Trust Enhanced Data Security in Free Roaming Mobile agents Using Symmetric Key cryptography

G.Geetha [1] C.Jayakumar[2]

[1]Assistant Professor , Jerusalem college of Engineering, Chennai,India ,
gee_dgl@yahoo.com
[2] Professor, RMK Engineering college ,Chennai , India,
cjayakumar2007@gmail.com

Abstract. *Mobile agents are emerging as useful paradigm in electronic commerce, for both wired and wireless environments Mobile agents are software programs that live in computer networks, performing their computations and moving from host to host as necessary to fulfill user goals.. In this research work, both chain relation and TTP (trusted host) has used for protecting data of free roaming mobile agents. We use TTP information with a host which is called Knowledge Based System (KBS) to maintain Trusted host Information. Agent delight at trusted Host as agent originator by means of symmetric key encryption. By using chain relation with trusted host, redundancy will be reduced and efficiency will be improved.*

Key words: Mobile agent, Trust, Data security, Symmetric key, Knowledge Based System, Cryptography

1 Introduction

Mobile agents migrate from originating hosts to intermediate servers to generate and collect data, and return to the originators to submit results after completing scheduled tasks. Mobile agents are computer programs, which are autonomous, proactive and reactive, and have ability to learn.

Mobile Agents are classified as free roaming Mobile agents, predefined itinerary mobile agents. Predefined itinerary Mobile agents know path in which they want to migrate.

Free roaming mobile agents are free to choose their respective next hops dynamically based on the data they acquired from their past journeys. Free-roaming agents have no pre-defined migration paths. They select their next hop at each hop they visit based on initial requirements and current conditions. Security of mobile agent systems are classified as

- **Security**
  1. Plat form security
  2. **Mobile Agent**
     1. single Hop
     2. **Multi Hop**
        1. Code security
        2. **Data security**
           1. Predefined itinerary
           2. **Free Roaming Mobile agent**

DOI : 10.5121/ijnsa.2011.3517

There are many security issues to be addressed in Data security in Free Roaming Mobile Agents for example data confidentiality, non reputability, insertion defense and truncation defense etc.

Generally security issues in mobile agents as

1. Protection of the host from malicious code

2. Protection of the agent from a malicious host trying to tamper the code and the agent data.

Agent Security is divided into code security (tampering attack, etc) and data security. Methods used to protect data in mobile agents count on move forms. The move forms of agent are pre defined itinerary and free roaming.

Security in free roaming agents is especially hard to achieve when the mobile code is executed in hosts that may behave maliciously. Data security in free roaming agent without itinerary information may face more complex attacks. This paper focus on Data Security in free roaming mobile agents.

1.1 Mobile Agent Work Flow

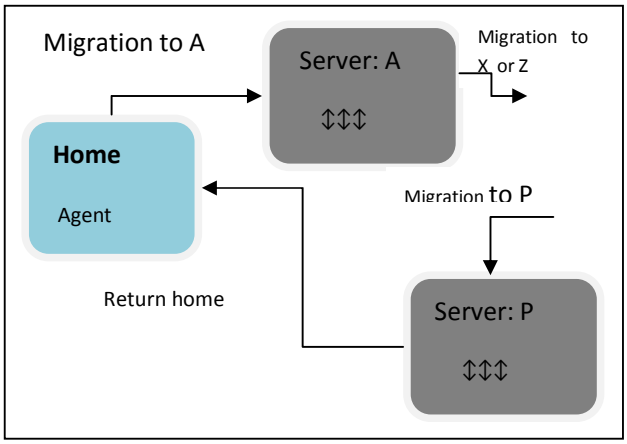

Fig. 1.The Mobile Agent Workflow

1.2 Applications of Mobile Agents

A) **Reduction of the network traffic** :

Mobile agents reduce the data flow in the network by dispatching it to a destination host and Making the conversation in destination host for the agent to compute. The benefit of the mobile agent is that the computation moved to the place where the data is available.

