SIAS: A Secure Shopping Information Agent System

Anthony H. W. Chan, T. Y. Wong, Caris K. M. Wong, and Michael R. Lyu
Department of Computer Science and Engineering, the Chinese University of Hong Kong
Shatin, N.T., Hong Kong
Fax: (852) 2603-5024
{hwchan1, tywong,kmwong1,lyu}@cse.cuhk.edu.hk

ABSTRACT

In this paper, we build a *Shopping Information Agent System (SIAS)* based on mobile agent technology. We discuss possible security attacks by malicious hosts to agents in the system, and present our solutions to prevent these attacks. We analyze the security of our solutions, and evaluate the performance overhead introduced.

Keywords

Mobile agents, electronic commerce, information agents, security

1. INTRODUCTION

Mobile agents can bring benefits such as reduced network load and overcoming of network latency [1]. Nevertheless, security is one of the blocking factors of the development of these systems. The main unsolved security problem lies on the possible existence of malicious hosts that can manipulate the execution and data of agents [2].

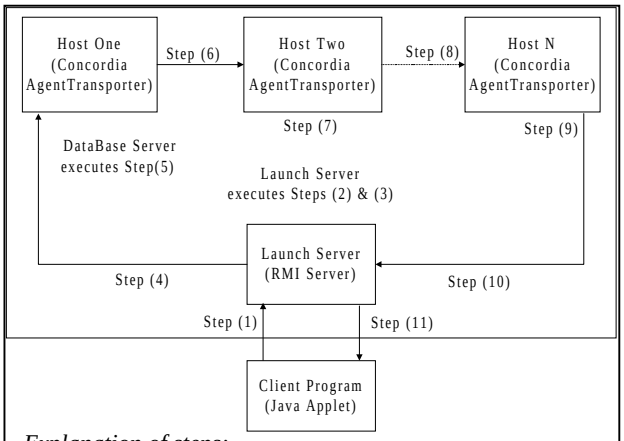
In this paper, we build a *Shopping Information Agent System (SIAS)* using the Concordia [3] architecture. The system is useful to collect and compare the prices of a set of products specified by users from different seller hosts in an electronic market. We address the security issues of the system, describe possible attacks by malicious hosts to the system, and devise and implement our solutions to protect the system against these attacks.

2. OVERVIEW OF SIAS

SIAS is a web-based mobile agent system that provides users with information of products for sale in an electronic marketplace. Advantages of SIAS include such properties as reduction of communication costs and delegation of tasks, which are the intrinsic advantages of a mobile agent system. It is written in the Java programming language and on top of the Concordia [3] application-programming interface (API).

The Concordia architecture, among different mobile agent platforms developed worldwide, is chosen for implementation of SIAS mainly because of the API simplicity, and the ability it allows to manipulate agent code execution, which is good for simulating malicious attack behaviors. The Java programming language is a natural choice for the Java-based Concordia API.

SIAS implements mobile agents to retrieve product information in an electronic market for users. An electronic market consists of hosts that sell products on the network. Each seller maintains a database that stores the prices and quantities in stock of different products available at that host. The control flow of the system is described by Figure 1.



Explanation of steps:

1. *Client program launches a request to the Launch Server object upon user input using Java Remote Method Invocation (RMI);*
2. *Launch Server creates an Agent object;*
3. *Launch Server initializes the agent with user-specified products and quantities, and the itinerary of agent;*
4. *Launch Server sends the agent to the network;*
5. *Database Server on Host One retrieves the required information for the incoming agent;*
6. *Agent goes to the next destination;*
7. *Database Server on Host Two repeats Step (5);*
8. *Agent goes to other hosts in the itinerary;*
9. *Database Server on each host repeats Step (5);*
10. *Launch Server receives the returning agent and calculates the cheapest purchasing combination;*
11. *Launch Server reports the cheapest purchasing combination to client program by Java RMI.*

Figure 1: Control flow of SIAS.

3. SECURITY DESIGN OF SIAS

Both host security and agent security [4] would be issues of SIAS. However, we are primarily interested in agent security. The Java sandbox model has largely simplified the host security part.

