Security on Mobile Agent Based Communication System

A Thesis Submitted
In Partial Fulfillment of the Requirements
For the Degree of
Master of Technology
In
Information Technology (Wireless Communication and Computing)

By

Abhishek Pandey
MW200501
under
Prof. R. C. Tripathi

Indian Institute of Information Technology, Allahabad

Indian Institute of Information Technology, Allahabad

July 2007

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Allahabad

(Deemed University)

(A Centre of Excellence in Information Technology Established by Govt. of India)

Date ______________

I/We hereby recommend that this report prepared under my/our supervision by **Abhishek Pandey** entitled "***Security on Mobile Agent Based Communication System***" be presented in the partial fulfillment of the requirements for the degree of Master of Technology in Information Technology Specialization *"Wireless Communications & Computing"* for Examination.

COUNTERSIGNED

____

Prof. R.C. Tripathi

(THESIS ADVISOR)

____

Prof. U. S. Tiwary

DEAN (ACADEMIC)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY

Allahabad

(Deemed University)

(A Centre of Excellence in Information Technology Established by Govt. of India)

CERTIFICATE OF APPROVAL*

The foregoing thesis is hereby approved as a creditable study in the area of Information Technology carried out and presented in a manner satisfactory to warrant its acceptance as a pre-requisite to the degree for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it is submitted.

COMMITTEE ON ______________________________

FINAL EXAMINATION ______________________________

FOR EVALUATION ______________________________

OF THE THESIS ______________________________

*Only in case the recommendation is concurred in

Declaration

This is to certify that this thesis work entitled “***Security on Mobile Agent Based Communication System***” which is submitted by me in partial fulfillment of the requirement for the completion of M. Tech in Information Technology specialization in Wireless Communications and computing to Indian Institute of Information Technology, Allahabad comprises only my original work and due acknowledgement has been made in the text to all other material used.

Abhishek Pandey
M.Tech. IT (Wireless Communications and Computing)
MW200501

Contents

Abstract

Mobile Agent Systems is a new technology it was developed for the communication between different mobile devices they have some good features but security remains the major problem in all Mobile Agent Systems for research point of view.

In mobile based Communication System, there were many problems in network like low bandwidth, slow data rate, data are not secure (because signals being available in open). The subject communication runs into cyber security problems. Our main objective is to provide a highly secure environment that is simple to use and deploy. So that in this project we will use agent base communication which is more effective in mobile communication and improve security for obtaining the goal. We create a new security protocol (that work on like data and code encryption, digital signature) and secure protocol architecture for the mobile communication. In this proposed system Aglets and Java are used as a language tools. In our application we will create a mobile agent which has the ability to find out the contact number for a given person in network using security protocols (which will be designed by us)

Acknowledgements

Before, I get into thick of things; I would like to add a few heartfelt words for the people who were part of my thesis in numerous ways, people who gave unending support right from the beginning. During this period, the faculty members and my batch mates took keen interest and participated actively. They are very efficient and qualified in their respective disciplines.

I express my sincere gratitude to **Prof. R. C. Tripathi, Indian Institute of Information Technology-Allahabad** for all his affectionate encouragement and guidance during the entire Thesis. His views and inputs are very helpful throughout the process.

I would like to thank **Dr. M. D. Tiwari, Hon'ble Director, Indian Institute of Information Technology-Allahabad** for the facilities and environment for research.

Not the least I would like to appreciate the support and guidance of all my Teachers specially Dr. U.S.Tiwary, Mr. Vijay Kumar Chourasiya and my all M.Tech. Friends without whose help and support the thesis could not have been a success.

Lastly I would like to thank my family for their love, support and encouragement that they have given me through the past months, helping me to persevere in my studies.

List of Figures and Tables

Figures

Tables

Chapter 1

Introduction

1.1 Motivation

Today there are so many network technologies using which we can connect computers with each other and spreading information all over the world. For this purpose we can use various distributed computer resources through the computer networks like [1]

- Mobile Client – Fixed Server Model (MC- FA Model)
- Mobile Client – Fixed Agent – Fixed Server Model (MC-FA-FS Model)
- Mobile Clients with Agent- Fixed Server with Agent Model (MCA-FSA Model)
- Peer – Peer Model (P-P Model)
- Mobile Client – Mobile Agent – Fixed Server Model (MC-MA-FS Model)

However, when any user wants to use these resources he must understand the location of distributed resources, predict their current statuses and select some suitable resources. Last one MC- MA- FS model (Mobile Agent) technologies are getting popular as means of an efficient way to access the remote resources on computer networks [1]. Mobile agents, in these technologies are processes that migrate from a node to node in the network autonomously to achieve result for user. The mobile agent provides the result to the user thought the migration of mobile agent system without any knowledge of network environment [2, 15].

1.2 Problem Definition

Today in communication fields research is basically based on the means to provide mobility. If we try to implement mobility then we need modifications in hardware as well as software in existing system. For solving the software problems, a new model is generated, *mobile agent based communication system,* but this system has still some problems.

In mobile agent based communication system there are many problems in networks like low bandwidth, slow data rate and data's are not secure because signals being available in open[4, 11].

1.3 Mobile Agent

Mobile Agent is a kind of program that migrates form one host to another in a distributed network mobile agent has many advantages over existing distributed techniques like resource utilization, reduced network traffic etc. As mentioned in [16, 29] there are following features of a mobile agent.

Features of Mobile Agent

Mobility – With the help of mobility feature mobile agent migrate between node to node, communication processes can be performed in a wireless as well as a fixed network.

Autonomy – In autonomy, mobile agent executes autonomously on the behalf of the some other process.

Communication – Mobile agent has the ability to communicate with another mobile agent, servers (i.e. either mobile or fixed) and other clients.

Learning – One of the most exciting features of mobile agent is its learning ability i.e. it changes the behavior according to its past experience.

Interoperability – Mobile agents have property to operate on different platform or over different clients and adapt to changes in the environment.

Persistence – mobile agent has no need to establish continuous connection for execution of programs i.e. it deals with disconnected operations.

1.4 Benefits of Mobile Agents

When to use Mobile Agents?

Mobile Client – Mobile Agent – Fixed Server Model (MC-MA-FS Model)

A mobile agent is a program or object that migrates from node to node according to the program's objective. At each node, program i.e. mobile agent checks the availability of the resources that he wants. If the resources are available then mobile agent executes the task and then returns back and if resources are not available then simply go to next node or terminates. In this model if a client wants any information form server, client use to generate request and give the detail information about that which it wants. Then mobile agent takes information from client and move to server host. Mobile agent runs on behalf of client to give request to server host and take response of the request and get back to the client. Mean while the client is free for other work or movement because it is the responsibility of mobile agent to take and give information between client and server.

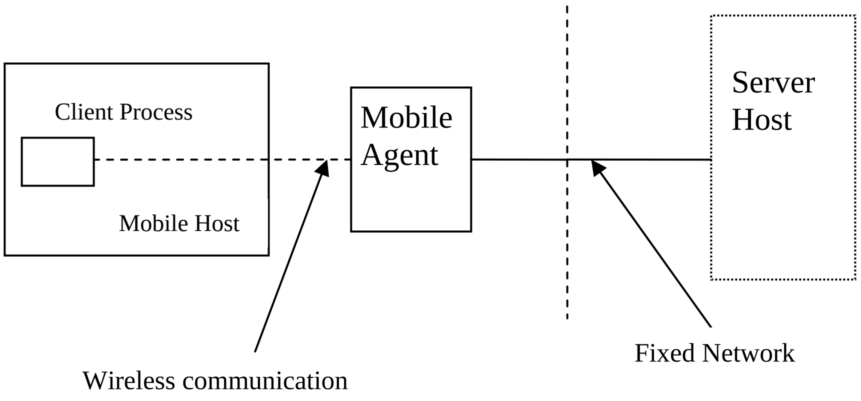


Figure 1

Normally, there are multiple agents running and performing different tasks on the same platform. In this multi-agent system some agents have same property and some have different types. The Mobile Client – Mobile Agent - Fixed Server is the best model in distributed system in wireless and mobile computing environment Mobile agent has ability to run over thin and think client [1, 3].

As discussed in [1] there would be following advantages and disadvantages of Mobile Client- Mobile Agent - Fixed Server model in wireless and mobile computing environment.

Advantages

- MC- MA- FS model has ability to deal with disconnection operation; when mobile agent is executed at host server then that time there is no need for network connection because agent has all information that is related for execution.
- In this model mobile agent optimizes data and services between client and server.
- This model is also applicable on thin client because mobile agent has property to run on thin as well as think client.
- With the help of mobile client we can also manage the problem of weak connectivity.
- Mobile agent works on both i.e. wireless and wired network so there is no need to worry about fixed network connections.

- Mobile clients used to communicate each other directly with the help of wireless networks.
- In this model mobile agent migrates to find resources in the network.

Disadvantages

- MC- MA-FS model is a very sophisticated model.
- There is need to manage the location of the agent.
- In this model a ***possible security thread*** is required.

1.5 Security Requirements

A mobile agent system has the following four security requirements [4]

a) Confidentiality

b) Integrity

c) Availability

d) Accountability

1.5.1 Confidentiality

Confidentiality says that the private data of a mobile agent is stored in agent program or on a platform which must be confidential i.e. reading and writing operation will not be performed on it without the authority. We must secure our data from snooper because they can change the data by new fake agent, thereby they can read confidential information. The mobile agent framework insures that there local and remote communication is confidential. Snooping is performed by external entity that creates fake information for own profit for example, the objective of a mobile agent is to collect the information of air ticket. For this purpose a mobile agent moves different airbus service provider and find out various information like time of airbus and rate of the ticket. Any one of the airbus create a malicious agent that can change the information of the mobile agent like altering the charges of the cheapest airbus for own profit.

Mobile agent uses the proxy technology by which mobile host hides its own location on a particular platform whenever it is required.

All the information of mobile agent’s activities at the platform is stored in the audit logs so it must be confidential and protected form others. Different security domains have different operations on audit logs, sometimes they exchange there information for checking there past history.

1.5.2. Integrity

In mobile agent based communication system security depends on the integrity of the local and remote agent’s operation. In mobile agent based system we ensure that the data, state and code of the mobile platform or agent must be modified by only authorized agent. Mobile agent can not restrict malicious agent or platform to change the state code or data but only detect these changes; however some times it is difficult to detect these modifications

In mobile agent system there must be some access control mechanism that provides restriction for unauthorized users. For these restrictions number of research is still going on. In this case we can also check that security mechanisms, performance and development cost, all must be in favor of mobile agent attribute.

Malicious host and agent are trying to crack the integrity of the system by exchanging the content of intra or inter platform messages. Reusing an old message, deleting the message, and changing the source and destination address all these types of attacks are goal oriented. Agent platforms rely on lower-level protocols to ensure the integrity of the agent communication [4]. Like in case of TCP/IP when any error occurs then communication stack resend that data to ensure the communication integrity.

