Evaluating AES and RSA performance in MANETs for Mobile Agent based Communication

Dissertation submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Information Security

Submitted By

Mandeep Kaur

(801533011)

Under the supervision of:

Dr. Sharad Saxena

Assistant Professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2017

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, **"Evaluating AES and RSA performance in MANETs for Mobile Agent based Communication"**, in partial fulfilment of the requirements for the award of degree of Master of Engineering in *Information Security* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Sharad Saxena* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Mandeep Kaur

Mandeep Kaur
801533011
ME (IS)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Sharad Saxena)
Assistant Professor, CSED
Thapar University, Patiala

ACKNOWLEDGEMENT

First of all, I would like to express my heartfelt appreciation to my supervisor Dr. Sharad Saxena for his constant guidance, great support, immense patience and valuable advice throughout my research at the Thapar University, Patiala. He helped me in finding research topics, proposing solutions and verifying results. Without his cooperative attitude, endless efforts and advices, this research would have never been possible. It has been a great honor and pleasure for me to do research under his supervision. I sincerely would like to thank Dr. Maninder Singh, Head of the Department, Computer Science engineering, Thapar University, Patiala for his support during my work.

Last but not the least I would like to thank my family, my colleagues and friends for their encouragement and love.

Mandeep Kaur
(801533011)

ABSTRACT

Mobile ad hoc networks (MANETs) are a type of wireless ad hoc network (WANETs). MANETs consists of peer-to-peer, self-forming and self-healing network. It is a collection of mobile devices which can form a temporary network if there is no fixed infrastructure provided.

The mobile hosts don't need assistance of base stations for communication with each-other. They can be initiated at any moment of time and organized quickly everywhere because there is no difficulty in setting up the infrastructure.

Due to lack of centralized architecture, security becomes the crucial aspect of ad-hoc network. The mobility of nodes also makes the network more prone to attacks. Security is a recent active topic for research in wired networks; even then, MANET has some nontrivial challenges that need a secure design. In order to deal with this issue, we have worked on mobile agent model.

Mobile agents are software modules that contain data and code with themselves. They are able to move autonomously from one host to other to perform tasks and automatically come back to the host where they were initialized. Although they have so many qualities such as dynamism, autonomy, flexibility etc. that are required for some disciplines yet Mobile Agent have some limitations mainly the security issues due to mobility. In our work, we have secured mobile agents' communication using AES and RSA algorithms in a MANET environment. The time taken to perform encryption and decryption by agents has shown the effective results using the proposed communication model.

Keywords: MANETs, Mobile Agents, Security, AES and RSA.

TABLE OF CONTENTS

Certificate……..i

Acknowledgement……..……..ii

Abstract……...……iii

Table of Contents…….…..……..iv

List of Figures…….…….…...vi

List of Tables……..…...vii

CHAPTER 1: INTRODUCTION

1.1 Introduction to Mobile Agents……..…...……2

1.1.1 Mobile Agent Architecture……..……...……...3

1.1.2 Features of Mobile Agents…….……...……4

1.1.3 Life Cycle of Mobile Agent…….…….…….5

1.2 Comparing Client Server Model with Mobile Agent communication…….6

1.2.1 Client Server Model……...6

1.2.2 Agent Based Model…….……..…….……7

1.3 Threats to Mobile Agents' Information……..…….……8

1.4 Objective of Dissertation……..……..…12

1.5 Research Methodology……..…13

1.6 Organization of Dissertation……..……13

CHAPTER 2: LITERATURE SURVEY

2.1 Available tools for Mobile Agent Development……14

2.2 Existing Security Mechanisms……...16

2.2.1 Platform Dependent……..……..…..16

2.2.2 Platform Independent……..…….……21

2.2.3 Dependent/Independent……..…….23

2.3 Issues Identified in Literature survey……..…..……26
2.4 Conclusion……..……...27

CHAPTER 3: PROPOSED WORK

3.1 Proposed Mobile Agent based Communication Model…….……28
3.2 Agent Development in JADE……29
3.3 Security Protocol Implementation…….……30
3.3.1 Agent Development and Communication……...…….……31
3.4 Data Encryption and Decryption……...……34
3.4.1 RSA Algorithm……...……35
3.4.2 AES Algorithm……...…….....36

CHAPTER 4: RESULTS AND DISCUSSIONS

4.1 Encryption……..……39
4.2 Decryption…….…..……...41

CHAPTER 5: CONCLUSION AND FUTURE SCOPE

5.1 Conclusion…….……46
5.2 Future Scope…….……47

REFERENCES…….…..……..…..……..…….48

LIST OF PUBLICATIONS……..……..……...53

VIDEO LINK…….…….....54

LIST OF FIGURES

Figure 1.1: Mobile Agent Communication……...……...2

Figure 1.2: Mobile Agent Architecture……..……...……...4

Figure 1.3: Life Cycle of Mobile Agent……..6

Figure 1.4: Comparison of Client Server and Agent Based Model…….7

Figure 1.5: Threat Categories……..9

Figure 2.1: Security Layers…….……...……16

Figure 2.2: Mobile Trust Model……..……...……17

Figure 3.1: Secure Agent Communication……...…..28

Figure 3.2: JADE Architecture……..29

Figure 3.3: Creation of Containers……...……..31

Figure 3.4: Creation of Vendor Agents……..32

Figure 3.5: Vendor-1 providing Book and its Price……...32

Figure 3.6: Vendor-2 providing Book and its Price……...33

Figure 3.7: Initialization of two Vendor Agents and a Customer Agent...33

Figure 3.8: Lowest Price Book selected by Customer Agent……34

Figure 3.9: Communication between Customer and Vendor Agents……34

Figure 3.10: AES Algorithm…….…….36

Figure 3.11: 4X4 Array of 16 Bytes……..37

Figure 3.12: Round Key Generation……..38

Figure 4.1: Encryption on Sender Side……..39

Figure 4.2: Encryption Time Comparison…….....41

Figure 4.3: Decryption on Receiver Side……...41

Figure 4.4: Decryption Time Comparison…….43

Figure 4.5: Time Delay Comparison…….……43

Figure 4.6: Throughput Comparison of AES and RSA…….44

LIST OF TABLES

Table 1.1: Attacks on each Layer in MANETs……..……12

Table 2.1: Toolkits and Their Features…….……..……15

Table 2.2: Various Techniques for Securing Mobile Agents……23

Table 4.1: Encryption time of AES and RSA for different file sizes …...41

Table 4.2: Decryption time of AES and RSA for different file sizes……43

Table 4.3: Comparison of AES and RSA……..46

ABBREVATIONS

WANET	Wireless Ad-Hoc Network
MANET	Mobile Ad-Hoc Network
WSN	Wireless Sensor Networks
RPC	Remote Procedure Call
REV	Remote Evaluation
DoS	Denial of Service
OSI	Open System Interconnection
AES	Advanced Encryption Standard
RSA	Rivest, Shamir and Adleman
ASDK	Agent Software Development Kit
FIPA	Foundation for Intelligent Physical Agents
JADE	JAVA Agent Development Environment
SSL	Secure Socket Layer
EEOS	Extended Elementary Object System
ERC	Extended Root Canal
MIP	Malicious Identification Policy
PMI	Private Management Infrastructure
CCAP	Code Change Authorization Protocol
BROSMAP	Broadcast Based Secure Mobile Agent Protocol

AMS	Agent Management System
DF	Directory Facilitator
ACC	Agent Communication Channel
JVM	JAVA Virtual Machine
IIOP	Internet Inter-Orb Protocol
GUID	Globally Unique Identifier

CHAPTER 1

INTRODUCTION

Wireless ad hoc networks (WANETS) are autonomous nodes that interact with one another via multi-hop radio network. In WANETS, connectivity is maintained in a decentralized way. Clients form different network topologies interact with one another without any interface such as wireless access points. The network is controlled by the nodes themselves because each node works as a host as well as a router.

WANET are categorized in two main types: MANET (mobile ad hoc network) and WSN (wireless sensor network).WSN are typically sensors that are distributed in an area. They cooperate with each other in order to fulfill a common goal. On the other hand, in MANET, each node works individually for its own. We have chosen MANET for our work because mobility of MANET increases the level of challenge of their security. Also, MANET doesn't have much resource constraints as compared to WSN. Therefore it's easy to run heavy applications in MANET, such as IDS.

In general, the routing process decides which intermediate node should be considered for delivery to destination node. In MANET, there are two steps for routing, i.e., Detection of route and Maintenance of route. Detection of route phase describes the route which should be followed for delivery of packets from sender node to receiver node. In route maintenance, any node in the network can detect if one or more than one link in the route have failed [1]. Any intruder can hamper route in order to pretend himself as some other node by giving fake routing information. Therefore both phases of communication should safe in order to provide security.

In wireless networks, each mobile device has a restricted area for transmission. These mobile devices act as nodes as well as routers. A

number of mobile devices are needed in order to transmit a packet before it reaches to final destination. WANET should be initiated at any moment of time and organized quickly everywhere because there is no difficulty in setting up the infrastructure. There are a number of applications of this type of networks such as military communications.

1.1 INTRODUCTION TO MOBILE AGENTS

Mobile Agents are software modules that move from platform to platform with its own data and code. After reaching to destination, the tasks are executed in the way they are meant to be. Mobile Agents communicate with each other when they move within the environment on the same network. An agent platform include the services such as; the capability for an agent to spawn or create new agents, terminate any spawned agents, clone itself, locate other agents at platform or at another platform, relocate itself on another platform and send messages to other agents.

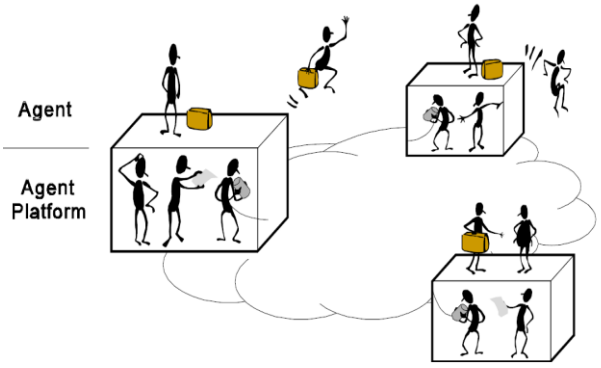

Figure 1.1: Mobile Agent Communication [2]

The platform of Mobile agents acts as distributed middleware. Therefore, platforms have the responsibility of sending, receiving, forming, executing, demolishing and transferring mobile agents. There are a number of models that explain the system of agents. In our work, we will use a simple approach for security purpose. It will include Agents, and Agent's Platform [3]. We assumed agents as a collection of program, control information and facts which is carried during execution on hosts they visit or reside. For agent platforms, we have assumed that it is the

environment where the agents reside and execute. The platform where agent is initialized is called the home platform. This is the most trustful and consistent platform. Mobile agents provide us the advantage of creating distributed applications. These applications results in overcoming network latency, reduction of network load, disconnected operations and faster interaction [4]. As mobile agents move from one platform to another for their execution, they are useful in case of applications that retrieve information from distributed environments. Some people use mobile agents to tackle with their monotonous and time consuming jobs.

Data Communication is also an appropriate application in case of mobile agents. In e-commerce this technology can be used to automate different stages that require a lot of time during the procedure of buying and selling. Mobile agents are customized and autonomous as compared to other "traditional" software's. Mobile agents can find products asked by user across every provider because of its ability to migrate within the network. While migrating, the exchange of information between mobile agent and the vendor in local way. Also, the exchange of information is not over network. Therefore they reduce overload and latencies in network. These special features of Mobile agents are helpful in optimization of selling and buying skills. This will also upraise the communication over the internet.

1.1.1 MOBILE AGENT ARCHITECTURE

Mobile Agents have become a promising technology in design, implementation and maintenance of Distributed Computing Systems, as they can travel form one node to another along with their internal state. Mobile agents are capable of performing tasks which need a lot of configuration. These days, mobile agents are widely used in number of applications because of their intelligence.

