ISSN:2231-0711
www.ijcset.net

Distributed System in Platform Security Using Mobile Agent

B.Sriprathha, S.Muthukumaraswamy

Dept. of PG Studies in Engineering, S.A.Engineering College,
Anna University, Chennai, Tamil Nadu, India.

sriprathanand@gmail.com, muthusans@gmail.com

Abstract: **An agent from the client going to migrate to all the current servers which are going to be connected directly or indirectly through wired or wireless network. An agent is a part of a program which is developed in the client and it migrates automatically when a request is generated. The agent carries the query to all the servers which are connected and collect the required data from the entire server which is connected and finally it delivers to the client. When the agent is travelling, both the agent and the data which is carrying should be protected, from the hackers or by the system user's hacking knowingly or by unknowingly. Using the Ring signature algorithm path is designed and the path can be extended when there is a need for new path in a distributed system. By this procedure the travelling time and the selection can be restricted to reduce the time delay.**

Keywords: migrate, agent, hacking, distributed systems

1. INTRODUCTION

As the Internet constantly expanding the availability of on-line information and major transaction, it has become compulsory issue to develop the efficiency of the system to retrieve the data to and fro without any internal and external difficulties. The solution to these criteria can be satisfied by 'mobile agent'.

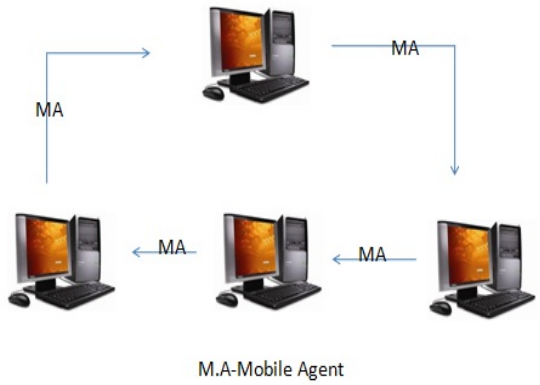
Mobile agents are processes or it can treated as a part of a program (i.e., executing programs) that can migrate from one machine of a system to another machine (usually in the same system) in order to satisfy requests made by their clients. A mobile agent executes on a machine that hopefully provides the resource or service that it needs to perform its job. If the machine does not contain the needed resource/service, or if the mobile agent requires a different resource/service on another machine, the state information of the mobile agent is somehow saved, transfer of the mobile agent to the machine containing the necessary resource/service is initiated, and the mobile agent resumes execution at the new machine.

Advantage:

a) Low network bandwidth,

b) It continue its execution even when disconnected from the network,

c) ability to clone itself to perform parallel execution, easy implementation and deployment, and reliability.



Basic Mobile agent Model

Mobile agents have been developed as an extension to and replacement of the client-server model. In the clientserver model, a server is a machine that provides some service and a client makes requests for those services. Communication between a client and a server is usually through message passing. So, when a client needs a particular service, it usually sends a request message to a server that contains the needed service.

1.1. Mobile Agent System Types

There are six mobile agent systems which includes Agent TCL as first which was later renamed as D'Agents mobile agent system created at to address the weaknesses of existing mobile agent systems, such as insufficient security mechanisms, support for only specific and complex languages, difficult or nonexistent communication between agents, and inadequate migration facilities.

The second system is ARA which is a platform for the portable and secures execution of mobile agents in heterogeneous networks. Mobile agents in this sense are programs with the ability to change their host machine during execution while preserving their internal state.

The next by mobile agent system is the Concordia is provided for the development and management of network-efficient mobile agent applications for accessing information anytime, anywhere, and on both wire-based and wireless device supporting java. The JVM is used for Concordia's runtime environment.

The fourth agent is the Mole which is the first Mobile Agent System that has been developed in the Java language. Mole provides a stable environment for the development and usage of mobile agents in the area of distributed applications. In Mole system, agent model based on Agents and places. Each Agent's identifier is created at the creating of each agent, which uniquely identifies that agent globally.

The fifth agent is the TACOMA is a piece of code that can be installed and executed on a remote computer. Such an *agent* may explicitly migrate to other hosts in the network during execution. The TACOMA project focuses on operating system support for agents and how agents can be used to solve problems traditionally addressed by other distributed computing paradigms, e.g. the client/server model.

