Trust and Reputation-Based Model to Prevent Denial-of-Service Attacks in Mobile Agent System

Praveen Mittal and Manas Kumar Mishra

1 Introduction

Distributed system provides resource availability at various geographical locations. Mobile agent system uses this concept of distributed system as a key point and executes its line of code at various locations of resources. An agent is a program that assists people and acts on their behalf. Agents function by allowing people to delegate work to them [1]. This program migrates from one platform to another for the partial or full execution of its line of code. Not only that, but it can migrate from one platform to another for accessing various resources. A platform is nothing but a computer system, which can create number of mobile agents for various tasks.

In mobile agent system, deploying an agent on various platforms involves the possibility of denial-of-service attacks on agent. Hence, the mobile agent environment should be secure and reliable for agent to execute. The proposed model maintains reputation of the platform to which the agent will get executed and trust of the mobile agent. In the following section, we analyze some of the mobile agent's development kits available in market.

1.1 A Mobile Agent Kit

Mobile agent is a program that can be developed with the help of mobile agent development kits available in market such as Concordia, Jacada, Aglets, Voyager,

P. Mittal (✉) · M. K. Mishra
GLA University, Mathura, India
e-mail: praveen.mittal@gla.ac.in

M. K. Mishra
e-mail: manas.mishra@gla.ac.in

S. Chakraverty et al. (eds.), *Towards Extensible and Adaptable Methods in Computing*, https://doi.org/10.1007/978-981-13-2348-5_22

Mole, and Odyssey [1]. Following section describes more about these mobile agent development kits.

Concordia is a full-featured framework made up of Java virtual machine for the development and management of network-efficient mobile agent applications, which extend to any device supporting Java. Concordia consists of various components, which are written in Java and combine together to provide a complete, robust environment for applications.

Jacada is another mobile agent software development kit which is freely available for download with a license key on a trial basis. It is designed only for Windows, size of a file is 160 MB, and system requirements are 5 GB hard disk space, 1 GB RAM, operating system must be higher than Windows XP SP2 [1].

Aglet is a very popular and famous mobile software development kits and freely available over the Internet. Tryllian is a company which developed various mobile agent development kits for different platform such as PDA and smart phones. A management server monitors the deployed agents and able to act accordingly. Tryllian offers services of license and subscription. Deploying an agent on the Tryllian network and installing agent development kit on server are the opportunities comes with subscription and license, respectively.

Voyager from object space is a Java-based platform. It facilitates objects to migrate as agents in the network. Voyager provides the use of object request broker with those of a mobile agent system.

Mole is a Java-based framework designed in University of Stuttgart for mobile agents. Major functionalities of include messages, RPC, and sessions. The research areas include security issues, communication between agents and agent groups and transactional by agents.

Another manufacturer of mobile agent is General Magic which creates Telescript, the first commercial mobile agent system. As Java is very popular language for Web-based applications. So General Magic planned to deploy a mobile agent paradigm in its Java-based Odyssey. Java class library was developed to create mobile agent applications.

1.2 Advantages of Mobile Agents

Mobile agent system should not be accepted on the basis of technology but should be chosen on the basis of benefits. Few benefits are as follows:

The main motive of mobile agent is to migrate computation to data rather than migrating data at the place of computation. In distributed system, lots of communication take place between autonomous systems, which not only consumes significant amount of time but also requires security. In the case of mobile agent, code gets execute locally. Since it can migrate itself to the executing environment, hence the significant amount of latencies also get reduces. As it does not require continuous connection for communication, it can work in disconnected mode.

Mobile agents can behaves dynamically to an unexpected situation, if a platform is being failed, all agents executing on that platform will be alert to continue their execution on other platform.

1.3 Mobile Agent Applications

Various applications of mobile agents are as follows:

Mobile agents are suitable for e-business. Mobile agent can negotiate the rate on online shopping or can compare rates with different quotes.

