TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 15, No. 2, August 2015, pp. 381 ~ 389
DOI: 10.11591/telkomnika.v15i2.8375

Security for Mobile Agents: Trust Estimate for Platforms

Razouki Hassan*, Hair Abdellatif
Laboratory of Modeling and Computation (LMC), Faculty of Science and Technology,
University Sultan Moulay Slimane, Beni Mellal, Morocco
*Corresponding author, e-mail: razouki.hassan@gmail.com

Abstract

The mobile agent has been seen as a promising distributed computing technology. The mobility characteristic of mobile agent makes it to travel often in open network. In this scenario, it is obvious that the mobile agents are vulnerable to various security threats. Protecting free-roaming mobile agents from malicious host and from other mobile agents has drawn much attention in recent years. The protection of mobile agents is considered as one of the greatest challenges of security, because the platform of execution has access to all the components of the mobile agent. In this paper, we present a new architecture paradigm of mobile agents, which allows the separation of the implementation tasks of the agent and its security mechanisms. Our approach is based on using two strategies of adaptation to adapt the mobile agent security at runtime, depending on the sensitivity of the services required to perform the duties of the agent and the degree of confidence of the visited platforms.

Keywords: *mobile agent, multi-agent systems, security, cryptography, software components, dynamic adaptation.*

1. Introduction

The mobile agent, as a typical distributed computing technology, has attracted many researchers' attention and developed quickly to satisfy many distributed applications. The ability of mobile agents to autonomously migrate from host to host transferring their code and internal state enables them to accomplish tasks in distributed environment more robustly and efficiently than traditional approaches. Despite of significant benefits, security presents a crucial point in mobile agent systems and may hinder the expansion and use of this paradigm [1-2].

The interaction of mobile agent with its platform will bring some security problems [3]. Four threat categories are identified:

1) Agent attacks against an execution platform.
2) Platform attacks against a mobile agent.
3) Agent attacks against another agent in the same execution platform.
4) Exterior entity attacks against an agent or a platform.

These attacks are primarily focused on the communication capability of the platform to exploit potential vulnerabilities [4].

The security of mobile agent has become an serious problem that needs deep consideration [5], because the platform of execution has access to all the components of the mobile agent. Since, a platform has the responsibility to execute a mobile agent; it is assumed that the platform must have full access to agent's code and data. A platform might be malicious and may try to execute the code in a manner in which it is not authorized to do. A platform may try to change agent's state, code or routing during agents' execution [6].

The Proposed approach is to find a new mobile agent paradigm architecture, which can protect the mobile agent via two strategies of adaptation. It takes into account the dynamic aspects of the security needs of a mobile agent in each runtime environment.

The first is a static adaptation performed by the MSAS (Management System of Agents Security)based on the sensitivity of the services requested by the agent, theMSAS adds additional security components and determines the policy of the dynamic adaptation to be followed by the mobile agent during its execution.

Received April 17, 2015; Revised June 26, 2015; Accepted July 13, 2015

The second is a reflexive dynamic structural adaptation performed by the mobile agent itself. According to the degree of confidence on the platform visited, the mobile agent selects and adapts security components to the tasks to be performed by this platform.

This paper is organized as follows: in section 2 we begin by identifying the state of art and present the different approaches to protect mobile agents. Section 3 presents the new architecture of our system, identifies the functions of the various components and trust estimation. Finally a conclusion is presented in section 4.

2. The Security of Mobile Agents

2.1. Security Issues Related to Mobile Agents

Three types of problems arise with regard to security in the concept of mobile agent: the security of the agent migration, the protection of the platform against agents and malicious platforms, and the protection of the agent against other agents and malicious platforms [7].

2.1.1. Malicious Agent

It is a program or software that functions for an intruder with a purpose to attacking the itinerary host. It can pose a severe risk to the hosts serving a platform to the mobile agent. Hence these hosts are vulnerable to attack by such agents executing on them. There are several kinds of such agents and these can be virus, worms or spying agents. The protection of the visited hosts against attacks carried by malicious mobile agents is a problem that is now fairly well controlled.

2.1.2. Malicious Host

When a mobile agent moves from one designated host to another designated host in the network to complete its task, there exists some designated hosts who tries to hamper the integrity, functionality and confidentiality of the agent in order to benefit themselves in some way. For example they can change the agent code with an intention to get some task done on its behalf as agent moves to other hosts to harm the reputation of the agent owner or to harm the other hosts in some way. Also a malicious host can try to access some unauthorized data belongs to an agent thus attacks on its confidentiality. It is more difficult to protect a mobile agent from a malicious host rather than protecting a host from a malicious mobile agent.

