PROTECT MOBILE AGENT AGAINST MALICIOUS HOST USING PARTIAL-MOBILITY MECHANISM

Tarig Mohamed Ahmed

Faculty of Mathematical Sciences, University of Khartoum, Sudan

ABSTRACT

A mobile agent is a promising area in distributed systems. It is a new technology for computers to communicate. Despite the multiple benefits of the mobile agent, but there are several obstacles to its spread. The mobile agent protection is one of these obstacles. In this paper a new mechanism has been proposed to protect mobile. The mechanism is called Partial-Mobility Mechanism (PMM). The main idea behind this mechanism is to allow mobile agents to visit malicious hosts partially by using a One-Hop-Agent (OHA). OHA is a type of the mobile agent that contains only a task that will be executed in a malicious host. By avoiding the mobile agent to visit the malicious host, PMM completely protects the mobile agent’s secrecy and integrity. PMM has been implemented using .Net framework and C# technologies. Some experiments have been conducted to test the feasibility and performance of the mechanism. Full analysis of the results have been presented and discussed.

KEY WORD

Mobile Agent, Mobility, Security, Privacy

1. INTRODUCTION

Mobile agents (MAs) are independent objects capable to achieve tasks in heterogeneous networks on behalf of users. Based on user’s requests, the MAs start their journey and move autonomously among hosts. The users need a very short time of connection to network in order to dispatch MAs and after that they can go offline. This a big win in terms of reducing the communication cost. The MAs is working based on a concept of remote programming and asynchronous communication mode. So, the problem of the network interruptions and the network latencies are avoided. Based on these features, MAs could be used in many applications of distributed systems, for example, networks maintenance, applications deployments, information retrieval ...etc.

The MA’s body consists of three parts: first part is a code which represents the behavior or tasks of the MA that will be executed in the hosts. The second part represents the MA’s data space which is updated according the execution of the first part. The third part is the execution state that keeps a execution start point in each host. The key feature of MAs system is a mobility that allows MAs to move among network nodes. There are two types of the mobility: the first is called strong mobility which allows MAs to move with code the three parts. When the MA arrives to next station during the journey, it will start the execution using its execution state. The second type is a weak mobility which allows to MA to move with the code and the data only. In this type,
the tasks will be executed similarly in all nodes but the data may change. Communication methods in MA systems are available between MAs themselves, MAs and their users or between the MA and the hosts. The communication provides some kind of information exchange between different entities. As general the message passing concepts are used in the MA systems.

A security is one of the major important issues that should be carefully planned when developing a MA model. The model must protect all parties in the system such as: MAs, Hosts, the mobility and communications. This area has taken a wide range of researchers’ attention. And it could be classified as one of the biggest challenges. The importance of the security comes from the nature of the MAs itself; that contains distributed entities any gape of security will affect overall the system. The MA needs a protection against other malicious MAs and hosts. The host needs a protection against malicious MAs. Also, the channels of the mobility and the communication should be secured. Gary [4] has defined these security areas in details. This paper deals with MAs protection against malicious hosts. The MA may face many risks form hosts such as stealing sensitive information like credit cards’ numbers, attacking the integrity of the MAs, preventing the MA to continue its journey...etc. All these possible attack have been resolved by PMM. This mechanism is completely protected the MA against malicious hosts. A full description of the mechanism will be presented later. As road map of the paper: section 2 presents some recent researches related to the MA protection. Section 3 provides full description of PMM. Section 4 presents some experiments related to PMM’s feasibility and performance with results discussion. Finally, section 5 concludes this paper and giving some recommendations as future works.

2. RELATED WORK

Many risks surround The MAs system. It is important to find a way that could mitigate these risks. The following points represent some of them [5]:

- Eavesdropping: Try to attack the secret or sensitive information in the MA. For example, information that is collected from others hosts or Credit Card Number.
- Intercepting and altering: this type of attack affects the behavior of the MA by making some modification of the MAs’ tasks, the itinerary table or results that collected hosts for example.
- Reply: illegally sending a MA’s copy to perform a malicious action.
- Masquerade: An entity of MA’s system pretends to be a different entity.
- Capturing: Hosts or service provides may capture the MAs and prevent them to continue their journeys.

This area of security has taken grade attention from the researches and some valuable mechanisms have been proposed to protect the MAs against malicious hosts. [6, 7]:

Self-Modifying Code Mechanism: This mechanism proposes an obfuscation algorithm based on self-modifying code to prevent attacks against a MA’s code at function level. The mechanism’s algorithm has been implements and it proves the efficacy [20].

