**American Journal of Applied Sciences 10 (12): 1638-1647, 2013**
ISSN: 1546-9239

doi:10.3844/ajassp.2013.1638.1647 Published Online 10 (12) 2013 (http://www.thescipub.com/ajas.toc)

# RESEARCH ON SECURITY PROTOCOL FOR COLLABORATING MOBILE AGENTS IN NETWORK INTRUSION DETECTION SYSTEMS

**Olumide Simeon Ogunnusi and Shukor Abd Razak**

Department of Computer Science, Faculty of Computing,
Universiti Teknologi Malaysia, 81310, Skudai, Johor Bharu, Malaysia

Received 2013-09-26, Revised 2013-10-04; Accepted 2013-11-07

## ABSTRACT

Despite the popularity of mobile agents in academic and commercial arena, the security issues associated with them have hindered their adoption on large scale distributed applications. However, researchers are making relentless effort to overcome the security impediments so that the interesting properties inherent in mobile agent application, especially in the field of intrusion detection, can be harnessed. Such properties include: adaptability, autonomous nature, low bandwidth utilization, latency eradication, mobility and intelligence. A number of protocols have been developed by researchers for different key distribution techniques to enhance their performance and to protect communicating entities against malicious attacks that can hinder their activities. However, they do not take into account the availability and fault tolerance of the protocols in case of any possible attack despite the authentication methods offered by encryption. This study therefore, proposes a fault-tolerant key distribution protocol for distributed mobile agents (communicating entities) in network intrusion detection system to facilitate hitch-free collaboration geared towards intrusive packets detection in Wireless Local Area Network (WLAN).

**Keywords:** Wireless Local Area Network, Fault-Tolerant, Mobile Agent, Key Distribution Protocol, Intrusion Detection

## 1. INTRODUCTION

Since Mobile agent systems found its application in a distributed environment, it becomes clear that they are naturally vulnerable to various security threats (Marikkannu *et al*., 2011). The application of mobile agent systems in distributed network enhances the efficiency of the network but the security aspect of the agent paradigm is a worrisome issue that requires persistent attentions of the researchers. The significance of computer networks in all facets of life necessitates collective efforts to ensure its availability and sustainability. Individuals and organizations nowadays rarely carry out their duties without using network facilities and internet technologies. This development has inspired the intruders to view computer network and internet as veritable platforms to execute their nefarious acts due to the unlimited data resource repository of the platforms.

To deploy mobile agent based security mechanism in a distributed network, one must also consider security issues associated with the mobile agents since they are potential target of the attackers. The intruders attack mobile agents to make the target network vulnerable. The vulnerability of the network hence paves way for their ultimate mission. In fact, no individual or organization would accept to host mobile agents if no security mechanism in place to facilitate the protection of the hosts and of the static agents running on the host (Leila and Barka, 2008).

**Corresponding Author:** Olumide Simeon Ogunnusi, Department of Computer Science, Faculty of Computing, Universiti Teknologi Malaysia, 81310, Skudai, Johor Bharu, Malaysia

An agent system (Hayzelden and Bourne, 2001) is made up of two actors: the mobile agents and platforms. An agent is an autonomous software entity that implements a task on behalf of its owner in a distributed environment. It can either be a static agent or mobile agent. As the name implies, a static agent resides in the network hosts as 'home guard' to monitor the activities of users, programs and intruders within the host machine. Mobile agents move from one host to another to perform the designated tasks as instructed by their owners. The second actor is the agent execution environment. Mobile agents make use of the platform resources (such as database and communication) to perform their tasks. The platform is also responsible for the creation of mobile agents. An agent comprises of three components: code, data and state. The agent code contains program instructions that state the task(s) to be performed by an agent. Data are the local raw facts needed for the agent code execution while the state spells the execution state of the program.

This study proposes a fault tolerant key distribution protocol for mobile agents in network intrusion detection system. It considers a scenario of collaborative mobile agents saddled with a collective task of detecting intrusive packets in data traffic in a Wireless Local Area Network (WLAN). The agents hopping region is confined to the WLAN. However, visiting (external) agents are welcome to the WLAN provided they comply with the internal security policy of the network domain. To minimise computation and memory overheads, an Elliptic Curve Cryptosystem (ECC) and digital signature are used to make the agent collaboration environment and inter agent communication secured. The reason for the choice of ECC is because of its high efficiency and shorter key length compared to RSA while digital signature is to guide against repudiation. Date-constraint key management scheme (Chen *et al*., 2010) is employed to give validity period to the key. Once the key of an agent expires, such agent will be retracted and terminated. The main contributions of this study are: The design of mechanisms to protect the protocol against single point of security failure and making it fault tolerant using backup agent server in case of the failure of the main agent server and making the appointment of the execution platform dynamic. The backup server, which is configured to have the same resource capability as the main agent server, automatically takes over the responsibilities of the main server at the instance of failure of the main server; the agent server dynamically select the execution platform for the collaborative agents among the host platforms in the network so that it will be difficult for attackers to track and attack the execution platform.

