Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 9, September 2014, pg.510 – 516

RESEARCH ARTICLE

SECURED TRUST COMMUNICATION IN MULTI-AGENT SYSTEMS

Anuradha Balasubramanian
Assistant Professor, Info Institute of Technology

Abstract: - Nowadays security and privacy issues take a major role in multi-agent system. Mostly multi-agent systems are open and dynamic in nature. This nature surely introduces a problem by providing secured communication. Message-Digest5 is presented to some security problems in multi agent systems based on distributed trust and the delegation of permissions and credibility. In particular, an agent will receive requests and assertions from other agents and must decide how to act on the requests and assess the credibility of the assertions, because sometimes malicious agents start to behave in unpredictable way. The multi-agent systems which is becoming critical for sustaining good service quality, is the even distribution of workload among service providing agents. For that a dynamic trust computation model called secured trust is introduced. This reduces to authentication the reliable identification of agents' true identity. In this project the Multi-Agent System (MAS) concepts is applied to facilitate the authentication and the authorization process in order to work with multi-clients more dynamically and efficiently. The key pair and Certification Authority are deployed to encrypt/decrypt electronic data or transaction, or sign/authenticate the sender and the recipient.

Index Terms: Multiagent system, secured trust, load balancing, dynamic trust, malicious agents

I. INTRODUCTION

Network security consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Once authenticated, a firewall enforces access policies such as what services are allowed to be accessed by the network users. Though effective to prevent unauthorized access, this component may fail to check potentially harmful content such as worms or Trojans being transmitted over the network. The components of Network Security are:

- Anti-virus and anti-spyware.
- Firewall, to block unauthorized access to your network.
- Virtual Private Networks (VPNs), to provide secure remote access.

In a Multi-agent System (MAS), agents interact with each other to achieve a definite goal that they cannot achieve alone. Multi-agent Systems are increasingly becoming popular in carrying valuable and secured data over the network. Malicious agents are always seeking ways of exploiting any existing weakness in the network. Multi-agent System is a system composed of multiple interacting intelligent agent within an environment. A multi-agent system (MAS) is a system composed of multiple interacting intelligent agents. Trust issues have become more and more popular since traditional network security approaches such as the use of firewall, access control and authorized certification cannot predict agent behavior from a ‘trust’ viewpoint. An agent creates trust-preferred securities by creating a trust issuing debt to the new entity, while the trust issues the trust preferred securities. We have used a novel policy of utilizing exponential averaging function to reduce storage over head in computing the trust of agents.

A reputation-based trust model collects, distributes and aggregates feedback about participants’ past behavior. These models help agents decide who to trust, encourage trustworthy behavior and discourage participation by agents who are dishonest. Reputation-based trust models are basically divided into two categories based on the way information is aggregated from an evaluator’s perspective. They are “Direct/Local experience mode” and “Indirect/Global reputation model” where direct experience is derived from direct encounters or observations (firsthand experience) and indirect reputation is derived from inferences based on information gathered indirectly (secondhand evidence such as by word of mouth). So, in the case of global reputation model an agent aggregates feedback from all the agents who have ever interacted with the target agent, i.e., an agent has a view of the network which is wider than its own experience, thus enabling it to quickly converge to a better decision. However, global reputation model is much more complex to manage than local experience model as malicious agents have the opportunity to provide false feedbacks.

Most of the existing global reputation models can successfully isolate malicious agents when the agents behave in a predictable way. However, the model suffer greatly when a

gents start to show dynamic personality, i.e., when they start to behave in a way that benefits them. This model also fails to adapt to the abrupt change in agents’ behavior and as a result suffer when agents alter their activities strategically. Moreover, some of the models show little effect in dealing with more complex attacks such as dishonest or unfair rating and collusion. Another aspect which is slowly becoming more critical for the proper maintenance of service quality is the appropriate distribution of workload among the trusted service providers. Without a proper load-balancing scheme, the load at highly reputable service providers will be immense which will eventually cause a bottleneck in the system’s service quality. To the best of our knowledge, none of the existing trust models consider load balancing among service providers.

With these research problems in mind, we propose a key authentication dynamic trust computation model named Secured Trust which can effectively detect sudden strategic alternation in malicious behavior with the additional feature of balancing workload among service providers. Secured-Trust considers variety of factors in determining the trust of an agent such as satisfaction, similarity, feedback credibility, recent trust and historical trust, sudden deviation of trust and decay of trust. We have used a novel policy of utilizing exponential averaging function to reduce storage overhead in computing the trust of agents. We have also proposed a new load-balancing algorithm based on approximate calculation of workload present at different service providers.

