International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 22 (2017) pp. 11824-11829
© Research India Publications. http://www.ripublication.com

Mobile Agent Data Security using Triple Data Encryption Standard

[1]Prapulla S B, [2]Trisha Venkatesh, Jayanth Chandra[3], Sindhu B Dinesh[4], Vindhya Nagaraj[5]

[1]Assistant Professor, Department of Computer Science and Engineering, Rashtreeya Vidyalaya College of Engineering, R V Vidyanikethan Post, Mysuru Road, Bangalore, Karnataka, India.
[2,3,4,5] Student, Department of Computer Science and Engineering, Rashtreeya Vidyalaya College of Engineering, R V Vidyanikethan Post, Mysuru Road,Bangalore, Karnataka, India.

[1,2,3]Orcid Id: 0000-0003-1897-5079, 0000-0001-9679-9096, 0000-0003-0366-5019,

[4,5]Orcid Id: 0000-0002-0217-4982, 0000-0001-9980-0312

Abstract

A mobile agent is an autonomous programming element consisting of code and data (state), which is programmed to represent a network user. Their applications are vast, owing to the ability to use them in economical communication channels with lower bandwidth and higher latency. Agents might attempt to acquire information from other agents or nodes in the cluster to accomplish their objectives in agent-based frameworks. Based on the processing performed at each node, the agent saves its state decides its path autonomously and resumes operation at the next node. Such operations in open situations may lead to many threats to agents like attack by malicious hosts, other malicious agents. This paper proposes a model to satisfy the security concern. A secure trusted network is established before allowing for the communication among the agents. This provides an access control mechanism against malicious hosts. Each agent is facilitated with its own set of public and private keys using DSA and Triple-DES secures the messages by encrypting them, thus protecting them from malicious agents. This paper gives insight into the proposed model by implementing the agent framework using JADE. The experiments carried out show that the proposed method used for encryption is very secure since Triple DES encryption is done and the possibility of breach is very less.

INTRODUCTION

A mobile agent is a self-ruling and objective situated programming element which works together and conveys with other programming elements and people. There are different definitions of a mobile agent, yet the most widely recognized attributes of agents are: self-governance, social capacity, reactivity, and pro-movement. Some authors' attributes, the operator worldview has turned into a promising innovation to build applications in open, taken, mixed situations. Truly, mobile agent specific frameworks are normally created in free appropriated situations, particularly in e-commerce, mobile computing, and information recovery regions. Mobile agents show advancement in assuming that licenses for total flexibility to organize applications in assistive stages in shaping a tremendous range, openly coupled scattered structure. Regardless of the way those there different models that can be utilized to delineate mobile agent frameworks, a direct version containing a pair of sections: an administrator and an agent stage seems adequate to talk about security of agents. We understand that a mobile agent contains the code and state data necessary to perform calculations. The Mobile agent stage gives the computational condition in which a mobile agent works. [1] The stage from which a mobile agent starts or is made is said to be the home stage. Multi agent itinerary plays a significant role in data aggregation.

In agent specific structures, agents might endeavour to get data from various agents, nodes or acquire data from remote expert association agents to fulfil their tasks. Shockingly, in open circumstances in which agents can uninhibitedly move around, such activities would be dangerous and sketchy in light of the way to decide reliability of agents and which external gets to don't prove a threat. Lacking suitable and sufficient responses for these security issues, there are chances that some amount of fragile data might be spilled. In any case, agent's traits, for instance, freedom make it strenuous to ensure the security of the structures. To cover agent specific structures from security threats, we require guaranteeing basic security attributes such as confidentiality, honesty and non-repudiation through different instruments, for instance, affirmation, endorsement, et cetera. Various examiners have separated security vulnerabilities and recognized security necessities and difficulties. Also, potential security strikes have been inspected, and relevant methods for security against attacks have been suggested. From an arrangement of security issues, the security in these agent systems is said to be the principle convergence of various researchers. A mobile agent is a said to be an agent which can move beginning with one host then onto the following where it can proceed with its enforcement. The agent's adaptability is not an obligatory trademark regardless, it has starting late pulled in thought owing of its central focuses. Mobile agents can decrease composing traffic and beat organize latencies which are its greatest benefits. In addition, in case we can handle the security issues pertaining to mobile agents, at that point the above-mentioned game plans can be easily associated with deal with the security issues of any kind of agent-based structure. Many existing security courses of action have been researched for their pertinence to agent specific structures. Other than security thoughts for agent specific structures, a couple of experts have moreover proposed utilizing agents to give security organizations.