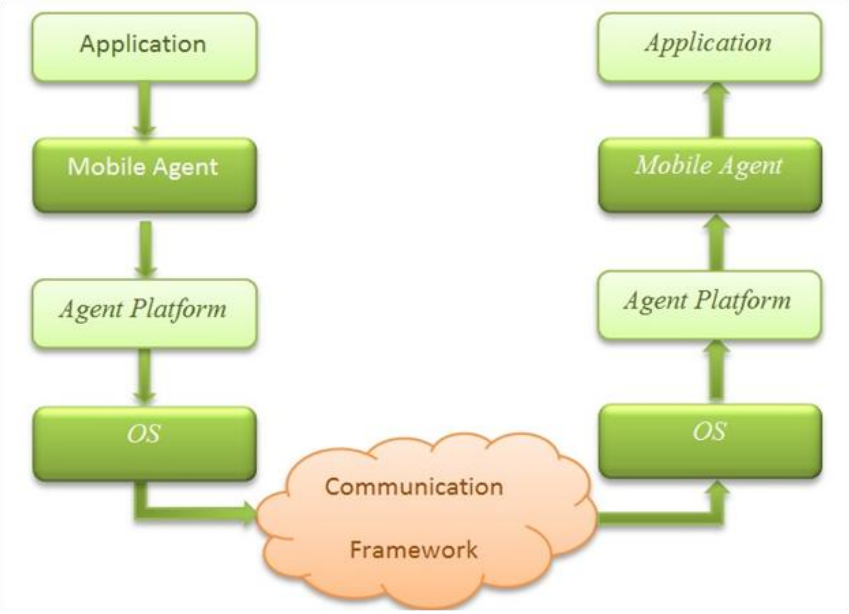

Figure 1.2: Mobile Agent Architecture

Mobile agents are software modules, contain data itself and able to move autonomously from one host to another to perform tasks and automatically come back to the host where they were initialized.

1.1.2 FEATURES OF MOBILE AGENT

Features of Mobile agent are as follows [5]:

- ✓ **Autonomous**: Mobile agents should be able to act according to their environment. They should have decision making ability, i.e., where to go, when to go and what to do. They should have the ability to control their states and information.
- ✓ **Proxy**: sometimes mobile agents act as somebody else for particular benefit.
- ✓ **Mobility**: It is the most important key feature of mobile agent. Mobile agent should be able to carry itself from one platform to another platform in a dynamic network.
- ✓ **Intelligent**: Mobile agents are capable of taking decision because of regress learning.
- ✓ **Proactive**: Mobile agent should be able to respond to the network and therefore should be goal-oriented.

- ✓ **Disconnected operation**: mobile agent should have the capability of working in the absence of network connection.
- ✓ **Cooperative**: Mobile agent should have the ability to synchronize themselves with other agents to fulfill a common goal.
- ✓ **Learning**: mobile agents should have the ability to accumulate knowledge. It is beneficial in case of decision taking.

  In mobile agent technology, the data is treated locally Instead of transferring Information over network. Hence they reduce network traffic which is very important for huge volume of data in real time systems.

1.1.3 LIFE CYCLE OF MOBILE AGENTS

There are set of events that a mobile agent can have during its lifetime which completes the mobile agent model. Common events of mobile agents are as Follows:

- ➢ **Creation** – creation of an agent directly Corresponds to an object's constructor. Event handler is responsible for initializing the state of agent. Further the event handler will also prepare the agent for more instructions that will be going to carry out.
- ➢ **Dumping** – Dumping of an agent is Similar to the object's destructor. All the resources which are being used by agent should be set free by the event handler during this event.
- ➢ **Transmit** – Transmitting the agent means Sending signals to plan for leaving present location and going to a latest one. This event can be initialized by any other agent who wants that agent to migrate form that location or agent itself can also initiate explicitly by a migrate request.
- ➢ **Arrival** – Here the migrated agent sends signals that it has arrived at new location successfully and performance of duties has been started.
- ➢ **Communication** – Communication basically means inter-agent connection. In this event, notifications of handling incoming messages are sent from other agents.

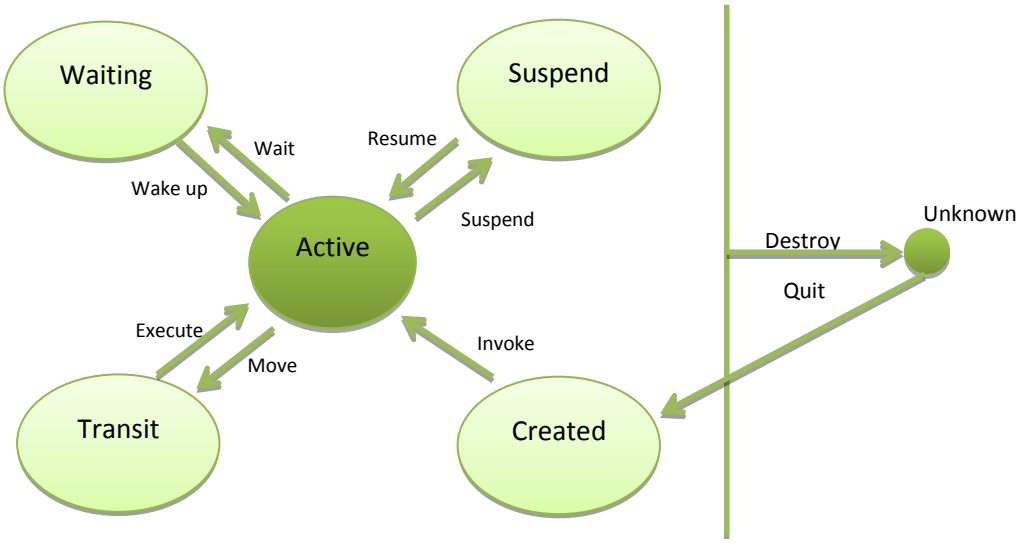

Figure 1.3: Life Cycle of Mobile Agent [6]

1.2 COMPARING CLIENT SERVER MODEL WITH MOBILE AGENT BASED COMMUNICATION

The communication in Client server and Mobile agent based environment is shown below:

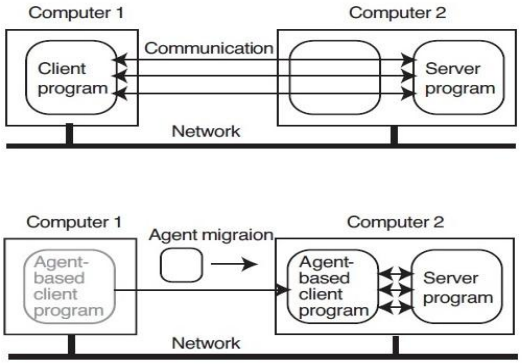

Figure 1.4: Comparison of client server and agent based model

1.2.1 CLIENT SERVER MODEL: An architecture in which a computer or a process sends request to another in a network is called a client server model. Here, the requesting computer or process is called a client and the computer to which it has requested is called the server. Basically the server serves the requests of the clients on its behalf. There are some methods to achieve communication between client and server such as RPC

(Remote procedure call), Message passing, and REV (Remote Evaluation). In RPC, the procedures are called through procedure call by the client. The client is called caller and sever is called callee. On Server side, the procedure is executed and the output of procedure is sent as a response. In remote evaluation, methods are downloaded directly for client by the server. In Client Server architecture, network based communication is classified as two processes that run over multiple processors, i.e., Front end and Back end. Front end and Back end work together by interacting with each other such that different applications are completed as single task. The client Server model doesn't proved to be good for many applications. Some of the disadvantages of client-server model are as follows [7]:

1. A large portion of application runs on server. Hence the client will use resources continuously. This can lead into a condition of resource congestion.

2. Composite methods are needed for back end and front end procedures because of lack of development tools compatibility.

3. Some network operating systems are not much scalable. So there is a Lack of scalability.

4. There is a total dependence on server for fulfilling the request. When the server goes down the client cannot get his request fulfilled.

1.2.2 AGENT BASED MODEL: Mobile Agent Technology introduces a new method of communication where the agents migrate themselves from one node to another with their code and data. It has also overcome the limitations of client server model. In RPC, the procedures are called through procedure call by the client. The client is called caller and sever is called callee. On Server side, the procedure is executed and the output of procedure is sent as a response. In remote evaluation, methods are downloaded directly for client by the server. [7].

On the other hand, mobile agents carry the code and execution state with itself and hence, overcome the drawbacks of other approaches mentioned above. Mobile agent reduces network bandwidth, it requires low power requirements and also support for mobile units, provide low latency interaction.

Benefits of Mobile Agents are as follows:

- ✓ **Less bandwidth**: mobile agents do not require large bandwidth as bandwidth is needed only when the agent migrates form one place to another.
- ✓ **Support dynamic environment**: Mobile agents are developed in Java. Java is platform independent because of its byte code. That means it is independent of network and computer as byte code can run on any system with JVM installed.
- ✓ **Easier Development**: Mobile agents are basically distributed by nature; therefore it is comparatively easy to build a distributed system.
- ✓ **Efficiency**: Mobile agents require less network resource because movement is from computation to data instead of data to computation. Hence it increases efficiency.
- ✓ **Fault Tolerance and Robustness**: The agents can react dynamically according to situations. As a result, it boosts fault tolerance in highly complex distributed systems.
- ✓ **Support Data Communication**: Mobile agents can be used in e-marketing also.

1.3 THREATS TO MOBILE AGENTS' INFORMATION

A threat can be called as a possible danger in which vulnerability can be exploited in order to breach security and can harm the system. Threats to mobile agents are categorized into following 4 categories:

- ➢ **Malicious agent attacking host:** This type of attack occurs when the host or platform is vulnerable. This means that there is a possibility of attack in

absence of proper access control mechanisms and authentication. Attacks such as Denial of service (DoS) can take place by denying platform services and allowing computational resources.

- **Malicious host attacking agent**: The host can also be malicious. It can attempt to refuse requested services, corrupt agent's code, steal its data, reinitialize agent, return illegal system call values or even destroy the agent completely. It can also bluff the agent by retarding the agent until the task is no more applicable. The Host may also evaluate and invalidate the agent.

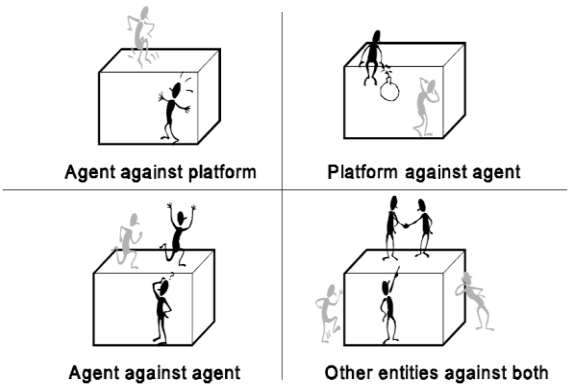

Figure 1.5: Threat Categories [2]

- **Malicious agent attacking another agent**: Sometimes malicious agent invokes public methods of another agent to hinder with its work.
- **Other entities against both:** Both outsider and insider entities of agent framework may try to harm, subvert disrupt the agent framework. The methods include attacking the inter-platform and inter-agent communications via masquerade. An attacker may also intercept messages or agents while transmitting and substitute other contents, modify their contents, or simply replay the transmission dialogue in order to disrupt the integrity of entire system.

Primary goals for security in ad-hoc network can be provided by authentication, confidentiality, integrity, non-repudiation, etc. Complete protection spanning should be provided to network. No single mechanism can provide all security service in MANET. The most common techniques are described below [8] [9]:

- **Confidentiality**: It guarantees that unauthorized person will never be disclosing certain information.
- **Authentication**: It makes sure of identity of the communicating node by enabling a node. To overcome authentication the attacker would imitate a node, and will gain unauthorized access to perceptive information and resources.
- **Availability**: Regardless of DoS attacks it guarantees the survivability. Attacker uses congestion techniques on physical and media access control layer to hamper physical channel of communication. The routing protocol can be interrupted by attacker on network layer. The attacker can get down the higher layers, high level services like key managing check.
- **Integrity**: The message transmitted in the network is never corrupted.
- **Non-impersonation**: The attackers uses fabrication that include fabrication message which is a fake routing message generated by attacker. Nobody else can act as if to be another authorized element to study any useful data. These attacks are hard to sense.
- **Non-repudiation**: The sent message by source can't be refused is ensured. Following are some of the attacks on ad-hoc network:

  • **Black Hole**: A malicious node is injected or inserted because of which bogus route replies are received for sent requests such that it advertises about having the smallest destination path [10]. Redirection of network traffic through malicious node is made up by false replies and is for eavesdropping. Denial of service which is a type of black hole attack attracts all traffic to it in order to execute DoS by dropping packets that are received.

