International Journal of Intelligent Computing and Information Science

SECURE MOBILE AGENT TRADER SYSTEM

S. M. Koriem R. M. Hassaan

Computers & Systems Eng. Department, Faculty of Engineering,
Al-Azhar University, Cairo, Egypt

samirkoriem@yahoo.com rana_hassaan@live.com

Abstract: *Mobile agent systems (MASs) have lately been used in all kinds of fields. MASs are programs that travel autonomously through a computer network in order to perform some computation or gather information on behalf of a human user or an application. MASs can be very useful in implementing E-business applications. In most applications, the security of mobile agents is of the utmost importance. In this paper, we propose a MAS model for implementing a secure trading system in JADE environment.Public key infrastructure is used in our proposed model by all parties to achieve two-way authentication. Policy-based security management has become a growing research area for mobile agent security. To ensure authorization inour proposed model a role-based access control (RBAC) is used to grant privileges to agents according to their roles. X.509 certificate is the most widely used data format for public key certificates. In our proposed model, X.509 certificateisused to store role information in its OU (organization unit) field, to make sure that only authorized agents in a scalable environment are given access to certain data or resources according to their roles. Authorized agents may be tampered by malicious agent to misuse its privileges. To ensure agent's integrity, agent code is encrypted, signedand added in X.509 certificate to be checked before agent execution. To ensure confidentiality, information that needs to be protected from disclosure like agent's shopping list is encrypted while travelling through communication channels to prevent malicious agents from eavesdropping.*

Index-Terms: *Mobile agent, security, role-based access control, X.509 certificate, public key infrastructure.*

1. Introduction

Mobile agent has embraced a bright prospect in E-commerce field, due to its great adaptability and advantages. Mobile agent has been widely applied in E-commerce and distributed computing. However the security issue related to mobile agent is still a problem remaining to be solved. Taking into consideration that E-commerce has been growing rapidly, ensuring security of mobile agent has become a vital constraint towards further development of E-commerce using mobile agent technology [1].

Mobile agent technology offers the possibility of executing a large number of tasks, which must be performed to manipulate large amount of data or resources, with minimal human intervention [2].Mobile agents have several advantages in the development of various services in smart environments in addition to distributed applications. Reduced communication costs is a great advantage of mobile agents, since mobile agent technology enables remote communications to operate as local

communications which is very useful for distributed applications. Another advantage of mobile agents is the asynchronous execution, where a mobile agent is able to continue processing at its destination even when the agent owner computer is shutdown or the network between source and destination is disconnected [3].

The internet has revolutionized the way business is performed. Internet has fundamentally changed the interaction between a business organization and its customers. Recently, multi-agents systems have been recognized as a very promising technology to develop next generation E-commerce systems by addressing many of the issues prevailing today. It is believed that software agents with decision autonomy and location autonomy (mobility) will make a paradigm shift in the evolution of E-commerce systems [4].

To illustrate the proposed work, we describe the development of mobile agent technology due to its advantages. Also, we describe the revolution of E-commerce and the security requirements of mobile agent due to this revolution. In Section 2, we discuss mobile agent applications related to our work. In Section 3, we describe the paper research methodology. In Section 4, we propose a trader system architecture to be used as an E-commerce system. In Section 5, we propose a security model to be implemented with system proposed in section 4 to achieve a secure trader system architecture. In Section 6, we study the impact on performance when security model proposed in Section 5 is complemented with trader system proposed in Section 4. Finally, in Section 7, we summarize our findings, discuss their implications and conclude.

1.1. Mobile Agents

After the idea of mobile software agents had spread in the middle of the 1990s, research on mobile agentsconcluded that MAS would be one of the major design principles for future distributed systems [5].A mobile Agent is a computer program that runs within a certain environment and can transport itself from one system to another [6].Mobile agent uses the network infrastructure to run in remote sites to gather information, cooperate with other sites and return to the home site after completing the assigned tasks [7].

Mobile agents are autonomous software agents that travel in a computer network to execute and perform tasks on different hosts on behalf of their owners. Autonomous mobile agents bring advantages such as task delegation, network communication, and cost reduction for distributed tasks [8].In addition to the inherited capabilities of stationary agents, mobile agents represent a class of agents whose main functions are their transmission capabilities between nodes on the same network or different networks. Mobile agents represent a direct extension of the client-server approach. The agent approach allows agents to continue to run after leaving a node, even if they lose connection with the node where they were created. Moreover, agent approach reduces traffic in the network thereby increasing the communication speed. Besides, an agent can move on to other machines when necessary and can delegate tasks to other mobile agents in order to achieve parallel distributed application. Finally, MAS is reliable due to mobility and autonomy which allows the agent to move from one point to another in the network and provide services and meet predefined goals without intervention [7].

For a number of years now, researchers promise that the agent technology is about to change the ways we construct software as well as have a much broader impact on the field of human-computer interaction. The characteristics of mobile agents make them ideal for E-commerce applications in open networks. A mobile agent can search for special products or services and negotiate on behalf of its owner with other entities. Furthermore, mobile agents can be used as selling agents. Unfortunately,

promises didn't materialize. To the contrary, it is relatively difficult to point to a successful large-scale implementation of agent systems . Implementation of MAS in real systems is obviously a matter of scalability. Also, security affected the implementation of mobile agents in real systems.Mobile agents are vulnerable to several attacks.Without achieving the proper security countermeasures, the use of agent-based applications will be severely impeded [9] [10].

We are interested in securing the agent platform in E-commercesystem. We have selected one of the best currently existing agent platforms, JADE version 4.3 to simulate our system model [11].

1.2. Security Threats and Requirements

Mobile agents can provide many benefits to the development of distributed applications, but their use also poses many security threats. Research on mobile agent technology has identified and solved a number of security-related issues but there are still many remaining unsolved [12].As the sophistication of mobile software increases, the associated security threats and vulnerabilities also increases. Threats to the security of mobile agents generally fall into three comprehensive classes; (1) disclosure of information, (2) denial of service, and (3) corruption of information[9].

Components of an agent system are used to categorize those classes of threats in greater detail, to identify the possible source and target of an attack. The threats that are discussed have counterparts in conventional client-server systems and have always existed in some form in the past. Mobile agents simply offer a greater opportunity for abuse and misuse, which means that it broadens the scale of threats significantly [9].Mechanisms for mobile agent security need to be tied to the threats and requirements that exist in a mobile agent environment. Developing good requirements for mobile agent security and matching those with existing security mechanisms will be important for success of mobile architectures long term [13].

Four major categorizations of attacks in a mobile agent environment include the following : (1) attacks by malicious agents against hosts; (2) attacks by malicious hosts against agents; (3) attacks by agents against other agents; and (4) attacks by other entities against the host platform [13].The majority of mobile agent security research is split into two broad categories: defending against malicious code threats and defending against malicious hosts threats. The problem of host security was the focus of early attention in the field because malicious code draws many parallels to migrating viruses. The malicious host problem is all about how to compute securely in an untrusted environment and protect any data that should not be divulged to the host that does the computation [13].

