Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 5, Issue 1, pp: (5-15), Month: January - April 2018, Available at: www.noveltyjournals.com

Design Tools for Mobile Agent Security from Malicious Host Platforms

Omoghenemuko, Greg I.[1], Asagba, Prince O.[2], Ogheneovo, Eward E.[3]

[1]Department of Computer Science, College of Education, Warri, Delta State, Nigeria
[2.3] Department of Computer Science, University of Port Harcourt, Rivers State Nigeria

Abstract: **Mobile agent security is imperative driving force for recent researches on heterogeneous distributed systems. The mobility of code awakens major security problems. This article introduces the taxonomy of security threats' to mobile agents' from the malicious hosts domain and takes cursory look at five fundamental limitations of mobile code paradigmatics: integrity; availability; confidentiality; authentication; and non-repudiation. The authors conceptualize and presented a proposed secure enhanced mobile agent security architecture that can protect mobile agents from malicious hosts' attacks in heterogeneous computation environment. Unified Modeling Language (UML) diagrams are employed to explain the proposed security architecture in. The authors presented thirteen typology of UML diagrams but employ five of the UML diagrams in the proposed system: one-jump and two- jump sequence use case diagrams, which shed light on sequence of security processes followed in security design of architectural mechanism for mobile agents' security from malicious hosts' attacks; class diagram, which shows different objects and classes in our security architectural design; state diagram that shows chronological events flow in the security system design; an event diagram that shows mobile agent source code obfuscation to maintain code integrity and mobile security system's input-process-output chart, and a detailed algorithm of the entire system's security is also presented. The authors conclude that state diagram is suitable for design of security elements in mobile agents to protect mobile agents from malicious hosts' attacks.**

Keywords: **Design tool; model; UML, taxonomy; typology, malicious hosts; integrity and confidentiality.**

1. BACKGROUND

The nucleus of object-oriented problems' solving, is design of model, and the design of model requires certain tools for its achievement. A model is a map through which real world problems' are solved, and modeling is art of constructing model and using the model for problems' solving. A model composes of objects, which interact by transmitting messages. Objects know things (*attributes*) and they do things (*operations*). An object *state* is a function of object's attributes and *classes* are a map of objects. A class envelops (data) and operations' (methods or functions) in single unique entity called object, and objects are *instances'* of classes.

Architect designs buildings. Builders create buildings' from architectural designed model made from design tools. Blueprint is standard graphic language engineers must understand to do their work. Developing software is unlike building construction, and the more complex the software system, the more crucial the link among people and all components involved in designing and extending applications.

Unified Modeling Language (UML) was introduced as applications' blueprint language for systems' analysts and software designers' to simplify software development tasks. Presently, UML is major part of applications' design tool. UML is a tool for visualizing a software program using a collection of diagrams. "The annotation has metamorphosed in Grady Booch, James Rumbaugh, Ivar Jacobson's work and effort of Rational Software Corporation for object-oriental design, but

it has deployed to traverse wider variety of software-design projects. Today, UML is accepted by the Object Management Group (OMG) as the standard for modeling software development" (Donald, 2003). UML specification to cover wider scope of software creation efforts in addition to agile practices. It improves integration between structural models like class diagrams' and behavior models like activity diagrams, adds ability to define a hierarchy and decompose software system into components and sub-components.

The UML gives everyone: business analyst; software developers a common lexicon to talk on software design. So, Object Management Group (OMG) established UML; to provide advanced community with permanent and universal design language to create computer applications'. One of the rationales of UML was to establish unified standard modeling notation that IT professionals can read and transmit systems' structure and design plans just as construction engineers have been doing for years with buildings' blueprints.

The UML is for object-oriental problem solving through programming, and object-orientation concept begins with construction of model, yet many modeling tools are enveloped in UML. UML is a standardized modeling language because it visualizes the modeling of software and it is programming (or implementation) language independent. The most useful UML diagrams' are: use-case diagrams, class diagrams, sequence diagrams, event diagrams, state diagrams, component diagrams, and deployment diagrams. It is however, outside scope of our research to dwell on details on types of UML diagrams. We will provide our readers with definitive information for generic comprehension of UML diagrams, as our interest is on employing UML diagrams as software design tools to show how mobile agents' could be prevented from malicious hosts' attacks in a multifarious computing environment.

2. MOBILE AGENTS AND MOBILE AGENTS' ENVIRONMENTS

A mobile agent model has sub-models, which include: an agent model, a life-cycle model, computational model, security model, configuration model and finally, navigation model

Taxonomy of Threats to Mobile Agent Paradigmatics:

Jansen (2000) identified four categories of threats to the general acceptability of mobile computing paradigmatics. They include:

- Threats imposed by mobile agent to the host;
- Threats imposed by mobile agent's Host to Mobile agent;
- Threats imposed by mobile agent to mobile agents; and
- Threats imposed by other entities to mobile agents

In providing a secure architecture for mobile agents, threats stemming from attacks imposed by a malicious host on mobile agent, is of main concern in this paper and these threats from malicious hosts' platforms are resultantly examined.

