

## **A Framework for Mobile Agent Security in Distributed Agent-Based E-Business Systems**

**A. Kannammal**

*Department of Computer Applications  
Coimbatore Institute of Technology  
Coimbatore 641 TamilNadu, India  
[kannaphd@yahoo.com](mailto:kannaphd@yahoo.com)*

**N. Ch.S.N. Iyengar**

*School of Computing Sciences, VIT University  
Vellore-632014, TamilNadu, India  
[nchsniyr@yahoo.com](mailto:nchsniyr@yahoo.com)*

### **ABSTRACT**

Mobile agent systems provide a great flexibility and customizability to distributed applications like e-business and information retrieval in the current scenario. Security is a crucial concern for such systems, especially when they are used to deal with money transactions. Mobile agents moving around the network are not safe since the remote hosts that accommodate the agents can initiate all kinds of attacks and attempt to analyze the agents' decision logic and the agents' accumulated data. Hence, mobile agent security is one of the most challenging problems unsolved. This paper analyzes the security attacks to mobile agents by malicious hosts and proposes solutions based on public key authentication technique and cryptography to address some of these problems. An experimental application is developed, and security and performance of proposed solutions are also evaluated. A performance model is developed in order to tune the parameters of execution environment to meet the desired level of performance and security.

**Keywords:** mobile agent security, e-business, performance model, distributed systems

## **1. INTRODUCTION**

Mobile agents are autonomous software agents that travel in a computer network to execute and perform tasks on different hosts for their owners. Autonomous mobile agents bring advantages like task delegation, network communication, cost reduction, etc., for distributed tasks [Danny and Oshima, 1999]. Security is one of the blocking factors of the development of these agent-based systems. The problem of mobile agent security can be divided into two parts: protection of hosts against agents, and protection of agents against hosts. Host security problems can be effectively solved by strong authentication of code sources, verification of code integrity, and limitation of access rights. The solution is realized in the Java security model [Sun]. Agent security problems derive from the possible existence of malicious hosts that can manipulate the execution and data of agents [Fritz and Hohl, 1998]. The lack of a trustworthy computing base [Tomas and Tschudin, 1998] also adds new complexities to the problem.

The inherent advantages provided by agent-based systems are hampered mainly by security concerns, especially when agents are used to deal with money transactions. To experiment with mobile agents, a mobile agent system named *Shopping Consultant Agent System (SCAS)* is built using the Java Agent Development Environment (JADE). The system is useful to collect the prices of a set of products specified by users from different seller hosts in an electronic market. Security issues of the system, possible attacks by malicious hosts, and solutions to protect the system against these attacks are devised and implemented. A performance model is developed to evaluate the overhead introduced when mobile agents are implemented with the proposed solution and also to tune the parameters of execution environment to meet the desired level of performance and security. The experimental system is updated to a distributed system in order to make the security enhancements feasible for real-time distributed e-business systems. The performance model is also updated and analyzed to observe the potential benefits of the proposed solutions.

## **2. SECURITY ISSUES OF AGENT-BASED E-BUSINESS SYSTEMS**

Any distributed system is subject to security threats such as eavesdropping, corruption, masquerading, and denial of service, replaying, and repudiation. A mobile agent system is subject to the same threats. Therefore, issues such as encryption, authorization, authentication, and non-repudiation should be addressed in a mobile agent system. Moreover, a secure mobile agent system must protect the hosts as well as the agents from being tampered by malicious parties.

Mobile agents moving around the network are not safe because the remote hosts that accommodate the agents can initiate all kinds of attacks and attempt to analyze the agents' decision logic and the agents' accumulated data. The hosts, in which the agents execute, have complete control over the agents. When the

mobile agent arrives at the host, the agent is loaded into the host's memory [Fritz and Hohl, 1998]. The host machine is armed with the external environment like the system clock and the code library to access the system or to access other host-specific information, to target the mobile agent. A program code can be inserted, modified, deleted, and selectively executed. The vulnerabilities of the mobile agent to execute in the hostile environment are hence readily reflected. If the host is malicious, the agents are exposed to security threats that may violate confidentiality, integrity, authentication, availability, non-repudiation, etc. A number of solutions are proposed to protect agents against malicious hosts [Tschudin, 1999], which can be divided into three streams: establishing a closed network, agent tampering detection, and agent tampering prevention. None of these proposed solutions solve the problem completely, however.

