Journal of Computer Science 10 (7): 1207-1215, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.1207.1215 Published Online 10 (7) 2014 (http://www.thescipub.com/jcs.toc)

SECURE COMMUNICATION PROTOCOL FOR PROTECTING COMPUTATION RESULT OF FREE ROAMING MOBILE AGENT

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

[1]Department of CSE, Jerusalem College of Engineering, Anna University, Tamilnadu, India
[2]Department of CSE, RMK College of Engineering, Anna University, Tamilnadu, India

Received 2013-11-28; Revised 2014-01-06; Accepted 2014-02-19

ABSTRACT

Mobile agent plays an important role in developing applications of open, distributed and mixed environments, such as the internet. Mobile agent or mobile software agent is piece of software that can operate autonomously to accomplish user assigned task. To explain more, mobile agent is the process which can migrate to hosts autonomously. As an agent travels to do execution in different environment in different host or servers, the agents are in need of protecting themselves and their data from various types of attacks. So providing security to the mobile agent (static code) and its data (dynamic code) is an emerging need in Mobile Agent Technology. The change in Mobile Agent (MA) code can be identified due its static nature where as finding change in mobile agent data is the biggest challenge especially in malicious host's attacks. This study presents Clone Return Process (CRP) method to protect the data of free roaming mobile agent against colluded truncation attack. By using CRP, malicious host are identified and recovery of mobile agent is easily done. So free roaming mobile agent communicates with other servers and protects its computation results (data) in an efficient way.

Keywords: Mobile Agent, Free Roaming, Protection, Attacks, Data Security, Code Integrity, Data Integrity, Migration Code, Aglets

1. INTRODUCTION

Mobile agent is a emerging paradigm for Distributed Computing. Mobile Agent has been developed quickly and widely used by researcher to satisfy many distributed applications (Wang *et al*., 2011). The software programs that live in computer networks are called mobile agents which have the feature of autonomy, social ability, learning and most important mobility. They can migrate from one host to another host to perform computations for fulfilling the goals of the user. Comparison of mobile agent technology with traditional methods is shown in **Table 1**. Free roaming mobile agents is a kind of mobile agent that roams in the network to do task of its owner without any given itinerary path.

When a mobile agent decides to migrate, it saves its own state and transports this saved state to next host and resume execution from the saved state on the remote host. In strong mobility mobile agents resumes its execution at exact the state where it stops the execution in previous host where as in Weak mobility, mobile agents does not resume its execution at same state where it stops the execution in previous host.

Figure 1 describes the General the mobile agent system which consists of Home Sever (originator), Mobile Agent and Server (The host that MA moves). The agent starts its execution after it reaches server. There is no communication between home server and other severs except the server next to originator and last server.

A mobile agent consists of three components shown in **Fig. 2**. Those are Code (program that defines the agent's behavior), State (the agent's internal variables which allow it to resume its actions after moving to another host) and Attributes (information about its origin, owner, agent identity I_a, its movement history, data d_i, resource requirements and authentication keys). Mobile Agent can access the attributes but it cannot modify them. Model notations and cryptographic notations are shown in **Table 2**.

Corresponding Author: Geetha, G., Anna University, CSE Jerusalem college of Engineering, Tamilnadu, India

Table 1. Comparison of mobile agent technology with traditional methods

Paradigms /attributes	Mobile agent	Remote evaluation	Client server
Implementation	Hard	Easy	Very easy
Security	Very low	Low	Very high
Performance	High very	High	Low
Mobility of elements			
a) Data	semi mobile	static mobile	mobile
b) Code	Mobile	Mobile	Static
c) Stack	Mobile	Mobile	Static
Itinerary	Static/dynamic	Static/dynamic	Static
Mobility	Code to data	Code to data	Data to code
Platform	Dependent	Independent	independent
Programming code	Hard	Hard	Easy
Tool	Aglets	Aglets	CORBA

Table 2. Model notations and cryptographic notations

S_i	i^{th} the host
D_i	Data collected at Si
ED	Enhanced data
OD	out data
Enc PbS0	Encryption using Public key of originator S_0
Sig	Encryption using Private key of originator S_i
EH_{codei}	Encrypted hash code foe mobile code at i^{th} host
H(x)	Hash of X
Msg	Message
α	Threshold value for data size
β	Threshold value for host count
γ	Threshold value for execution time
HC	Host count
T_{tot}	Total execution time
t exe i	Execution time of Si
t tra i	Time taken travel from S_{i-1} to S^i
σ	Delay time
τ, υ	Constants
α	Threshold value for data size

