Protecting Mobile Agent using Enhanced Reference Monitor based Security Framework

Oladeji P. Akomolafe
Department of Computer Science
University of Ibadan, Ibadan, Nigeria

Attah H. Honesty
Department of Computer Science,
University of Ibadan, Ibadan, Nigeria

ABSTRACT
The introduction of mobile agent technology to distributed systems presents lots of benefits which include reduced network bandwidth consumption and network latencies, load balancing, etc. One critical issue with mobile agent technology is the security of its data and code against malicious hosts. The different security mechanisms proposed to offer protection seems to suffer some shortcomings that impact the performance of mobile agent's execution. This paper presents an efficient security framework for the protection of mobile agents.

Our proposed enhanced security framework deploys a trusted third party technique designed to allow the offloading of the computation-intensive verification mechanism for execution.

This approach ensures that resource consumption by the verification mechanism of the reference monitor agent on the hosts is largely reduced, thus performing more efficiently than the existing system

General Terms
Mobile agent system, Security

Keywords
Security, Mobile Agent, Reference Monitor, Trusted Third Party

1. INTRODUCTION
Before the advent of mobile agent programming paradigm, distributed computing over the years has been implemented using the most basic message passing model, remote procedure calls, peer-to-peer model, distributed objects, publish-subscribe model and the widely-used client/server computing model to mention a few. These models require stable and consistent open network connection for interactions among the computing components situated in remote hosts. More so, as such the volume of data exchange causes more network traffic. The method of communication is either through message passing or remote procedure call (RPC) which usually takes a long time since it is a synchronous mode, [1]

The mobile agent computing presents lots of other benefits which include reduced network bandwidth consumption and network latencies, load balancing, fault tolerance and network management especially in wireless networks, thus, making it possible to implement many applications like e-commerce, real-time control systems and cloud computing more flexibly and efficiently, [2]

However, mobile agents are faced with different security threats which has been categorized into four major perspectives [3] to include:

- Agent attacks against an execution platform.
- Platform attacks against a mobile agent.
- Agent attacks against another agent in the same execution platform.
- Exterior entity attacks against an agent or a platform.

These attacks are primarily focused on the communication capability of the platform to exploit potential vulnerabilities. The security of mobile agent has become a serious problem that needs deep consideration [4], because the platform of execution has access to all the components of the mobile agent. Since, a platform has the responsibility to execute a mobile agent; it is assumed that the platform must have full access to agent's code and data. A platform might be malicious and may try to execute the code in a manner in which it is not authorized to do. A platform may try to change agent's state, code or routing during agents' execution [5]. Thus, the focus in this section will be geared towards research proposals that address threats posed to mobile agents by malicious hosts

Mobile Agent consists of code, data (information storage for intermediate results and private variables) and state information. The mobile agent system provides an environment where mobile agents execute by hopping from one host's execution platform to another host's platform in order to accomplish its tasks. During migration, the mobile agent suspends its execution on the current host's platform (i.e. in a frozen sate), carries its code, data (partial results) and state information to the next host where it resumes execution from where it left off. There are two types of mobility: the first, strong mobility which implies that the mobile agent moves with code, data and execution state. When the mobile agent arrives at next node, it will start its execution from the end point in the previous node. The second being weak mobility means that the mobile agent moves with the code and the data only. In this case, the execution will start from the beginning upon arrival at the next node. Hence, they have the capability to execute in an open network environment, characterized by heterogeneously trusted and untrusted host platforms. The implication is that, this raises a serious security concerns that need urgent solutions that will ensure the availability, integrity and confidentiality of agent's code, data and itinerary requirements. Therefore, the proposed method concentrates on tackling the malicious hosts' problem in this work.

The rest of this paper is structured as follows: section 2 discusses related work. section 3 presents the proposed enhanced security framework based on trust provided by the Reference Monitor for providing a trusted computing environment. Section 4 presents the experimental setup used and results. Finally, section 5 presents the conclusions drawn and outlines future work.