Two of the proposed solutions that look most feasible and interesting to protect mobile agents are protected agent states and mobile cryptography.

**Protected Agent States** [Neeran and Tripathi, 1999], which basically involve the signing and encrypting of agent states based on public key cryptography that helps to achieve the confidentiality and integrity of the agent's data. They are effective and feasible to protect agent states, because of the well-established cryptography theory underneath. However, they do not protect the code integrity and confidentiality of agents.

**Mobile Cryptography** [Tomas and Tschudin, 1998], which is a possible approach to protect agent code integrity works as follows. *If Alice wants Bob to evaluate a function  $f$  for her, based on Bob's data  $x$ , she would encrypt the function  $f$  to produce  $E(f)$ , and implement a program  $P$  that evaluates the encrypted function  $E(f)$ , and send  $P$  to Bob. When Bob runs  $P$ , he would not produce plaintext output  $f(x)$  that he can read and modify. Instead, he can produce only the encrypted output  $E(f(x))$ , which would be readable only by Alice, who has the key to decrypt. Moreover, Bob cannot modify the execution of  $P$ , because it implements an encrypted function that Bob would not understand.*

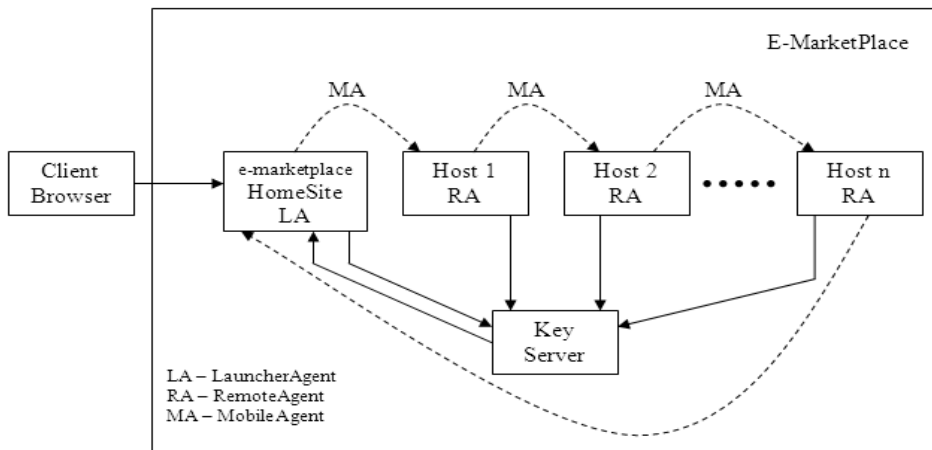
This approach is proved to be possible for polynomial functions only. However, if it really can be extended to other functions, such that arbitrary functions can have an encrypted but executable form that can be evaluated on a remote host, the problem of malicious hosts will be effectively solved.