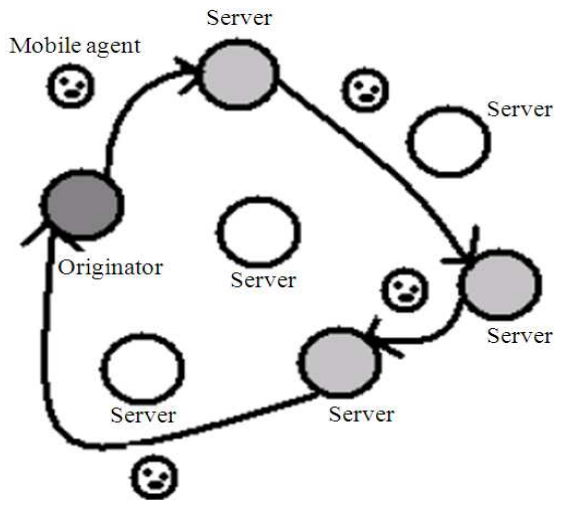

Fig. 1. Migration of Mobile Agent from originator to various hosts

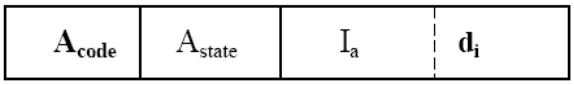

Fig. 2. Components of Mobile Agents

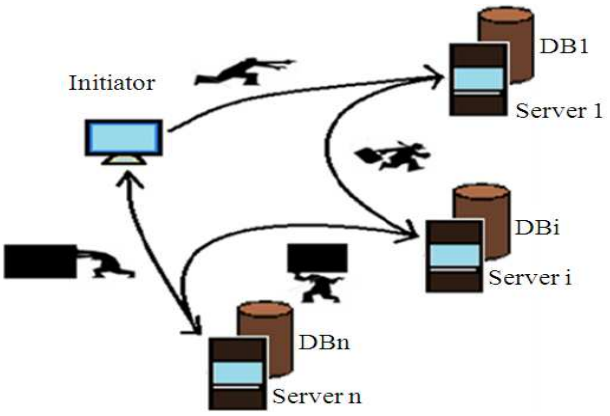

Fig. 3. Application example-Mobile agent as data collect

1.1. Security Issues in Mobile Agents

Though Mobile agent moves from one host to another host in Network, the security of mobile agent plays wide role in mobile agent technology. The Mobile agent security can be classified as follows **Fig. 3**.

In **Fig. 4**, it is clearly shown the classification of mobile agent Security. In another aspect i.e., according to attackers of agent Security classified as:

- Malicious Host Attacks the Agent
- Malicious Agent Attacks Agent
- Others Attacks the Agent

Here the security of agent platform is not mentioned due to focus the study of Data security in free Roaming the Mobile agent from attack of malicious Host.

Malicious Host May try to tamper the mobile agent's Static code(agent program) or Dynamic code (data) even Both while the agent is in migration to process computational results(data of agent). To provide secure agent execution in various host or severs, the Security of agent mechanism divide into two major categories as follows:

- Detection mechanism
- Avoiding mechanism

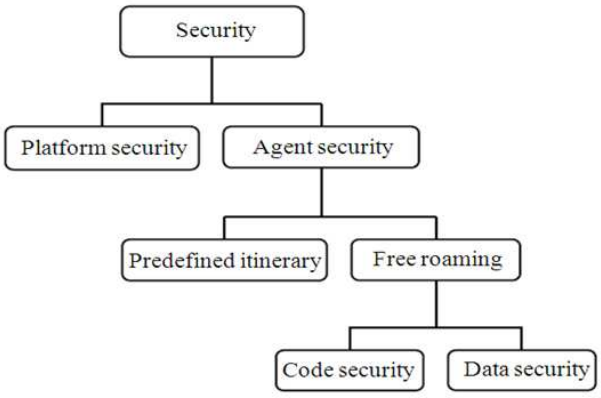

Fig. 4. Classification of mobile Agent system Security

1.2. Detection Mechanism

To find wether the host is malicious host or not, the methods used are:

- Trusted Third Party (TTP)
- Chain relation method
- Multi agent co operating mechanism

1.3. Trusted Third Party (TTP)

TTP is used to protect the mobile agent's data by recording itinerary information directly or indirectly. Mobile agents need at least on TTP to communicate for their execution and protection.