2. RELATED WORKS

Several approaches for the protection of mobile agent have been proposed by a number of researchers. They try to ensure the access of the mobile agent to hosts in which it may have confidence or detect those that are malicious. Essentially, they are intended to detect attacks or render them ineffective. The proposed mechanism is based on preventing attacks from malicious hosts.

An apparent solution proposed by [6] is the use of tamper-proof hardware at the site of every remote host as provided by a trusted third party to interact with the agent's platform. The goal of this approach is to impede any form of modification of agent's code, data and state by creating a completely secured executing environment for the agents. Even though this mechanism ensures confidentiality, integrity and secure communication between the agents and its host's platform, it introduces high cost of installation and maintenance at every host to be visited by the mobile agents in an open environment, consequently making it unattractive.

[7] defined obfuscation as an approach where an agent owner (home agent platform) enforces a security policy by applying a behavior preserving transformation to the code before it is dispatched to run on unknown and known hosts. The code obfuscation creates a blackbox entity where agent's program is illegible and hides data thereby rendering it difficult for attackers (malicious hosts) to read and modify the agent's code, its data and partial results. However, this approach offers short-lived code obfuscation as given time, the malicious platform will be able to de-obfuscate the code

[8] proposed an approach based on time limited code obfuscation. He applied a conversion mechanism where new agents (messed up programs) are generated from the original agent using different configurations of parameters. These new agents are then given the protection time attribute specifying how long they will remain obfuscated after which their code and data will be susceptible to any attack and no longer be needed, even though it has been successfully attacked. Nevertheless, this technique suffers from sabotage and blackbox testing threats. Another method where the execution time of mobile agents is being monitored was proposed by [9]. The implication is that, if the mobile agent spends longer than expected time on the agent platform, then there is an indication that the agent platform might be attempting to de-obfuscate it or replay it. Although this technique provides security for mobile agents, it requires a connected and synchronous mode to allow agents return partial results home on every visited platform, which represents its shortcoming.

Moreover, among the mobile agent's security frameworks designed to prevent most threat attacks is the multi-facet security approach proposed by [10]. This security strategy is purposely to contain most of the threats dogging multi-agent systems, especially for the malicious hosts attack. In their work, Time-To-Live TTL and heartbeat methods were adopted as technique to tackle the DoS attack, encryption method provided confidentiality and integrity of agents' code. However, the mechanism failed to offer protection against the repudiation attack. [11] proposed a pragmatic technique for providing security for mobile agents against malicious host platforms in an open network environment. This mechanism is based on Reference Monitor Concept, a reference validation mechanism, earlier introduced by [12] and adopted for securing computer and network. This security framework combines cryptography methods, data encryption and access control methods to tackle major threat attacks. It deploys reference monitor RM agents to enforce system's security policy on remote platforms for mobile agent's execution. The drawback is that, the RM agents' integrity and authenticity verification process results to increased computational overheads on the agent platforms due to the several hash value computations and high memory consumption from several destination RM agents' files.

[13] have proposed the use of a chain of digital envelopes with platform registries to support dynamic agent's itineraries in open network environment. The main advantage is that the proposed scheme exhibited better performance when compared to the results obtained from obfuscation methods in terms of data integrity and security. The main drawback is that the proposed scheme consumes a little more time visiting platform registries and executing complex cryptographic functions than the obfuscation methods. [14] presented a security solution that protects both the mobile agent itself and the host resources that encrypt the data before passing it to mobile agent and decrypt it on the visited host sides. No provision is given to secure code from other malicious agents. [15] proposed a secure system for deployment of mobile agents. The system provides methodology that spans a number of phases in agent's lifetime: it starts from agent creation and ends with agent's execution. It addresses classification, validation, publishing, discovery, adoption, authentication and authorization of agents. This system is based on secure web services and uses RBAC XACML policies and SAML protocol. Though the work authenticates the code, the integrity of the code is not assured, and uses asymmetric encryption repeatedly, thus increases the execution time of the mobile agents.