Confidentiality is breached when important and private information in a mobile agent is disclosed to unauthorized user and it is an obligation to protect mobile agents private information. The integrity of mobile agents should be protected by ensuring that the mobile agent migrating from one host to another is not intercepted, altered or modified in the migration process. Authenticity is the process of verifying mobile agents identities. Mobile agents are normally required to pass through authentication in order to have access to specific resources [14]. Therefore, confidentiality, integrity, and authenticity are the three commonly used categories to define security requirements for computer systems [15].

1.2.1. Malicious Agents

Malicious agent can attempt to either gain unauthorized access to host resources or wrongly use the authorizations that have been granted by the host [16].Denial of service attacks and eavesdropping can be executed by malicious agent. Changes to agent state or code can cause an agent to become malicious

in nature itself [13]. Therefore, mobile agents owners must be authenticated to get authorized to communicate together, migrate, and get access to resources.As a result, three essential needs concerning the host platform when dealing with mobile agents are required: (1) authentication; (2) authorization; and (3) allocation of resources. [17].

1.2.2.Malicious Hosts

The execution environment of the agent is under the complete control of a malicious host without other means of protection. A desired property of any host framework would be the ability to execute code on behalf of a user without gaining any knowledge of what that code is accomplishing [18].Inspection, modification, denial of service, replay and masquerading can be executed by malicious host. Therefore, the agent code, itinerary and data state must be protected from those various threats. Information protection can be assured when seller agent is authenticated and authorized to limit the access to the buyer agent private information [13].

2. Related work

2.1Mobile Agent Applications

JADE platform has been used in implementing a lot of systems. An Integrated agent system for e-mail coordination was implemented, where a method is proposed to prioritize the unread e-mails according to the users' interest and priority. The Agent mechanism is guided using JADE Middleware and its underlying architecture [19].

Also a mobile agent based distributed information platform for autonomous health care monitoring was implemented. Distance medical advice and continuous monitoring of medical conditions for critical needs of patients are offered by this platform. In health care field, agent technology has been applied to improve the performance of information systems in terms of interoperability, scalability and re-configurability. The platform offers a diagnostic tool time critical situations, enabling them to make vital decisions faster. Fast decision making is due to immediate communication and exchange of real time data [20].Another health care application was presented, that is composed of agents that provide medical services. The system contains agents developed using JADE.The health care application allows the user to search for medical centers satisfying a given set of symptoms. Also, user is allowed to access his medical record or to make an appointment to be visited by a particular kind of doctor in his location. User can also chat with or mail a doctor, and can get various tips about health care and diseases. User agent is used by patient to enter symptoms and gets the list of available doctors in clinic and hospital. Moreover, user agent can chat or mail doctors and view his health profile. On the other hand, doctor agent can update medical records and prescribe medicine. All these tasks are executed with the help of the main agent that searches doctors directory and fixes appointments. [21].

A conceptual architecture of a multi-agent E-commerce system was proposed in [22].Another approach for E-commerce systems was presented, where development of automatic negotiations is one of the more important research issues in which agents change their negotiation protocol and strategy through dynamic loading of negotiation modules [23].Another implementation of E-commerce was proposed,where a mobile agent is designed to search and to filter information of interest from electronic markets. Also security techniques and its robustness were considered. A sound security of information gathered is ensured throughout agent's itinerary against various security attacks, as well as truncation attacks [24].

The design and implementation of a multi-agent system capable of executing some of the most important auctioning and voting protocols was also presented. The experiments prove the possibility of using such a system on a large scale, with the benefits of reduced costs and flexibility to participate in auctions or elections organized in different settings [25].

2.2. Host Protection

Mechanisms used to prevent malicious agent behavior can lead to unnecessarily restricting mobile code with good intentions while failing to discern and restrict code that has hostile purposes [12].The overall goal of host protection is to somehow limit the overall power of the execution environment while reducing the overall vulnerability of a host to a malicious mobile agent [13].

Many mechanisms like sandboxing, safe code interpretation, digital code signature, and policy management were used for host protection. Sandboxing mechanism was proposed as means to provide a separate but protected place for unsafe code to execute in, when it migrates to a remote host. The disadvantage of this proposal is that sandboxing can limit the usefulness of applications by restricting mobile code with good intentions [26].

Safe code interpretation is another provided early method that offers security feature for remote code execution than compiled environments because instructions can be examined for their effects before execution. Sandboxing and safe interpretation do not help establish trustworthiness of a given mobile agent program [27].

Mobile agent code digital signature has been used to verify the identity of the mobile agent. A verified signature does not guarantee the mobile code to be trusted. The trust model is all or nothing in the sense that code is allowed to execute with some set of privileges once signature verification is done. Policy statements can establish how to interpret valid code signatures and at a minimum trust between an agent originator and the remote host [28].

Another method of host protection is policy management. Before allocating host resources to an agent, mobile agents identity must be authenticated and their authorization level determined. Policy-based security management has become a growing research area for mobile agent security. Benefits of policy frameworks when used to implement security are reusability, extendibility, efficient and verifiable [29].

2.3. Agent Protection

Defending agents against malicious host attacks is the second major category for considering security mechanisms. Some mechanisms focus on code protection, others are more geared towards data state protection, and even others focus on itinerary protection. Some mechanisms are preventative while others detect malicious activity. Some schemes require more proprietary changes to the agent framework than others, some mechanisms are more expensive in cost and performance, and some do not offer strong cryptographic levels of security. In all, each must be evaluated according to the security requirements that a mobile agent application needs in the face of possible malicious host activities [13].

Execution tracing is a cryptographic tracing mechanism that is used to detect illegal changes to the data or code. Tan and Moreau introduce a trusted third party that serves the role of verification authority for traces generated by an agent server. Traces only indicate if a given result is a possible execution of the program and not necessarily the actual execution of the program. Trusted third parties decreases the openness of this solution [30].

here are some information regarded public and some information regarded private, which requires protection. All types of data require integrity verification. Digital signature is a method of protection that can offers integrity protection. In this approach, each host signs its data result after encryption takes place (normally with the public key of the agent owner). A trusted node is responsible for verifying that each host in itinerary did not tamper data results.

Policy management as mentioned previously in Section 2.2.is a method used to protect from both malicious host and malicious agent. Policies play a key role in defining mobile agent security architecture[23].

2.4.Security Mechanisms

Some security mechanisms by their nature provide protection for both agents and hosts, mainly due to the fact that a malicious host can subvert an agent into a malicious variant – thus requiring host and agent protection equally.A large category of protection approaches are geared toward determining whether an agent state has been illegally modified or not. This property is referred to as integrity of execution and seeks to verify the agent is in a state that is consistent with a normal execution given the set of input from each host in the agent itinerary [13].

