CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In the growth of the Internet world, many new network related technologies are incorporated to improve the quality of service. In this vision, the message passing systems, Remote Procedure Call (RPC) (Tay and Ananda 1990) and Distributed Object Systems (Minar 1998) are developed in the line of distributed computing. In message passing systems, the programs available at two ends can communicate with each other by sending simple messages over the network. In Remote Procedure Call (RPC), a program on the client communicates with other program available on the remote server by calling functions. In Distributed Object Systems, instead of calling a function, the remote machine invokes the objects residing on the server, and therefore is possible to access the properties and methods of objects. However, those functions and objects are pre-defined and lack the flexibility for customization. Mobile agent technology, continued in this line of evolution by introducing client customization and autonomy, improves the smartness of the distributed systems. The study (Sau-Koon Ng 2000) reveals that the effectiveness of the dynamic code deployment and remote data processing in the mobile agent, reduces the network’s total latency time and traffic volume. But, the challenge for the well-featured mobile agent technology is the protection against various attacks. This thesis focuses on developing protection and recovery models for secure mobile agent migration in an open system environment.
1.2 MOBILE AGENT

The mobile agent (Hohl 1999) is a software agent acting on behalf of its owner with the extra capabilities of mobility. Mobility refers to the migration of the agent from one node to another node to perform certain computations on behalf of its owner. The platform, from which a mobile agent originates, is called the home platform and the user who creates the mobile agent is referred to as the originator. The mobile agent consists of the state (execution status of the agent), code (computation to be performed) and data. If a mobile agent decides to move from one node to another node, it saves its state and dispatch with the saved state to the next machine and resumes execution from the saved state. In addition to this, the mobile agent has the capability to execute asynchronously and to be fault-tolerant among other ones (Kotzanikolaou et al 2000). The mobile agent supports both strong and weak mobility. The migration of data and code is the weak mobility, and the migration of data, code and state is the strong mobility.

1.3 MOBILE AGENT LIFE-CYCLE

Figure 1.1 shows the mobile agent’s life cycle model. In general, a typical mobile agent has several operations to perform in its lifespan. The basic operations are:

- 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 it has reached the end of its life time limit period, it is to be removed by the host or the agent is killed by 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, for instance, if an agent sends a letter to a customer it will wait for a response.

- **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 AGENT

To perform efficient computation on the distributed environment, different types of mobile agents are used. The mobile agent is mainly classified into two types based on the number of hops it has traversed. The mobile agent that returns to its home after visiting a single remote host as shown in Figure 1.2 is referred to as the single hop mobile agent.

![Single Hop Mobile Agent Migrations](image)

**Figure 1.2 Single Hop Mobile Agent Migrations**

The mobile agent that returns home after visiting more than one remote host in the single dispatch from the home is referred to as the multi-hop agent. The multi-hop agent is further divided into three types (Jha and Iyer 2001) based on its itinerary and order as the (i) Static Itinerary Static Order (SISO), (ii) Static Itinerary Dynamic Order (SIDO) and (iii) Dynamic Itinerary and Dynamic Order (DIDO). Itinerary is a set of sites that an MA needs to visit. This could either be static (fixed at the time of MA initialization), or dynamic (determined by the MA logic). The Order represents the sequence in which the mobile agent has to visit the remote hosts available in the itinerary. If the order to be visited is predefined by the owner then it is called a static order; otherwise, it is a dynamic order.

In a Static Itinerary, the list of the remote host’s address is given by the owner at the time of dispatching the agent. The mobile agent should visit
only the listed remote hosts and return to its home. No privilege is granted to anybody to modify the itinerary during the mobile agent’s journey.

In the static Itinerary, the order may be static or dynamic, and these are referred to as the Static Itinerary Static Order (SISO) and Static Itinerary Dynamic Order (SIDO). In the SISO, the agent should visit only the given remote hosts 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.

![Figure 1.3 Multi Hop Mobile Agent with SISO](image)

In SIDO, the agent should visit only the given list of remote hosts in the dynamic order. The order of visiting the remote hosts is based on the current conditions (shortest path or network traffic) of the hosts where the agent is currently 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.
In Dynamic Itinerary, the mobile agent itinerary can be dynamically determined by the mobile agent itself or by the remote host where it is currently residing. The decision of the succeeding host is fully based on the current conditions and requirements. Here, there is no order in visiting the host; it is fully dependent on the requirements (where the required information for the mobile agent is available) and conditions (shortest path or network traffic). It may be noted that a dynamic itinerary always implies 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.