Host Revocation Authority: The main idea of this mechanism is to use a Trusted Third Party, the Host Revocation Authority. The HoRA plays an important role by controlling malicious actions that are done by the hosts in the past. The MA’s sender must consult the HoRA before sending
the MAs to the hosts with bad history [8]. This mechanism fails to protect the MA against malicious hosts in case no history related to that hosts.

A Secure Mobile Agents Platform: By using access control and authentication, this mechanism protects the MAs. The host controls all the resources available on it. Each MA defines its own control policy for other MAs by using an Interface Definition Language (IDL) [9].

Execution Tracing: Gary [10] proposes to detect the malicious actions from the MAs after returning home. Vigna [11] proposes a mechanism to detect the attack by using cryptographic traces and looking to the MAs history file (log) where the MA’s user will know if the MA has achieved its duties correctly or not [12]. In this mechanism the MA must maintain large log information and this is a drawback. Also, a secure protocol is required for transferring cryptography hashes for external entities.

Obfuscated Code: The main idea behind this mechanism is to create a Black Box out of an original MA to execute the same task of the MA as an original MA, but by different arrangement [13]. Obfuscated Code has disadvantages, for example, no black-box algorithms exist that work for arbitrary data. Sander and Tschudin [14] used cryptography in their approach in special cases by having the MA’s program computes not the original, but an encrypted version of it. The result of this function is decrypted by the MA’s user. But, cryptography theory has not a schema that computes arbitrary function in a non-interactive manner.

The Ajanta mechanism: This mechanism proposes three approaches for protecting the MA [15]. The first is to allow the programmer to define parts of the MA’s state as Read-Only and if any modification occurs to these parts, the MA’s user can detect using the digital signature mechanism. The second approach is let the MA creates append-only data states container where the data stored in this container can not be deleted or altered without detection by MA’s user. The third approach is to let programmers to define data states to specific hosts and no other hosts can deal with these data states. These mechanisms use the encryption, the decryption and the digital signature.

Partial Result Encapsulation: This mechanism detects attacks by using encapsulating the results of MA actions at each host, for subsequent verification or when it returns to the home [16]. The disadvantage of this mechanism, for example, does nothing to ensure MA’s privacy. Also, the results to encapsulate may not be immediately clear.

Environment Key Generation mechanism: When some environment condition is occurred, this mechanism allows the MA to take an action. A key is generated is used to unlock some executable code that was encrypted [17]. This approach has weakness such as: the control of the MA could simply modify. The host limits the capability to execute a MA’s code that not related dynamically, sense it is considered an unsafe operation. The mechanism is connected with other protection mechanism.

KeyLets mechanism: This mechanism based on partitioning a MA as units according to the task type [18]. By using secret keys, it encrypts each unit to protect them. The distribution of keys to different hosts is done through the execution of specific type of a MA that is termed a Keylet. The disadvantages of this approach: Propagation requires a third party code producer that can supply the MA by a template the MA’s owner. Also, a large number of transactions related to the keylet
and a host may not be willing to support the increased of computation. Moreover, key revocation is not good in quality. In addition, it requires a complicated mechanism to categorize tasks of the MA. Also, this mechanism does not protect the MA code completely.

3. PARTIAL-MOBILITY MECHANISM

Partial-Mobility Mechanism (PMM) is a new mechanism to protect MAs integrity and privacy against malicious hosts. In PMM the MA has two types: the first one is an One_Hop_Agent (OHA) which can visit only one host. The second is a Multi-Hop-Agent (MHA) which can visit multiple hosts. The MHA can contain multiple of OHAs. The main idea behind PMM is to allow to the One-Hop-Agent to visit untrusted hosts only. So, the MA will not visit any host that is classified as untrusted host. In PMM, all hosts will be visit by MAs are classified in two categories, trusted and untrusted hosts. Now, a full detail of PMM is described as follows:

3.1 MA in PMM

In PMM the MA has two types: OHA which represents the task that will be executed in an untrusted host. OHA is valid to work in only one host. OHA consists of three main parts: part one, Data_Input which represents the inputs data that for the task that will be implemented in the untrusted host. Part two, Task which represents the task required from the untrusted host. Part three, Output_Info that represents the results after executing the task in the untrusted host. There is also other information related to system like mobility’s information. Figure 1 presents OHA.

