New Computing Model for Securing Mobile Agents in IP Networks

Jean Tajer, Mo Adda and Benjamin Aziz
School of Computing, Portsmouth University, U.K.
{mo.add,benjamin.aziz}@port.ac.uk, up828996@myport.ac.uk

Keywords: Mobile Agent, Encryption, Trusted Nodes, Integrity, Performance Analysis.

Abstract: This paper deals with the prevention of security issues on mobile agents in IP Networks. We propose a new security computing model based on trusted server to avert Eavesdropping and Alternation attacks. The new protocol will be implemented using IBM mobile agent platform, Aglet. The new framework consists of components that provide support to the mobile agent while it is touring hosts in the agent space. It also protects the confidentiality and integrity of parts of the mobile agent. We conduct performance analysis over different types of mobile agents over a real IP Traces under malicious actions.

1 INTRODUCTION

Multi-Agent Systems (MAS) are designed using independent, autonomous known as agents which can perform their tasks independently or collectively in different types of environments. The agents can be considered as processes with the ability to perform an action on the environment on behalf of user. These systems allow distribution of complex tasks amongst agents. One of the basic properties of multi-agent system is its ability of self-organization which makes it utterly desirable for autonomous and flexible system designs such as graphical applications, logistics, transportation, search engines, network management etc .

Mobile Agent Systems can be divided based into programming language by which they are developed and use: Java and non-Java based. Around 85% of Mobile Agent systems available today are built using Java, due to its inherent support to Mobile Agent programming.

Mobile Agents are becoming a focus of modern research because of their applications in distributed systems which are replacing traditional client-server architectures rapidly. However, one of the key concerns in practical implementation of Mobile Agent is the lack of protection against any threats.

The rest of this paper is organized as follows. Related work is provided in section 2. Section 3 provides the security issues that a Mobile Agent can counter while visiting another host in the network. The proposed approach design is explained in section 4. In section 5, we present our experimental works and check the capability, reaction and performance of the mobile agents based on the developed design. Finally in section 6, we present the conclusion and our future work.

2 RELATED WORKS

Several researches have been conducted over mobile Agents.

Some Articles showed what exactly it is makes Java such a powerful tool for mobile agent development, also it highlighted some shortcomings in Java language systems that have implications for the conceptual design and use of Java-based mobile agent systems.

Other studies concentrate their work on the fault tolerance techniques in mobile agents, network management applications based on mobile agent technologies and how the fault tolerance techniques can improve their performance.

Other articles worked on an agent-based intelligent mobile assistant for supporting users prior to and during the execution of their tasks.

In addition, some works have been performed to integrate the mobile agents with the e-commerce. Some technical relevant issues are well presented.

Some researches concentrated their work on security concerns (i.e masquerading, denial of service, unauthorized access and repudiation) of mobile agents and how to protect them by several techniques like for example providing logical framework designed to support large-scale

heterogeneous mobile agent applications, on safe code interpretation, digital signatures, path histories, State Appraisal and Proof-Carrying Code (PCC).

In this paper, we will provide a brief about the security threats that a mobile agent can counter while hosting other hosts in the network. A new approach for preventing against these attacks will be proposed based on trusted server. It consists of components that provide support to the mobile agent to secure its mission in the agent space. It also protects the confidentiality and integrity of parts of the mobile agent. We will conduct performance analysis over a real IP Traces integrated with these attacks

3 SECURITY ISSUES AND COUNTERMEASURES

Security is one of the key factors of MAS. In fact, a MA is one of the potential threats to computer systems and vice versa, from the host system to the MAS itself. In this part, we will talk about the main security issues related to MAS.

The security threats for MASs could be divided as follows:

- Agent Protection: A remote host can threaten an agent; for example an untrusted host could execute an agent, observe its data and make some changes to them. Moreover, the threat could be from another agent in the domain, agent to agent threaten. Also unauthorized third parties threaten an agent. Through the agents communications a malicious entity may captures these messages and alter them.
- Host protection: An arriving malicious agent might threaten a host, the agent can access other data and files on the local host and might induce a serious damage. The other threat of the host is from unauthorized third parties; it is possible to send a host many spam agents that stress the host and take all its available processing power.
- Network protection: Incoming malicious agents threaten the network; in this case, the agents can clone themselves and flood the network with an excessive number of agents. Possible attack here is called denial of service attack (DOS).