Mobile agent is suitable to work as a scheduler to schedule meeting on behalf of a platform. It can travel from one platform to another to schedule meeting according to the majority availability.

A secure transactional group can be made with only trusted mobile agents. The trusted platforms could let their mobile agents to meet on a secure platform. These mobile agents can purchase or sell various commodities.

Web crawler is an example of mobile agent. Information retrieval is a domain where mobile agent travels over multiple Web sites and collects the various information based on keywords

Mobile agent can work like a broadcasting and multicasting. All kind of push messages and software update can be done by mobile agent system.

Parallel processing can be done with the help of mobile agent. Any task can be divided into many parts, and each part may assign to various mobile agent. Once each subdivided task gets completed, it will be handover to the task originating platform.

With so many applications and advantages, mobile agent has security as an issue. Before deploying any agent on the platform, the originating platform should ensure about the reliability of the visiting platform. Next section discusses the security issues of mobile agent system.

1.4 Security Issues

Issues in mobile agent systems can be broadly categorized as issues originating from an agent attacking an agent platform, an agent platform attacking an agent, and an agent attacking another agent [2]. They are further classified into many attacks, but proposed work deals only with denial-of-service attack. Hence, all other classifications are not discussed here.

Agent Platform to Agent Attacks.
In these attacks, platform acquires the agent's code and data. This domain includes masquerading, denial of service, eavesdropping, and alteration. These kinds of attacks are very easy as platform can access code and data of the agent. Denial-of-service attack in this category is very easy. A malicious mobile agent could trap

the visiting mobile agent and denial to provide service for which mobile agent came. It may produce intolerable delay while execution of the line of code or even forcefully terminate the execution without intimation.

Agent to Platform.
In these attacks, agent acquires the execution environment of agent platform. This domain includes masquerading, denial of service, and unauthorized access. Denial-of-service attacks seem like a virus attacking over the system. A malicious agent can trigger a denial-of-service attack by acquiring excessive amounts of the platform's services and resources. A malicious agent may want to consume all computing resources of the executing platform so that the services provided by the platform cannot be used by other agents.

Agent to Agent.
In these attacks, a malicious agent acquires the code and data of other agent. This domain includes masquerading, denial of service, unauthorized access, and repudiation. Denial-of-service attack is very common in this category. Malicious agent can attack over other agent with the intention of denial of its services. For example, sending multiple requests again and again will cause unwanted pressure on the request handling routines of the agent.

2 Related Work

Trust and reputation have become important in security field. Many trust and/or reputation models are available recently. In this section, the most popular ones are elaborated.

In [3], authors suggest TRUMMAR as a reputation model in mobile agent systems where the reputation is computed on the basis of loss of reputation with time, prior-derived reputation, first impression, hierarchy of platforms, and reported reputations from other platforms that wish to volunteer for information. The reputation value is calculated as

$$\begin{aligned} \mathrm{RepA/B(0)} = \mathrm{repA/B} + \mathrm{q}\left(\sum \alpha \mathrm{irepA/Bi}\right)/\left(\sum \alpha \mathrm{i}\right) \\ + \mathrm{r}\left(\sum \beta \mathrm{irepA/Bi}\right)/\left(\sum \beta \mathrm{i}\right) + \mathrm{s}\left(\sum \delta \mathrm{irepA/Bi}\right)/\left(\sum \delta \mathrm{i}\right) \end{aligned} \quad (1)$$

where

- RepA/B(0) is reputation being computed now of A with respect to B.
- RepA/B is last computed reputation of A in view of B.
- $\sum \alpha$i repA/Bi is the weighted sum of reputations of A as informed by the one hop distance of B (Bi).
- $\sum \beta$i repY/Xi is the weighted sum of reputations of A as informed by the nodes that are more than two hops distance of B (Bj).