II. RELATED WORK

In this section, we look into some of the most recent and popular research works done on reputation model. Here, we discuss the key ideas of the following models – Bayesian network trust model trust model

Bayesian network-based trust model [3] believes that trust is multidimensional and agents need to evaluate trust from different aspects of an agent’s capability. This model uses Bayesian Network and Bayesian Probability to calculate trust. This model’s main flaw lies in the authors’ assumption that all the agents have identical Bayesian network architecture which is unrealistic because different agents have different requirements which leads to different network architecture. In the case of aggregating recommendation from other agents, this model assumes that all the agents are truthful in providing their feed-backs. This assumption is also not realistic as malicious agents will often provide false feedback to other agents to disrupt the system.

The Trust model provided by addresses different aspects in determining the trust of an agents such as recent trust, historical trust, expected trust and confidence in its trust for other agents. However, in computing direct trust, this model uses simple averaging function which fails to assign any time relative weight to the transactions. Another drawback of this model is that the formulation of both historical trust and credibility requires the storage of previous values up to a certain interval and this causes storage overhead.

The models do not address a critical aspect of trust theory which is decay of trust value with the elapse of time. Since, at percent, the network is highly dynamic and unpredictable, trust values should decay as time elapses in absence of interaction. However, these models fail to simulate real life decay function which has a small decay rate in the initial phase while displaying higher decay rate as more and more time elapses. We have incorporated such decay function in our trust model along with many other issues which have not been addressed by existing trust models.

Another aspect which is slowly becoming vital for sustaining service quality is the balanced distribution of workload among service providers. Almost all trust models have ignored this issue so far. In fact, none of the models discussed so far address the aspect of balancing load among the trusted agents for proper maintenance of service quality. In a trust computing environment, an agent with the highest trust is normally selected s service provider, so highly reputed agents handle bulk of the total service requests. This can degrade the overall service quality of the system if these highly reputed agents are assigned too much workload. So, a load-balancing algorithm which distributes service requests to all capable (i.e., a bit less reputable but trustworthy) agents is required to maintain a satisfactory level of service quality. We have proposed such a load-balancing algorithm.

III. IMPLEMENTATION

Secured Trust:

The main objective of this paper is to provide a dynamic trust computation model for effectively evaluating the trust of agents even in the presence of highly oscillating malicious behavior. Our model also provides an effective load-balancing scheme for proper distribution of workload among the service providing agents. A number of parameters have been considered in our trust model for computing the trust of an agent. Now, some of these parameters have been previously discussed in [2] but it can fully cope with the strategic adaptations made by malicious agents. The mathematical and logical definitions used for these parameters also cannot reflect the true scenarios faced in real life. For the following sections, we assume that agent p (called evaluator) needs to calculate the trustworthiness of agent q (called the target agent).

Load Balancing Among Agents:

Load balancing is a computer networking methodology to distribute workload across multiple agents or other resources, to achieve optimal resource utilization, maximize throughput, minimize response time and avoid overload. For selective scenario, we first compute the trust of agents who respond to a transaction request and then we select the agent with the highest trust value. The problem that will arise from this disproportionate allocation of workload is that the quality of service will fall greatly due to the heavy workload present at the highly trusted agents.

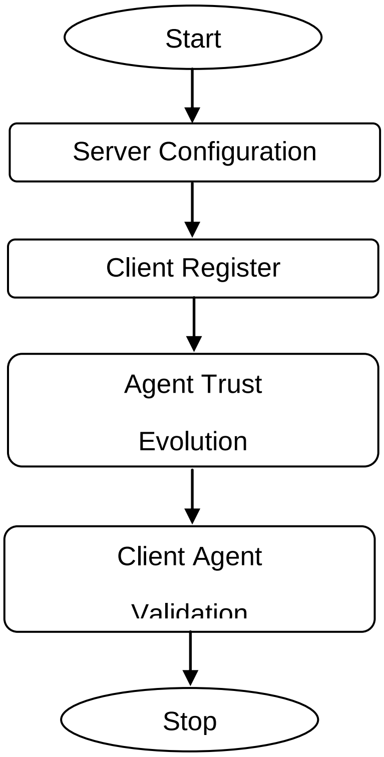
In our load-balancing algorithm, we either calculate a heuristic value of workload and choose the agent with the smallest load or make a probabilistic choice based on the computed trust value of agents. We start our load-balancing algorithm by first classifying the responders (agents that respond to a transaction request) into two groups, namely-good service providers (G) and unknown service providers (U) based on a threshold value of trust (γ). We then first seek to choose an agent from G by computing an approximate value (heuristic value) of load present at each responder in G. Sorting the responders in increasing order of load; we take the responder with the smallest workload. In the case of no responders being present in the class G, we select an agent from U either probabilistically based on its trust value or randomly.