1.4. Chain Relation Method

Mobile agents form a chain relation among the previous and the following hosts where they compute and collect data. If a malicious host modifies the data, the mobile agents can detect the modification through this chain relation. Different chain relations with different mechanisms used to detect various attacks especially colluded truncation attacks.

1.5. Multi Agent Co Operating Mechanism

More than one agent involves in mobile agent applications. Mobile agents are classified in to different classes. For example Task agent, secondary agent, data computation agents, data collection agents.

1.6. Avoiding Mechanism

This Mechanism gives idea of that the agent should not move to malicious host or un trusted host to protect the data. To list Avoiding mechanism used in Mobile Agent Technology, we have:

- Trust computing
- Dynamic interaction
- Private Key consignment

1.7. Trust Computing

In Trusted computing, mobile agent execution is based on trust and reputation values of the host. Trust values of host are calculated by various methods to protect mobile agent and its data.

1.8. Dynamic Interaction

During the Information collected the interaction an environment key is generated. That key allows to infer the host‘s trust degree and permits the mobile agent to adapt its execution.

1.9. Private Key Consignment

Private Key Consignment method protects the private key of the agent by consigning the private key to a tamper proof hardware which enables convenient and secure use of the private key.

In rest of the paper is organized as follows: Section 2 discusses a background on the threats and some related works on mobile agent security. Section 3 is about security requirements of mobile agent systems. Section 4 offers a detailed description of the proposed protocol. Section 5 presents implementation details of the protocol and an analysis. As a conclusion, section 6 presents synthesis of work and its limitations.

1.10. Related Works

Methods used to protect mobile agents data includes:

- Partial Results Authentication Code (PRAC)
- Set authentication code
- Ring signature
- Chan hash chaining

Partial Result Authentication Code (PRAC) proposed to ensure the integrity of data collected from hosts by Yee (1997). In this agent and its originator maintain a list of secret keys or key generation function used to calculate Message Authentication Code (MAC) upon the result of each host. The agent uses a key to encapsulate the collected offer and destroys the key. Yee defines forward integrity in which the first visited malicious host cannot modify or forge any PRACs of previously visited hosts.

Extended from of Yee's Partial Result Authentication Code (PRAC) is KAG method which proposed by Karjoth *et al*. (1998). In KAG, each host generates a signing key for its successor and certifies the corresponding verification key. Using the received

signature/verification key pair, a host signs its partial result and certifies a new verification key for the next host.

Marikkannu *et al*. (2011) developed a protocol, which is immune to most types of known attacks. The protocol uses the techniques of trip marking, digital signing and MIP, to overcome most types of attacks. Enhanced KAG Scheme is proposed by Cheng and Wei (2002) to Defend two colluder truncation attack. In this scheme, a host is first required to get a counter signature of its partial result from its predecessor before sending it to the next host.

Linna and Jun (2010) proposed the Signature Trust Chain Mechanism (STCM) in which data was encrypted as a whole for protection and identity information was sent to trusted third party to resist any attack. This mechanism uses the TTP for verification. In this mechanism two types of agents are used. It compares the path of the collected data with path of identity information. If any mismatch occurs the host will identify that there is an attack.

Method to use integrity measurement feature and the integrity reporting feature is proposed by Silei *et al*. (2008). In this Integrity measurement is the process of obtaining metrics of platform characteristics where as integrity reporting is the process of attesting to integrity. But this mechanism has two agents, task agent and secondary agent platform configuration register.

Signature Trust Chain Mechanism is proposed Linna and Jun (2010). In STCM data was encrypted in to a whole for protection and sending identity information to trusted third party to resist attack. This mechanism use TTP for Verification.

Songsiri (2005) proposed method using TTP for protecting data of mobile agents. It has two protocols: Online TTP and off line TTP. Again this method is in need of TTP.

The agents transfer commitments to other Co-operating Agents method is described by Roth (2001) in which those agents performs task like storing gathering and verifying But idea behind this approach is TTP.