The final agent is the Voyager is 100% java agent-enhanced Object Request Broker (ORB). Goals of this product to provide programmer to create state of the art distributed programs quickly and easily while providing a lot of flexibility and extensibility for the products that are being created with the voyager system. This is a 100% pure java based system. Using these virtual objects, several tasks can be performed besides sending messaging the remote objects, which are as follows.

1) Remote object creation.

2) Connection with existing remote object in different applications.

3) Allows to move code and objects to another clients.

2. RING SIGNATURE ALGORITHM

In cryptography, a ring signature is a type of digital signature that can be performed by any member of a group of users that each have keys. Therefore, a message signed with a ring signature is endorsed by someone in a particular group of people. One of the security properties of a ring signature is that it should be difficult to determine which of the group members' keys was used to produce the signature. Ring signatures are similar to group signatures but differ in two key ways: first, there is no way to revoke the anonymity of an individual signature, and second, any group of users can be used as a group without additional setup. Ring signatures were invented by Ron Rivest, Adi Shamir, and Yael Tauman, and introduced at ASIACRYPT in 2001.

The name "ring signature" comes from the ring-like structure of the signature algorithm. In the original paper, Rivest, Shamir, and Tauman described ring signatures as a way to leak a secret. For instance,

a ring signature could be used to provide an anonymous signature from "a high-ranking White House official", without revealing which official signed the message. Ring signatures are right for this application because the anonymity of a ring signature cannot be revoked, and because the group for a ring signature can be improvised.

Another application, also described in the original paper, is for deniable signatures. A ring signature where the group is the sender and the recipient of a message will only seem to be a signature of the sender to the recipient: anyone else will be unsure whether the recipient or the sender was the actual signer. Thus, such a signature is convincing, but cannot be transferred beyond its intended recipient.

Ring Signature Following the formalization about ring signatures pro- posed in, we explain in this section the basic definitions and the properties eligible to ring signature schemes. A regular ring signature scheme consists of the following three-tuple (Key-Gen, Sign and Verify):

- Key-Gen is a probabilistic polynomial algorithm that takes a security parameter(s) and returns the parameters.
- Sign is a probabilistic polynomial algorithm that takes system parameters.
- Verify is a deterministic algorithm that takes as in- put a message M ,a ring signature _, and the public keys of all the members of the corresponding ring, then outputs "True" if the ring signature is valid, or "False" otherwise.

The resulting ring signature scheme must satisfy the following properties:

- **Correctness:** any verifier with overwhelming probability must accept a ring signature generated in a correct way.
- **Anonymity:** any verifier should not have probability greater than 1/l to guess the identity of the real signer who has computed a ring signature on behalf of a ring of l members. If the verifier is a member of the ring distinct from the actual signer, then his probability to guess the identity of the real signer should not have greater than $1/(l-1)$.
- **Unforgeability:** any attacker must not have non- negligible probability of success in forging a valid ring signature for some message M on behalf of a ring that does not contain him, even if he knows valid ring signatures for messages, different from M, that he can adaptively choose.

3. PROPOSED PROTOCOL

Pricing query has been considered as one of the most viable applications of mobile agent technology. The most famous example of the pricing query scenario is the air-fare survey agent in which Alice wants to fly to Paris next Sunday with the lowest price. So she sends a mobile agent to all airline servers in order to check the availability, the price, as well as other useful information

In order to get a fair and meaningful outcome, Alice's agent must keep those offers provided by former airlines secret to the later ones; otherwise later visited airlines may decide their offers according to former ones. Likewise, for guaranteeing integrity of the survey, no collected data will surfer from unauthorized alteration, including truncation, insertion and modification.

Cryptographic technique of confidentiality and integrity can be useful for this purpose of developing secure pricing query protocol.

In the KAG family, the first protocol provides publicly verifiable forward integrity (PVFI) and the second protocol provides Forward privacy (FP) both assuming the existence of public key Infrastructure (PKI).

Publicly verifiable forward Integrity of the collected data prevents the agent originator from receiving useless data since unauthorized modification can be detected during the agent's itinerary. On the other hand, forward privacy protects the anonymity of offer providers against malicious hosts as well as any unauthorized observer.

Unfortunately, these two important features cannot coexist within a single protocol in existing solutions because of the essence of contradiction when traditional cryptographic techniques, precisely the digital signatures, will be employed.

A new protocol integrating both the above two important features will be proposed by employing the ring signature instead of traditional signature schemes. The protocol not only can be applied into e-commerce scenarios nowadays, but works in the generalized theoretical scenario as well.

