SECURING MOBILE AGENT AND REDUCING OVERHEAD USING DUMMY AND MONITORING MOBILE AGENTS

Rajesh Shrivastava*

Pooja Mehta (Gahoi)**

Abstract:

A mobile agent is a composition of computer software and data which is able to migrate (move) from one computer to another autonomously and continue its execution on the destination computer. The host computer offers the runtime environment for the mobile agents. Mobile agent technology offers several advantages over conventional client-server computing paradigm: reduced network traffic, efficient resource access, dynamic system adaptation, and support for mobile users. Platforms executing mobile agent can be malicious and try to discover the agent intention, to read data transported by agent or modify the agent data, code or state. One obvious security issue is the protection of hosts against possible attacks from suspicious nodes. The cases where the security of data or reliability of agent became must, we need some method to insure the things discussed above. For increasing the security of original mobile agent, we use dummy agent and monitoring agent which increases the overhead. To reduce the overhead, we reduce the size of data of dummy agent.

Keywords: Mobile agent, monitoring agent, distributed systems, security, overhead.

* Head, Deptt. of Computer Science, Shri Ram Institute of Engg. and Tech., Jabalpur, India.

** Deptt. of Computer Science, Shri Ram Institute of Engg. and Tech., Jabalpur, India.

I. INTRODUCTION:

Mobile agents are mobile autonomous processes operate on behalf of users in a distributed computing environment. The autonomous agent concept has been proposed for a variety of applications on large, heterogeneous, distributed systems (e.g., the Internet) [4]. These applications include a specialized search of a middleware services such as an active mail system, large free-text database [5], electronic malls for shopping, and updated networking devices. Mobile agent systems have many advantages over traditional distributed computing environments. They use less network bandwidth, increase asynchrony among clients and servers, dynamically update server interfaces and introduce concurrency [6].

Due to the problems with security of Mobile agents have limited their popularity. Mobile agents are composed of code, data, and state. Agents migrate from one host to another taking the code, data and state with them. The state information allows the agent to continue its execution from the point where it left in the previous host.

II. PREVIOUS WORK:

Mobile agent protection is difficult because of a host's complete control over executing programs. While many approaches have been proposed to defend mobile agents from malicious hosts, none adequately addresses every aspect of security. We survey three proposed approaches for the problem of mobile agent protection. The three approaches are chosen because each approach is very uniquely implemented and has strengths that other approaches do not have; we choose Partial result authentication code approach because it can protect results from mobile agents. Computing with encrypted functions approaches is chosen because it tries to scramble code and data together. An obfuscated code approach is chosen because it scrambles an agent's code in such a way that no one is able to gain a complete understanding of its function.

Yee [9] introduced Partial Result authentication Codes (PRACs). The idea is to protect the authenticity of an intermediate agent state or partial result that results from running on a server. PRACs can be generated using symmetric cryptographic algorithms. The numbers of encryption keys are used by agent. The agent's state or some other result is processed using one of the keys, producing a MAC (message authentication code) on the message when the agent migrates from a

host. The key that has been used is then disposed of before the agent migrates. The PRAC can be verified at a later point to identify certain types of tampering.

A similar functionality can be achieved using asymmetric cryptography by letting the host produce a signature on the information instead.

The new scheme is proposed by Sander and Tschudin [10] where an agent platform can execute a program embodying an enciphered function without being able to recognize the original function. For example, instead of equipping an agent with function f, the agent owner can give the agent a program P(E(f)) which implements E(f), an encrypted version of f. The agent can then execute P(E(f)) on x, yielding an encrypted version of f(x). With this approach an agent's execution would be kept secret from the executing host as would any information carried by the agent. For example the means to produce a digital signature could thereby be given to an agent without revealing the private key. However, a malicious platform could still use the agent to produce a signature on arbitrary data. Sander and Tschudin therefore suggest combining the method with undetachable signatures. Although the idea is straightforward, the trick is to find appropriate encryption schemes that can transform functions as intended.

Hohl proposes what he refers to as Blackbox security to scramble an agent's code [11] in such a way that no one is able to gain a complete understanding of its function.

However, no general algorithm or approach exists for providing Blackbox security. A time-limited variant of Blackbox protection is proposed as a reasonable alternative.

This could be applicable where an agent only needs to be protected for a short period. One serious drawback of this scheme is the difficulty of quantifying the protection time provided by the obfuscation algorithm.

III. PROPOSED ALGORITHM:

The mobile agent has script and data to execute it at the other nodes connected in the network. Because of self mobility of software it travels in the network to solve the purpose but there may be the problem of the security in the network. Hence to protect the agent we propose an algorithm.