A mobile agent however can move (ricochet) starting with one execution condition then onto the next in a framework. This recent condition, known as the host condition, expects full command on an agent's programs, information and evaluation condition [2][3].

This kind of host's authority on the running code creates a difficult scenario to shield compact administrators from noxious hosts[4] what's all the more, in light of current circumstances opens them to various security perils. On a very basic level, the security requirements of any PC structure are secrecy, honesty, verification, approval, non-disavowal and accessibility [5][6][7]. The security necessities of an agent can be handled by a malignant host in different fashions. This consolidate contradiction of advantage, listening stealthily,

piece endeavor, alteration, replays and masking [5][8][9][10][11]. While strategies, for instance, get the opportunity to control, mystery word protection besides, sand boxes have been made to guarantee administrator stages against debilitating administrators [12], not one of the philosophies in securing agents from malevolent nodes has sufficiently tended to each piece of security [13].

Mobile Agent Security: Dangers Utilizing a clear model containing agent and agent stage, safety risks in the agent structures are masterminded into the following arrangements specifically agent-to-stage, agent-to-agent, stage to-agent and other-to agent stage.

Classification of Threats to Mobile Agents:

a. Agent to platform: A class which speaks to an arrangement of dangers where mobile agents abuse security shortcomings in the agent stage or dispatch assaults against the latter. This arrangement of dangers incorporates disguising, disavowal of administration and unapproved get to.

b. Agent-to-Agent: This classification speaks to the arrangement of dangers where mobile agents abuse security shortcomings in different agents or dispatch assaults against the latter. This arrangement of dangers incorporates disguising, unapproved get to, foreswearing of administration and disavowal.

c. Stage to Agent: This classification speaks to the arrangement of dangers wherein stages bargain the security of mobile agents. This arrangement of dangers incorporates disguising, disavowal of administration and listening in.

d. Other to Agent Platform: This class speaks to the arrangement of dangers in which outer elements, along with agents and their stages undermine the security of the mobile agent stage. This arrangement of dangers incorporates disguising, refusal of administration, unapproved get to, and duplicate and replay.

e. Platform to Agent Security Threats: The mobile agent has the highest security in its home stage because the place it is created. However, versatility suggests this trusted execution condition should be reached out to other host stages in the agents' schedule. Such trust is hard to stretch out past a solitary jump particularly in light of the fact that while the home stage could put stock in the following host of the system, this two-sided trust can't be transitive i.e. the home stage (x) puts stock in the following host (y), it cannot imply that a different host of the network(z) likewise confides in (y). This multifaceted nature presents a multihop security issue. A portion of the conceivable stage to agent security dangers incorporate this accompanying [5][8][3]:

- **Denial of Service:**

The agent stage ought to dependably execute an agent's requests, apportion fundamental assets and keep the endless supply of administrations. Be that as it may, a malignant agent stage, may disregard agent benefit demands, present inadmissible deferrals for basic undertakings, decline to run the agent's programs, in some cases may terminate an agent unnoticeably. Non-responsive agents on malevolent stages would be stopped or live locked.

- **Masquerade:**

A scenario wherein a pernicious stage asserts the character of another stage that a mobile agent ought to really visit. Thus, making the agent give up vindictive delicate data. When the disguising host can pick up the agent's trust, it might have the capacity to peruse or alter parts of an agent's programs, information and state. However, it can be counteracted by utilization of a solid validation convention to send confirmation between a host and an agent.