1.5.3 Accountability

All the process of mobile agent system viz. user, process host and agent must be accountable for their action and they must be uniquely identified, authenticated and audited. An event that is relevant to security is stored in platform security policy and they have a defined information agent name, time of event, security event success and failure of the event.

1.5.4 Availability

One of the A mobile agent system also requires availability, i.e. mobile host must be ensuring that the data and services both are available at remote and local agent. Mobile host must be able to detect problems, these problems are in either in software or in hardware they can also recover data, manage deadlock when problem occurred

In mobile communication, the agent platform must support easy ease of use of data and various services at both remote and local agents. The following are features of mobile platform concurrency control, concurrent access control, deadlock management, and restricted access. Shared data must be available in a usable form, capacity must be available

The mobile host are able to process multiple agent (provide information that he want dispatch agent, destroy agent and so on). In some mobile agent system some agent have to responsibility to maintain the network and provide recovery if any problem occurred

The agent platform is capable to handle huge number of requests of users and remote agents or risk creating an accidental denial of service. In case platform is overloaded or cannot entertain any processing or communication load then platform must generate notification to the agent to indicate that each can not provide expected quality of services to the requesting agents.

There is a need of control mechanism of platform and proper monitoring of available resources to protect them from denial of services attack. A denial of service can attack indirectly on the various agents running on platform and available resources of the platform. Many organization are involved in the process of hosting of an agents, denial of service attack can be a harmful for multiple agents running on same platform or organization.

Chapter 2

Background Study

2.1 Introduction to Mobile Agent:-

Mobile agent is a new technology that makes it much easier to design, implement and maintain distributed system. Mobile agents are capable to decrease network traffic load and supports functions to overcome network latency. To satisfy the user need Mobile agent can implement highly robust and fault-tolerant system [9, 11].

2.1.1 Agent Definition

A *mobile agent* is a program functions on behalf of a user in a distributed environment, and is able of migrate *independently* from one node to other to accomplish the assigned task of user [10, 15].

Mobile agent = agent + mobility

Mobile agent is the combination of Software agent technology and Distributed computing technology. Mobile agents are different from Remote Procedure Call (RPC) i.e. because mobile agents can move continuously from one host to another host and travels based on its own needs and choices. Mobile agents are unlike the common process migration, because the common process migration system can not decide where to go and when to go by itself. However, mobile agents can migrate to any where at any time. Mobile agents are different from Java Applets, since applets can travel only one way from server to client, while mobile agents can move in between the client and the server bi-directionally.[15]

2.1.2 Advantages of Mobile Agent

Following are the advantages, that provide by the mobile agent architecture is better than other distributed architecture [12, 15, 16, 29].

1 They reduce the network load:-

In a distributed system, for completing a task there are multiples number of times information will moves form one node to another node. This is especially true when security measures are enabled. So that the result is a lot of network traffic and size of data will increase because, security thread will add every time when data is sanded. Mobile agents allow us to dispatch all information of the task at a time when agent arrives at destination host the interactions can take place locally rather than transferred over the network. So that Mobile agents are also for removing the raw information (like security thread that is added every time when information migrates) in the network. The objective of mobile agent bases communication is: *move the computations to the data rather than the data to the computations.*

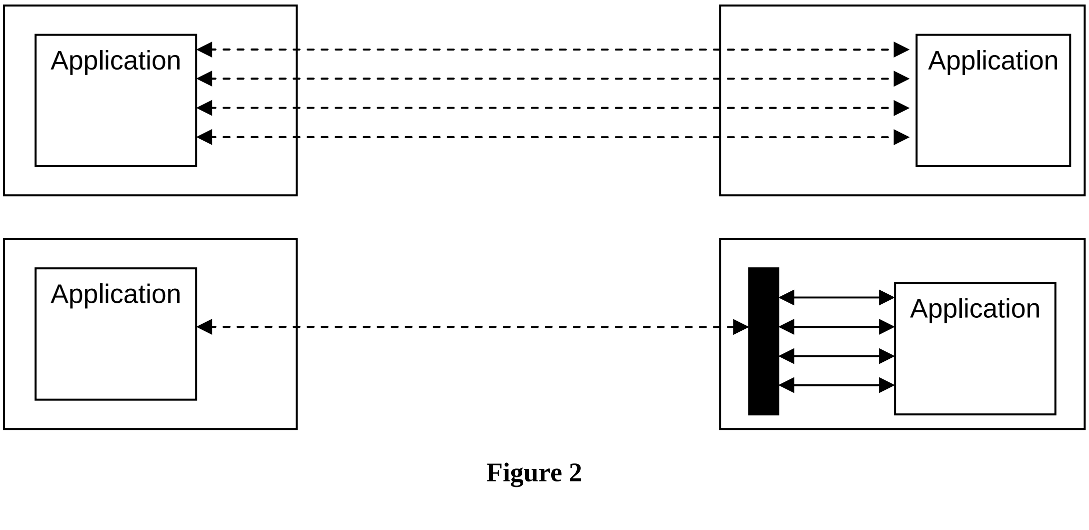


Figure 2

2 They overcoming network latency

Network latency can be reduced by Mobile agent; robots are the example of real time system which responds in real time scenario [4]. So latencies are not acceptable. A mobile agent provides a solution; mobile agent dispatched all information stored on it and at the destination mobile agent act as controller so the all information's are directly access form mobile agent.

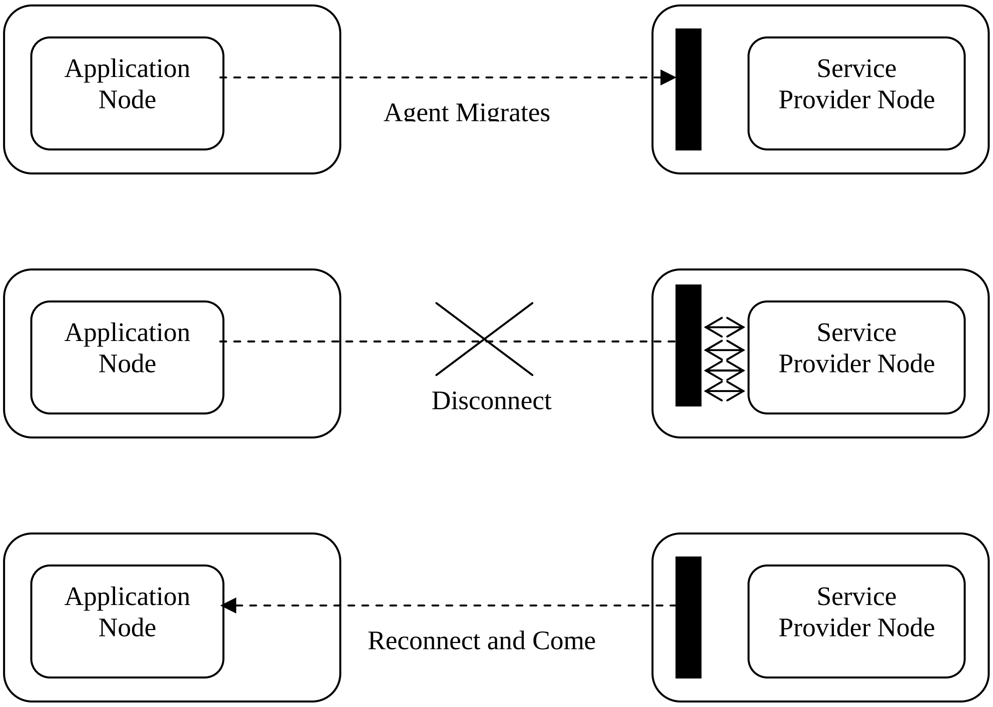


Figure 3

3 they encapsulate protocols

In mobile agent system each host have own information and function by which the mobile agent that will migrate and execute. Mobile agent have both data as well as code this technology is known as encapsulation and to provide the security on this have some rules known as protocol. The interpretation of incoming data and robust coding of out going data are part of basic functioning of protocol.

4 they execute asynchronously and autonomously

One of the most important advantages of mobile agent system is its asynchronous and autonomous execution of agent in wireless networks. Some times mobile devices are expensive for communication or weak network connections or slow speed for communication but some tasks require a continuously open connection between a mobile device and a communication network but its not economically or technically feasible. So with the help of mobile agent we can solve this problem, tasks embedded into mobile agents, and then dispatched it After that ,the mobile agent independent of the creating process and can operate asynchronously and autonomously in this time no need establishment of network connection. After completion of task mobile agent need to agene need to communication network.

5 They adapt dynamically

Mobile agents sense their execution background and react autonomously to changes if found any optimal solution. Mobile agents have the unique feature *cloning* by which they distribute themselves to the hosts in the network by this method they achieve the optimal configuration for solving a problem [29].

6 They are naturally heterogeneous

Mobile agent provides fundamental concept network computing is heterogeneity in both hardware and software. Mobile agents are generally depends on there execution environment and independent from computer- and transport-layer.

7 They are robust and fault tolerant

As mention in fifth advantage the mobile agent adapt dynamically, when a critical problem arrives in network, mobile agent adapt dynamic feature and change his path and provides robust and fault tolerant capability. Before the shut down process all the running agents will get warning message with time to dispatch their execution and resume processing on another connected host [29].

2.1.3 Mobile Agent Systems

There are number's of Mobile agent systems most of them are java based. Some of then are introduce as follow

Aglets:- Aglet was developed by IBM Tokyo Research Lab. Aglets (mobile agents) are java objects that execute on Aglets Workbench (the agent platform like Tahiti server). A developer can use the classes and methods defined in java Aglet API for aglet creation and manipulation. The mobility of the aglet is achieved by the sterilization and dynamic class loading techniques of java. An aglet serializes itself and dispatches to another Aglet Workbench, where it is loaded (executed) by the class loader.

A security model has been defined for the Aglet. Every aglet has an identifier, with the help of them appropriate security policies are applied. However, the system does not enforce the access control based on the aglet's owner. A server trusts the aglets if they were sent from the server in the same domain. The servers within a trusted domain must authenticate each other by using a MAC (Message Authentication Code). Aglets are shielded by proxy objects which provide language level protection as well as location transparency [15, 20]

Concordia is Java-bases mobile agent system and this is developed by Mitsubishi Electric Information Technology Center America. It provides a Concordia Server that executes on the top of the Java Virtual Machine as the agent platform

Three types of protection are available in Concordia security model: 1st protection of agent storage area 2nd protection of transmission of agent and 3rd protection of server resources. In Concordia access control is managed by using SecutiyManager in Java according to the user's identity. The passwords are stored in a global password file, which makes Concordia hard to scale up, if not impossible [15, 26].

Odyssey:- The research of General Magic Inc. developed the first commercial mobile agent system known as Tele-script which are based on network architecture and proprietary language. General Magic re-implements the mobile agent in java-based Odyssey. The above system has been implemented the Tele-script concept in the form of java classes. The result is a java class library that enables developers to create their own mobile agent applications [15].