  • **Location disclosure**: Attacker's target is the confidential necessities of an ad hoc network [11]. By using traffic analysis techniques, or by simple probing or monitoring, an attacker can find out the position of a node, or

even the formation of the whole network. It is known as location disclosure.

• **Blackmail**: For the recognition of malicious nodes the routing protocol that uses a method. Messages are transmitted that try to blacklist the criminal [12]. The attack floods the network with fraud or forged route with the help of malicious node. Packets are formed in order to use resources of participating nodes and disorder the genuine routes that are established. The sleep deficiency agony attack aims at the use of attacker that may formulate reporting message and try to separate nodes that are genuine from network. Non-repudiation security feature can be helpful in such cases as it combines node for the generated message, and hence is also known as blackmail attack.

• **Wormhole**: it is one of the most commanding attacks. Two malicious nodes collaboratively take part in the network [13]. Capturing routing traffic at one point of the network is done by one attacker, and channeled to another node of the network, if they share a personal communication link with each other. The traffic is tunneled back into the network by second attacker. The connectedness of the nodes that have created routes over the wormhole link is completely under the control of the two conspiring attackers. To get secured from wormholes we can use packet leashes.

• **Replay**: An attacker that performs a repeated attack into the network routing passage that has been previously captured. This attack usually targets the originality of routes.

• **Denial of Service**: Denial of service attacks is one of the most powerful attacks in ad-hoc network which aims at absolute interruption of the routing function and therefore the ad-hoc network perform intact operation [14]. Denial of service attacks' specific cases comprise of the overflow of routing table and the sleep deficiency agony. Overflow attack in a routing

table overflow the vulnerable node of the network by formation of packets with fake route in order to consume resources. The establishment of genuine routes is interrupted. The utilization of batteries of a specific node also increases by continuously keeping node engaged in routing decisions. It is known as sleep deprivation torture attack.

• **Routing Table Poisoning**: The table that holds information regarding routes of the network is preserved using routing protocols. The malicious nodes in poisoning attacks creates and sends fabricated signaling traffic, or some time alter messages that are genuine from supplementary nodes, for creating fake entries of the participating nodes in the tables [12].

Various types of attacks on different layers of OSI are summarized in Table below:

Table 1.1: Attacks on each layer in MANETS [9]

Layer	Attacks
Application Layer	Data duplicity, Repudiation
Transport Layer	SYN flooding ,Session hijacking, Traffic analysis, monitoring, disruption, Jamming, interceptions, eavesdropping
Network Layer	Resource utilization, Wormhole, black hole
Data link Layer	Examining ,Traffic Analysis, interruption
Physical Layer	Jamming, eavesdropping

1.4 OBJECTIVE OF DISSERTATION

The primary concern of our dissertation is to provide secure communication and information transfer using mobile agent. This can be done by preventing information carried by agent from malicious access. Following are the objective needed to be achieved:

- ✓ Providing communication using mobile agent technology.
- ✓ Securing information using ciphering techniques such as AES and RSA.
- ✓ Comparing the used Ciphering techniques.

1.5 RESEARCH METHODOLOGY

1. Literature Survey with survey papers.
2. Installation and configuration.
3. Setup of simulation environment.
4. Designing mobile agent communication model.
5. Ciphering of agents' information using suitable algorithms.
6. Comparison of performances of different algorithms.
7. Result Analysis and conclusion.

1.6 ORGANIZATION OF DISSERTATION

Chapter 1 gives a brief introduction about MANET, Mobile agent, working of mobile agents, Client server model, comparison between mobile agent and client server architecture and various threats to mobile agent in ad-hoc network. We have also discussed threats depending upon OSI layers.

Chapter 2 provides the survey of various techniques used for detection and protection of agent and its information. This chapter gives information of some techniques, system and tools that have been used for our work.

Chapter 3 describes the experimental setup and implementation details to achieve desirable objectives such as agent creation and communication between different agents.

Chapter 4 describes the usage of encryption and provides the results by comparing different cryptographic algorithms.

Chapter 5 concludes the work and provides future scope.

CHAPTER 2

LITERATURE SURVEY

Mobile agent technology has provided several distributed applications that got fundamental changes over traditional communicational approaches like RPC, REV etc.. Security is the biggest concern which darkens the advantageous side of mobile agent infrastructure. As soon as the mobile agent migrates from home platform, it goes out of the control of its owner that gives the opportunity to attacker (eavesdropper, network sniffer, execution environment, etc.).

2.1 AVAILABLE TOOLS FOR MOBILE AGENT DEVELOPMENT

1. IBM- Aglet Software Development Kit (ASDK)

IBM-Aglet or Aglet Software Development Kit is one of the environments for developing JAVA based mobile agents [15]. These toolkits are open source freely available; Aglet 2.5 alpha is the latest version. Good Graphical user interface is provided for development of agent. It primarily consists of two packages- Workbench of Aglet and Aglet Building Environment (ABE).

2. Voyager

An agent development tool developed by ObjectSpace named as Voyager, in mid- 1996 [16]. Voyager is one of the commercial products of ObjectSpace that has taken over by Recursion Software Inc. since 2001.

3. Anchor

Lawerence Berkeley National Laboratory, U.S.A. developed Anchor agent toolkit that facilitates secure management and transmission of mobile agents in a heterogeneous distributed environments [17].

4. Zeus

Advanced Applications & Technology Department of British Telecommunication labs developed Zeus, an integrated environment for the rapid development of collaborative agent applications [18]. This toolkit is open source and freely available. It is compatible with most hardware platforms as it is purely implemented in Java. It can be compiled with FIPA (Foundation for Intelligent Physical Agents) standards.

5. JADE

Tilab developed JADE (Java Agent Development Framework) for multi agent applications for peer to peer communication architecture. It's a software framework that is fully designed and implemented in Java language [15,19,20]. Multi-agent systems simplification has been done by the use of a middle-ware that complies with the latest Foundation for intelligent physical agents (FIPA) 2000 specifications. For debugging and deployment phases of agent development a set of graphical tools is provided which is supported by it.

Table 2.1: Tool Kits and their Features

Comparison of Agent Development tool kits	**Aglet**	Voyager	JADE	Anchor	Zeus
Features					
Production nature	Open Source and freely available	Commercial	Open Source and freely available	BSD license Available	Open Source and freely available
Implementation Standards	MASIF	- - - - -	FIPA	X. 509	FIPA
Technique for Communication	Both Asynchronous and Synchronous	Every method	Asynchronous	Asynchronous	Asynchronous
Security Technique	Poor	Weak	Good	Very Good	Good
Mobility of Agent	Weak	Weak	Good	Weak	Don't Support
Agent	Socket	RMI	RMI	Socket	Null

Migration					

2.2 EXISTING SECURITY MECHANISMS

Various techniques have been proposed for securing mobile agents. We have divided them in three categories:

- ✓ Platform Dependent
- ✓ Platform Independent
- ✓ Platform Dependent/Independent

2.2.1 PLATFORM DEPENDANT:

A trustworthy relationship using Mobile Agents: M.Uddin et. al. [23] proposed a security framework for cloud computing environment using mobile agents. This framework establishes trustworthy relationship among clients and service providers. It has 4 layers. Each layer is capable of performing verification, authentication and integrity during communication. SSL key exchange method is used for authentication. They have used mobile agents that Perform tasks requested by the clients and the cloud service provider. These tasks are first agreed by both client and service providers using SSL. This framework dynamically configures and adds services on the clusters in cloud data centers.

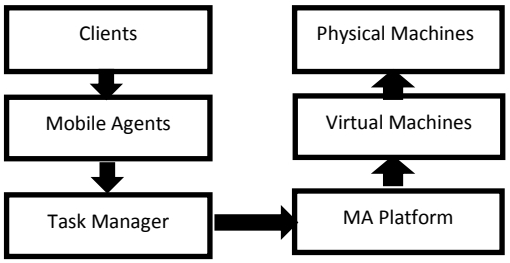

Figure 2.1: Security Layers [23]

Reference Clone: L.Benachenhou et. al. [24] proposed reference based execution for protection of the Mobile Agent from malicious attacks. In this method, code integrity is verified by comparing the under execution agent with its reference clone. The reference clone is kept under a trusted

server. As a result, it protects the agent from some types of modifications such as code modification, execution modification and data modification. In case of Denial-Of-Service attack, the reference clone cannot help the mobile agent. It is not capable of protecting state of mobile agent because the state of the Mobile Agent changes if it moves from one platform to any other platform and it is not possible by using the reference clone in trusted server.

Mobile Trust: C.Lin et al. [25] have proposed Trust Enhanced Security architecture for Mobile Agent system. In This method, the authors have used traditional security mechanisms with trust decisions. This is done with the help of certain relationship policies. A Trust Management Layer named Codifying Trust Evidence acts to present the security-related trust relationships. This provides a trust enhanced security mechanism. The trust evidences are derived, presented, evaluated and decisions are made on the basis of data in trust relationship database. This architecture has been worked out in Aglets platform and is very effective also.

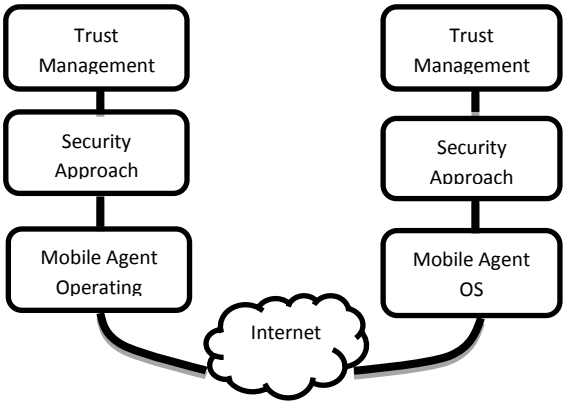

Figure 2.2: Mobile Trust Model [25]

Formal Modeling: L.Ma et. al. [26] proposed an extended elementary object system which is a coupling of P/T Nets and Object Orientation. The Reference Clone approach supports weak mobility. The EEOS supports strong mobility as well as it provides a secure communication. The transactions or executions are done by authentication between Mobile

Agent and Execution Platform. It also includes an approach to detect malicious attacks on Execution Platform. This model must involve more security mechanisms which would best fit into the generic model.

Malicious Availability and Code Integrity Test: S.Venkatesan et al. [27] proposed the eXtended Root Canal Algorithm (ERC algorithm) which is helpful in verifying code integrity and the Identifying Maliciousness of mobile agent. The ERC Algorithm is an extended version of the Root Canal Algorithm (RC Algorithm). The RC Algorithm was used to check the integrity of code of Mobile Agent. This method has a low time complexity as compared to other competitor techniques such as Code on Demand and SeMoA. Malicious Identification Policy (MIP) greatly helps in identification of malicious activities. The ERC Algorithm is efficient in terms of time complexity and space complexity. It also protects the Mobile Agent from all attacks other than the replay attacks.

Dual Check-Point Method: P.Marikkannu et al. [28] implemented the Dual Check-Point Analysis technique which addresses an unidentified attack specifically piggybacking attack, in which the Malicious Agent attaches to the Mobile Agent to attack either the Mobile Agent or the Execution Platform. It preserves an Authentication Table to check integrity and authenticity of code of Mobile Agent. The technique is named Dual Check-Point Analysis because it checks the mobile agent in terms of Digital Signature Verification at outer gate and in terms of Size verification as well at inner gate. This technique provides a good solution to the piggybacking attacks.

AIF (Artificial Immune System): S.Venkatesan et al. [29] implemented the Artificial Immune System based model of Mobile Agent. The Platform Protection gives a clean classification of division of duties and clones to tackle with foreign agents. It enhances the computational capabilities of the system. In this approach, the duties are clearly divided; therefore

protection is given by only those agents who are capable of identifying malicious activity. This method drastically reduces the computational cost.

K-response Recovery Model: S.Venkatesan et al. [30] presented K-response Recovery Model that recovers the various components of a Mobile Agents such as agent code, state, itinerary and data. This approach can recover components even after a malicious attack. In this mechanism, each host sending off the Mobile Agent keeps a clone of the Mobile Agent with its previous state. The host Sends message to the execution platform (where the mobile agent was previously executing) whether Mobile Agent is alive or dead. This technique may not work if the Mobile Agent lives through a host where all the execution platforms are malicious and they collectively coordinate attacks.