2.4.1.Policy Management

Before agents can be allocated resources on a local host, their identity must be authenticated and their authorization level determined. Policy-based security management has become a growing research area for mobile agent security as a spillover from work done in network security management [30].
Policy frameworks ultimately protect both agent and host because they concern themselves with expression and development of dynamic trust assessment in mobile contexts [13].There are several benefits of policy frameworks when used to implement security [31]:

- Reusable
- Extendable
- Verifiable
- Efficient
- Context sensitive

A privilegemanagement scheme for mobile agent systems was described with an attribute certificate that conveys the policy rules associated with an agent, and a policy certificate that conveys policy governing the behavior of all agents that may attempt to visit an agent platform or a specific place on an agent platform [33].

Prescribed security policies for both the agent and the hostshould be embodied externally in separate certificates. The agent's use of resources is governed by an attribute certificate while the governing rules for all agents visiting a host platform are embodied within a policy certificate. Attribute certificate is an external object used to carry the policy information. Attribute certificate include the identity of the owner, the identity of the issuer, algorithm used to protect the certificate and the subject attributes. Policy certificates and attribute certificates are nearly synonymous except policy certificates represent more than just a host platform and in some cases can be created and maintained offsite from the agent platform itself. Policy certificates express policy rules assigned to an agent platform instead of an agent.The proposed framework allows an agent to carry one or more attribute certificates to host in their itinerary, all of which determine the relevancy of a given certificate after verifying an issuer's

identification. Agents are granted privileges based on whether attribute certificates of the agent comply with policy certificates of the host [32].

Policies should help agents represent their security levels while also giving visibility to the underlying host resources that are accessible to them.Framework policies are another proprietary method for security that require infrastructure in both agents and frameworks to implement. Despite this drawback, they are considered by some the only hope of an extensible generalized means for combined agent and host defense.

2.4.2. Code Signature

Code signatures were introduced originally with the "Microsoft ActiveX" framework and are part and parcel of the Java environment in the form of signed applets [33]. Signatures can be used to verify the integrity of the code when used in conjunction with cryptographic functions. Hash of the mobile code can be generated and used as a message digest to be sent along with the mobile code. After reception of the mobile agent, the host runs the same hash function. If the result equals the message digest that came with the agent, then integrity is verified. Signatures are based on public key cryptography where a public/private key pair is associated with particular principle. Signatures can also be used to verify the integrity of the code when used in conjunction with cryptographic functions. A verified code signature does not guarantee that the code is trusted. Policy statements can be used to establish minimum trust level with a limited set of privileges [13].

3. Research Methodology

We aim to do in depth literature review of existing work to identify mobile agents security areas covered. Based on literature survey, we have identified makeable solutions to security problem, and therefore propose a design and prototype implementation for solutions. The first main objective of the proposed research work is to achieve confidentiality, where a mobile agent private information must be protected. The second main objective is integrity, where we must ensure that the mobile agent is neither altered nor modified during migration.The last main objective isauthenticity, where the mobile agent identity must be verified to get authorities.

3.1 JADE Background

"JADE" (Java Agent Development Framework) is a software development framework fully implemented in Java language. JADE is a middleware developed by the research and development department of "Telecom Italia" (Telecom Italia Lab – AKATILAB), that simplifies the development of applications. It simplifies the implementation of multi-agent systems through a middle-ware that complies with FIPA (Foundation for Intelligent Physical Agents) specifications. FIPA specifications represent a collection of standards which are intended to promote the interoperation of heterogeneous agents and the services that the can represent. FIPA specification can be categorized into agent communication, agent transport, agent management, abstract architecture and applications. Therefore, the goal of JADE is to simplify the development while ensuring standard compliance through a comprehensive set of system services and agents. Agent model is used by JADE to allow agent mobility, high runtime efficiency, software reuse, and the realization of different agent architectures. FIPA-compliant agent platform includes Agent Management System (AMS), Directory Facilitator(DF) and Message Transport System also known as Agent Communication Channel (ACC). All these three agents are automatically activated at the agent platform start-up. An agent platform can have multiple

agents for scalable application multiple platforms should be used to reduce the load on AMS, DF and ACC[34].

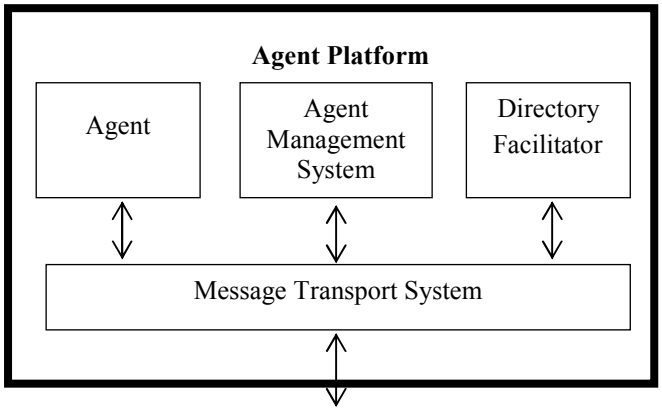

Figure 1: Architecture of a FIPA Agent Platform

JADE includes two main products: a FIPA-compliant agent platform and a package to develop Java agents. JADE has been fully coded in Java.The standard model of an agent platform, as defined by FIPA, is represented in Figure 1 [34].

AMS is the agent who exerts supervisory control overaccess to and use of the Agent Platform. Each platform will have only one AMS. AMSprovides white-page and life-cycle service. AMS also maintains a directory of agent identifiers (AID) andagent state. Agents must register with AMS on creation in order to get a valid AID.DF is the agent who provides the default yellow page service intheplatform.The Message Transport System, is thesoftware component controlling all the exchange of messages within the platform, includingincoming and outgoing messages of remote platforms [34].

JADE fully complies with FIPA reference. AMS and DF are created after launching JADE platform immediately. Also, the Messaging Service (implementing the ACC component) is always activated to allow message-based communication. Agent platform can be split on several hosts, as represented in Figure 2. Typically (but not necessarily) only one Java application, i.e. only one Java Virtual Machine (JVM), is executed on each host. Each JVM is a basic container of agents that provides a complete run time environment for agent execution and allows several agents to concurrently execute on the same host. The container where the AMS and DF lives is called the main-container. Other containers connect to the main container and provide a complete run-time environment for the execution of any set of JADE agents [34].

