ISSN No. 0976-5697

Volume 5, No. 7, September-October 2014

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Implementing Security in Mobile Agents by using AES Technique

Mitali Sachdeva[1], Amit Chhabra[1], Arvind Sharma[1]
Department of Computer Science
Swami Devi Dyal Group of Professional Institutions,
Barwala (Haryana), India

Abstract: Mobile Agent is a program which travels from one platform to another in order to get its work done. An agent is safe in its owner site but when it ventures out and gets executed on some other platform, it becomes vulnerable to many attacks. There are many counter measures that have been proposed in this paper to overcome these attacks. Cryptography based technique i.e. Symmetric Key Algorithm (specifically AES) is one such counter measure taken to provide security. An example case study of client server application has been considered to address the issue of security. Mobile agent is implemented on the aglet client agent host using a platform known as Aglet. Aglet is a java object which moves in a network and gets executed on host which are aglet enabled.
AES using Aglet and Java have been implemented to provide security to the mobile agent. The results show how the client agent and server agent can exchange information over internet in a secure manner. The private data entered by client agent is encrypted using AES key which is provided by the server agent to the client agent in an encrypted form. This means that only server agent can decrypt this because he has AES key.

Keywords: Mobile Agent [MA], Aglets, AES, Encryption, Decryption

I. INTRODUCTION

Mobile Agent (MA) is a program which travels from one platform to another in order to get its work done. During this process it carries its state and data with itself and resumes its execution from the state it had left on the previous platform. The Mobile Agent (MA) [1] paradigm seems to be a promising technology for developing applications in open, distributed and heterogeneous environments, such as the Internet. Many application areas [2], such as electronic commerce, mobile computing, network management and information retrieval can benefit from the application of the mobile agents technology. The exploitation of mobile agents offer several peculiar advantages, such as reduction of network latency, asynchronous execution, robust and fault tolerant behaviour. However, a wider diffusion of mobile agents is currently limited by the lack of a comprehensive security framework that can address the security concerns arising in mobile agent applications providing efficiency at the same time.

This paper is organized as follows:

Section II presents the literature survey in brief, Related Work for implementing the security policies is discussed in section III, section IV presents the proposed work of the research, section V provides the results and section VI concludes the paper.

II. LITERATURE SURVEY

The basic dictionary definition of agent is one who acts [1]. In order to achieve these goals they communicate and interact with other agents, they exchange information and take back the results to the user. The software agents behave in the same manner too. The agents may be stationary, may be resident or move from one host to another [1].

Agent technology is not a single technology; it's an integrated form of multiple technologies.

The current state of agent technology is as follows [1]:

a. The full set of technologies is not available.
b. Agent technology is still not a very widespread technology or has been widely accepted as it seems from its benefits.

Developers have found various forms of agents in Information Technology system. Few of these are discussed below:

a) Software Agents [3]
b) Intelligent Agents [4]
c) Autonomous Agents [5]
d) Adaptive Agents [6]
e) Mobile Agents [6]

A. Security Issues:

Alfalayleh and Brankovic [12] highlight the security issues for making the internet secure.

Computer networks connected to internet is always subject to security threats, so is a mobile agents system. Issues such as authentication, authorization, alterations, and encryption should be addressed in a mobile agent system.
The threats in mobile code systems can be broadly classified as [12]:

a. Threats that originate from an agent which can attack on another agent platform. This kind of attack can be protected in many ways e.g. sandboxing, software fault isolation, proof-carrying code, operating system access controls.
b. An agent attacking another agent on the agent platform, protection against this attack has been made as agent separation implemented on hosts.
c. An agent platform attacking an agent, protection of agents from malicious hosts remains a major problem in agent technology.

W. Jansen has discussed some threats which can occur due to malicious host [14]:

d. ***Eavesdropping:*** Eavesdropping threat involves the monitoring and interception of the secret information which is being communicated between authenticated

hosts.

e. ***Alteration:*** Alteration attack is extended attack of eavesdropping threat. If the agent is executing on malicious host it is exposing its code and data.

f. ***Masquerade:*** An agent platform can masquerade itself as another platform, to make itself appear to be an authenticated one, to a mobile agent. It may attract the agent to come and make it execute so that malicious host can extract sensitive information.

g. ***Denial of Service:*** This kind of attack occurs when an agent comes to a host to produce some results by utilizing the resources of the host.