Voyager:- Voyager is a java based platform for agents to enhanced distributed computing. Voyager supports various objects including object for messaging capabilities and object for movement of agents in the network. Voyager supports various properties of java based object request brokers in addition to mobile agent system properties. A user can create network applications in Voyager based on various agent development techniques it also supports distributed application development [15, 27].

2.1.4 Mobile Agent Standardization MASIF

There are many numbers of mobile agents technology like Aglets, AgentTCl, Odyssey etc. these all are have different architecture and implementations. There must be a standard for mobile agent technology that provides interoperability and system diversity.

Mobile agent system supports an interoperable interface using MASIF (Mobile Agent System Interoperability Facility), various interfaces and definitions are defined in MASIF [17]. Common Object Request Broker Architecture (CORBA) is the Object Management Group's (OMG) language-neutral and platform independent stander for distributed object system; MASIF uses service provided by CORBA.

There are two interface of the MASIF: MAFAgentSystem and MAFFinder; MAFAgentSystem interface provides operation for creation, management and transfer of agents. The interface of MAFFinder supports many operations including location agent's places, registration and un-registration operation and agent systems [29].

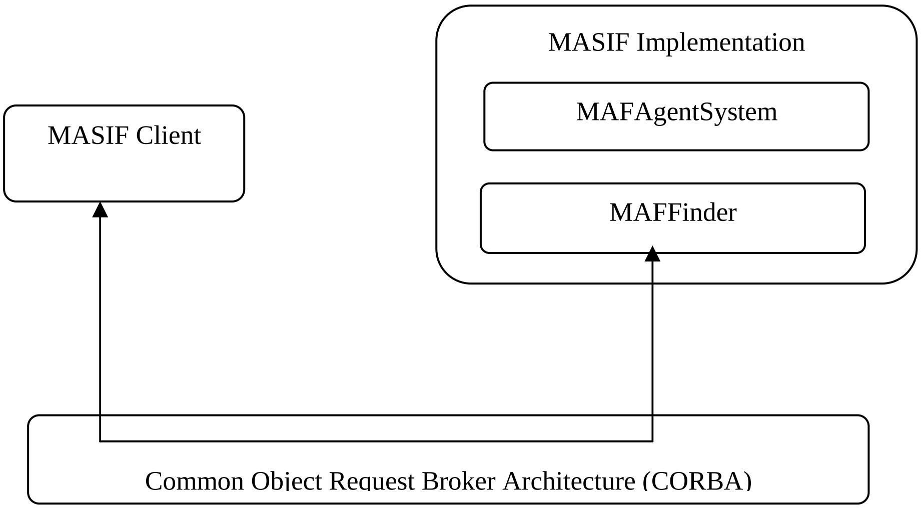


Figure 4

2.2 Mobile Agent Behavior

2.2.1 Creating an agent and Disposing

The place where mobile agent is created is called mobile host. There is another agent that resides on the same mobile host. This agent can initiates the creation process or any other agent system can initiate it. There is a need to authenticate this process by providing authority and information that the new agent will process. The initialization arguments for the new agent are also provided by the creator. As mentioned in the book [29] creation involves three steps.

1. Instantiation and identifier assignments
2. Initialization
3. Autonomous execution

Mobile agent life is ended by disposal initialization by the agent itself or by another agent residing in the same mobile host. An agent can also be disposed of the system for one of the following reasons

- End of life time
- No use
- Security violation
- Shutdown

Disposing of an agent is a two step process

- Preparing for disposal:- before the disposing of the mobile agent, agent give a chance to finalize its current task..
- Suspension of execution:- mobile agent suspends the execution.

2.2.2 Transferring an agent

Mobile agent can move around in any mobile host and this move instruction is provided by mobile agent itself. The transferring process is executed by dispatch process, transfer agent from its current location (origin mobile host) and received by the particular place (destination mobile host)[29].

2.2.3 Dispatching an agent

When we want to move mobile agent from one place to another place then it is necessary mobile agent must be able to identify its destination mobile host. If the destination mobile host is not define then its run in default place that is selected by the destination agent. Mobile agent system provide information that agent want to transfer itself to the destination mobile agent. This message is relayed via an internal API between the agent and agent system. When the agent system receives the agent's trip request, it should do the following [8, 29]

- Suspend the agent
- Serialize the agent
- Encode the serialized agent
- Transfer the agent

2.2.4 Receiving an agent

Before receiving any mobile agent, the static mobile agent determines whether this agent is acceptable or not. For that the static mobile agent want authentication of coming agent. After that dynamic mobile agent provides it's authenticated and if it is correct then they execute that dynamic mobile agent so following steps are takes place.

- Receiving the dynamic mobile agent.
- Decode the dynamic mobile agent.
- Run the dynamic mobile agent.
- Resume execution of dynamic mobile agent.

2.2.5 Agent Class Transfer

The mobile agent cannot resume execution in the destination without its class being present. Following are the ways to make the class available for the destination engine [8,11, 29].

- Class at origin:- if the class is already at destination
- Class at distention
- Code on demand

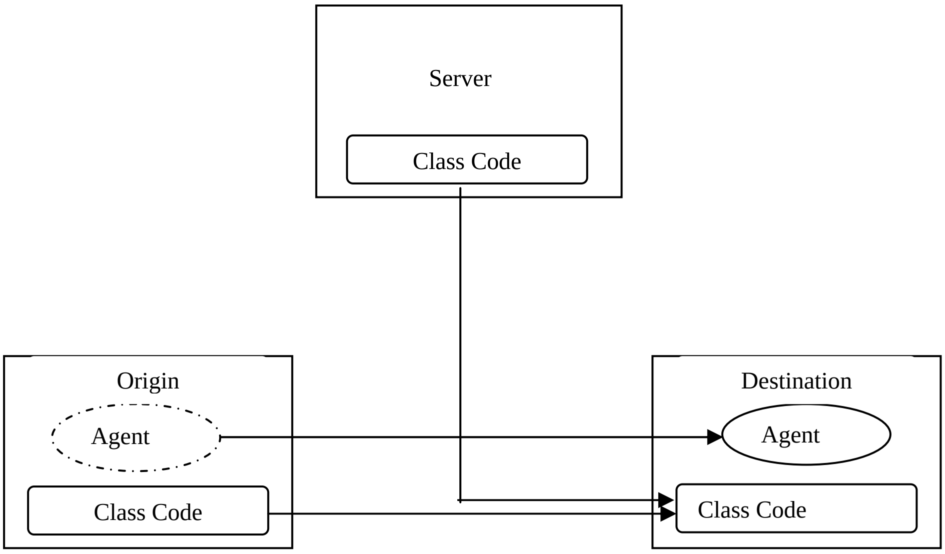


Figure 5

2.2.6 Communication

Agents can communicate with other agents. These other agents may be residing within same place or in other place. Agents messaging are either peer to peer or broadcast. In peer to peer communication mechanism only two hosts exchange information. On the contrary broadcasting is one to many messaging scheme. Broadcast mechanism is useful in multi-agent systems. Following three types of communication schemes [29]

- Now- type messaging
- Future type messaging
- One way type messaging

2 .3 Cryptographic Mechanisms

The recent development of cryptographic techniques makes security in the open environment possible. It provides methods to protect communication as well as to identify communicating parties. In this section we will give an introduction to the cryptography mechanisms used in mobile communication [25, 28].

2.3.1 One-way Hash Function

The one-way hash function is a very important concept in modern cryptography theory. “One-way” means it is easy to generate the code given the message, but nearly impossible to generate the message give the code. A hash function maps function maps an input of varying length into a fixed length output. A one –way hash function maps an input of varying length into a fixed length output. A one way hash function is like a fingerprint maker in that no matter how big the original data might be, we can always produce a unique and fixed size fingerprint. The fingerprint is unique, because given the fingerprint and the message; it is infeasible to produce another message that will produce the same fingerprint. MD5 (Message Digest algorithm 5) is a well known cryptographic hash function with a 128- bit resulting hash value. MD5 designed by Professor ***Ronald Rivest*** of MIT. The details are available in RFC-1321 (Request for Comments) [25].

2.3.2 Public- key Cryptography

In Public-key algorithms we use two keys by one we encrypt and second one is use for decryption these algorithms have the following characteristic

- It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and encryption key.
- A public key cryptography these are the steps
  - Plaintext:- Plaintext is a readable data that will be encrypted.
  - Encryption algorithm:- the encryption algorithms perform various mathematical computations transformation on the plaintext. For encrypt Plaintext.
  - Public and private key:- with the help of these pair of key one of them is use for encryption of data and other one is for decryption of the data.
  - Cipher-text:- this is the scramble message produced as output.
  - Decryption algorithm:- this algorithm accepts the cipher-text and the matching key and produce the original plaintext

2.3.3 Digital Signature

Digital Signature is an authentication technique based on Public key infrastructure. In this authentication method message sender attaches a unique identification code which is called signature. The signature guarantees the source and integrity of the message. Following are the requirements for a digital signature [28].

- It must be a pattern that depends on message being signed.
- It must be easy to produce the digital signature.
- It must be easy to verify and recognize.

2.3.4 Kerberos

Kerberos is a network authentication protocol that provides individuals communicating over an insecure network to prove their identity to one another in a secure manner. Kerberos published and implements by Massachusetts Institute of Technology (MIT) [25]. It design for client-server model and it provides mutual authentication- both the user and the server verify each other's identity [28]. RFC 1510 is an excellent resource for understanding the Kerberos protocol.

Chapter 3

Used Technology

3.1 Aglet

Basic Elements of Aglets

An Aglet is platform which combines *agent* and *applet* applications. The aglet model also supports the applet model. As discussed in [9, 10, 15, 16], following are the basic elements of an aglets.

Aglet: -Aglet is defined to be an autonomous java object that has mobility from one host to another host in a computer network environment. In other words it run's according to its own thread of execution when it is arrived at a host. It also reacts to in coming information as it is its ability.

Proxy: -. A proxy is used to protect aglet from open access. Location transparency is a feature of proxy which hides location of the aglet over the public network. A proxy is a strong representative which is used to maintain data confidentiality in aglets

Context: The workplace of an aglet is context. It is responsible for the management of aglets which are in running mode in a uniform execution environment and it is secure the host from malicious aglets. In computer networks each node is capable to provide functionality of multiple servers and each server can configure multiple contexts.

Contexts can be located over the network by their name which includes context's server address and context name.

Message: - The exchange of messages in aglet knows as object. It can support two types of message passing - synchronous message passing between aglets and asynchronous message passing between aglets. Aglets can exchange information as a message passing in a loosely coupled fashion.

Future reply: - Asynchronous message sending is used as a handler to collect the results for future reply.

Identifier: - An aglet contains an identifier which is unique and immutable throughout the life span of the aglets.

Fundamental operation of Aglets

Following are the fundamental operation of Aglets [29]

Creation: - The aglets are created in context. An identifier is allocated to every new aglet; it is included and initialized in context. Aglets start their execution after the initialization process.

