An Efficient Approach for Mobile Agent Security

Pankaj Mehta
Mtech Scholar
Computer Science & Engg.
Graphic Era Hill University
Bhimtal

Divya Bisht
Mtech Scholar
Computer Science & Engg.
Graphic Era Hill University
Bhimtal

Nripesh Kumar
Astt. Professor
Computer Science & Engg.
Graphic Era Hill University
Bhimtal

ABSTRACT:

Mobile agent technology provide a computing infrastructure in which a program in the form of a software agent can run at any host, suspend its execution, transfer itself to another host and resume execution at the new host. As the agent migrated between multiple hosts that are trusted to different degrees causes new security threats from malicious agents and hosts.

Mobile applications must balance security requirements with the help of available security mechanism in order to meet application level security goals. In this paper a new security mechanism is proposed to protect the agent from attacks. In this proposed mechanism the mobile code will be encrypted using Triple DES before it starts traversing in the network so that only authenticated hosts can read the current data and state of the mobile agent.

Keywords:

Mobile Agent, security issues, security requirements, protecting agent platform, protecting agent, Triple DES

1. INTRODUCTION:

With the advent of distributed programming [1, 2 and 3] and high speed internet computers have excogitated from huge monolithic devices having very little memory to client server environments that allow complicated and discrete forms of distributed computing. The technology has changed from remote job entry terminals to Java applets, from magnetic tapes to distributed databases, various restricted forms of code and data mobility have always endured. Mobile Agent is a newborn computing archetype which allows complete mobility of cooperating applications to supporting platforms to form a loosely-coupled distributed system.

Mobile agents are independent programs with the capability of changing their execution location when and where it chooses through a series of migrations and itinerary. Mobile agent applications determine and accomplish security requirements based on the unique interactions of agent servers (hosts) with static and dynamic agent components.

Even though having many realistic advantages, mobile agent technology results in significant new security hazards from malicious agents and hosts. The main cause of security threats is that, as an agent traverses multiple machines having different degrees of trust, its state can change in ways that adversely impact its functionality.

Generally, there are two forms of protection in mobile agent security first one is to prevent agent from altering the malicious parties and the other is to prevent parties/hosts from malicious agents. The formulation of trust between agent and host is considerable thought both in distributed application and mobile agent. The most relevant feature mobility complicates trust because the receiving execution host must make distributed trust decisions in spite of having little or no prior knowledge.

2. MOBILE AGENT:

Agents are the independent piece of code or software [5, 8] which acts on behalf of someone. Generally, a small and well defined task is performed by the agent. A mobile agent is a program [1, 3 and 4] that can move from host to host in networked architecture when and where it chooses. It can be migrated anywhere in the itinerary and continue its execution in new host. This contrasts with the client/server model where non-executable messages traverse the network, but the executable code remains permanently on the computer it was installed on.

Compared to traditional client server architecture, mobile agent transmits less data across the network which reduces the network traffic and rescues the network bandwidth. The mobile agent can move, with partial results, from one server to another until it has accomplished its task, and then return to the originating host, which may freely be disconnected during the agent’s trip.

Before the Mobile Agent [16] technology becomes so popular, the communication between the client and server is achieved by different approaches such as message passing, Remote Procedure Call (RPC) and Remote Evaluation (REV). In RPC method, the procedure abides at the server and client sends data to the procedure that will be executed at server, and the result is sent back to the client. The REV approach is different from the RPC, in which the procedure itself will be dispatched and the desired result is returned to the client.

In Client/Server model the server provides services to the client. Typically a client sends a request message to the server whenever it needs a service as shown in figure 1. In case, if the server is unable to satisfy the request made by the client due to the lack of the resources, the client sends request to other server having the required resource to satisfy the client that usually boosted the inefficient use of network bandwidth. As a result of this the network traffic increases and unusual delay occur.

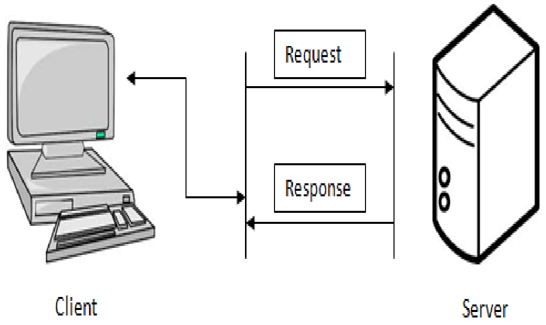

Figure 1: Client/ Server Model [16]

Mobile agent provides solution for this problem as they do not depend on the server operation. The connection between the client and server machine is no longer required as the mobile agent has migrated, later when mobile agent completes its job at the server, then it will reconnect to the client or host with the result shown in figure 2. This certainly saves the network bandwidth especially in the wireless environment where disconnection is frequent and bandwidth play a major role.