There are some important steps of our proposed algorithm-

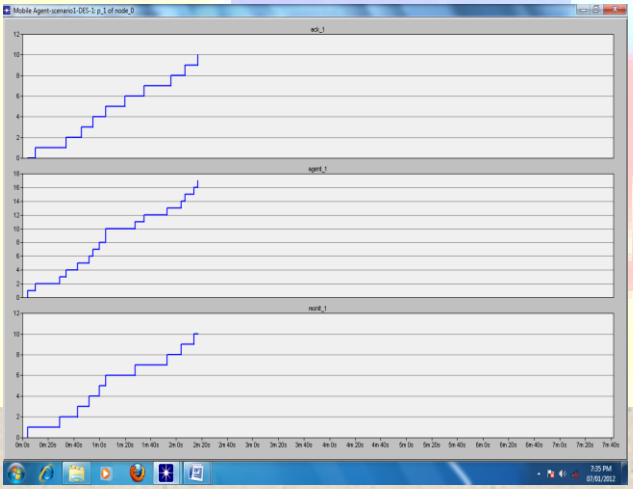
Step 1: The original agent creates a monitoring agent and a dummy agent with same script but with dummy data. With dummy agent, we send the actual script and dummy data but minimize the size of the dummy data to reduce the overhead. Monitoring agent sends the acknowledgement to original agent.

Step 2: Original agent sends the monitoring agent and dummy agent to next node to check the behavior of next node in the network. If monitoring agent finds the node suspicious, it sends the alert acknowledgment to original agent.

Step 3: If there is no harmful activity in next node then monitoring agent sends an ok acknowledgment to original agent to certify the security of original agent.

IV. SIMULATION RESULTS:

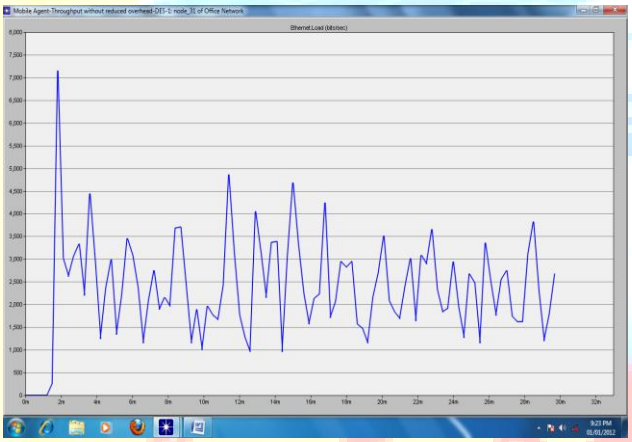
Result shown in Figure 1 shows the protection against suspicious node (system protects the agent to travel through suspicious node). This ensures the security of the original mobile agent from the suspicious nodes in the network.



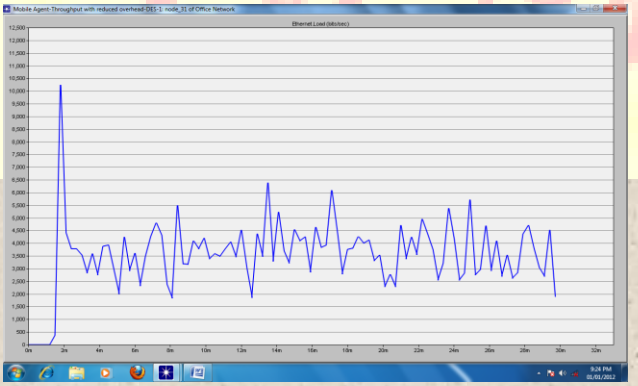
1. Simulation Result for Suspicious Nodes

We send 10 dummy agent and 10 monitoring agent to a node in the network. We get the conclusion that if we send 10 agents and get 10 acknowledgements means there is no suspicious node but if we get only 8 acknowledgements means there is 2 suspicious nodes. So we don't send original agent and data to that suspicious node.

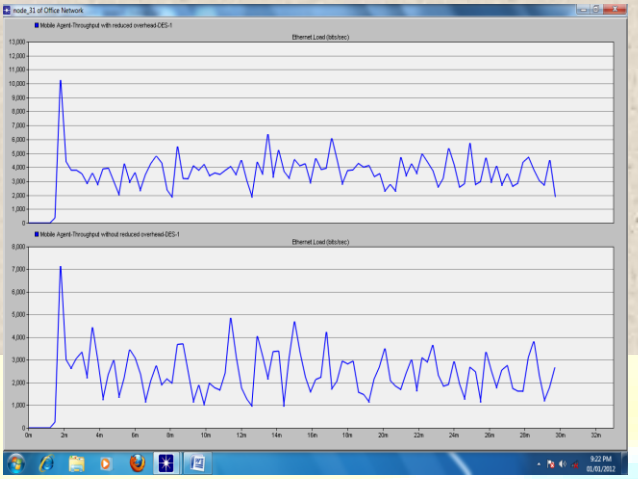
Result in Figure 2-5 shows the increment in the throughput when we minimize the size of data of dummy agent. In figure 2, throughput is shown when we don't reduce the overhead. In figure 3, throughput is shown when we reduce the overhead. In figure 4, throughput is shown with both the situation. In figure 5, a comparative analysis is shown, in which significant increment is seen in throughput. In Figure 5, Blue line is indicating the incremented throughput and red line is indicating the non-incremented throughput. We get the significant increment in throughput after minimizing the overhead.