- **Eavesdropping**

The agent's way of performing on the host implies that the host can document an agent's directions those are given to it. This suggests a vindictive host might attempt to decide the code, information or stream authority of agent. Despite the fact that the agent may not be straightforwardly uncovering mystery data, the stage might have the capacity to gather significance from the sorts of administrations asked for and from the personality of the agents with which it conveys. This type of assault is hard to forestall and distinguish.

- **Alteration**

A pernicious host can adjust the mobile agent by modifying the information, code and control stream with the goal that the agent does different assignments than what was expected by its maker. An agent that hits a few stages on its schedule is presented to another hazard on its travel and on each of such occurrences; an instance is created on another stage. Change may be distinguished by making the first creator carefully sign the agent's code. However, this location ends up noticeably troublesome for agents going by a few stages (the "multi-bounce" issue).

LITERATURE SURVEY

Multi mobile agent itinerary and security plays an important role in data collection. In [16] authors have proposed an algorithm called farthest node first and nearest node technique for itinerary of agents.

As indicated by Lange and Oshima in [14], there are three crucial safety problems concerning mobile agent frameworks, namely:

- Shielding the host (organize) from the mobile agent.
- Defending the mobile agent from other mobile agents, and
- Guarding the mobile agent from the host.

A portion of the proposition to shield an agent against a noxious host is talked about underneath:

a. Shadow and Essential Operator Approach: This method suggested by [11] expects to perceive and stay away from each blocking noxious host in the agenda of a mobile administrator. The plan uses an insistence and time - out framework to guarantee that a mobile administrator has passed by the host in its schedule and can be pulled back to the accompanying one, without affecting itself. A pair of agents are involved; a basic (PA) and a shadow (SA). Commonly, SA slacks one phase in the agenda behind PA. The supposition is that a host is said to be non - blocking should it empower the PA to proceed with its work and leave to the accompanying host unharmed. The SA estimates a pernicious operation in case it doesn't get an insistence inside a suitable time - out T after which it asks for aid from the home host to perceive the toxic host and make remedial move. As soon as the pernicious host is identified by the home host, it dispatches another occurrence of the PA to a secured host to meet SA. The latter passes on a duplicate of the accumulated information. The SA will reload the assembled data into the void PA. The as of late stacked PA will proceed with its agenda skirting the malignant host.

b. Authors of [7] Proposed Partial Mobility Mechanism (PMM) to secure the agents respectability and protection against noxious hosts. In PMM, the mobile administrator has a couple of sorts:

1. A One Hop Agent (OHA) - This addresses errands to be computation in case of untrusted have and can in a manner of speaking visit one has.

2. The Multi - Jump - Operator (MHA) - This addresses errands to be done in trusted in has. The MHA can strike different place stock in has. In PMM, the Mobile agent's home stage makes a mobile agent and decides every one of the hosts in the itinerary of the mobile agent. Hosts are appointed either trusted (serves MHAs just) or untrusted (serves OHAs figuratively speaking). One foundation of this part is a mobile agent's agenda dealing with the self-administration property of a mobile agent is gathered before. It is furthermore burdensome to screen the hosts' security status in an appropriated sort out.

Agents follow up on information and deliver information that correspond to after effects of their computation. This multi-faceted view makes execution in agents quicker by confining security controls. It additionally builds the framework. However, the implementation of this approach on Java based mobile agents has still not been done and studied.

c. Creators of [17] proposed a multi - faceted approach to manage security. This strategy gives Protection against Masquerading, Eavesdropping and Alteration and Blocking and Denial of Service. This multi - faceted method should begin with necessities designing stage the best method to deal with upkeep of the made structure. This method is responsible for the agent enforcement speedier by confining security controls. It moreover builds a powerful structure in light of the fact that exclusion.

d. The Drexel University has proposed the following architecture. SWAT (Secure Wireless Agent Testbed) is another technique used on agents and Ad Hoc networks for implementing secure communications. SWAT aims in integrating public and symmetric key encryption with agents to create multiple agents groups and establish inter-agent communication. SWAT addresses the previously occurring problems in ad hoc networks by integrating host, security and agent systems by facilitating a relationship among each of the entities. The major drawback is the limitational power of the iPAQ nodes, due to which the currently proposed SWAT architecture has to be optimized in order to act as an efficient solution for multi agent security.