Trust and Reputation Management: G.Geetha et al. [31] implemented the Trust and Reputation Management framework for colluded truncation attacks. The major focus on security is directed towards the path which is followed by the Mobile Agent. A routing table has been designed for the route of Mobile Agent with a secure path based on reputation and trust. In addition, cryptographic algorithms have been used to ensure the security of mobile agent.

Hierarchical Key Management: Y.F.T.L.Chen et al. [32] introduced the Hierarchical Key Management Scheme to reduce the complexity of key management by a Mobile Agent, since mobile agent has the load of being transferred from system to system. They have used Elliptic Curve Cryptography to increase the security. Therefore it boosts the performance of mobile agent because the overhead of transmitting the key is reduced. The keys automatically get modified according to time. This is very important, since a Mobile Agent that has access to a particular resource at a particular time may not have the access to that resource at any other time. Therefore, agents can access resources only when they are permitted

to access. This approach is resistive to reverse attack, external collective attack, collusion attack and date alteration attack.

Proxy Signature Protocol: X.Hong [33] presented threshold proxy signature protocol that uses a proxy signer to sign a digital signature. The digital signatures are done on behalf of owner to whom the agent belongs. The proxy signing process uses RSA algorithm and it is shared using Lagrange Formula. However flaws in this approach make it vulnerable to attacks. We have found that the security algorithm was unable to ensure secrecy, unforgeability, undeniability, prone to identifiability, and is more prone to attacks.

Trip marker: C.Garrigues et al. [34] proposed a good approach to prevent replay attacks in Mobile Agents by the usage of agent identifiers with trip markers. The trip marker keeps in track the route of the Mobile Agent. It is possible to avoid replay attacks, because the execution platforms have a real time track over the trip marker. It is possible only if the execution platform is not compromised. So this approach is platform dependent.

Self-Reliant Mobile Code: S.Srivastava et al. [35] proposed Self-Reliant Mobile Code which is a collaboration of different techniques based on confidentiality, integrity and self-protection. The authors have formalized a spontaneous symmetric key algorithm based on Petri net. One of the key components is distributed in a secure manner and other key component is derived from the collected data at run time or at execution time.

Elliptic Curve Cryptography: A.Zakerolhosseini [36] presented the coupling of the Blind Signature along with Elliptic Curve Cryptography which is a security mechanism to secure the Mobile Agent as a whole. The idea was the collaboration of Elliptic Curve Cryptography with a preliminary access control mechanism. This scheme is useful in case of Reverse Attack, Conspiracy Attack and “Man-in-the-Middle” attack. It is

obvious that most of the approaches are prone to attack. Also all type of attacks can't be prevented by those techniques. Therefore discussed approaches can be used if the application demands so.

RBAC: H.Idrissi et al. [37] proposed a mechanism for preventing unauthorized access to mobile agents. Their approach is applied on a hospital providing tele-medical services. It dynamically assigns roles within the architecture, where permissions and roles are simulated as cooperative mobile agents. For this task, they use attribute certificates issued by PMI (privilege management infrastructure), in addition to elliptic curve cryptographic primitives and refer to the role engineering database. According to our survey, this approach is effective for detection of unauthorized access in terms of robustness and time.

2.2.2 PLATFORM INDEPENDENT

Dynamic code: T.Wang et al. [38] implemented dynamically upgradeable code. In this technique the code can add new modules to Mobile Agent and delete the duplicate ones also. It can also be used to enhance privacy of code. To ensure integrity of code it uses process of Double Integrity Verification scheme and Code Change Authorization Protocols (CCAP). For integrity protection, an authorized mechanism is used. When the code of agent is changed the host validates the changed code by using that mechanism. However this approach does not provide a good solution for code integrity problem.

Proof Carrying Code: G.C.Necula et al. [39] implemented the Proof Carrying Code in which the code of Mobile Agent needs a proof which is essential for execution. The code is checked in order to cross check the security policies of that system.

MACPL: C.Garrigues et al. [40] focused on securing applications using Secure Mobile Agents. The proposed architecture helps in reducing the complexity in implementation of Cryptographic Protocols. It enhances the

reuse of the protocols by implementing the agent task and security mechanism as separate entities. The architecture uses Mobile Agent Cryptographic Protection Language (MACPL). This enhances the reuse of control codes. In this approach, platforms are not responsible for providing protection to the agent. Therefore this approach increases the agent size which would affect performance in real-time systems. This mechanism is also attack specific because it does not address all type of attacks.

Self-Modifying Code: S.L.Shan et al. [41] implemented the Self-Modifying Codes which uses the code obfuscation algorithm for modification of code by itself to avoid attack at functional level. Obfuscation has two types: static obfuscation and dynamic obfuscation. For achieving static obfuscation, there are many techniques such as Inheritance relationship modification, Pointer manipulation, Address manipulation, branch function obfuscation, array reconstruction and variable reconstruction. The protection provided by dynamic code obfuscation is more as compared with static code obfuscation. Dynamic obfuscation is provided by techniques such as Jump table spoofing, self-modifying code, Inter-process communication and signal obfuscation. The decision of obfuscating the function call to control flow or normal instruction is done by using Liveness analysis.

BROSMAP: D. Shehada et al. [42] proposed BROSMA protocol that works in real time and distributed applications. The copies of agents are sent at a time to all service providers. It continues its functioning even when one of the agents are lost or killed maliciously. This architecture is capable of providing applications with security and good performance. In this approach both symmetric and asymmetric cryptographic protocols are used along with digital signatures. It is proved to be a novel approach as it prevents several types of attacks such as MITM, replay attack, unauthorized access etc.

2.2.3 DEPENDANT/INDEPENDANT

Sandbox: T.Grandison et al. [43] implemented Sandboxes inside the browser using java interpreter. These are used for mobile agents. This security technique mainly consists of three elements namely class loader, security manager and verifier.

Code Signing: H.Reiser et al. [44] proposed a novel approach i.e. Code Signing which uses Microsoft Authenticode. In Microsoft Authenticode, ActiveX is used for code signing. The system consists of a particular policy. If that policy is changed by a Mobile Agent or any other external entity, it is considered as a threat. It introduces a trust model to differentiate between the trustworthy authors and untrustworthy authors.

Table 2.2: Various Techniques for Securing Mobile Agent

Authors and Techniques Used	Attacks Detected/Prevented	Security Approach	Comments
(M.Uddin et al.) A Trustworthy Relationship Using Mobile Agent"[23]	Unauthorized Access	SSL Key Exchange Method	-Ensures Security and Privacy of Client Data.
(L.Benachenhou et al.) Reference Clone[24]	DoS, Unauthorized Access, Alteration	Clone of Agent is kept under a trusted Server	-Can't protect state of mobile agent. -Supports weak Mobility.
(C.Lin et al.) Mobile Trust[25]	Unauthorized Access, Repudiation	Cryptographic Mechanisms such as DES , RSA etc.	-Effective Approach used in Anglets also.
(L.Ma et al.) Formal Modeling[26]	Unauthorized Access	EEOS Approach	-Strong Mobility -Require more refinement.
(S.Venkatesan et al.) Malicious Availability and Code Integrity Test[27]	Unauthorized Access, masquerading, alteration	eXtended Root Canal Algorithm, MIP	-Low Time and Space Complexity

(P.Marikkannu et al.) Dual Check Point Method[28]	Tailgating	Digital Signature Verification	-Good solution for Piggybacking Attacks.
(S.Venkatesan et al.) AIF[29]	Unauthorized Access, Masquerading	Pattern Matching	-Drastically reduces computational cost
(S.Venkatesan et al.) K-Response Recovery Method[30]	Masquerading	Keeps a Clone of Mobile Agent with Previous State.	-Can't work if Mobile Agent works through a malicious host.
(G.Geetha et al.) Trust and Reputation Management[31]	Repudiation, Alteration and Eavesdropping	Traditional Cryptographic Mechanism: RSA.	-Effective framework for colluded truncation attacks.
(Y.F.T.L.Chen et al.) Hierarchical key Management[32]	Alteration, Eavesdropping	Elliptic Curve Cryptography	-Resistive to Reverse Attacks, External Collective Attacks, Collusion Attacks and Data Alteration Attacks.
(X.Hong) Proxy Signature Protocol[33]	Unauthorized Access	Digital Signature, RSA Algorithm.	-More prone to attacks with timing constraints
(C.Garrigues et al.) Trip Maker[34]	Replay Attacks	Trip Maker	-Effective if and only if the Execution Platform is not compromised
(S.Srivastava et al.) Self-Reliant Mobile Code[35]	Unauthorized Access, Alteration	Symmetric Key Algorithm	-Reduces Computational cost
(A.Zakerothosseini) Elliptic Curve Cryptography[36]	Unauthorized Access, Repudiation, Alteration,eavesdropping	Blind Signature, Elliptic Curve Cryptography	-useful in case of Reverse Attack, Conspiracy Attack, Man-

			in-the-Middle Attack
(H.Idrissi et al.) RBAC[37]	Unauthorized Access	Elliptic Curve Cryptography	-Good at detecting unauthorized access
(T.Wang et al.) Dynamic Code[38]	Alteration	Double integrity Verification Scheme, CCAP	-Doesn't provide a complete solution for code integrity problem
(G.C.Necula et al.) Proof Carrying Code[39]	Unauthorized Access, Repudiation	Proof-Carrying Code	-Good approach for verifying the safety policies.
(C.Garrigues et al.) MACPL[40]	Unauthorized Access	Cryptographic Protection Language	-Agent size affects performance.
(S.L.Shan et al.) Self-Modifying Code [41]	Alteration	Code Obfuscation Algorithm	-A resistant method for code of Mobile Agent
(D.Shehada et al.) BROSMAP [42]	Replay attack, MITM, Masquerading, Unauthorized access	Both symmetric and asymmetric cryptographic protocols along with digital signatures.	-Good approach for preventing a number of attacks.
(T.Grandison et al.) Sandbox[43]	Eavesdropping	Sandbox Technique	-Good for separation of concurrent programs
(H.Reiser et al.) Code Signing[44]	Unauthorized Access	Microsoft Authenticode	-Good approach for differentiating between trustworthy and untrustworthy authors.

2.3 ISSUES INDENTIFIED IN LITERATURE SURVEY

List of various issues that we have seen above in our survey are as follows:

- ✓ The reference clone is kept under a trusteed server in the approach proposed by L.Benachenhou et al. [24]. If the trusted server gets compromised, the intruder can stop the execution of authorized agents and let the malicious agents run in the platform. Data can be tempered by the malicious agents. Any cryptographic technique must have been used so that the data is secure in compromised environment too.
- ✓ In K-response recovery model by S.Venkatesan et al. [30], components of mobile agents are recovered even after malicious attack. We can recover code and data of mobile agent using this technique, but this approach has not prevented the attack. It has been used to recover the agent components only.
- ✓ X.Hong [33], in his proxy signature protocol, has used RSA algorithm in proxy signing process along with digital signatures. It was more prone to attacks. It was unable to ensure secrecy.
- ✓ The Tripmaker approach, proposed by C.Garrigues et al. [34], mainly focused on the route of mobile agent. If the execution platform is vulnerable, it not possible to maintain secrecy of agents coming from other platforms. Along with Tripmaker, any symmetric or asymmetric cryptographic algorithm should have been used.
- ✓ A proof is essential for a mobile agent to execute in approach of G.C.Necula et al. [39]. If a platform is malicious the data of mobile agent will be exposed. Any Cryptographic algorithm should have been used to ensure the secrecy of data of mobile agent.

Mobile agent travels in ad-hoc environment where attack ratio is very high. As a result of literature survey, no reliable and secure environment is available for agent to migrate. Either the agent data should be securing enough or the environment. A lot of work has been done using RSA

cryptography. Although AES is considered to be fast algorithm, but it has never been used for agent based environments.

2.4 CONCLUSION

We came across through literature survey about many limitation and security issues that make information carried by mobile agent unsecure. Many researchers have used RSA for ciphering the mobile agent's data. Although AES has considered being a fast algorithm, but it is not been used in agent based environments. In our dissertation work we will prevent our data from malicious access by using AES and RSA, and finally we will compare their performance.