A self-adaptive model in which mobile agents are configured and may be reconfigured at runtime by assembling reusable components in response to changes in execution environment was proposed by [16]. [3] also proposed a similar model that provides security mechanisms as components to mobile agents according to the sensitivity of their required services and the degree of credibility of the agent platforms and allowing for static and dynamic adaptation of the security mechanisms by mobile agents during runtime. The shortcomings are space consumption and computational burden from the added extra data for the mobile agent.

3. ENHANCED REFERENCE MONITOR BASED SECURITY FRAMEWORK

The malicious platform is an entity in the mobile agent system that is able to execute an agent that belongs to another party and that tries to attack that agent in some way [3]. As the mobile agents are executed in the host, the host allocates all the resources, so that it can launch lots of different kinds of actions to attack the agents. Hence, reference monitor based security framework was adopted by making some modifications most especially to adequately equip the reference monitor agent to ensure better secure environment and allow efficient migration of roaming agents to host platforms in an open network. In order to achieve this, conceptually the system is divide into four subsystems which include:

- Agent Creation Point
- RM Agent Policy Specification Point
- Agent Evaluation Point
- Agent Execution Area

3.1 Agent Creation Point

This is the home platform where roaming agents are created, initialized with owner's or user's tasks, validated and published to the network environment.

3.2 RM Agent Policy Specification Point

This area requires the use of a trusted third party in which system-wide security policy to be enforced on every remote host that is to execute users' roaming agent are specified and the reference validation mechanisms created, which are then integrated to the reference monitor agent. Here also, the digital certificate and signature for RM agents are created The remote platform registers with the PSP, make requests to it for RM agent and then, it is safely distributed and integrated into the platform. The presence of the RM agents in any remote platform turns it into a trusted computing base in an open system

3.3 RM Agent Evaluation Point

This is actually where RM agents are verified. In the existing framework, the verification of the integrity and authenticity of RM agent was being carried on each host, but in the proposed approach, the novel approach introduced includes a trusted third party. Thus, the trusted third party is made to perform the entire computations required in the verification process before roaming agents are migrated to destination host. The figure 3.1 gives the model of the proposed system.