There are many security services that can be used for securing the agents systems, for example; authentication, integrity, confidentiality and authorization.

In case of the authentication, the host needs to know the sender of the delivered agent. The agent authentication process includes verifying the entity that programmed the agent and also verifying the entity that dispatched it to the host. Basically, the agent and the host need to know with whom they are talking and dealing with, here the public-key encryption or passwords can be used.

For integrity, checking the integrity of the agents is a technique that makes sure no one has made any changes to the agents, the agents travelling form on host to another, and communicates and exchanges their data with other hosts and other agents. In this case, we need to make sure that the agents have not been tampered with in relation to their state, code or data. Moreover, the agents could carry different types of data, for example some private data. These data should only be readable from a specific host or agents. This technique is very important to avoid an eavesdropping threat.

The last service which helps to protect the agents and the hosts is authorization; the incoming agents should have a specific right to access the host information, so different agents have different authority, to protect the hosts and also to protect themselves.

4 NEW PROPOSED APPROACH

Our proposed countermeasure is based on trust. A mobile agent can host either blind folded, based on policy enforcement, or based on control and punishment.

A blind folded is the Mobile Agent which “simply need to trust its entertaining host”. The host is free to do whatever it wants while giving services to the Mobile Agent. But it is trusted that it neither has malicious behaviour nor collaborate with other hostile hosts that perform some bad actions on the agent.

Policy enforcement is another trust. In this case, the Mobile Agent and the host have a prior contractual relationship in the form of policy. Both parties need to sign for their rights and obligations.

The last trust is control and punishment. There is no needed policy to be signed between the two parties as well as there is no contract signed. The trust assumes that hosts are not by nature malicious and give them a chance to behave accordingly. But it still uses control mechanism to punish the host if found guilty of misbehaviours.

Our solution is a combination of policy enforcement and on control and punishment.

The below section outlines the guidelines used to develop the countermeasure.

4.1 Proposed Countermeasure

Mobile Agents are subjected to any type of attacks because they are a lonely figure once sent to the agent space. Hence, the proposition modifies the computing model of the mobile computation in order to address hostile host threats.

Figure 1 shows an overview of the mobile computing model. It is mandatory that the home or owner of the Mobile Agent has a public-private key pair at its disposal, the public key is published to the world so that the Mobile Agent could retrieve this key while it is visiting hosts. These keys are used by the security protocol to protect the confidentiality and the integrity of parts of the Mobile Agent.

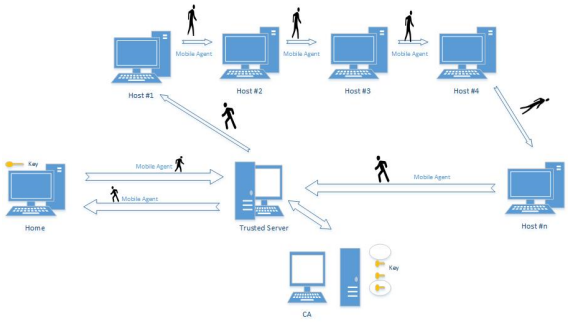

Figure 1: Proposed new model.

The proposal modifies the way by which the mobile computation is done. The arrows dictate that, the Mobile Agent first goes to the trusted node, creates a temporary storage element called active storage element (ASE), then moves to the first host to be visited. It goes there, sends the information it has retrieved from the corresponding host to be stored temporarily at ASE. The trusted node accepts the information and stores it. Each Mobile Agent that has a trust relationship with this node does the same, creates its own ASE at the trusted node and uses it to store the partial information it retrieves from each hosts. At the end, the Mobile Agent returns back to the trusted node and asks the corresponding ASE to hand it over the results it has been accumulating so far, carry back the result to its home as if it has been doing the job alone.

It is assumed that the agent space is divided into regions, within each region a node called trusted server is setup. These servers provide various services to the Mobile Agent while the agent is in the agent space. The Mobile Agent supported by these trusted nodes should be able to avert some of the evil acts from hostile hosts. The division of the agent space into regions is analogous to the cells in the mobile communication systems.