In designing enhanced mobile agent security architecture, we organise the possible threats from malicious hosts on mobile agents using different criteria depending on the attack. According to International Security Organisation-ISO (1988) cited in Elmarie and Elsabe (2002), the criteria by which a mobile agent is protected against a malicious host, depends on five basic requirements' of users of Internet services (namely:

- *Integrity*,
- *availability*,
- *Confidentiality*,
- *Authentication;* and
- *Non-repudiation.*

By using these basic requirements' for security, the criteria that should exist in design of mobile systems, are: integrity, confidentiality and authentication.

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 5, Issue 1, pp: (5-15), Month: January - April 2018, Available at: www.noveltyjournals.com

Integrity:

Integrity of mobile agents should be defended from manipulations by malicious hosts. This includes protection from manipulations of mobile agent's code, state, and data (Elmarie, 2004). In order to protect mobile agents', integrity sub-categories must be added in the security design:

Integrity interference: mobile agent should avoid executing hosts that tends to interfere with mobile agent's execution mission. In this scenario the hosts do tamper any information, but interferes with the execution of mobile agent completely, transmitting the agent to hosts unspecified on itinerary, or executing agents' precipitately.

Information modification: The sub-criteria include several actions', namely: alteration, corrupting, manipulating, deleting, wrong execution of agent's code, data, control flow or status. Second instance of information modification occurs when executing hosts interfere with interaction between agents and alters the communications' for its own well-being.

Availability:

When mobile agent reaches host, it receives privileges and access to resources' required for its design objectives. If authorised mobile agent is denied access to objects or resources it should have legal access, then availability-refusal occurs. *Availability refusal* is deliberate action perpetrated by executing platforms, to obstruct agent. This attack takes various forms' namely: *denial-of-service*, *delay-of-service* and *transmission refusal*

Denial-of-Service (DoS): Normally, this attack occurs where network crashes because it is overloaded with network traffic. Mobile agent's denial-of-service, means the requested resources' needed by agent to finish its mission are denied. It is practicable for malicious host to attack agent with much useless information that agent finds it difficult to finish its tasks and achieve its objectives. Attacks relating to non-reputation, where the agent platform denies that it has received an agent is also included here.

Delay-of-Service: This attack occurs where host allows mobile agent to long for service and only provides that service or access to required resources after some time. This delay can have effect on the actual purpose of mobile agent.

Transmission refusal: when malicious host disregards itinerary of mobile agent and refuses' to transmit agent to the next host stated in its itinerary, transmission refusal occurs.

Confidentiality:

When the assets of mobile agents' are illegitimately accessed by host, privacy of mobile agent is not respected and agent comes under attack. *Three subclasses of confidentiality attac*ks are described, namely eavesdropping, theft, and reserve engineering.

Eavesdropping is invasion of privacy that occurs when host spies on agent and collects information about mobile agent or about intercommunications between agents'. The access by remote host to mobile agent's code, state and data, give host a chance to monitor agents' for other purposes than protecting itself and its own resources. Although the host may not attempt to alter the agent, it can use this information for its own benefits (Elmarie, 2004; Parul and Sharma, 2012).

The classical threat of eavesdropping when electronically communicating, is more serious in mobile agent systems because an agent platform can, not only monitor communications, but attack all unencrypted code executed agent brings to host's platform and data gathered on the platform. An agent may be exposing proprietary algorithms, trade secrets or other sensitive information (Jansen and Karygiannis, 2000). Even if the platform finds it clumsy to automatically extract the secret information, it infers the meaning from the types of services requested. As with agent, which interact with travelling agents. Though the exact details of communications', is unknown to the platform, the communication could show that the person on whose behalf the agent is acts, is planning an itinerary in a future. The platform may sell this information to a suitcase manufacturer who may sends unrequested advertisements, or for worse, the platform may share this information with thieves who may target the home of the traveler (Jansen and Karygiannis, 2000).

Identity stealing or theft and eavesdroppings are related. In this subclass, the malicious hosts spy on agent, and removes information from agent. The malicious host may also steal the agent itself and use it for its own purposes, or simply kill it (Elmarie, 2004) and (Parul and Sharma, 2012).

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 5, Issue 1, pp: (5-15), Month: January - April 2018, Available at: www.noveltyjournals.com

Reverse engineering occurs when the malicious host captures mobile agent and analyzes its data and state in order of manipulates future or existing agents. While "reverse engineering attacks' makes host to design its own corresponding agents or update the profile of information access by agent

Authentication:

In the case of the malicious host problem, the agent correctly determines and authenticates its executing host. If agent hides identity or refuses to present its credentials, the host may hamper or jeopardize the intended objective(s) of mobile agent. *There are two subdivisions of authentication attacks*, namely *masquerading* and *cloning* (Elmarie, 2004).