B. Counter Measures:

The general solutions to the malicious host problem have been discussed:

a. ***Avoiding the Problem:*** Claessens et al. [15], describes one of the counter measures against malicious host i.e. avoiding the problem; this is the problem in which an agent is avoided to get executed on the untrusted remote host.

b. ***Hardware-Based Security:*** Hardware-based security is the most effective approach but it is not feasible enough. This approach requires installing a secure hardware on each node where a mobile agent can migrate to and execute [16].

c. ***Encryption-Based Security:*** Another approach is to use cryptography techniques. DES [17] and RSA [19] are the common encryption techniques. But in this paper AES technique is used which is also an encryption based seutiy technique

d. ***Enciphered Functions:*** T.Sander and C.F.Tschudin [18] have proposed a method whereby an agent platform can execute a program embodying an enciphered function without being able to discern the original function. In this, a function can be encrypted in such a way that they can still be implemented as programs

e. ***Co-operating Agents:*** By using cooperating agents the information and functionality can be divided among various agents. It gives benefit that if somebody tries to compromise one agent, may not be able to extract the complete information [18].

f. ***Execution Tracing:*** Execution tracing is used to detect unauthorized alteration of an agent by recording its execution detail at every platform. In this case each platform is supposed to maintain a log file of the execution detail of the agent on that system. Now the drawback is to maintain the various log files produced [13].

g. ***Obfuscated Code:*** Hohl has proposed code mess-up technique in which an agent's code is scramble in such a way that no one is able to gain a complete understanding of its function [13].

III. RELATED WORK

There are numerous works that suggest the implementation of security techniques in mobile agent based systems.

Bhawana Sharma, Arun Gangwar [2] has provided objectives, platform and architecture of mobile agents.

E.C. Vijjil [10] has provided complete research work on security issues of mobile agents including threats, countermeasures and future scope.

Rajesh Kumar, Yashpal Singh, S Niranjan [11] have highlighted on security issues of mobile agents.

M. Alfalayleh and L. Brankovic [12] highlight the security issues for making the internet secure. Without security and cryptography, e-commerce cannot work in the real world.

W. Jansen [14] has discussed some threats which can occur due to malicious host and also the countermeasures for protecting an agent.

Claessens et al. [15], describes one of the counter measures against malicious host i.e. avoiding the problem.

a. This is the problem in which an agent is avoided to get executed on the untrusted remote host. An agent is sent to only those hosts which have been authenticated.

b. The problem with this kind of approach is that it is not clear in advance which host to trust or not. This is very conservative approach and severely reduces the number of hosts on which an agent may migrate to.

The methodology behind the cryptography techniques AES is as follows:

A. AES:

The Advanced Encryption Standard (AES) [8] is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

The AES cipher is almost identical to the block cipher Rijndael [7] cipher developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, who submitted a proposal to NIST during the AES selection process. The Rijndael block and key size vary between 128, 192 and 256 bits. However, the AES standard only calls for a block size of 128 bits. Hence, only Rijndael with a block length of 128 bits is known as the AES algorithm.

The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data. The number of internal rounds of the cipher is a function of the key length. The number of rounds for 128-bit key is 10.

One Round of encryption and decryption is shown in figure 3.1

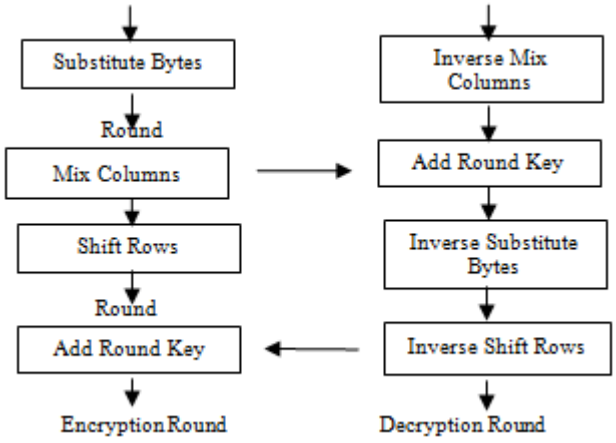

Figure 3.1: One Round of encryption and Decryption in AES

Unlike its predecessor DES, AES does not use a Feistel network. Feistel networks do not encrypt an entire block per iteration, e.g., in DES, 64/2 = 32 bits are encrypted in one round. AES, on the other hand, encrypts all 128 bits in one

iteration. This is one reason why it has a comparably small number of rounds.