Nowadays, it is familiar to introduce trusted server setup in a network handled by a third party. Plenty of servers deployed in the internet word uses a trusted server like Web servers, mail servers and Domain Name Services (DNS) servers. Trusted servers do not have the right to modify the Mobile Agent's content, as web servers do not modify the web page they host. But here the security protocol provides further protection to the Mobile Agent content at the trusted server. It is such a similar concept that the proposition wants to exploit. The nodes and the trusted servers could be set up, in a similar style as nodes of root Web servers, by the Mobile Agent user community.

In sections to follow, we will take a look at the main components of the proposed countermeasure approach and how should the components interact according to the security protocol.

4.2 Components of the Proposed Countermeasure

Figure 1 shows the overall view of the proposal shows that the countermeasure constitutes various components at various degree of multiplicity. Below is the component of our proposed approach:

- Home of the Mobile Agent
- Mobile Agent (MA)
- Trusted Node
- Active Storage Element
- Home

4.2.1 Home of the Mobile Agent

It is the computer running Mobile Agent platform and has sent the Mobile Agent to carry out a task on its behalf. It can also be defined as a computer running a Mobile Agent based distributed application. The Mobile Agent after completing its task will return to the home carrying the result.

4.2.2 Mobile Agent

It is a program that migrates from one node to another node in a computer network to accomplish a given task.

4.2.3 Active Storage Element

It is a temporary storage element created by each Mobile Agent, at the trusted server. It actively participates in the process of temporary information

storage and handing over of all the information to the Mobile Agent.

4.2.4 Host

It is a computer in the agent space running Mobile Agent platform and entertains any visiting Mobile Agent which would like to gather information from it. This component is at the center of the controversy, which could be hostile. The host provides all the necessary resources for the agent to execute there.

4.3 Security Protocol

A security protocol defines how components of the system (Home, Trusted Nodes and Active Storage Element) should interact with each other as well as what are the necessary tasks need to be performed at each level in order to secure the network from any malicious actions.

The security protocol is free to take or alter any action on the Mobile Agent. While the Mobile Agent is travelling between its home and trusted nodes the usual composition is deemed. But when the agent is visiting different nodes, it is assumed to be composed of only the two out of the three components that is usually associated with: code and state, to give hostile hosts no chance of disclosure of information collected from previous hosts.

Another function of the security protocol is to develop a mechanism that lets the user of the Mobile Agent to digitally sign the list of destinations it wants the Mobile Agent to visit. After creating the list of destination, it digitally signs the destination object using its private key, then the destination object is passed down to the Mobile Agent. The Mobile Agent, upon its arrival at each host in the network, verifies that it has a valid copy of the destination object before putting that object into use. By this, the Mobile Agent avoids the possibility that it would be directed to visit other hosts by altering the list of paths it has carried from its home, as any malicious host could not forged the digitally signed destination object.

We can find below the security protocol in action in every step in the agent step. It is assumed that, the home node has a public-private key pair (HPubK-HPrvK).The public key could be retrieved by the hosts from relevant authorities.

4.3.1 At Home

The user of the MAS specifies the address of the list of hosts that will visit using the Agent Based Application (ABA).

The ABA takes the list and forms a destination object. The destination object contains the list of hosts to be visited, the address of the trusted server (TS) and the home. The ABA then digitally signs the destination object and passes it to the MAS. The MAS accepts the signed object. By using HPubK, the MAS verifies that it has the right un-signed destination object from which the address of the next node to be visited is determined and dispatches itself to that node, as pointed out in the previous section it goes first to the Trusted Node, Figure 2.

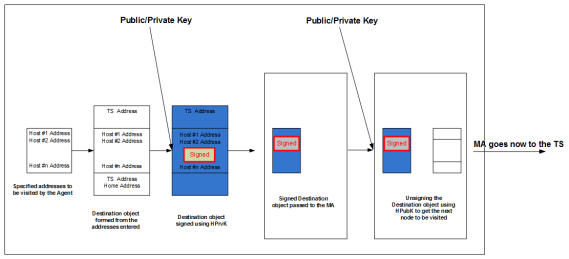

Figure 2: Proposed model at home.

4.3.2 At Trusted Server

The MAS arrives at the TS. It creates its own Active Storage Element (ASE). The MAS passes down the necessary information to the ASE so it can communicate with it. The MAS retrieves the public key of the home, HPubK. Using this key, the MAS un-signs the digitally signed destination object and determines the next node to be visited. In this case, it is the first host in the list and the MA Dispatches itself to that node, Figure. 3.