4. DESIGN PRINCIPLE

By simply replacing the traditional signature scheme in the original PVFl protocol with the above tailored ring signature, one will find out that forward privacy and publicly verifiable forward integrity compromise and coexist magically.

$$S_i \rightarrow S_{i+1} : \Pi, \{Nj \mid 0 \leq j \leq i\}; 0 \leq i \leq n$$

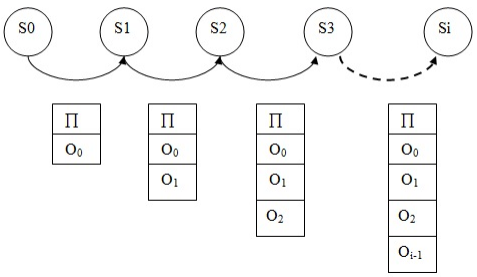
Although the protocol looks the same as the PVFI protocol, the encapsulated offers N_{is} are produced in a different manner. In the proposed protocol, when trying to provide a offer to the agent originator, the shop *Ss* first chooses a random number n_s and encrypts his plaintext offer o_s with the n_s by the originator's public key P_o. *Also,* he is free to choose the ring z members from those candidate shops. Note that the shop S_s, cannot choose the next visiting shop as a member of his ring signature. Otherwise, the next shop will have the ability to modify Ns provided by Ss, without any chance of being detected. Thus leads the protocol failed in achieving strong forward integrity.

As described in Section 3.2, the candidate list is predetermined at departure and, in other words, can be found in the agent code Π. Then, after deciding all ring members, the shop (now is also the offer by computes

RING - SIGN((ENC$_o$(o_s,n_s), h_s ,) ,n_s **, ,** P1,. . . . Pz,. S**,** S_s),

and the output pair

((ENCo (o_s,n_s.), h_s), σ)



Data Collecting During Itinerary is the encapsulated offer provided by shop S_s.

4.1. Problem Statement

Despite its many practical benefits, mobile agent technology results in significant new security threats from malicious agents and hosts. The primary added complication is that, as an agent traverses multiple machines that are trusted to different degrees, its state can change in ways that adversely impact its functionality.

Consider a mobile agent that visits sites run by airlines, hotel chains, and rental car companies searching for a travel plan that meets a customer's requirements. The mobile agent is programmed by the travel agency. A customer dispatches the agent to the Airline-1 server where the agent queries the flight database. With the results stored in its environment, the agent then migrates to the Airline-2 server where again it queries the flight database.

From either of these hosts it may visit a hotel host, where it may be eligible for a special deal depending on whether the customer will be travelling on an allied air carrier.

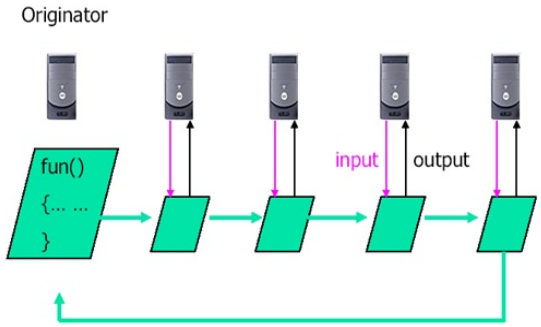
The agent compares flight and fare information decides on a travel plan, migrates to the appropriate airline and hotel hosts, and reserves the desired flights and rooms. Finally the agent returns to the customer with the results. The customer can expect that the individual airlines and hotels will provide true information on flight schedules and fares in an attempt to win her business. However, the airline servers are in a competitive relation with each other.

4.2. Competing Airline Carriers

Consider a mobile agent that visits the Web sites of several airlines searching for a flight plan that meets a customer's requirements. We focus on four hosts: a customer host, a travel agency host and two servers owned by competing airlines, for instance United Airlines and American Airlines, which we assume for the sake of this example do not share a common reservation system. The mobile agent is programmed by a travel agency. A customer dispatches the agent to the United Airlines server where the agent queries the flight database. With the results stored in its environment, the agent then migrates to the American airline server where again it queries the flight database. The agent compares flight and fare information, decides on a flight plan, migrates to the appropriate airline host, and reserves the desired flights. Finally, the agent returns to the customer with the results.

The customer can expect that the individual airlines will provide true information on flight schedules and fares in an attempt to win her business, just as we assume nowadays that the reservation information the airlines provide over the telephone is accurate, although it is not always complete. However, the airline servers are in a competitive relation with each other.