Each round consists of several processing steps, each containing four similar but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key.

AES consists of so-called layers. Each layer manipulates all 128 bits of the data path. The data path is also referred to as the state of the algorithm. There are only three different types of layers. Each round, with the exception of the first, consists of all three layers.

We continue with a brief description of the layers:

a. Key Addition layer: A 128-bit round key, or subkey, which has been derived from the main key in the key schedule, is XORed to the state.
b. Byte Substitution layer (S-Box): Each element of the state is nonlinearly transformed using lookup tables with special mathematical properties. This introduces confusion to the data, i.e., it assures that changes in individual state bits propagate quickly across the data path.
c. Diffusion layer: It provides diffusion over all state bits. It consists of two sublayers, both of which perform linear operations:

The ShiftRows layer permutes the data on a byte level.

The MixColumn layer is a matrix operation which combines (mixes) blocks of four bytes.

a) High-level description of the algorithm:

(a). Key Expansion— round keys are derived from the cipher key using Rijndael's key schedule. AES requires a separate 128-bit round key block for each round plus one more.

(b). Initial Round

i. Add Round Key—each byte of the state is combined with a block of the round key using bitwise XOR.

(c). Rounds

i. Sub Bytes—a non-linear substitution step where each byte is replaced with another according to a lookup table.

ii. Shift Rows—a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

iii. Mix Columns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.

iv. Add Round Key

(d). Final Round (no Mix Columns):

i. Sub Bytes
ii. Shift Rows
iii. Add Round Key

AES Decryption: Because AES is not based on a Feistel network, all layers must actually be inverted, i.e., the Byte Substitution layer becomes the Inv Byte Substitution layer, the ShiftRows layer becomes the Inv ShiftRows layer, and the MixColumn layer becomes Inv MixColumn layer. However, as we will see, it turns out that the inverse layer operations are fairly similar to the layer operations used for encryption. In addition, the order of the subkeys is reversed, i.e., we need a reversed key schedule.

IV. PROPOSED WORK

Fig. 4.1 shows the flow diagram of the work done where the e-commerce mobile agent based application has been taken as a case study and the security techniques have been implemented.

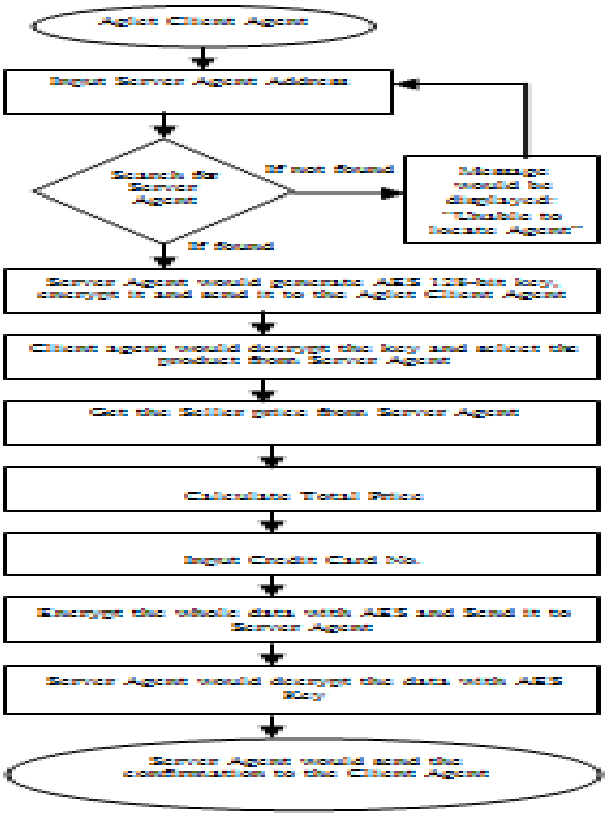

Figure 4.1: Flow diagram of work done

Symmetric key algorithm AES has been implemented on the agent code for the protection against malicious host.

