Computer Network Management & Applying Web Services with Mobile Agent

Prashant P. Rewagad **Gaurav Govind Rakhonde**

Abstract — **In a very simple form a Mobile Agent is an independent piece of code that has mobility and autonomy behavior. One of the main advantages of using Mobile Agent in a network is -it reduces network traffic load. The Mobile Agent technology invented to overcome the complexity resulting due to the increasing size of network components raises new network management schemes. Many prototype applications providing Mobile Agent capability have been proposed for being used in network management. E-commerce and information retrieval are some of them. The motive behind the agent mobility is that, it addresses some limitations faced by traditional centralized client-server architecture, which are mainly, minimizing bandwidth consumption, supporting network load balancing, enhancing scalability as well as flexibility, increase fault tolerance and solve problems caused by unreliable network connections. In this paper we present network management, propose a net patrol model for applying web services with the help of mobile agent and propose a solution to protect mobile agent with threshold cryptography in ad-hoc network**.

Key Words — **Ad hoc network, Mobile Agent, Security, Threats, Threshold Cryptography.**

I. INTRODUCTION

Broadly speaking an agent is any program that acts on behalf of a (human) user. A Mobile Agent, then, is a computer program that is capable of migrating autonomously from node to node, across a heterogeneous network, to perform some computation on behalf of the user. Applications can inject Mobile Agents into a network, allowing them to roam the network, either on the predetermined path or one that the agents themselves determine based on dynamically gathered information. Having accomplished their goals, the agents can return to their home site to report their results to the user. In managing networks we can achieve more benefits from the use of Mobile Agents due to the fact that, the approach is based on a decentralized computation. The comparison between Mobile Agent performances with traditional centralized approach based on Simple Network Management Protocol (SNMP) show that SMNP does not scale well when the size and complexity of the network increases Today SOA (Service Oriented Architecture) is a fast emerging successor of the Object Oriented and Distributed Object Oriented paradigms. WS (Web Services) Technology is an implementation of the SOA Model. Since 'The Web' today is omnipresent, 'Web Services', which are Services offered on the Web, enable ubiquitous as well as distributed processing.

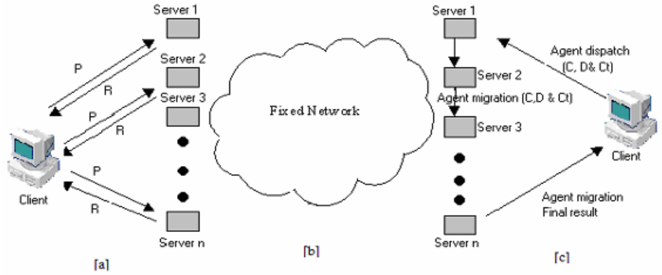
Both Mobile Agents and Web Services are Distributed Computing paradigms, and are well suited for a domain such as NM (Network Management) which is innately distributed in nature. Though there is a lot of published research material available, where Web Services and Mobile Agents are used independently for NM, there is very few research outcomes published which focuses on the convergence of these two technologies.

A. Mobile Agent

A mobile agent is an independent piece of code with autonomy and mobility features [4]. A mobile agent can migrate from one host to another autonomously to resume its execution [5]. Here autonomy means to take decision and to execute an action without direct user or human interaction. One of the main advantages of using mobile agent is – it reduces network traffic load noticeably as it does not require any continuous connection or communication between the server and the client [6]. Besides, use of mobile agent makes access of remote resource more efficient and flexible. Agent can adapt dynamically to an environment and can operate in heterogeneous environments.

B. Comparing Mobile Agent With traditional SNMP System

The following figure shows the comparison between tradional SNMP protocol and Mobile Agent.



P: Parameter, R: Results, (C, D & Ct): Code data and context

Figure 1*:* Comparison between SNMP vs. Mobile Agent paradigms. (a) SNMP Paradigm (b) Fixed Network (c) Mobile Agent paradigm

In section 2 we present the agent characteristics. Advantages and problems of Mobile Agents are also discussed in this section. Section 3 tackles the theme of the paper i.e. proposes the Mobile Agent Management infrastructure and shows how the major functional components are implemented. Proposed Net patrol model is discussed in section 4. Section 6 presents proposed model for security of Mobile Agent in ad hoc networks using Threshold Cryptography. Conclusive remarks are given in section 7, followed by a list of references in section 8.