2.2. Approaches to the Protection of the Mobile Agents

Several approaches for the protection of mobile agent have been proposed. They try to ensure the access of the mobile agent to hosts in which it may have confidence or detect those that are malicious. They are primarily intended to detect attacks or render themineffective, our proposal is based on protection.

Xinwen Zhang [8] presents a mobile policy framework to protect the information and resources imported by mobile code and agents in runtime environments with trusted computing technologies. This framework includes policy specification and definition, as well as implementation architecture in Java. The benefit of this enforcement architecture is that he can define and implement the permission class in a mobile policy, maintaining the flexibility and compatibility with current runtime technologies. However, the downfall is that the system uses ‘Trusted Computing Devices (TCD)’ and depends on ‘Trusted Runtime Environment (TRE)’. Therefore, for this system to work, TCD and TRE are mandatory.

Ibharalu et al. in [9] have proposed the use of a chain of digital envelopes with platform registries to support dynamic agent’s itineraries in open network environment. This scheme protects and allows mobile agents to roam freely in open networks environment without being compromised in a malicious hosts. The main advantage is that the proposed scheme exhibited better performance when compared to the results obtained from obfuscation methods in terms of data integrity and security. The main drawback is that the proposed scheme consumes a little more time visiting platform registries and executing complex cryptographic functions than the obfuscation methods. Though the data is protected from hosts, the code is vulnerable to attack by other malicious agents residing in the host.

Nisha et al. presented a security solution [10] that protects both the mobile agent itself and the host resources that encrypt the data before passing it to mobile agent and decrypt it on the visited host sides. The method of "computing with encryption function" has been used. It

solves the problem of malicious host that can harm mobile agent or the information it contains. Here also, the itineraries are encrypted well in advance indicating the usage of static itinerary. No provision is given to secure code from other malicious agents.

Shibli et al. [11-12] Proposed a secure system for deployment of mobile agents. The system provides methodology that spans a number of phases in agent's lifetime: it starts from agent creation and ends with agent's execution. It addresses classification, validation, publishing, discovery, adoption, authentication and authorization of agents. This system is based on secure web services and uses RBAC XACML policies and SAML protocol.Though the work authenticates the code. The integrity of the code is not assured, and uses asymmetric encryption repeatedly increases the execution time of the mobile agent.

Leriche and Arcangeli [13] proposed a model of the mobile agent that is self-adaptive, by assembling reusable components. The agents are configured and may be reconfigured at runtime so that they are able to respond to changes in the execution environment.

The analysis, at the end of this study, showed that all approaches use the same security mechanism to protect mobile agents against different malicious platforms, without taking into account either the security requirements of each agent (the sensitivity of the information contained in the mobile agent) and the credibility of platforms. Indeed, each mobile agent requires different security mechanisms based on their services and based on the credibility of each visited platform.

3. A New Perspective of Security

Each mobile agent requires different security mechanisms based on their services andbased on the credibility of each visited platform. These security mechanisms are stored in the MSAS (the Management System of Agents Security) as security components (see Figure 1). The MSAS must support and update all existing security mechanisms. This approachis based on using two strategies of adaptation to adapt the mobile agent security at runtime, depending on the sensitivity of the services required to perform the duties of the agent and the degree of confidence of the visited platforms.

We have conceptually structured the overall system into three functional areas: a) Creation Area: where agents are created, validated, appraised, and published; b) Deployment Area: adds security components according to the sensitivity of the services required by the agent, and determine the adaptation policy to be followed by the mobile agent during its execution, and c) Execution Area: which contains actual runtime components for (physical network) agents. Agents traverse the network and perform their tasks in the Execution Area. Agent selects and adapts the components of security tasks to run on this platform. This adaptation is performed by the mobile itself depending on confidence level of the platform agent visited.

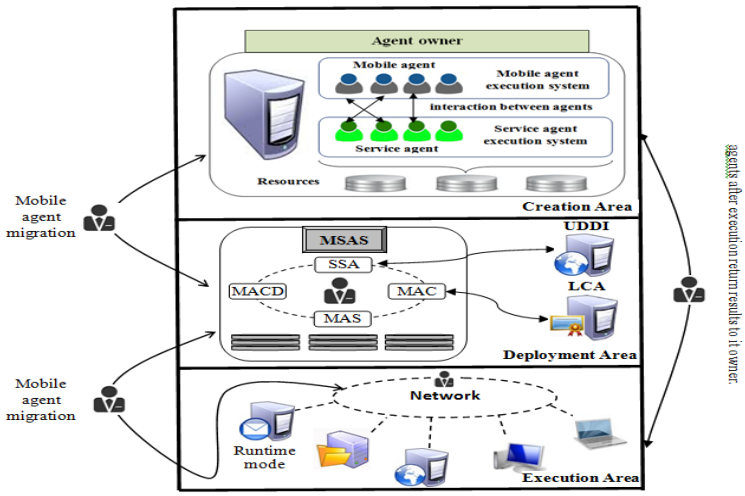

Figure 1. General architecture of the system

In order to increase the level of security offered and provide a satisfactory protection model, we also use traditional protection mechanisms such as password, cryptography and digital signature. These mechanisms help to preserve the integrity of the data of our mobile agent, and to control access to its resources.