CHAPTER 3

PROPOSED WORK

Development of agents as well as their wide usage requires good underlying infrastructure. Literature survey indicates scarcity of agent development tools in initial years of research which limited the exploitation of this beneficial technology. However, a wide variety of tools are available for developing robust infrastructure.

3.1 PROPOSED AGENT BASED COMMUNICATION MODEL

We have proposed a secure communication model for mobile agent data and information transfer. In our dissertation work, we have provided security at transport layer. This can be done by securing information of agent using encryption technique thus preventing transport layer from malicious attacks. Detailed discussion about methodology is seen in rest part of chapter.

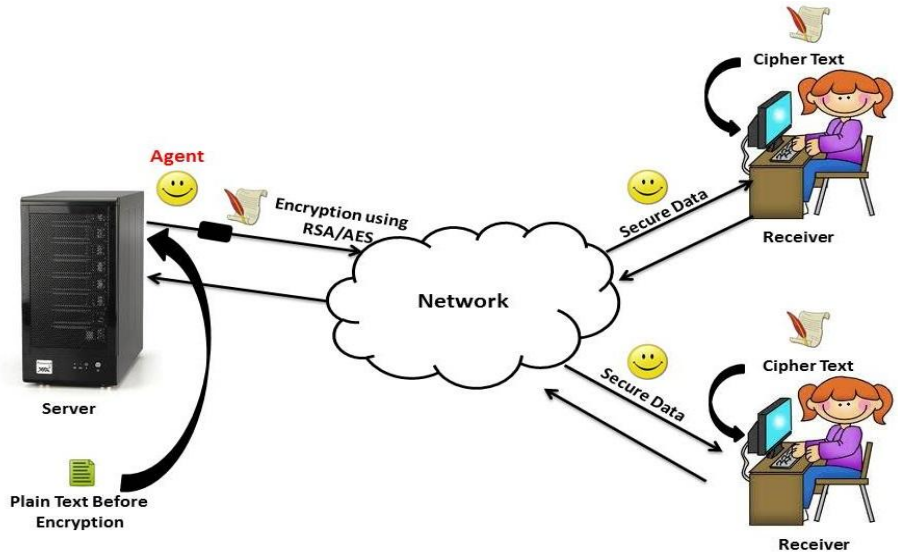

Figure 3.1: Secure Agent Communication

It is shown in the above figure that the information is first encrypted by the agent and then the agent migrate form home platform to another platform along with encrypted data. We are using JADE pslatform for agent development and communication.

3.2 AGENT DEVELOPMENT IN JADE

JAVA agent Development Environment (JADE) is a middleware for developing multi-agent system very easily [45]. It includes a library of classes that programmers use in order to develop agent, a runtime environment in which the jade agent resides and a collection of graphical tools that permits monitoring and administrating the running agent activity. Sometimes, agents include artificial intelligence algorithms in order to become more intelligent. JADE is a FIPA based compliant. Agents are created and kept in a repository which is called container. Agents have the addresses of source and host so that they can migrate from host to host. Agents have buffer to store metadata. They also have a unique identity for distinguishing among all agents. It defines the platform with three important services. JADE Agent Platform, includes the AMS, the DF (Directory Facilitator), and the ACC (Agent Communication Channel). All these three agents are automatically activated at the agent platform start-up. Management of agent is done by DF and AMS.

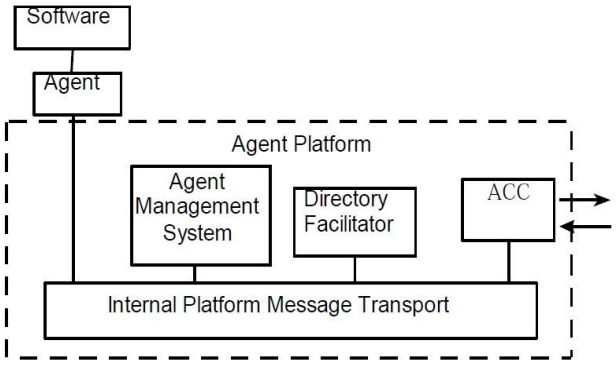

Figure 3.2: JADE Architecture [45]

JADE provide us with the following features:

- ✓ JADE Agent Platform, includes the AMS (Agent Management System), the DF (Directory Facilitator), and the ACC (Agent Communication Channel). All these three agents are automatically activated at the agent platform start-up. Management of agent is done by DF and AMS.

- ✓ It is a Distributed agent platform that can split on several hosts. For single Java application, single Java Virtual Machine, is executed on each host. Agents are implemented as one Java thread and Java events are used for effective and light-weight communication between agents on the same host. JADE does scheduling of these tasks in a more efficient compared JVM for threads.

- ✓ A number of DFs (Directory Facilitator) can be started at run time in order to implement multi-domain applications.

- ✓ Transport mechanism and interface to send/receive messages to/from other agents.
- ✓ It Consist of IIOP protocol to connect different agent platforms.
- ✓ It provides automatic registration of agents with the AMS.
- ✓ The start-up agents obtain their GUID (Globally Unique Identifier) from the platform.

3.3 SECURITY PROTOCOL IMPLEMENTATION

As we discussed in Chapter 2, many researchers have provided different solutions for protecting agent's information but it's still an area of concern for researchers. In our dissertation work we have introduced security protocol for providing protection to mobile agent against different threats/attacks. Migration property of mobile agent that helps agent to move to different remote location makes this communication architecture superior than others. Our research work is providing mobile agent security

during communication or execution thus targeting confidentiality. We have provided transport layer protection using AES and RSA algorithm.

3.3.1 AGENT DEVELOPMENT AND COMMUNICATION

Agents are created inside containers. In our scenario, we have created 2 containers other than main container i.e. Container-1 and Container-2. We are taking an example of a customer and a vendor, in which the server agent is customer agent named “customer”. It is created inside main container which is the default container for agent creation. The vendor agents are created in Container-1 and Container-2.

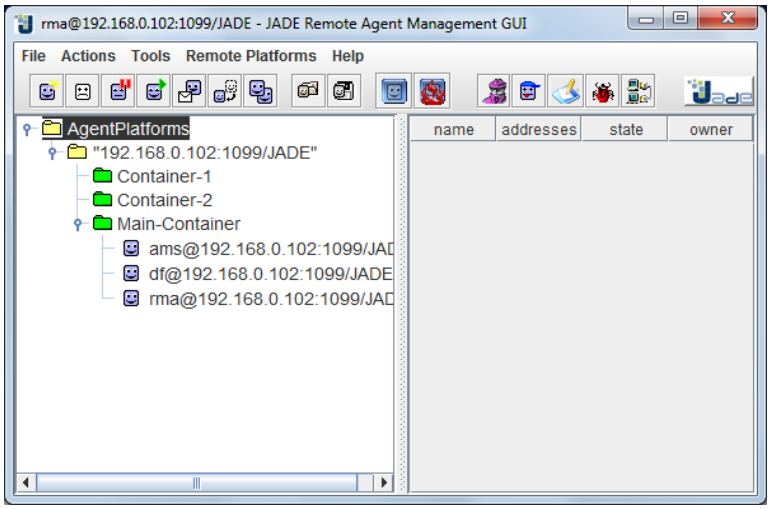

Figure 3.3: Creation of Containers

The customer agent will put a query for a book and its price. Other vendor agents will tell prices for that book if that book is available in their store. The customer agent will select that book which has the lowest price among all. The vendor agents will reside in container-1 and 2 respectively as shown in figure below:

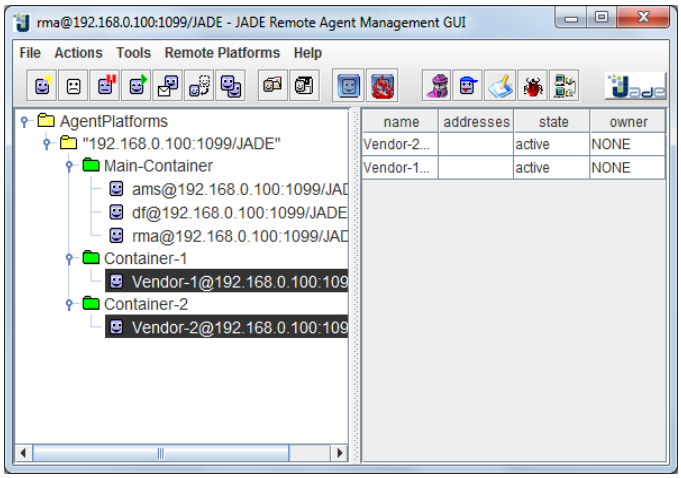

Figure 3.4: Creation of Vendor Agents

After creation of vendor agents, different vendor agents can quote price according to themselves. These agents will tell the price of various books. We are taking example of a book of Operating System as shown in figure below. We will apply security protocols on data that is described in the next section of this paper. The vendor enters the price as soon as vendor agent is created. Following Figure shows Vendor-1 and Vendor-2 that entered price as “447” and “337” respectively for book named “Operating System”. These values get stored in buffer.

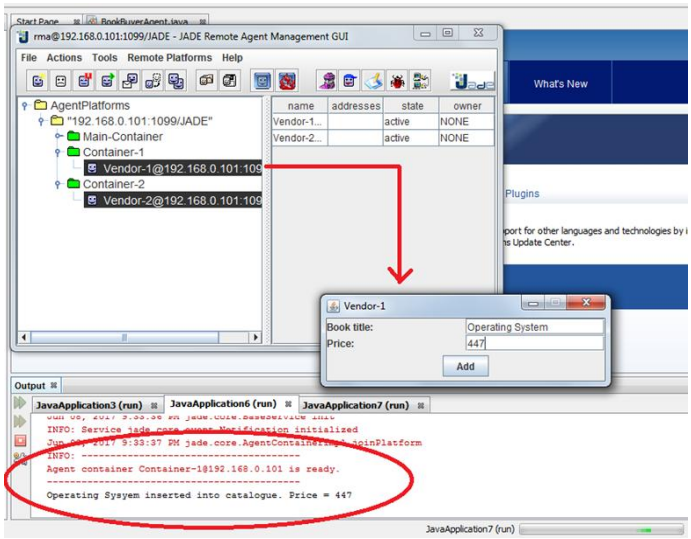

Figure 3.5: Vendor-1 providing book and its price

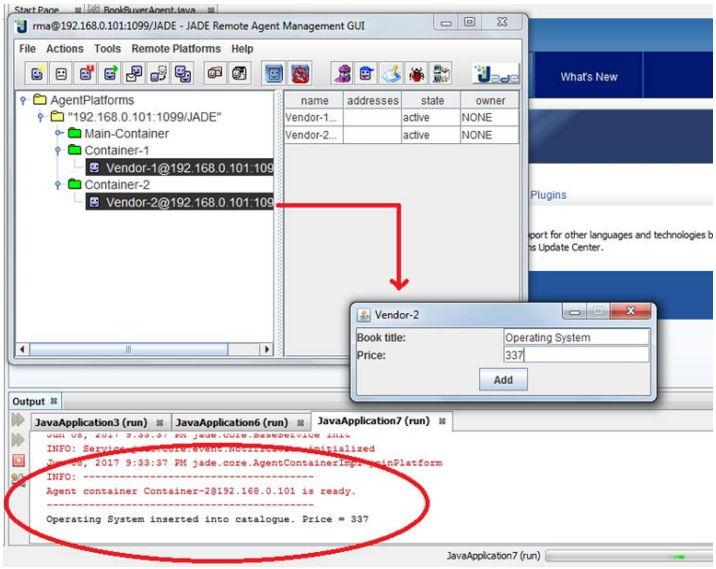

Figure 3.6: Vendor-2 providing book and its price

The vendor agents had been created which will respond to the query fired by the customer agent. In the below figure, a customer agent has been created which will search for a particular book and the vendor agents will respond back with price of that book if available in catalogue.