II. AGENT CHARACTERISTICS

Mobile Agents may possess several or all of the characteristics [09, 10] summarized below in Table 1. Every agent satisfies the first four properties (i.e. reactive, autonomous, goal oriented, and temporary continuous). Other properties are defined on adding hierarchical classification.

Table 1: Characteristics of Mobile Agent

Autonomy	Exercise control over its own actions
Temporary continuous	It is a continually running process
Reactivity	Respond in a timely fashion to changes in the environment
Goal oriented	Does nor simply act in response to the environment
Proactively	Able to change event and make things happen
Social ability	Can communicate and collaborate
Mobility	Able to transport itself from one machine to another
Cognitive	Able to learn and adapt environment
Flexibility	Characterized by capability to adapt changing environment

A. Problems With Mobile Agent

Several Mobile Agent systems have been proposed [1]. However, the technology is still not yet widely accepted, due to the fact that there are still several issues to be solved. Major problems associated with Mobile Agent include:

- *Coordination:* One of the fundamental activities in Mobile Agent application is coordination between agents and entities they encounter during execution. The mobility of an agent raises problems. Multiple agents are likely to visits the same site at the same time. It is forbidden several agents attempting to have access to the same resource at the same time, as this might lead to deadlock. For the agent-based application to be successful, coordination between agent and network components is an issue that needs to be well addressed.
- *Resource management:* Since agents are autonomous congestion during resource access is inevitable. Resource allocation of agents must be governed in order to avoid congestion or system breakdown.
- *Security:* The introduction of mobile code in a network raises several security issues. In open network such as Internet, servers run the risks of system penetration by malicious agents, as these can cause undesirable consumption of resources. On the other hand parts of the agent states might be sensitive and might need to be kept secret when they travel on the network. Security bleach could result in the modification of the agent's code as it traverses the network. Researches admit that protecting the agent against hostile hosts and vice versa are still a difficult issue [7] further points out that security is the major obstacle preventing the wide spread acceptance of the Mobile Agent paradigm. In Section we propose noble mechanisms for protecting the agent's code and their hosting sites of execution.

B. Security Issues with Mobile Agent

Security of mobile agent is essential in any mobile agent based application. Besides security of agent platform is also important. To discuss the security aspects of a mobile agent system we have considered the following security services: Confidentiality, Integrity, Authentication, Authorization and Non-Repudiation.

- *Confidentiality:*

Confidentiality ensures that, data and code carried by an agent is not accessible by unauthorized parties (unauthorized agent or unauthorized agent server).

- *Integrity:*

Integrity guarantees that agent's code and baggage cannot be altered or modified.

- *Authentication:*

Authentication enables a mobile agent to verify its identity to an agent server as well as an agent server to a mobile agent. Without authenticity an attacker could masquerade an agent's identity and could gain access to resources and sensitive information.

- *Authorization:*

Authorization ensures that an agent can access the resource or information only those are allowed for it to access.

- *Non-Repudiation:*

Non-repudiation assures that the agent-server or the mobile agent cannot repudiate the actions it has performed.

III. PROPOSED MOBILE AGENT MANAGEMENT ARCHITECTURE

In this section we propose an infrastructure that provides framework for network management functionality and code mobility. The architecture emphasizes autonomy and mobility of agents. The entire network is viewed as being made up of small and easily manageable groups of nodes called *Domains*. Based on this vision the infrastructure is equipped with two major entity categories namely, *management entities* and *managed entities*. The management entities include Domain Managers (DMs) and Manager of Managers (MoMs), all performing the management roles in a hierarchical level. The Domain Managers have control over the nodes within a domain, and MoMs directly exercise power on their immediate subordinates DMs, who's in turn exercise the powers delegated to them, to the managed entities. The managed entities include Agent Servers, Directory servers, Service/Resource agents and Mobile

Supporting Server, each performing specific function. To ease the management, domains consist of serves not more than a certain number. Likewise the MoM is bound to serve not more than certain maximum number of DMs. The grouping will largely depend on the geographical layout of the networks and the average load on the entire network. The registration of a node in a group may be done at the time anode is being installed, and a node can be shifted to another domain later as the network administrator may prefer. The mobile supporting server is used to enable the participation of the mobile user in communicating with the network components. The infrastructure is illustrated in figure 3.