- $\sum\delta$i repA/Bi is the weighted sum of reputations of A as informed by unknown node in the network that voluntarily to give reputation of B.
- δ, β, α are weighing factors computed according to the reputation of the individual one-hop distance nodes, more than two-hop distance node, and unknown in the host network, respectively.
- p, q, r, and s are weighing factors corresponding to the previous reputation of A in view of B, one-hop distance nodes of B, nodes greater than two-hop distances of B, and alien in the agent network.

Cubaleska and Schneider [4] propose a mathematical model for a posteriori identification of malicious platform to develop a trusted system. On the basis of trust value of originating host to other, it can either prepare an itinerary in which reliable platform to be visited are mentioned, or it can choose the platform.

In [5], authors proposed a method for computing reputation on the basis of position of every person of a community within the social network. Local information is used to compute reputation value. A different algorithm, node ranking, was proposed to obtain such computations.

In [6], authors proposed Sporas as a reputation model in mobile agent system, where the reputation is calculated recursively and latest rating gets more weight. Hence, reputation rating at time i, Ri, is computed recursively from the previous reputation Ri−1 and the purchase rating Pi as:

$$\mathrm{Ri} = \mathrm{Ri} - 1 + (1/\theta) \cdot \Phi(\mathrm{Ri} - 1) \cdot (\mathrm{Pi} - \mathrm{Ri} - 1) \tag{2}$$

where θ is the ratings, Pi is a rating taken from user i, and function Φ is defined to gradually decrease the value for reputable users as:

$$\Phi(\mathrm{Ri} - 1) = 1 - 1/(1 + \mathrm{e} - (\mathrm{Ri} - 1 - \mathrm{M})/\sigma) \tag{3}$$

where M is the maximum reputation and σ is the accelerating factor of the function Φ. So, lesser the value of σ, the steeper the damping factor Φ(R).

3 Proposed Model

Designing a reliable and secure trust and/or reputation model for the execution of line of code without denial-of-service attack is a new research area. In this section, a model for computing the trust and reputation value is presented. Let us consider the situation in Fig. 1, where an agent platform S1 wants to send a mobile agent MA1 on anther platform say S2 in order to execute line of code for the task. Platform S1 will not send the agent unless it is sure that platform S2 is trusted one, i.e., that platform S2 will provide a trusted environment which will not masquerade, alter or edit the agent's code, data, or status. In the mechanism of finding the trusted platform,

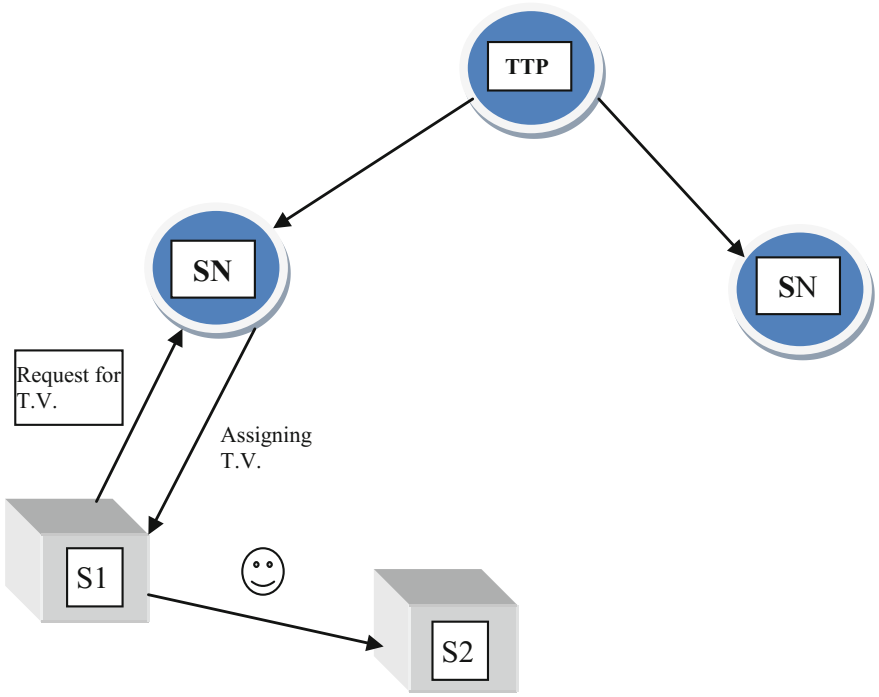

Fig. 1 The proposed model (trust and reputation model)

platform S1 will calculate a reputation value for the platform S2. This reputation is calculated by taking the average of all its mobile agent's trust values.