3.1. Components of the System

3.1.1. SSA (System of Security Adaptation)

Systems based on mobile agents are characterized by a very dynamic aspect. This is due mainly to the migration of agents to multiple systems with different behaviors and security policies. Indeed, while visiting a new system, the agent must adapt dynamically to the security requirement.

The definition of an adaptation policy and security components of the mobile agent is a crucial step for the effective implementation of security in a mobile agent system. The goal of the SSA is to protect the mobile agent via a static adaptation (see Figure 2). This adaptation is to transform the mobile agent to a secure mobile agent. TheSSA adds additional security behaviors (security components) and determines its dynamic adaptation policy during execution.

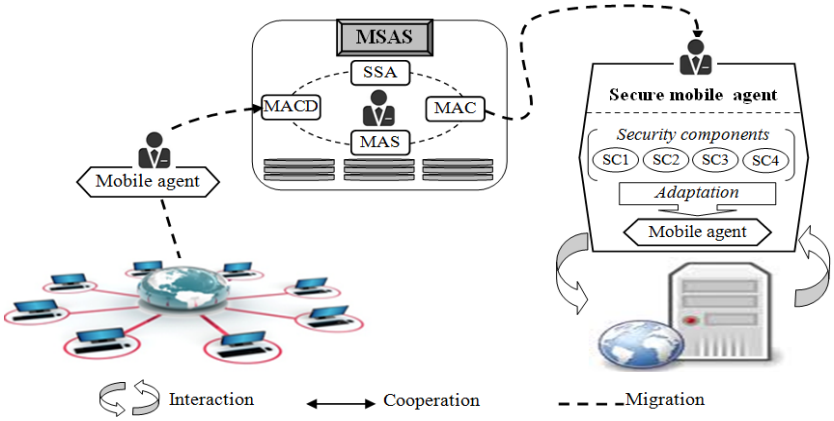

Figure 2. The adaptation of a mobile agent to theSSA

3.1.1. MACD (Management Agent of Confidence Degree)

Security agent requires dynamic assessment of the credibility of the host to visit, which can increase or decrease the security of the mobile agent. The mobile agent trusts the host when considering its degree of confidence as very favorable, or increase the level of security in the case of an unfavorable confidence degree. Hence, it is important that mobile agents can authenticate and identify the list of hosts in their itinerary.

MACD adds the necessary information for the estimation the degree of confidence by the mobile agent. This estimate is based on information stored in the mobile agent and the information collected from the environment visited. MACD updates such information following an inspection / observation by the mobile agent during its execution on the host visited.

3.1.2. MAS (Management Agent of Services)

The MAS is used to analyze and determine the sensitivity of the services requested by mobile agents (e.g. Top Secret, Secret, Confidential, Restricted, and Unclassified), then search and filter hosts that provide these services from UDDI server (see Figure 1). The objective of services classification is to protect the officer conducting sensitive tasks.

3.1.3. MAC (Management Agent of Certificates)

The MAC uses symmetric and asymmetric cryptography to prevent the behavior analysis of the mobile agent. Symmetric cryptography is used to encrypt / decrypt sensitive tasks of the mobile agent. The list of secret keys is shared between the MSAS and the hosts. Asymmetric cryptography is used to ensure authentication and security of communication

between the different entities of the system. For this, it uses reliable mechanisms such as encryption, hashing and digital signature. The MAC can also record and check the hosts'certificates (ID certificate, validate date ...).

3.1.4. UDDI Server

UDDI Server is based on the standard concept of UDDI servers, as specified by OASIS [14], i.e. web services publishing and discovery. In our system, UDDI Server acts as a registry containing a special category of services: agent provision web services. Any business entity (i.e. Agent Factory) who wants to provide agents to the community (i.e. end users) must properly expose a web service interface and correctly publish this web service interface.

3.1.5. Certificate Authority Local

Certificate authority local is an entity that issues digital certificates. The digital certificate certifies the ownership of a public key by the platform of the certificate. This allows the mobile agent to rely upon signatures or assertions made by the private key that corresponds to the public key that is certified. In this model of trust relationships, a CA is a trusted third party that is trusted by both the platform of the certificate and the agent relying upon the certificate. CAs are characteristic of many public key infrastructure (PKI) schemes.