A Security protocol that protects mobile agents from malicious platform attacks through the use of reference clone is proposed by Benachenhou and Pierre (2006). This clone, a copy of the agent is executed on trusted servers in parallel in order to verify the mobile agent execution.

Software architecture by Garrigues *et al*. (2010) is based on implementing agent-driven approach using. That provides Mobile Agents with a code that manages their own protection and execution. That code is referred to as the agents control code.

Raji and Ladani (2010) proposed a protocol in any host cannot learn either the true identity of the agent owner, or the path that the agent has traversed through so far and both of the agent execution results and the agent itinerary are maintained in the agent state in such a way that its owner can only be aware of them.

Two advanced models are proposed by Venkatesan and Chellappan (2010b) for platform and agent code protection with the policy and the additional signature to improve the efficiency of the existing Malicious Identification Police model for scanning the incoming agent to detect the malicious activities and to overcome the availability of vulnerabilities in the existing Root Canal algorithm for code integrity checks.

Senthilnathan and Purusothaman (2012) presents the results depicting the advantageous of using agents in data replication, which includes reduction in data communication cost under different circumstances like change in mobility of nodes, read write ratio of nodes and replication schema.

Ogunnusi and Razak (2013) proposes a fault-tolerant key distribution protocol for distributed mobile agents (communicating entities) in network intrusion detection system to facilitate hitch-free collaboration geared towards intrusive packets detection in Wireless Local Area Network (WLAN).

From the above analysis, several methods are proposed to protect mobile agent, mobile agent data and mobile agent itinerary. Each and every mechanism has its own strength and weakness with reference to different environments. But all proposed mechanisms fail to protect mobile agent against colluded truncation attack. In this study, Clone Return Process method is proposed to protect mobile agent, mobile agent's data and its itinerary against all most all type of attacks especially colluded truncation attack.

1.11. Security Requirements

Mobile agent security rests on Confidentiality, integrity and availability as like computer security.

1.12. Confidentiality

Confidentiality is the concealment of information or resources.

1.13. Data Confidentiality

Data confidentiality defines the protection of data from unauthorized disclosure. The originator (host on agent created) only can obtain data which computed from other hosts.

1.14. Forward Privacy

The originator can only extract the visited host's integrity.

1.15. Integrity

Integrity refers to trustworthiness of data or resource which should be prevented from improper or unauthorized change. Integrity includes data integrity, origin integrity.

1.16. Data Integrity

Data Integrity is an assurance that data received are as exactly as calculated and sent by the host to which agent has moved on. So the intermediate host cannot modify, insert delete previous host's data

1.17. Code Integrity

Code Integrity is a assurance that code received are exactly as the code sent by the originator (host in which agent has created). So intermediate host cannot modify, insert delete code of mobile agent.

1.18. Authentication

Authentication provides assurance that source of received data is as claimed. So that any host where agent migrates should authenticate the data which computed on its platform.

1.19. Anti-Insertion-Attack

Any host cannot have the data to insert redundant data.

1.20. Truncation Resilience

The chain of encapsulated offer can be broken in between colluded malicious hosts.

1.21. Malicious Host Identification:

The originator can identify the malicious host by verifying the chain of encapsulated offer.

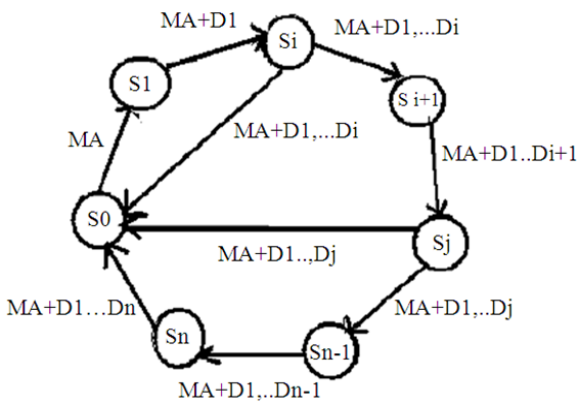

Fig. 5. Clone return process method

1.22. Availability

Availability refers to the ability to use the information or resource desired. Attempts to block availability called denial of service attacks.

1.23. Non-Repudiation

Non-repudiation provides protection against denial by one of the entities involved in a communication. So that no host which agent has moved on can deny data results computed on them and agent's passing through.