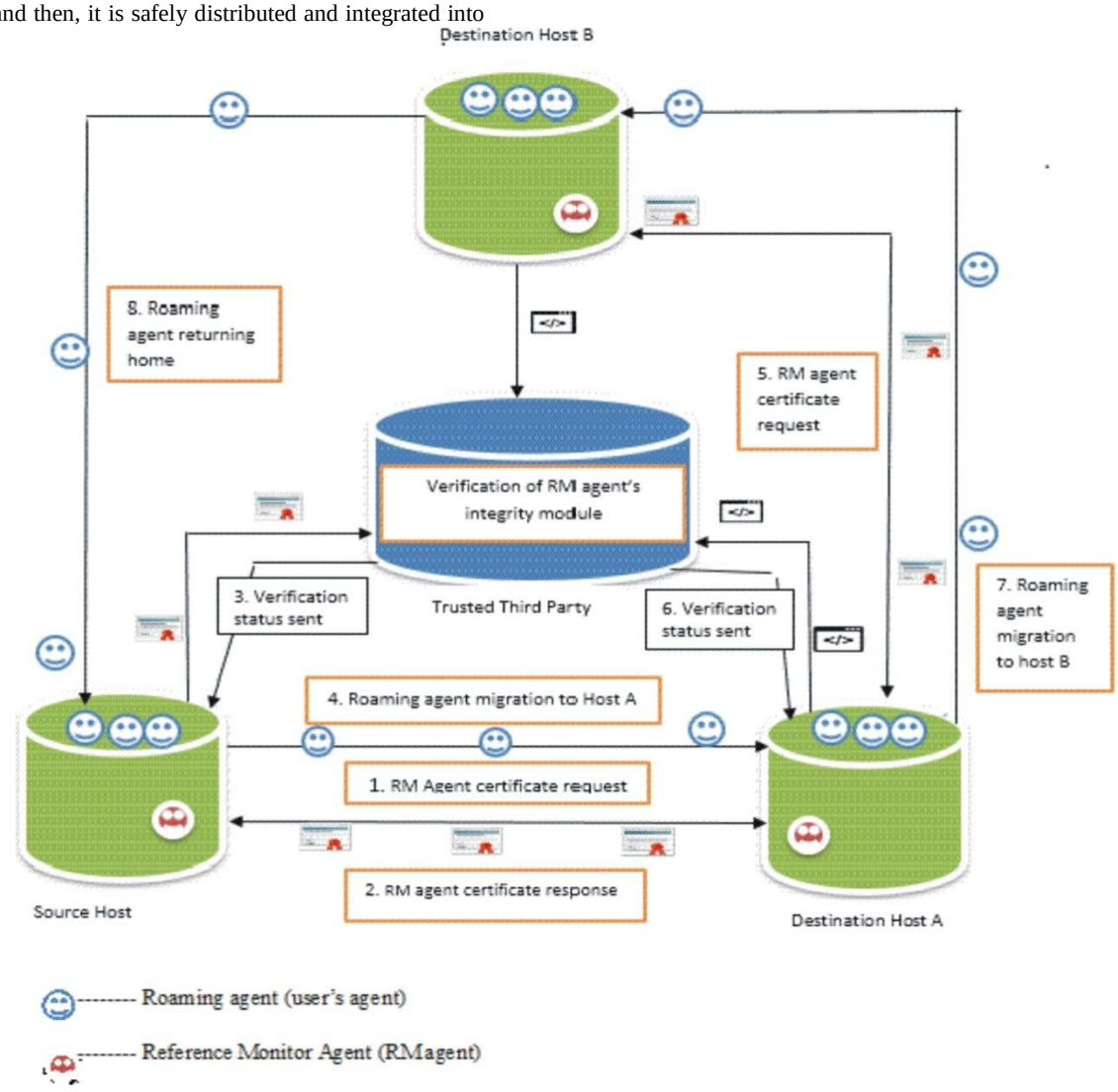

Fig 3.1 : System Architecture of Proposed Model

3.4 Agent Execution Area

This contains actual runtime component that provide needed services for the roaming mobile agents' execution and the reference monitor agent.

3.5 RM agent Verification Mechanism

The figure 3.2 clearly depicts the steps required in the verification of the integrity of the RM agent as performed by the trusted third parry. Firstly, the trusted third party check the validity of the destination host RM agent certification, thereafter concatenate the source RM agent challenge with source file of RM as distributed and that of the running source file of the RM agent executing on the next destination host, then compute their hash values. Thus, it then compares these values to find whether there is a match. In the case where there is a match, it can sufficiently infer that the next destination host is execution-safe for the roaming agent destined there, else it reports destination host is unsafe and next destination host in roaming agent itinerary be considered and the process repeats. In the existing security framework, this verification process was being performed by source RM agents using the limited local resources of host.