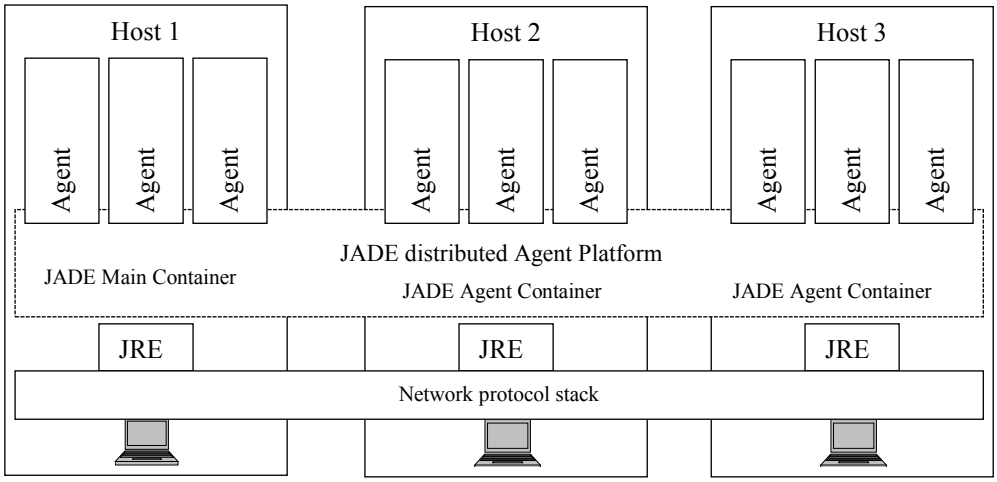

Figure 2: JADE Agent Platform distributed over several containers

4. Trader System Architecture

The commercial activity is a significant part of the network infrastructure allowing an open market of the services. A trading mobile agent system is proposed as a system that can have multiple sellers and multiple buyers. It helps buyers to find seller of the required service. A trading mobile agent system is represented in Figure 3.

Each buyer creates a buyer agent to execute specific tasks to complete the purchasing process. Also, each seller creates a seller agent to be registered in the DF to sell the sellers products. When the buyer enters its shopping, a buyer agent is created to request agents that supports all required services fromDF. The DF returns a list of agents supporting those services. The seller agents list returned by DF is considered as the itinerary of the buyer agent. And the buyer agent moves to each seller in this itinerary to buy all items in the list.

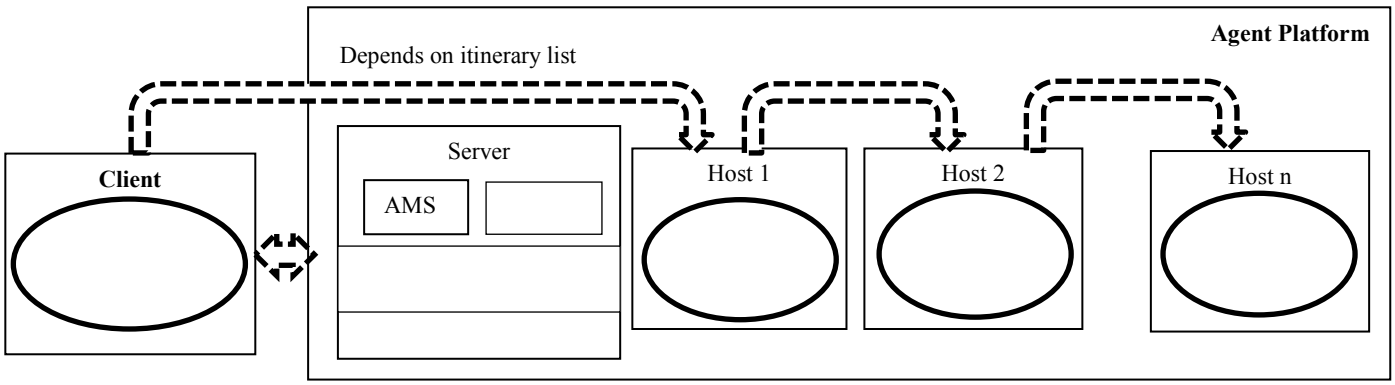

Figure 3: Trading System before applying security

5. Secure Trader System Architecture

In this paper, Secure Trader System (STS) architecture is proposed. STS provides all infrastructural and functional components needed for a secure MAS. A MAS is said to be applicable and practical when it hinges on realistic security techniques, especially for E-commerce systems. STS security model proposed is represented in Figure 4. On mobile agent creation, mobile agent requests ID certificate from

AMS with its ID , public key and requested role. Consequently, AMS assigns a role to the mobile agent, signs the agent code and requests ID certificate from CA for this agent with role and code signature assigned as certificate attributes.

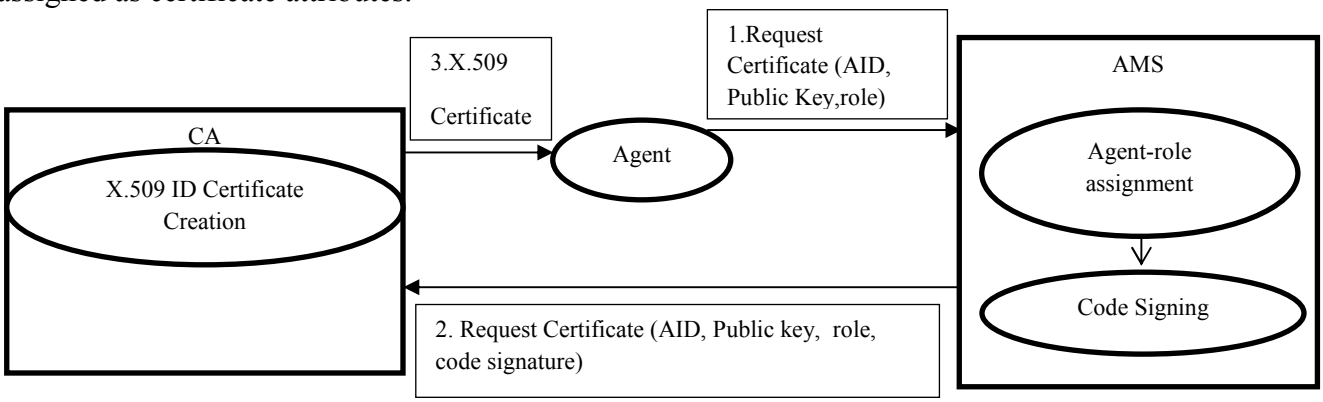

Figure 4: Security Model

5.1. Phases of STS

We have conceptually structured the use of the overall system into four functional phases:

a)Initialization and Deployment Phase –On buyer/seller registration buyer/seller agents are being created, authenticated and policies are granted according to role.

b) Itinerary Retrieval Phase–Buyer agent retrieves seller agents according to service they are registered with in the DF;

c) Verify Certificate Phase–Checks the validity of the seller agent certificate; and

*d) Migration and Execution Phase.*Code signature checked and executes its specific task, after agents migrate to the trusted host.

Mobile agents' development starts in the first phase i.e. *"initialization and deployment phase"*. The outputs of the first phase serve as inputs for the next phase, and so on. Each phase provides input to the subsequent phase. Each buyer/seller agent should have a certificate created by the authenticated buyer/seller, where the agent's signature and role is attached to it where each buyer has only one buyer agent and each seller has only one seller agent.