AES needs only one key to encrypt the message. And both the client and server agent need the same key to decode the message.

In the mobile agent based application, the client agent would first input the server agent's address from where it can get its desired item. As soon as it is found, the server agent would generate the AES key, encrypt it and send it to the aglet client agent. Now the client agent would enter all its desired information and encrypt it using AES and send it to server agent. The server agent can decrypt the whole object code using AES key which is the desired information of the product given by the client agent and give the confirmation to the client agent.

It is very difficult to crack the code because the key used in this application is too big (128 bit long).

The following work is done:

a. Develop an Aglet based Client Agent.
b. Develop an Aglet based Server Agent.
c. Implement Encryption for security.

The proposed work is to achieve the security in mobile agents while they are travelling over internet and carrying confidential data.

V. RESULTS

The mobile agent system used in this thesis is Aglet Software Development Kit (ASDK) 2.0.2. The application is

shown with the help of an example of Aglet Client-agent GUI (Graphical User Interface) which inputs data from user and uses that information for doing its task.

Figure 5.1 shows the Aglet Client Agent. It has some fields as:

a) ***Server Location:*** This field shows the server location where the agent will migrate to.

b) ***SEARCH:*** This button will search the server location where the agent will migrate to.

c) ***AES HEX Key Received:*** This field provides the AES key provided by the Seller Agent in encrypted form.

d) ***Decrypted AES key:*** This field provides the original AES key.

e) ***Product Name:*** This field specifies the product to be bought by the agent.

f) ***Unit Price:*** This field specifies the unit price of the product which is to be bought by the agent.

g) ***Quantity:*** This field tells the number of products to be purchased.

h) ***Total Price:*** This field tells the total price of the number of quantities of the product to be purchased.

i) ***Credit Card Number:*** This field has the credit card number which the agent carries with it. This is the private information which the agent is carrying while travelling.

j) ***Encrypt:*** This button will encrypt the information entered by the Buyer Agent which is passed to the Seller Agent and the AES encrypted object data is placed in the field below it.

(a). ***Order Product:*** This button will confirm the order of the product placed by the user from the seller agent's side.

(b). ***Close:*** This button closes the agent.

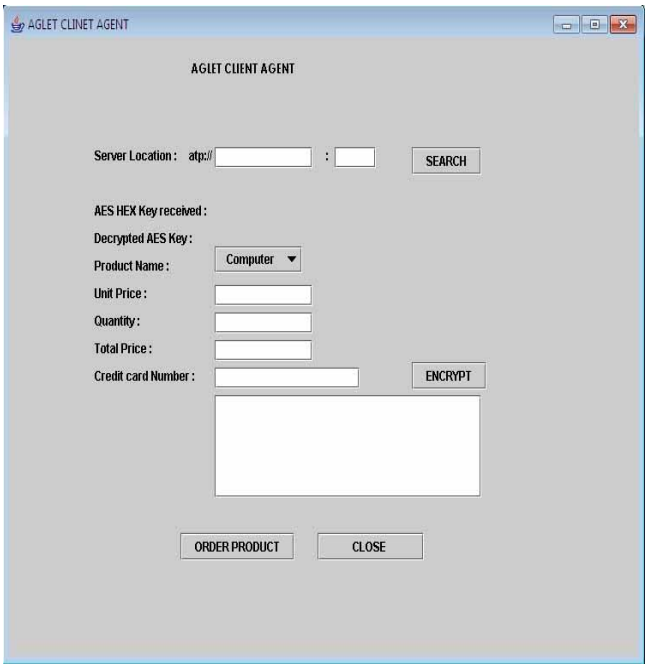

Figure 5.1: GUI of Aglet Client Agent

Server Agent is an agent who is present on the remote host and provides the information to the client agent wherever it is present and the cost of the product set by the sever-agent host is suppose to take by the client agent. In Figure 5.2 user enter the location address of the server and search whether the entered server is found or not.