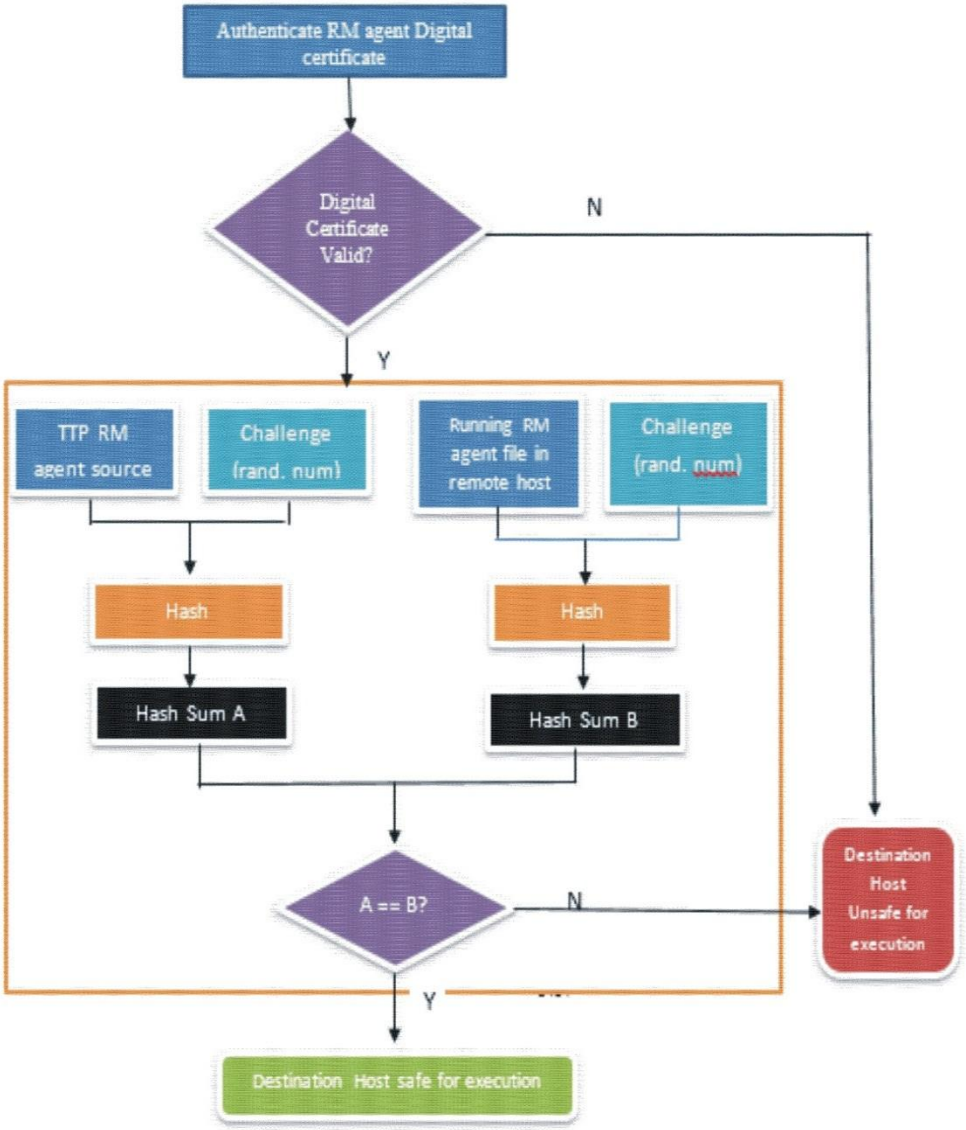

Fig 3.2: Verification of RM agent integrity and authenticity Mechanism [7]

4. EVALUATION AND RESULTS

The implementation builds the system architecture designed and generates result accordingly which is then use to perform the evaluation process. The experiment was performed using a prototype model with which agents migrate and give responses. The implementation was carried out using JADE. JADE (Java Agent Development Environment) is a software development framework aimed at developing multi-agent systems and applications conforming to FIPA standards for intelligent agents. For the purpose of a comprehensive evaluation, two experiments were performed which entails the implementation of both the existing and proposed security frameworks adopting the Book Trading System as a test bed. In this environment, all the hosts had the reference monitor agent deployed to enforce security policy that ensures secure environment for agent's execution. One of the hosts was dedicated as the agent home platform where the user's roaming agent (book buyer agent) was created and initialized with user's task (book title, deadline, maximum price) that encapsulates the behavior of the agent, while, the other two platforms hosted the bookshop service providers (on which book seller agents are registered to serve the book buyer agent request).

The evaluation metrics used in this work include memory usage and runtime of the verification mechanism in both security frameworks.