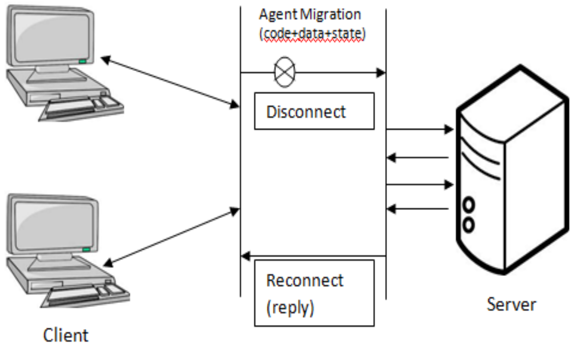

Figure 2: Mobile Agent Model [16]

3. SECURITY ISSUES:

By providing a computing infrastructure [1] mobile agent systems helps execution of agents belonging to different and probably un-trusted users. The communication medium is intrinsically insecure and the different agents and agent systems may have conflicting objectives. In this scheme, a variety of attacks may be possible. Such as network traffic may be eavesdropped by unauthorized users or agents may be observed, or worse an active intruder may modify the code, data or state of an agent. Sometimes agents may attack the agent platform/host to get unauthorized access to the resources.

A new security affair can be introduced in mobile agent system that is protection of mobile agents from malicious sites. Sites could alter with agents' code or state, forcing them to disclose or change private information, run a program with different initial states multiple times and observe the results, "brainwash" them to attack other agents/hosts, delay access to specific and critical resources or deny services to enable other hosts or agents to gain unfair advantage.

Firewall model [6] used for the security of agents in network is not quite successful. There are some assumptions are made in this model that only internal users requests for the execution of programs. Firewall prevents unauthenticated access to or from a private network. Any type of running code can be downloaded with the documents like an applets or a plug-in. This type of agent can run internally in the network and act as a mobile agent. It disrupts the security issues of a network as the agent can generate any security threat inside the network which fails the assumption of firewall that an attacker could only from the outside.

In some situation cryptographic algorithms provide little help. Such as using digital signature it is authenticated that the mobile agent is received from the appropriate source and guarantees that the agent is not tampered in travel. But the cryptography make no guarantees about what the agent might do when executed.

4. SECURITY REQUIREMENTS:

Security is an elementary [3] concern for a mobile agent system. There are a number of desirable security goals for a mobile agent system. Most of these are related to the interaction between the agent and interpreters. The user on behalf of whom an agent works wants it should be protected from malicious interpreters and intermediate hosts which are exist in the transmission route.

Mobile agent [7, 8] has two distinct properties, mobility and agency, mobility concerns with a self-contained and identifiable software component that can move across the network and agency means it act on behalf of users. The use of mobile agent raises a number of security matters. Agents need protection from the other agents and from the host on which they execute. Alike hosts need to be protected from agents and from other malicious parties that can communicate with host. The complications due to the malicious hosts to agents appear more complicated to figure out. As the agent run inside the host so host has entire control over agent, so a malicious host can affect the agent by the following ways

•By observing the code, data and flow control,
•By manipulating the code, data and flow control
•Incorrect execution of code – including re execution,
•Denial of execution – either in part or whole,
•Masquerading as a different host,
•Eavesdropping on agent communications,
•Manipulation of agent communications,
•False system call returns values.

Mobile agent paradigm [9] needs to satisfy the following security requirements.

1) Confidentiality

It is essential to assure that the information carried by a mobile agent or stored on a platform is accessible only to authorized parties. Agent frameworks must be able to ensure that their intra and inter-platform communications remain confidential.

2) Integrity

Integrity means to prevent unauthorized modification. The platform should protect agents from unauthorized modification of code, data and state. Only authorized agents and processes carry out any modification of shared data.

3) Availability

The agent platform must assure that the data and services should be available to local and remote agents. And the platform must be able to provide controlled concurrency, support for simultaneous access, deadlock management, and exclusive access as required.