PROBLEM DEFINITION

In the survey carried out, it has been found that the existing systems suffer from the following problems:

a. One of the methods surveyed requires two agents, an SA and a PA. This creates unnecessary agents.

b. One of the existing works has the problem of monitoring the security standing of hosts in an appropriated organize. The usage of the One Hop Agent (OHA) likewise restricts transportability of the operator and in this manner mistreats the primary idea of a mobile agent.

PROPOSED SYSTEM

The system being proposed focuses on making the security process more efficient. The system has to establish a secure network before data is sent. This is achieved by a user authentication method where the base station creates a user id and SSH-key where both of this is used for validating the authenticity of the nodes. The data being carried is encrypted with the key generated by the DSA algorithm. These keys are very quick and easy to generate. It provides for message authentication as it allows the receiver to verify that the key used belongs only to the sender. The proposed method does not need any extra agents to accomplish the task at hand. This work can be used in applications like sensitive data collection, monitoring where the information being carried has to be secure.

IMPLEMENTATION

The entire implementation is explained in the following steps:

1. Establishing the communication: The first step in establishing a communication is to create a secured and trusted network where the nodes and base station are authenticated so that that the communication would be secured. This is achieved by a user authentication method where the base station creates a user id and SSH-key where both of this is used for validating the authenticity of the nodes. This is used by agents for registering with the base station. This forms a trusted network of nodes with the base station.

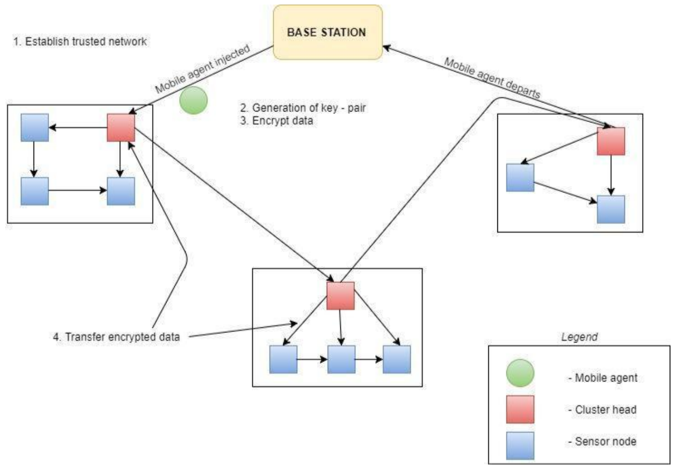


Figure 1: System layout

2. Key generation: Public and Private keys are generated by DSA algorithm, as mentioned earlier. This key-pair once generated are stored in order to be used later during data encryption. The algorithm used for the generation of public and private keys is Digital Signature Algorithm (DSA).

Each agent has a pair of public and private keys.

The keys generated by this are used to encrypt and decrypt the data being sent between two agents.

The process involved in key generation is as follows:

- Select a prime number q (called the prime divisor). It is N bits.

- Decide on another prime number p, so that p-1 mod q = 0. p is termed as the ‘prime modulus’. It is L bits in length.

- Select an integer g, subject to constraints 1 < g < p, g^q mod p = 1 and g = h^((p–1)/q) mod p. In other words, g is a number such g's multiplicative order modulo p is equal to

q. The set (p,q,g) can be distributed among different users.

The following steps are for the key generation for a user given the above parameters.

- Select an integer x, such that 0 < x < q. x is the secret or private key that can be chosen by any method.

- Compute y so that g^x mod p. y is the public key. □ Bind {p,q,g,y} as the public key.

- Bind {p,q,g,x} as the private key.

It has been implemented in Java and it uses the **java.security** package. The keys are of length 512 bits.

Of all the simple cryptography methods, DSA proves to be very handy and convenient. It provides for message authentication as it allows the receiver to verify that the key used belongs only to the sender. This is achieved by a hash function. Suppose, the data is modified by an intruder, it will fail to be verified by the hash function.