### 3. EXPERIMENTAL AGENT-BASED E-BUSINESS SYSTEMS

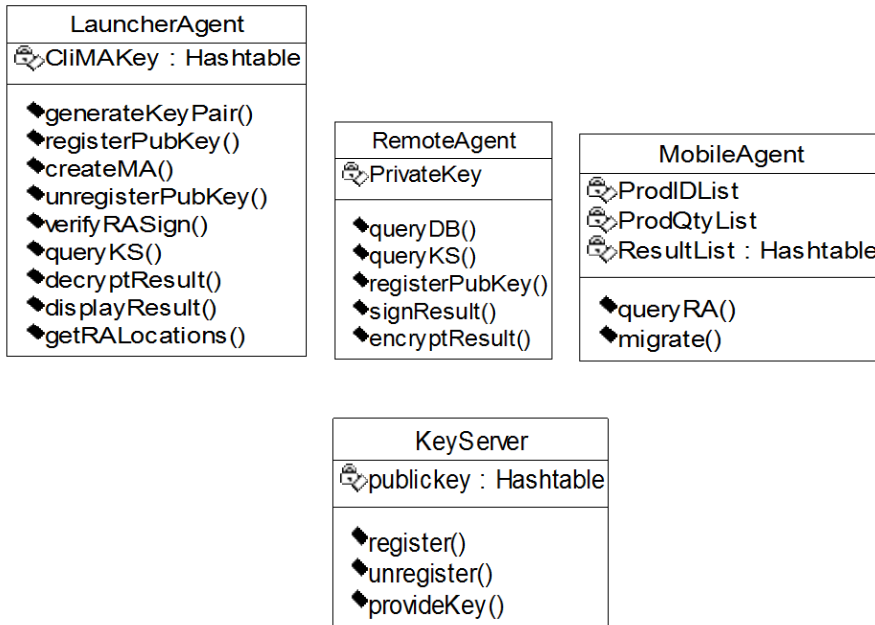
An experimental multi-agent system – namely, the Shopping Consultant Agent System (SCAS) – is designed and developed to identify the security issues that may arise in agent-based e-business systems [Zachary, 2003]. The SCAS is a Web-based mobile agent system that provides users with information on the products for sale in an electronic marketplace (e-marketplace). SCAS allows

users to specify a set of products and the corresponding quantities they want to buy. An agent is created for the user, which will collect price details from hosts in the e-marketplace. The itinerary of the agent is determined before the agent is launched. After the agent visits all hosts specified in its itinerary, it returns to its sender and reports the prices. The SCAS architecture and component design are presented in Figure 1 and Figure 2, respectively.

The security issues and attacks that violate the security requirements are well explained in Kannammal, Ramachandran, and Iyengar, [2006]. It also enhances the security by means of two approaches; namely, establishing a closed network, and agent tampering prevention. This paper is an extension to the work, which formulates a performance model for the secured SCAS. Also, the e-marketplace is updated to a distributed e-marketplace with necessary updating in the performance model.



**Figure 1. SCAS Architecture**



**Figure 2. SCAS Component Design**

**4. PERFORMANCE MODEL FOR SECURED SCAS**

A performance model has been developed for the secured SCAS in order to evaluate the impact of proposed security solutions on the performance. The parameters selected for this case study are based on the workflow given in Kannammal, et al. [2006], and are summarized in Table 1.

Since the mobile agent does not perform the encryption task before it is launched, it does not generate the result of the task. It consists of code and state information. Its size  $D_{init}$  is given by

$$D_{init} = D_{code} + D_{state} \tag{1}$$

The mobile agent signs the product ID and the quantity specified (the details of query). Hence, its initial time for migration  $T_{init}$  is

$$T_{init} = D_{state} * (1 / R_s) \tag{2}$$

When a mobile agent migrates among nodes, it performs its task and generates the result of the task. While a mobile agent is migrating among nodes, its size is defined as

$$D_{mig} = D_{code} + D_{state} + D_{data} \tag{3}$$

**Table 1. Parameters for Evaluating Security Enhancements**

Name	Description
N	Number of Nodes
D <sub>code</sub>	Code size of MobileAgent (MA)
D <sub>state</sub>	Status data size of MA (prod id and qty and data other than that collected from remote hosts)
D <sub>data</sub>	Amount of data collected from remote hosts
T <sub>reqPK</sub>	Time taken to request public key
T <sub>respPK</sub>	Time taken to respond for public key
R <sub>s</sub>	Rate of signature creation
R <sub>v</sub>	Rate of signature verification
R <sub>e</sub>	Rate of encryption
R <sub>d</sub>	Rate of decryption
R <sub>se</sub>	Rate of database search
R <sub>th</sub>	Network throughput
T <sub>reqdata</sub>	Time taken to request data
D <sub>init</sub>	Size of MA before migrating
T <sub>init</sub>	Initial time required for MA migration
D <sub>mig</sub>	Size of MA while migrating
T <sub>mig</sub>	Time taken for MA to migrate and retrieve information
T <sub>Math</sub>	Total time for MA migration (MA throughput)
T <sub>MAAuth</sub>	Time taken for MA authentication
T <sub>Sdata</sub>	Time taken for signing and encrypting the result
T <sub>MASec</sub>	Time to provide security services
T <sub>SMA</sub>	Total Execution time for Secured SCAS

The total execution time of the model amounts to the time for information retrieval and the time for security services such as authentication, integrity, and confidentiality.