Masquerading: if remote host hides as destination on mobile agent's itinerary, then, masquerading occurs. A remote host may masquerade as third party that is trusted and accept mobile agents'; to withdraw information from agents. Remote host that masquerades can harm both visiting mobile agents' and host whose identity it has used (Elmarie and Elsabe, 2002). Masquerading agent may pose as authorised agent in an effort to gain access to services and resources to which it is not entitled. Agent may masquerade as an unauthorised agent to shift the blame for any actions for which it does not want to be held liable (Parul and Sharma, 2012).

Cloning: each agent carries its own credential to gain authorised permission to service of its executing hosts. If a host duplicate of mobile agent, it causes specific agent authentication problems.

3. CHALLENGES OF EXISTING SECURITY SYSTEM

The challenges of existing system include:

1. Insufficient security mechanism to secure computation result data;
2. Mobile code tampering is possible but can be detected;
3. Inadequate migration facility (weak mobility) since mobile agent is only been transmitted on trusted execution environments.

Protecting Mobile Agent from Malicious Hosts:

Protecting mobile agent in a multifarious network environment to promote mobile agent computing is a serious issue for investigation in scientific computations. Although, it is somewhat clumsy to design a foolproof model or architecture to protect mobile code, we are poised to design an architecture that is applicable to develop security tactics that comfortably protect mobile agents from hackers' attacks. In this section, we use the UML design tools, to present captivating logical and physical secure enhanced mobile agent security architecture design structure for mobile agent protection from malicious hosts' on Internet.

Typology of UML Diagrams:

The UML diagrams group into two unique groups: structural diagrams' and interaction or behavioural diagrams.

Structural UML Diagrams

1. Class diagrams
2. Package diagram
3. Object diagram
4. Component diagram
5. Composite diagram
6. Deployment diagram

Behavioural UML Diagrams

7. Activity or Event diagram

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 5, Issue 1, pp: (5-15), Month: January - April 2018, Available at: www.noveltyjournals.com

8. Sequence diagrams
9. Use-case diagrams
10. State diagram (state's machine diagrams or statechart)
11. Communication diagram
12. Interaction diagram
13. Time diagram

Class Diagram- is central to object-oriental method, involving UML, because it describes static architecture of a system.

Object Diagram- It depicts instances not classes. It is useful for explaining decomposed sections of complex relationships, especially repetitive relationships.

State Diagram- it depicts feasible state of object and the transitions' , which cause change in object's state.

Package Diagram- This is part of class diagrams, which arranges elements of system into interrelated groups to reduce dependencies among packages.

Use-Case Diagram- It enlightens analysts through life scenarios, to visualize functional requirements' of a system and shows sequences of actions, which provide measurable value to actor and it is represented with horizontal elliptic. It explains at glance what a system does to observers. Use cases emphasises *what* a system would do and not *how*.

Sequence Diagram (or Collaboration diagram)- it depicts interaction of each object with other objects of a system and how operations actualize.

Activity or Event Diagram: shows the dependability of one activity on the others and the process for an event.

State diagram- focuses on object under-going a process (or a process as object),

Component Diagram-component is module of a code and physical representations' of class diagram.

Deployment Diagram- depicts physical arrangements of software and hardware. They show relationships existing among systems' hardwares and softwares.

Time diagram- shows time for a process to execute

Proposed Secure Enhanced Mobile Agent Security Architecture

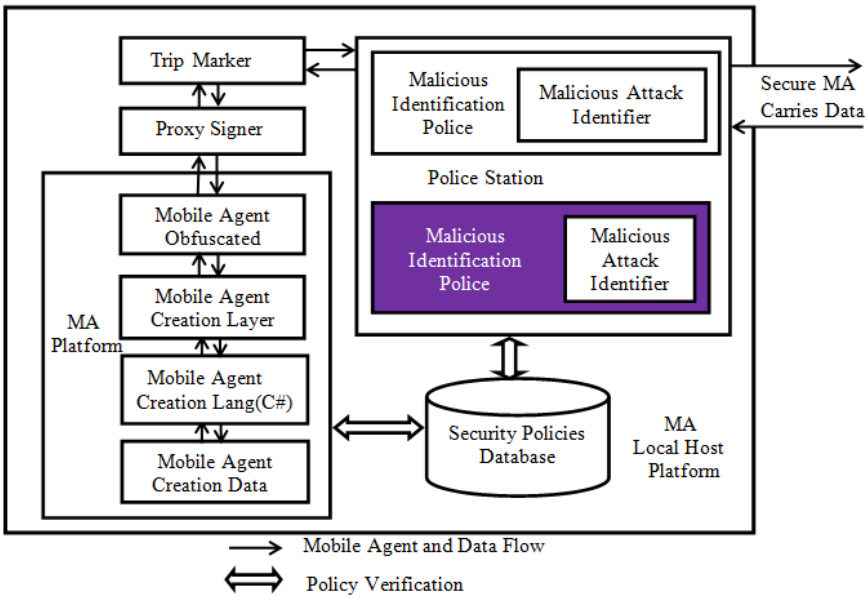

Figure 1: Proposed secure enhanced mobile agent security architecture

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 5, Issue 1, pp: (5-15), Month: January - April 2018, Available at: www.noveltyjournals.com