Cloning: - The cloning is a method, by which an aglet generate identical copy of the original aglet at the same context. The only differences are that execution restarts in the new aglet and assign new identifier. *Note that execution threads are not cloned.*

Dispatching: - with the help of dispatching, an aglet moves from one context to another, it will remove from its source context and insert it into the destination context, where aglet will restart execution (execution threads do not migrate). We say that the aglet has been "pushed" to its new context.

Retraction: - The retraction of an aglet is like a pull mechanism, it force the agent to migrate from its current context to its context from which the retraction was requested.

Activation and deactivation: - The property of mobile agent is, temporarily stop the execution of aglet and store its state in secondary memory with the help of deactivation operation. For restore the aglet in a context we use activation operation.

Disposal: - The termination process of an aglet will stop its active execution and also eliminate it from the context.

Messaging: - Aglet are communicated with each other via Messages passing to each other by notifying various operations like transmitting, receiving, and handling synchronous and asynchronous message management.

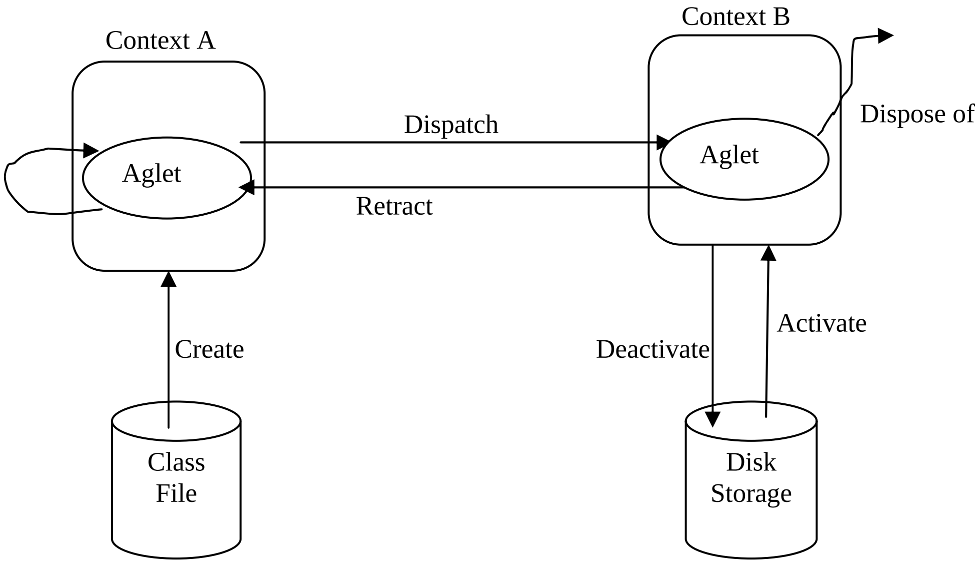


Figure 6

3.2 JDK1.2

The JDK1.2 security architecture provides application verification and location based access control management. This security architecture provides more sophisticated methods to manage security. The access rights are specified by the security policy. The security policy can specify its own set of access rights to each signer and location combination. [21, 30].

3.3 JDBC (Java Database Connectivity)

The JDBC API is a Java API that used at server side to establish a communication between java and database. With the help of JDBC API user can execute SQL statements in java program, retrieve the fetched results and modify data sources to update database

In java JDBC DriverManager object class is used to provide connection between java application and JDBC driver. The DriverManager is very small in size and simple to operate and its very important function in the JDBC architecture JDBC provides these three programming activities [21]

- Connect with data source
- Perform queries and update statements
- Retrieve the result from the database in answer to our query

JDBC includes following components

- The JDBC API: -that provides access method in java by which we access data from relation database management system. The JDBC API can operate in distributed or heterogeneous environment by connecting multiple data source. All API's are either in *java.sql* or *javax.sql.*package.
- JDBC Driver Manager:- JDBC Driver Manager is backbone of the JDBC architecture by it we also connect with java application to JDBC driver.

- JDBC –ODBC Bridge:- with the help of JDBC ODBC Bridge we provide a platform on it JDBC access via ODBC drivers.

3.4 Oracle (Relational Database Management System)

A database is a collection of logically related information that is divided into rows and columns (tabular manner) where each column represents the attribute and rows are the logically related data of the object. Oracle is used at server to store data of the customers and transactions. Oracle allows data to be stored and executed from within the database. Oracle improves the following technical aspects:

Database provides, ease of management, scalability, security, availability and application areas: Internet content management, e-commerce integration, packaged applications, Business Intelligence [22].

Chapter 4

System Designing and Implementation

4.1 Design Criteria

Security measures for mobile agent

Many commercial and research tasks for Mobile Agent Security architecture have been implemented and many are still under development. The security issues in mobile agents are the challenging research areas and also have attracted the attention of the many researchers and security communities.

Problem

In malicious host problem any agent can transfer their processing on the host and user cannot stop the host to protect it from malicious accessibility. A simple example how malicious host can affect any program such as a search agent has been sent out to find the contact number of any person. An agent stores some initial information that is required during the processing, such as name of the person and it is to be sent out to find the contact number of that person. For that result, the agent will have to visit every node and query their data base. A malicious host can interfere by erasing all information collected by the agent or it may provide a wrong number. Some other problems are also with the agent based communication system like malicious agent that also introduces problem in the mobile agent based communication system. Malicious agent can affect the system in the same way as mobile host and it is also a kind of virus that can crash the mobile host.

Some research papers are provides solution for these problems the are as [9, 14]

Contractual agreements:- Under this agreement the operators of agent platform gives assurance by making a contractual agreements to maintain their environment security, confidentiality, integrity and privacy related to its processing data and computation.

Trusted hardware: - The trusted hardware can be implemented to protect from m*alicious host.* Trusted hardware is a tamper resistant devices which installed with the host and make interaction with the agent platform.

Encrypted payload:- Asymmetric (public key) cryptography is frequently used to implement security of a mobile agents, it also generate results and transfer them to the agent's owner. In this case no need to hide encryption key secretly.

Environmental key generation: - An agent can hold encrypted code or secret data with the help of Environmental key generation. The decryption of an encrypted data is possible only when pre decided environmental conditions are defined as a true.

All these are the proposed solution for the protecting our agent with malicious host but they all are not very effective due to some of them are not possible to implement for thin platform as in trusted hardware. Some of them increase the network load if we use existing techniques and some of them are not so secure like environmental key generation, Contractual agreements. So that we are define new security model. In our model, we define a new encryption and decryption technique that did not increase network load and security is implemented for application layer so no need to changes in hardware and network architecture

4.2 Security Model

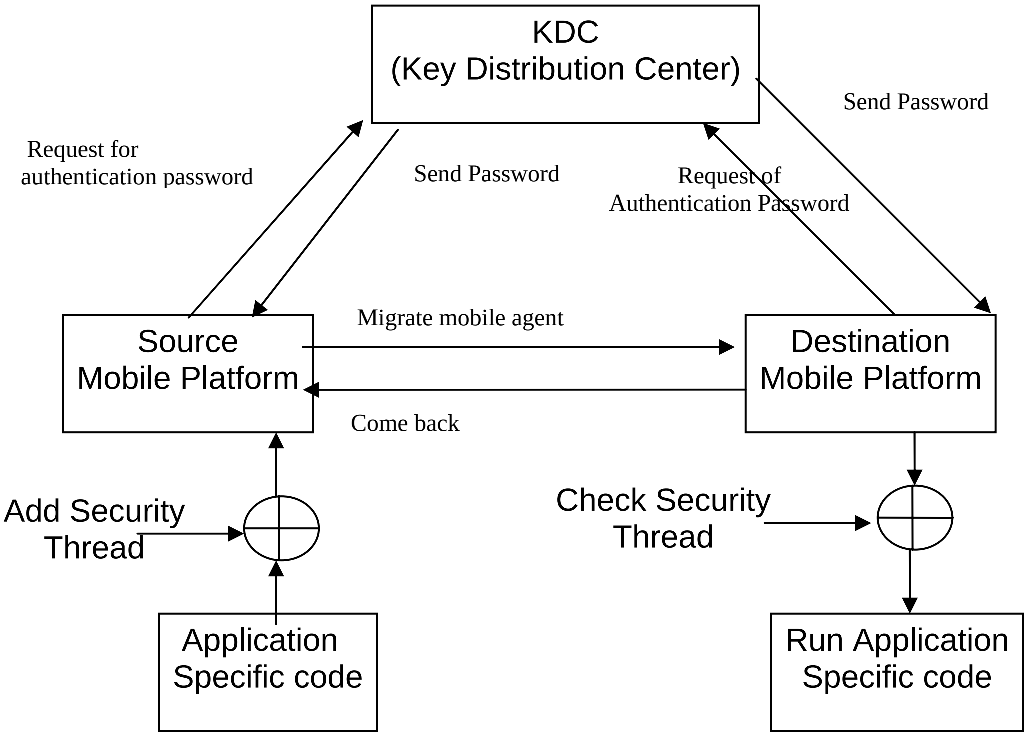


Figure 7

This is my proposed model. In this model we are trying to remove all security problem of mobile agent detailed are as

4.2.1 Application Specific Code:-

First of all we create an application which implements the security issues. In our application we created a mobile agent for finding contact number of a given person with in a network.