Let  $T_{mig}$  be the time required that a MobileAgent moves to a remote host and retrieves the information.  $T_{mig}$  is given by

$$T_{mig} = (1 / R_{se}) + (D_{mig} / R_{th}) \quad (4)$$

The total time required for a mobile agent to execute the assigned task,  $T_{MAth}$  is given by

$$T_{MAth} = (D_{init} / R_{th}) + T_{init} + (D_{data} / R_{th}) + (N - 1) * T_{mig} \quad (5)$$

The remote host has to authenticate the mobile agent by requesting the public key of the mobile agent from the key server. The necessary time for mobile agent authentication,  $T_{MAAuth}$ , is given by

$$T_{MAAuth} = T_{reqdata} + T_{reqPK} + T_{respPK} + (D_{state} * (1 / R_v)) \quad (6)$$

Time for signing and encrypting the results to be provided by the remote host,  $T_{Sdata}$ , is defined as

$$T_{Sdata} = D_{data} * ((1 / R_s) + (1 / R_e)) \quad (7)$$

The time for providing integrity and confidentiality is defined as:

$$T_{MASec} = (D_{data} / R_s) + (D_{data} / R_e) + (D_{data} / R_d) + (D_{data} / R_v) + (N * (D_{state} / R_v)) + (N - 1) * T_{Sdata} \quad (8)$$

If the mobile agent visits  $N$  nodes, it needs to  $N + 1$  migration and authentication. The total execution time for the secure SCAS,  $T_{SMA}$ , is defined as,

$$T_{SMA} = T_{MAth} + (N + 1) * T_{MAAuth} + T_{MASec} \quad (9)$$

The overhead introduced in the case with security enhancements is due to the extensive use of the RSA algorithm to encrypt and decrypt each item, which is time-consuming especially when the key is long (a longer key gives a stronger protection to the system). Hence, a trade-off between performance and security for SCAS is identified.

## 5. SECURITY MODEL FOR LARGE-SCALE DISTRIBUTED ENVIRONMENT (MULTIPLE E-MARKETPLACES)

The performance of the secured SCAS gradually degrades because of the security implementation, which, in turn, is due to the extensive cryptographic operations involved. Also, when the mobile agents are launched to travel multiple e-marketplaces, then the system should meet the scalability requirements, also. When the agent has to collect information from multiple e-marketplaces, then the performance overhead would become too high to be possible to apply in a real-

time scenario. Keeping the key server as a centralized one makes it a bottleneck for the system.

Though security requirements are met, it is difficult to apply this model to the large-scale distributed environment because of the centralized key server. Also, it causes the execution time to increase sharply because of the use of costly security operations during travel.

Hence, a new distributed security model is proposed in this section, which aims to address the security problems identified in the previous sections, simultaneously handling performance degradation issues. The proposed model is based on Trusted Domain Guide Manager (TDGM) proposed by You and Lee [2004].

### **5.1. Secured and Distributed SCAS (SDSCAS) Architecture**

The experimental multi-agent system – namely, Shopping Consultant Agent System (SCAS)—is updated in such a way that the mobile agent is launched to collect information from multiple e-marketplaces. Figure 3 shows an overview of the architecture of a distributed e-marketplace, along with the proposed security enhancements. The client can send a request to any of the ‘k’ number of e-marketplaces in the given environment. Each of these e-marketplaces maintains the address details of other e-marketplaces that are part of the architecture. Also, each of these e-marketplaces maintains the public key details of participating hosts present in its own system. Here, the concept of a closed network of e-marketplaces is used. Again, the closed network of hosts is maintained.