Algorithm:

```
Selection of service providing agent (p,S)
Input: Evaluating agent p and the set of agents responding to a service request S
Output: Service providing agent q
for each x Є S do
    compute Trust (p,x)
    if Trust (p,x) > $\gamma$ then
    $G \leftarrow G \cup \{x\}$
    else
    $U \leftarrow U \cup \{x\}$
    end if
end for
if $G \neq 0$ then
    for each x Є G do
    compute load N(p,x)
    end for
    sort G in increasing order of load N
    return agent q with the smallest load N
else
    Total_trust ← 0
    for each x Є U do
        Total trust ← Total_trust + Trust(p,x)
end for
if Total_trust > 0 then
    for each x Є U do
    compute Prob(p,x)
    end for
    return agent q with probability Prob(p,q)
    else
        return any agent q randomly
    end if
end if
```

IV. SYSTEM MODULES

DATA FLOW DIAGRAM



Server Configuration

A server is a physical computer dedicated to running one or more services to serve the needs of the users of other computers on the network. Depending on the computing service that it offers it could be a database server, file server, mail server, web server, or some other kind of server. The clients either run on the same computer or connect through the network. Every agent has different needs, and the different servers all represent different sets of trust.

1. Client Registration

Different agents have different requirements for registered agents. Typically, the agent must be a legal resident of the state that allows entities to serve as registered agents. Client Application Services is the name of the client and services framework. In order to import multiple clients, you must:

- ❖ Create the clients definition file
- ❖ Validate and import the clients definition file

2. Agent Trust Evolution

Agents generally interact by making commitments to one another to carry out particular tasks. In most realistic environments there is no guarantee that a contracted agent will actually enact its commitments. It could interact with all agents and then derive trust measures from the history of interactions. The calculation and measurement of trust in unsupervised virtual communities like multi agent environments involves complex aspects such as credibility rating for opinions delivered by peer agents.

3. Key Pair Generation

The Key Pair Generator class is used to generate pairs of agents. All key pair generators share the concepts of a key size and a source of randomness. There is an initialize method in this Key Pair Generator class that takes these two universally shared types of arguments.

4. Client Agent Validation

The agents in the Multi-Agent System are able to gather data by generation of logs as well as provide run-time validation and verification support by watch agents and also agents to check any violation of invariants at run-time. The validation is an essential parts of the model development process if agent to be accepted and used to support decision making. The service agent expects all the credentials necessarily at the time of request. In order to use its services, a requesting agent must send all required credentials along with the request for service.

5. File Transfer

File transfer is a generic term for the act of transmitting files over a computer network like the Internet. There are numerous ways and protocols to transfer files over a network. File transfer is the movement of one or more files from one location to another. The File Transfer is a common way to transfer a single file or a relatively small number of files from client to sever.

V. CONCLUSION

We have presented a novel trust computation model called Secured Trust for evaluating agents in multi agent environments. Secured Trust can ensure secured communication among agents by effectively detecting strategic behaviors of malicious agents. We also provide a model for combining all these factors to evaluate trust and finally, we propose a heuristic load-balancing algorithm for distributing workload among service providers. This approach is particularly useful in open environment in which agents must interact with other agents with which they are not familiar.

REFERENCES

[1] K. Aberer and Z.Despotovic, **“Managing trust in a peer 2 peer information system,”** in Proceedings of the tenth international conference on Information and knowledge management (CIKM). ACM, 2001, pp. 310–317.

[2] B.Li, M.Xing , J.Zhu , and T.Che, **“A dynamic trust model for the multi agent systems,”** in Proceedings of IEEE International Symposiums on Information Processing (ISIP), 2008, pp. 500–504.

[3] Y.Wang and J.Vassileva, “**Bayesian network based trust model**,” in Proceedings of IEEE/WIC International Conference on Web Intelligence (WI), Halifax, Canada, October 2003, pp.372–378.

[4] M. Gupta, P. Judge, and M. Ammar, “**A Reputation System for Peer-to-Peer Networks**”, Proc. 13th Int’l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV ’03), pp. 144-152, 2003.

[5] E. Damiani, S.D.Capitani, S.Parabo schi, and P.Samarati, **“Managing and sharing servants’ reputations in P2P systems”,** IEEE Transaction on Knowledge and Data Engineering, vol.15, pp. 840–854, 2003.

[6] A.A. Selcuk, E.Uzun , and M.R.Pariente, **"A reputation based trust management system for P2P networks,"** in Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid (CC GRID), 2004, pp . 251–258.

Biography

B.Anuradha had completed Bachelor degree in Electronics and Communication at K.S.Rangaswamy College of Technology,Tiruchengode.she had worked as faculty in networking areas. Currently she had completed master degree in VLSI at Karpagam University,Coimbatore. Now working as a Assistant Professor in Electronics and communication Engineering at INFO institute of Engineering. Her Areas of interest are Networking,Testing, and IC fabrications.