1.24. The Proposed Protocol

The proposed Clone Return Process Method (CRP) consists of agent migration, code integrity verification, data collection, encryption and hashing, signing, threshold checking, cloning and returning (**Fig. 5**). Instead returning to originator after migration to n host, the mobile agent can interact with originator in between the collection of data from other servers based on some threshold:

1) Creation of mobile Agent at S_0
   - i. H_{code} = H(Agent Byte Code)
   - ii. $EH_{code0} = E_{PR0}(H_{code})$
   - iii. $S_0 \rightarrow S_i$: EH_{code0}

2) Execution of mobile Agent at Remote host Si
   - i. Received H_{code} = DPUi-1($EH_{code\ i}$)
   - ii. H_{code} = H(Agent Byte Code)
   - iii. If (Received H_{code} == H_{code}) then

A. ODi-1 = ED_{i-1}|| H(ED_{i-1})
B. If (Received H(ED_{i-1})=H(Received ED_{i-1}))
   - B.1. Collect D_i
   - B.2. $ED_i = E_{PU0}(S_i || S_{i+1} || Sig(D_i) || H(D)_i || OD_{i-1})$
   - B.3. OD_i=ED_i|| H(ED_i)
   - B.4. $EH_{code} = E_{PRi}(H_{code})$
   - B.5. If threshold checking is not true
     - B.5.1. Si→S_{i+1}: EH_{codei}||OD_i
     - Else
     - B.5.2. Si → S_0: OD_i
   - B.5.3 $S_i \rightarrow$Si+1: EH_{codei} || OD_i
   - Else
   - $S_i \rightarrow S_0$: Msg (previous data mismatch)

Else
$S_i \rightarrow S_0$: Msg (Error code)

Three types of protocols are used based on the different constrains for returning to the originator. Threshold Values α, β, γ is chosen for Data size (α), number of host to be migrated (β) and execution time (γ). After retrieving the data from other host it verifies the threshold value. The mobile agent either migrates to the next server or does both the process migrating to the next

sever as well as to the originator by cloning if the threshold value is reached. Clone Agent will return to originator with partial data and the original agent migrates to the next server.

1.25. Data Size as Threshold

The mobile agent migrates to N number of host to collect data from each host. After migrating to N number of host the agent will collect the data for processing in the originator. While executing code in unknown severs, the agent may face various types of attacks especially multi colluded truncation attack. Confidentiality and Integrity must be ensured when the mobile agent migrates to other severs or host to collect data. To provide the above mentioned security features the collected data is signed, encrypted and attached with the hash value of the encrypted data in each host:

$$ED_i = E_{PU0}\left(S_i \| S_{i+1} \| Sig_{PRi}(D_i) \| H(D)_i \| OD_{i-1}\right)$$

$$OD_i = H(ED_i) \| ED_i$$

$$S_i \rightarrow S_{i+1} : OD_i$$

Before each migration the mobile agent checks for the threshold value (i.e.,) data size α. The threshold value is compared with the collected data size i.e., OD_i. Based on the comparison the mobile agent either migrates or does both migration and cloning:

```
If α > size of(OD_i) then
        Migrate to next host
Else
{
Calculate α = α+ size of (OD_i) or α = α+ constant τ
 Does cloning
 Return clone agent to originator with partial data
 Migrate to next host
}                                                    (1)
```

1.26. Host Count as Threshold

The mobile agent migrates to N number of host to collect data from each host. The mobile agent can interact with the originator after β hosts for preventing itself from various attacks. Initially the HC will be 0.

After visiting each host, the host count HC is incremented by one:

$$HC = HC + 1$$

Before migrating to the next host, the mobile agent verifies its threshold value β. Based on the value of β and HC, the agent either migrates to the next host or does cloning and migration:

```
If β>HC then
Migrate to next host
Else
  {
Calculate HC = HC +1
Does cloning
Return clone agent to originator with partial data
Migrate to next host
HC = 0
}                                                    (2)
```

1.27. Execution Time as Threshold

The mobile agent takes texe time to execute its code in each host and ttra time to travel from one host to another host. Total execution time:

$$T_{tot} = \sigma \sum t_{exei} + \sum t_{rai} + \sigma$$

σ – delay time

The mobile agent may clone and return based on the threshold time γ. In each host after execution the total execution time T_{tot} is calculated based on the above formula. If total time T_{tot} exceeds the threshold value γ then the agent communicates with the originator by cloning and return the partial data:

```
If γ<T_tot then
    Migrate to next host
Else
{Calculate γ = γ + T_tot or α = α+ constant υ
Does cloning
Return clone agent to originator with partial data
Migrate to next host
}                                                    (3)
```

1.28. Experimentations

Mobile Agent is usually implemented for a distributed application of information retrieval from large number of database residing in remote servers. The data retrieved from the remote servers are securely transmitted until it reaches the originator.