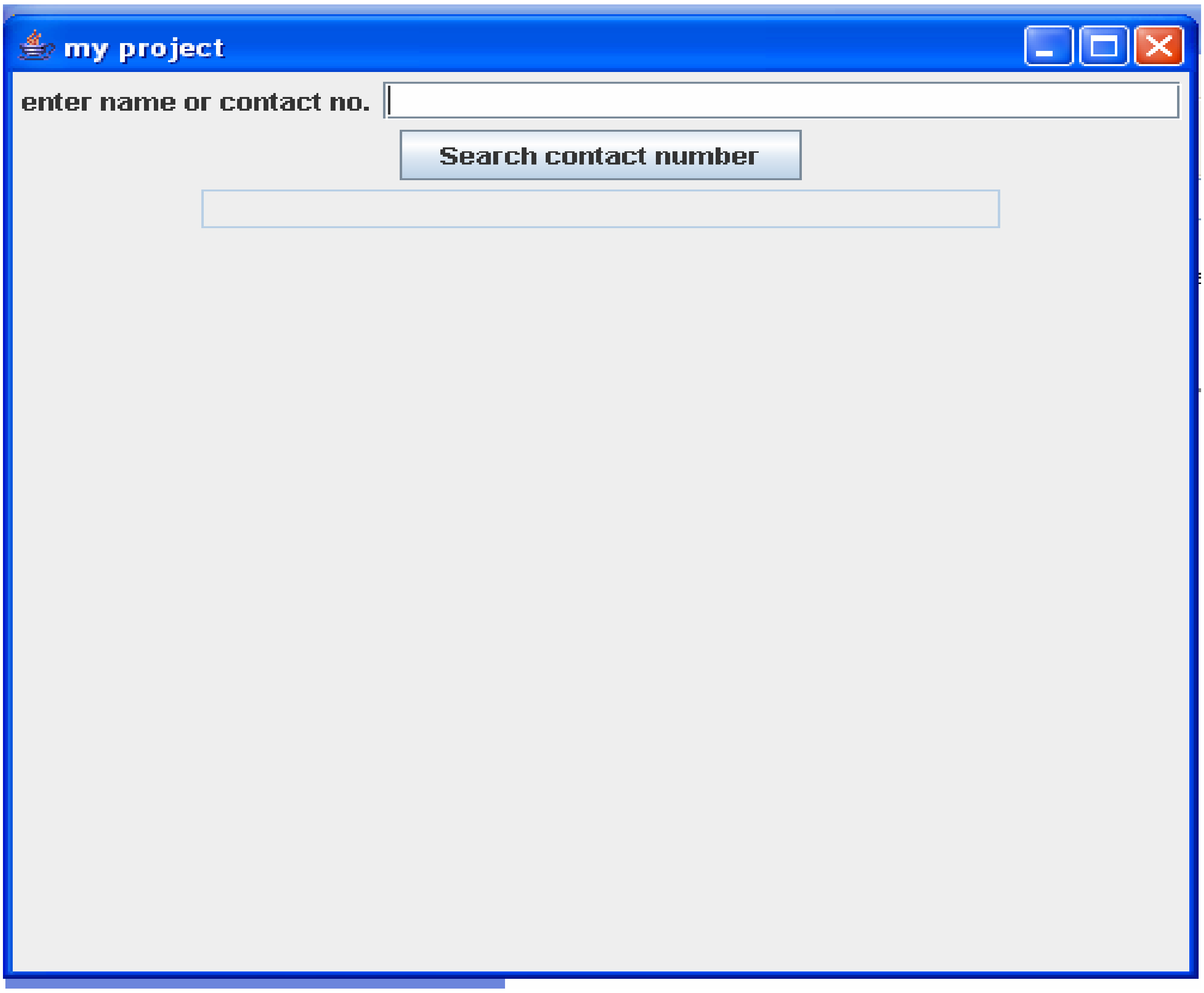
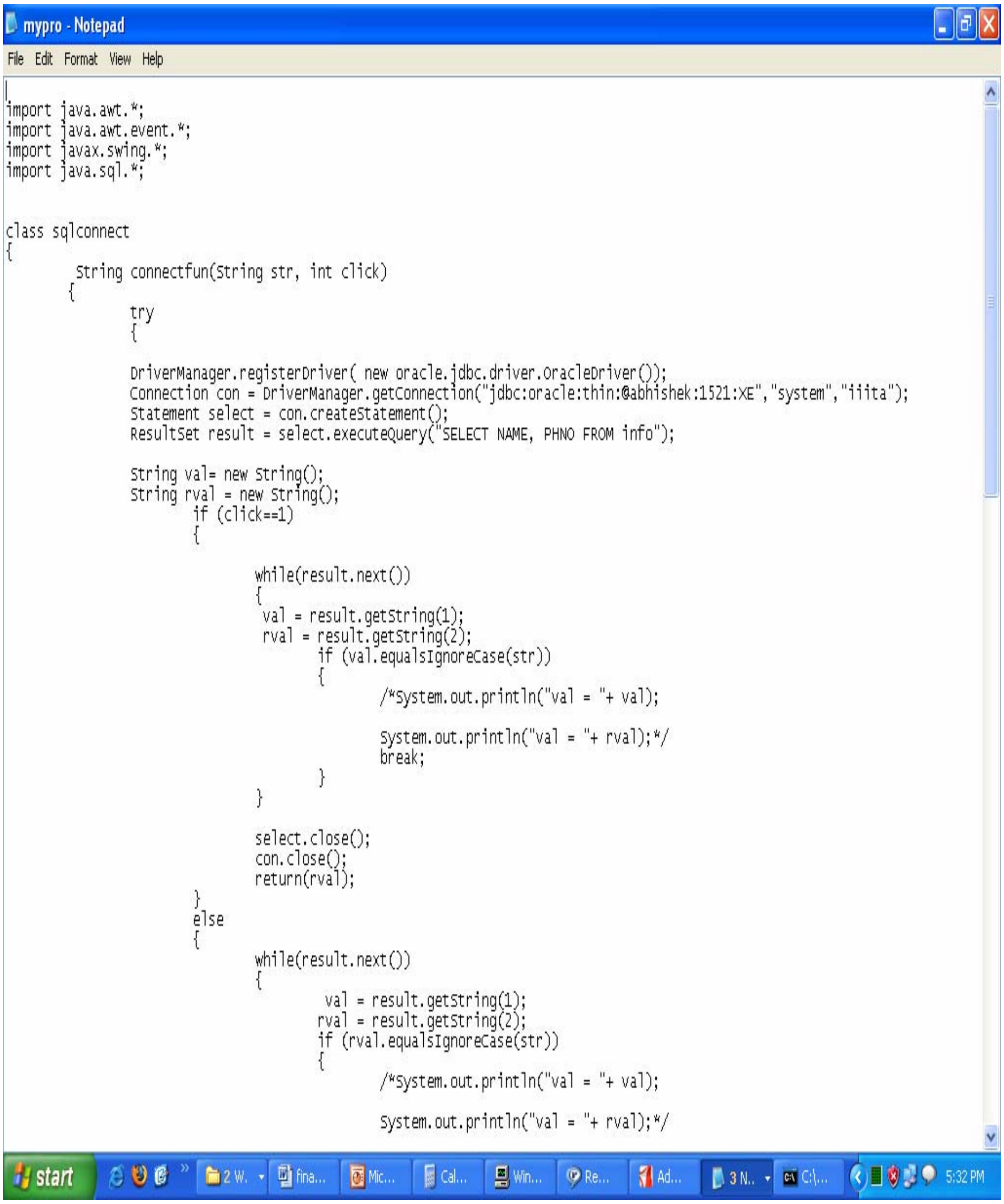


Figure 8

4.2.2 Add Security Thread:-

As we already discussed, our main objective is to protect mobile agent, mobile platform so in this security thread first of all we use MD5 (Message Digest 5) one way hash function. In java JDK1.1 introduced the notion of a *Cryptographic Service Provider or* "provider" for short. This term refers to a package (or a set of packages) that supply a concrete implementation of a subset of the cryptography aspects of the JDK Security API (java.security.MessageDigest). With the help of MD5 we insure that our data or code can not be interrupted (modify) by any Malicious Host or Malicious Agent.

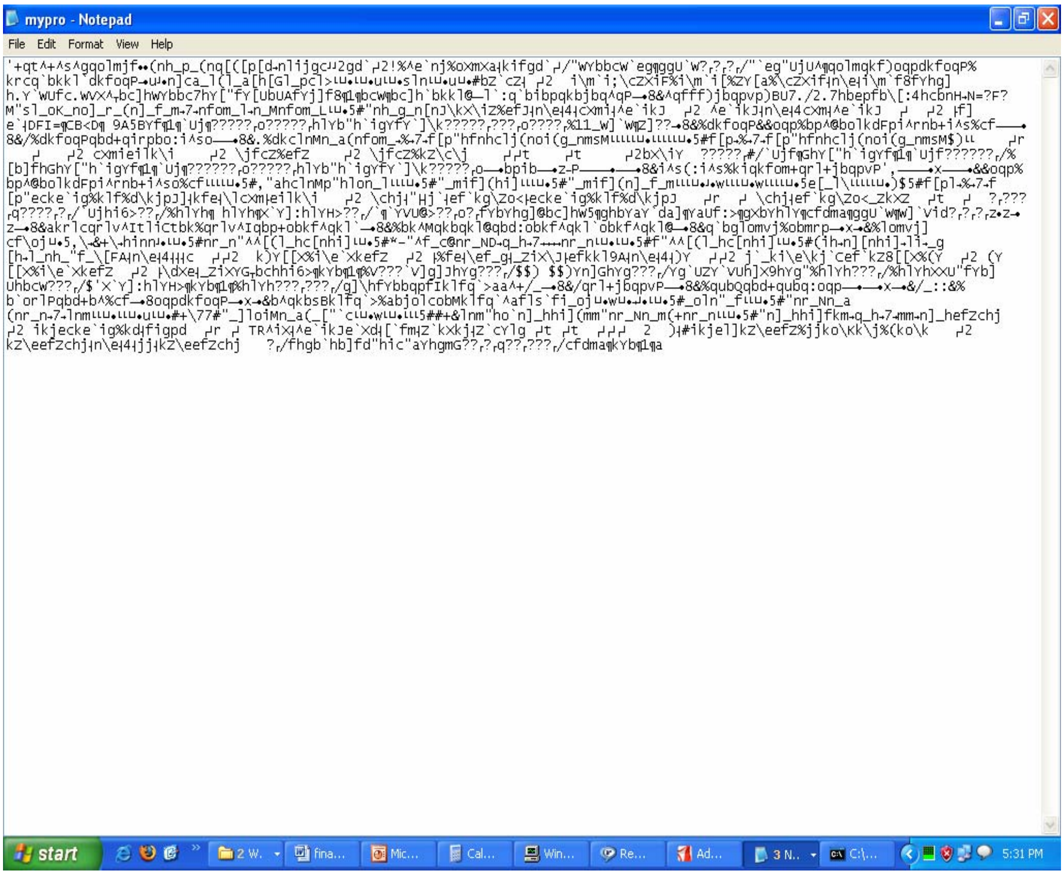
In following figure, this is code file (mypro.java) which will encrypt and then dispatch through mobile agent.



```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.sql.*;


class sqlconnect
{
         String connectfun(String str, int click)
        {
                try
                {

                DriverManager.registerDriver( new oracle.jdbc.driver.OracleDriver());
                Connection con = DriverManager.getConnection("jdbc:oracle:thin:@abhishek:1521:XE","system","iiita");
                Statement select = con.createStatement();
                ResultSet result = select.executeQuery("SELECT NAME, PHNO FROM info");

                String val= new String();
                String rval = new String();
                        if (click==1)
                        {

                                while(result.next())
                                {
                                 val = result.getString(1);
                                 rval = result.getString(2);
                                        if (val.equalsIgnoreCase(str))
                                        {
                                                /*System.out.println("val = "+ val);

                                                System.out.println("val = "+ rval);*/
                                                break;
                                        }
                                }

                                select.close();
                                con.close();
                                return(rval);
                        }
                        else
                        {
                                while(result.next())
                                {
                                          val = result.getString(1);
                                        rval = result.getString(2);
                                        if (rval.equalsIgnoreCase(str))
                                        {
                                                /*System.out.println("val = "+ val);

                                                System.out.println("val = "+ rval);*/
```

