CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The enormous and rapid growth towards the use of Internet in modern days has paved way to improvise its Quality of Service and performance with emerging technologies. In such a view, message passing systems such as Remote Procedure Call (RPC) (Tay and Ananda 1990) and Distributed Object Systems (Minar 1998) were developed to be used in a Distributed Computing Environment. In a Message Passing System, systems which are connected to the terminals communicate with one another using simple messages understandable by the systems. A client invokes the methods in the client system by means of function calls in Remote Procedure Call (RPC). In case of Distributed Object Systems, the client invokes the objects which are residing inside a server, thereby accessing the properties and methods of the objects. Both the mechanisms could use functions and objects which are predefined and therefore lack the property of flexibility and customization. The evolution continued with the origin of Mobile Agent Technology that introduces autonomy, flexibility and customization which makes the Distributed Systems smarter. A detailed study on Mobile Agent Technology (Sau-Koon Ng 2000) clearly describes the advantages of the use of dynamic code deployment and remote data processing in Mobile Agents. This has reduced the overall traffic latency and traffic volume. Though the computational capability of the Mobile Agents is high they are prone to various attacks. The research focuses on the development of protection and
recovery techniques for the Mobile Agents to move securely in a Distributed Environment.

1.2 MOBILE AGENT

Mobile Agent (Hohl 1999) is a software module which works on behalf of its owner with certain capabilities of mobility, autonomy and flexibility. The most distinctive property of a Mobile Agent is mobility. Mobility refers to the movement of the Mobile Agent from one node to another to perform some operation on behalf of its owner. The platform which gives origin to the Mobile Agent is called the Home Platform and is usually termed as the originator. The Mobile Agent has three fields namely the state (execution state of the agent), code (computation to be performed) and data. If a Mobile Agent decides to move from one node to another, it saves its state. Then it dispatches the saved state to the next machine and resumes execution from the saved state. Apart from this, the Mobile Agent is capable of asynchronous execution (Kotzanikolaou et al 2000). The Mobile Agent extends support to both strong and weak mobility. The migration of data and code is termed as weak mobility, and the migration of data, code and state is termed as strong mobility.

1.3 MOBILE AGENT LIFE CYCLE

Figure 1.1 shows the Mobile Agent’s life cycle. In general, a typical Mobile Agent has several operations to perform during its lifespan. The basic operations are given below.

- Create: The agent owner should properly develop an instance and should sign for authorized development.
• Dispatch: The agent can be sent or migrated to any of the remote hosts to perform certain computations.

• Clone: In some cases, many users want to use private instances of the same agent, and it becomes important to make a copy (clone) of an agent. The clone may also be useful in the case of agent recovery.

• Dispose: When the agent completes its execution or has reached the end of its life time limit period, it is to be removed by the host or the agent kills itself.

• Halt: During the execution and service search states, an agent can be halted. “Halt” happens if there are some interruptions; for example, no intended service is found or an agent needs to wait for a certain event to occur.

• Retract: The agent dispatched by the originator to the remote host for computation can be revoked by the originator for some reasons.

• Execute: An agent in the execution state performs certain computations on the originator or remote hosts for which it is created.

Figure 1.1 Life Cycle of a Mobile Agent
In an agent’s life span, it is in a continuous repetitive “execute-halt-migrate-halt” process at different visiting servers and the events that cause a transition from one state to another and the actions that result (Yao 2004).

1.4 TYPES OF MOBILE AGENTS

Various types of Mobile Agents are used to achieve effective computing in a distributed environment. Based on the hops made, Mobile Agents are usually classified into two. A Mobile Agent that returns to its Home platform after visiting a single remote host is called Single-Hop Mobile Agent. Figure 1.2 shows a Single-Hop Mobile Agent.

A Mobile Agent that returns to its home platform after visiting more than one remote host in a single dispatch from the home platform is referred to as the Multi-Hop agent. The Multi-Hop agent can further be divided into three types (Jha and Iyer 2001) based on the itinerary and order as follows:

(i) Static Itinerary Static Order (SISO)
(ii) Static Itinerary Dynamic Order (SIDO) and
(iii) Dynamic Itinerary and Dynamic Order (DIDO).

![Figure 1.2 Single-Hop Mobile Agent Migrations](image-url)
Itinerary is a set of sites that a Mobile Agent needs to visit. This could either be static (fixed at the time of Mobile Agent initialization), or dynamic (determined by the Mobile Agent logic). The order represents the sequence in which the Mobile Agent needs to visit the remote platforms available in the itinerary information. If the order to be visited is predefined by the administrator then it is said to follow static order; else, it is said to follow dynamic order.

In a Static Itinerary, the list of remote host’s to be visited is given by the administrator at the time of dispatching the agent. The Mobile Agent is restricted to visit only the remote hosts listed in the itinerary and must return home. No other users are given privilege to modify the itinerary during the journey of the Mobile Agent.