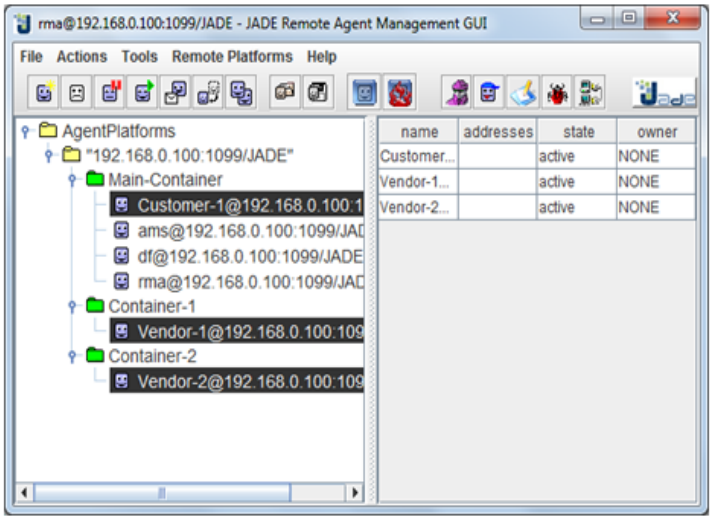

Figure 3.7: Initialization of two vendor agents and a customer agent

The customer agent will select the book having minimum price as shown below in figure:

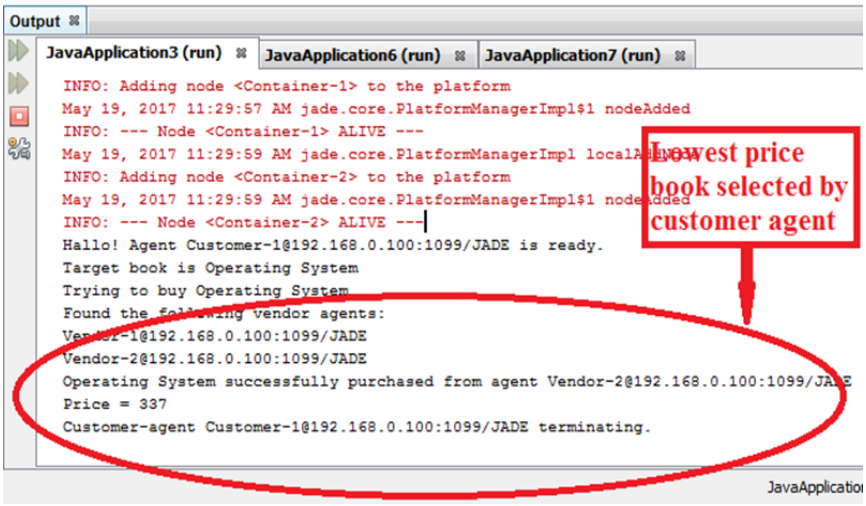

Figure 3.8.: Lowest price book selected by customer agent

The following figure shows the communication between customer agent and vendor agents. This is done with the help of a sniffer agent.

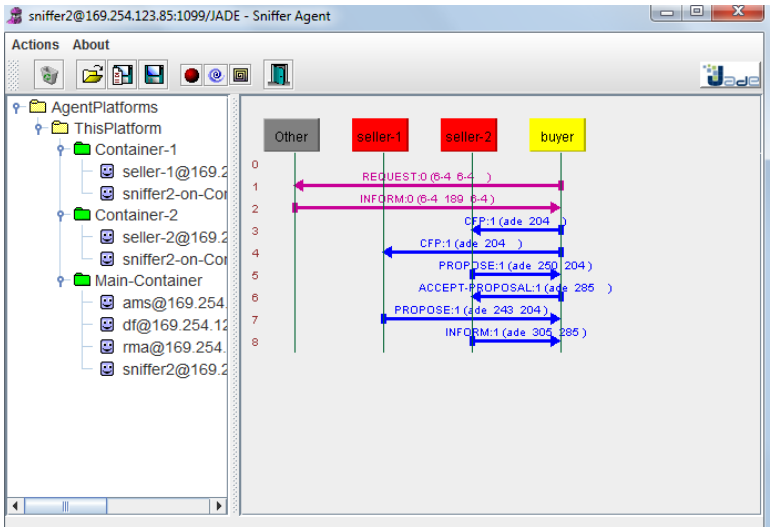

Figure 3.9: Communication between customer and vendor agents

3.4 DATA ENCRYPTION AND DECRYPTION

We have taken a case for information transfer which is an example of book-buying that includes two entities ‘book name’ and ‘book price’. An agent named Customer (data initiator) is created in the main container and other agents named Vendor-1 and Vendor-2 who respond to the query fired by buyer. The agent lies in ad-hoc network hence, the intruder can

put all the efforts to get the information from the agent. But if any agent itself will encrypt the data by using some encryption method and then forward the information to another agent, cracking key will be a difficult task. Hence no other agent would be able to access other agents' information.

TEXT CIPHERING

In our mobile agent security architecture, we have mainly focused on securing communication rather than securing agents or platforms. Cryptographic protocols are used for transport layer security. The communication process is completed in three phases:

Setup (Generation of private and public key)

Encrypt (converting any arbitrary string into unreadable form)

Decrypt (Decryption using corresponding private key).

Encryption of data is done with RSA algorithm and AES algorithm. The data is in the form of a text file. The data sent form one agent is secured by encrypting it and the receiving agent decrypt it using key. Below is the brief discussion of algorithms (AES & RSA) and their working:

3.4.1 RSA ALGORITHM

1. Select two large prime numbers p and q, where p! =q.
2. n<-p *q
3. c(n)<- (p-1)*(q-1)
4. Select a number e such that 1 < e < c(n) and e is co-prime to c(n).
5. k<- e^{-1} mod c(n)
6. Public key <- (e,n) //Known to all
7. Private key <- d //Known to receiver
8. Generate private key and public key.

For encryption, we have taken RSA parameters as:

p=3, q=11, p*q=3*11=33. Here, c(n) = (p-1)*(q-1) that is; 2*10=20, and value of e is taken between 1 and c(n), that is; e=7. The numerical values of alphabets will be a=0, b=1, c=2, d=3…. z=25. Steps are as follows:

- Compute value of k where k= (k*e) % c (n)-1. One of the possible answers of the statement is: k-> 3[(3*7) % 20-1]
- Public key is (e, n) -> (7, 33)
- Private key is (k, n) -> (3, 33)

RSA is an asymmetric encryption algorithm. Therefore the sender agent encrypts data using public key and the data will be decrypted by only that agent who knows the private key [12]. Theoretically there is no limit for key size of RSA but it is not possible yet to compute factors for numbers which are greater than 768 bits. That is why modern systems use a minimum key length of 3072 bits. We have used key size of 16 bytes.

3.4.2 AES ALGORITHM

The second algorithm on which we have worked is Advanced Encryption Standard (AES) [13]. AES provide us with key length of 16, 24, 32 bytes. We have used 16 bytes key length.

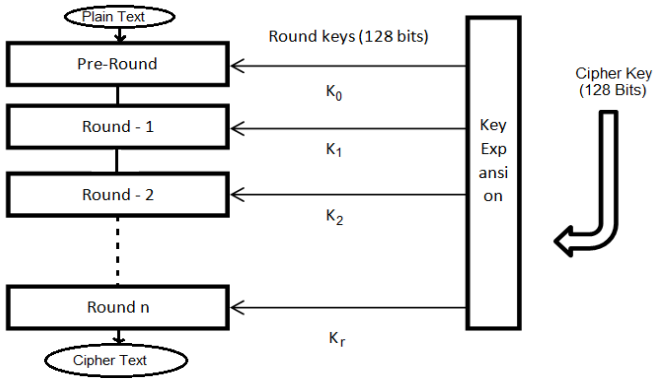

Figure 3.10: AES Algorithm

The encryption process in AES uses derived key sets, which are called round keys. These keys are applied on array of data that is going to be encrypted along with some other operations. This is called state array.

The following steps have been taken to encrypt a block of data (128 bit) using AES:

1. Derived the round keys from the main cipher key.
2. Initialized the state array with plaintext.
3. Added the starting state array to the first round key.
4. Performed nine rounds for manipulation of state.
5. Performed the final and last round for manipulation of state.
6. Copied the final state array out as the cipher text.

The tenth round has a bit different manipulation than others. That is why first nine rounds are different from the tenth final round. Each round has its own round key that is derived from the original 128-bit encryption key. Each round contains four steps. One of the four steps performs XORing of the state array with the round key.

Key Expansion:

AES Key Expansion algorithm is used to derive a 16-bytes round key for each round. It is derived from the main 16 bytes encryption key. Here the logic is; if 1 bit of encryption key is being changed, it will make a difference in round keys for coming several rounds. First of all the algorithm arranges the 16 bytes encryption key in a 4×4 array, as shown below.

$$\begin{bmatrix} k_0 & k_4 & k_8 & k_{12} \\ k_1 & k_5 & k_9 & k_{13} \\ k_2 & k_6 & k_{10} & k_{14} \\ k_3 & k_7 & k_{11} & k_{15} \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} w_0 & w_1 & w_2 & w_3 \end{bmatrix}$$

Figure 3.11: 4x4 array of 16 bytes

Here the words [w0 to w3] are bitwise XOR'ed with the input block of data before the round processing starts.

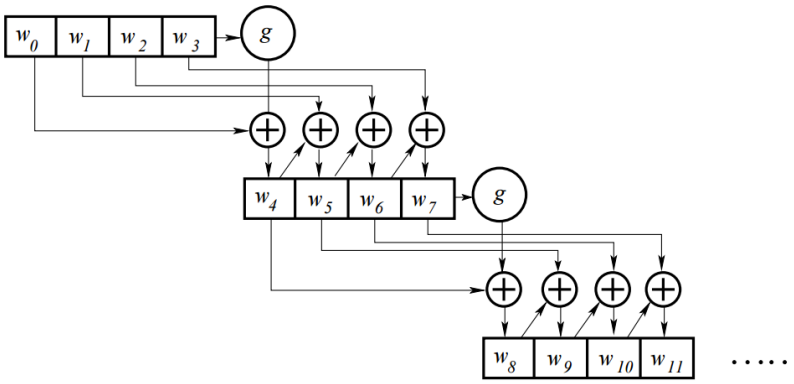

Figure 3.12: Round Key generation

On the receiver side, all the steps are taken in reverse to decrypt the cipher text using AES 16 bit private key.

Any vendor agent who wants to communicate with customer agent will encrypt his data. Therefore no other vendor would be able to know what price the other one has mentioned. Hence no unauthenticated person would be able to manipulate, delete or read that data. When the customer will get the price list of all the vendors he will decrypt the information using private key. We had taken an example having book name “Operating System” and whose price is “337”. This information is stored in buffer in encrypted form only the customer can decrypt it by using its key.

CHAPTER 4

RESULTS AND DISCUSSION

Mobile agent migrates from one place to another in ad-hoc environment, which is prone to various threats. Thus, information carried by agent is not secure. To provide security to information we are using encryption technique. In our example of customer and vendors, we have used both the algorithms and analyzed their performance in agent environment for a text file. We have encrypted the text "Operating System and 337".