4. DESIGN

The design of another system is inevitable due to the identified challenges exhibited by existing system. In design, we specify architecture, components, modules, interfaces and data that satisfy system requirements. Criteria to establish protective scheme of mobile agents against malicious hosts, are based on fundamental concerns of users of Internet, namely: integrity, availability, confidentiality, authentication (Elmarie, 2004) and authorisation. The challenges identified in existing system are used as foundation for requirements for integrated mobile agent security architecture. The requirements for integrated mobile agent security architecture are proposed as following:

1. Architecture supplies security, dependent on type of environments of implementation.

2. Architecture supplies security levels defined by mobile agent type and goals.

3. Architecture maintains mobility and autonomy attributes of mobile agents.

4. Additional software and hardware requirements for implementations of agent and agents' host platforms must be at minimum, to decrease (time and cost) of computing.

5. Communication sessions between mobile agent and hosts, and other agents must be minimised,

6. Cost of implementation of strategies for securing and maintaining agents must be minimalised.

7. Agent implementation cost must be affordable. That is, costs of countermeasures implementation are a function of security.

8. Fundamental mechanisms that help security of mobile agent like integration of additional function of security services are adopted.

Design Tools Employed:

Immediately the researchers determine the objects' behaviours and responsibilities, the researchers can create a detailed model of how the object will interact with each other to provide the functionality specified in each design use case. The Unified Modeling Language (UML) provides thirteen types of diagrams to model design and implementation aspects of the system. In this research design for mobile agent security against malicious host, we employ the following UML diagrams:

(1.) Sequence Use Case Diagram;

(2.) Object-Class Diagram;

(3.) Activity or Event Diagram;

(4.) Input-Process-Output Charts; and

(5.) State Diagrams

Sequence Use Case Diagram:

Sequence use case diagrams show us great detail of how the objects interact with each other.

Protection of mobile agent conveying computation results to local host through the Internet requires the sequence in figures 2 and 3

(1) Mobile agent visits *Remote Host A*, to execute.

(2) Secure computations' are completed at untrustworthy host and results are conveyed to mobile agent.

(3) Results obtained at *Remote Host A* are taken to local host.

(4) But if mobile agent must visit more hosts to do computation or gather information before returning home, then

(5) Mobile agent would embark on itinerary to *Remote Host B.*

(6) Secure computations are completed.

(7) Mobile agent embarks on itinerary to *Remote Host C*, and secure computations' are completed

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 5, Issue 1, pp: (5-15), Month: January - April 2018, Available at: www.noveltyjournals.com

(8) Mobile agent embarks on itinerary to *Remote Host N*, and secure computations' are completed

(9) Results obtained at *Remote Hosts A, B, C* and N are conveyed to local host safely.

Figure 2 shows One-Hop Sequence use case diagram for proposed secure enhanced security system. Figure 3 shows Multi-hop sequence use case diagram for proposed secure enhanced security system.

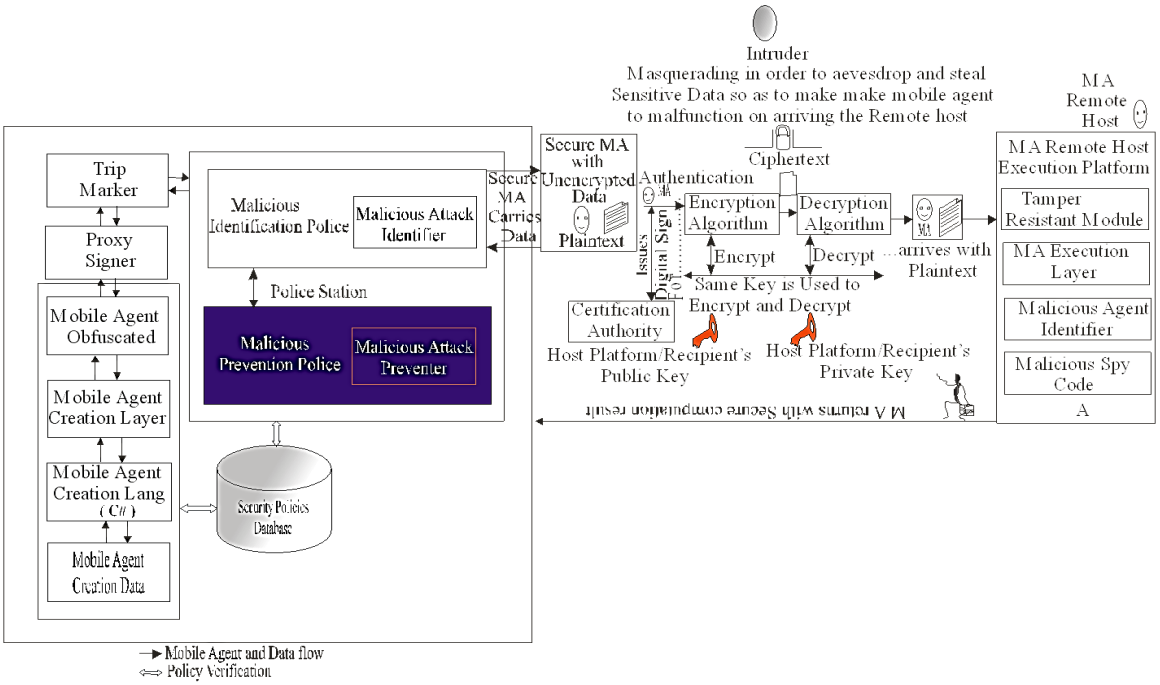

Figure 2 shows One-Hop Sequence use case diagram for proposed secure enhanced security system