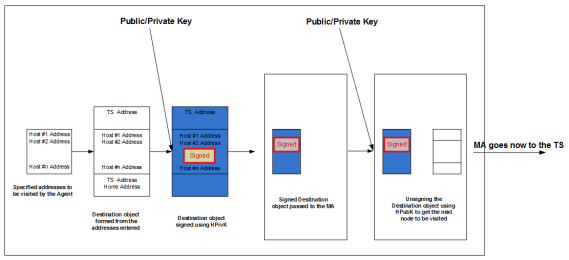

Figure 3: Proposed model at Trusted Server.

4.3.3 At Trusted Server, After the Reach of the Nth Host

The MAS arrives at the TS. It asks the corresponding ASE to hand it the information it has been accumulating (a pair of HPubK(SymK_i) and SymK_i(Info_i) retrieved from each host), and takes these information. After un-signing its destination

object, it looks for the address of the next node to be visited. In this case for sure it is the home node and the MAS dispatches itself to its home, Figure 4.

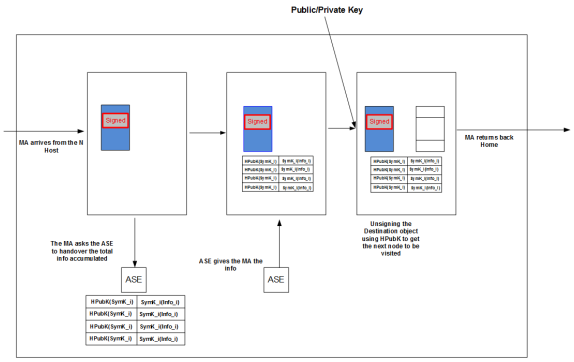

Figure 4: Proposed Model at Trusted Server, after the reach of the Nth Host.

4.3.4 At Home

The MAS arrives back at home after completing its mission. The MAS contains a pair of encrypted information. The MAS hands the overall information to the ABA. For each pair of encrypted information retrieved from each host, the ABA does the following:

- First, using its private key (HPrvK) to decrypt the encrypted symmetric key, HPubK (SymK_i) ,
- Second, using the decrypted symmetric key, SymK_i, it decrypts the information which is encrypted using the same key, SymK_i (Info_i),

The ABA does the same process for each pair of information retrieved from every host the agent goes to collect information. At last the ABA displays the result to the user, Info_i.

4.4 Security Protocol Summary for N Hosts

You can find below a summary for the security protocol:

- N hosts addresses digitally signed by the home node.
- One ASE created at TS.
- N symmetric random keys generated at each host.
- N information retrieved will be encrypted by the corresponding N symmetric keys.
- The N symmetric keys will be encrypted by the public key (RSA) of the home.
- The encrypted N information and encrypted keys stored at the ASE.
- Decryption at home node and displaying the plain text result to the user.

5 EXPERIMENTALS RESULTS

In this section, we present the result of the implementation of the security policy as well as the result of the performance comparison between different types of mobile agents.

The following techniques and tools are used: Two workstations with 8 GB and 768 MB of RAM respectively, which run Windows Server 2003 and a number of Mobile Agents (Proposed MA, DS MA and Normal MA) as described below, are used.

- *Normal Mobile Agent (Normal MA):* An Agent that executes mobile computation in the usual way
- *Proposed Mobile Agent (Proposed MA):* A Mobile Agent which is directed by the security protocol mentioned in section 4.
- *Digitally Signed Mobile Agent (DSMA):* A Mobile Agent that supports digital signing of the destination object while still performing computation.

We have considered the above describe mobile agents will have to execute the similar path.

To measure the capability of the proposal towards eavesdropping threat, a test environment is set up using the above mentioned computers as shown in Figure 5. Computer A is considered to act as trusted server (TS) and computer B runs many host nodes simulated through various port numbers as well as the home node in a virtualized mode. Wireshark Network Packet Analyzer will be running regularly over computer A. its job is to capture, sniff packets in a network and store them.

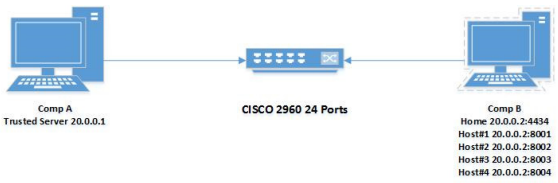

Figure 5: Experimental Lab.