B) **Reduce network latency** :

Real-time systems need to respond to real time changes in their environments. Controlling such systems through a large network of a substantial size, involves considerable latencies. For critical real-time systems, such latencies are not acceptable. Mobile agents offer a solution, because they can be dispatched from a central controller to act locally and execute the controller's directions directly.

C) **Asynchronous autonomous interaction***:*

Mobile agents can be delegated to perform certain tasks even if the delegating entity does not remain active. This makes it an attractive for mobile application and disconnected operations.

D) **Dynamic Adaption**:

Mobile agents can sense with their execution environment and react autonomously to changes.

E) **Robust and fault-tolerant**:

If a host in which agent executing its code is being shut down, all agents executing on that Machine are warned and given time to dispatch and continue their operation on another host in the network.

F) **Client Customization**:

In distributed computing models,. Clients are confined to the service provided by the server. In case the clients want to have a new service, the service must be installed on the server. But with mobile agent, the clients are virtually installing programs on to the server when the mobile agent migrates from one host to the others.

G) **Efficiency savings***:*

CPU consumption is limited, because a mobile agent execute only on one node at a time. Other nodes do not run an agent until needed.

H) **Space savings***:*

Resource consumption is limited, because a mobile agent resides only on one node at a time. In contrast, static multiple servers require duplication of functionality at every location. Mobile agents carry the functionality with them, so it does not have to be duplicated.

1.2 Related Works

Mobile agent's security problems divided into three threats: the mobile agent's transfer security problems, mobile agent platform's security problems and mobile agent security issues in malicious platform. Mobile agent security issues in malicious platform are very difficult to solve because agent's run-time code, data and communications are fully exposed to the host which agent running.

Mobile agents must have strong security properties to protect themselves and the collected data while leaving their homes and migrating to other potentially malicious server. A malicious server may expose, modify, insert or truncate data the agent collected from other previously visited servers to benefit it self .This problem is serious for free roaming mobile agents .

The mechanism used to protect data of free roaming mobile agents can be mainly divided in to two categories. Detecting mechanism and avoiding mechanism. Sorting Detecting Mechanism into

(i) TTP (Trusted Thirty Party)

(ii) (ii) Using Chain relation.

(iii) Multi agent co-operation

(iv)

We use TTP to record itinerary information directly or indirectly. The main problem is that we need one TTP at least, and the mobile agent need to communicate with it, so the TTP will

become a bottleneck and even cause single-point failure. It is not easy to find a TTP in the open internet.

In chain relation we form a chain relation among previous and following hosts' computing data generated by the mobile agent. They can detect the modification made on the data by malicious hosts with this chain relation. Different chain relations decide different mechanisms and also decide their ability in detecting colluded truncation attacks.

Reference [1] has designed a trust model TAMAP. The model obtains trust score through interaction between hosts. Mobile agent adjusts its implementation according to trust score. The model is mainly consists of protection agreements, trust estimation mechanism and dynamic self-adaptive process, which can resist colluded truncation attack. Compared with other models, TAMAP has an advantage of simplifying complex, costly infrastructure, such as reputation database. But TAMAP is modeled on a real-world trust reputation mechanisms to achieve network security, need strong network management as support, no matter how sophisticated the design, how thoughtful the consideration it is difficult to be applied in practice.

Reference [2] uses partial result encapsulation for chain association certification, use digital signatures and Hash function to form association between results of two hosts adjacent. Through this association the mechanism can prevent data changed maliciously, but it cannot resist colluded truncation attack.

Reference [3] is an improvement of reference [2]. Data generated by agent on each host have relationship with data generated on consecutive two hosts before and after. The mechanism can resist two or noncontiguous multi-hosts' colluded truncation attack, but cannot resist contiguous multi-hosts' colluded truncation attack.