A. *System Implementation*

In the proposed infrastructure, the managing entities provide an interface to the user to specify policies for Mobile Agent and dispatch the Mobile Agent. They also have the capability to create the Mobile Agent based on the information provided by the user. The travel action plan and security of the management information are specified at the Domain Manager before launching the Mobile Agent. When the Mobile Agent returns with the collected information, the Domain Manager processes the information and presents it to the user in a Graphical User Interface. In addition, a Domain Manager keeps track of the Mobile Agent, and it is ready at any time to service any special request from any of the managing entities.

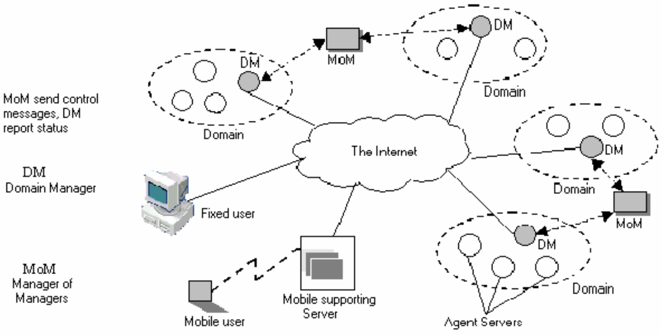


Figure 2*:* Mobile Agent Infrastructure

Agent Servers to execute in parallel, thus increasing system performance. Once the Mobile Agent is launched, the DM is available for other actions such as processing the received results, launching the new agent etc. The user at the user interface defines the policies before dispatching the Mobile Agent. We adopted, with slight modifications, the traveling plan proposed by [1] it consists of a list of nodes to be visited, potentially in a specified order. However, the order given at the launching entity is not mandatory; the intelligence of a Mobile Agent still enables it to make decisions based on the situations at any Agent Server. Intelligence can also be used in making decisions such as finding the next destination thus optimizing the travel plan. It can also be used to detect abnormal situations as the agent travel around the network. The security feature, discussed in section 5, provides the way to protect the agent, the information collected at the entity, as well as from other agent hosts or entities. The Agent server is designed to receive and execute mobile code. It thus provides execution environment (see figure 5), and it is responsible for receiving a Mobile Agent, authenticating the Mobile Agent and executing under local environment. It must also provide mechanism by which the local resources can be accessed. The Agent Server program specifies the policies that govern the Mobile Agent interaction with the local resources. At the same time the Agent Server has the authority to deny any service to the Mobile Agent that violates contract. The Domain Manager strictly specifies interactions between the local environments of the agent server. In order to effectively distribute processing load and control of management station, the hierarchical management approach has been adopted. To keep system manageable, the infrastructure provides only two level hierarchies as illustrated shown in Figure 3.