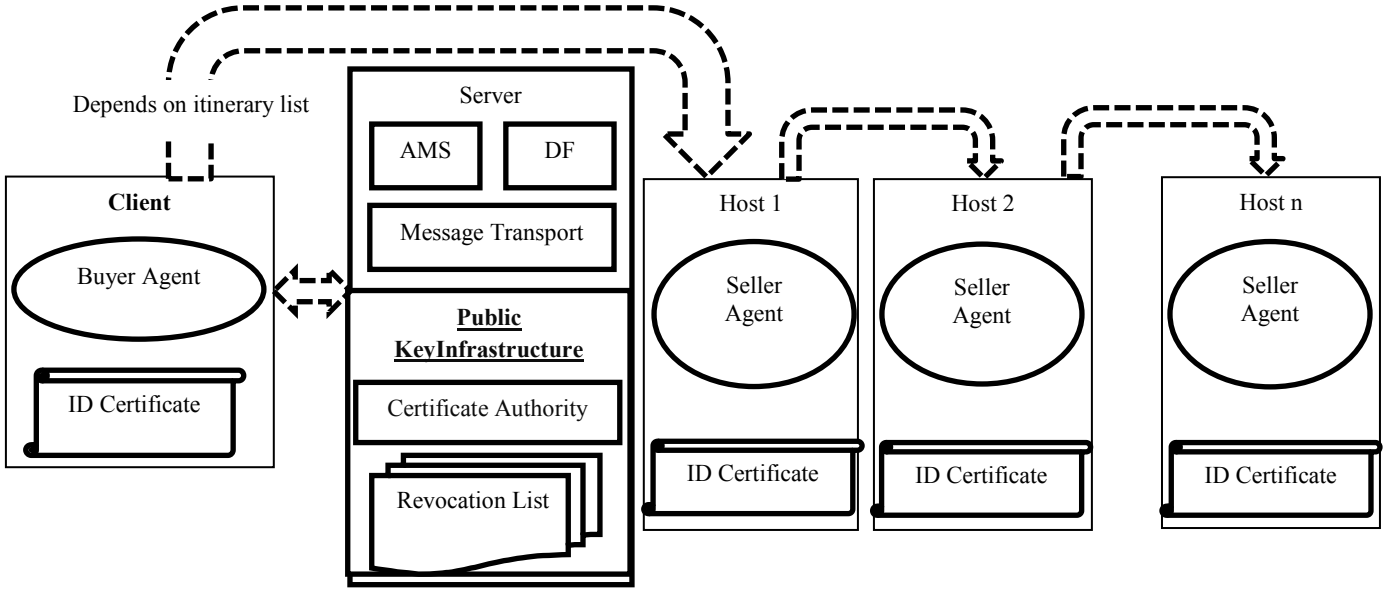

Figure 5: Trading System after applying security

In STS infrastructure we have proposed comprehensive agent's development and execution life cycle, which comprises ninestages.Those stages are spread over four phases mentioned above. Each phase includes different stages and thus solves specific security requirement of a particular phase. Agent creation, agent privileges assignment according to role and agent code encryption and signing belongs to initialization and deployment phase.The next phase is the itinerary retrieval, where buyer agent owner authentication and retrieving list of seller agents are the two stages of this phase.Seller agent certificate verification is the next stage in certificate verification phase.Finally, the last phase contains buyer agent migration, validating buyer agent code signature, and buyer agent execution stages. Trading system is shown in Figure 5 after applying the proposed security model. The next section explains different security requirements and motivation of security issues related to mobile agents' security infrastructure in each specific phase. Those requirements are then addressed in subsequently, where security procedures corresponding to each phase have been discussed in detail.

- **First Phase: Mobile Agents Initialization and Deployment**

a) Buyer Initialization and Deployment

When a client needs to buy anything, it creates a buyer agent with the required shopping list, and the buyer agent registers itself at the AMS to get an AID. Buyer agent requests a certificate from AMS with a buyer role and encrypted agent code to be signed. AMS signs the encrypted agent code and requests X.509 ID certificate for buyer agent from CA.X.509 IDcertificateisassigned to the buyer agent containing buyer role information,as well as the signature of the encryptedagentcode, both signed with the public key of thecertificate authority (CA). Privileges are granted according to the agent's role. Buyer shopping list is encrypted to avoid information disclosure. Buyer initialization and deployment phase protocol is shown in Figure 6.

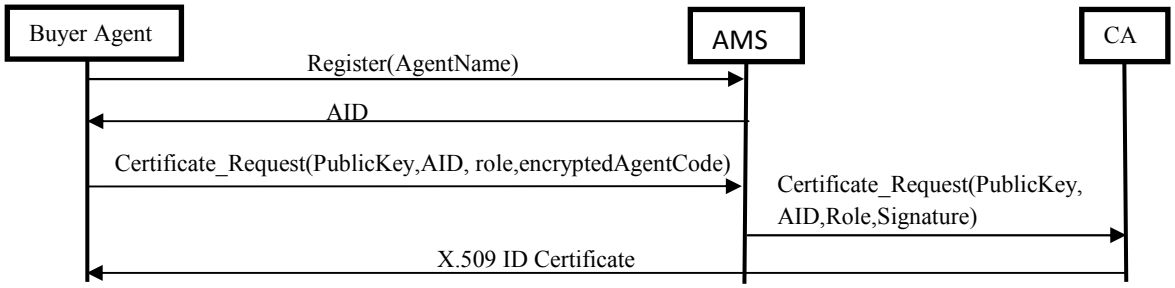

Figure 6: Buyer Initialization and Deployment Phase

b) Seller Initialization &Deployment

Each host creates its seller agent, and the seller agent registers itself at the AMS to get an AID. Seller agent requests a certificate from AMS with a seller role and encrypted agent code to be signed. AMS signs the encrypted agent code and requests X.509 ID certificate for seller agent from CA.X.509 certificate with seller role, and seller code signature is assigned to the seller agent, both signed with the CA public key. After that the agent registers the supported service at the DF. Seller initialization and deployment phase protocol is shown in Figure 7.

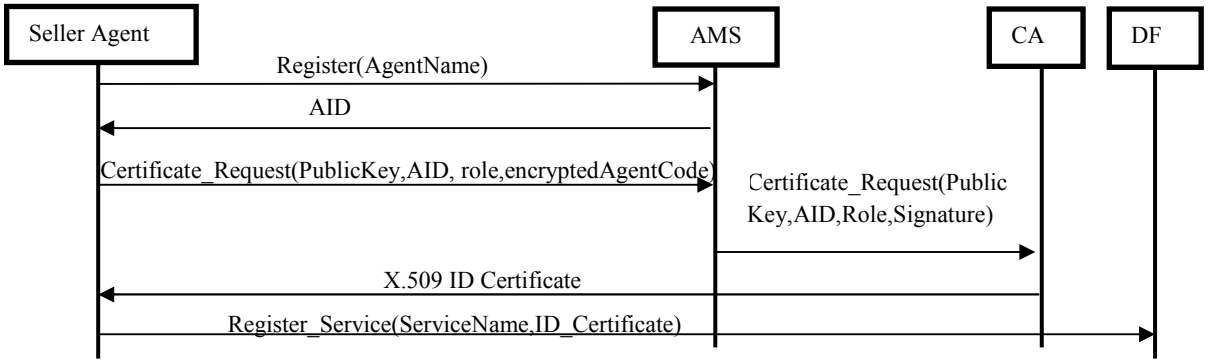

Figure 7: Seller Initialization and Deployment Phase

X.509 ID certificate is easy to add to public key infrastructure. X.509 ID certificate privileges can be managed easily. However, changing privileges require revocation of identity certificate. On the other hand, X.509 attribute certificate privileges can be managed easily and change in privileges doesn't require revocation of ID certificate. However, X.509 attribute certificate needs more cost to implement privileges management infrastructure. Using X.509 ID certificate and adding the role info to it will allow to have X.509 ID certificate that is easily added to public key infrastructure and also doesn't require revocation of ID certificate on changing privileges. That is due to the RBAC mechanism used, where the role is added to X.509 ID certificate without privileges. Therefore, the role can be maintained without modification, while modifying the privileges of this role.