Figure 9

After the encryption of above file (mypro.java) is generate new file (mypro.dec) which looks like as



```
'+qt^+^s^gqolmjf••(nh_p_(nq[([p[d→nlijgc┘┘2gd`┘2!%^e`nj%oxmxa┤kifgd`┘/"wYbbcw`eg¶ggU`W?,?,?,/"`eg"UjU^¶qolmqkf)oqpdkfoqP%
krcq`bkkl`dkfoqP→⊔•n]ca_l(l_a[h[Gl_pcl>⊔⊔•⊔⊔•⊔⊔⊔•slnⅼ⊔•⊔⊔•#bZ`cZ┤ ┘2  i\m`i;\cZxiF%i\m`i[%ZY[a%\cZxif┤n\e┤i\m`f8fYhg]
h.Y`wUfc.wVX^,bc]hwYbbc7hY["fY[ubUAfYj]f8¶1¶bcw¶bc]h`bkkl@—l`:q`bibpqkbjbq^qP—•8&^qfff)jbqpvp)BU7./2.7hbepfb\[:4hcbnH→N=?F?
M"sl_oK_no]_r_(n]_f_m→7→nfom_l→n_Mnfom_L⊔⊔•5#"nh_g_n[nJ\kX\iZ%efJ┤n\e┤4┤cxmi┤^e`ikJ  ┘2 ^e`ikJ┤n\e┤4cxm┤^e`ikJ  ┘  ┘2 ⊦f]
e`┤DFI=¶CB<D¶ 9A5BYf¶1¶`Uj¶?????,o?????,hlYb"h`igYfY`]\k?????,???,o????,%11_W]`W¶Z]??→8&%dkfoqP&&oqp%bp^@bolkdFpi^rnb+i^s%cf——•
8&/%dkfoqPqbd+qirpbo:i^so——•8&.%dkclnMn_a(nfom_→%→7→f[p"hfnhclj(noi(g_nmsM⊔⊔⊔⊔•⊔⊔⊔⊔•5#f[p→%→7→f[p"hfnhclj(noi(g_nmsM$)⊔    ┘r
  ┘   ┘2 cxmieilk\i    ┘2 \jfcZ%efZ    ┘2 \jfcZ%kZ\c\j     ┘┘t     ┘t      ┘2bX\iY  ?????,#/`UjfqGhY["h`igYf¶1¶`Ujf??????,/%
[b]fhGhY["h`igYf¶1¶`Uj¶??????,o?????,hlYb"h`igYfY`]\k?????,o—•bpib—•z-P———•———•8&i^s(:i^s%kiqkfom+qrl+jbqpvP',———•x———•&&oqp%
bp^@bolkdFpi^rnb+i^so%cf⊔⊔⊔•5#,"ahclnMp"hlon_l⊔⊔⊔•5#"_mif](hi]⊔⊔⊔•5#"_mif](n]_f_m⊔⊔⊔•┘•w⊔⊔⊔•w⊔⊔⊔⊔•5e[_l\⊔⊔⊔⊔•)$5#f[pl→%→7→f
[p"ecke`ig%klf%d\kjpJ]┤kfe┤\lcxm⊦eilk\i   ┘2 \chj┤"Hj`┤ef`kg\Zo<⊦ecke`ig%klf%d\kjpJ   ┘r  ┘ \chj┤ef`kg\Zo<_ZkXZ  ┘t  ┘  ?,???
,q????,?,/`Ujhi6>??,/%hlYh¶ hlYh¶X`Y]:hlYH>??,/`¶`YVU@>??,o?,fYbYhg]@bc]hw5¶ghbYaY`da]¶YaUf:>¶gxbYhlY¶cfdma¶ggU`W¶W]`Vid?,?,?,z•z→
z—•8&akrlcqrlv^Itlictbk%qrlv^Iqbp+obkf^qkl`—•8&%bk^Mqkbqkl@qbd:obkf^qkl`obkf^qkl@—•8&q`bglomvj%obmrp—•x→&%lomvj]
cf\oj⊔•5,\→&+\→hinn┘•⊔⊔•5#nr_n"^^[(l_hc[nhi]⊔⊔•5#*-"^f_c@nr_ND→q_h→7→→→nr_n⊔⊔•⊔⊔•5#f"^^[(l_hc[nhi]⊔⊔•5#(ih→n][nhi]→li→_g
[h→l_nh_"f_\[FA┤n\e┤4┤┤┤c  ┘┘2  k)Y[[X%i\e`XkefZ  ┘2 ⊦%fe┤\ef_g┤_ZiX\J⊦efkkl9A┤n\e┤4┤)Y  ┘┘2 j`_ki\e\kj`cef`kZ8[[X%(Y  ┘2 (Y
[[X%i\e`XkefZ  ┘2 ⊦\dXe┤_ZiXYG,bchhi6>¶kYb¶1¶%V???`V]g]JhYg???,/$$) $$)Yn]GhYg???,/Yg`UZY`VUh]X9hYg"%hlYh???,/%hlYhXXU"fYb]
UhbcW???,/$'X`Y]:hlYH>¶kYb¶1¶%hlYh???,???,/g]\hfYbbqpfIklfq`>aa^+/_—•8&/qrl+jbqpvP——•8&%qubQqbd+qubq:oqp——•——•x—•&/_::&%
b`orlPqbd+b^%cf—•8oqpdkfoqP—•x→•&b^qkbsBklfq`>%abjolcobMklfq`^afls`fi_oj⊔•w⊔•┘•⊔⊔•5#_oln"_f⊔⊔•5#"nr_Nn_a
(nr_n→7→lnm⊔⊔•⊔⊔•⊔⊔•#+\77#"_]loiMn_a(_["`c⊔⊔•w⊔⊔•⊔⊔5##+&lnm"ho`n]_hhi](mm"nr_Nn_m(+nr_n⊔⊔•5#"n]_hhi]fkm→q_h→7→mm→n]_hefZchj
┘2 ikjecke`ig%kd┤figpd  ┘r  ┘ TR^iX┤^e`ikJe`Xd┤[`fm┤Z`kXkj┤Z`cYlg ┘t  ┘t   ┘┘┘  2  )┤#ikjel]kZ\eefZ%jjko\Kk\j%(ko\k    ┘2
kZ\eefZchj┤n\e┤4┤jj┤kZ\eefZchj   ?,/fhgb`hb]fd"hic"aYhgmG??,?,q??,???,/cfdma¶kYb¶1¶a
```

Figure 10

After that we use encryption technology, for that we created own algorithms. This algorithm is derived by the system date. In my algorithms for encryption and decryption keys are identical but it will change when date is change. Basically we create two keys first for find out the block size of the encrypting data and second key is by which we can change the data. Following is the methods by which we generate these keys.

a) Mention our keys are date dependent so first of all we assume that current date is 23.6.2007 and encryption key is generated by this way

$=2^0*7+2^1*0+2^2*0+2^3*2+2^4*6+2^5*3+2^6*2$

=469

Then we divide it by 20 and find out the reminder, if reminder is less than 10 then we add 10 (because in my algorithm block size must be between 10 to 20)

469%20 =9

So we add 10 on it and this become 19.

So first block is 19 character and 2nd is 6 more than previous one.

So variable block sizes are 19,25,31,37 …………

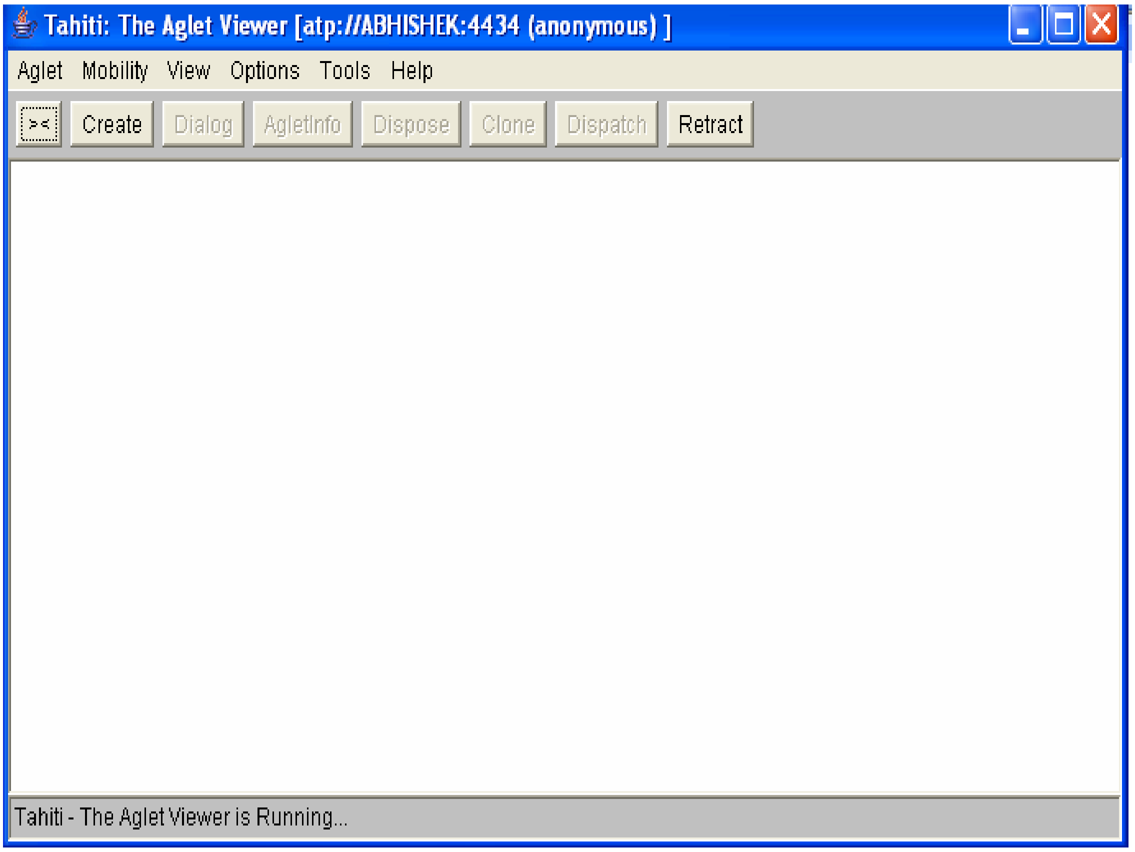
b) Then reverse the content of each block and calculate the encryption key by the following methods

23.6.2007
2+3+6+2+0+0+7
=20 = 2+0
=2

i.e. ASCII code of each character of a first block is incremented by 2 and after completion of every block our key is increment by 4. One think is also important if our key is exceeding by 20 then we again start increment by original key that is 2 in this example. Means in this case our key’s values are 2, 6, 10, 14, 18, 2, 6 …

4.2.3 Source Mobile Platform:-

In our thesis we can use Aglet for mobile agent communication and our platform is Tahiti. Tahiti server is providing GUI (Graphical User Interface) that is used as default user interface to the user. Tahiti presents a main window with a menu bar a list of running agents and toolbar.



Tahiti - The Aglet Viewer is Running...