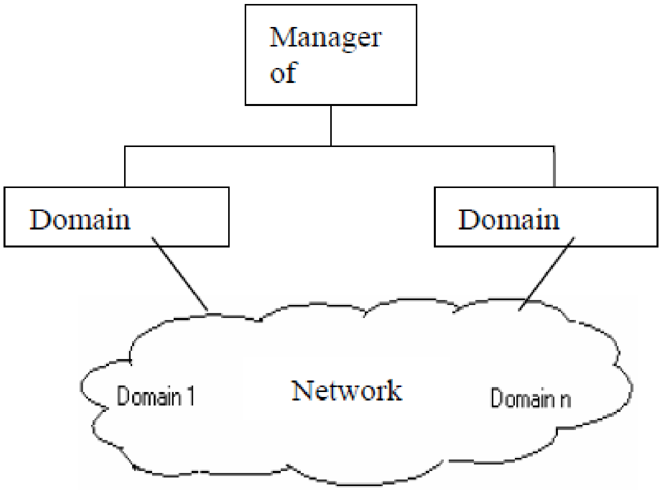


Figure 3: Two level hierarchy network

B. *Agent Server Implementation*

The agent server provides the resources, execution environment, as well as communication support for Mobile Agents. Every node in a Mobile Agent system must, therefore, be equipped with the Agent Server. The execution environment consists of Java interpreter called Java Virtual Machine, (JVM), which is a stand-alone platform. When the Mobile Agent arrives in the execution environment with the request to execute the server first performs security operations upon the agent to ensure its safety and regality; it is then instantiated and starts execution. After execution, the server provides storage support for the intermediate results. An Agent Server supports the transport mechanism with the help of Aglet Transfer Protocol (ATP). Aglets is a Java system developed by IBM, it enables the agents to move from one server to another by invoking special methods which execute automatically when the agent finishes execution and serving the context in a current host. In the proposed architecture, when the Mobile Agent wants to move first the Agent server packs it along with the context and encrypt the code for transit protection. When this is done the method *goTo* execute automatically, and thus enables the agent's code to move to the next destination.

C. *Life Cycle Of Mobile Agent*

In The proposed architecture a user creates the agent containing the request, mobile code, state information and other parameters. Other attributes include information about the Mobile Agent, such as the launcher, movement history, resource requirement, identification, authentication and encryption information. After creation, the user consults the manager for the agent launch. After arriving at the new hosts a Mobile Agent perform security and initializing process to activate itself. Once activated, the Mobile Agent collects the necessary information through interaction with local resources and performs execution. After the completion of all the specified tasks for that particular entity, the results are served into the container of the Mobile Agent. The agent will continue visiting nodes one after the other until the assigned task is completed. Figure 4 describes the Mobile Agent journey in its lifetime.

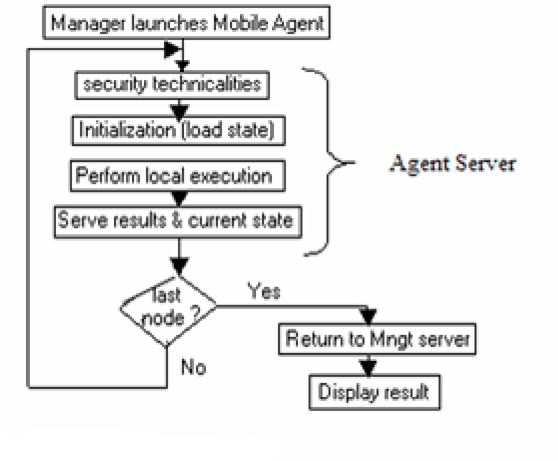


Figure 4: Mobile Agent Life Cycle

IV. Proposed Net Patrol Model For Applying Web Services With Mobile Agent

A. *Why the Name is Net Patrol?*

As mentioned in the Introduction Section, our research work aims to fill the gap, caused due to paucity of research outcomes where convergence of WSes with MAs and SNMP is used. The inspiration for this name came from a related work done, way back in 1995, by Zapf et al [8], which they call 'Net Doctor', that uses MA and SNMP to manage networks. In this context, we gave the name 'Net Patrol' to our framework, because, the Mobile Agents 'patrols' / 'tours' the network by migrating to the network nodes scrutinizing its Memory, CPU, Bandwidth Utilizations etc. The network Throughput is watched and prospective congestion segments are tracked and reported by overseeing the Packet Discard Rate at the NIC. As shown in Figure below, the framework broadly consists of three components namely:

i) The Manager Node.
ii) The Enterprise.
iii) The Managed Node.