Here a typical e-commerce application of e-ticketing is chosen. i.e., single client searching for information about a finding convenient price from the catalogs of several on line travel agencies. The client requires highly customized query, which is not supported by the standard query interface of on line shop. Such query would require the client to fetch a relevant subset catalog and implement a search at its end.

1.29. Aglets

The Clone Return Process is experimented using IBM Aglet. Aglet is Mobile Agent framework which supports interoperability Aglet was developed by IBM Tokyo Research Laboratory and is now open source. An Aglet is a composite Java object that includes mobility and persistence and its own thread of execution. Aglets uses a call-back model based on the Java event delegation model. Various action and mobility interfaces are supported by Aglets framework which determine what to do when a specific event happens.

An Aglet interacts with its environment through an Aglet Context object. Aglets are always executed in Aglet Contexts. To interact with each other, Aglets go through Aglet Proxy objects. An Aglet Proxy object acts as an interface of an Aglet and provides a common way of accessing the Aglet behind it. In a way, an Aglet Proxy object becomes the shield that protects an agent from malicious agents.

Agent Transfer Protocol (ATP) is a simple application-level protocol designed to transmit an agent in an agent system-independent manner. An ATP request consists of a request line, header fields and content. The request line specifies the method of the request, while the header fields contain the parameters of the request. ATP defines the following four standard requests methods:

- Dispatch: The dispatch method requests a destination agent system to reconstruct an agent from the content of a request and to start executing the agent. If the request is successful, the sender must terminate the agent and release any resources consumed by it
- Retract: The retract method requests a destination agent system to send a specified agent back to the sender. The receiver is responsible for reconstructing and resuming the agent. If the agent is successfully transferred, the receiver must terminate the agent and release any resources consumed by it
- Fetch: The fetch method is similar to the GET method in HTTP; it requests a receiver to retrieve and send any identified information (normally class files)
- Message: The message method is used to pass a message to an agent identified by an agent-id and to return a reply value in the response. Although the protocol adopts a request/reply form, it does not lay down any rules for a scheme of communication between agents

1.30. Experimental Setup

The Clone Return Process is implemented on 8 terminals of Pentium IV core 2deo, 2.67 GHZ, 1 GB RAM connected through a 10mbps LAN.

For secure migration of mobile agent, the following advanced levels of cryptographic algorithms are used:

- Elliptic Curve Cryptography
- RSA
- SHA1
- Digital Signature

The following parameters are considered for comparing the performance of the implementation strategy:

- Database size
- Size of the data retrieved
- Processing time
- Number of hosts
- Key size

In this turnaround time is taken as the performance metric. Turnaround time is the time that elapsed between posting the request and receiving the results. This time includes agent creation, migration to other servers, information retrieval and the time to process for extracting the required data. CRP is given better results with respect to Performance and Security.

1.31. Security Analysis

1.31.1. Confidentiality

1.31.2. Data Confidentiality

As the retrieved data is encrypted with the public key of the originator, only the originator can decrypt the data for processing. As only the host in which the mobile agent was created can obtain the data computed from other hosts, the protection of data from unauthorized disclosure was ensured:

$$ED_i = E_{PU0}\left(S_i \| S_{i+1} \| Sig_{PRi}(D_i) \| H(D)_i \| OD_{i-1}\right)$$

1.32. Forward Privacy

The encrypted data and hash of the data are appended with pervious collected offers and it is processed with public key cryptography. Hence the malicious host cannot discover the pervious host's address and data which implies forward privacy:

$$ED_i = E_{PU0}\left(S_i \| S_{i+1} \| Sig_{PRi}(D_i) \| H(D)_i \| OD_{i-1}\right)$$