The rest of the study is structured as follows: In Section 2, we discuss the background of our study and our survey of the existing related work. This has helped us to establish our focus and identify where we can contribute to knowledge with the state of the art security protocol to protect collaborative mobile agents saddled with the responsibility of detecting intrusive packet in wireless local area network. The architecture of our protocol is presented in Section 3, where we also present the key authentication technique of our security protocol and assumptions made. In Section 4, the key distribution protocol is presented with the notational algorithm of the protocol. We conclude our study in section 5.

## 2. BACKGROUND AND RELATED WORK

In this study, we are focusing on the protection of collaborative mobile agents against malicious visiting agent. Wireless network is an unguided or open network which is vulnerable to security threats. A malicious visiting agent can intercept and attack legitimate collaborative agents. Therefore, protecting these agents against such threat would facilitate hitch-free and successful intrusion detection task. Security attacks against mobile agents are categorized into (Berard *et al*., 2001): confidentiality attacks, integrity attacks and availability attacks. Confidentiality attacks are concerned with the illegal access to the agent components (code, data and state) by a malicious mobile agent. Such malicious mobile agent can spy, alter, steal or eavesdrop on, the legitimate agent code or data. Integrity attacks comprise of all attacks which modify, or interfere with the agent code, data or state. Availability attacks occur when a legitimate agent cannot access the resources for which it is given right. Such attacks include Denial of Service (DoS) attack. For example, an agent among the collaborative agents may deprive other agent from accessing its data which is among those to be considered for decision on intrusive data traffic in the network.

Intrusion occurs when an attacker takes advantage of the security vulnerabilities of a network and thus violates the confidentiality, integrity and availability of the objects (agents) on the network. Security mechanism is a fundamental requirement of wireless network and therefore, it is necessary that this security concern must

be articulated right from the beginning of the network design and deployment (Surraya *et al*., 2012).

Although many researchers have proposed different approaches to protect mobile agents against malicious agents but they are limited in some aspects, for example, the security mechanisms adopted in turn result to increased computation and memory overheads and consequent degradation of the network. More researches are still ongoing to minimize these overheads with the view to enhance the efficiency of the network.

Various cryptographic based mechanisms have been proposed to alleviate the problem of mobile agents communication security (Guan and Huanguo, 2010; Srivastava and Nandi, 2011a; 2011b). Cryptographic protocol generally runs in closed network domain where at least information regarding nature of the network is known (Chen *et al*., 2010) and the behaviours of the hosts and the agents running on the host can be monitored.

Venkatesan *et al*. (2010) proposed an advanced models to improve the efficiency of the existing Malicious Identification Police (MIP) model for scanning the incoming agent to detect the malicious intent and to restrain the attendance of vulnerabilities in the existing Root Canal algorithm for code integrity checks. They extended the MIP model with the policy to uniquely identify the agent owners in the distributed environment. The Root Canal algorithm is improved upon and the improved version is named eXtended Root Canal algorithm. This model is efficient in integrity protection of agents and agent platform. However, the confidentiality of the agents components necessary to prevent passive attack (such as spy and impersonation of agent code for future use) is not considered.

Carles *et al*. (2010) present a software architecture and a development environment for the implementation of applications based on secure mobile agents. They focus at facilitating and speeding up the process of implementing cryptographic protocols and to allow the reuse of these protocols for the development of secure mobile agents. This proposal uses agent builder to implement agent protection protocols using Mobile Agent Cryptographic Protection Language (MACPL), which is domain specific. The architecture does not provide a new method of protecting mobile agents but advocating a simpler way of implementing agent protection without expertise in cryptographic application programming and language that provides high level cryptographic functions and promotes code reuse.

Woei-Jiunn (2012) developed security schemes based on cryptographic solutions for preventing agents and hosts against illegal alteration. This study develops a proxy signature scheme and a proxy authenticated encryption scheme for the protection of mobile agents against malicious agent hosts using Elliptic Curve Cryptographic (ECC) based self certified public key cryptosystem. The proxy signature scheme is capable of protecting users' private keys stored in smart cards. The cryptosystem proposed by the author is constructed using the ECC integrated with identity based public key cryptosystem. This security scheme is implemented for the protection of Linux based mobile agent networks in an electronic auction application.