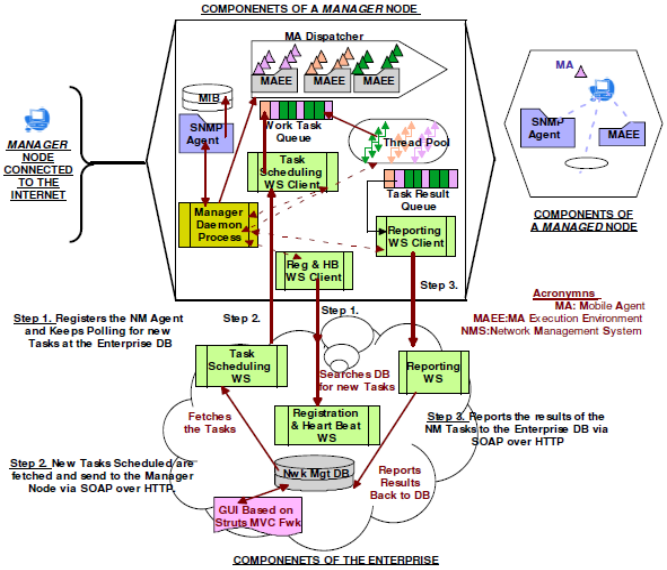

COMPONENETS OF A MANAGER NODE
MA Dispatcher
MAEE
MAEE
MAEE
MIB
SNMP Agent
Work Task Queue
Thread Pool
Task Scheduling WS Client
Task Result Queue
Manager Daemon Process
Reporting WS Client
Reg & HB WS Client
Step 3.
MANAGER NODE CONNECTED TO THE INTERNET
MA
SNMP Agent
MAEE
COMPONENTS OF A MANAGED NODE
Acronymns
MA: Mobile Agent
MAEE:MA Execution Environment
NMS:Network Management System
Step 1. Registers the NM Agent and Keeps Polling for new Tasks at the Enterprise DB
Step 2.
Step 1.
Task Scheduling WS
Searches DB for new Tasks
Reporting WS
Fetches the Tasks
Registration & Heart Beat WS
Step 3. Reports the results of the NM Tasks to the Enterprise DB via SOAP over HTTP
Step 2. New Tasks Scheduled are fetched and send to the Manager Node via SOAP over HTTP
Nwk Mgt DB
Reports Results Back to DB
GUI Based on Struts MVC Fwk
COMPONENETS OF THE ENTERPRISE

Figure 5: Proposed Net Patrol Model

B. *The Manager Node*

As shown in Figure 5, the Manager Node of 'Net Patrol' framework has a plethora of components working together. They are:

- A Daemon Process: It brings up the Agent.
- *MAEE* (*M*obile *A*gent *E*xecution *E*nvironment): *'Net Patrol'* used *Aglets* Framework.
- SNMP Agent: *'Net Patrol'* uses *AdventNet* SNMP API.
- *MIB* (*M*anagement *I*nformation *B*ase): Maintains state of network parameters of itself.
- Three SOAP Based WS Clients: Vital for Enterprise-Agent SOAP message exchange.
- A Pool of Threads: Made ready to execute the scheduled NM tasks, simultaneously.
- Result Queue: Maintains the task execution results until fetched by the Reporting WS.
- Set of library files developed by us to ease manager & managed node communication.

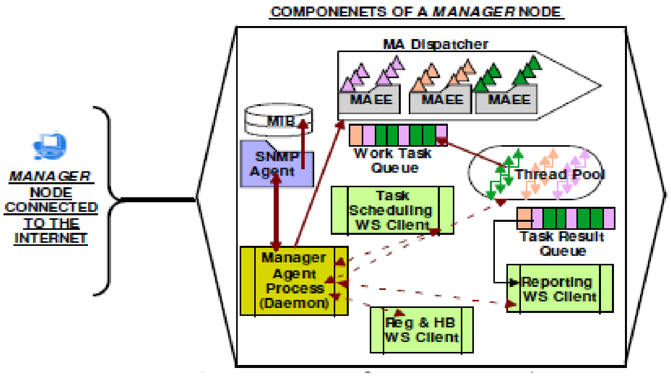

COMPONENETS OF A MANAGER NODE
MANAGER NODE CONNECTED TO THE INTERNET
MA Dispatcher
MAEE
MAEE
MAEE
MIB
SNMP Agent
Work Task Queue
Thread Pool
Task Scheduling WS Client
Task Result Queue
Manager Agent Process (Daemon)
Reporting WS Client
Reg & HB WS Client

Figure 6: Components of Manager Node

C.The Enterprise

Net Patrol’ uses a *Service Oriented Enterprise*, with the Web Services deployed on the Internet, thus enabling pervasive monitoring and management of the computer network. As depicted in Figure 7, the framework has three Web Services, namely

a) Registration and Licensing Web Service
b) Task Scheduling Web Service
c) Result Reporting Web Service