4.1 ENCRYPTION

The data is encrypted before migrating by sender agent as shown below:

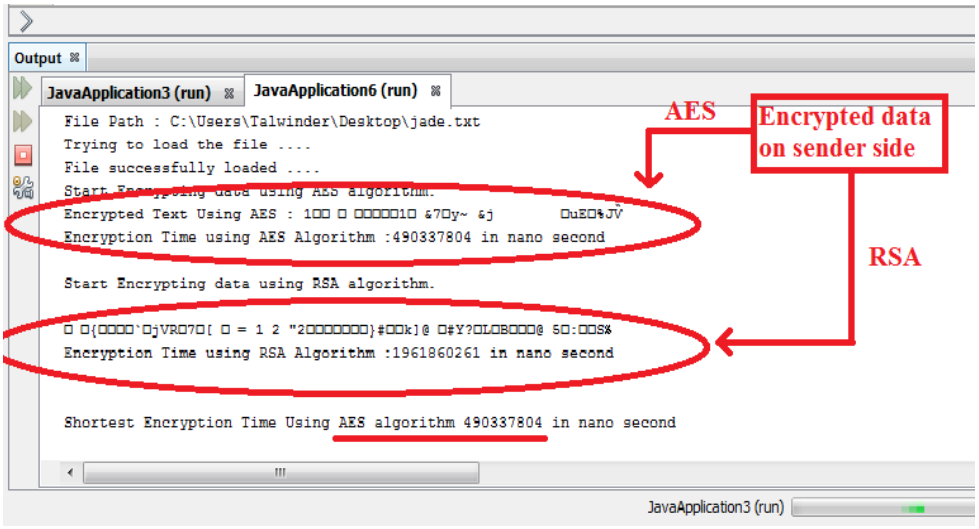

Figure 4.1: Encryption on sender side

Plaintext: Operating System and 337

Cipher text by AES:

1��������1�&7�y~ &j ̈ ̈ ̈

&#x1f;&#x1f;&#x1f;

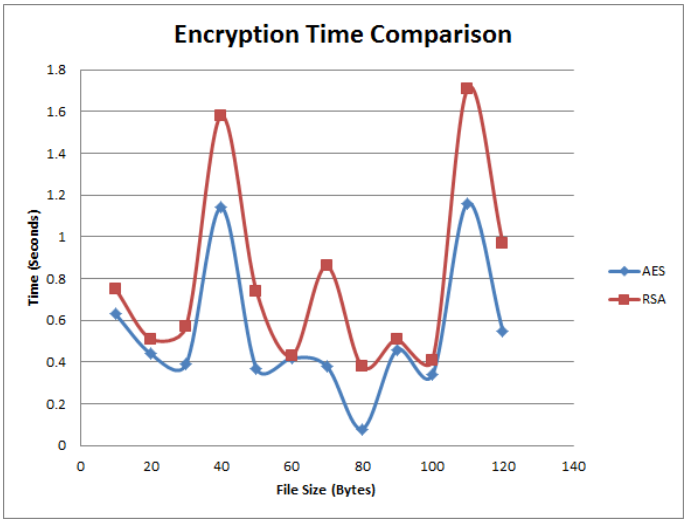

Figure 4.2: Encryption Time Comparison

4.2 DECRYPTION

On receiver side the data is decrypted by receiver agent using the decryption key. The received data after decryption is shown in the output console.

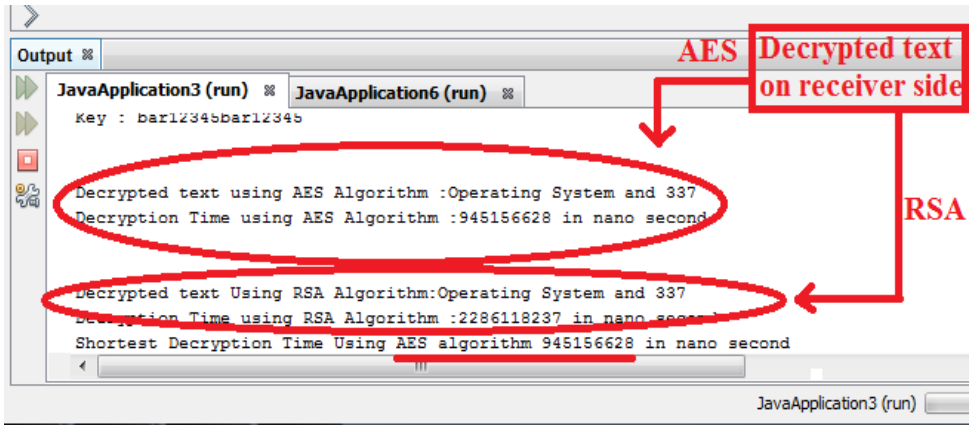

Figure 4.3: Decryption on Receiver Side

The data can be decrypted by those agents who have the key. The decryption is also done by both the algorithms.

Table 4.2: Decryption time of AES and RSA for different file sizes

DECRYPTION TIME

File Size (Bytes)	AES (Seconds)	RSA (Seconds)
10	0.69	1.21
20	0.43	0.67
30	0.43	0.63
40	1.31	1.55
50	0.36	0.55
60	1.13	1.59
70	0.41	0.54
80	0.41	0.82
90	0.75	1.09
100	0.38	0.53
110	0.62	0.9
120	0.45	1.44

The comparison graphs of AES and RSA on the basis of time taken to decrypt the files of different sizes is shown below:

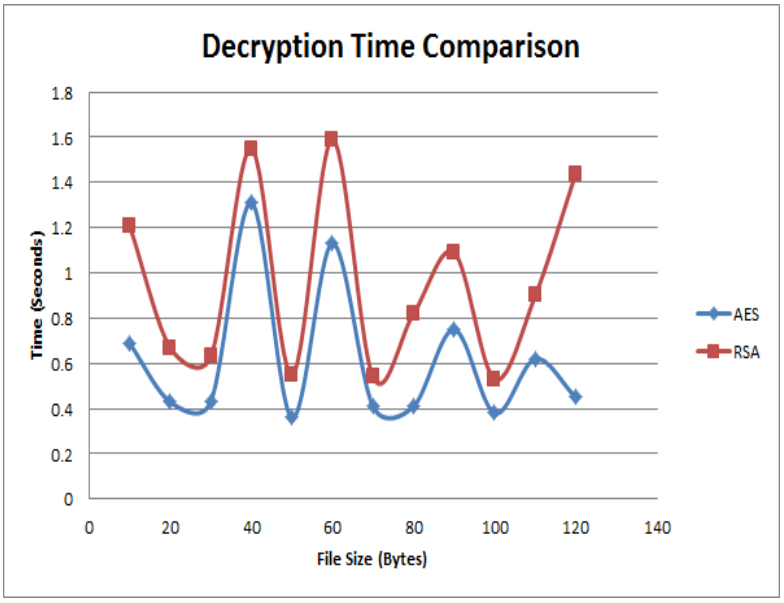

Figure 4.4: Decryption Time Comparison

We can conclude that for any file size, AES takes less time to encrypt and decrypt the data as compared to RSA.

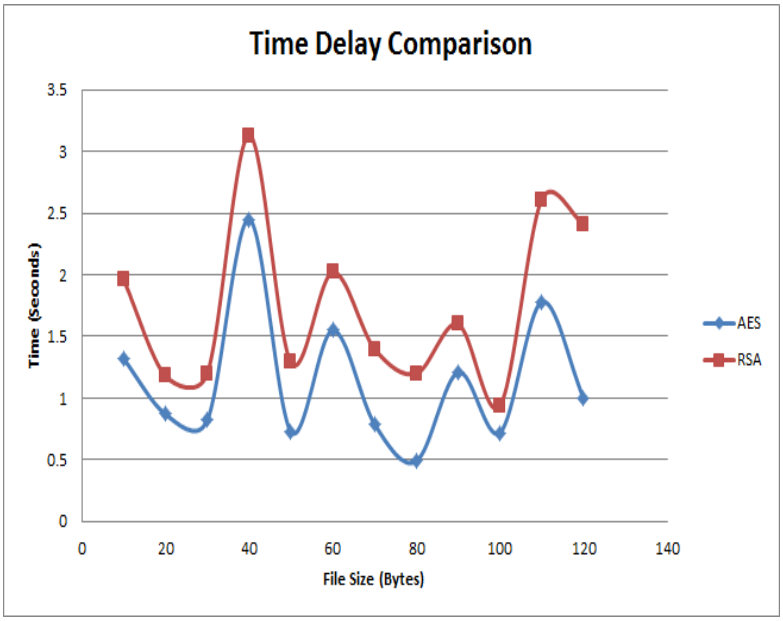

Figure 4.5: Time Delay Comparison

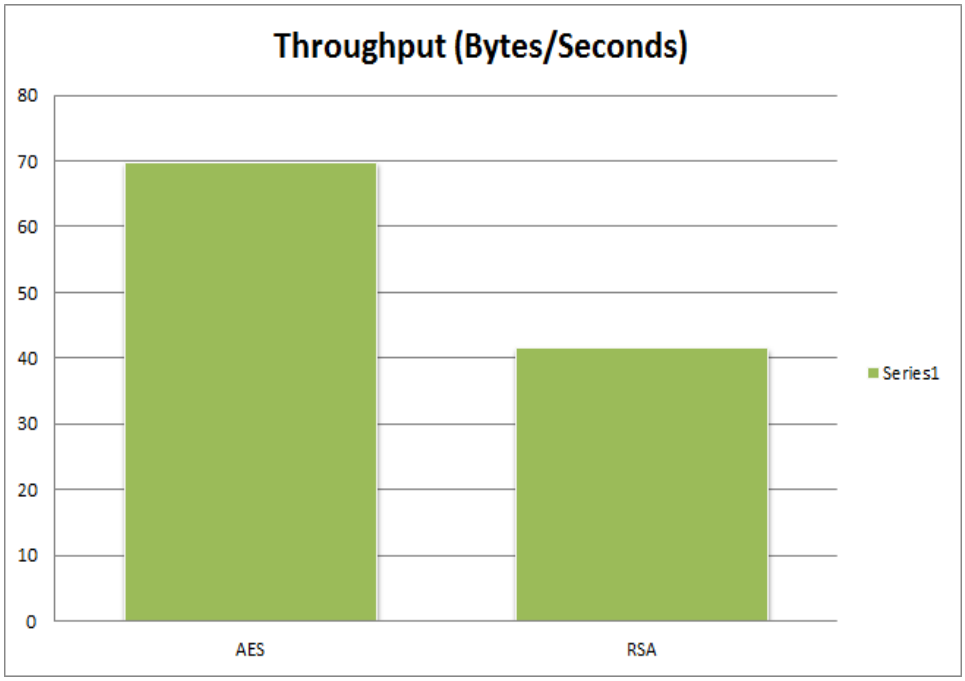

Figure 4.6: Throughput comparison of AES and RSA

In both the cases of encryption and decryption, RSA takes more time than AES for all file sizes.

- ✓ We have found that securing communication is better than securing platform or agent or both because it is independent of all of them.
- ✓ Securing communication can avoid a number of attacks such as eavesdropping, unauthorized access, alteration, sniffing and Man in the Middle Attack.
- ✓ RSA algorithm takes more time than AES; therefore RSA will take more time for any intruder to decrypt the data (if and only if the attacker knows the private key). In that scenario RSA will be more beneficial.
- ✓ AES is really fast, but suffers from the security risks of key exchange (which can be solved using RSA). That means, key exchange should be very secure so that no attacker can guess or steal the key. If the key is secured enough, then using AES is a better option. We can exchange key with the help of RSA and the data can be transmitted using AES.

Table 4.3: Comparison of AES and RSA

Factors	**AES**	**RSA**
Key Type	Single Private Key	One Public and one Private key
Encryption	Fast	Slow
Decryption	Fast	Slow
Ciphering & deciphering key	Same	Different
Block Size	16 Bytes	Minimum 64 Bytes
Rounds	10,12,14	1
Key Security	Less	More
Data Security	Excellent	Not Secure Enough
Key Size (Bytes)	16,24,32 Bytes	>128 Bytes
Brute force	Hard	Not so hard
Effective Security	16 bytes (AES-16)	10 Bytes (128--RSA)
Power Consumption	Low	High
Ciphering and Deciphering Algorithm	Different	Same

- ✓ Using RSA algorithm can also increase the delay in communication.

  Encryption algorithm plays very important role in communication security. Our research work analyzed the performance of existing encryption techniques like AES and RSA algorithms. Based on the text files used and the experimental results, it was concluded that AES algorithm consumes very less encryption and decryption time as compared with RSA. From the simulation result, we evaluated that AES algorithm is much better than RSA algorithm.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

5.1 CONCLUSION

In this dissertation we have discussed information communication using two nodes in ad-hoc way by using client-server and mobile agent based communication model. We have compared the two communication strategies and found that mobile agent based communication is more relevant and fruitful for ad-hoc communication. We have also discussed various threats to mobile agents. Threats are due to the mobility of agents, which is necessary for data forwarding from one agent to another. Since the communication channel is wireless and information is carried by Mobiles Agents form one station to another station, information is not secure. In other word information is highly prone to attacks and threats. Therefore, in this work we have proposed and discussed a method for secure mobile agent information between two nodes and provided secure communication using cryptography. For agent development and transformation we have used JADE development environment. For the purpose of ciphering, we have used RSA and AES algorithm and then we have compared both. The key research findings are as follows:

- ✓ Comparison of Client Server and Mobile Agent based communication models.
- ✓ Proposed a Mobile Agent based secure communication model.
- ✓ Provide security to the information using RSA algorithm and AES algorithm.
- ✓ Compared both the Algorithms.