4) Accountability

It is necessary to keep track of all visiting mobile agent's actions in order to keep them accountable for their actions. Audit logs are used to keep track which are also necessary and valuable when the platform must recover from a security breach, or a software or hardware failure.

5. PROTECTING AGENT PLATFORMS:

Various security checks [10] are required to protect an agent platform from malicious agent both when an agent arrives and while it is executing. Before the agent starts execution, the hosting node should provide protection against malicious agent logic which is defined as a set of instructions that originate a site security policy to be violated. If the agent code is proved as a secure code, the host should authenticate the incoming agent and should arbitrate agent operations on needed resources by means of access control checks.

5.1 Sandboxing and Safe Code Interpretation

The agent [8, 17] moves between different platforms so the agent should be platform independent. Thus the agents are created in such scripting or programming languages which provide this facility. Java is most widely used programming language for the creation of mobile agents. Java also utilizes sandboxing and signed code. In sandboxing the un-trusted programs are executed within their virtual address space so the agents are prevented to interfere with other applications and a unique identifier allows access to the system resources. Sandboxing also provides the memory and method access in a restricted way and provides an execution domain in a mutually exclusive manner.

5.2 Proof Carrying Code

Proof-Carrying Code (PCC) permits a computer system to determine that the program code or agent provided by another system is safe to install and execute. It [11] requires the author of an agent to properly prove that the agent conforms to a certain security policy. The platform where the agent executed checks the agent and the proof before executing it. The agent can then be run without any further restrictions. PCC is useful in many applications. It enhances the ability of a collection of software systems to interact flexibly and efficiently by providing the capability to share executable code safely. Typical examples of code consumers include operating system kernels and World-Wide Web browsers, which must allow un-trusted applications and Internet hosts to install and execute code. The major drawback of this approach is that it is difficult to generate such formal proofs in an automated and efficient way.

5.3 Signed Code

Protecting the agent [12, 17] using the digital signature is the fundamental approach for securing the agent. The digital signature is an electronic signature generated by the creator, user or by the reviewer of the agent. As the agent works on behalf of the some end user, the signature implies the authority and ensures the authenticity, integrity and the origin of the agent. Digital signatures are based on public key cryptography in which two keys are generated one private and one public. To create a signature a hash value is created which is encrypted by the private key at the sender's end and that hash value is decrypted by the public key at the receiver's end. Any change in the data results the different hash value. The decrypted hash value is matched with the computed hash if both are same that means there is no change in agent since it has been signed.

5.4 Path Histories

The agent traverses [12, 17] multiple hopes in its routing path some of them can harm the agent. So it is necessary to keep the track of the safe and unsafe hosts. In this approach the record of the previously visited platforms are maintained so that a newly visited host can decide to allow the execution of the host or not. If the agent is running in an un-trusted host, the newly visited host stops the execution and transfers the agent to a new host. To maintain the path history a signed code is added which contains the identity of current host and the identity of the next platform to be visited. The path history is transferred to the next host and so on. One drawback of this method is that if path history increases the verification become costly.

6. PROTECTING AGENTS:

The main issues to be absolutely addressed to protect agents from malicious hosts are integrity, secrecy and execution. The agent should not be hijacked by un-trusted host or the execution should not occur on false environment. Some more general-purpose techniques for protecting an agent include the following.

6.1 Trusted Hardware

In this approach [14] a trusted third party supply trusted hardware, in the form of tamper resistant device that are placed at the site of the host and interact with the agent platform. This technique is used if the operators of the available execution environments can't be trusted. Smart card is an example of a tamper resistant device. Such trusted hardware either can protect entire agent's execution environment or perform certain security sensitive tasks.

6.2 Trusted Nodes

By offering trusted nodes [8] into the mobile agent framework, sensitive information can be prevented from being sent to un-trusted hosts, and certain misdeed of malicious hosts can be traced. The origin host which launched the mobile agent is assumed to be a trusted node.

In addition to this, service providers can operate trusted nodes in the infrastructure.

6.3 Environmental Key Generation

Environmental key generation [13] grants an agent to carry encrypted code or information. The encrypted data can be decrypted when some predefined environmental condition is true. Using this method an agent's private information can be encrypted and only disclosed to the environment once the predefined condition is met. This requires that the agent has access to some predictable information source. Once the private information has been disclosed, it would, of course, be available also to the executing host. However, if the condition is not met on a particular host, the private information is not disclosed to the platform.