### **5.2. Components Design of SDSCAS**

The components present in the secured SCAS are redesigned to reflect the updates introduced in the new security model. Following is a detailed description of each of the components.

Each e-marketplace is responsible for maintaining the public key and private key details of each of the hosts registered with it.

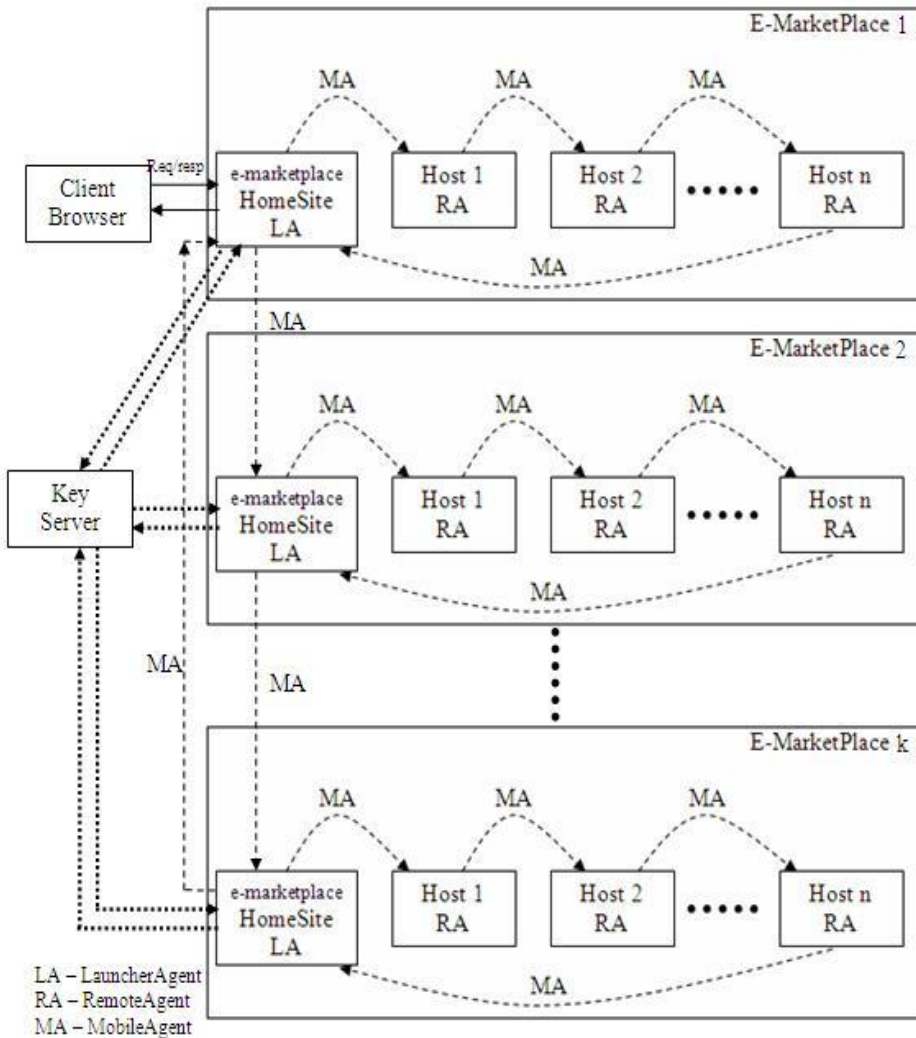
The KeyServer is a kind of trusted third party that provides services in the same way as a centralized key management authority. It resides outside any e-marketplace, as a separate independent entity, which is responsible for:

- Storing the public key details of LauncherAgent present in the home-site of participating e-marketplaces
- Registering and de-registering the MobileAgent public key details
- Providing the public key details to LauncherAgents

The LauncherAgent present in the e-marketplace that receives the client’s request is responsible for:

- Creating a MobileAgent for the client

- Creating a private key/public key pair for the MobileAgent



**Figure 3. Secured SCAS Architecture for Distributed E-Marketplaces**

- Registering the public key of the MobileAgent with the KeyServer
- Signing the query using its own private key
- Providing the query details to the MobileAgent
- Providing the list of hosts to be visited by the MobileAgent in the current e-marketplace