3. Encryption and decryption: The data that is transferred has to be encrypted in order to remain safe from tampering.

This paper makes use of Triple Data Encryption Standard Algorithm (Triple DES) for encryption, which is a symmetric-key encryption algorithm which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block. DES uses a key size of 56 bits which is very easily prone to brute-force attacks. Hence, Triple DES provides a more effective algorithm for such attacks as the key size is longer due to three levels of encryption.

Triple DES uses a key bundle that comprises three DES keys, K_1, K_2 and K_3, each of 56 bits (excluding parity bits which take up 8 bits). The encryption algorithm is:

$C = E_{K3} (D_{K2} (E_{K1} (P)))$

Where, C-cipher text, P-plain text and E and D denote encryption and decryption respectively.

A message is encrypted with K1 first, then decrypted with K2 and encrypted again with K3.This increases security as the key length effectively increases from 56 to 112 or 168. Managing several keys isn't an issue as they are all encoded into a single key. This is the same key-pair which is generated in step 1. Using this key-pair, the data encryption is carried out.

DES encryption has been implemented in Java using javax.crypto.spec.DESedeKeySpec package. The encrypting and decrypting are done using sun.misc.BASE64Decoder and sun.misc.BASE64Encoder packages respectively. Once a secure network is set up and the data has been encrypted, the first entity (sender) sends the data from its end. Once the second entity (receiver) receives the data, it has to be decrypted for further usage. The mobile agent communication is being implemented using JADE.

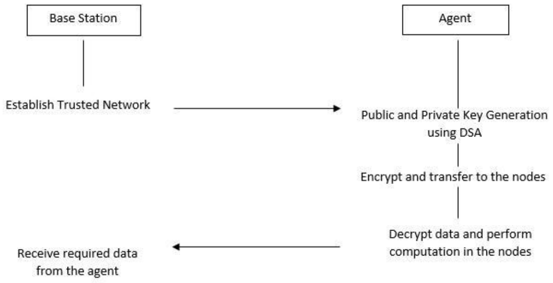
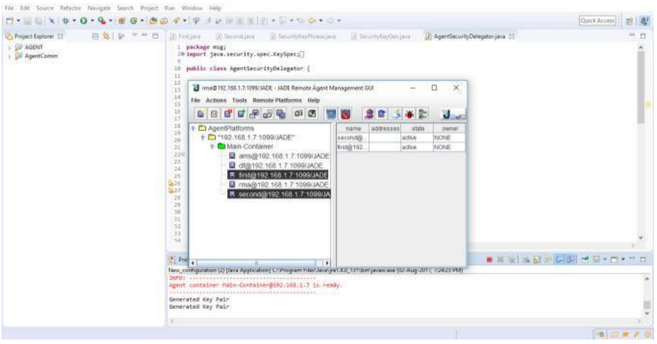

Base Station
Agent
Establish Trusted Network
Public and Private Key Generation using DSA
Encrypt and transfer to the nodes
Decrypt data and perform computation in the nodes
Receive required data from the agent

Figure 2: Sequence of events

RESULT

Once the agent framework has been implemented using the proposed approach in JADE, a main container is created with the two entities, i.e., the base station and the mobile agent.


File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Project Explorer
AGENT
AgentComm
First.java
Second.java
SecurityKeyPhrase.java
SecurityKeyGen.java
AgentSecurityDelegator.java
rma@192.168.1.7:1099/JADE - JADE Remote Agent Management GUI
File Actions Tools Remote Platforms Help
AgentPlatforms
"192.168.1.7:1099/JADE"
Main-Container
ams@192.168.1.7:1099/JADE
df@192.168.1.7:1099/JADE
first@192.168.1.7:1099/JADE
rma@192.168.1.7:1099/JADE
second@192.168.1.7:1099/JA

```
package msg;
import java.security.spec.KeySpec;

public class AgentSecurityDelegator {
```