In reference [4] agent copies itself in each host, then the copy reach next host and test the host is malicious or not if not, the host become next hop, otherwise, agent chooses another host, but the mechanism cannot resist multi-hosts' colluded truncation attack

2 Trust Enhanced Symmetric Key Cryptography

To solve the data security problems of free roaming mobile agent, we establish following model

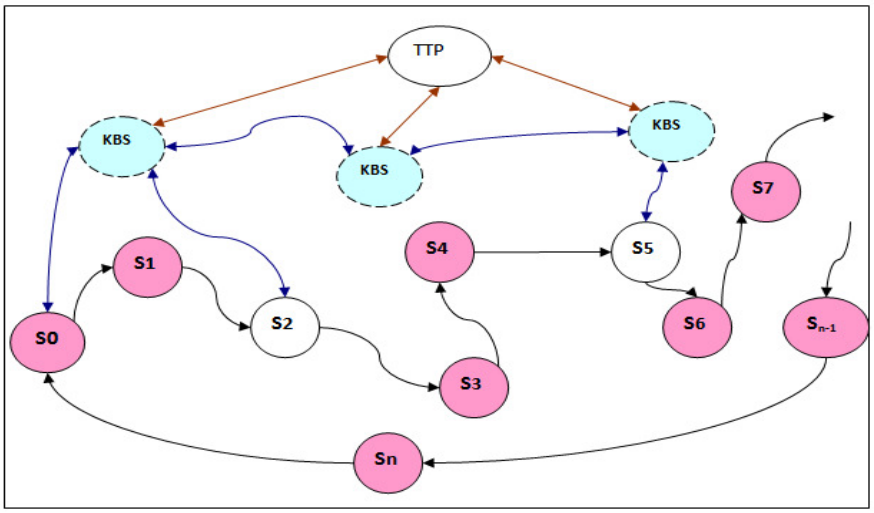

Fig. 2 Agent Migration With KBS

As shown in fig 1, S_0, S_1, S_2.... . ,S_n are network hosts, So is a task Sponsor or initiator or Originator or creator . Agent starts from So, chooses its next hop according to certain principles and on the new host it collects data , then chooses next hop, This process continues until the completion of scheduled tasks and return to S_0.

Now let's assume mobile agent's route is S1, S2 .....Sn. Among these hosts some malicious hosts may exist, and even there are multiple malicious hosts, which conspire with each other to modify, cut off data collected by agent to achieve their attack purpose.

Based on model above, we established trust enhanced symmetric key cryptography on the following premises

Premise I: Malicious hosts have impact on mobile agent's choosing its next hop if mobile agent is running on it. Even worse, malicious hosts can control its next hop.

Premise 2: On non-malicious hosts, mobile agent's choosing its next hop mainly depends on current network environment and so on but not depend on current host. We can assume that each host has equal possibility of being next hop.

Premise 3: We have established Symmetric key certificate system. Each KBS (host) has Key agreement with originator. KBS maintain trust host list by interface with TTP.

Premise 4: Mobile agent's route is generated dynamically. KBS should not only have ability to resist colluded truncation attack, but also meet following security requirements:

(1) Data confidentiality: only S0 can obtain data without encryption generated by hosts which agent chooses.

(2) Non-repudiation: No host which agent has run on can deny data results generated on them and agent's passing through.

(3) Anonymity: for some certain purpose, No host knows the task sponsor and mobile agent's migration path.

(4) Integrity: if malicious host modify other hosts' data mechanism can detect illegal alteration.

(5) Anti-insertion-attack: any host cannot have the data to insert redundant data.

2.1 The Basic Idea

General idea of Trust enhanced symmetric key cryptography is based on encryption, decryption and Signature Verification Principle. By using KBS, data was encrypted into a divisible whole for protection. Getting identity information from trusted third party via KBS to make key agreement with trusted host. If the host is trusted host, verification and summarization of previous offers are done. If it is not a trusted host offer collected according to security policies to resist colluded truncation attack.

When agent reaching a host, it concatenates the data generated on it with data carried by agent then encrypts them and verifying identity information. KBS gets identity information from the trusted third party periodically. When agent comes back to host after completes its work without any attack.

2.2 Model Description

We descript TKBS through 5 stages.