- **Second Phase: Itinerary Retrieval**

The buyer agent submits the requested services of the shopping list to the DF. The DF replies with the agent's itinerary, a list of seller agents' AIDs and X.509 certificates for every requested service after it checks therevocation list to make sure that buyer agent certificate is still trusted, and checks buyer role permissions andbuyerencrypted code signature. Itinerary retrieval phase protocol is shown in Figure 8.

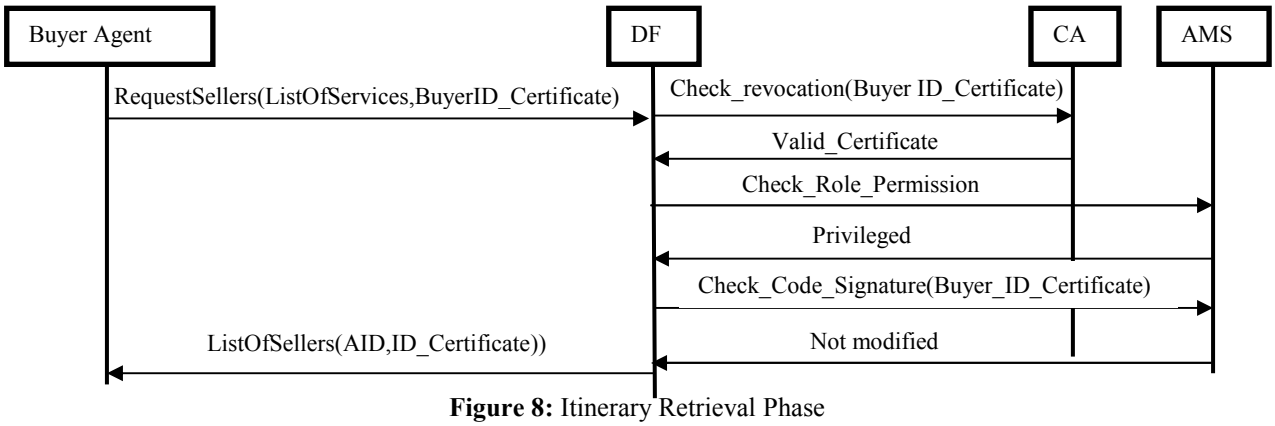

Figure 8: Itinerary Retrieval Phase

- **Third Phase:VerifyCertificate**

Buyer agent loops on the itinerary list to check revocation of the seller agents' certificates before migration and execution. Certificate verification phase is shown in Figure 9.

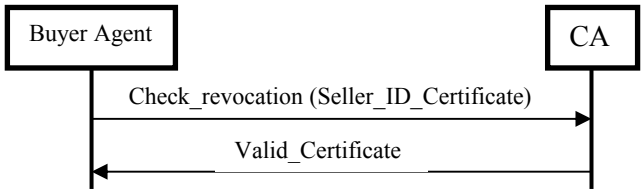

Figure 9: Verify Seller ID Certificate

- **Fourth Phase: Move & Execute**

Buyer agent migrates to seller agent and execute its task after buyer X.509 certificate trust, buyer role permission and buyer encrypted code signature are checkedfor validity. After buyer agent execution,the third and fourth phase are repeated to the next Seller ID certificate in the itinerary until the itinerary list is finished. Move and execute phase protocol is shown in Figure 10.

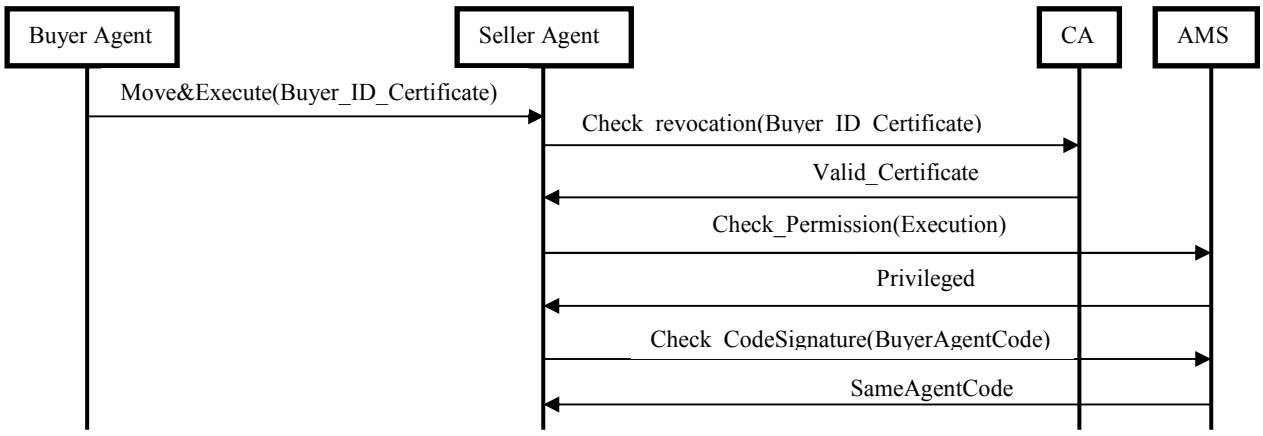

Figure 10: Move and Execute Phase

6. Performance Analysis

During development of STS, we found that an agent and host can use resources unrestrictedly**.** To avoid this threat we need to restrict in some way the resources usage. That was the reason of using RBAC. Also, we noticed that a malicious agent can eavesdrop the host's communication channel with other agents, and we need to secure important data traveling through those channels, so that important messages can't be cloned by malicious agents. Therefore, eavesdropping was the reason of encrypting important data (shopping list) while travelling through communication channels. Moreover, we faced a problem that due to multi-hopping, even non-malicious agents can become malicious at some point and abuse the privileges they are. And therefore, we realized that we need to monitor agent's integrity by receiving hosts, so that when host detects integrity problem, it marks this agent as untrusted and the agent becomes unauthorized to execute its task.Therefore, agent tampering was the reason of using agent code signature.

As a conclusion, authenticity, is achieved by implementing fine grained resource control based on security roles (RBAC – Role Based Access Control). Concerning confidentiality, it's achieved by encrypting important data before traveling through communication channels. Finally, having agent's code signature on creation and monitoring its change after each migration will ensure integrity. Mobile agent real application system performance is greatly influenced by security mechanisms. A mobile agent system performance evaluation is performed according to certain metrics to evaluate mobile agent system applicability. It's usually required to identify security mechanisms that doesn't affect the performance negatively. The factor considered from the performance perspective is the round trip time (RTT). The performance is studied with respect to the scalability.