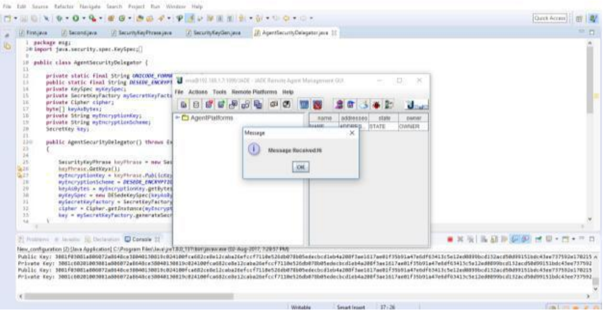
name	addresses	state	owner
second@...		active	NONE
first@192...		active	NONE

```
New_configuration (2) [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (02-Aug-2017, 7:24:23 PM)
INFO: ----------------------------------------
Agent container Main-Container@192.168.1.7 is ready.
----------------------------------------

Generated Key Pair
Generated Key Pair
```

Figure 3: 'first' and 'second' agents created in JADE

These agents are then made to communicate with each other by using their private and public keys using SecurityKeyGen class. The 512 bit key generated is used for message exchange. Using these keys for encrypted communication, they transfer messages over the network. Successful delivery of the message is shown by a pop-up dialog with the intended message to be transferred.


File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
First.java
Second.java
SecurityKeyPhrase.java
SecurityKeyGen.java
AgentSecurityDelegator.java
JADE Remote Agent Management GUI
File Actions Tools Remote Platforms Help
AgentPlatforms
Message
OK
Problems
Javadoc
Declaration
Console
Writable
Smart Insert
37 : 26

```
package msg;
import java.security.spec.KeySpec;

public class AgentSecurityDelegator {

    private static final String UNICODE_FORM
    public static final String DESEDE_ENCRYPT
    private KeySpec myKeySpec;
    private SecretKeyFactory mySecretKeyFact
    private Cipher cipher;
    byte[] keyAsBytes;
    private String myEncryptionKey;
    private String myEncryptionScheme;
    SecretKey key;

    public AgentSecurityDelegator() throws Ex
    {

        SecurityKeyPhrase keyPhrase = new Se
        keyPhrase.GetKeys();
        myEncryptionKey = keyPhrase.PublicKey
        myEncryptionScheme = DESEDE_ENCRYPTI
        keyAsBytes = myEncryptionKey.getBytes
        myKeySpec = new DESedeKeySpec(keyAsBy
        mySecretKeyFactory = SecretKeyFactory
        cipher = Cipher.getInstance(myEncrypt
        key = mySecretKeyFactory.generateSecr
```

name	addresses	state	owner
NAME	ADDRESS	STATE	OWNER

Message Received:hi

```
New_configuration (2) [Java Application] C:\Program Files\Java\jre1.8.0_131\bin\javaw.exe (02-Aug-2017, 7:28:57 PM)
Public Key: 3081f03081a806072a8648ce38040130819c024100fca682ce8e12caba26efccf7110e526db078b05edecbcd1eb4a208f3ae1617ae01f35b91a47e6df63413c5e12ed0899bcd132acd50d99151bdc43ee737592e170215
Private Key: 3081c60201003081a806072a8648ce38040130819c024100fca682ce8e12caba26efccf7110e526db078b05edecbcd1eb4a208f3ae1617ae01f35b91a47e6df63413c5e12ed0899bcd132acd50d99151bdc43ee737592
Public Key: 3081f03081a806072a8648ce38040130819c024100fca682ce8e12caba26efccf7110e526db078b05edecbcd1eb4a208f3ae1617ae01f35b91a47e6df63413c5e12ed0899bcd132acd50d99151bdc43ee737592e170215
Private Key: 3081c60201003081a806072a8648ce38040130819c024100fca682ce8e12caba26efccf7110e526db078b05edecbcd1eb4a208f3ae1617ae01f35b91a47e6df63413c5e12ed0899bcd132acd50d99151bdc43ee737592
```

Figure 4: Screenshot showing successful transfer of data

CONCLUSION

Thus, the proposed work used for encryption is highly secure because of Triple DES encryption and the probability of breach is very less. This method where Triple DES is combined with DSA has not been tried before. The experiments have shown that it is simple and efficient. The key generation process is simple and fast, thus improving the overall efficiency. This system currently uses generation of access key method for authenticating each node and thereby ensuring a trusted and secure network among the nodes before starting the communication process. Thus, additional processing is not required from third party entities.