1. S0 key agreement with KBS
2. Task sponsor S0 generates MA
3. MA migrates to host Si
4. MA migrates to Trust host Si

5. MA reaches Task sponsor
6. MA finds results at Task sponsor

In 6 stages, MA is a agent, who finishes the task through migration among hosts. KBS is a host which establishes key agreement with task sponsor, trust hosts, and other KBS. With KBS information, Trust host react as task sponsor.

2.2.1 Symbol description

S_0	Mobile agent's task sponsor
S_i	The no.i host
MA	Mobile Agent, an mobile agent to finish the task
Third party	third party
d_0	S_0's logo, generated by S_0
d_i	Data collected on host S; without encryption
D_i	Data produced by agent running on host S; with
H	Hash operation
K	Symmetric key used for encryption
$Enc_K(m)$	Encrypt message m using key K
$Sig_{pr}(m)$	Sign message musing S_i 's private key
$Dec_K(m)$	Decrypt message m using key K
A →B: m	A sends message m to B
$Enc_{Tpu}(m)$	Encrypt message m using temporary public key
$Dec_{TpR}(m)$	Decrypt message m using temporary private key

Table 1: SYMBOLS USED IN SYSTEM MODEL

2.3 The Proposed Algorithm

1. S_0 's key agreement with KBS

Task sponsor (initiator) is S_0,. KBS is host trusted by TTP which maintains trusted host list. Before creating Mobile Agent, task Sponsor makes key agreement with KBS by using Deffie-Hellman Key exchange algorithm.

Deffie-Hellman Key exchange algorithm

q α are known to S_0,KBS and trusted hosts.

q is prime number and an integer α is primitive root of q.

S_0:

Generate random number X_A

Calculate $Y_A = \alpha^{X_A} \bmod q$

$S_0 \rightarrow$ KBS: Y_A

KBS:

Generate random number X_B

Calculate $Y_B = \alpha^{X_B} \bmod q$

$S_0 \rightarrow$ KBS: Y_B

For calculate key K for encryption

S_0: $K = (Y_B)^{X_A} \bmod q$

KBS: $K = (Y_A)^{X_B} \bmod q$

Temporary pubil key and private kye is created by So

S0: (tPU,tPR)

S0→KBS : $PK = Enc_K$ (tPR)

2. S_0 generates Mobile agent MA

Task sponsor (initiator) is S_0, who generates MA. S_0's logo is d_0. S_0 determines its next hop S_1 according to current network environment and counts:

$$D_0 = Enc_{tPU} (S_0 \| S_1 \| d_0 \| Sig_{pr0} (d_0))$$

S_0 uses its private key to sign on d0, which then be encrypted to form D0.

$h_0 = H (S_0 \| S_1 \| d_0 \| Sig_{pr0}(d_0))$

$D_0' = D_0 \| h_0$

TA migrates to S_1 with D_0'.

$S_0 \rightarrow S_1 = D_0'$, PK

3. MA migrates to host S_i

When reaches host S_i with data D_{i-1}, MA generates original data d_i through computation on host Si then chooses its next hop S_{i+1} according to current environment. S_i signs on d_i with its private key, concatenate d_i. S_i, S_{i+1} and D_{i-1} to form:

$$D_i = Enc_{tPU} (S_i \| S_{i+1} \| d_i \| Sig_{pri} (d_i) \| D_{i-1}')$$

Then compute Hash value:

$h_i = H(S_i \| S_{i+1} \| d_i \| Sig_{prl}(d_i) \| D_{i-1}')$

$D_i' = D_i \| h_i$

MA migrates to next hop with encrypted data D_i':

$Si \rightarrow S_{i+1}: D_i'$

Meanwhile, Si sends its identity information to task initiator S_0:

$Si \rightarrow S_0: Enc_{tPU} (Sig_{pri} (S_i) \| S_i)$

S_0 can obtain Si's identity through decryption and certificating signing when receives data above:

$Dec_{tPR} (Enc_{tPU} (Sig_{prl} (S_i) \| S_i))$

4. MA migrates to Trust host Si

We assume that MA to migrates to Trust host Si after passing through n hosts. In trust host agent has two works.