![Diagram](image)

**Figure 1.3 Multi-Hop Mobile Agent with SISO**

In the static Itinerary, the order of visit may be static or dynamic, and these are referred to as the Static Itinerary Static Order (SISO) and Static Itinerary Dynamic Order (SIDO) respectively. In the SISO, the agent should visit the remote hosts only in the given order. The order may change only...
when the destination remote host fails. Figure 1.3 shows the Multi-Hop Mobile Agent migration based on the SISO, where the table consists of the itinerary (list of remote host address) for the agent to visit in the static order.

In SIDO, the agent should visit the given list of remote hosts only but the visit can be in dynamic order. The order of visiting the remote hosts is based on the current conditions such as shortest path, network traffic etc. of the hosts where the agent is presently residing. The Multi-Hop Mobile Agent with the Dynamic Order is also referred to as the free roaming Mobile Agent with the SIDO. Figure 1.4 shows the Multi-Hop Mobile Agent with the SIDO.

![Figure 1.4 Multi-Hop Mobile Agent with SIDO](image)

In Dynamic Itinerary, the Mobile Agent determines the itinerary dynamically by itself or by the remote host where it is presently residing. The decision of the succeeding host is fully based on the present conditions and requirements. There is no order in visiting the host. It is fully dependent on the requirements, where the required information is available for the Mobile Agent, and conditions such as shortest path or network traffic. It may be noted that a Dynamic Itinerary always infers a dynamic order. This type of Multi-
Hop Mobile Agent is also referred to as the free roaming Mobile Agent with DIDO.

1.4.1 Comparison of the Single and Multi-Hop Mobile Agent (MA)

The Single-Hop Mobile Agent and the Multi-Hop Mobile Agents differ from one another on the basis of visiting the remote hosts. The Single-Hop Mobile Agent will visit the multiple remote hosts by multiple dispatches from the client or multiple agents can visit the multiple remote hosts in a single dispatch. The Multi-Hop Mobile Agent will visit the multiple remote hosts by a single dispatch from the client.

![Comparison of Single-Hop vs Multi-Hop Mobile Agent](image)

**Figure 1.5 Single-Hop Vs Multi-Hop Mobile Agent**

Figure 1.5 shows the comparison of performance between the Single-Hop Mobile Agent and the Multi-Hop Mobile Agent to collect information from the four remote hosts. It shows that the Multi-Hop Mobile Agent takes less time to visit and gather information from the set of remote
hosts than the time taken by the Single-Hop Mobile Agent from the same set of remote hosts. It shows that the Multi-Hop Mobile Agent is better than the Single-Hop Mobile Agent.

Mathematically consider a scenario with \( N \) number of remote hosts that are connected at equal distance from one another. The client, which is linked at the same distance, can dispatch its agent to visit \( N \) hosts using the Single-Hop and Multi-Hop Mobile Agent technique. Then, the visiting time of both Single and Multi-Hop Mobile Agent technique is based on the number of hops given in equations (1.1) and (1.2). It is observed that the Multi-Hop Mobile Agent performs better than the Single-Hop Mobile Agent.

\[
\text{Single-Hop MA} = N^* (2*Mt) \tag{1.1}
\]

\[
\text{Multi-Hop MA} = (N^*Mt) + Mt \tag{1.2}
\]

Here, \( Mt \) denotes the migration time taken to transmit between two hosts. Since the hosts are equidistant from one another, the migration time is considered to be constant.

1.5 ISSUES IN MOBILE AGENT ENVIRONMENT

Mobile Agents are considered as the most powerful forms of code mobility but have not been well acknowledged by the Internet community because of security issues (Vigna 2004). The security attacks in the Mobile Agent environment are categorized as shown in Table 1.1.
Table 1.1 Types of Attacks on Mobile Agent Environment

<table>
<thead>
<tr>
<th>Attacks</th>
<th>Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent to Agent Attack</td>
<td>Masquerading</td>
</tr>
<tr>
<td></td>
<td>Unauthorized Access</td>
</tr>
<tr>
<td></td>
<td>Denial of Service</td>
</tr>
<tr>
<td></td>
<td>Repudiation</td>
</tr>
<tr>
<td></td>
<td>Tailgating</td>
</tr>
<tr>
<td>Agent to Platform Attack</td>
<td>Masquerading</td>
</tr>
<tr>
<td></td>
<td>Denial of Service</td>
</tr>
<tr>
<td></td>
<td>Unauthorized Access</td>
</tr>
<tr>
<td></td>
<td>Tailgating</td>
</tr>
<tr>
<td>Platform to Agent Attack</td>
<td>Masquerading</td>
</tr>
<tr>
<td></td>
<td>Denial of Service</td>
</tr>
<tr>
<td></td>
<td>Eavesdropping</td>
</tr>
<tr>
<td></td>
<td>Alteration</td>
</tr>
<tr>
<td>Other to Agent Attack</td>
<td>Masquerading</td>
</tr>
<tr>
<td></td>
<td>Denial of Service</td>
</tr>
<tr>
<td></td>
<td>Unauthorized Access</td>
</tr>
<tr>
<td></td>
<td>Copy and Replay</td>
</tr>
<tr>
<td></td>
<td>Tailgating</td>
</tr>
<tr>
<td>Platform to Agent to Platform Attack</td>
<td>Masquerading</td>
</tr>
<tr>
<td></td>
<td>Denial of Service</td>
</tr>
<tr>
<td></td>
<td>Eavesdropping</td>
</tr>
<tr>
<td></td>
<td>Alteration</td>
</tr>
<tr>
<td></td>
<td>Unauthorized Access</td>
</tr>
</tbody>
</table>