To prevent the problem of continuous use of the old key in Volker and Mehrdad (1998) by the users and the need to keep modifying the key so as to change the original access rights of the key, which could cause unnecessary error and risks, in addition to the large amount of computations that the system needs to perform, Chen *et al*. (2010) proposes date constraint key management scheme. This scheme attached a date to the key, so as to give it a validity period.

Srivastava and Nandi (2013) introduced a security protocol for the protection of mobile agent from various types of attacks. This protocol is built on the foundation of self protection approach based on agent driven security and integrity based confidentiality of mobile agents. The self-protection is instituted to make mobile agent less interactive during its execution. The idea of symmetric key's component distribution was adopted in which a key component is distributed in secure manner while the other key component is derived from ensuring integrity of data collected at run time. However, the concept of self protection of agent may not be useful in a situation where an application is agents interaction specific. That is, an application environment where the interaction of mobile agents is the ultimate for specific problem solving.

Abdelhamid *et al*. (2007) presents a protocol that protects mobile agents against malicious hosts. In this protocol, four concepts were combined: the collaboration between a sedentary agent and a mobile agent; the cryptography; the reference execution; the digital signature to facilitate secure communication between agents and time limited execution (timeout). The reference execution is a reliable platforms which shelter the collaborative sedentary agents).

In Secured routing using reputation value and trust value (Rajeshwar *et al*., 2012), the authors used two agents. One is to generate routing table and the second agent to retrieve data securely from non-malicious hosts. The data are secured using Elliptic Curve Cryptography (ECC) algorithm and Secure Hash Algorithm 1 (SHA1) (Padma *et al*., 2010; Megha and Jadhav, 2011). This model actually provides confidentiality protection for the agent data but imposes high computation overhead

as a result of the two level cryptographic processes taking place in each host the agent visits. If the agent has to visit all the hosts in the network, the model will cause performance degradation of the network when a large number of hosts are involved.

Protected Agent States (Neeran and Tripathi, 1999) model proposed the signing and encrypting of agent states using public key cryptography. The model achieved the confidentiality and integrity protection of the agent states but could not protect the agent code integrity and confidentiality.

Shopping Consultant Agent System (SCAS) (Kannammal and Iyengar, 2008) is developed for the security issues associated with agent based electronic business systems (Zachary, 2003). SCAS is a web based agent system, which hops the net for information on the products for sale in an electronic marketplace. With this agent system, users specify items to buy and the corresponding quantities. The agent then collects the price details from sellers hosts in the electronic marketplace. The agent visits all the hosts in its itinerary and returns to its sender with the report stating the prices of the items. This model enhances the security of the agent by establishing a closed network and using agent tampering prevention. The issue here is that the established closed network limits the mobility and autonomy of the mobile agent.

Hyungjick *et al.* (2004) focused on extending mobile agent cryptography to agent security proposed by (Sander and Tschudin, 1998a; 1998b; 1998c) in terms of privacy and integrity by considering composite functions and additivemultiplicative homomorphism to encrypt mobile agents such that the encrypted mobile agent can run on any host without decryption. The execution of the mobile agent will also generate encrypted result which will be decrypted by the agent owner on the agent arrival. However, this security technique does not protect the mobile agent against Denial of Service attack of the host platform on which the agent is running. Also, the use of additive multiplicative homomorphism is a cryptographic scheme that wraps function inside another function, which requires more computations than a simple cryptographic function thereby imposes high computation overhead on the network. This, in effect, diminishes the performance of the network.

## 3. SYSTEM ARCHITECTURE

The architecture (**Fig. 1**) considers two agent security domains: the sending security domain and the receiving security domain. An agent security domain is the local network domain of an agent system. In this case, the Agent Server (AS) of the receiving security domain generates a private/public key pair locally, requests for its certificate from the Certification Authority ($CA_2$) and stores them in its keystore. It appoints the execution platform among the host platforms in the network.