1. Verify and collect all the previous host's data
2. Current host data collection

To verify, MA needs S0 's tPR key . So that trust host gives request to KBS,

And gets S0 's temporary private key .

KBS: $Dec_K(PK) = tPR$

Si→ KBS : $Enc_{PRSi}(S_0)$

KBS→ Si: tPR

Now MA is carrying with data D_n'.

$D_n' = D_n \| h_n.$

$D_n \| h_n = Enc_{tPU} (S_n \| SA\| d_n \| Sig_{prn}(d_n) \| D_{n-1}) \| H(S_n \| SA\| d_n \| Sig_{pm}(dn)\| D_{n-1})$

$= ...$

In trust host MA retrieves D_n and h_n through D_n'. MA decrypts D_n using temporary private key. Operate hash function on decryption result, then compare hash value with h_n. If they are same to each other, we can be sure that data has not been damaged on host S_n. Then MA decrypt D_{n-1}, continue to this proceed. If hash value is different from corresponding h_i, that is to say, data has been damaged. After finishing the decryption, MA can obtain data collection data and address collection add1.

$data = \{d_0, d_1 ... d_n\}$

$add1 = \{S_1, S_{2.....}S_n\}$

After steps above, if all the data has not been illegally modified through verify, MA encrypts original data and sends them to S_0, otherwise sends information to show damage .

We use 1 bit to express data received by MA is reliable or not. Bit 1 represents success (data received by S_0 is correct), 0 means failure (data received by S_0 has been damaged). Sending information as below when data has been transmitted correctly:

$S_i \rightarrow S_0: Enc_{tPU} (1 \| Sig_{prsi} (S_i) \| d_i \| d_2 \| \cdots \| d_n)$

Sending information as below when data has been damaged:

$S_i \rightarrow S_0$: Enc_{tPU} (0 || Sig_{prSi} (S_i) || S_j || ... || S_j)

Si. ... ,Sj; are malicious hosts judged by S_0.

When receiving Trust host Si's data, agent sponsor decrypts the data with Temporary private key then decide what the results mean according to the one bit 1 or 0.

Trust host S_i determines its next hop S_{i+1} according to current network environment, and counts similar at S_0:

$D_i = Enc_{tPU}$ (Si || S_{i+1} || d_i || Sig_{pri} (d_i))

That is to say, Si uses its private key to sign on d, which then be encrypted to form D_i. Then count:

$h_i = H(S_i || S_{i+1} || d_i || Sig_{pri}(d_i))$

$D_i' = D_i || h_i$

TA migrates to S_i with D_i'.

$S_i \rightarrow S_{i+1} = D_i'$

5. MA returns to Task Sponsor S_0

We assume that MA returns to task sponsor after passing through n hosts. Now MA is carrying with data D_n'.

$D_n' = D_n || h_n. = Enc_{tPU} (S_n || S_{n-1} || d_n || Sig_{prn}(d_n) || D_{n-1}) || H(S_n || S_{n-1} || d_n || Sig_{pm}(dn) || D_{n-1}) = ...$

MA retrieves D_n and h_n through D_n'. MA decrypts D_n using its private key. Operate hash function on decryption result, and then compare hash value with h_n. If they are same to each other, we can be sure that data has not been damaged on host S_n.

Then MA decrypt D_{n-1}, continue to this proceed. If hash value is different from corresponding h_i, that is to say, data has been damaged. After finishing the decryption, SA can obtain data collection data and address collection add1

data = {d_o, d1 ... d_n}

add1 = {S_1, S2…Sn}

Meanwhile, S_0 receives identity information from hosts agent passed:

Enc_{PK} (Sig_{prl} (S_1) || S_1), Enc_{PK} (Sig_{pr2} (S_2) || S_2),… Enc_{PK} (Sig_{pm} (S_n) || S_n)

Through decryption and certificate signing on data above,S_0 can obtain address collection add2:

add2 = {S_1, S_2,. ..S_i ...S_n}

If add2 is the same as add1, damage has not occurred, otherwise data has been damaged.