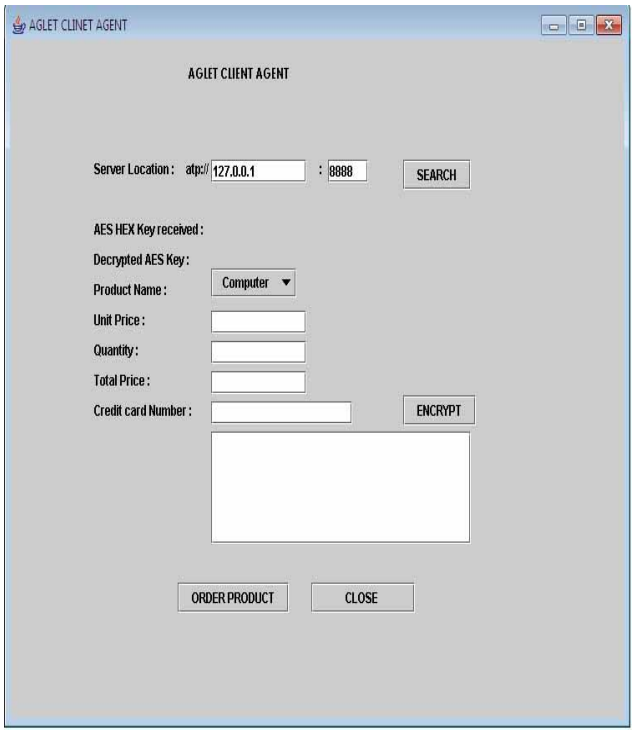

Figure 5.2: Server Location entered by the Aglet Client Agent

Figure 5.3 shows the list of information which the user entered to get the desired product, its quantity and the private information which is the credit card number to buy.

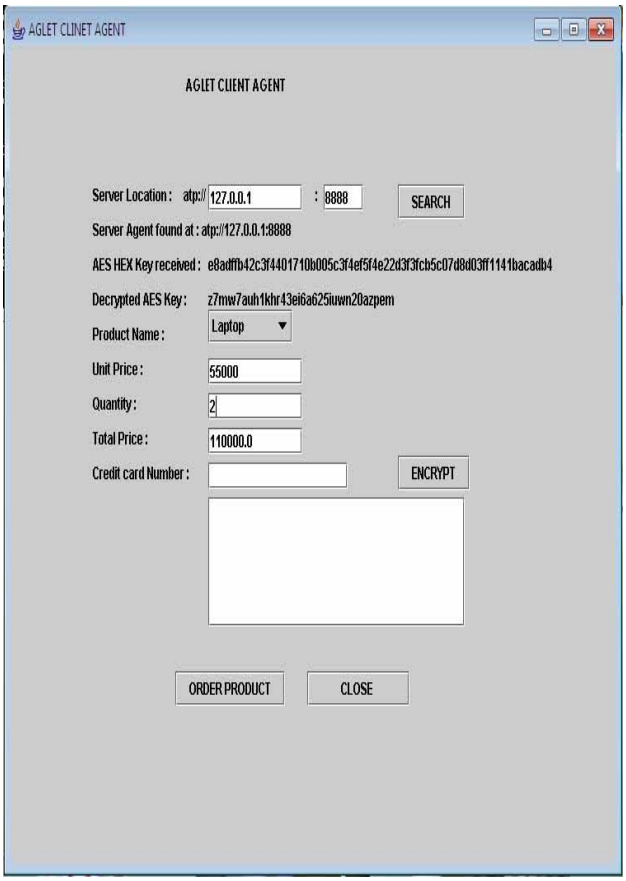

Figure 5.3: List of Information of Aglet Client Agent

Figure 5.4 shows the encrypted result of Aglet Client Agent done by AES (on all the information included credit card number) which the user entered to buy the desired product.