In our research work, the performance metrics RTT is considered under the effect of changing the number of nodes. The RTT is the time required for a buyer agent to travel around from a seller agent to another to find the required shopping list, and purchasing list. To run STS, an assumption is made that the network bit rate is 256 kbps, 512 kbps & 1Mbps so that we can use agent size before migrationto simulate the time took to migrate the agent between hosts. Also, we can use communication messages size to simulate the time took sending messages between agents. Finally, number of nodes includes the total number of seller agents that are used for simulation. RTT is used to estimate the impact of increasing the number of nodes and applying security. The performance of STS can be studied by running STS and observing the RTT when increasing the number of nodes to evaluate the overhead of security applied on STS. A simulation software is developed to simulate STS client and hosts, to be run on one computer taking network latency of different assumed network bit rates into consideration during simulation process.

The performance study consists of five steps. In the first step the agents are implemented and the experiment is simulated with five nodes without applying security mechanisms. In every step we increase five nodes till we reach twenty five nodes. In all steps, RTTvalues are collected to be analyzed. The five steps are repeated again after applying security mechanisms, so that we can compare data collected before and after applying security mechanisms. Performance of mobile agent system is studied for five configurations before and after applying security mechanisms. These different setups are intended to study performance overhead after applying security. For each simulation run, the RTT increases.

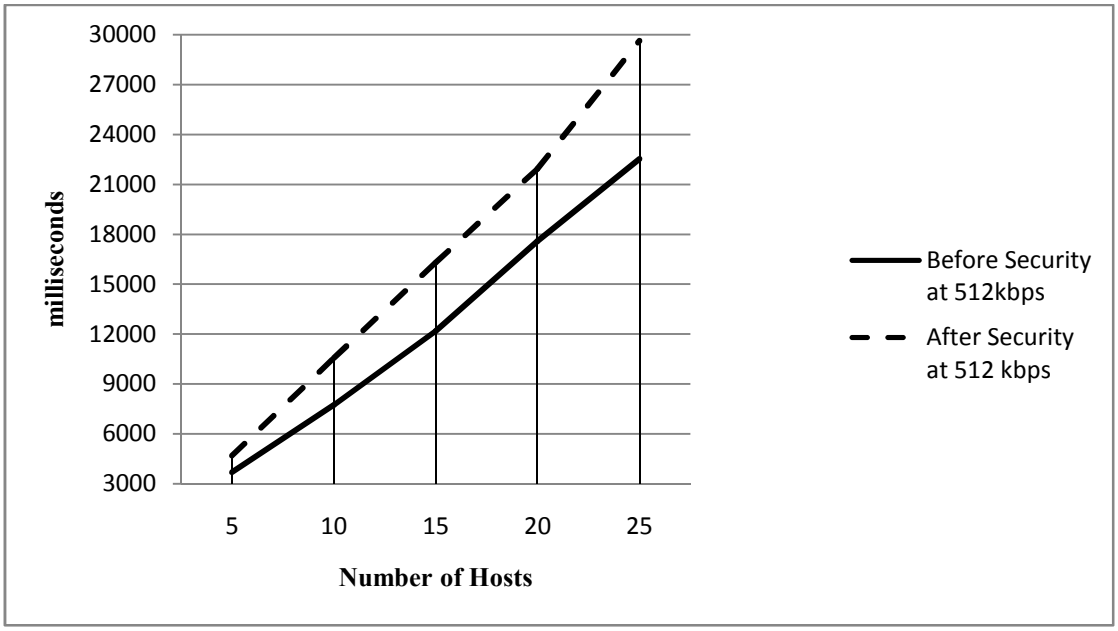

Figure 12: RTT

Observing five nodesconfigurationRTT behavior, it has been noticed to have minor difference before and after applying security. Observing ten nodes configuration, 512 kbps assumed bit rate we found that before applying security the RTT is 7736.91ms and increased to 10552.59ms after applying security. The difference between RTT value before and after applying security will continue to increase rapidly till we reach twenty five nodes configuration, 512 kbps assumed bit rate where the RTT is 22521.971ms before applying security and increased to 29618.323ms after applying security.

RTT increases by increasing the number of nodes after applying security to STS. RTT is affected by the longer life cycle of the agent. On initialization and deployment phase an overhead is introduced due to requesting certificate from CA, and requesting a role and agent code signature to be assigned to this certificate by the AMS. Moreover, any request from the agent is subjected to authentication check by validating the agent's certificate, authorization check by checking the agent's role and privileges, and integrity check by validating the agent's code current signature. Taking into consideration that the migration process is repeated number of time depending on the number of nodes in the itinerary and each migration requires authorization check and integrity check.

7. Conclusion

In this paper, we had performed an in depth review of existing work to identify mobile agents implementations and security areas covered. The design and implementation of secure agent platform for an E-commerce system were introduced. Our proposal in STS is to use public key infrastructure by all parties to achieve two-way authentication, where X.509 certificate will be used in implementation as it is a standard certificate and will not need any specific infrastructure at the buyers and sellers machines. We defined specific security roles like a buyer role and a seller role, and each role has been granted by limited privileges to regulate resource access to execute the required task without accessing forbidden resources. Each agent has its own role appended in its X.509 certificate to ensure authorizing the agent with privileges to achieve its task without abusing resources. On agent creation, an encrypted agent's code signature is appended to its X509 certificate. Encrypting confidential data like buyer agent

shopping list will secure those data during traveling through communication channels, preventing any malicious agent from eavesdropping those confidential information. Every time an agent travels to a host, the host checks modifications in the agent's code signature to ensure integrity before allowing the agent to execute its task. We have implemented a simulator forSTS using JADE framework to be run on one computer. All modules in the system were explained in detail. All phases in the agent life cycle were explained. We described the security mechanisms used inSTS. All parties in the proposed framework use public key infrastructure to achieve two-way *authentication*, which ensures that only authenticated agents can use STS. We used role-based policy management mechanism to grant privileges to users according to their roles, which ensures that each authenticated agent has a limited authority to use STS. In conjunction with "Role-Based Access Control", agent code is encrypted and signed to ensure *integrity*, to ensure that agent code isn't tampered and that this code belongs to an authenticated agent. User role and code signature are added in X.509 certificate.Furthermore, information that needs to be protected from disclosure is encrypted while travelling through communication channels to ensure confidentiality.

As expected, overhead in STS performance is faced due to time taken to create certificate for every agent, assigning a role to certificates, and signing and encrypting agents' code. Also, checking privileges of the agents after any request and signature appraisal after every migration will add extra time to the whole life cycle.