The Figure 6 shows analysis of the packet captured while the Normal MA and DS MA are in execution.

As it is shown, in either cases it is possible to eavesdrop what information is retrieved and exchanged at each host: **"OS Architecture: x86; OS Version: 5.2"**.

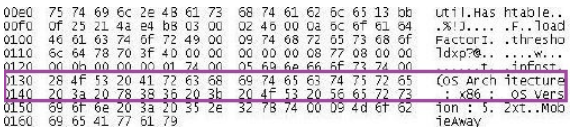

Figure 6: Captured packet analysis for DS MA and normal MA.

Figure 7 shows analysis of the captured packet while the Proposed MA is in operation. As it can be seen from the figure, unlike the above case since the information is sent to the TS in encrypted form, it is not possible to look into its content. Hence, the security protocol provides the required confidentiality of the information while it is being stored at ASE.

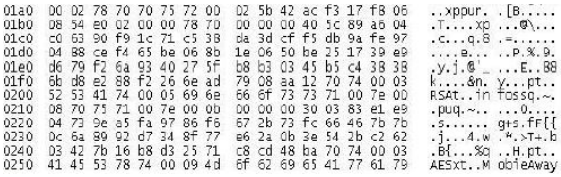

Figure 7: Captured packet analysis for Proposed MA.

A different type of attacks is performed: Alteration threat. The similar test environment as in the above case is used, except that all of the nodes are simulated in computer.

A malicious node is interfering on a different port number in the Computer B. This node is planned to behave maliciously towards the Proposed MA; its goal is to supply a wrong public key to the MA as it arrives there and is in the process of un-signing its digitally signed destination object. But fortunately the MA cannot un-sign the signed object using the public key just supplied. This is because the destination object is signed by the private key of the home node, not by a private key which corresponds to the public key supplied by the hostile node. Therefore, any attempt of alteration of destination object will be detected by the MA.

Performance Comparison:

To measure the performance of every mobile agents (Normal, DS, Proposed MA) a similar test environment as above is used. Their performance is compared in terms of their average turnaround time, measured in milliseconds (ms).

This performance parameter is the average time in milliseconds (ms) each Mobile Agent requires to do the job, after dispatched till it returns and handovers the result to the user.

As expected DS Mobile Agent takes in between of the two. Comparing the execution time of the Normal MA with the Proposed MA, the Proposed MA needs approximately 5x more time as shown in Figure 8. This substantial amount of time is a price to pay to achieve the corresponding security: Generation of the keys, encryption of partial information, verification of destination object at each visited host and at last collecting the results back to the Mobile Agent from the TS all add up to form a big turnaround time.

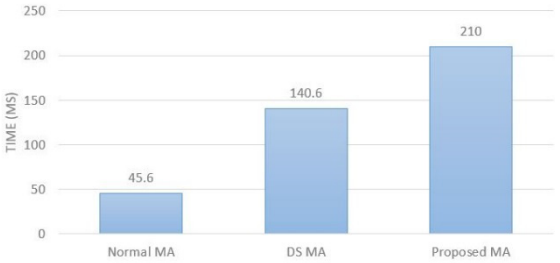

Figure 8: Performance Comparison Between Different Types of Mobile Agents.

Comparing the performance time between DS MA and Proposed MA, the DS MA needs less time. This is due to the fact that DS MA does not carry out some of the functions the Proposed MA performs like: Generation of keys, Encryption. It takes time since it verifies that the destination object is valid copy on its arrival at each and every host.

Figure 9 compares the execution time for all Mobile Agent cases. As the number of nodes to be visited is steadily increased, we notice that the turnaround time increases. This is tribute to the fact that there are more jobs to be done.

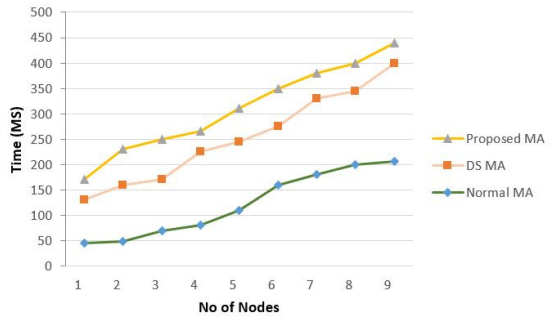

Figure 9: Performance time trend as the number of nodes visited increases.

