# Matrix Hop Mobile Agent (MHMA) - A Novel Framework for Security Architecture

**K. Dhanalakshmi [a,*], Dr.G.M. Kadhar Nawaz [b,1]**

*Abstract* - **Mobile agent technology is a new computing paradigm appropriate for distributed computing environment. To eliminate various security issues and its corresponding solutions available in existing mobile agent systems like free roaming mobile agent, an alternative model has been developed called as MHMA (Matrix Hop Mobile System) system. In MHMA, the offer (offer means data) collected from the remote host is delivered to the owner before the mobile agent migrates to the next host. Also, this architecture protects the mobile agent platform against spam attacks carried out by the spam agents. This MHMA architecture includes fault tolerant mechanism which will helpful to resume the process even if the malicious host attacks the mobile agent and also used to identify the malicious host.**

***Index Terms*** - data protection, fault tolerant, Matrix hop mobile agent, Spam attacks, security.

## I. INTRODUCTION

A distributed system is a major concern in the today's network world for fast accessing. The development of distributed systems is affected by the need of flexibility, adaptability and autonomy. Mobile agent technology is an active solution to fulfill the needs of flexibility, adaptability and autonomy in distributed systems, to make them smart. For effective mobility, agent platforms use the Agent Transfer Protocol (ATP) to transfer the mobile agent between the hosts effectively. Nevertheless, security is a major issue in mobile agent technology that degrades its performance. As per the Yao (2004) report, security falls mainly into two areas:

a) Protecting a Server from Malicious Agents: The server in a mobile agent environment can be attacked by malicious agents with illegal codes. Attacks may be the denial of service or Unauthorized Access or Masquerade. To protect the servers from the malicious agent, authentication and byte code verification are the major security requirements.
b) Protecting a Mobile agent from Malicious Servers: A mobile agent roaming in the distributed network can be attacked by malicious platforms. A malicious platform may modify or kill the agent. Protecting the mobile agent from the malicious server is a more critical issue than protecting the server because the agent dispatched to remote servers is fully under the control of the remote server. The major security requirements for protecting the mobile agent environment are confidentiality and integrity. To protect the agent and platform from alternating attacks, new security architecture for mobile agent is proposed with few protection mechanisms.

## II. Security Properties

The mobile agent and its environment are protected by having the various security properties against the malicious attackers. The security properties are:
i. Confidentiality: A Mobile agent will roam in the network to collect the information. The information collected from the remote machines should be kept secret from other remote machines during the travel. For example, Subbu wants to fly to Australia and he wants to know the cost details from various travel agencies. An agent is dispatched to collect the information on behalf of Subbu. It will visit multiple agencies and gather the information. The information gathered from one agency should be kept secret from the other agency. Otherwise, an agency will fine-tune its cost details based on the other agency to give preference to choose its flight.
ii. Integrity: Accuracy is the essential thing in all the scenarios. In the mobile agent environment, the agent code and its information should be original (it has to be always the same), not modified by any of the malicious servers.
iii. Non-repudiation: The server which sends the Information to the owner of the agent or to the remote server should not deny that he is not the owner of the particular information and agent.
iv. Authentication: The mobile agent or its information from the remote host must be authenticated before it is allowed to execute or perform execution on that. The server, which receives the agent should validate whether the data or agent is from the valid user or not.
v. Byte Code Verification: The agent server is protected by verifying the code of the migrated agent from the remote host. The code is verified before the execution of the agent starts by the server. The verifier should scan the code of the agent to identify whether it has any illegal operation.

## III. Requirements for Security Properties

Security properties to the mobile agent environment are provided by incorporating various cryptography requirements. The requirements to fulfill the security properties of the mobile agent environment are:
i. Digital Signature: It is used to prove that the message was generated by a particular domain. It is particularly used for the non-repudiation purposes.