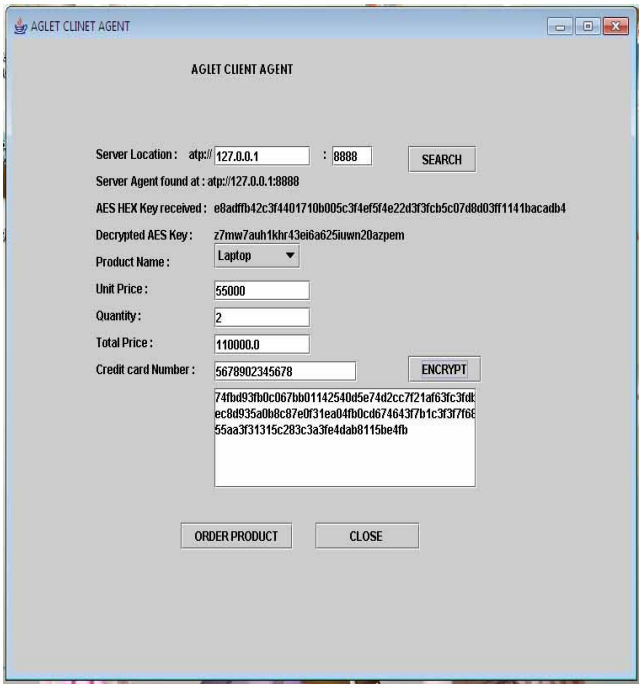

Figure 5.4: Encrypted Result of Aglet Client Agent

Purpose of the experiment: The case study studied in the above section is the encrypted information related to the Client Agent before sending it to the remote host. So by doing this, author is making the agent more secure as it is using the technique of cryptography i.e. Symmetric key (AES) for its protection.

The purpose of this experiment is to provide security to the private data carried by the Mobile Agent over internet. As there is a danger that any malicious host can get the AES key and easily encrypt the private data, so the AES key is provided by the Server Agent in an encrypted form.

A. Analysis of result:

The analysis of results presented in previous section is described below:

Aglet based Client Agent is developed on a host machine.

The GUI of an Aglet based Client is shown in figure 5.1.

As the client agent enters a server location i.e. address of the server agent with which it wants to exchange information, and presses the SEARCH button, the server agent with the particular address is searched (as shown in figure 5.2).

And the message is displayed "Server Agent found at: ....." and also an AES HEX Key is received at the Aglet Client Agent.

This AES HEX Key is actually the encrypted AES key provided by the Server Agent to the Client Agent.

The Aglet Client Agent then decrypts the AES key and generates the original AES key.

The Aglet Client then selects product out of the list of products provided by the Server Agent.

He then enters the Unit Price of the product selected and also the quantity of the product. Then total price is calculated (as shown in figure 5.3).

After calculating the total price of the product, the Aglet Client Agent then enters the private information which is to be secured i.e. his Credit Card Number.

After entering all the data, he presses the ENCRYPT button. As soon as the ENCRYPT button is pressed all the data is converted to an encrypted form using AES (as shown in figure 5.4).

And finally the Client presses the ORDER PRODUCT button.

Now, the mobile agent executing at the Aglet Client Agent takes all the AES encrypted data and moves to the server location provided by the Client Agent.

At the server location, when the mobile agent reaches it provides all the data to the server agent.

Now, the server agent knows the actual AES key. So he decrypts all the information using AES key and retrieves the original data.

If he finds the information provided by the client agent relevant then he confirms the order.

In this way selling and buying of items over the internet has been made safe.

Obviously, in this fast world everyone wants to buy and sell items on just one click of mouse. No-one has enough time to go to the market and buy sell items there.

So this application provides an efficient solution to all those people by providing security on the online transactions.

VI. CONCLUSION

Mobile agent paradigm has received a great deal of attention. Mobile agent technology can offer a new paradigm for communication over heterogonous network channels. Number of advantages [6] of using mobile agent computing paradigms include: overcoming network latency, reducing network load, executing asynchronously and autonomously, adapting dynamically, operating in heterogeneous environments, and having robust and fault-tolerant behavior. While mobile agents have generated considerable excitement in the research community, they have not translated into a significant number of real world applications due to various security issues. It has been found that without addressing any security issues, mobile agent technology cannot be exploited fully.