Figure 1.5 shows the performance comparison of both 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.

![Figure 1.5 Single Hop Vs Multi Hop Mobile Agent](image)

In the form of mathematics, consider a scenario with $N$ number of remote hosts that are connected with an equal distance between them. The client, which is also connected at the same distance, can launch 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 shows that the multi-hop mobile agent is better than the single-hop mobile agent.
Single Hop MA= \( N^* (2^* Mt) \)  

Multi Hop MA= \((N^* Mt) + Mt\)  

1.5 ISSUES IN THE MOBILE AGENT ENVIRONMENT

Mobile agents are considered as the most powerful forms of code mobility but have not been well received by the internet community because of security issues (Vigna 2004). The security issues (attacks) in the mobile agent environment are categorized into five types as shown in Table 1.1.

<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>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>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>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>● Unauthorized Access</td>
</tr>
<tr>
<td></td>
<td>● Alteration</td>
</tr>
</tbody>
</table>
1.5.1 Types of Threats

The various threats in the mobile agent environment created by the malicious agent and platforms are:

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 integrity, confidentiality and availability.

ii) Unauthorized Access: The malicious agent can access the platform and its services without having the proper authorization of the platform.

iii) Denial of Service (DoS): DoS is a critical issue in all over the network world. It reduces the availability of resources to the legitimate users. The DoS can be launched intentionally by running attack codes to exploit system vulnerabilities, or unintentionally through programming errors (Jansen 2000). The DoS attack can cause “deadlock” – where agents on other servers waiting for the results of non-responsive agent on a malicious server, and “livelock” where the agent on the malicious server is not able to finish some critical stage of its task; therefore it can never catch up or achieve its goal (Jansen 2000).

iv) Repudiation: A malicious agent or server, having conducted a communication exchange with the legitimate agent or server, later denies the communication exchange. For example, the bidding agent can deny having provided information like the bidders preferred price.
v) Eavesdropping: The attacker can gain information from the legitimate agent or platform without its knowledge. The information gained from the victims is used for further attacks. It is a passive attack, not an active attack. For example, the attacker can gain the credit card information from the victim and use the details in the future.

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

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

In addition to the multiple types of attacks, the loss (failure) of an agent is another issue. The reason for the agent loss may be the attacker or the mobile agent platform failure. The agent roaming in the network may visit a different platform with different characteristics. The malicious remote platform may modify or kill the agent. The agent killed by the malicious platform may have a lot of valuable information. The loss of information gathered from the number of remote hosts may lead the agent owner to the crisis.

1.6 CLASSIFICATIONS OF ATTACKS ON DIFFERENT MOBILE AGENT SYSTEMS

Figure 1.6 shows the classifications of the attacks in different types of mobile agent systems. The probability of 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.
Figure 1.6 Classifications of the Attacks in Different Types of Mobile Agent Systems
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, viz. Direct and Indirect attack.

In the case of a Direct Platform Attack, the malicious platform will send its malicious agent to attack the remote platform. In case of an Indirect Platform Attack, the malicious platform will modify the behavior of the agent to attack the forthcoming hosts in the agent’s itinerary.

1.7 MOTIVATIONS

In recent years, a mobile agent is used in many applications like Data mining (Klusch et al 2003), Grid computing (Kuang et al 2002), P2P networks (Lu and Fu 2006), Network routing (Manvi and Venkatram 2007), etc. The key reasons for incorporating the mobile agent concept in various applications are:

- Reduction of the network load: Mobile agents will reduce the data flow in the network by packing the conversation and dispatching it to a destination host for the agent to compute. The main advantage of the mobile agent is that the computation moved to the place where the data is available.

- Reduce network latency: Critical real-time systems, such as robots in manufacturing processes, need to respond to real-time changes in their environments. Controlling such systems through a factory network of a substantial size, involves significant latencies. For critical real-time systems, such latencies are not acceptable. Mobile agents offer a solution,
because they can be dispatched from a central controller to act locally and execute the controller’s directions directly.

- Dynamic Adaption: Mobile agents can sense their execution environment and react autonomously to changes.