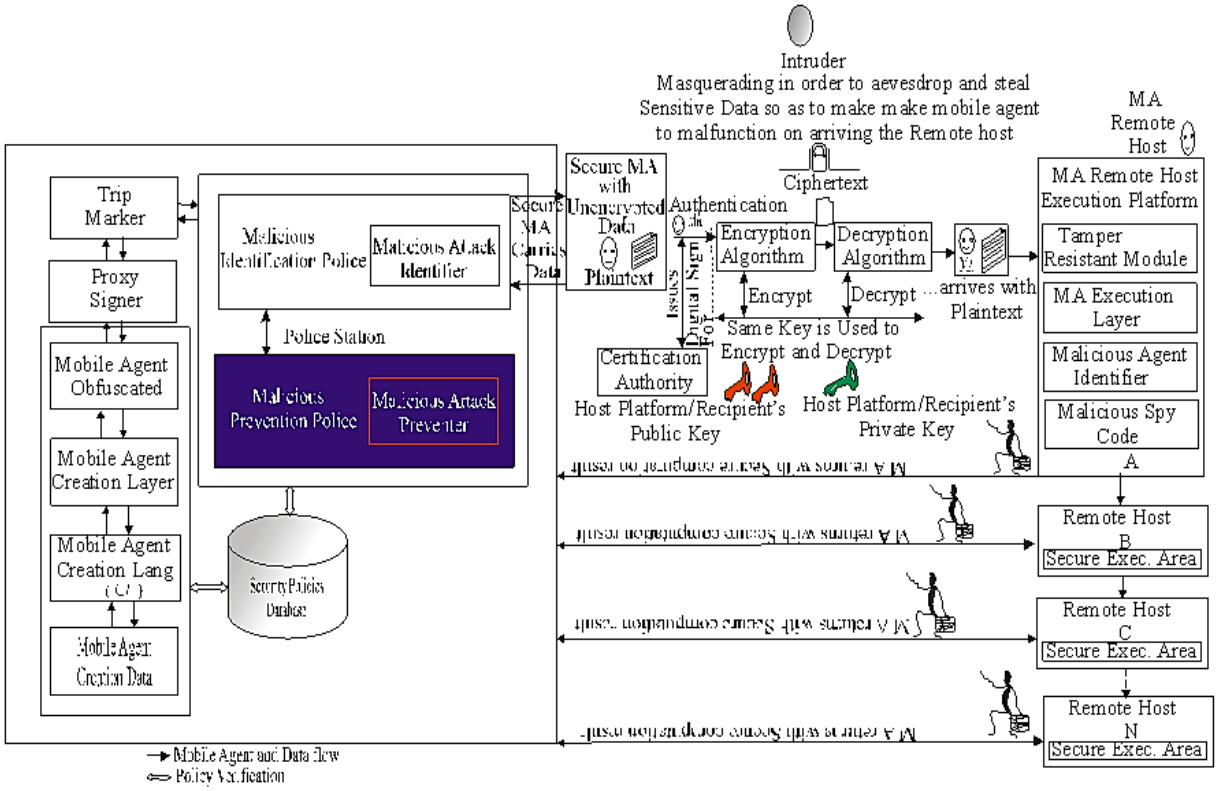

Figure 3: Multi-hop sequence use case diagram for proposed secure enhanced security system

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 5, Issue 1, pp: (5-15), Month: January - April 2018, Available at: www.noveltyjournals.com

5. CLASS DIAGRAM OF PROPOSED MOBILE AGENT SYSTEM

Class diagram is used to extend the use-case. Object-class diagram specifies objects, attributes, technique of operations and object-class relationship. Conventionally, class diagram is depicted with rectangle containing three horizontal sections. The Upper section shows the name of the object class, middle part contains the attribute and the lower part contains the type of operation on object. Figure 4 shows Class diagram of proposed secure enhanced mobile agent security system, and Figure 5 shows proposed secure enhanced mobile agent security system state diagram