6.4 Execution Tracing

Execution tracing [15] is a technique for distinguishing unauthorized modification of an agent through the reliable recording of the agent's behavior during its execution on each agent platform. In this technique each platform involved to create and retain a non-repudiatable log or trace of the operations performed by the agent while traversing in the network, and to submit a cryptographic hash of the trace. A trace is composed of a sequence of statement identifiers and platform signature information. This technique also provide platform to convey agents and associated security related information among the various parties. If any doubtful results occur, the appropriate traces and trace summaries can be obtained and verified, and a malicious host identified.

7. PROPOSED SYSTEM:

Protecting Agent using Triple DES- A new idea is proposed in this paper to protect the agent using triple DES cryptographic algorithm. As discussed earlier mobile agent is a piece of code or software program which migrates from one host to another host within the network with its data and state. The code can be protected using triple DES by encrypting it in such a form that any other host or agent can't access agent's data and state. Hence any unauthenticated party can't make any changes on its data or state.

In Triple DES three 64 bits keys are used. The code is first encrypted with the first key, then decrypted using the second key, and finally encrypted again with the third key.

Cipher Text= $E_{k3}(D_{k2}(E_{k1}(\text{Plain Text})))$

Decryption is vice-versa. The effective key strength in triple DES is 168 bits which make the mobile code more secure and difficult to trace. But due to the meet in the middle attack, the effective security it provides is only 112 bits. Triple DES is believed to be secure up to at least 2^{112} security which is quite not breakable with today's technology.

Using Cipher Block Chaining mode of operation the first 64 bit key acts as the initialization vector to DES. 64 bit block of mobile code is executed and the resulting cipher text is then XORed with the next plaintext block to be encrypted, and the procedure is repeated.

