Mobile Agent Authentication and Authorization

Sheng-Uei Guan
Brunel University, UK

INTRODUCTION

With the increasing usage of the Internet, electronic commerce (e-commerce) has been catching on fast in a lot of business areas. As e-commerce booms, there comes a demand for a better system to manage and carry out transactions. This leads 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 under debate (Milojicic, 1999), using mobile agents in e-commerce attracts much research effort, as it may improve the potential of their applications in e-commerce (Guan & Yang, 1999, 2004). One advantage of using agents is that communication cost can be reduced. Agents traveling and transferring only necessary information saves network bandwidth and reduces the chances of network congestion. Also, users can schedule their agents to travel asynchronously to the destinations and collect information or execute other applications, while they can disconnect from the network (Wong, Paciorek, & Moore, 1999).

Although agent-based technology offers such advantages, the major factor holding people back from employing agents is still the security issues involved. On 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 designed well. Authentication checks the credentials of an agent before processing an 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 resources it requested.

BACKGROUND

Many intelligent agent-based systems have been designed to support various aspects of e-commerce applications in recent years, for example, Kasbah (Chavez & Maes, 1998), Minnesota AGent Marketplace Architecture (MAGMA) (Tsvetovatyy, Mobasher, Gini, & Wieckowski, 1997), and MAgNet (Dasgupta, Narasimhan, Moser, & Melliar-Smith, 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, Kotz, Cybenko, & Rus, 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 & Oshima, 1998; Ono & Tai, 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 will authenticate the aglet and restrict the aglet under the context's security policy.

Ajanta is also a Java-based mobile agent system (Karnik & Tripathi, 1999, 2001; Karnik, 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 et al., 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 the other party can be sure that you are the only person who can produce it.

Secure agent fabrication, evolution & roaming (SAFER) was proposed as an open architecture (Zhu, Guan, & Yang, 2000) for an evolutionary agent system for e-commerce. Issues such as agent fabrication, evolution, and roaming were elaborated in Guan, Tan, and Hua (2004), Guan and Yang (1999), Guan and Zhu (2002, 2004), Guan, Zhu, and Ko, (2000), Wang, Guan, and Chan (2001), and Zhu and Guan (2001). The next section provides more details on SAFER. This article gives an overview of the authentication and authorization issues on the basis of the SAFER architecture.

Authentication and Authorization

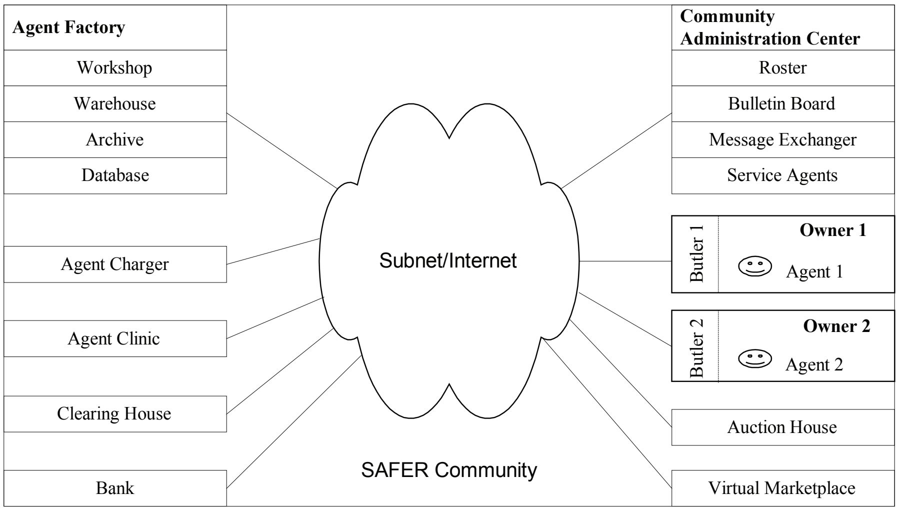
This section presents an overview of the architecture based on secure agent fabrication, evolution & roaming (SAFER) (Zhu et al., 2000) to ensure a proper authentication and authorization of agent. Here, the public key infrastructure (PKI) is used as the underlying cryptographic scheme. Also, agents can authenticate hosts to make sure they are not heading to a wrong place. According to the level of authentication that an 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,

Figure 1. SAFER architecture


Agent Factory
Workshop
Warehouse
Archive
Database
Agent Charger
Agent Clinic
Clearing House
Bank
Subnet/Internet
SAFER Community
Community Administration Center
Roster
Bulletin Board
Message Exchanger
Service Agents
Butler 1
Owner 1
Agent 1
Butler 2
Owner 2
Agent 2
Auction House
Virtual Marketplace