All trust and reputation values will be maintained by some centralized supporting nodes (SN) which are behaving like a trusted third party (TTP). Since the mobile agent has to work in distributed environment, hence the number of supporting nodes (SN) varies according to the geographical area. Figure 2 shows the process in trust and reputation model.

In this model, information about the trust and reputation can be retrieved by any SN. These collected trust and reputation values then aggregated for choosing the right platform for the execution of the mobile agent. Once the execution gets over, the values of trust and reputation get modified according to the feedback given by executing platform.

Algorithm
Input: TTP, SN, S1, S2, MA1
Output: Reputation and trust values to S1, S2, and MA1
Step 1: S1 is creating a mobile agent MA1
Step 2: S1 will send its unique address (MAC) to Supporting Node SN
Step 3: SN will check S1's unique address (MAC) and compute trust and reputation values for S1

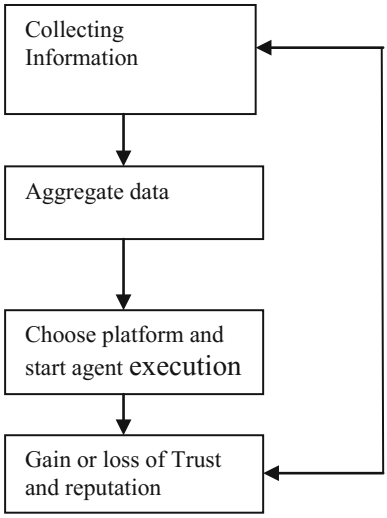

Fig. 2 The process in the model (flow chart)

(a) Assigning initial trust value to platform (If S1 is gaining trust and reputation values first time)
- Initial trust value will lie between α and β, where α is initial value of the trust value, which is considered to be minimum and β is maximum value of the trust value a mobile agent can have.
- Initial reputation can be calculated by taking the average of the trust values of all mobile agents under that originating platform.

(b) Increment and decrement of trust and reputation values on the basis of type of services performed, i.e., read and write.
- Increment in trust value on successful execution of services (reported by executor)

I. If service performed is read, then its trust value get increased by γ
II. If service performed is write, its trust value get increased by δ
- Decrement in trust value on execution of services more than the number of times allowed (reported by executor)

I. If service performed is read, then its trust value get decreased by γ
II. If service performed is write, its trust value get decreased by δ

Step 4: Computing the maximum number of service request for preventing DoS. Maximum number of request Maxrq can be calculated as follows:

$$\text{Maxrq} = \text{M x}\left(\text{TV(S1)}/\sum \text{TV}_{\text{Si}}\right) \tag{4}$$

where M is the total number of services and a platform S1 can perform over a time interval.