- Providing the list of e-marketplaces to be visited by the MobileAgent in the entire environment
- Launching the MobileAgent

The LauncherAgent present in the e-marketplace (which is not the one that has created the MobileAgent) that receives the MobileAgent from another e-marketplace is responsible for:

- Retrieving the public key of MobileAgent
- Verifying the signature of the query using public key of LauncherAgent of home e-marketplace
- Providing the list of hosts to be visited by the MobileAgent in the current environment
- Launching the MobileAgent

The LauncherAgent present in the e-marketplace (which is the one that has created the MobileAgent), that receives the MobileAgent from another e-marketplace is responsible for:

- Verifying the signature of the query using its own public key
- Retrieving the public key of LauncherAgents from KeyServer
- Verifying the signature of query results
- Decrypting the result using private key of MobileAgent
- Sending the response to the client

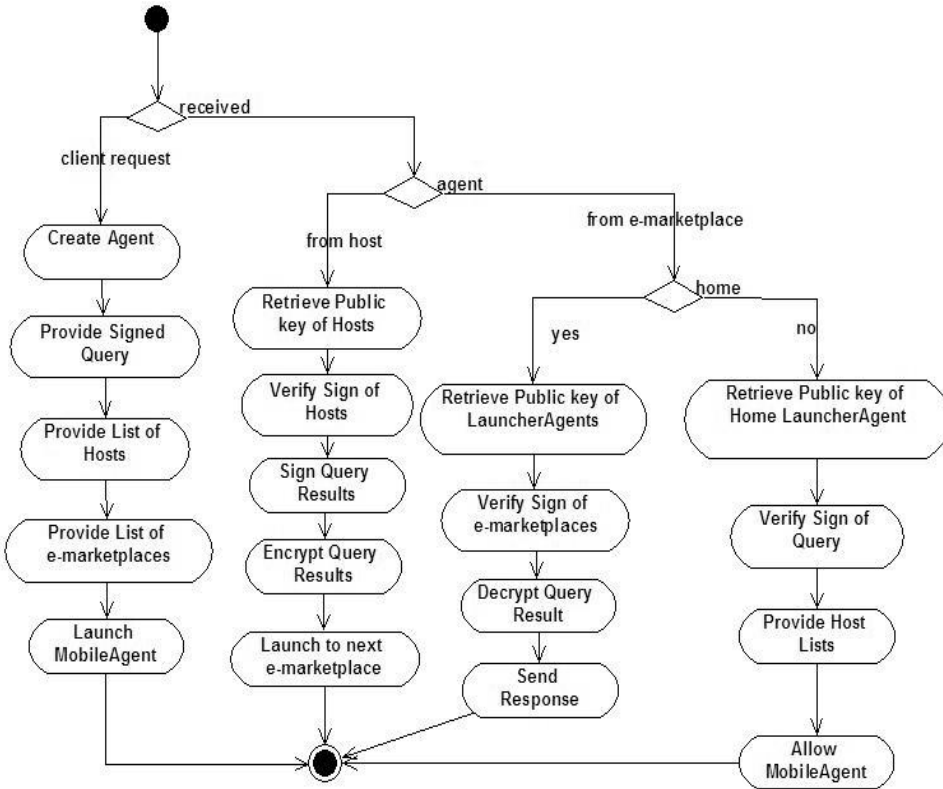
The LauncherAgent present in the e-marketplace that receives the MobileAgent from a host present in the same e-marketplace is responsible for:

- Verifying the signature of the query using public key of LauncherAgent of home e-marketplace
- Verifying the signature of query results provided by each of the hosts
- Encrypting the query results by public key of MobileAgent
- Launching the MobileAgent into the next e-marketplace

The different types of activities performed by the LauncherAgent in the e-marketplace according to different situations are depicted by Figure 4.