1.33. Integrity

1.33.1. Data Integrity

The retrieved data is digitally signed and hashed. The enhanced data is appended with pervious collected offers and it is processed with public key cryptography. Thus the malicious host cannot change the pervious host's address and data:

$$ED_i = E_{PU0}\left(S_i \,\|\, S_{i+1} \| Sig_{PRi}(D_i) \| H(D)_i \| OD_{i-1}\right)$$

1.34. Code Integrity

Code Integrity is assured by verifying the received hash code with the hash code of mobile agent code in the current host. While dispatching the agent the host has to sign the hash code for next host verification:

- $RH_{code} = DPU_{i-1}(EH_{code})$
- $H_{code} = H(\text{Agent Byte Code})$
- Verify if RH_{code} equals to H_{code} or not

1.35. Authentication

Authentication is provided for data and the mobile code through the private key encryption. The host where the mobile agent migrates authenticates the data computed on its platform and the hash code of the mobile code by its private key encryption:

$$ED_i = E_{PU0}\left(S_i \,\|\, S_{i+1} \| Sig_{PRi}(D_i) \| H(D)_i \| OD_{i-1}\right)$$
$$EH_{code} = E_{PRi}\left(H_{code}\right)$$

1.36. Anti-Insertion-Attack

One host cannot access the data of another host. Also the host can neither insert nor modify the data collected from the previous host because in each host the collected data is encrypted by host's private key and the originator public key. Mainly chained hash values are generated to avoid anti insertion attack i.e., previous and current offers were put together to find hash value of current host:

$$ED_i = E_{PU0}\left(S_i \| S_{i+1} \| Sig_{PRi}(D_i) \| H(D)_i \| OD_{i-1}\right)$$
$$OD_i = ED_i \,\|\, H\left(ED_i\right)$$

1.37. Truncation Resilience

The chain of encapsulated offer could not be broken in between the colluded malicious hosts due to the cloning and return of mobile agent in between them.

1.38. Malicious Host Identification

The originator could identify the malicious host by verifying the chain of encapsulated offer:

$$ED_i = E_{PU0}\left(S_i \| S_{i+1} \| Sig_{PRi}(D_i) \| H(D)_i \| OD_{i-1}\right)$$
$$OD_i = ED_i \,\|\, H\left(ED_i\right)$$

1.39. Availability

1.39.1. Non-Repudiation

The host to which the mobile agent has moved on could not deny data results computed on them and agent's passing through due to digital signature on the data and the hash value of the mobile agent code.

2. CONCLUSION

Mobile agents are very much important in to today's e-world. The protection of mobile agent data plays a major role in mobile agent applications. The mobile agent security is guaranteed through Clone Return Process (CRP). Multi colluded truncation attack is avoided by partial returning of data in between processing. The execution time to collect all data is reduced due to CRP.

3. REFERENCES

Benachenhou, L. and S. Pierre, 2006. Protection of a mobile agent with a reference clone. Comput. Commun., 29: 268-278. DOI: 10.1016/j.comcom.2005.01.006

Cheng, J. and V. Wei, 2002. Defenses against the truncation of computation results of free-roaming agents. Proceedings of the 4th International Conference on Information and Communications Security, Dec. 9-12, Springer Berlin Heidelberg, Singapore, pp: 1-12. DOI: 10.1007/3-540-36159-6_1

Chung, Y.F., T.S. Chen and M.W. Lai, 2008. Efficient migration access control for mobile agents. Comput. Standards Interf., 31: 1061-1068. DOI: 10.1016/j.csi.2008.09.039

Dadhich, P., K. Dutta and M.C. Govil, 2010. Security issues in mobile agents. Int. J. Comput. Appli., 11: 0975-8887. DOI: 10.5120/1574-2104

Garrigues, C., S. Robles, J. Borrell and G. Navarro-Arribas, 2010. Promoting the development of secure mobile agent applications. J. Syst. Soft., 83: 959-971. DOI: 10.1016/j.jss.2009.11.001

Hacini, S., Z. Guessoum and Z. Boufaida, 2007. TAMAP: A new trust-based approach for mobile agent protection. Comput. Viro, 1: 267-283. DOI: 10.1007/s11416-007-0056-y

Jha, R. and S. Iyer, 2010. Performance evaluation of mobile agents for e-commerce applications. Proceding of the 8th International Conference on High Performance Computing, Dec. 17-20, Springer Berlin Heidelberg, India, pp: 331-340. DOI: 10.1007/3-540-45307-5_29

Karjoth, G., N. Asokan and C. Gulcu, 1998. Protecting the computation results of free roaming agents. Proceeding of the 2th Workshop Conference On Mobile Agents Mobile Agents, (MA' 98), Springer-Verlag Londen UK, pp: 195-207.