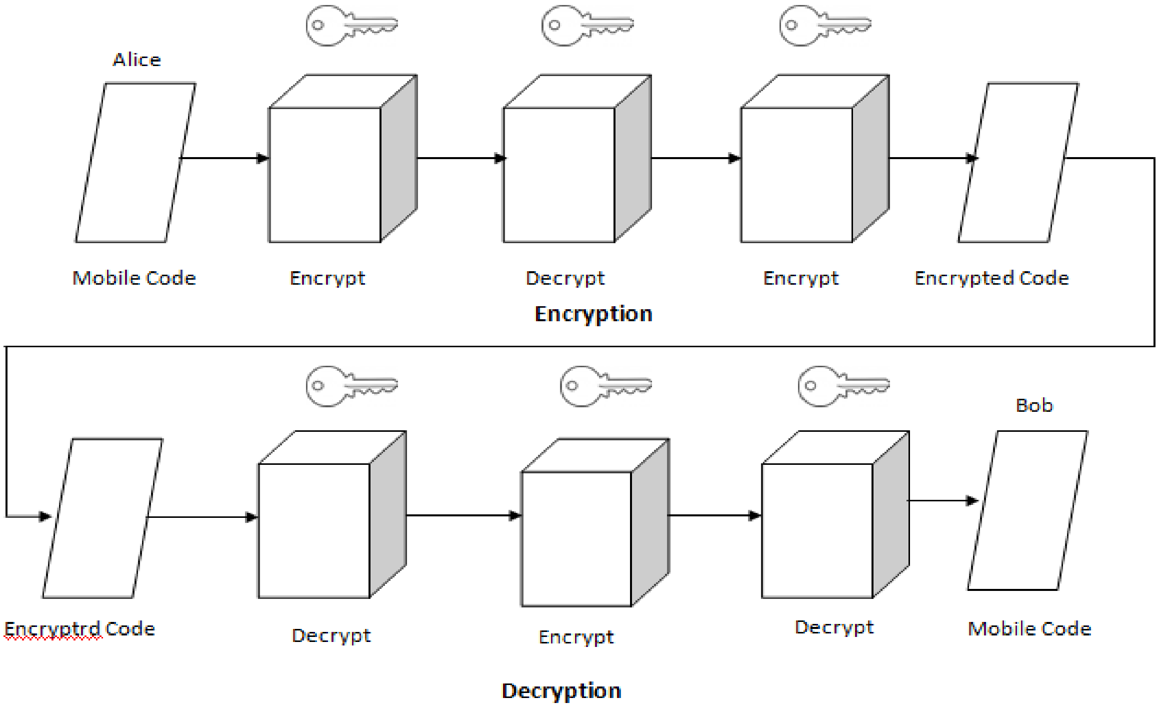

Figure 3: Encryption/ Decryption of Mobile Code using Triple DES

8. CONCLUSION:

The Mobile Agent technology is used in distributed environment so it cannot be expected that the participants will trust each other. So a well suited security mechanism is required to protect both the agent and the host platform. The agent's security problems are very hard. There does not seem to be a single solution to the security problems introduced by mobile agents unless trusted hardware is introduced, which is likely to prove too expensive for most applications. In this paper the newly presented security mechanism will provide protection to the agent. The future scope is to implement this proposed system practically and to test the agent's security by a number of test functions with various level of difficulty. If the agent's code will successfully encrypted using the Triple DES then any intruder cannot make changes to its data and state. This Paper also contains solutions for some of the problems identified in the analysis of the security requirements of Mobile Agent technology.

9. REFERENCES:

[1] E. C. Vijil and Kanwal Rekhi, "Security Issues in Mobile Agents", School of Information Technology, Indian Institute of Technology, Bombay, Mumbai, 400 076.

[2] J. Todd McDonald and Alec Yasinsac, "Security Models for Mobile Agent Systems", www.elsevier.com/locate/entcs

[3] William M. Farmer, Joshua D. Guttman, and Vipin Swarup, "Security for Mobile Agents: Issues and Requirements", The MITRE Corporation 202 Burlington Road Bedford, MA 01730 1420.

[4] David Kotz, Robert Gray, and Daniela Rus, "Future Directions for Mobile-Agent Research", Technical Report TR2002-415, Department of Computer Science, Dartmouth College, January 29, 2002

[5] What is agent? -A world class definition from the webopedia computer dictionary, http://www.webopedia.com/TERM/A/agent.html

[6] Yang Kun, Guo Xin and Liu Dayou, "Security in Mobile Agent System: Problems and Approaches", Department of Computer Science, JiLin University, Changchun, 130023.

[7] V. A. Pham and A. Karmouch, "Mobile software agents: an overview", IEEE Communications Magazine, Vol. 36, No. 7, 1998.

[8] Niklas Borselius, "Mobile agent security", Mobile VCE Research Group Information Security Group, Royal Holloway, University of London Egham, Surrey, TW20 0EX, UK

[9] Rajan Sahota, "An Overview of Security Techniques to Protect Mobile Agent from Malicious Host", International Conference on Computing and Control Engineering, 12 & 13 April, 2012.

[10] Michael S. Greenberg, Jennifer C. Byington, Theophany Holding and David G. Harper, "Mobile Agents and Security", Tufts University.

[11] George C. Necula, and Peter Lee. "Safe, untrusted agents using proof-carrying code". In G. Vigna, editor, Mobile Agents and Security, volume 1419 LNCS, pages 61–91. Springer-Verlag, Berlin, 1998.

[12] Wayne A. Jansen, **"**Countermeasures for Mobile Agent Security", National Institute of Standards and Technology, Gaithersburg, MD 20899, USA.

[13] James Riordan and Bruce Schneier, "Environmental key generation towards clueless agents". G. Vigna, editor, Mobile Agents and Security, volume 1419 in LNCS, pages 15–24. Springer-Verlag, Berlin, 1998.

[14] Uwe G. Wilhelm, Sebastian Staamann, and Levente Buttya, "Introducing Trusted Third Parties to the Mobile Agent Paradigm", J. Vitek and C. Jensen, editors, Secure Internet Programming, volume 1603 in LNCS, pages 471–491, New York, NY, USA, 1999. Springer-Verlag Inc.

[15] G. Vigna, "Protecting Mobile Agents Through Tracing", Proceedings of the 3rd ECOOP Workshop on Mobile Object Systems, Jyvälskylä, Finland, June 1997. http://www.cs.ucsb.edu/~vigna/listpub.html

[16] Yashpal Singh , Kapil Gulati and S Niranjan, "Dimensions and issues of Mobile Agent Technology", International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.5, September 2012.

[17] Wayne Jansen and Tom Karygiannis, "Mobile Agent Security", NIST Special Publication 800-19 –, National Institute of Standards and Technology.