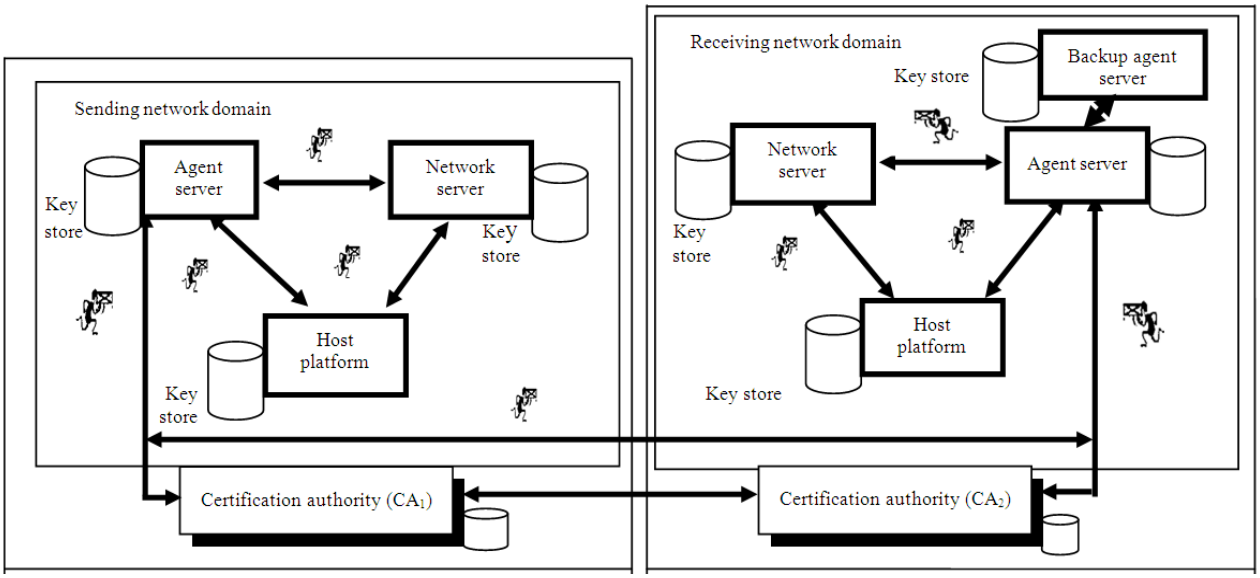


**Fig. 1.** System architecture

It creates mobile agents, distributes to them certificates and public keys and encrypts each agent components (data and code) with the agent's private key. Then sign the agent components and its certificate with the AS's private key before they are dispatch to the network. Having taken samples of the network traffic, the mobile agents migrate to the execution platform. The execution platform authenticates and registers them before they are allowed to execute and collaborate to attain their collective goal. The Backup server is designed to take charge of the responsibility of the main agent server whenever it fails to perform it function. So, all activities of the agent server are automatically duplicated in the backup agent server.

The sending security domain is assumed to have its independent Certification Authority ($CA_1$) as evident in the architecture but have communication link with $CA_2$ for the purpose of authenticating the visiting agent.

The method adopted for the key distribution in our work includes the following components:

- Mobile agent: A mobile agent is an autonomous software agent that has the capability of moving from one host to another with its components (i.e. data, code and state) to achieve its designated task of collaborating with other agents to decide on intrusive packets. To communicate with other agents, the agent invokes methods which will be translated into messages by the underlying messaging mechanism. In our architecture, all the collaborating mobile agents are created by the agent server, named and dispatched to collect network traffic data.
- Agent server: The agent server is the creator of all the mobile agents. It requests for the certificates and private/public key pairs from $CA_1$, allocates the certificates and public keys to each of the mobile agents at creation. The agent server is also responsible for maintaining the database of received certificates and private keys. To prevent against single point of security failure, the agent server has a Backup Agent Server (BAS) that possesses same resources as the agent server. The agent server duplicates all its records and processes in the BAS to forestall total breakdown of the protocol and the agent system if the agent server is incapacitated for any reason.
- Backup Agent Server: It is a replica of the agent server and is designed to play the role of the agent server if it breaks down (fault tolerance). Its purpose is to ensure uninterrupted availability of the agent server and its designated functions. The agent server must constantly backup its activities such as creation of new agents, private/public key pairs and digital certificates allocated to newly created agents, termination of existing agent, revocation of certificate of an expired agent, retraction of an erring agent, admission of an external agent and its associated key pair and digital certificate
- Certification authority: The certification authority is a trusted third party appointed to facilitate reliable intercommunication between two or more network (security) domains. The communicating network domains may most likely have different Certification Authorities (CA) as evident in our protocol. In our architecture, we considered two security domains with different CAs ($CA_1$ and $CA_2$). In this case, the Visiting Agent (VA) from the sending security domain must be authenticated and certified trusted and granted certificate by its domain CA ($CA_1$) before forwarding it to the receiver CA ($CA_2$). Thus $CA_2$ will verify and grant certificate before it is allowed to migrate to agent server of the receiving security domain. On getting to the agent server of the receiving security domain, it is allocated local private/public key pair before it can be allowed to execute on any of the agent server, network server or host platform
- Security domain: A security domain is the network domain of an agent system. It consists of an agent server, backup agent server, network server, host, keystore and mobile agents. Mobile agents in the WLAN can move within the security domain to perform their function. The WLAN is a confined network domain for the mobile agents since their activities are limited to the domain. Their activities are simply capturing network traffic data and collaborate to detect intrusive packets in the traffic data
- Messaging system: This is part of the agents execution environment. In our architecture, any of the network host can be selected by the agent server as the execution host. This is to relieve the agent server of the responsibility of collaborating agents execution and concentrate on security-critical issues such as agent creation and key generation and distribution. Messaging system provides facilities for agents' local and remote communication. It establishes communication links between collaborating agents
- Execution platform: The execution platform is a host platform or network server platform appointed