Step 5: Declaring the denial-of-service attack

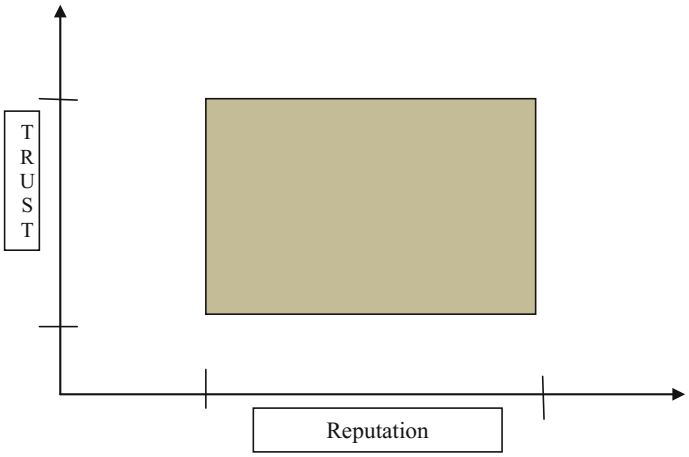

Fig. 3 Trust and reputation thresholds

Once the trust value of any mobile agent gets less than α, the mobile agent will be declared as a malicious agent and its contribution in reputation of its originator will be canceled. Ultimately, the reputation of the originator gets reduced.

In Fig. 3, shaded area shows the trusted and reputed platform over which the mobile agent is secure from denial-of-service attack. Reputation is maintained by taking the average of trust values of all mobile agents of a platform.

$$\text{Rep(S1)} = \sum \text{TV}_{\text{MAi}}/\text{n} \tag{5}$$

where n is total number of mobile agents generated by S1.

If the trust value of any mobile agent gets increased, then reputation of its originator automatically increased. Similarly, the decreased in trust value of any mobile agent reduces the reputation of its originating platform.

4 Results and Analysis

In order to compare proposed model with other trust and reputation model, consider the environment shown in Fig. 1. S1 and S2 are autonomous platform capable enough to create mobile agents and provide execution environment to them. Supporting nodes (SN) provide replica of database stored at trusted third party. As mobile agent works in distributed network; hence, the supporting nodes (SN) are situated at various geographical regions.

The table for maintaining the data for agent's trust and platform's reputation is maintained by SN, and it gets updated on the basis of agent's platform interaction.

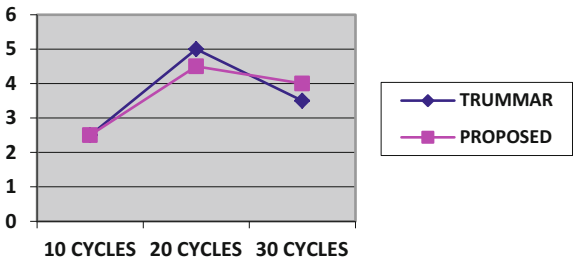

Fig. 4 Behaviour of reliable platform's reputation

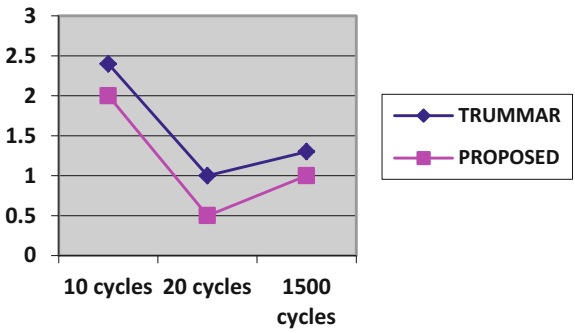

Fig. 5 Behavior of malicious platform

To verify proposed model, implementation is done in JADE. The system consists of four platforms S1, S2, S3, and S4. Only one communication can occur at any time. The minimum trust value for any mobile agent is 0.5, and the maximum trust value for an agent is 5. Similarly if a platform is found "untrustworthy," its corresponding reputation value gets reduced by the trust value of the agent, otherwise its value gets increased. The trust value of a reliable agent increases gradually from an initial value from 0.5 to a value around 5. An agent's trust value decays exponentially with respect to the number of execution. However, the reputation of the platform depends on average trust value of all its mobile agents.

A comparison between the TRUMMAR [3] and proposed model is shown in Fig. 4, where x-axis shows the number of times the services are demanded from platform and y-axis shows the reputation values of the platform.

Model proposed in [3] increases and decreases the reputation value more drastically but proposed model, changes reputation value gradually as shown in Fig. 4.

The reputation of malicious platform decays from initial value to a value less than 0.5. The same effect that was analyzed for reliable platform is also analyzed for malicious platform. Figure 5 shows that the reputation value increases gradually, once it reaches to its minimum.