The host (RemoteAgent that resides in the host) that receives a MobileAgent is responsible for:

- Retrieving the results by querying its own database
- Signing the query result using its own private key
- Encrypting the query result using the public key of LauncherAgent present in its own e-marketplace



**Figure 4. Activities of the LauncherAgent**

In this model, the services provided are: trusted domain management, security policy management, and inter-marketplace authentication. Each of the marketplaces is authenticated and each of the hosts present in the e-marketplace is in the trusted domain. Hence, the MobileAgent cannot migrate to an agent system or a host that is not trusted. As this distributed model executes domain-level security operations and enables the MobileAgent to migrate among nodes using domain-centric itinerary, it can provide higher performance than the secured SCAS. Also, this model is more scalable as the bottleneck at the centralized KeyServer is updated to a distributed one. The proposed security mechanisms are evaluated using performance models.

**5.3. Performance Model for Secured and Distributed SCAS**

A performance model has been developed for the secured and distributed SCAS in order to evaluate the impact on system performance introduced by the proposed solutions for the security problems. The parameters selected for this case study are based on the workflow given in the previous section, and the

parameters required for this model, in addition to those given in Table 1, are summarized in Table 2.

**Table 2. Additional Parameters for Evaluating SDSCAS**

Name	Description
$T_{SDTh}$	Total execution time for MA to complete the task
$C_{AP}$	Number of agent platforms (e-marketplaces)
$T_{SDAuth}$	Time required for authentication
$T_{SDSec}$	Time required for providing security
$T_{itnry}$	Time required for providing itinerary to MA
$T_{SDMA}$	Total execution time in SDSCAS

The total execution time of a MobileAgent in SDSCAS amounts to the time for the information retrieval and the time for security service. Whenever the MobileAgent visits an e-marketplace to perform its task, authentication service is done twice by the Main Container of the agent platform: first, to check the authenticity of the MobileAgent before beginning the task, and, second, to check the authenticity of participating hosts at the time of completion of the assigned task at the e-marketplace. The hosts present in that e-marketplace assume that the MobileAgent is safe because it is authenticated by the domain controller (LauncherAgent present in the home site of e-marketplace), based on the predefined security policies.

The total execution time of MobileAgent to complete the assigned task,  $T_{SDTh}$  is given by the relation,

$$T_{SDTh} \leq T_{MAth} + (C_{AP} * (D_{mig} / R_{th} )) \quad (10)$$

The time required for authentication is given by

$$T_{SDAuth} = C_{AP} * T_{MAAuth} \quad (11)$$

The time for providing integrity and confidentiality,  $T_{SDSec}$ , is given by

$$T_{SDSec} \leq T_{MASec} + (C_{AP} * ((D_{code} / R_v) + T_{SDData} )) \quad (12)$$

The total execution time in the SDSCAS model,  $T_{SDMA}$ , consists of the time for searching the data, the time for authentication services, the time for providing

the travel plan, and the time for providing integrity and confidentiality. It is defined as

$$T_{SDMA} \leq T_{SDTh} + T_{SDAuth} + C_{AP} * T_{itnry} + T_{SDSec} \quad (13)$$

If the proposed model would provide higher performance than the centralized SCAS, then the following inequality must be satisfied.

$$T_{SDMA} \leq T_{SMA} \quad (14)$$

Expanding equation 14, using equations 13 and 9, yields:

$$\begin{aligned} T_{SDTh} + T_{SDAuth} + C_{AP} * T_{itnry} + T_{SDSec} \\ \leq T_{MAth} + (N + 1) * T_{MAAuth} + T_{MASec} \end{aligned} \quad (15)$$

When the number of e-marketplaces increases, the migration of the MobileAgent becomes frequent. Each time the MobileAgent migrates, the corresponding e-marketplace has to revise the travel plan for MobileAgent; therefore, the performance of this model degrades. In spite of that, the time for authentication also decreases, as on behalf of 'n' number of hosts, only one Main Container in the e-marketplace can authenticate the incoming MobileAgent. Hence,  $T_{SDMA}$  is improved more than  $T_{SMA}$ . This distributed model is appropriate to develop large-scale distributed information retrieval because such applications can very well satisfy equation 15.