by the agent server to coordinate, monitor and supervise the execution and collaboration of the cooperating agents. It reports any erring mobile agent to the agent server for retraction. The execution platform decrypts all the mobile agents on arrival and makes available all needed resources to facilitate their effective collaboration for intrusive packets detection

- Keystore: Each of the agent server, network server and host platform has a local database used for storing and retrieving the agents' public key and their digital certificates. All digital certificates signed by the CA authority are also stored in the agent server database

It is worth noted that the sending network domain is an outside network with its own independent service agents not considered in our security mechanism. The service agent of the outside network is what we considered as a Visiting Agent (VA) to our local area network.

### 3.1. Key Authentication

The focus of our research is on mobile agent communication security in an agent based intrusion detection system in Wireless Local Area Network (WLAN). The WLAN infrastructure is protected using certification authority model. The communicating mobile agents within the security domain are protected against possible attacks by visiting agent from another security domain using same model. First and foremost, the agent servers of the two security domains must engage in mutual authentication (explained in Section IV). To protect the mobile agents, certificate and public key of Visiting Agent (VA) must be authenticated. This is done using certification authority and key authentication models (also explained in section IV). However, the agent server of the receiving security domain constraints the VA from visiting the execution platform, where agents collaboration is taking place.

### 3.2. Assumption

Our assumptions for the security protocol are:

- The two network domains considered in our protocol are using two distinct Certification Authorities
- The visiting agent is not having the same mission as the cooperating mobile agents
- The network server, execution platform, host platform and the cooperating mobile agents are trusted entities since they are all components of a trusted Local Area Network. The essence of protecting the cooperating agents is to secure them against man in the middle attack while in transit and any possible attack occasion by intrusive agent during their collaboration session. However, authentication of the agents and agent platforms and VA denial of visitation to execution platform are still considered necessary in our protocol

## 4. KEY DISTRIBUTION PROTOCOL

Our protocol (**Fig. 2**) adopts the phases in key distribution protocol as proposed by (Leila and Barka, 2008) with some modification as follows:

- Pre-configuration phase: The security domain administrators configure the local keystores of the agent servers which are used for the storage of the private/public key pairs
- Initialization phase: The security administrator starts the agent server with the following actions
- The agent server create the mobile agent
- The agent server generates public/private key pair for the agent
- The agent server also request for a certificate from its CA for the mobile agent
- Operations b(i) to b(iii) are repeated by the agent server for the number of mobile agents created
- Appointment and registration phase: Appointment of the execution host and the registration of both the execution host and other hosts in the network take place at this phase with the following actions
- The agent server appoints an execution host (execution platform), generate public/private pair and allocate to it a private key
- the agent server request for the digital certificates from $CA_2$ for both the execution host and the other hosts
- $CA_2$ issued the agent server certificates for both the execution host and the other hosts
- Mobility phase: This mobility phase is broken into two sub phases
- Mobility within the security domain of the cooperating mobile agents: The initialized local mobile agents are dispatched with their attached public keys for decryption on arrival at the execution host
- Mobility across security domain: This occurs when a mobile agent from one security domain is visiting another security domain. The following actions take place