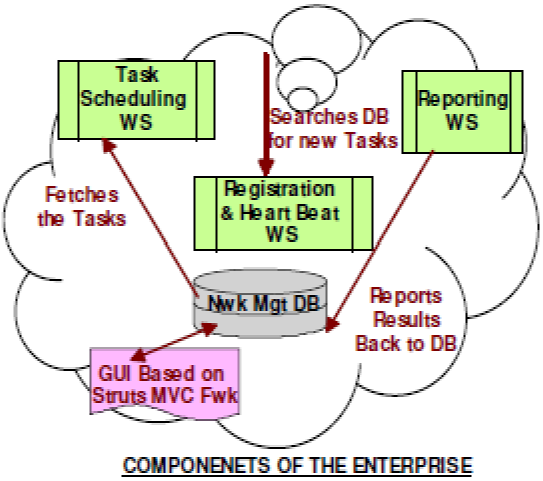

Task Scheduling WS
Searches DB for new Tasks
Reporting WS
Fetches the Tasks
Registration & Heart Beat WS
Nwk Mgt DB
Reports Results Back to DB
GUI Based on Struts MVC Fwk
COMPONENETS OF THE ENTERPRISE

Figure 7: Components of the Enterprise

The first thing the Agent Daemon process at the Manager Node does, when it starts up, is to invoke the Enterprise Registration and Licensing WS, where it registers itself with a Network Monitoring ID, which is unique to the entire enterprise database and has an Encrypted License Key. It is imperative to complete this Registration Process in order to ensure that the Enterprise recognizes it as a valid Agent. Agent validity is determined by checking the application’s contract expiry date with the Enterprise. If the contract has expired, the agent brings itself down, making the entire system un-usable, until a valid license is got.

We have designed the Enterprise User Interface so that the same can be deployed on a desktop web browser or a mobile device web browser. This is done, so that the Enterprise User who in this case is the network-admin, can access the backend NM application ubiquitously. To provide a unified view of the NM nodes and servers we provide an interface to the network-admin to schedule the routine NM tasks for the nodes of the LAN he is managing. Some of the typical tasks are Bandwidth Utilization, Throughput, Process Information, Memory Utilization etc. The Agent constantly polls the Enterprise Task Scheduler Web Service, to check if there are any NM that are assigned to it. The polling interval of the Agent is configurable. If there are such tasks, they are fetched by the Task Scheduler WS and using our WS execution framework these tasks are sent to the Task Scheduler WS Client integrated with the Agent. The results of these tasks are sent back to the Enterprise through the Enterprise Result Reporting Web Service.

D. The Managed Node

As in below figure 8, the Managed Node of *‘Net Patrol’* has the following components installed in it:

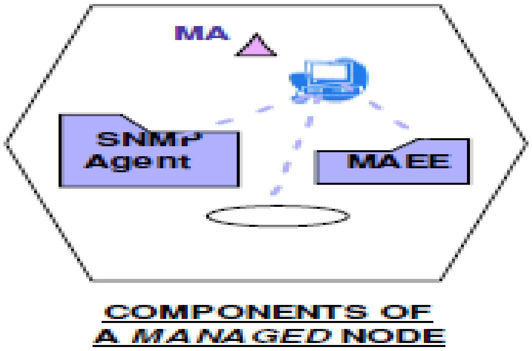


Figure 8: Managed Node

• MAEE (Mobile Agent Execution Environment): ‘Net Patrol’ used Aglets Framework.
• SNMP Agent: ‘Net Patrol’ uses Advent Net SNMP API.
• MIB (Management Information Base): Maintains state of network parameters of itself.
• Set of library files developed by us to ease manager & managed node communication.