The architecture of mobile agent, the aglet model and the methodology have been discussed in this paper. A case study of client server application has been taken and the counter measures i.e. the cryptography technique (AES) has been selected for providing security. AES has been applied on the whole agent's code. By applying the cryptography technique, the application has been made more secure because in order to crack the code, AES key is needed which is 128-bit long.

No software system can provide a complete solution in the protection of the mobile agents Ref. [1, 2]. An executing host must have deep knowledge about the agent's code and the data it uses. A host has possibility to interpret a visitor mobile agent, to extract the data it uses. Mobile agent systems perform quite well on secure networks, but they need more autonomy and intelligence to react in more risky and changing environments. Future distributed applications are expected to exploit the flexibility of agent-based software in a variety of ways.

VII. ACKNOWLEDGMENT

The authors wish to thank all the faculty members of our department and our family for making this research successful.

VIII.REFERENCES

[1] Needam, MA, Agent Technology, Object Management Group, Green Paper 1, pp. 11, Sept. 2000.

[2] Bhawana Sharma, Arun Gangwar, "MOBILE AGENTS: Objectives, Platforms and Architecture," International Journal of Engineering and Management Research, Issue 2, April 2012.

[3] H.S. Nwana, Software Agents: An Overview, Intelligent Systems Research, Advanced Applications & Technology Department, Ipswich, Suffolk Cambridge University Press, 1996.

[4] D.Lange and M.Oshima, Seven Good Reasons for Mobile Agents, Communications of the ACM, Volume 42, pp. 88-89, 1999.

[5] The Foundation for Intelligent Physical Agents (FIPA), http://www.fipa.org , 2006.

[6] D. B.Lange and M. Oshima, Programming and Developing Java Mobile Agents with Aglets, Addison Wesley, pp. 1-7, 1998.

[7] Joan Daemen, Vincent Rijmen, "The Design of Rijndael: AES – The Advanced Encryption Standard" Springer, 2002, ISBN 3-540-42580-2

[8] Christof Paar, Jan Pelzl , "The Advanced Encryption Standard", Chapter 4 of *Understanding Cryptography*, A Textbook for Students and Practitioners", Springer, 2009.

[9] K.P.Tripathi, "A Multi-Paradigm approach for Mobile Agents," Proceedings of the 5th National Conference;INDIACom-2011

[10] E.C. Vijil, "Security Issues in Mobile Agents," Indian Institute of Bombay, Mumbai.

[11] Rajesh Kumar, Yashpal Singh, S Niranjan, "Review of Security Issues for Mobile Agent Technology," International Journal of Engineering Research and Development, Issue 4, May 2013

[12] M. Alfalayleh and L. Brankovic, An Overview of Security Issues and Techniques in Mobile Agents, The School of Electrical Engineering and Computer Science, University of Newcastles, Newcastle, NSW 2308, Australia

[13] F.Hohl, "Time Limited Blackbox Security," G. Vigna, editor, Mobile Agents and Security, Springer-Verlag, Volume1419 of LNCS, pp.97-102, June 1998.

[14] W. Jansen, Countermeasures for Mobile Agent Security, Computer Communications, Special Issue on Advances in Research and Application of Network Security, 2000.

[15] J. Claessens, B. Preneel, J. Vandewalle, How Can Mobile Agents Do Secure Electronic Transactions on Untrusted Hosts? A Survey of the Security Issues the Current Solutions, ACM Transaction on Internet technology, Vol. 3, no.1, pp.38-41, Feb. 2003.

[16] W. M. Farmer, J. D. Guttman, and V. Swarup, Security for Mobile Agents: Authentication and State Appraisal, European Symposium on Research in Computer Security (ESORICS), 2002.

[17] Davis, R., The Data Encryption Standard in Perspective, Proceeding of Communication Society magazine, IEEE, Volume 16 No 6, pp. 5-6, Nov. 1978.

[18] T.Sander and C.F.Tschudin, Protecting Mobile Agents against Malicious Hosts, G.Vigna, edior, Mobile Agents and Security, Volume 1419 of LNCS, pp. 44-60. Springer-Verlag, June 1998.

[19] R.L.Rivest, A.Shamir, and L.Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, Volume 21 No. 2, Feb. 1978.