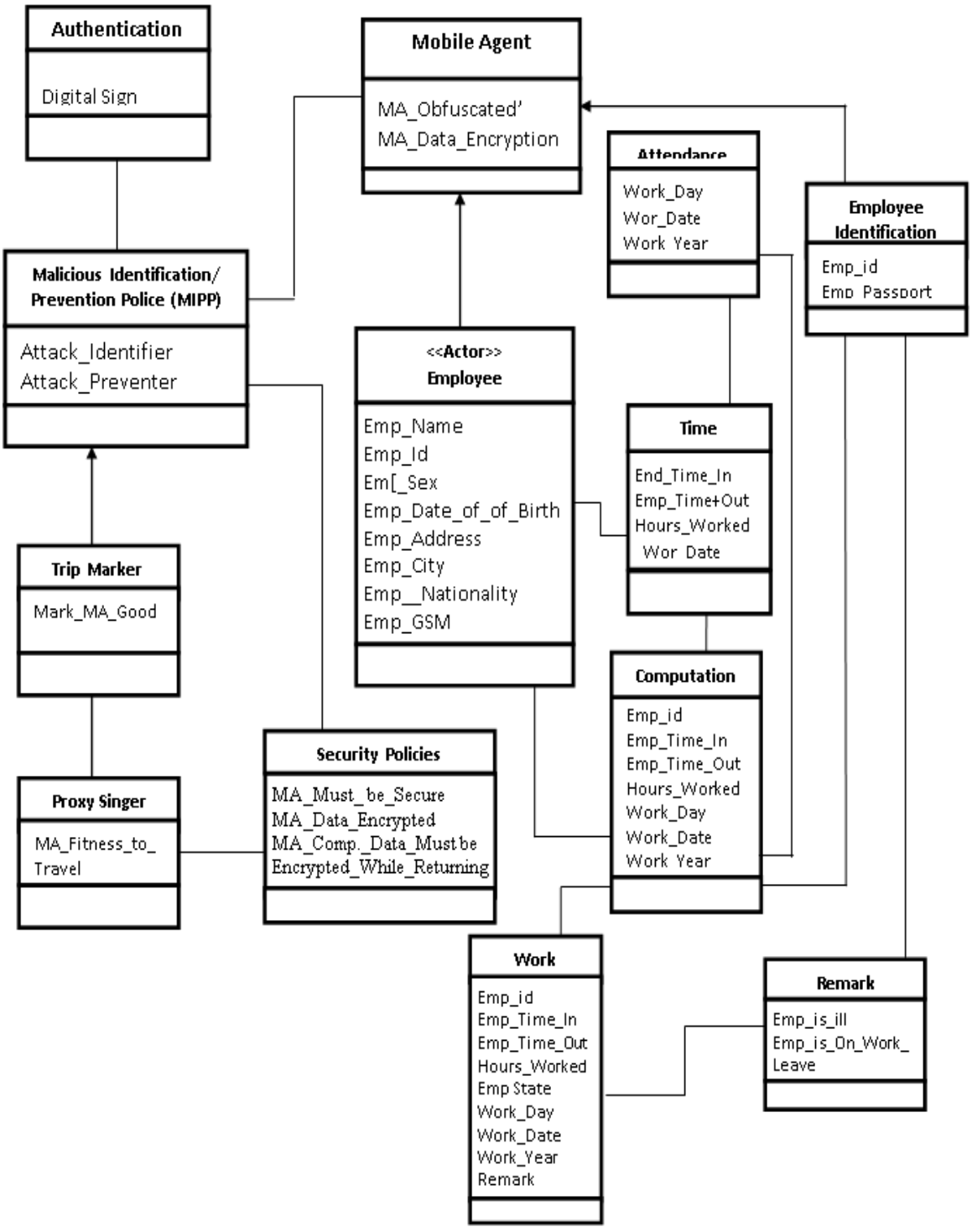

Figure 4: Proposed secure enhanced security system class diagram

Novelty Journals

ISSN 2394-7314

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 5, Issue 1, pp: (5-15), Month: January - April 2018, Available at: www.noveltyjournals.com