- Robust and fault-tolerant: If a host is being shut down, all agents executing on that machine are warned and given time to dispatch and continue their operation on another host in the network.

- Client Customization: In distributed computing models like Remote Procedure Call (RPC) and distributed objects (RMI), the exposed functions are defined and established on the server and there is no opportunity for client customization. Clients are confined to the service provided by the server. In case the clients want to have a new service, the service must be installed on the server. But with the concept of mobile agent, the clients are virtually installing programs on to the server when the mobile agent migrates from one host to the others.

Despite its many practical benefits, users of the mobile agent technology suffer from various issues, mainly, security threats. To protect against these types of attacks, various solutions were developed, but the developed security models for a mobile agent environment do not give a guarantee to protect it from new types of attacks, and also no recovery model is reported for the multi-hop mobile agent. The work reported in this thesis has been motivated by this fact to develop advanced protection and recovery models for the mobile agent environment.
1.8 OBJECTIVES

The objective is to design an advanced mobile agent security architecture to protect the mobile agent environment from various attacks with a focus on

(i) To protect the mobile agent code from malicious host attacks or malicious agent attacks.

(ii) To protect the mobile agent platform from the malicious agent of a malicious host.

(iii) To protect the entire data carried by the mobile agent throughout its journey.

(iv) To develop mechanisms to recover the mobile agent (data, code, itinerary and state) after the failure or loss of an agent.

1.9 CONTRIBUTIONS

Despite most of the benefits of the mobile agent paradigm, the security issue is a paramount problem in its usage. Though many solutions are proposed, each of them has its own drawbacks in terms of a different environment. In this context, a set of new security models are proposed to protect the mobile agent environment from various attacks from multiple malicious hosts.

Model-I: The Malicious Identification Police (MIP) model has been formulated towards the mobile agent platform protection. The Attack Identification Scanner (AIS) of the MIP in every mobile agent platform will scan the incoming agent to detect the malicious activity of the agent. A policy framework is developed to scan the agent in terms of its originator privileges.
Model-II: The Root Canal (RC) and eXtended Root Canal (XRC) algorithms are proposed to maintain the integrity of the mobile agent code. The RC integrity model protects the agent code against accidental and intentional alteration by malicious hosts. The XRC integrity model offers security assurance over false malicious claims from the malicious succeeding hosts.

Model-III: The 3-Identification (3-ID) algorithm is developed to assess the modification of the original information or the insertion of fake information on the agent roaming in the network.

Model-IV: The $K$-response recovery model is offered to provide the confidence to the originator that the original form of the agent is returned. This model is also equipped with the capability to recover the original agent in case of any negative impact (altered or killed agent).

The developed protection and recovery models are assessed with respect to their capabilities in e-Health environments. Since the models stated here are fine grained and quite narrow in nature, this study can definitely answer a majority of the related questions for all environments. In seeking to argue for the research goals, it is clear that this work differs in flavor from the majority of the existing research works as it dealt with experimental results and quantitative outputs.

1.10 ASSUMPTIONS OF THE THESIS

This thesis focused to protect the mobile agent environment from different types of attacks using the different protections models. The protection models require some assumptions, are as follows:
i) The public key of the hosts are already distributed to the remote hosts connected in the network.

ii) The proposed models to protect the environment against the attack are the same in all hosts and also it is known by all the hosts of the network.

iii) All illegal activities are reported or intimated to the administrator to take further actions. The actions taken by the administrator is blocking the agent owner and host which perform the indirect platform attack from the network permanently or temporarily until the clearance of the complaint.

1.11 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 presents the proposed MIP scanning model to protect the mobile agent platform against malicious agents by scanning the entire incoming agent. Additionally, the policy-based scanning mechanism is described to improve the efficiency of the method.
Chapter 5 describes the proposed RC and XRC based code integrity checking model to protect the mobile agent from a malicious host, and also to protect the genuine host from any false malicious claim from a malicious host.

Chapter 6 describes the proposed 3-ID algorithm to protect the information carried by the multi-hop mobile agent against attacks from single or multiple malicious hosts.

Chapter 7 briefly presents the $K$-response recovery model to recover the mobile agent after it is destroyed or altered by the malicious attackers.

Chapter 8 illustrates the multi-hop mobile agent application in the e-health environment with experimental results.

Chapter 9 concludes by showing how the research goals have been met and providing a future research road-map.