Figure 11

On the mobile Platform, there is static mobile agent by which we add Security thread in Dynamic mobile agent. So static mobile agent have following tasks

- Find out authentication key for dynamic mobile agent
- Encrypt dynamic mobile agent (Algorithms describe above)
- Find out MD5 of Dynamic mobile agent
- Create a GUI platform by which dynamic mobile agents authenticate each other and check that authentication.
- Migrate dynamic mobile agent to a given destination
- And last display result which will get by mobile agent.

4.2.4 KDC (Key Distribution Center):-

KDC is just like a central server that provides the authentication passwords for all mobile agents. Key Distribution Center; generate nine characters (bytes) password that is distributed to all mobile agents, by using public encryption (digital signature). When mobile agent move form one platform to another platform both static and dynamic both agent will authenticated by exchanging at password. Dynamic agent provide 1,3,5,7,9th location character from that nine character and static mobile agent provides 2,4,6,8th location character form that nine character password. Because both have that nine character password then both are able to check its validity. Following is the algorithm by which we generate that nine byte password and assume that current date is “23.6.2007".

- First of all we fetch each digit of date from left to right and out prime no in sequence from 101and multiply it with digit of date so the equation is like

  101* 2 + 103*3 + 107*6 + 109*2 + 113*0 + 137*0 + 137*7

  =202+309+642+218+0+0+959

  =2330

  Then find out sum of digit and if digit is more than 10 then again find out sum of those digits. So

  2+3+3+0 = 8

So first digit of that password is “8”.

- For finding the second character, each digit of date is subtract by 9

  9-2 , 9 -3 , 9-6 , 9-2 , 9-0 , 9-0 ,9-7

  =7, 6, 3,7,9,9,2

  Then first to last (left to right) each is added by 0,10,20,30,40,50,60 and then add it and at last find out sum of sum

  7+16+23+37+49+59+62

  =253%26 = 9

  By this remainder we find the 9th character is “J” so this is 2nd character of my password.

- In date find out difference between 1 digit by its next digit after that find out sum of those numbers

1+3+4+2+0+7 = 17

If sum is more than 10 then again find out sum of digit.

1+7=8

So third character of our password is “8”

- For 4th character of password we use following method

  2^n *date_digit (n)

  Where n is 1 to 8 and date_digit (n) is 1st location of digit 2nd location of digit ------nth location of digit

  So.

  $2^1*2 + 2^2*3 + 2^3*6 + 2^4*2 + 2^5 *0+ 2^6 *0+ 2^7*7$

  =4+12+48+32+0+0+896

  =992

  Divide this number by 26 and find out the remainder and its equivalent character

  =992%26 => 4

  Character of 4th location is “E”

  So 4th character of password is “E”.

- For finding the 5th character of password we multiply each digit of date by sequentially odd numbers and then add that numbers

  3*2 + 5*3 + 7*6 + 9*2 + 11*0 + 13*0 + 15*7

  =6+15+42+18+0+0+105

  =186

  After that divide this number by 26 and find-out remainder. Then subtract that number by 26 and then finds its equivalent number.

  186%26 = 4

  26-4 = 22 its equivalent character is “W”

  So the 5th character of password is “W”

- For finding 6th character of password simply concatenate date and year and then after divide it by 23 and find outs its equivalent character

  232007%23 = 6

Its equivalent character is “G”
So 6th character of password is “G”

- For finding 7th character of the password, reverse date month and year and sun it
  32+6+7002
  =7040
  Then find out sum of the digits while it not less than 10
  7+0+4+0 = 11
  1+1 = 2
  So 7th character of a password is “2”.

- For finding the 8th character of password, first of all find out the sum of day month and year and then multiply them
  2+3 *6 * 2+0+0+7
  =6*6*9 = 324
  Divide this number by 10 and find-out remainder
  324%10 = 4
  So the 8th character of the password is “4”.

- For finding the last character of the password, simply add first digit of date by last digit of date, second digit of date by second last digit of date and so on then multiply them
  (2+7)* (3+0)* (6+0)* 2
  =9*3*6*2
  =324
  Then reverse it. Divide it by 26 and find out the remainder
  423%26 = 7
  its equivalent character is “H”
  so the 9th character of password is “H”

All these characters of password are stored in an array (8 J 8 E W G 2 4 H)
Finally this is a password which is provided by KDC.

4.2.5 Destination Mobile Platform:-

This Mobile Platform is a destination for dynamic mobile agent when a dynamic mobile agent arrive at Mobile Platform2, on this platform there is a static mobile agent that have a program for checking the authentication of that mobile agent and mobile agent also check the authentication of Mobile platform2 by provided authentication (which is distributed by KDC) of static mobile agent. Static mobile agents at mobile Platform2 also read security thread of the mobile agent and check it is correct (which is distributed by KDC) and then execute it.

4.2.6 Check Security Thread:-

Before the execution of the application, check security thread read information about the mobile agent that is

- ❖ Decrypt dynamic mobile agent
- ❖ Find out the MD5 of the coming mobile agent and check this is same or no.
- ❖ If yes then execute other wise ignore that mobile agent (dispose)

4.3 Testing

4.3.1 Encryption and Decryption Technique

In general encryption and decryption techniques are perform following steps

- ❖ Take simple text (plaintext) which will be encrypted and divide it in blocks
- ❖ Create any encryption technique (algorithms)
- ❖ Pass secret key (cipher-text) as input of the encryption algorithm then it will generate encrypted information
- ❖ In decryption algorithms we pass secret key as input and then it generate decrypted file which is same as simple file.

Like RC5 encryption block of plaintext of length 32, 64, 128 bits into blocks of cipher-text of the same length. [24]

But in our technique, no need for cipher-text because in mobile communication band with is low and we tried to reduce network traffic load so if we remove cipher-text then we reduce up to 50% of data load from network because RC5 have same length for plaintext as well as cipher-text. So that, we use dynamic blocking technique in our algorithm block size is calculated by the help of system date. Detail of the algorithm is already discussed.

4.3.2 Authentication Password

As mention above when a mobile agent dispatch from one platform to another platform we check that platform as well as agent both must be valid so that we use nine byte an authentication password. Our model is more secure and feasible because of following.

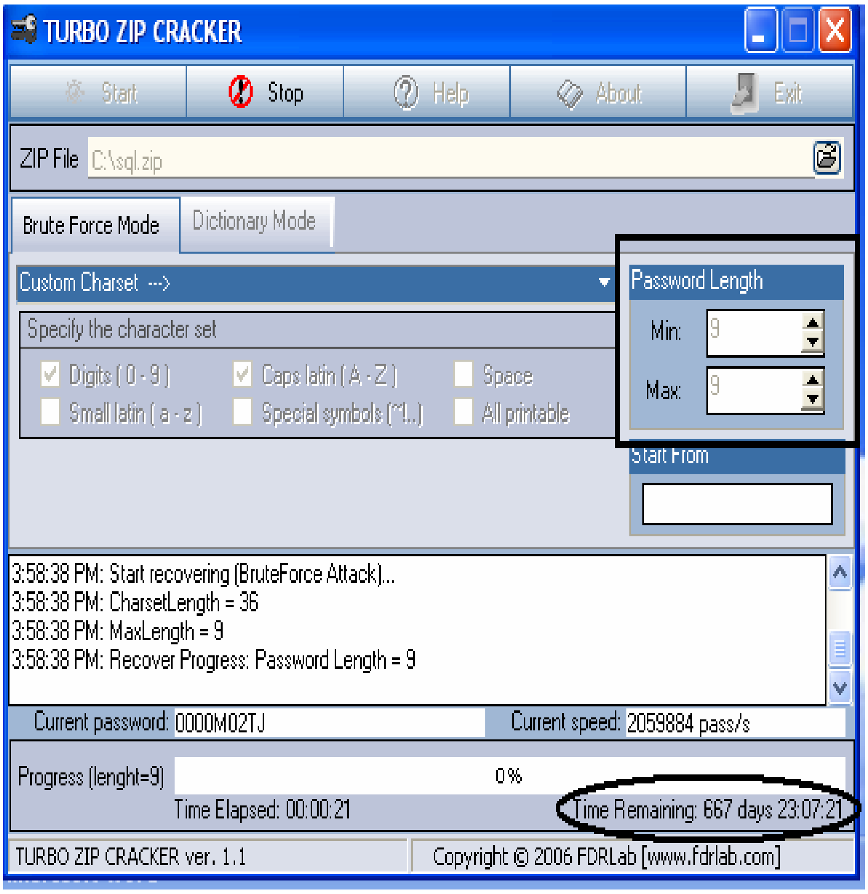
- ❖ Every mobile agent must have a nine byte an authentication password and this password is generated by KDC. That password is dynamically changed in every so each mobile agent must update his password every day and once mobile agent obtain password then no need to find password on that day. And this is secure because following are the method by which cracking is possible [24, 26].
    - ✓ **Dictionary attack:-** a dictionary attack exploits the tendency of people to choose weak password and usually this password is related to themselves in some way like birth place, friend’s name, automobile licenses number and so on. In this case person creates a dictionary in that he stores all information and the cracker crack password by this dictionary. So this technique is not effective in this case.
    - ✓ **Brute Force Attack:** - in this technique cracker try every possible password. A brute force attack will always be successful since password will crack but in Brute Force technique is slow means if we want to find out nine byte password then it take more than 600 days (by testing) but our

password will change in every day so that our password is also secure by this attack.

We use Turbo ZIP Cracker for the testing first of all we found the authentication password by above KDC algorithm and this password is apply to ZIP file protection and trying to crack it by Turbo ZIP Cracker

As shown in the figure-12 and table2 [24, 26] brute force attack is also not feasible for cracking this password

- ✓ in last technique is algorithm is public and weak encryption and decryption technique are used then cracker can crake the password but our algorithm is not public and KDC is responsible to distribute password not algorithm.



ZIP File C:\sql.zip

Specify the character set

☑ Digits (0 - 9) ☑ Caps latin (A - Z) □ Space
□ Small latin (a - z) □ Special symbols (!@..) □ All printable

Password Length

Min: 9

Max: 9

Start From

```
3:58:38 PM: Start recovering (BruteForce Attack)...
3:58:38 PM: CharsetLength = 36
3:58:38 PM: MaxLength = 9
3:58:38 PM: Recover Progress: Password Length = 9
```

Current password: 0000M02TJ Current speed: 2059884 pass/s

Progress (lenght=9) 0%

Time Elapsed: 00:00:21 Time Remaining: 667 days 23:07:21

TURBO ZIP CRACKER ver. 1.1 Copyright © 2006 FDRLab [www.fdrlab.com]

Figure 12

Key Size (bits)	**Number of Alternative keys**	**Time required at 1 decryption /µs**	**Time required at 10^6 decryptions/ µs**
32	$2^{32} = 4.3*10^9$	2^{31} µs = 35.8	2.15 milliseconds
56	$2^{56}=7.2*10^{16}$	2^{55} µs = 1142 years	10.01 hours
128	$2^{128}=3.4*10^{38}$	2^{127} µs= $5.4*10^{24}$ years	$5.4*10^{18}$ years
168	$2^{168}= 3.7*10^{50}$	2^{167} µs = $5.9*10^{36}$ years	$5.9*10^{30}$ years
26 Character permutation	$26! = 4 *10^{26}$	$2*10^{26}$ µs = $6.4*10^{12}$ years	$6.4*10^6$ years

Table 1

Chapter 5
Conclusions

5.1 Contributions

In this thesis, we have proposed the framework of security architecture for Mobile agent based communication, along with partial implementation of the system. The design is quite general, so that it can be easily integrated into other mobile agent systems. More over our focus is over an application-layer security for agent based communication to provide end to end authentication and data confidentiality between mobile agents.