6. MA finds final result at S_0

After steps above, if all the data has not been illegally modified through verify, MA encrypts original data and sends them to S_0, otherwise sends information to show damage

We use 1 bit to express data received by MA is reliable or not. Bit 1 represents success (data received by S_0 is correct), 0 means failure (data received by S_0 has been damaged). Sending information as below when data has been transmitted correctly:

S_0: Enc_{tPU} (1 || Sig_{prsi} (S_i) || d_i || d_2||⋯ || d_n)

Sending information as below when data has been damaged:

S_0: Enc_{tPU} (0 || Sig_{prSi} (S_i) || S_j || ... || S_j)

Si. ... ,Sj; are malicious hosts judged by SA.

When receiving S_i's data, agent sponsor decrypts the data with its private key then decide what the results mean according to the one bit 1 or 0.

3 Explanation and Analysis

3.1 Realization of trusted third party (KBS)

As mentioned above, KBS is the main factor of success. Because KBS act on behalf of TTP and serve as key distribution center. There are many methods of choosing n trust host. For example, analysis and improvement on a secure .threshold group signature scheme, which can resist colluded attack and forge signature attack; Reference [9] designed and realized a trusted computer program based on embedded way. The program embeds ESM (Embedded Security Module) on General-purpose computer motherboard to achieve more complete security architecture.

3.2 Efficiency of TKBS

We assume there are n hosts those agents passing through in TKBS. It will need 2n+2 encryptions (include signature) and 2n+2 decryptions (include signature verification). The efficiency of encryption and decryption times is O(n) if no single Trust host is not exit. If n trust nodes are avail, we need less than 2n+2 encryptions and decryptions. The efficiency of encryption and decryption times is less O (n) if n trust host is available in a chain

4 Security Analysis

We analyze security of TKBS as below:

• General security analysis

(a) Data confidentiality: data D_i' carried by agent passing through host Si is encrypted in S_0's Public Key, So only S_0 can decrypt Di' in its private key to get original data. S_0 can obtain the original data by Trust host S_i. However, other hosts in agent's migration can only know original data generated by it because of lacking S0's private key.

(b) Non-repudiation: host S_i uses its private key to sign on original data d_i generated on it, that is $Sig_{pri}(S_i)$,these signature information will be certified on S_i. So other hosts cannot deny data generated on it.

(c) Anonymity: in data D_i, any information about identity is encrypted in S_0's public key, so only S_0 can obtain hosts' identity through decryption. Other hosts cannot know which host MA has passed except its last hop, Trust host and next hop. Similarly, other hosts don't know the task sponsor S_0

(d) Integrity: if a malicious host S_{k+i} modifies data D_{k-1}' carried by MA, S_0or trust host can testify whether the data was damaged or not because the data has been encrypted in S_0's public key. When the data returns to S0, S0 can retrieve $D_{k-1}$and h_{k-1} through decryption. If D_{k-1} and h_{k-1} is different, that shows attack has happened.

5 CONCLUSION

To resolve the problem of mobile agent's security, especial on attacks on mobile agent data, paper analyzed current protection mechanism and put forward KBS to enforce its security with symmetric key cryptography. The analysis shows that Trust enhanced symmetric key cryptography can protect data of free roaming mobile agent effectively and realize some security needs such as data confidentiality, integrity, and anonymity.

ACKNOWLEDGMENT

I wish to express my sincere gratitude to my beloved Panel member **Dr.A.Kannan,** Professor**,** Computer Science &Engineering ,Anna University Chennai and **Dr.N.Sreenath ,**Professor Computer Science &Engineering ,Pondicherry University, Chennai for providing an opportunity and extending his timely assistance.

I am very much indebted to **Dr.P.Narayanasamy,** Director, Anna university of Technology, Chennai for extending his timely assistance and kind co-operation for this Work.