**Manuscript received, 19/May/ 2012.**
**Manuscript selected, 19/June/ 2012.**

**K. Dhanalakshmi [a,*],**
Chettinad College of Engineering and Technology, Karur, India
E-mail: kdhanalakshmimca@gmail.com
**Dr. G.M. Kadhar Nawaz [b,1],**
Sona College of Technology, Salem, India
E-mail: nawazse@yahoo.co.in

ii. Hash function: The hash function or message digest is a mathematical transformation that takes the arbitrary length of the message as the input and computes a fixed length of the hexadecimal number. It is a one-way function because it is not able to figure out what input corresponds to a given output.

iii. Key Cryptography: It is used to send information between participants in a way that prevents others from reading or modifying it.

iv. Scanner: It is used to scan the incoming agent for identifying the illegal (malicious codes) activities in the agent byte code to attack the mobile agent platform.

## IV. MATRIXHOP MOBILE AGENT SERVER SECURITY ARCHITECTURE

The mobile agent in a distributed network will roam among remote hosts to collect the information by computing some task. To protect the agent and the platform during migration, each platform should perform some verification and computation based on previous works. The Matrix Hop Mobile Agent (MHMA) server security architecture proposed in this thesis also requires verification and computation as shown in Figure 1. In the design of the mobile agent security architecture, the decision of the next host (Next Host Decider) is added to give smartness to the multi-hop mobile agent with a dynamic order.

Initially, the agent from its originator consists of the byte code, state, itinerary, credential (identification) and dummy information. The host which receives the agent will start the protection models in the architecture to detect spamming agent and protect mobile agent platform against Denial of Service (DoS) attacks. The proposed mobile agent server security architecture has five models, namely: (i) Agent Type Analyzer (ATA), (ii) Agent Tracker (AT), (iii) Next Host Decider (iv) Agent Cloning (AC) (v) Fault Tolerant System (FTS).

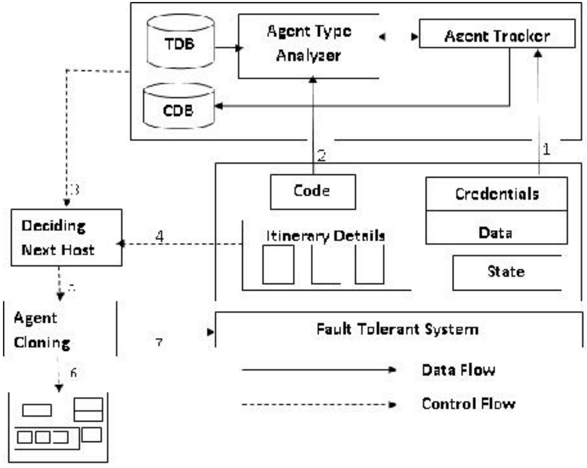

Figure 1. MHMA Server Security Architecture

The process of the security architecture after receiving the agent is: Every platform should initially start the ATA to scan the incoming agent' byte code to identify the type of application. If the ATA verified that the mobile agent is first time enters to the mobile agent platform in the time interval, it will scans the code of the agent to determine the type of application and identifies the threshold value for the mobile agent using TDB (Threshold Data Bases) and updates CDB (Counter Data Base) then, it will send the control to the AT (Agent Tracker). The AT updates the count, starting time, and finishing time in the CDB database. Thereafter, each time an agent visits the platform, the AT allows the agent for normal execution only if the value of the count must be less than its threshold value within the time interval.

Then the Next Host Decider will take control to choose the succeeding remote host from the static itinerary or dynamic itinerary and then the clone is taken by Agent Cloning (AC). The AC retrieves the IP address of the succeeding host and transmission time between the currently residing host and next chosen host. And, AC sends the master mobile agent to the succeeding host and cloning mobile agent to the owner along with the offer produced from the visited host, IP address and transmission time to the owner. The Fault Tolerant System (FTS) ensures the arrival of offer from each visited host by the mobile agent and also ensures the whether the mobile agent is alive or not.