References

1. C. Zhang and X. M. Wei, "The study of the trust mechanism based on mobile agent in the E-commerce agricultural commodity trade," *Advance Materials Research,* vol. 1037, pp. 444-446, 2014.
2. R. J. Conejar and H.-k. Kim, "Designing a Mobile Multi-Agent Based for U-Healthcare," *Advanced Science & Technology Letters,* vol. 95, pp. 111-114, CIA2015.
3. I. Satoh, "Coordination of large-scale multi-agent systems," in *Mobile Agents*, Springer US, 2006, pp. 231-254.
4. A. C. Ojha, "A review of security issues in mobile agent-based E-commerce," *IUP Journal of Information Technology,* vol. 11(1), no. 53, 2015.
5. M. Feldmann, "An approach for using the web as a Mobile Agent infrastructure," in *International Multiconference on Computer Science and Information Technology*, 2007.
6. W. jihong and H. a. k. H. jianping, "Security Design of Mobile Agent System," in *Database and Expert Systems Applications, 11th International Workshop*, 2000.
7. A. Outtagarts, "Mobile Agent-based Applications: a Survey," *International Journal of Computer Science and Network Security,* vol. 9, no. 11, pp. 331-339, November 2009.
8. K. a. Iyengar, "A Model for Mobile Agent Security in E-Business Applications," *International Journal of Business and Information,* vol. 2, no. 2, 2007.
9. W. A. Jansen, "Countermeasures for Mobile Agent Security," *Computer Communications,* no. Special Issue on Advances in Research and Application of Network Security, 2000.
10. P. Kotzanikolaou, M. Burmester and V. Chrissikopoulos, "Secure Transactions with Mobile Agents in Hostile Environments," in *Information Security and Privacy*, Springer-Verlag, 2000, pp. 289-297.
11. K. Chmiel, D. Tomiak, M. Gawinecki, P. Karczmarek, M. Szymczak, M. Paprzycki, "Testing the Efficiency of JADE Agent Platform," in *Third International Symposium on Parallel and Distributed Computing/Third International Workshop on Algorithms*, 2004.

12. J. Zachary, "Protecting mobile code in the wild," *IEEE Internet Computing,* vol. 7, no. 2, pp. 78-82, 2003.
13. J. T. McDonald, A. Yasinsac and W. C. Thompson, "Taxonomy for Defining Mobile Agent Security," ACM Computing Surverys, May 2005.
14. S. A. Shanola and M. S. Joy, "Security framework for mobile learning environments," in *7th International Conference of Education, Research and Innovation*, Nov2014.
15. M. Bishop, Computer Security, Art and Science, Addison-Wesley, 2003.
16. C. Cubillos and F. Guidi-Polanco, "Security Issues on Agent-Based Technologies," in *VIP Scientific Forum of the International IPSI-2003 Conference*, 2003.
17. C.F.Tschudin, "Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet," in *Mobile Agent Security*, Springer-Verlag, 1999, pp. 431-445.
18. F. Hohl, "Time limited blackbox security: Protecting mobile agents from malicious hosts," in *Mobile Agents and Security*, Springer-Verlag, 1998, pp. 92-113.
19. B. S, "A JADE Implementation of Integrated Agent System," *International Journal of Computer Applications,* vol. Volume 58– No.5, November 2012.
20. C.-J. Su, "JADE implemented mobile multi-agent based, distributed information platform for pervasive health care monitoring," *Applied Soft Computing,* vol. 11, pp. 315-325, January 2011.
21. M. Gupta, A. Sarkar, I. Pramanik and B. Mukherjee, "Implementation Scheme for Online Medical Diagnosis Implementation Scheme for Online Medical Diagnosis Implementation Scheme for Online Medical Diagnosis Implementation Scheme for Online Medical Diagnosis Implementation Scheme for Online Medical Diagnosis," *International Journal of Scientific and Research Publications,* vol. 2, no. 6, June 2012.
22. A. Pîrvănescu, C. Bădică, M. Ganzha and M. Paprzycki, "Conceptual Architecture and Sample Implementation of a Multi-Agent E-commerce System," in *Proceedings of the 15 th International Conference on Control Systems and Computer Science CSCS'15*, Bucharest, Romania, 2005.
23. M. Ganzha, M. Paprzycki, A. Pirvanescu, C. Badica, C. Bădică and A. Abraham, "JADE-BASED MULTI-AGENT E-COMMERCE ENVIRONMENT; INITIAL IMPLEMENTATION," in *6th Int. Symposium SYNASC04*, Timisoara, Romania, 2001.
24. R. Al-Jaljouli and J. Abawaju, "Agents Bases E-commerce and Securing Exchanged Information," in *Pervasive Computing*, Springer London, 2009, pp. 383-404.
25. A.-I. Pandichi and F. Leon, "Design and Implementation of a Multiagent System for Auctioning and Voting," *Bulletin of the Polytechnic Institute of Iasi, Automatic Control and Computer Science Section,* pp. 87-101, 2011.
26. H. D. Johansen, E. Birrell and R. V. Renesse, "Enforcing Privacy Policies with Meta-Code," in *6th Asia-Pacific Workshop on Systems*, 2015.
27. J. O. Agushaka, H. A. Alaku and E. O. Adetula, "A framework for intrusion detection system using mobile agents with platform security," in *The Inter-Disciplinary Academic Conference on Uncommon Development*, Jan2015.
28. D. P. Sharma, "Mobile Agent-Based Authentication: A Model for User Authentication in a Distributed System," *International Journal of Computer Applications,* vol. 112, no. 13, 2015.
29. H. Idrissi, E. M. Souidi and A. Revel, "Secutrity of mobile agent platforms using access control and cryptography," in *Agent and Multi-Agent Systems: Technologies and Applications*, May2015, pp. 27-39.

30. H. Tan and L. Moreau, "Extending execution tracing for mobile code security," in *2nd Intl Workshop on Security of Mobile MultiAgent Systems (SEMAS2002)*, Bologna, Italy, July 2002.
31. S. Wright, R. Chadha and G. Lapiotis, "Special Issue on Policy Based Networking," *IEEE Network,* vol. 16, no. 2, 2002.
32. P. Bellavista, A. Corradi, C. Federici, R. Montanari and D. Tibaladi, "Security for Mobile Agents: Issues and Challenges," in *Handbook of Mobile Computing*, M. Ilyas, 2004.
34. W. Jansen, "A Privilege Management Scheme for Mobile Agent Systems," in *International Conference on Autonomous Agents*, Montreal, Canada, May 2001.
35. W. Jansen and T. Karygiannis, "NIST Special Publication 800-19 - Mobile Agent Security," National Institute of Standards and Technology, 2000.
36. TILAB, "JADE TILAB," 8 April 2010. [Online]. Available: http://jade.tilab.com/doc/programmersguide.pdf.
37. M. S. Greenberg, J. C. Byington and D. G. Harper, *IEEE Communications Magazine,* 1998.
38. D. Singlee and B. Preneel, "Secure E-commerce using mobile agents on untrusted hosts," Cosic Internal Report, May,2004.
39. B. S, "A JADE Implementation of Integrated Agent System for," *International Journal of Computer Applications,* vol. Volume 58– No.5, November 2012.