The airline server illustrates a crucial principle: *For many of the most natural and important applications of mobile agents, we cannot expect the participants to trust one another.*



Problem Formulation

There are a number of attacks they may attempt. For instance, the second airline server may be able to corrupt the flight schedule information of the first airline, as stored in the environment of the agent. It could surreptitiously raise its competitor's fares, or it could advance the agent's program counter into the preferred branch of conditional code. Thus the mobile agent cannot decide its flight plan on an airline host since the host has the ability to manipulate the decision. Instead, the agent would have to migrate to a neutral host such as the customer's host or travel agency hosts, make its flight plan decision on that host, and then migrate to the selected airline to complete the transaction. This attack illustrates a principle: *An agent's critical decisions should be made on neutral (trusted) hosts.*

A second kind of attack is also possible. The first airline may hoodwink the second airline, for instance when the second airline has a cheaper fare available. The first airline's server surreptitiously increases the number of reservations to be requested, say from two to 100. The agent will then proceed to reserve 100 seats at the second airline's cheap fare. Later, legitimate customers will have to book their tickets on the first airline, as the second believes that its flight is full. This attack suggests a third principle: *migrating agent can become malicious by virtue of its state being corrupted.*

5. CONCLUSION

Mobile Agents are a burgeoning field of study in the autonomous agent's arena. Their key advantages include their local and real-time interaction and server flexibility. Mobile Agents play an important role in electronic commerce, for both wired and wireless environments. Mobile agents are software codes with the ability of moving from host to host, and the ability of accomplishing tasks by computational powers of visited hosts. This technology brings new inspirations for applications like e-commerce, however, many security issues for investigating.

Collected data protection in the pricing query scenario is one of those emerged topics in agent protection. In this paper, a new protocol with two important security properties of this line, forward privacy and publicly verifiable forward integrity, is presented. These two properties did not coexist in a single protocol in the literatures but now are integrated by using the recent cryptographic technique, ring signature.

In most pricing query applications, itineraries with predetermined candidate hosts are envisaged as the most realistic and reasonable ones nowadays. The proposed protocol works as well in the generalized itineraries, thus is also valuable in theoretic point of view. Besides, when enabling the shops to decide the size of its own ring, shop-configurable security can be achieved and thus leads more flexible implementations.

REFERENCES:

[1] L. Ma, J. J. P. Tsai, Y. Deng, and T. Murata, "Extended elementary object system model for mobile agent security," in *Proc. World Congr. Integr. Des. and Process Technol.*, 2003, pp. 169–178.

[2] L. Ma, J. J. P. Tsai, and T. Murata, "Secure mobile agent system model based on extended elementary object net," in *Proc. 25th IEEE Int. Comput. Softw. Appl. Conf.*, 2004, pp. 218–223.

[3] David M. Chess, Colin G. Harrison, and Aaron Kershenbaum. "Mobile Agents: Are they a good idea?", IBM

[4]M. Köhler and H. Rölke, "Modelling mobility and mobile agents using nets within nets," in *Proc. 2nd Int. Workshop MOCA*, 2002, pp. 141–157.

[5] D. Chess, C. Harrison, A. Kershenbaum, "Mobile agents: Are they a good idea?" in *Proc. 2nd Int. Workshop Mobile Object Syst.*, 1996, pp. 25–47.

[6] W. M. Farmer, J. D. Guttmann, and V. Swarup, "Security for mobile agents: Issues and requirements," in *Proc. 19th Nat. Inf. Syst. Security Conf.*, 1996, vol. 2, pp. 591–597.

[7] G. Vigna, "Protecting mobile agents through tracing," in *Proc. 3rd Workshop Mobile Object Syst.*, 1997, pp. 137–153.

[8] H. K. Tan and L. Moreau, "Trust relationships in a mobile agent system," in *Proc. 5th Int. Conf. Mobile Agents*, 2001, vol. 2240, pp. 15–30.

[9] F. Hohl, "A model of attacks of malicious hosts against mobile agents," in *Proc. ECOOP Workshop Distrib. Object Security, 4th Workshop Mobile Object Syst.: Secure Internet Mobile Comput.*, 1998, pp. 105–120.

[10] T. Sander and C. F. Tschudin, "Towards mobile cryptography," in *Proc. IEEE Symp. Security Privacy*, May 1998, pp. 215–224.