I take immense pleasure in expressing my sincere gratitude to **Dr.C.Jayakumar ,** Professor, Department of Computer Science & Engineering ,RMK Engineering college ,Kavarai Pettai,Chennai for his constant encouragement, valuable guidance and personal support during the work period, without which this work would not have attained completion.

Finally, I would like to express my sincere thanks to **my Family** without whom this work would not have been completed successfully.

References

1. Salima Hacini,Zahia Guessoum,Zizette Boufaida "TAMAP: a new trust-based approach for mobile agent protection " Journal of Compute Viro 1.2007 vol 3, Page no 267-283.
2. M. Yao, E. Foo, E. P. Dawson and K. Pengo An Improved Forward integrity Protocol for Mobile Agents. In proceedings of 4th International Workshop on Information Security Applications (WI SA 2003), Computer Science, 2004, pp; 272--285.
3. Darren Xu , Lein Ham, Mayur NarasimhanAn , Luo Junzhou" Improved Free-Roaming Mobile Agent Security Protocol again Colluded Truncation Attacks. Proceedings of the 30th Annual (COMPSAC'2006), IEEE Computer Society 2006 Volume 2, pp: 309- 314.
4. Y.c. Jiang, Z.Y. Xia, Y.P. Zhong, S.Y. Zhang. Defend mobile agent against malicious hosts in migration itineraries [J]. Microprocessors And Microsystems 28 (2004), pp: 53 1-546.
5. S Green, L Hurst. Software Agents: A Review [J]. Trinity College Dublin Broadcom Eireanm Research Ltd, I997
6. Zhang Yunyong, Liu Jinde., Mobile agent technology [M].Beijing: Tsinghua University Press, 2003.
7. Shane Balfe, Eimear Gailery. Mobile Agents and the Deus Ex Machina 21st International Conference on Advanced Information Networking and Applications Workshops (AINA W'07), pp: 486-492.

8. Wu Shuiqing, Wang Guocai, Yang Jian. Analysis and improvement on Secure threshold group signature scheme [J].Computer Engineering And Applications.2008, 44(26), pp: 113-115
9. Xiao Jing, Yu Chao. Design and Implementation of a Trusted Computer System [J]. Journal of Wuhan University of Technology.2007, 29(7) 1671- 4431 (2007) 07- 0148-04.
10. F.Silva and R.Popsecu-Zeletin, "mobile agent-based transaction In open environments." IEICE/IEEE JOINT Special Issue Autonomous De-centralized Systems, vol.E83-B, no.5, pp.973-987, May 2000.
11. M.Breugst, I.Busses, S.Covaci, and T.magdanz "Grasshopper- A Mobile agent platform for IN based service environments", In Proc. IEEE Intelligent Networks Workshop, Bordeaux, France, May 1998, pp279-290.
12. T.Eiter, E.Erdem, and W.Faber. Plan reversals for recovery in Execution monitoring. In Non-monitoring Reasoning, 2004.
13. A.Fedoruk and R.Deters. Improving fault-tolerance by Replicating Agents. In AAMAS'02, pages 737-744. ACM Press, 2002.
14. D.Morley and K.Myers. The SPARK agent framework. In AAMAS'04 NY NY, 2004H.
15. S.Pears, J. Xu, and C.Boldyreff. Mobile agent fault tolerance for information retrieval application: An exception handling Approach. In the Sixth international Symposium on Autonomous Decentralized Systems, 2003.
16. A.Unrh, H.Harjadi, J.Bailey. Semantic-Compensation-Based recovery in Multi-Agent Systems. In IEEE.2005
17. Ashish Kumar Srivastava et. al Secure Mobile Agent based Information Gathering in Wireless Network. / (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 05, 2010, 1685-168

G.Geetha received her B.E computer science and engineering from Bharathiar University and her M.E computer science from sathayabama university .Now she is working as Assistant professor in Jerusalem College of Engineering, Chennai, and doing research in Anna University Chennai

Professor C **Jayakumar** received his M.E computer Science and PhD from Anna University Chennai. Now he is working as Professor in CSE, RMK Engineering College Chennai.