### A. *ATA Analyzer*

The purpose of the Agent Type Analyzer (ATA) is a scanner used to protect the mobile agent platform from a spamming agent (malicious agent). Its process is to scan the byte code of the agent to detect what is the type of application the mobile agent is developed. Today's Java has a code verification mechanism to identify the object initialization, stack overflow and underflow. Also, MIP mechanism scans the byte code of the mobile agent to detect whether the agent consists of any illegal code or not. But, those techniques will not prevent spam requests raised by the spam agent. The ATA with the AT (Agent Tracker) will prevent all types of attacks generated by the spam agents.

### B. *Agent Tracker (AT)*

AT allows an agent to visit and perform its computation in the machine only when the count value is less than its threshold value within a particular interval of time. Every time an agent enters into the platform, this AT will validate the number of times the agent has already visited the machine during the particular time interval. If the current visiting count is valid, AT allows the agent to perform its computations in the machine. After the successful completion of its execution, the agent will dispatch either to its home platform or to its next visiting host depends upon its itinerary information also count is incremented by one and updates CDB.

### C. *Next Host Decider*

The Next Host Decider of the architecture is a non-security model to route the multi-hop mobile agent with a dynamic order. The Next Host Decider has two functions: (i) one is to generate the routing table periodically to improve the journey smartness of the agent and (ii) the other one is to decide the succeeding host for the agent (iii) Attach the IP address of the decided succeeding host and link cost between the currently residing host and selected succeeding host to the data produced.

i. Generating the routing table: Every host in the environment should have the agent to calculate the link cost among the hosts and to generate the routing table and update it periodically. The routing table consists of the remote host IP address and the link cost (time to reach the respective remote host). The link cost is calculated, based on the agent's dispatch time from the home (TD) and the arrival time of agent (TA) at the remote host, as follows:

$$\text{Link Cost} = TA - TD \qquad (1)$$

ii. Deciding Next Host: An agent roaming in the network with a static itinerary dynamic order depends on the remote hosts to decide the succeeding host. The decision of the next host is based on the condition that may be based on the shortest path or the network traffic. This work concentrates on the shortest path because the network traffic is known by every host in the network. The shortest path details of all the hosts are available with the routing table to decide the next nearest host. The routing table available with the host is not constant all the time, for it will be modified periodically considering the network traffic. The link cost has to be calculated periodically to update the table from time to time because of the network traffic. Hence, no other host can identify the decision of the succeeding host.

iii. Next host decider chooses the next succeeding host based on the link cost information available in the routing table. Then, attach the IP address of the decided succeeding host, link cost to the data produced in the currently agent residing host. These details will be used by the Fault Tolerant system (FTS) to calculate the estimated waiting time for the arrival of data from host.

### *D. Agent Cloning*

After choosing succeeding host by the next host decider, now the agent cloning process will be performed by the agent Cloning module. The AC performs the agent clone to create the duplicate agent. Among the two agents, one agent will be send to the owner of the agent with data, IP address, link cost details. And, another agent will be dispatched to the succeeding host to resume the process.

### *E. Fault Tolerant System (FTS)*

Agent failure recovery is one of the important research works to improve the tolerance of the agent failure which will enhance performance of the mobile agent in distributed computing environment. The Fault Tolerant System (FTS) model is proposed for MHMA system to identify the aliveness of the agent and to determine the expected arrival of data from each remote host visited by the agent. As per this model, the owner calculates the estimated waiting time for the arrival of data from each succeeding host by using the IP address and link cost retrieved by the agent from each host. If the agent is alive and host is genuine it is possible to receive the data within the estimated waiting time. If not, the agent is failed and the owner resends the agent directly to the host from which it is expecting data.