Linna, F. and L. Jun, 2010. A free-roaming mobile agent security protocol against colluded truncation attack. Proceedings of the 2nd International Conference on Education Technology and Computer, Jun. 22-24, IEEE Xplore Press, Shanghai, pp: 261-265. DOI: 10.1109/ICETC.2010.5530034

Marikkannu, P., R. Murugesan and T. Purusothaman, 2011. AFDB security protocol against colluded truncation attack in free roaming mobile agent environment. Proceedings of IEEE International Conference on Recent Trends in Information Technology, Jun. 3-5, IEEE Xplore Press, Chennai, pp: 240-244. DOI: 10.1109/ICRTIT.2011.5972321

Ogunnusi, O.S. and S.A.D Razak, 2013. Research on security protocol for collaborating mobile agents in network intrusion detection systems. Am. J. Applied Sci.,10:1638-1647. DOI: 10.3844/ajassp.2013.1638.1647

Park, J.Y., D.I. Lee and H.H. Lee, 2001. Data protection in mobile agents; one-time key based approach. Proceedings of the 5th International Symposium Autonomous Decentralized Systems, Mar. 26-28, IEEE Xplore Press, Dallas, pp: 411-418. 10.1109/ISADS.2001.917446

Raji, F. and B.T. Ladani, 2010. Anonymity and security for autonomous mobile agents. IET Inform. Security, 4: 397410. DOI: 10.1049/iet-ifs.2009.0217

Roth, V., 2001. On the robustness of some cryptographic protocols for mobile agent protection. Proceedings of the 5th International Conference on Mobile Agents (MA' 01), Springer-Verlag, pp: 1-14. DOI: 10.1007/3-540-45647-3_1

Senthilnathan, T. and T. Purusothaman, 2012. A multi-agent based data replication mechanism for mobile grid. Am. J. Applied Sci., 9: 542-552. DOI: 10.3844/ajassp.2012.542.552

Silei, L., Z. Rui, L. Jun and X. Junmo, 2008. A novel security protocol to protect mobile agent against colluded truncation attack by cooperation. Proceedings of the International Conference on Cyberworlds, Sept. 22-24, pp: 186-191. IEEE Xplore Press, Hangzhou, DOI: 10.1109/CW.2008.28

Songsiri, S., 2005. A new approach for computation result protection in the mobile agent paradigm. Proceedings of the 10th IEEE Symposium on Computers and Communications, Jun. 27-30, IEEE Xplore Press, pp: 575-581. DOi: 10.1109/ISCC.2005.14

Srivastava, A.K and A. Goel, 2010. Secure mobile agent based information gathering in wireless network. Int. J. Comput. Sci. Eng., 2: 1685-1689.

Venkatesan, S. and C. Chellappan, 2008. Protection of mobile agent platform through Attack Identification Scanner (AIS) by Malicious Identification Police (MIP). Proceedings of the International Conference in Emerging Trends in Engineering and Technology, July 16-18, IEEE Xplore Press, Nagpur, Maharashtra, pp: 1228-1231. DOI: 10.1109/ICETET.2008.89

Venkatesan, S., C. Chellappan and P. Dhavachelvan, 2010a. Performance analysis of mobile agent failure recover in e-service applications. Int. J. Comput. Standard Interf., 32: 38-43. DOI: 10.1016/j.csi.2009.06.001

Venkatesan, S., C. Chellappan, T.V. Engattaraman, P. Dhavachelvan and A. Vaish, 2010b. Advanced mobile agent security models for code integrity and malicious availability check. J. Netw. Comput. Appli., 33: 661-671. DOI: 10.1016/j.jnca.2010.03.010

Wang, X., M. Chen, T. Kwon and H.C. Chao, 2011. Multiple mobile agents itinerary planning in wireless sensor networks: Survey and evaluation. IET Communi., 5: 1769-1776. DOI: 10.1049/iet-com.2010.0638

Yee, B.S., 1997. A sanctuary for mobile agents. Technical Report CS97-537, UC San Diego, Department of Computer Science and Engineering.