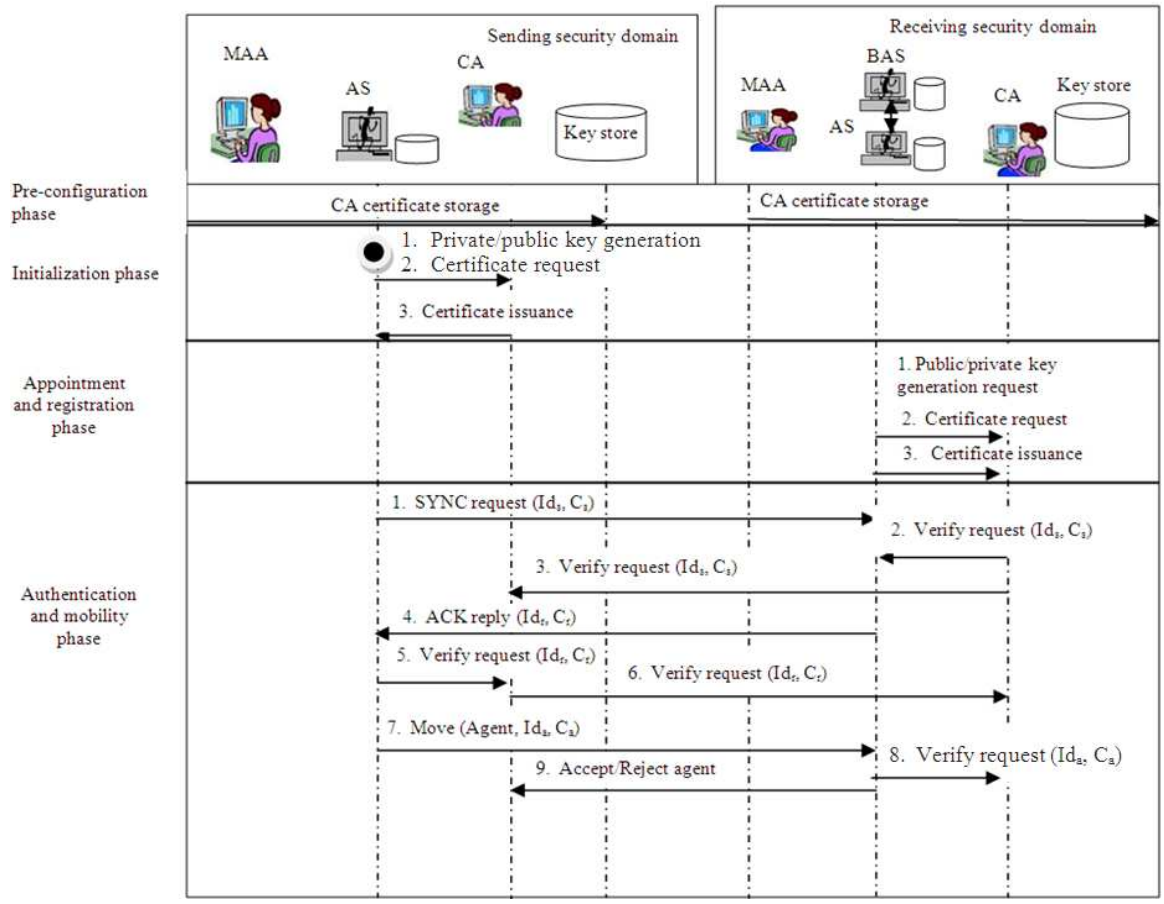


**Fig. 2.** Key distribution protocol

- The sender agent server sends SYNC request with its identity and certificate to the receiver agent server
- VERIFY operation at the receiver agent server works as follows
- If the identity and certificate of the sender agent server is not with the receiver agent server, the receiver agent server requests for it from $CA_2$. If the $CA_2$ of the receiver agent server does not have the certificate, the $CA_2$ requests for it from $CA_1$
- The receiver agent server sends an ACK reply with its identity and certificate to the sender agent server.
- VERIFY operation at the sender agent server works as follows
- If the sender agent server does not have the identity and certificate of the receiving agent server, it verifies from $CA_1$. If $CA_1$ does not have the information, it sends request to the $CA_2$
- The sender agent server sends the message with the agent identity and certificate to the receiver agent server
- VERIFY operation at the receiver agent server follows the steps below
- The receiver agent server request for the verification of the certificate of the visiting mobile agent since it does not have it in its local key store
- The $CA_2$ of the receiving agent server acknowledged and signed the certificate of the visiting agent and forwarded it to the receiver agent server
- The receiver agent server accepts or rejects the visiting agent and informs the sender agent server about the acceptance or rejection of the visiting agent. The notational algorithm of the protocol is shown in **Fig. 3** below