## V. Conclusion

In this paper, we have proposed a new security architecture MHMA to prevent the mobile agent environments against various attacks. The agent who migrated from one server to another server will immediately verify and validate its number of visiting time within a time interval. After the validation is successful, the agent is allowed to continue its computation in the environment and generates data. If the count of visiting the mobile agent platform is exceeding its threshold value within a time interval, the agent is not allowed to perform its execution and can be killed, returned to the owner's host or intimated to the agent owner or administrator of the distributed environment to take necessary action against the malicious agents. This mobile agent server security architecture is useful for incorporating the mobile agent in any environment like e-Voting (Robles 1999), ad hoc networks (Levy et al 2005), sensor networks (Wang and Qi 2004), Intrusion Detection System (Hijazi and Nasser 2005), etc.

## References

[1] Aderounumu G.A., Oyatokun B.O. and Adigum M.O. (2006), 'Remote Method Invocation and Mobile Agent: A Comparative Analysis', The Journal of Issues in Informing Science and Information Technology, Vol. 3, pp. 1-11.

[2] Lange D.B. and Aridor Y. (1997), 'Agent Transfer Protocol - ATP/0.1', IBM Tokyo Research Laboratory, Draft No.4.

[3] Xavier L. (2003), 'Java Bytecode Verification: Algorithms and Formalizations', International Journal of Automated Reasoning, pp. 1-40.

[4] Robles S. (1999), 'Applying Mobile Agents Systems into Large Scale Voting System Designing', Master's thesis, Universitat Autonoma de Barcelona, Catalan.

[5] Levy R., Carlos P.S., Teittinen A., Haynes L.S. and Graff C.J. (2005), 'Mobile agents routing – a survivable ad-hoc routing protocol', Proceedings of the IEEE military communications conference (MILCOM '05), IEEE Computer Society, Vol. 5, pp. 2903-2909.

[6] Wang X. and Qi H. (2004), 'Mobile agent based progressive multiple target detection in sensor networks', Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04), IEEE Computer Society, Vol. 2, pp. 285-288.

[7] Hijazi A. and Nasser N. (2005), 'Using mobile agents for intrusion detection in wireless ad hoc networks', Proceedings of the 2nd IFIP International Conference on Wireless and Optical Communications Networks (WOCN '05), IEEE Computer Society, pp. 362-366.

[8] Dhanalakshmi K. and Kadhar Nawaz G.M. (2011) 'Protecting Mobile Agent Platform against Agent Based Attacks (Spam Attacks) using –Anti Spam Techniques (AST)' European Journal of Scientific Research, Vol. 64 No. 2 (2011) , pp. 19-31.

[9] Dhanalakshmi K. and Kadhar Nawaz G M. (2010) 'Performance enhanced mobile agent for e-commerce based applications', IEEE Proceedings of 2nd International Conference on Computing Communication and Networking Technologies (ICCCNT' 2010) , India, pp.1-7.

[10] Dhanalakshmi K. and Kadhar Nawaz G M. (2011) 'Matrix Hop Mobile Agent (MHMA) System for E-service Applications', Elsevier Science Direct Proceedings of International Conference on Communication Technology and System Design (ICCTSD' 2011), India.

## BIOGRAPHY

**K. Dhanalakshmi** is working as a Senior Assistant Professor in MCA Department, Chettinad College of Engineering and Technology, India. She is doing her part time Ph. D at Anna University of Technology Coimbatore, India. She did her B. Sc (Mathematics) in Alamelu Angappan College of Science for Women, India in 1997 and her MCA in Sri Saradha Niketan College of Science for Women, India in 2000. She received her M.Phil from Bharathidasan University in 2007. Her research is on Security for Mobile Agents.

**Dr.G.M. Kadhar Nawaz** is a Professor and Director in the Department of School of Computer Applications at Sona College of Technology, Salem, India. He received his B.Sc in Computer Science and MCA from Vysya College, Salem, India in 1994 and 1997. He received his Ph. D from Periyar University, India in 2007. He has 14 years of experience in teaching and research. He has published more than 60 papers in reputed International Journals and Conferences. His research area is Network Security.