Using cryptography increases the complexity of the entire scenario. We have compared the performance of both the algorithms (RSA and AES). The insecurity of key is greater in AES. RSA is better for security point of

view as it uses a public and a private key. On the other hand, using RSA can increase the communication delay as it takes more time to encrypt and decrypt the data. AES is relatively fast. So, we can use RSA for exchanging of AES, and encryption and decryption can be done using AES.

5.2 FUTURE SCOPE

This is starting to implement security on agents' data at transport layer. Although the work is a startup of mobile agent based communication that we have furnished in this dissertation, we have planned to expand the work for multi agent based communication and for multiple platforms. We have also planned to work on intrusion detection in agent based environments. Further we have planned to optimize communication in agents based environment regarding fast accessing and security.

REFERENCES

[1] R.bindu, "Mobile Agent Routing Protocol with Security for MANET," *International Journal of Applied Engineering Reasearch, Dindigul*, vol. 1, no. 1, pp. 92-101, 2010.

[2] W.A.Jansen, "Mobile Agents Security," *Journal of Computer applications*, vol. 23, no. 17, pp. 1667-1676, 2000.

[3] H.yang, H.luo, L.Zhang, "Security in Mobile Ad Hoc Networks: Challenges and solution," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 38-47, Febrauary 2004.

[4] W.Jansen, T.Karygiannis, "Mobile Agent Security," *NIST Special Publication*, pp. 800-19, 1999.

[5] T.M.Ahmed, "Increasing Mobile Agent Performance," vol. 6, no. 4, June 2007.

[6] B.Ales, "Client-Server Architecturre," *McGrew-Hill, IEEE-802*, 1992.

[7] G.S.Hura, "Client-Server Computing Architecture:An efficient paradigm for project management," in *Proc. Engineering Management Conference IEEE* , June 1995.

[8] D.B.Large, M.Oshima, "Programming and Developing Java Mobile Agents with Aglets," *Addison-Wesley publication*, 1998.

[9] W.Bing, J.Chen, J.Wu, M.Cardei, "A survey of Attacks and Contermeasures in Mobile Ad Hoc Networks," *Wireless Network Security, Springer US*, pp. 103-135, 2007.

[10] A.Ghaffari, "Vulnerability and Security of Mobile Ad Hoc Networks," in *6th WSEAS International Conference on Simulation, Modeling and Optimization*, 2006.

[11] X.Wu, B.Bhargava, "AO2P: Ad Hoc On-Demand Position -Based Private Routing Protocol," *Mobile Computing*, vol. 4, no. 4, pp. 335-348, 2005.

[12] I.Mahammad, "The Hand Books of Ad-Hoc Wireless Networks-II Series," *CRC Press LLC, USA,* 2003.

[13] L.Zhou, Z.J.Haas, "Securing Ad Hoc Networks," *Special Issue on Network Security,* vol. 13, no. 6, pp. 24-30, 1999.

[14] C.S.R.Murthy, B.S.Manoj, "Adhoc Wireless Networks-II Edition," *Technology and Engineering, Pearson Education of India,* May 2004.

[15] G.Naguyen, T.T.dang, L.Hlucy, M.Lacclavik, Z.Balogh, I.Budinska, "Agent Platform Evaluation and Comparison," in *Institute of Informatics, Solvak Acadamy of Sciences*, Pellucid 5FP IST-2001-34519, June 2002.

[16] D.Horvat, D.Cvetkovic, V.Milutinovic, P.Kocovic, V.Kovacevic, "Mobile Agents and Java Mobile Agent Toolkits," in *Proceeding of 33rd Hawaii IEEE International Conference on System Sciences (HICSS)*, Maui, Hawaii, USA, 2000.

[17] S.S.Mudumbai, J.William, E.Adbelliah , "Anchor Toolkit-ASecure Mobile Agent System," june 2008. [Online]. Available: http://escholarship.org/uc/item/2594j56c.

[18] D.Camacho, R.Aler, C.Castro, M.J.Molina, "Performance Evaluation of Zeus, Jade and Skelton Agent Frameworks," in *System Man and Cybernetics, Vol 2*, Systems, Man and, 2002.

[19] F.Bellifemine, G.Caire, A.Poggi, G.Rimassa, "JADE: A White paper," Available: http://exp.telecomitalialab.com, vol 3, No. 3, 2003. [Online].

[20] S.Vitabile, V.Conti, C.Militello, F.Sorbello, "An Extended JADE-S Based Framework for Developing Secure Multi-Agent Systems," *Computer Standards and Interfaces* , vol. 31, pp. 913-930, 2009.

[21] A.Nagesh, "Distributed Network Forensics using JADE Mobile Agent Framework," Student Trade Show and Project Reports, Arizona State University, 2006.

[22] P.Ahuja, V.Sharma, "A Review on Mobile Agent Security," *International Journal of Recent Technology and Engineering,* vol. 1,

no. 2, pp. 83-88, 2012.

[23] M.Uddin, J.Memon, R.Alsaqour, A.Shah, M.Z.A.Rozan, "Mobile Agent based multi-layer security framework for cloud data centres," *Indian Journal of Science and Technology*, vol. 8, no. 12, pp. 1-10, 2015.

[24] L.Benachenhou, S.Pierre, "Protection of mobile agent with a reference clone," *Computer Communications, Canada,* 2006.

[25] C.lin, V.Varadharajan, "MobileTrust: a trust enhanced security architecture for mobile agent systems," *International Journal of Information Security,* vol. 9, no. 3, pp. 153-178, 2010.

[26] L.Ma, J.J.P.Tsai, "Formal Modeling and Analysis of a Secure Mobile-Agent System," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* , 2008.

[27] S.Venkatesan, C.Chellappan, T.Vengattaraman, P.Dhavachelvan, A.Vaish, " Advanced mobile agent security models for code integrity and malicious availability check," *Journal of Network and Computer Applications,* vol. 33, no. 6, pp. 661-671, 2010.

[28] P.Marikkannu, A.Jovin, T.Purusothaman, " A Secure Mobile Agent System against Tailgating Attacks," *Journal of Computer Science,* vol. 7, no. 4, pp. 488-492, 2011.

[29] S.Venkatesan, R.Baskaran, C.Chellappan, A.Vaish, P.Dhavachelvan, " Artificial Immune System based mobile agent platform protection," *Computer Standards and Interfaces,* vol. 35, no. 4, pp. 365-373, 2013.

[30] S.Venkatesan, C.Chellappan, P.Dhavachelvan, " Performance analysis of mobile agent failure recovery in e-service applications," *Computer Standards & Interfaces,* vol. 32, no. 1-2, pp. 38-43, 2009.

[31] G.Geetha,C.Jayakumar, "Implementation of Trust and Reputation Management for Free-Roaming Mobile Agent Security," *IEEE Systems Journal,* vol. 9, no. 2, pp. 556-566, 2015.

[32] Y. F. T.L.Chen, "An efficient date-constraint hierarchical key management scheme for mobile agents," *An International Journal of*

Expert Systems with Applications, vol. 37, no. 12, pp. 7721-7728, 2010.

[33] X.Hong, " Efficient threshold proxy signature protocol for mobile agents," *An International Journal of Information Sciences,* vol. 179, no. 24, pp. 4243-4248, 2009.

[34] C.Garrigues, N.Migas, W.Buchanan, S.Robles, J.Borrell, ""Protecting mobile agents from external replay attacks," *Journal of Systems and Software,* vol. 82, pp. 197-206, 2009.

[35] S.Srivastava, G.C.Nandi, "Self-reliant mobile code: a new direction of agent security," *Journal of Network and Computer Applications,* vol. 37, pp. 62-75, 2014.

[36] A.Zakerolhosseini,M.Nikooghadam, " Secure Transmission of Mobile Agent in Dynamic Distributed Environments," *Wireless Personal Communications,* vol. 70, no. 2, pp. 641-656, 2013.

[37] H.Idrissi, A.Revel, E.M.Souidi, "Security of mobile agent platforms using RBAC based on dynamic role assignment," *International Journal of Security and its Applications,* vol. 10, no. 4, pp. 117-134, 2016.

[38] T.Wang, S.U.Guan,T.K.Chan, "Integrity Protection for Code-on-Demand mobile agents in e-commerce," *The Journal of Systems and Software,* vol. 60, no. 3, pp. 211-221, 2002.

[39] G.C.Necula, P.Lee, "Safe, Untrusted agents using proof-carrying code," *Lecture Notes in Computer Science, France,* 1998.

[40] C.Garrigues, S.Robles, J.Borrell, G.Navarro-Arribas, "Promotiong the development of secure mobile agent applications," *Journal of Systems and Software,* vol. 83, no. 6, pp. 959-971, 2010.

[41] S. L.Shans, " Mobile Agent Protection with Self-Modifying Code," *Journal of Signal Processing Systems,* vol. 65, no. 1, pp. 105-116, 2011.

[42] D.Shehada, C.Y.Yeun, M.J.Zemerly et al., "BROSMAP: A Novel Broadcast Based Secure Mobile Agent Protocol for Distributed Service Applications," *Security and Communication Networks,* pp. 1-

18, 2017.

[43] T.Grandison,M.Sloman, "A survey of trust in internet applications," in *IEEE Communications Survey Tutorial, Fourth Quarter*, 2000.

[44] H.Reiser, G.Vogt, " Security requirements for management systems using mobile agents," in *Proceeding of the Fifth IEEE Symposium on Computers and Communications*, France, 2000.

[45] F.Bellifemine, A.Poggi, G.Rimassa, "JADE-A FIPA-compliant agent framework," in *Proceedings of Practical Applications of Intelligent Agents*, 1999.

[46] N.Garg, R.P.Mahapatra, "MANET Security Issues," *IJCSNS International Journal of Computer Science and Network Security*, vol. 9, no. 8, pp. 241-246, 2009.

[47] B.Bhatia, M.K.Soni, P.Tomar, "Vulnerability Analysis of Mobile Agents Praxis in Mobile," *International Journal of Computer Applications*, vol. 163, no. 7, pp. 58-65, 2017.

[48] R.Bindhu, "Mobile Agent Based Routing Protocol with Security for MANET," *International Journal of Applied Engineering Reasearch Dindigul*, vol. 1, no. 1, pp. 92-101, 2010.

[49] C.Chowdhury, S.Neogy, "SECURING MOBILE AGENTS IN MANET AGAINST ATTACKS USING TRUST," *International Journal of Network Security & Its Applications*, vol. 3, no. 6, pp. 259-274, 2011.

[50] A.Hijazi, N.Nasser, "Using mobile agents for intrusion detection in wireless ad hoc networks," in *FIPI International Conference on Wireless and Optical Communications Networks*, Dubai, United Areb Emirates, 2005.

LIST OF PUBLICATIONS

[1] M.Kaur, S.Saxena, "A Review of Security Techniques for Mobile Agents", *International Conference on Computing, Communication and Automation* (ICCCA), 5th and 6th May, 2017 (Accepted).

[2] M.Kaur, S.Saxena, "Mobile Agent Based Communication using AES and RSA", *International Conference on Advanced Computational and Communication Paradigms* (ICACCP)**,** Sikkim Manipal Institute of Technology (SMIT), September 2017 (Accepted).

[3] M.Kaur, S.Saxena,"AES compared with RSA on Mobile Agent based communication", *in information security in Biomedical Signal Processing (IGI Global Book Submission)* 2017 (Chapter Proposal accepted).

VIDEO LINK

https://www.youtube.com/watch?v=Nr7Cptce-Ow&t=4s

m

ORIGINALITY REPORT

%11	%2	%9	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Kaur, Jasleen, Sharad Saxena, and Mohd Abuzar Sayeed. "Securing mobile agent's information in ad-hoc network", 2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), 2014. Publication	%3
2	Bhalaji, N., Sinchan banerjee, and A. Shanmugam. "A Novel Routing Technique against Packet Dropping Attack in Adhoc Networks", Journal of Computer Science, 2008. Publication	%2
3	www.omicsonline.org Internet Source	%2
4	Srivastava, Shashank, Divya Kumar, and Shuchi Chandra. "Trust analysis of execution platform for self protected mobile code", 2015 International Conference on Advances in Computing Communications and Informatics (ICACCI), 2015. Publication	%1