FUTURE WORK

In future, improvements can be made to the DSA algorithm by including elliptic curve cryptography (ECDSA- elliptic curve DSA). This can further reduce the size of the keys and the time required.

ACKNOWLEDGEMENT

The authors would like to thank Dr. Shobha G for her guidance, support and valuable insights which assisted this work.

REFERENCES

[1] Jansen W. A., (1998)Mobile Agents And Security. National Institute of Standards and Technology, USA.

[2] Shzrivastava, S. & Nandi, G.C.,(2014)Fragmentation based encryption approach for self protected mobile

agent. Journal of King Saud University -Computer and Information Sciences, 26, pp.131-142.

[3] Razouki, H. & Hair, A.,(2014). Towards A New Security Architecture of Mobile Agents. International Journal of Soft Computing and Engineering (IJSCE), Vol 3 Issue 6, 55-60.

[4] Dey,S & Sinha, D., (2014)A Survey on Protection Techniques of Mobile Agents from Malicious Hosts. International Journal of Innovative Research in Computer and Communication Engineering, Vol 2, Issue 8.

[5] PAhich, P., Dutta, K., Govil, M. C. (2010) 'Security Issues in Mobile Agents', International Journal of Computer Applications,11(4), pp. 1-7.

[6] Ahmed, T.M. (2013) 'Protect Mobile Agent Against MaliciousHost Using Partial-Mobility Mechanism',International Journal in Foundations of Computer Science & Technology (IJFCST),3(6), pp. 41-52.

[7] Singh, D., Thakur, A., Gupta, D. (2015) 'A Review of Mobile Agent Security', International Journal of Advanced Research in Computer Science and Software Engineering,5(2), pp. 188-190.

[8] Pai, P., Shinde, S.K., Khachane, A.R. (2012) 'Security in Mobile Agent Communication', International Journal of AdvancedEngineering Research and Studies,1(4), pp. 74-80.

[9] [9] Mahmoodi, M., Varnamkhasti, M.M. (2013) 'A Secure Communication in Mobile Agent System', International Journal of Engineering Trends and Technology (IJETT),6(2), pp. 186-188.

[10] Ahmed, T.M. (2013) 'Protect Mobile Agent Against Malicious Host Using Partial-Mobility Mechanism',International Journal in Foundations of Computer Science & Technology (IJFCST),3(6), pp. 41-52.

[11] Ahmed, T.M. (2009) 'Using Secure-Image Mechanism to Protect Mobile Agent against malicious Hosts ', International Scholarly and Scientific Research & Innovation,3(11), pp. 364-369.

[12] Lee, H., Alves-Foss,J., and Harrison, S. (2004)'The Use Of Encrypted Functions For Mobile Agent Security'. Hawaii International Conference On System Sciences. Hawaii: DARPA.1-10.

[13] Shrivastava, R., Mehta, P. (2012) 'Securing Mobile Agent And Reducing Overhead Using Dummy And Monitoring Mobile Agents', International Journal of Management, IT and Engineering,2(4), pp. 296-303.

[14] Ebietomere, E.P. & Ekuobase, G.O., (2014)Issues on Mobile Agent Technology Adoption. African Journal of Computing & ICT, 7.

[15] D'Anna, L., Matt, B., Reisse, A., Vleck, T.V., Schwab, S. and LeBlanc, P (2003) Self-Protecting Mobile Agents Obfuscation Report, DARPA: Network Associates Laboratories.

[16] Prapulla S B, Jayanth Chandra, Madhwesh B, Mudakavi, Shobha G, Thanuja T C, "Multi Mobile Agent Itinerary planning using Farthest Node First Nearest Node Next (FNFNNN) Technique *International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS),* Bengaluru, October 2016 PP: 105 - 111

[17] Anthony M. Ngereki, Andrew M. Kahonge, " A Multi – Faceted Approach to Mobile Agent Security", International Journal of Computer Applications, Volume 120 , No.21, June 2015, pp 20-26.