This section describes a secured and distributed agent based model for large-scale distributed e-business environments. This model provides better performance than the centralized model; hence, it is suitable to meet the scalability requirements of large-scale real-time enterprise applications.

## 6. CONCLUSIONS

Deployment of agents in real-time distributed applications are hampered by the security issues explored with it. This paper aims to address security issues concerned with a distributed mobile agent-based e-business system.

Security requirements of a general e-business system – such as authenticity, integrity, confidentiality, and non-repudiation – are addressed using public key cryptographic mechanisms. Results show that a single solution is feasible to address all these requirements. The proposed solution introduces performance degradation when a greater number of hosts is involved. This is due to the large number of encryption and decryption computations involved. Hence, a tradeoff between security and performance is identified. A performance model is developed to evaluate the proposed solution according to the requirements of the implementation environment.

To avoid the bottleneck at the centralized key server in the secured SCAS model when it is scaled for distributed e-business system, a distributed model is proposed that improves the performance of the security model, simultaneously

offering better performance than the centralized model. Architecture for secured and distributed agent-based enterprise application has been designed with solutions to address the security requirements. The performance model of the centralized model is enhanced to suit the distributed model and both the models are compared for their relative performance. Results show that the distributed model is appropriate for the development of the large-scale real-time enterprise applications.

## REFERENCES

- Hohl, Fritz. 1998. A model of attacks of malicious hosts against MobileAgents, *In* Fourth Workshop on Mobile Object Systems (MOS'98): Secure Internet Mobile Computations, <http://cuiwww.unige.ch/~ecoopws/ws98/papers/hohl.ps>, 1998.
- Kannammal, A.; Ramachandran, V.; and Iyengar, N.Ch.S.N. Secure MobileAgent system for e-business applications, *Proceedings of 4<sup>th</sup> ACS/IEEE International Conference on Computer Systems and Applications*, Dubai/Sharjah, UAE, March 8-11, 326-329.
- Karnik, Neeran M., and Tripathi, Amand R. 1999. Security in the Ajanta MobileAgent System Technical Report, Department of Computer Science, University of Minnesota, May 1999.
- Lange, Danny B., and Oshima, Mitsura. 1999. Seven good reasons for MobileAgents, *Communications of the ACM*, 88-89.
- Sander, Tomas, and Tschudin, Christian F. 1998. Protecting MobileAgents Against Malicious Hosts. *In* Giovanni Vigna (ed.), *MobileAgents and Security*, LNCS 1419, 44-60. Springer.
- Sun Microsystems. Java Security Architecture. <http://java.sun.com/products/jdk/1.2/docs/guide/security/spec/securityspecTOC/fm.html>
- Tschudin, C. 1999. MobileAgent Security, *Intelligent Information Agents: Agent Based Information Discovery and Management in the Internet*, 431-446, Springer.
- You Eung-Gu and Lee Keum-Suk. 2004. A MobileAgent security management, *Proceedings of the 18<sup>th</sup> International Conference on Advanced Information Networking and Application (AINA '04)*.
- Zachary, J. 2003. Protecting Mobile Code in the Wild, *Internet Computing*, IEEE, 7(2), March-April.

## ABOUT THE AUTHORS

**A. Kannammal** is a faculty member in the Department of Computer Technology and Applications at Coimbatore Institute of Technology, India. She received her Ph.D. in computing science from VIT University, Vellore, India, in 2007. Her research interests include electronic business, agent technology, and Web services. She has published and presented papers in journals and at conferences. She serves as a reviewer for the International Journal of Patterns, Software Architecture and Software Use.

**N. Ch. S. N. Iyengar** is a professor in the School of Computing Sciences at VIT University, Vellore, India. He received his master of science degree in applied mathematics and his Ph.D. from the Regional Engineering College, Warangal (now known as NIT Warangal), Kakatiya University, Andhra Pradesh, India. He received his M.E. degree in computer science and engineering from Anna University, Chennai, India. His research interests include fluid dynamics (porous media), information security models, e-business applications, agent technologies, QoS in networks, ontology-based Web technologies, and cryptography.