V. PROPOSED SECURITY MODEL

To describe our model we will take help of the following figure 9 of an Ad hoc network. As we know that in an ad hoc network it is not possible to use a central server or a single point of trust, so trust should be distributed here among the available nodes. Distribution of trust in our proposed model is attained by using ‘Threshold Cryptography’. According to Threshold Cryptography to sign a certificate (for a service or server) there is a Master public/private key pair (Kpb / kpr) which is called the key pair of the Key Management Service. The master public key is known by all the Agent Servers and all nodes in the network trust any certificate signed by the master private key (kpr). According to Threshold

Cryptography to sign a certificate (for a service or server) there is a Master public/private key pair (Kpb / kpr) which is called the key pair of the Key Management Service [3]. The master public key is known by all the Agent Servers and all nodes in the network trust any certificate signed by the master private key (kpr)

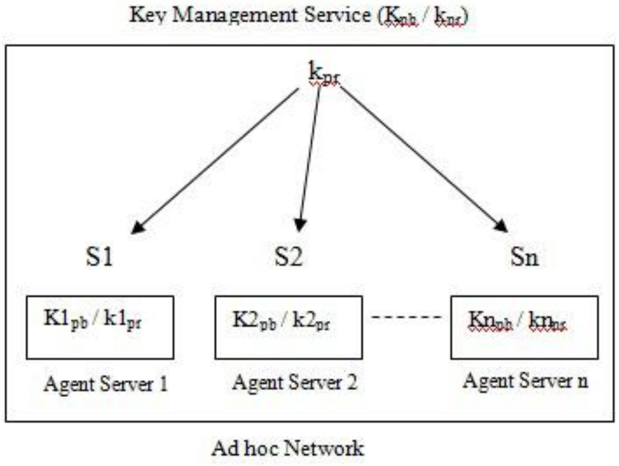


Figure 9: Mobile agent security in ad hoc network using Threshold Cryptography

This master private key is divided into n shares. Each server has one share of the private key (kpr). So S1, S2,S3…..Sn are the key shares for Agent-Server1 to Agent-Server-n respectively. Each Agent-Server has its own public-private key pair. Each server knows public keys of all other Agent Servers. As our model is based on (n, t+1) cryptography, where n >=t+1, here in the network n number of servers shares the ability to sign digital certificate and generate the corresponding private key of the Master Public Key (Kpb). Any (t+1) servers can perform this operation jointly. Here, *t* is the threshold value for the network and the system can tolerate up to *t* compromised servers. An important use of Mobile Agent may be to collect data from a network. For example, in a meeting people may want to share real time data among them. Here we consider that, the people in the meeting have established an Ad hoc network to get connected. We also consider that communicating devices (i.e. Laptop) they are using have the ability to process an Agent code, means agent platform is installed there. Now, assume that person 1 (Agent Server 1) wants to gather some data from other members and has launched a mobile agent for that purpose. That agent will move from one agent server to another, collect data and at last return with the collected data to the agent server who launched it. The whole processing or workflow can be described by the following steps:

1. At first Agent-Server1 launches a Mobile Agent. The agent traverses to Agent-Server2. Before launching the agent, Agent-Server1 calculates Message Integrity Code (MIC) of the agent code and digitally sign with its private key (k1pr). This digital signature provides authentication service. After signing Agent-Server1 encrypt the package (agent code plus digital signature) by the master public key (kpb). This provides confidentiality service.
2. Now when Agent-Server2 receives the full package it calculates the master private key by using its own partial share of the key with share of other t agent servers. After generating the key Agent-Server2 verifies it using Master Public Key (kpb). If any Agent Server is compromised and it provides incorrect key share then it will not be able to generate the correct private key (kpr). If it happens then Agent-Server2 tries another set of *t+1* shares. This process continues until Agent-Server2 gets the correct key.
3. After having the master private key Agent-Server2 decrypts the package which it has got from Agent-Server1. Then it verifies the signature of Agent-Server1 as it knows Agent-Server1's public key (k1pb). Then Agent-Server2 becomes ensured that the package is from Agent-Server1. Then Agent-Server2 calculates the MIC of Agent code and compare, thus check the integrity of the code. Then the Mobile Agent executes its operation on Agent-Server2.
4. After finishing its execution on Agent-Server2 the Mobile Agent moves to Agent-Server3. But before that, Agent-Server2 calculates the MIC of the agent code, data and sign digitally with its private key (k2pr). Then encrypt the whole package with the master public key (kpb).
5. This process repeats until the Mobile Agent finishes its tasks and return to the launcher agent server (Agent-Server1).