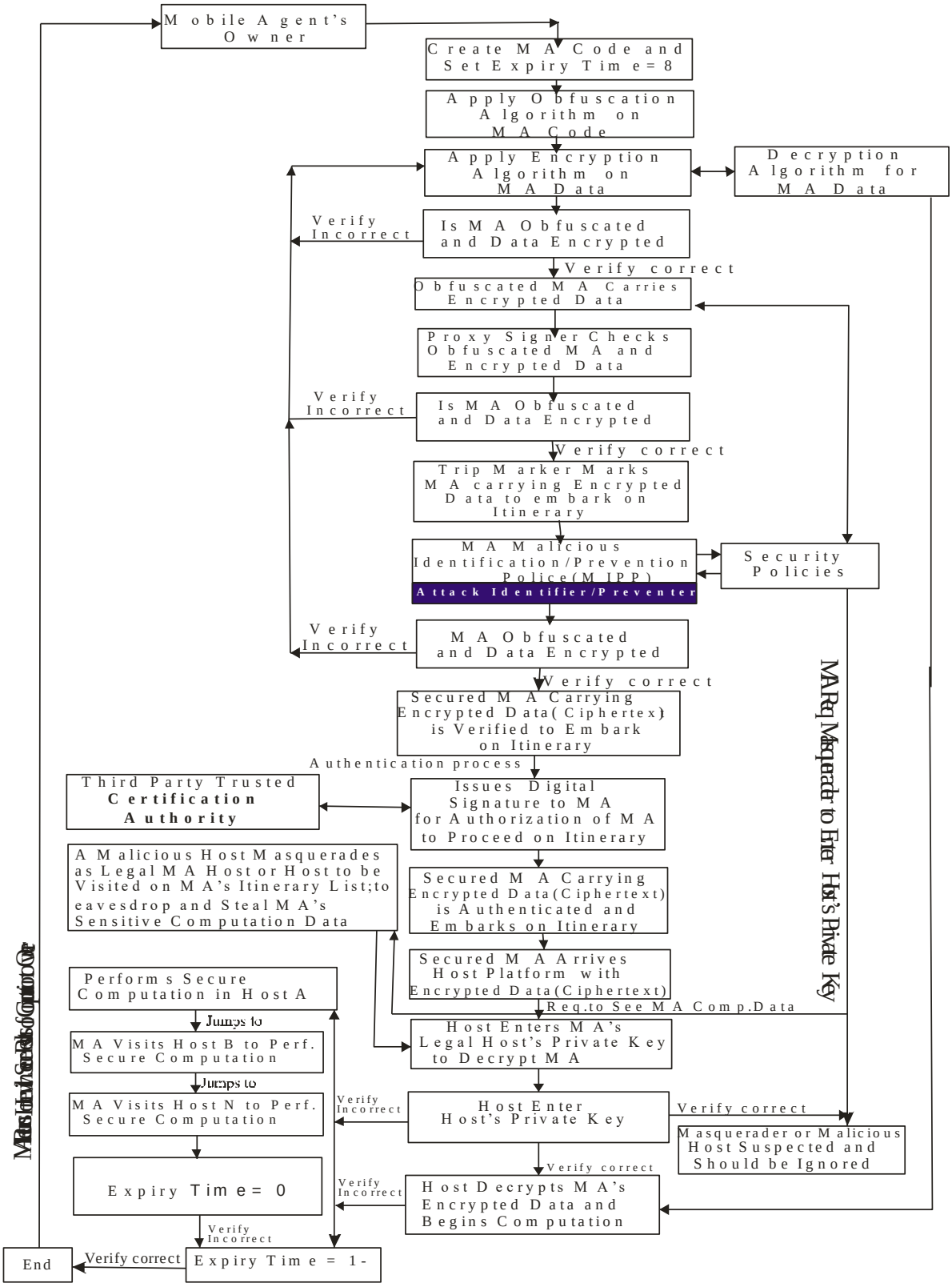

Fig. 5: Proposed secure enhanced mobile agent security system state diagram

Input-Process-Output Design:

Input-Process-Output (IPO) model uses charts to specify programs' inputs, storage, processing stages needed to process data and generate output. Figure 6 input-process-output chart of the proposed mobile agent security system

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 5, Issue 1, pp: (5-15), Month: January - April 2018, Available at: www.noveltyjournals.com

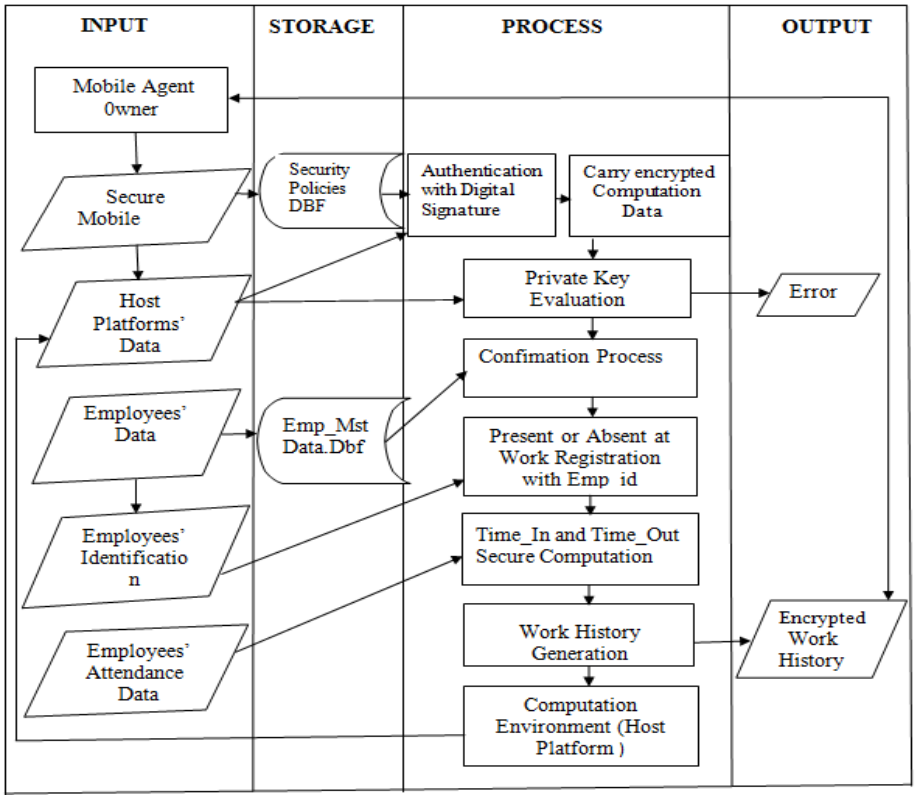

Figure 6: Mobile agent security system's input-process-output chart

Proposed Mobile Agent Code Obfuscation Algorithm Event Diagram:

Figure 6 shows the proposed mobile agent code obfuscation algorithm event diagram

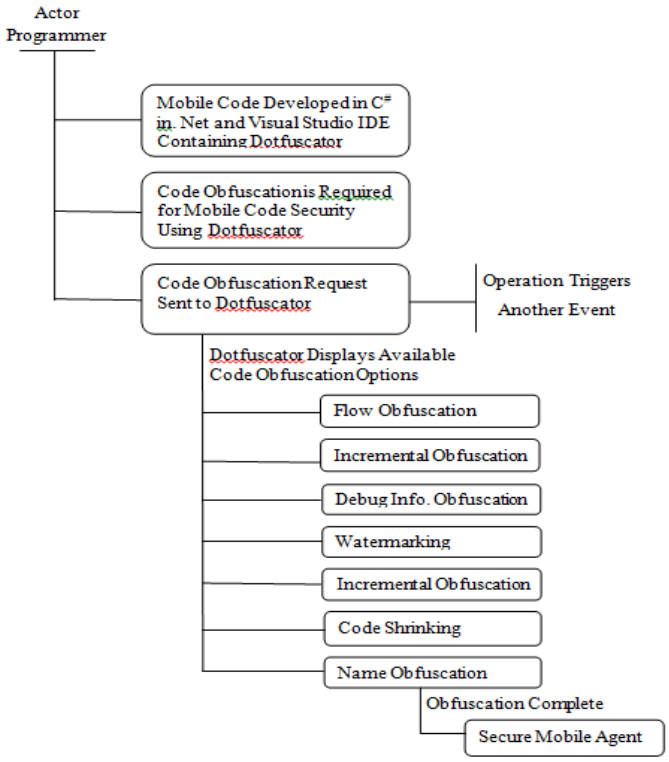

Figure 6: shows the proposed mobile agent code obfuscation algorithm event diagram

International Journal of Novel Research in Computer Science and Software Engineering
Vol. 5, Issue 1, pp: (5-15), Month: January - April 2018, Available at: www.noveltyjournals.com

Preferred UML Diagram in our study and Why?

The state diagram in figure 4., is preferred as a design tool for creating secure mobile agent that can successfully overcome malicious hosts' attacks in a heterogeneous computation environment. The reason is, state diagram describes dynamic behaviour of a system in response to external stimuli. State diagrams are especially useful in modeling reactive objects whose states are triggered by specific events-which makes it suitable for secure mobile agent software design against malicious hosts' attacks.

6. CONCLUSION

Design is salient phase of software creation, and the tools employed for specific design to resolve specified problem, help in reducing software failure to solve the given problem, and thus create blue-prints for software solutions that satisfy user's requirements'. This study examines at mobile agent security from malicious host platforms, and employed the following design tools: sequence diagram use case; object-class diagram; activity or event diagram; input-process-output charts; and state diagrams to ensure foolproof security of mobile agent from malicious hosts in heterogeneous computation environment.

REFERENCES

[1] Donald, B. (2003): An introduction to the Unified Modeling Language. https://www.ibm.com/developerworks/ rational/library/769.html

[2] Elmarie, B.(2004): Framework for Protecting Mobile Agent Against Malicious Host. PhD Thesis submitted to Department of Computer Science, University of South Africa

[3] Elmarie, B. And Elsabe, C.(2002): Classification of Malicious Hosts Threats in Mobile Agent Computing. Proceedings' of Annual Research Conference' of South African Institute of Computer Scientists and Information Technologists on Enablement Through Technology.[Retrieved From: http://dl.acm.org/citation.cfm?id=581526&dl= ACM&coll=DL&CFID=580514408&CFTOKEN=95723381]

[4] Marikkannu, P.; Adri, J.J.; and Purusothaman, T. (2011): An Enhanced Mobile Agent Security Protocol. European Journal of Scientific Research, ISSN:1450-216X,Volume 51,No.3,pp.321-331. [Retrieved From: http://www. microsoft.com/mspress /books/3873.aspx] <Accessed 2013>

[5] Parul, A. and Sharma, V.(2012), *A Review on Mobile Agent Security*. International Journal of Recent Technology and Engineering (IJRTE). ISSN:2277-3878,Volume1, Issue 2, June 2012.

[6] Randy, M. (1994): A Hand-On Introduction for Developers. https://edn.embarcadero.com/article/31863