The second type is MHA which can visit multiple hosts. The MHA consists of different items such as: an itinerary table, Tasks for each host, Data State (in PMM, strong mobility is used), Data and OHA’s table. The OHA’s table contains one task per each untrusted host. Figure 2 presents MHA.
Also, MHA has additional information related to communication, security, execution state…etc. Based on user’s request, a MA is created. Suppose the MA will visit N hosts. If M (0<M <N) hosts are classified as untrusted hosts. In this case, the MA will contain N-M tasks which represent the tasks in trusted hosts in the MHA. For untrusted hosts, M tasks for OHAs will be created and embedded in the MHA.

3.2 Mobility in PMM

In PMM the strong mobility is used. As mentioned above, in PMM there are two types of MAs, MHA which represents the tasks that will be executed in trusted hosts and OHA which represents a task that will be executed in an untrusted host. To represent one MA, PMM needs only one MHA and at least one OHA which embedded in MHA.

The mobility in PMM has two ways. The first one is related MHA which the MA movies normally among hosts using the itinerary table. But, if the next station of the MA is an untrusted host, the second way of the mobility will be used. It is related to OHA. The MHA will not visit the untrusted host, instead of that; the specified OHA will visit the untrusted host. The security of PMM comes from this point that the MHA which contain the all tasks and others OHAs are not allowed to visit any host that classified as untrusted host. By this way, the MAs are completely protected against malicious hosts. Figure 3 presents mobility mechanism in PMM.
As seen in figure 3, the hosts are classified as trusted and untrusted hosts. If the trusted host is a next station in the itinerary table, the MAH will move to that host. On the other hand, if the next host is untrusted the OHA will move to that host. By this approach, untrusted host will never deal with content of the MA.

3.2 PMM implementation

Base on above PMM concepts, a full MA’s system has been developed using C# language and Dot Net Framework as application platform. The system consists from different entities such as following:

a. MA’s Home

MA’s home plays important role in the system. It creates the MA’s according to users’ requests. This entity identifies all hosts that will be visit by the MA. There is a C#’s class to represent the MAs. By using the class, the MA object is created.

b. MA’s Server

This entity aims to receive the MAs after finishing their journeys. It connects to MS-Access 2007 as a database to store all information that has been collected by the MAs. The server can receive many MAs simultaneously. After MAs arriving to this place, the server extracts and stores the information in the database.

c. Hosts

The hosts are classified into two classes: trusted and untrusted hosts. These hosts as general provide the MAs by services. The main different between two classes is in dealing with MAs. The trusted can receive and serve only MHAs and the other can receive and serve only OHAs. Also, the trusted hosts can generate OHAs based on the task required to implemented in case the next station is an untrusted host.

Figure 4 presents the architecture of the PM system.
4. EXPERIMENTS

To test the feasibility and the performance of PM many experiments have been done using the PMM’s implementation as following:

a) At first, a MA has been generated to visits 50 hosts all hosts are assigned to be trusted hosts. The hosts have been selected randomly to create the itinerary table. This process is
repeated 10 times and in each turn, the itinerary table is created randomly. After MA completes the journey, the time is computed in msc. Figure 5, 6, 7 and 8 present some snapshots of these tasks.

Figure 5 MA Test visit trusted host no 1

Figure 6 MA Test visit untrusted host no 6
This process is repeated but one of the hosts is assigned to an untrusted host to see how PMM deals with the untrusted host. Also, the time is computed in each turn. Figure 9 shows the result of this experiment.
In figure 10 test No1 represents the cost of time in msc of the MA journey that has visited 50 trusted hosts. Test No2 represents the cost of time of the MA that has visited 50 hosts one of them is an untrusted host. In test No2 PMM Mechanism has been used and we see, PMM has a little effect in performance.

In next experiment, a MA has created to visit 50 hosts randomly 11 times. In each turn, the number of untrusted hosts is increased by one. The experiment starts with 0 untrusted hosts to 10 untrusted hosts. A figure 6 presents the result.
By using Regression linear model, the following equation has been developed to represents the performance:

\[ y(x) = \alpha + \beta x \]

X represents the number of untrusted hosts. Based on the experiment, the model of the performance is:

\[ y(x) = 491.2 + 152.1x \]

The error has been computed and it is equal to 0.016 msc

5. CONCLUSION

In this paper a new mechanism has been proposed. The mechanism is called Partial-Mobility Mechanism (PMM). The main idea of PMM is to protect the MAs against malicious hosts. PMM has two types of MAs, the MHA which can visit only trust hosts and the OHA which can visit only untrusted hosts. The trusted hosts help PMM to generate OHAs. PMM components have been explained and implemented. Some experiments have been done to test the feasibility and the performance in terms of time cost.

REFERENCES