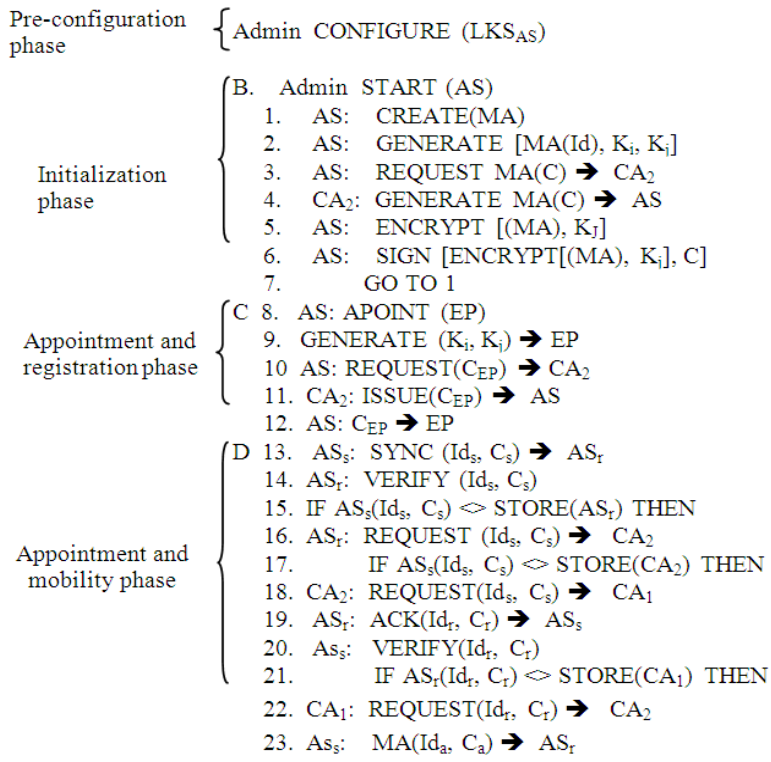


**Fig. 3.** Notational algorithm of the key distribution protocol

**Table 1.** Notation and meaning

| WLAN | Wireless Local Area Network |
|---|---|
| $CA_1$ | Certification Authority of the sending security domain |
| $CA_2$ | Certification Authority of the receiving security domain |
| $C_{EP}$ | Certificate of Execution Platform |
| $Id_s$ | Identity of agent server of the sending security domain |
| $C_s$ | Certificate of agent server of the sending security domain |
| $AS_r$ | Agent server of the receiving security domain |
| $AS_s$ | Agent server of the sending security domain |
| $Id_r$ | Identity of agent server of the receiving security domain |
| $C_r$ | Certificate of agent server of the receiving security domain |
| $Id_a$ | Identity of visiting agent from the source security domain |
| $C_a$ | Certificate of visiting agent from the source security domain |
| AS | Agent server |
| $K_j$ | Public key |
| EP | Execution Platform |
| VA | Visiting agent |
| $K_i$ | Private key |
| CA | Certification Authority |
| C | Certificate |
| BAS | Backup Agent Server |
| Id | Identity |
| DoS | Denial of Service |
| MA | Mobile Agent |
| LKS | Local Key Store |
| MAA | Mobile Agent Administrator |

**Table 1** shows the notations used in this article and their coresponding meanings.

## 5. CONCLUSION

In this research, we have focused on the design of fault tolerant security protocol for the protection of collaborating mobile agents against possible attacks that can threaten the confidentiality of the agents while migrating to the execution platform where they will exchange views on their captured suspected intrusive packets in order to arrive at a collective decision and report on the detected intrusion in the network and also during their collaboration session. This design is just a part of our ongoing research work. The protocol exploits two possible means at ensuring fault tolerance and availability. Firstly, the establishment of a backup agent server, which has the same resource capability as the main server and all activities of the main server are automatically duplicated in the backup agent server. Whenever the main server is incapacitated, the backup agent server takes up its responsibility immediately. Secondly, the appointment of the execution platform is dynamic. The agent server randomly selects the execution platform among the host platforms therefore making the execution platform uncertain to any intruder to target. After the research exercise, our protocol will be compared with some other existing similar protocols with the view to measure its efficiency.

## 6. ACKNOWLEDGEMENT

This study was supported by the Universiti Teknologi Malaysia and Federal Polytechnic, Ado-Ekiti, Nigeria. We are thankful to the two institutions.

## 7. REFERENCES

Abdelhamid, O., S. Pierre and H. Boucheneb, 2007. A security protocol for mobile agents based upon the cooperation of sedentary agents. J. Netw. Comput. Applic., 30: 1228-1243. DOI: 10.1016/j.jnca.2006.04.008

Berard, B., M. Bidoit, A. Finkel, F. Laroussinie and A. Petit *et al*., 2001. Systems and Software Verification: Model-Checking Techniques and Tools. 1st Edn., Springer, Berlin, ISBN-10: 3540415238, pp: 190.

Carles, G., S. Robles, J. Borrell and G.N. Arribas, 2010. Promoting the development of secure mobile agent applications. J. Syst. Soft., 83: 959-971. DOI: 10.1016/j.jss.2009.11.001

Chen, T.L., Y.F. Chung and F.Y.S. Lin, 2010. An efficient date-constraint hierarchical key management scheme for mobile agents. Expert Syst. Applic., 37: 7721-7728. DOI: 10.1016/j.eswa.2010.04.069

Guan, H., H. Zhang X. Meng and J. Zhang, 2010. A communication security protocol of mobile agent system. Wuhan Univ. J. Natural Sci., 15: 117-120. DOI: 10.1007/s11859-010-0206-9

Hayzelden, A.L.G. and R.A. Bourne, 2001. Agent Technology for Communication Infrastructures. 1st Edn., John Wiley, Chichester, ISBN-10: 0471498157, pp: 296.