Table 4.1: Memory usage of the existing framework

Host	No of roaming agent	No of migrating agent	Memory (KB)
A	5	2	5.856
B	10	5	19.712
C	20	10	48.650
D	30	15	78.840

The table 4.1 shows the results obtained from existing security framework. This clearly demonstrates that the memory consumption of the RM agent on the host machines increases linearly as the number migrating agents. Also table 4.2 shows that the execution time for the verification process rises proportionately to the number of agent on host machines.

Table 4.2: Verification runtime of the existing framework

Host	No of roaming agent	No of migrating agent	Runtime (sec)
A	5	2	2.262
B	10	5	9.31
C	20	10	20.62
D	30	15	29.93

The results elicited from the proposed security framework is displayed in table 4.3 and table 4.4 and show the memory usage and verification runtime respectively.

Table 4.3 Memory Usage of proposed framework

Host	No of roaming agent	No of migrating agent	Runtime (sec)
A	5	2	3.586
B	10	5	15.512
C	20	10	35.615
D	30	15	70.740

Table 4.4:Verification runtime of the proposed framework

Host	No of roaming agent	No of migrating agent	Runtime (sec)
A	5	2	1.051
B	10	5	7.41
C	20	10	16.72
D	30	15	24.13

Comparing the values got from the evaluation process, it is quite clear that the resource consumption of the proposed system reduced substantially across the agent platforms. The figure 4.1 and figure 4.2 give a graphical representation of the comparison.

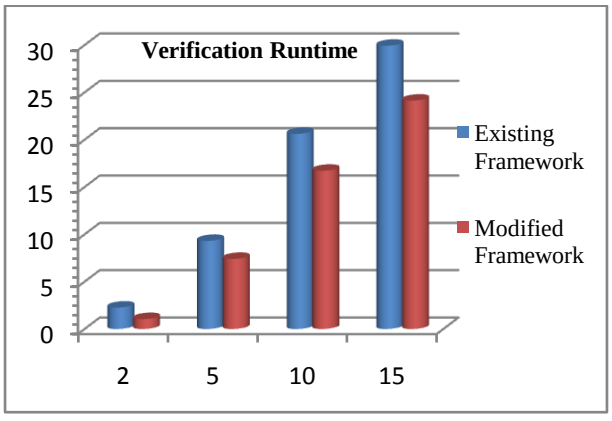

Fig 4.1: Verification runtime comparison

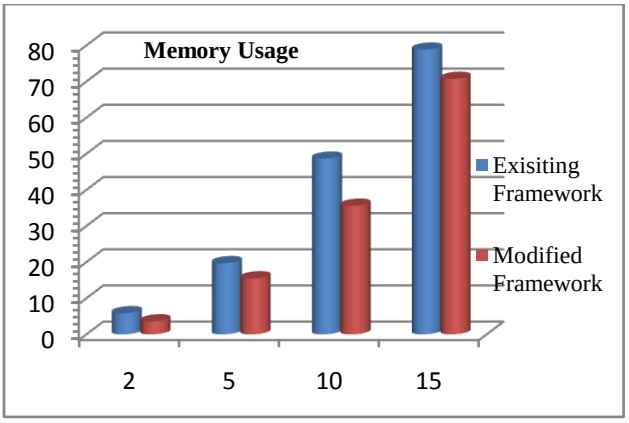

Fig 4.2: Memory usage comparison

The following computations summary the results for Memory Usage (MU) and Verification Runtime (VRT) comparison.

MU = ((5.586 - 3.586)/5.586) *100 = **35.80%**

VRT = ((2.262 – 1.051)/2.262) * 100 = **53.54%**

From the results above, it becomes evident that with this approach, we have been able to reduce the resource consumption of the reference monitor based security framework significantly, that is to say, the memory usage is decreased to **35.80%** and the time to complete the verification mechanism downgraded to **53.54%**

5. CONCLUSION

The main idea in this work is that we introduced a Trusted Third Party, as a fundamental component of the enhanced framework. By this way, the Trusted Third Party is allowed to implement the verification mechanism of reference monitor concept, thus enabling the RM agent to duly performs the reference validation mechanism through the enforcement of the system's security policy. The Trusted Third Party fulfils the requirements of an efficient computing environment to mobile agents. Altogether, with the enhanced security framework, it is still possible for a mobile agent to migrate to any agent platform (AP) with the assurance of security. The enhanced security framework is most suitable for open systems where protection or security against malicious hosts and support for mobile agent computing with static as well as dynamic itineraries are required.

