Ajanta is also a Java-based mobile agent system (Karnik, 1999, 2001, 2002) employing a challenge-response based authentication protocol. Each entity in Ajanta registers its public key with Ajanta's name service. A client has to be authenticated by obtaining a ticket from the server. The Ajanta Security Manager grants agents permissions to resources based on an access control list, which is created using users' Uniform Resource Names (URNs).

iJADE (intelligent Java Agent Development Environment) (Lee, 2002) provides an intelligent agent-based platform in the e-commerce environment. This system can provide fully automatic, mobile and reliable user authentication.

Under the public key infrastructure (PKI), each entity may possess a public-private key pair. The public key is known to all, while the private key is only known to the

key owner. Information encrypted with the public key can only be decrypted with the corresponding private key. In the same note, information signed by the private key can only be verified with the corresponding public key (Rivest, 1978; Simonds, 1996). The default algorithm that generates the key pairs is the digital signature algorithm (DSA), working in the same way as a signature on a contract. The signature is unique, so that the other party can be sure that you are the only person who can produce it.

MAIN THRUST OF THE ARTICLE

This article presents an overview of the architecture based on SAFER (Secure Agent Fabrication, Evolution & Roaming) (Zhu, 2000) to ensure a proper authentication and authorization of agent. Here, the public key infrastructure (PKI) is used as the underlying cryptographic scheme. Also, agents can authenticate the hosts to make sure that they are not heading to a wrong place. According to the level of authentication that the incoming agent has passed, the agent will be categorized and associated with a relevant security policy during the authorization phase. The corresponding security policy will be enforced on the agent to restrict its operations at the host. The prototype has been implemented with Java.

Design of Agent Authentication and Authorization

Overview of the SAFER Architecture

The SAFER architecture comprises various communities and each community consists of the following components (see Figure 1): Agent Owner, Agent Factory, Agent Butler, Community Administration Center, and so forth. The Agent Owner is the initiator in the SAFER environment, and requests the Agent Factory to fabricate the agents it requires. The Agent Butler is a representative of the Agent Owner authorized by the owner to coordinate the agents that are dispatched. Owner can go offline after dispatching his or her agents, and thereafter the butler can take over the coordination of the agents. The Agent Factory fabricates all the agents. This is the birthplace of agents and is thus considered a good source to check malicious agents. The Community Administration Center (CAC) is the administrative body, which has a roster that keeps the data of the agents that are in the community. It also collects information, such as addresses of new sites that agents can roam to.

Agent Structure and Cryptographic Schemes

In SAFER, mobile agents have a uniform structure. The agent credentials (hard-coded into the agent (Guan, 2000, 2001)) are the most important part of the agent body and are immutable. This part includes FactoryID, AgentID, Expiry Date, and so forth. The Agent Factory then signs this immutable part. When the receiving host accepts an agent, it can verify with the Agent Factory's public key whether the agent's credentials have been modified. The mutable part of the agent includes the Host Trace, which stores a list of names of the hosts that the agent has visited so far. Upon checking, if any distrusted host is found, a host may decide not to trust this agent and impose a stricter security policy on it.

In SAFER, the main cryptographic technology used is the PKI. The public keys are stored in a common database

Figure 1. SAFER architecture