Three particular security problems for SIAS can be identified:

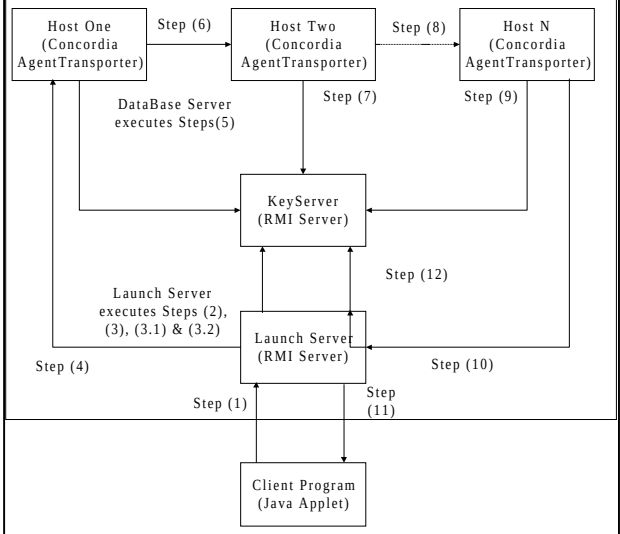
a) Modification of query products by a malicious host;

b) Modification of query quantities by a malicious host; and

c) Spying out and modification of query results.

This is only a subset of possible attacks. There are other attacks such as replaying of query results and masquerading of hosts. For the time being, we consider the four attacks only, for simplicity.

Having figured out the above system vulnerabilities, we develop a simple but original approach to protect agents in SIAS against attacks from malicious host, based on cryptographic techniques. We introduce a public-key infrastructure into the system, and require each host and agent in the system to possess a pair of keys for encryption and decryption. Therefore, each agent or host can encrypt or digitally sign the data items carried by an agent, and thus all the a) query products, b) query quantities, c) query results can be protected. Figure 2 illustrates changes to the system for security enhancement.



Explanation of additional / modified steps:

3.1. Launch Server generates a key pair for agent;

3.2. Launch Server signs the product and quantity lists for agents and registers the public key of agent to Key Server;

5. Database Server on Host One retrieves public key of agent from Key Server, and verify the signatures of product and quantity lists of agents. Then, the Database Server retrieves the required information for the incoming agent, signs the results using its own private key, and encrypt the results using the public key of agent;

11. Launch Server decrypts the query results, and verifies the signatures of the query results. It also detects change of agent itinerary by decrypting the chain of encrypted itineraries, and finally reports the cheapest purchasing combination to client program.

12. Launch Server deletes the public key entry of the finished agent from the key server.

Figure 2: Control flow of security-enhanced SIAS.

4. EVALUATION OF THE SECURE SIAS

The security of the additional measures lies mainly on the introduction of a key server that facilitates the use of public key cryptography. Assuming the key server, the communication channel with the key server are secure enough, and the keys are managed properly, the prevention of modification of the signed product and quantity lists of an agent by a malicious host is supported by the security of the RSA encryption algorithm. The time complexity for breaking the RSA cryptosystem depends on the length of the key in number of bits. The longer the key is, the more secure the system would be. In our implementation, we have chosen a key length of 128 bits. This would be sufficiently secure for our security purpose.

To evaluate the performance overhead introduced, we have tested the times for SIAS to launch a single agent with and without security measures. Round trip times (RTTs) required for an agent to travel around an electronic market of three hosts, with and without security enforcement, are measured respectively. Queries of different sizes (number of product items) have been tested.

Results show that, without security measures, the RTT for an agent to travel in SIAS does not change much over when the size of query varies. However, when security is enforced, the RTT increases very fast and linearly with the size of query. This can be explained by the extensive use of the RSA algorithm to encrypt and decrypt each item, which is time consuming, especially when the key is long. In addition, we simulate malicious hosts trying to modify the query list of an agent in SIAS, and measure the overheads introduced by the actions of malicious hosts. The results show that it takes a little bit more time for an agent to travel around when there is attack from malicious host. This suggests that the agent round trip time may be used as a measure for tampering detection.

5. CONCLUSION

We studied the technology of autonomous mobile agents and the problem of malicious hosts in a mobile agent system. We implemented SIAS as a sample application of mobile agents. We addressed some security problems of malicious hosts in SIAS, and developed a primitive approach to protect the agents. We analyzed the security of our approach, and believe it is strong enough for our application. We measured the performance overhead of the security measures, saw a trade-off between performance and security for SIAS, and learned that it takes time for a malicious host to attack an agent.

6. References

[1] Danny B. Lange and Mitsuru Oshima. "Seven Good Reasons for Mobile Agents", *Communications of the ACM*, p.88 - 89, 1999 Mar.

[2] F. Hohl. "A Model of Attacks of Malicious Hosts Against Mobile Agents", *Proceedings of the ECOOP Workshop on Distributed Object Security and 4th Workshop on Mobile Object Systems: Secure Internet Mobile Computations*, p. 105 - 120, INRIA, France, 1998.

[3] "Concordia - Java Mobile Agent Technology". http://www.meitca.com/HSL/Projects/Concordia/

[4] C. Tschudin. "Mobile Agent Security", *Intelligent Information Agents: Agent Based Information Discovery and Management in the Internet*, p. 431 - 446, Springer, 1999.