CONCLUSION

In this paper, a Mobile Agent-based management system has been discussed. The framework differs from most other Mobile Agent frameworks in that, it proposes a hierarchical level of management, which provides, to network components, smooth coordination and fault tolerance mechanism. It also provides an efficient way of locating the agent. This is achieved by having the agent creating a registration path as it passes though the servers. The approach enables the message being sent to trace the agent, easily follow the path and go directly to the agent. The spotlight of our work was to demonstrate the research outcomes of using an approach for Network Management combining two distributed computing paradigms, namely, Web Services and Mobile Agents. But, we did not want to bring in a totally new approach, replacing the tried and tested SNMP, a stable, sturdy, simple Network Management Protocol prevalent for more than two decades. In our hybrid 'Net Patrol' NMS framework, we have adhered to the wise old thought process that the charm of novelty should not obliterate the fact that it is unwise to change a working solution! Therefore, we employed a convergence of three techniques for Network Management, namely, SNMP, Mobile Agents as well as Web Services. This paper provides a solution for securing mobile agent in an ad hoc network. We have used Threshold Cryptography in our model, because it provides solution to the problem of central certificate authority (CA) and trusted third party in PKI, by distributing trust among several network nodes. Though it is tough to provide 100% security in an ad hoc network, to detect and prevent vulnerabilities and intrusions, use of mobile agent can play a tremendous role.

REFERENCES

[1] Mohammed A.M.Ibrahim "*Distributed Network Management with Mobile Agent Support"* 2006 International Conference on Hybrid Information Technology (ICHIT'06

[2] Mydhili K Nair and V.Gopalakrishna "*Applying web services with Mobile Agent for Computer Network Management*" International Journal of Computer Networks & Communications (IJCNC) Vol.3, No.2, March 2011.

[3] S.M. Sarwarul Islam Rizvi, Zinat Sultana, Bo Sun, Md. Washiqul Islam " *Security of Mobile Agent in Ad-hoc Networks using Threshold Cryptography*" World Academy of Science, Engineering and Technology 70 2010

[4] Niklas Borselius, *Mobile agent security*. [Online] Information Security Group, Royal Holloway, University of London. Available at: http://www.agent.ai/doc/upload/200402/bors02_1.pdf [Acces sed 1March 2010].

[5] Tarig Mohammad Ahmed, 2009." *Using Secure-Image Mechanism to Protect Mobile Agent against malicious Hosts"* [Online] World Academy of Science, Engineering and Technology.

[6] Mieso K. Denko and Qusay H. Mahmoud, *Mobile Agents for clustering and Routing in Mobile Ad Hoc Networks*, In : S. Pieere, M. Barbeau, E. Kranakis, Eds. 2nd International Conference on Ad Hoc Networks and Wireless (ADHOC-NOW'03), 8-10 oct. 2003, Montreal, Canada, Springer, 2003, pp. 271-276

[7] K. Hosson W.Gottfried R. Lunderer and B. Subbiah, "*A Mobile Agent Framework for Distributed Network management*", In IEEE proceedings, 1997

[8] Zapf M, Herrmann K, Geihs K(1999),Decentralized SNMP Management with Mobile Agents, *Proceedings of 6th Int. Symposium on Integrated Nwk Mgt*, Boston, MA, USA, 24-28 May, 623-635

[9] Lange D. B. and O. shima M. "*Seven good reasons for Mobile Agents*", in Communications, ACM, 42(3) pp. 355 – 395, 1999

[10] N. Karnik and Atripathi, "*Agent Server Architecture for Ajanta Mobile-Agent systems*", Proc. int'l conf. Parallel and Distributed Processing Techniques (PDPTA '98) CSREA Press, pp. 63-73, 1998.

AUTHOR'S PROFILE

	Prashant P. Rewagad Head of Department of Computer Engineering, G.H.Raisoni Institute of Engineering and Management, Jalgaon.
	Gaurav Govind Rakhonde Pursuing Masters Degree in Computer Science & Engineering from G.H.Raisoni Institute of Engineering and Management, Jalgaon and doing research work under guidance of Mr.Prashant P. Rewagad sir.