6 CONCLUSIONS

In this paper, we proposed a new framework based on trusted server and developed by IBM platform Aglet to prevent security issues over mobile agents. The proposed approach evaluated how different types of mobile agents can react based on real traces with attacks. Our experimental results show that the proposal mobile agent needs approximately 4x more time than a normal agent to execute its job. This is cause of the Generation of the keys, encryption of partial information, verification of destination object at each visited host and at last collecting the results back to the Mobile Agent from the TS all add up to form a big turnaround time. Indeed, we compared the execution time for all Mobile Agent cases. As the number of nodes to be visited is increased, we notice that the turnaround time increases. This is tribute to the fact that there are more jobs to be done.

In our future work, we will focus on detecting flooding attacks over mobile agents. We will propose a new framework for the detection of flooding attacks by integrating Power Divergence over Sketch data structure. The performance of the proposed framework is investigated in terms of detection probability and false alarm ratio.

We also intend to provide a method for reducing the amount of monitoring data on high speed networks, and to analyze the impact of sampling on the precision of this divergence measure.

REFERENCES

HU, Jiang-Ping, Zhi-Xin LIU, Jin-Huan WANG, Lin WANG, Xiao-Ming HU. "Estimation, Intervention and Interaction of Multi-agent Systems." *Acta Automatica Sinica 39*, no. 11 (2013): 1796-1804.

Umar Manzoor, Samia Nefti, Yacine Rezgui, "Categorization of malicious behaviors using ontology-based cognitive agents", *Data & Knowledge Engineering, Volume 85*, May 2013, Pages 40-56.

Umar Manzoor, Samia Nefti, "iDetect: Content Based Monitoring of Complex Networks using Mobile Agents", *Applied Soft Computing, Volume 12*, Issue 5, May 2012, Pages 1607-1619.

Chen, Bo, Harry H. Cheng, and Joe Palen. "Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems." *Transportation Research Part C: Emerging Technologies 17*, no. 1 (2009): 1-10.

Maria Zubair, Umar Manzoor. "Mobile Agent based Network Management Applications and Fault-Tolerance Mechanisms", The Sixth International Conference on Innovative Computing Technology (INTECH 2016)

Mouhammd Alkasassbeh, Mo Add. "Network fault detection with Wiener filter-based agent", *Journal of Network and Computer Applications 32*(4) (4):824-833 · July 2009

Talal Rahwan, Tarek Rahwan, Iyad Rahwan, and Ronald Ashri. "Agent-based Support for Mobile Users using AgentSpeak(L)", *Agent-Oriented Information Systems Volume 3030 of the series Lecture Notes in Computer Science* pp 45-60

Tu, Griffel and Lamersdof. "Integration of intelligent and mobile agent for E-commerce"

Ryszard Kowalczyk, Mihaela Ulieru and Rainer Unland. "Integrating Mobile and Intelligent Agents in Advanced e-Commerce: A Survey)", *Agent-Oriented Information Systems Volume 3030 of the series Lecture Notes in Computer Science* pp 45-60

Jansen W. and Karygiannis "T. Mobile Agent Security", *National Institute of Standards and Technology*, Gaithersburg, MD 220899.

Rahula Jha, "Mobile Agents for e-commerce", *M. Tech. Dissertation*, IIT Bombay, India, 2002.

Altalayleh, M. and Brankovic, L. "An Overview of Security Issues and Techniques in Mobile Agents", University of NewCastle, Australia.

Danny B. Lange, Mitsuru Oshima. "Mobile Agents with Java: The Aglet API", September 1998, Volume 1, Issue 3, pp 111–121

Sun: Java 2 SDK security documentation. (2003).

Guido J.van 't Noordende, Frances M. T. Brazier, Andrew S. Tanenbaum. "Security in a Mobile Agent System", 2004*, IEEE Symposium on Multi-Agent Security and Survivability*

Michelle S. Wangham, Joni da Silva Fraga, Rafael R. Obelheiro. "A Security Scheme for Agent Platforms in Large-Scale Systems", 2013*, IFIP International Conference on Communications and Multimedia Security Mobile*, pp 104-116

Gray, R., Kotz, D., Cybenko, G., Rus, "Security in a multiplelanguage, mobile agent systems*". LNCS 1419*. Springer-Verlag (1998)

Karnik, N. "Security in Mobile Agent Systems". *PhD thesis*, University of Minnesota (1998)