Hyungjick, L., J.A. Foss and S. Harrison, 2004. The use of encrypted functions for mobile agent security. Proceedings of the 37th Hawaii International Conference on System Sciences, Jan. 5-8, IEEE Xplore Press, pp: 1-10. DOI: 10.1109/HICSS.2004.1265700

Kannammal, A. and N.C.S.N. Iyengar, 2008. A framework for mobile agent security in distributed agent-based e-business systems. Intern. J. Bus. Inform., 3: 129-143.

Leila, I. and E. Barka, 2008. Key distribution framework for a mobile agent platform. Proceding of the 2nd International Conference on Next Generation Mobile Applications, Services and Technologies, Sept. 16-19, IEEE Xplore Press Cardiff, pp: 281-287. 10.1109/NGMAST.2008.61

Megha, K. and A. Jadhav, 2011. Implementation of elliptic curve cryptography on text and image. Intern. J. Enterprise Comput. Bus. Syst., 1: 1-13.

Neeran, M.K. and A.R. Tripathi, 1999. Security in the ajanta mobile agent system. Technical Report, Department of Computer Science.

Padma, B.H., D. Chandravathi and P.P. Roja, 2010. Encoding and decoding of a massage in the implementation of elliptic curve cryptography using koblitz's method. Intern. J. Sci. Eng.

Rajeshwar, B., A. Saravanan, R.S. Balaji, G. Geetha and C. Jayakumar, 2012. Secure information retrieval using mobile agent. Proceedings of the International Conference on Computing and Control Engineering, (CCE' 12).

Sander, T. and C. Tschudin, 1998b. On software protection via function hiding. Inform. Hiding, 1525: 111-123. DOI: 10.1007/3-540-49380-8_9

Sander, T. and C. Tschudin, 1998c. Protecting Mobile Agents Against Malicious Hosts. In: Mobile Agent Security, Vigna, G. (Ed.), Springer-Verlag, Heidelberg, Germany, ISBN-10: 3540647928, pp: 44-60.

Sander, T. and C.F. Tschudin, 1998a. Towards mobile cryptography. Proceedings of the IEEE Symposium on Security and Privacy, May 3-6, IEEE Xplore Press, Oakland, CA., pp: 215-224. DOI: 10.1109/SECPRI.1998.674837

Srivastava, S. and G.C. Nandi, 2011a. Detection of mobile agent's blocking in secure layered architecture. Proceedings of the IEEE International Conference on Communication Systems and Network Technologies, Jun. 3-5. IEEE Xplore Press, Katra-Jammu, India, pp: 43-48. DOI: 10.1109/CSNT.2011.16

Srivastava, S. and G.C. Nandi, 2011b. Protection of mobile agent and its itinerary from malicious host. Proceedings of the 2nd IEEE International Conference on Computer and Communication Technology, Sept. 15-17, IEEE Xplore Press, Allahabad, India, pp: 405-411. DOI: 10.1109/ICCCT.2011.6075189

Srivastava, S. and G.C. Nandi, 2013. Self-reliant mobile code: A new direction of agent security. J. Netw. Comput. Applic., DOI: 10.1016/j.jnca.2013.01.004

Surraya, K., M. Usman and A. Alwabel, 2012. Mobile agent based hierarchical intrusion detection system in wireless sensor networks. Int. J. Comput. Sci., 9: 101-108.

Venkatesan, S., C. Chellappan, T. Vengattaraman, P. Dhavachelvan and A. Vaish *et al*., 2010. Advanced mobile agent security models for code integrity and malicious availability check. J. Netw. Comput. Applic., 33: 661-671. DOI: 10.1016/j.jnca.2010.03.010

Volker, R. and J.S. Mehrdad, 1998. Access control and key management for mobile agents. Comput. Graphics, 22: 457-461. DOI: 10.1016/S0097-8493(98)00035-1

Woei-Jiunn, T., 2012. Secure communication for electronic business applications in mobile agent networks. Expert Syst. Applic., 39: 1046-1054. DOI: 10.1016/j.eswa.2011.07.105

Zachary, J., 2003. Protecting mobile code in the world. IEEE Internet Comput., 22: 457-461. DOI: 10.1109/MIC.2003.1189192

Marikkannu, P., J.J.A. Jovin and T. Purusothaman, 2011. Fault-tolerant adaptive mobile agent system using dynamic role based access control. Int. J. Comput. Applic., 20: 1-6.