Protection of Mobile Agent Data

Sheng-Uei Guan
Brunel University, UK

INTRODUCTION

One hindrance to the widespread adoption of mobile agent technology is the lack of security. Security will be the issue that has to be addressed carefully if a mobile agent is to be used in the field of electronic commerce. SAFER—or Secure Agent Fabrication, Evolution, and Roaming—is a mobile agent framework that is specially designed for the purpose of electronic commerce (Zhu, Guan, Yang, & Ko, 2000; Guan & Hua, 2003; Guan, Zhu, & Maung, 2004). Security has been a prime concern from the first day of our research (Guan & Yang, 1999, 2002; Yang & Guan, 2000). By building strong and efficient security mechanisms, SAFER aims to provide a trustworthy framework for mobile agents, increasing trust factors to end users by providing the ability to trust, predictable performance, and a communication channel (Patrick, 2002).

Agent integrity is one such area crucial to the success of agent technology (Wang, Guan, & Chan, 2002). Despite the various attempts in the literature, there is no satisfactory solution to the problem of data integrity so far. Some of the common weaknesses of the current schemes are vulnerabilities to revisit attack when an agent visits two or more collaborating malicious hosts during one roaming session and illegal modification (deletion/insertion) of agent data. Agent Monitoring Protocol (AMP) (Chionh, Guan, & Yang, 2001), an earlier proposal under SAFER to address agent data integrity, does address some of the weaknesses in the current literature. Unfortunately, the extensive use of PKI technology introduces too much overhead to the protocol. Also, AMP requires the agent to deposit its data collected to the agent owner/butler before it roams to another host. While this is a viable and secure approach, the proposed approach—Secure Agent Data Integrity Shield (SADIS)—will provide an alternative by allowing the agent to carry the data by itself without depositing it (or the data hash) onto the butler.

Besides addressing the common vulnerabilities of current literature (revisit attack and data modification attack), SADIS also strives to achieve maximum efficiency without compromising security. It minimizes the use of PKI technology and relies on symmetric key encryption as much as possible. Moreover, the data encryption key and the communication session key are both derivable from a key seed that is unique to the agent's roaming session in the current host. As a result, the butler can derive the communication session key and data encryption key directly. Another feature in SADIS is strong security.

Most of the existing research focuses on detecting integrity compromise (Esparza, Muñoz, Soriano, & Forné, 2006) or on bypassing integrity attacks by requiring the existence of a cooperating agent that is carried out within a trusted platform (Ouardani, Pierre, & Boucheneb, 2006), but which neglected the need to identify the malicious host. With SADIS, the agent butler will not only be able to detect any compromise to data integrity, but to identify the malicious host effectively.

BACKGROUND

Agent data integrity has been a topic of active research in the literature for a while. SADIS addresses the problem of data integrity protection via a combination of techniques discussed by Borselius (2002): execution tracing, encrypted payload, environmental key generation, and undetachable signature.

One of the recent active research works is the security architecture by Borselius, Hur, Kaprynski, and Mitchell (2002). Their security architecture aims at defining a complete security architecture designed for mobile agent systems. It categorizes security services into the following: agent management and control, agent communications service, agent security service, agent mobility service, and agent logging service. SADIS addresses the agent communication service as well as agent security services (integrity protection), while previous research on SAFER addresses agent mobility service.

While many of the security services are still under active research, the security mechanisms for protecting agents against malicious hosts were described by Borselius, Mitchell, and Wilson (2001). Their paper proposes a threshold scheme to protect mobile agents. Under the mechanism, a group of agents is dispatched to carry out the task, each agent carrying a vote. Each agent is allowed to contact a merchant independently and gathers bids based on the given criteria. Each agent votes for the best bid (under a trading scenario) independently. If more than n out of m ($m > n$) agents vote for the transaction, the agent owner will agree to the transaction.

Such a mode of agent execution effectively simplifies agent roaming by allowing one agent to visit one merchant only. While the approach avoids the potential danger of having the agent compromised by the subsequent host, it does not employ a mechanism to protect the agent against the current host. Most important of all, the threshold mechanism's security is based on the probability that no more than n hosts out of m are malicious. In other words, the security is established based on probability. Different from this approach, SADIS's security is completely based on its own merits without making any assumption about probability of hosts being benign or malicious. This is because the author believes that in an e-commerce environment, security should not have any dependency on probability.

Other than the research by Borselius, there are related works in the area. One such work on agent protection is SOMA, or Secure and Open Mobile Agent, developed by Corradi, Cremonini, Montanari, and Stefanelli (1999). SOMA is a Java-based mobile agent framework that provides for scalability, openness, and security on the Internet. One of the research focuses of SOMA is to protect the mobile agent's data integrity. To achieve this, SOMA makes use of two mechanisms: Multi Hop (MH) Protocol and Trusted Third Party (TTP) Protocol. MH protocol works as follows. At each intermediate site the mobile agent collects some data and appends them to the previous ones collected. Each site must provide a short proof of the agent computation, which is stored in the agent. Each proof is cryptographically linked with the ones computed at the previous sites. There is a chaining relation between proofs. When the agent moves back to the sender, the integrity of the chained cryptographic proofs is verified allowing the sender to detect any integrity violation.

The advantage of MH protocol is that it does not require any trusted third party or even the agent butler for its operation. This is a highly desirable feature for agent integrity protection protocol. Unfortunately, MH protocol does not hold well against revisit attack when the agent visits two or more collaborating malicious hosts during one roaming session (Chionh et al., 2001). Roth (2001) provides more detailed descriptions on potential flaws of the MH protocol.

Another agent system that addresses data integrity is Ajanta (Tripathi, 2002). Ajanta is a platform for agent-based application on the Internet developed in the University of Minnesota. It makes use of an append-only container for agent data integrity protection. The main objective is to allow the host to append new data to the container, but to prevent anyone from modifying the previous data without being detected. Similar to the MH protocol, such an append-only container suffers from revisit attack.

From these attacks on existing research, the importance of protecting agent itinerary is obvious. In SADIS, the agent's itinerary is implicitly updated in the agent butler during key seed negotiation. This prevents any party from modifying the itinerary recorded on the butler and guard against all itinerary-related attacks.

There is one recent research work on agent data integrity protection called One-Time Key Generation System (OKGS) researched at the Kwang-Ju Institute of Science and Technology, South Korea (Park, Lee, & Lee, 2002). OKGS does protect the agent data against a number of attack scenarios under revisit attack, such as data insertion attack and data modification attack to a certain extent. However, it does not protect the agent against deletion attack, as two collaborating malicious hosts can easily remove roaming records in between them.

Inspired by OKGS's innovative one-time encryption key concept, SADIS will extend this property to the communication between agent and butler as well. Not only the data encryption key is one time, but the communication session key is as well. Using efficient hash calculations, the dynamic communication session key can be derived separately by the agent butler and the agent with minimum overhead. Despite the fact that all keys are derived from the same session-based key seed, SADIS also ensures that there is little cor-