Model proposed in [7] uses group in-charge and itinerary to detect malicious host. The proposed model uses the reputation value for detecting malicious host. The proposed model consumes less time in comparison with model in [4] as shown in Fig. 6.

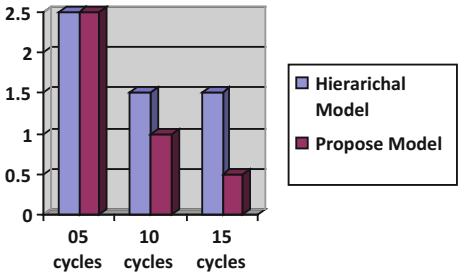

Fig. 6 Comparison between two models

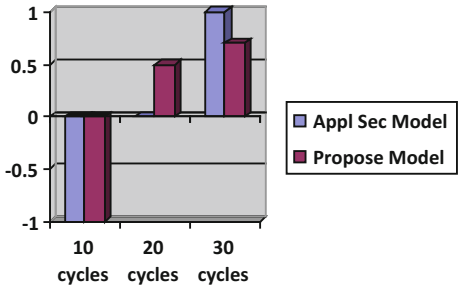

Fig. 7 Comparing reputation values for two models

Model proposed in [8] uses trust value as (HT, T, ND, U, HU) from [−1, 1] scale. It builds a trust relationship with security requirement and security mechanism. The proposed model is more accurate as it computes trust in discrete values, while model in [8] uses range to check trustworthiness of an agent. Figure 7 shows the results.

5 Conclusion

Proposed model is a trust and reputation-based model that mobile agent system can implement to avoid denial-of-service attacks from agent-to-platform as well as platform to agent. Numerous of trust and/or reputation models have been previously implemented in research, but proposed model can handle both agent-to-platform and platform to agent denial-of-service attacks.

References

1. Lange, D.B., & Oshima, M. *Introduction to mobile agents.* General Magic Inc., Sunnyvale, California, USA.
2. Jansen, W., & Karygiannis, T. (1998). Mobile Agent Security, U.S. Department of commerce, NIST, Spec. Publ. 800-19, p. 44 (Oct. 1999), Gaithersburg, MD 20899-8930.
3. Derbas, G., Kayssi, A., Artail, H., & Chehab, A. (2004). Trummar-a trust model for mobile agent systems based on reputation. In *Proceedings of the IEEE/ACS International Conference on Pervasive Services, 2004 (ICPS 2004)* (pp. 113–120). IEEE.
4. Cubaleska, B., & Schneider, M. (2002). Applying trust policies for protecting mobile agents against DoS. In *Third International Workshop on Policies for Distributed Systems and Networks (POLICY),* Moterey, California.
5. Pujol, J. M., Sangüesa, R., & Delgado, J. (2002). Extracting reputation in multi agent systems by means of social network topology. In *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems*, Bologna, Italy.
6. Zacharia, G., & Maes, P. (2000). Trust management through reputation mechanisms. *Applied Artificial Intelligence, 14,* 881–907.
7. Aggarwal, M., & Nipur, P. (2012). Hierarchal model to prevent DoS attack in mobile agents. In *International Conference in Recent Trends in Information Technology and Computer Science (ICRTITCS-2012) Proceedings published in International Journal of Computer Applications®(IJCA)* (pp. 0975–8887).
8. McDonald, J. T., & Yasinsac, A. (2006). Application security models for mobile agent systems. *Electronic Notes in Theoretical Computer Science*, *157*(3), 43–59; In *Proceedings of the First International Workshop on Security and Trust Management (STM 2005).*
9. www.tryllian.com/technology-advancement-with-mobile-agentsoftware/12/05.2013.
10. Mármol, F. G., & Pérez, G. M. (2010). Towards pre-standardization of trust and reputation models for distributed and heterogeneous systems. *Computer Standards & Interfaces*, *32*(4), 185–196.