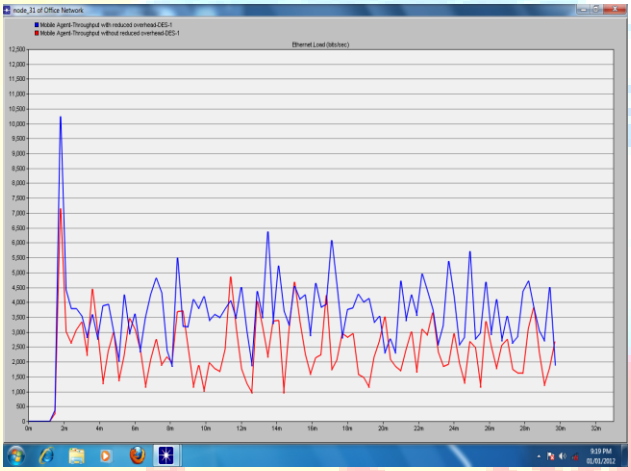
2. Throughput without reduced overhead



3. Throughput with reduced overhead



4. Throughput with and without reduced overhead



5. Comparative result of both the situation

We send dummy agent which holds script and dummy data not the original data, we reduce the size of the dummy data to reduce the overhead which is the burden on the network. We get the maximum throughput 7,000 bits/sec. without reducing the overhead and get maximum throughput 10,250 bits/sec. after reducing the overhead. So we get the 46.42% increment in the throughput by reducing the overhead.

V. CONCLUSIONS:

The opnet simulator is used to study the behavior of mobile agents in insecure environment. We get the success in protection of the mobile agent using dummy and monitoring agents and simultaneously reduced the overhead by minimizing the size of dummy data of dummy agent. We get the increment in the throughput by reducing the overhead.

REFERENCES:

- Ichiro Satoh. *Selection of Mobile Agents*. In Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04). IEEE Computer Society Press, 2004.
- J. White, "*Mobile Agents White Paper*," General Magic Inc., 1996.
- D. Milojici, "*Mobile agent applications*", IEEE concurrency, July-Sep 1999, pp 80- 90.
- Chandra Krintz, *Security in agent-based computing environments using existing tools*. Technical report, University of California, San Diego, 1998.
- Joshua D. Guttman and Vipin Swarup. Authentication for mobile agents. In LNCS, pages114–136. Springer, 1998.
- Neeran Karnik. Security in Mobile Agent Systems. PhDthesis, Department of Computer Science and Engineering. University of Minnesota,1998.
- Tomas Sander and Christian F. Tschudin. Protecting Mobile Agents Against Malicious Hosts.In Giovanni Vigna, editor, Mobile Agent Security, pages 44–60. Springer-Verlag: Heidelberg,Germany, 1998.
- Bennet Yee. Using Secure Coprocessors. PhD thesis, Carnegie Mellon University, 1994.
- Bennet Yee. A sanctuary for mobile agents. In J. Vitek and C. Jensen, editors, Secure Internet Programming, volume 1603 in LNCS, pages 261–274, New York, NY, USA, 1999. Springer-Verlag Inc.
- Tomas Sander and Christian Tschudin. Towards mobile cryptography. In Proceedings of the IEEE Symposium on Security and Privacy, pages 215–224, Oakland, CA, May 1998. IEEE Computer Society Press.

- Tomas Sander and Christian F. Tschudin. Protecting Mobile Agents Against Malicious Hosts.In Giovanni Vigna, editor, Mobile Agent Security, pages 44–60. Springer-Verlag: Heidelberg,Germany, 1998.
- Fritz Hohl. Time limited blackbox security: Protecting mobile agents from malicious hosts. In G. Vigna, editor, Mobile Agents and Security, volume 1419 in LNCS, pages 92–113. Springer-Verlag, Berlin, 1998.
- Neelesh Kumar Panthi, Ilyas Khan, Vijay k. Chaudhari, "Securing Mobile Agent Using Dummy and Monitoring Mobile Agents", IJCSIT Vol. 1 (4) , 2010, 208-211.