We have suggested two- way authentication protocol to authenticate mobile agents. This solution can be implemented in aglet (Java based technology) without any changes in underlying protocols and mobile agent communication infrastructure.

5.2 Future Works

The security architecture which we have made is still partial in its implementation with respect to programming i.e. code writing. It is due to the fact that the security in mobile agent is too comprehensive and large to get complete in a master's thesis. As our thesis work is associated to security algorithms it is generally based on the relevance of current system's date, so it might have substantial dependency on date. It would be the challenge of present time to put emphasis and endeavor for completing this sophisticated task as well as to minimize the date dependency.

Appendix A
Aglet Source Development kit

In this thesis we use Aglet’s Source Development kit 2.0.2 for Mobile Agent System. This is open source. Few details about the aglet technology are presented here including how to install Aglet’s, how to run the Tahiti Server.

Aglet’s Software Development Kit was originally created by IBM Tokyo Research Laboratory. This is a open-source so it is available at “http://sourceforge.net/projects/aglets/” detailed about installation process given in the user manual which is also available in same location [18].

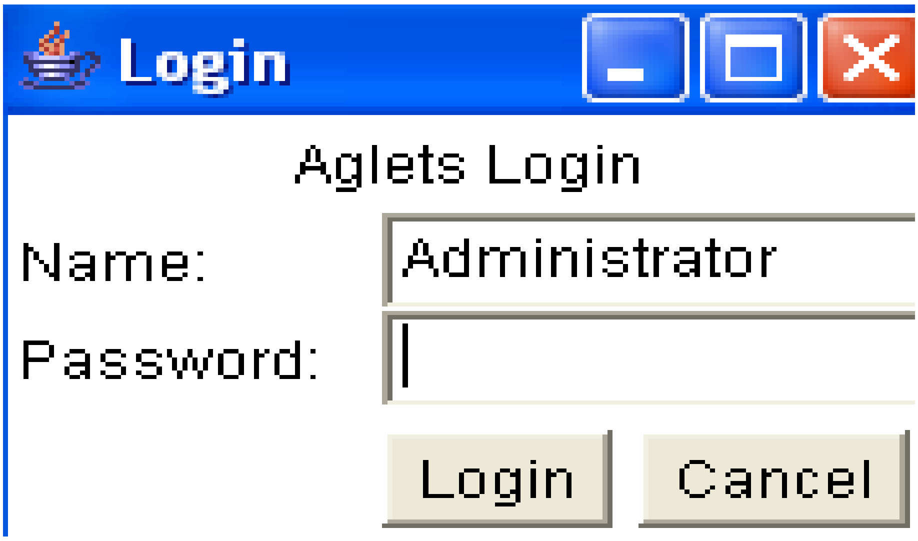
To install the ASDK there are following steps

1) We need Java 2 Run-time Environment (JRE), even if it recommended installing the full Java 2 Source Development Kit (J2SDK) which allows compiling agents in Aglets platform.
2) After downloading the Aglet’s we found aglets-2.0.2.jar file.
3) Decompress the aglets-2.0.2.jar file by the *jar xvf aglets-2.0.2.jar* command.
4) Once decompress that file we get set of sub directories like *bin ,cnf, public lib,* and some file like README and INSTALL.
5) After the move bin folder and run *ant.bat* batch file. This batch file generates a *agletsd.bat* file
6) Then modify the path environment variable and create some new environment variable

```
AGLETS_HOME = c:\aglets
AGLETS_PATH = %AGLETS_HOME%
PATH=%PATH%;\%AGLETS_HOME%\bin
Classpath=%classpath%;c:\aglets\lib\aglets-2.0.2.jar
```

7) Then run *agletsd.bat* batch file for loading the Tahiti Server

8) And at last come following screen in which you provide user name and password by default user name is *anonymous* and password is *aglets* if you change the password and it is necessary, the length of password must be 6 characters long.



Name: Administrator

Password:

Figure 13

Appendix B

Existing Mobile Agent Systems

System	**Organization**	**URL**
Aglet	IBM	http://www.trl.ibm.com/aglets/
AgentSpace	INESC	http://berlin.inesc.pt/agentspace/main-eng.html
Ajanta	Univ. of Minnesota	http://www.cs.umn.edu/Ajanta/
Ara	Univ. of Kaiserslautern	http://wwwagss.informatik.uni-kl.de/Projekte/Ara/index_e.html
Concordia	Mitsubishi	http://www.merl.com/projects/concordia/WWW/index.html
D’Agent	Dartmouth College	http://agent.cs.dartmouth.edu/
JATLite	Stanford Univ.	ftp://java.stanford.edu/JATLite/
Jumping Beans	Aramira Co.	http://www.jumpingbeans.com/
MESSENGERS	UC Irvine	http://www.ics.uci.edu/~bic/messengers/
MOLE	Uni. Of Stuttgart	http://mole.informatik.uni-stuttgart.de/?N=D
OAA	SRI International Inc.	http://www.ai.sri.com/~oaa/
SOMA	Univ. Of Bologna	http://www-lia.deis.unibo.it/Research/SOMA/
TACOMA	Cornell Univ.	http://www.tacoma.cs.uit.no/
Voyager	ObjectSpace Inc	http://www.objectspace.com/products/voyager
WAVE	Univ. Of Surry	http://www-zorn.ira.uka.de/wave/

Table 2

Appendix C

Abbreviations

MC- FA Model	Mobile Client – Fixed Server Model
MC- FA-FS Model	Mobile Client – Fixed Agent – Fixed Server Model
MCA-FSA Model	Mobile Clients with Agent- Fixed Server with Agent
P-P Model	Peer – Peer Model
MC-MA-FS Model	Mobile Client – Mobile Agent – Fixed Server Model
RPC	Remote Procedure Call
MAC	Message Authentication Code
IBM	International business machine
MASIF	Mobile Agent System Interoperability Facility
CORBA	Common Object Request Broker Architecture
OMG	Object Management Group's
MD5	Message Digest algorithm 5
RFC	Request for Comments
MIT	Massachusetts Institute of Technology
JDBC	Java Database Connectivity
JDK	Java Development Kit
API	Application Program Interface
JNDI	Java Naming and Directory Interface
RDBMS	Relational Database Management System
KDC	Key Distribution Center

Table 3

Reference

Journals and conference Paper:-

[1] Venkatraman, S. (2005). “Mobile Computing Models - Are they Meeting the Mobile Computing Challenges?”, *Association for Computing Machinery New Zealand Bulletin*, 1 (1) (ISSN 1176-9998), in 2005.

[2] Zhang Jun-yan; Li Yi-chao; Min Fan; Yang Guo-wei; “Mobile agent-based Security model for distributed System”, Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT’2003. *Proceedings of the fourth International conference* on 27-29 Aug. 2003 Page(s):194-197

[3] Alf Inge Wang Carl-Fredrik, Sorensen Eva Indal. ”Mobile Agent Architecture for Heterogeneous Devices”, Dept. of Computer and Information Science, Norwegian University of Science and Technology, N-7491 Trondheim, Norway in 2003.

[4] Wayne Jansen, Tom Karygiannis “NIST Special Publication 800-19 – Mobile Agent Security”, National Institute of Standards and Technology Computer Security Division Gaithersburg, MD 20899 in October 1999

[5] Samaras. G; Dikaiakos, M.D.; Spyrou.C; Liverdos.A; “Mobile agent platform for web databases: a qualitative and quantitative asswssment”, 3-6 Oct 2001 Page 50-64.

[6] Kastidou.G; Pitoura.E; Samaras.G, ”A scalable hash-based mobile agent location mechanism”, 19-22 May 2003 Page(s): 472-477.

[7] Agent Systems, Mobile Agents and Applications; Vol. 1882-(Lecture Notes in Computer Science)

[8] Mobile agents: 5th international conference*; proceedings/ MA 2001* Atlanta, GA, USA, December 2001

[9] Mobile agents for telecommunication applications: 3rd international workshop; proceedings/ MATA 2001, Montreal, Canada August, 2001

[10] Mobile agents :5th international conference; proceedings/ MA 2002 Barcelona, Spain, October 2002

[11] ‘Mobile Agents with Java: The Aglet API1” Danny B. LangeGeneral Magic Inc. 420 North Mary Avenue Sunnyvale, CA 94086 U.S.A.danny@acm.org, Mitsuru Oshima
IBM Tokyo Research Laboratory1623-14 Shimotsuruma, Yamato-shi Kanagawa-ken 242, Japan moshima@trl.ibm.co.jp, 2000.

[12] Danny B. Lange and Mitsuru Oshima, "Mobile Agents with Java: The Aglet API", World Wide Web Journal, 1998.

[13] MASIF The OMG Mobile Agent System Interoperability Facility, Dejan Milojicic, Markus Breugst, Ingo Busse, John Campbell, Stefan Covaci, Barry Friedman, Kazuya Kosaka, Danny Lange, ouichi Ono, Mitsuru Oshima, Cynthia Tham, Sankar Virdhagriswaran and Jim White

[14] Niklas Borselius, “Mobile agent Security”, this paper appears in “Electronics & communication Engineering Journal”, oct 2002 Volume 14 Page(s) 211-218.

[15] Dan Shiao, IBM Chaina, “Mobile Agent: New Model of Intelligent Distributed computing”, October, 2004.

[16] Lange, Danny B.; Oshima, Mitsuru, “Programming and Deploying Java Mobile Agents with Aglets”.

[17] Mobile Agent Facility Specification Edition 2000

[18] The Aglets 2.0.2 User's Manual October 2004

Websites

[19] http://aglets.sourceforge.net/

[20] http://www.trl.ibm.com/aglets/

[21] http://java.sun.com/docs/books/tutorial/jdbc/overview/index.html

[22] http://www.oracle-base.com/articles/9i/Articles9i.php

[23] http://www.smart-soft.co.uk/Oracle/oracle9i-new-features-part1.htm

[24] http://en.wikipedia.org/wiki/Password_cracking

[25] http://en.wikipedia.org/wiki/Main_Page

[26] http://www.merl.com/projects/concordia/WWW/documents.htm

[27] http://www.objectspace.com/products/voyager

Books

[28] William Stallings, "Cryptography and Network Security", Third edition, Pearson Education, 2003.

[29] Programming and Deploying Java Mobile Agents with Aglets by Danny B.Lange, Mitsuru Oshima.

[30] "The Complete Reference Java 2 Fourth Edition" by "Herbert Schildt",