The future scope of this idea revolves around integrating this mechanism into multi-agent systems where remote hosts are unknown and untrusted.

5.1 Further Work

Future work can focus on how to integrate the RM agent into the user's roaming agents and to further dynamically equip the RM agents to provide better security.

6. ACKNOWLEDGMENTS

We wish to express our profound gratitude to all the anonymous reviewers for their valuable and detailed comments.

7. REFERENCES

[1] K. Neeran 1998 "Security in Mobile Agent Systems, Ph. D. dissertation," Deparment of Computer Science and Engineering,, University of Minnesota.

[2] Lange, D. B. and Oshima, M. 1999 "Seven GoodReasons for Mobile Agents," Communication of the ACM, vol. 29, no. 3, pp. 88-89.

[3] H. Abdellatif 2014 "Self-adaptive security for mobile agents," International Journal of Computer Applications, pp. 50-65.

[4] R. Wang, Hu T. and Xu X. 2004 "Research in Mobile agent security," Journal of Chongqing University of Posts and Telecommunications, vol. 16, no. 3, pp. 81-86.

[5] M. P. a. M. R. Lukasz N. 2006 "Mobile agent security," 1 Thomas edition, Information assurance and computer security, IOS press, pp. 102-123.

[6] Uwe G. W., Sebastian S. and Levente B. 1, 1999 "Introducing Trusted Third Parties to the Mobile Agent Paradigm," Secure Internet Programming, vol. 1603, pp. 471-49.

[7] Sandya. A., N. M. and N. N. 2015 "Mobile Agent Security using Reference Monitor-based Security Framework," in In Proceedings of the Ninth International Conference on Emerging Security Information Systems and Technlogies.

[8] F. Hohl. 1998 "Time Limited Blackbox Security: Protecting Mobile Agents from Malicious Hosts," Springer Verlag,.

[9] Esparza .O, S. M., L. Munoz J. and J. Forne 2003 "A protocol for detecting maliciouss hosts based on limiting the execution time of mobile agents," in In Proceedings of IEEE Eight International Symposium on Computers and Communciation (ISCC'03), Kerner-Antalya, Turkey.

[10] A. M. Ngereki. and A. M. Kahonge. 2015 "A Multi-Faceted Approach to Mobile Agent Security," International Journal of Computer Applications, vol. 120, no. 21, pp. 20-25.

[11] Sandya A., & A., Caully 2011 "Obfuscation Techniques for Mobile Agent code confidentiality," Journal of information & Systems Management, vol. 1, no. 1, pp. 25-36.

[12] P. J. Anderson 1972 "Computer Security Technology Planning study, Bedford MA: Technial Report ESD-TR-72-51," Air Force electronice Division, Hanscom AFB.

[13] Ibharalu FT., A. B. Sofoluwe and T. A. A. , 2011 "A reliable protectioin architectur for mobile agents in open network system," International Journal of Computer applications, vol. 17, no. 7, pp. 6-14.

[14] P. Nisha, K. Sunil and A. B. 2010 "Security on Mobile Agent BAsed Crawler," International Journal of computer applications, vol. 17, no. 15, pp. 5-11.

[15] A. Shibli, Y. I. and M. S. 2010 "MagicNET: Security system for protection of mobile agent," in In Proceedings of the IEEE International Conference on Advanced Information Networking and Applications.

[16] S. Leriche and Arcangeli J. , 2010 "Flexible architectures of adaptive agents: the agents approach," International Journal of grid computing and multi agent systems, vol. 1, no. 1, pp. 55-75.