The research work address the Tailgating attacks in addition to the attacks addressed by Vigna (2004).
1.5.1 Types of Threats

The various threats in the Mobile Agent Environment are caused due to Malicious Agents and Platforms are given below.

i) Masquerade: The Mobile Agent or platform may spoof the identity of the legitimate agent or platform to steal the resources (CPU time) and confidential data (password, credit card number from a bank agent). Masquerade leads to the compromise of security services such as integrity, confidentiality and availability.

ii) Unauthorized Access: The platform and its service can be accessed by the malicious agent without having the proper authorization of the platform.

iii) Denial of Service (DoS): DoS is a serious issue in the world of Distributed Systems. It is a great threat to the availability of resources to an authorized user. The DoS can be launched intentionally by running malicious codes to exploit system vulnerabilities, or unintentionally with the programming errors (Jansen 2000). The DoS attack may result in deadlock where agents on other servers waiting for the outcomes of non-responsive agent on a malicious server, and live-lock where the agent on the malicious server is not able to finish some critical stage of its task and thereby failing in its actual goal (Jansen 2000).
iv) Repudiation: A malicious agent or server, having communication exchange with the legitimate agent or server, denies the communication exchange later on.

v) Eavesdropping: The attacker can gain access to information from the legitimate agent or platform without its knowledge. The information gained from the victims is used for further attacks.

vi) Copy and Replay Attack: The malicious server captures the copy of the legitimate agent or its session key or agent message and is retransmitted to another server for illegal purposes.

vii) Alteration: The malicious platform may alter the agent (state, code, data or itinerary) prompting it to change its behaviour or to cheat the owner of the agent.

In addition to the various attacks discussed above, other issues like loss of an agent and tailgating attacks are also present. The reason for the agent loss is due to the attacker or the Mobile Agent platform failure. The agent roaming in the network may visit different platforms with different characteristics. The malicious remote platform may modify or kill the agent. The agent killed by the malicious platform may have some significant information. The loss of information gathered from the number of remote hosts may lead the agent owner to the crisis. The tailgating attacks focuses on a Mobile Agent, enters the Agent Platform and attack the platform.
1.6 CLASSIFICATION OF MOBILE AGENT SYSTEM ATTACKS

Figure 1.6 shows the classifications of the attacks in different types of Mobile Agent systems. It is observed that an agent attack on the Single-Hop Mobile Agent is very low compared to the Multi-Hop Mobile Agent system, as the Single-Hop Mobile Agent reaches its home after visiting the single remote host.

The loss of agent (data, code and state) in the Multi-Hop Mobile Agent system is a very sensitive issue, as the Mobile Agent has already collected a lot of information from the preceding hosts. The classification consists of two types of platform attacks namely, Direct and Indirect attack.

In the case of a Direct Platform Attack, the malicious platform sends its malicious agent to attack the remote platform. In case of an Indirect Platform Attack, the malicious platform modifies the behaviour of the agent to attack the forthcoming hosts in the agent’s itinerary.
Figure 1.6 Classifications of Mobile Agent System Attacks
1.7 ORGANIZATION OF THE THESIS

The organization of the thesis is as follows:

Chapter 1 presents the introduction about the Mobile Agent, and its types based on its itinerary, and the issues related to the agent and platform attacks and also the motivation of the proposed work.

Chapter 2 discusses the existing works and the issues related to the work presented in this thesis.

Chapter 3 describes the design of the newly framed security architecture to increase the security level of the Mobile Agent environment.

Chapter 4 discusses a new issue in the Mobile Agent System namely Tailgating Attack which focuses on the Mobile Agent Platform. It also provides an effective mechanism to encounter such an attack. It also describes the improved Malicious Identification Police which uses a Tripmarker to identify the External Replay attacks.

Chapter 5 discusses the Address Forward and Data Backward (AFDB) protocol which is used to secure the data carried by the Mobile Agent from Colluded Attacks.

Chapter 6 briefly explains the usage of Trusted Environment with Reference Clone (TERC) which is mainly used for the purpose of recovery. An effective implementation of TERC technique in a DIDS is also discussed. This chapter also presents the eXtended Volunteer Algorithm (XVA) which helps to attain a fault-tolerant environment. The Adaptive Mobile Agents help the System to achieve this task.

Chapters 7 shows the issues which have been addressed by this research and sow the seeds for future research.