3.2. Protection Protocol

This protocol aims to protect a mobile agent code against malicious hosts. The estimate the degree of confidence occupies the center of our works in ceit reveals the trust degree of the target host (see Figure 3).

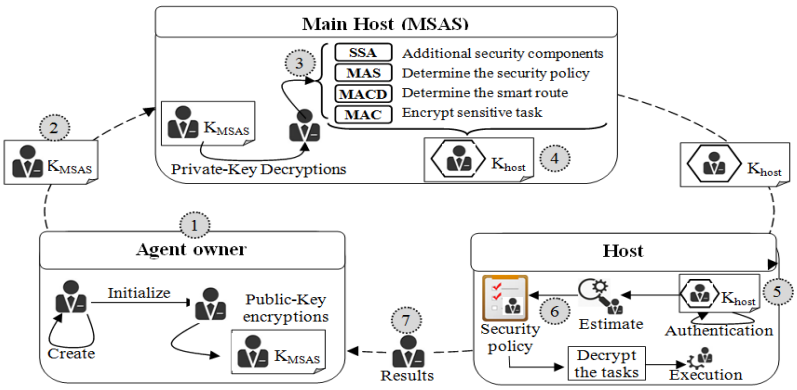

Figure 3. Scenario of mobile agent execution

1) The agent owner creates and initiates a mobile agent to perform the tasks requested by the customer. The behavior of the mobile agent is encrypted using the public key of MSAS, which allows it to guarantee the confidentiality and integrity of migration between owner and MSAS.

2) The agent owner sends the mobile agent to the MSAS.

3) The MSAS decrypts the behavior of the mobile agent using its private key, and then analyses and determines the sensitivity of the services requested by the agent, then adapt the agent's security and determine the smart itinerary.

4) The MSAS encrypt the sensitive tasks of the mobile agent. The list of secret keys is shared between the MSAS and the visited hosts.

5) The mobile agent must authenticate the host in question, and verify some information necessary for the performance of its task. To do so, the agent must obtain certain information from the runtime environment (for example, the identity of the host visited the password, digital certificate...). Then, the mobile agent encrypts the information collected using the public key of MSAS, to compare them with those it holds.

6) After authentications of the host, the mobile agent estimate the degree of confidence and extract the tasks that need to be executed on this platform. Depending on the

adaptation policy, the agent may decide to decrypt sensitive tasks and execute them on the host if the confidence is very favorable, add or replace security components if the degree of confidence is not favorable, or stop and leave the host to go to the next while notifying the MSAS about this failure.

7) The mobile agent returns to the agent owner with the results obtained from the different hosts.

3.3. Internal Structure of a Secure Mobile Agent

Figure 4 outlines the general architecture of a secure mobile agent model adapted by the MSAS. This architecture defines four main parts in an agent. The interface between the agent and the runtime platform. The basic level is the "operative part" of the agent. The meta level is non-functional services of the agent as security, communication, mobility. The agent's memory contains the information needed to perform the work of the mobile agent.

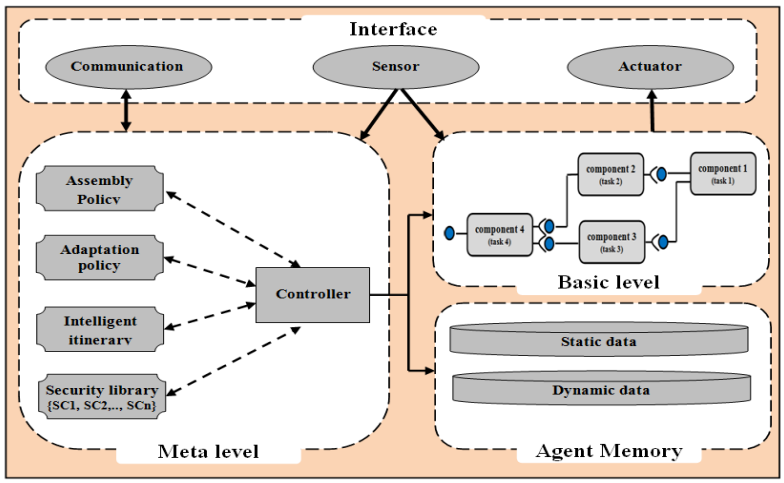

Figure 4. Internal structure of a secure mobile agent

3.3.1. Interface

This is the component through which the agent communicates and interacts with the runtime environment. This component allows authenticating platforms, detecting accesses of the external hosts to its resources, receiving requests and providing services in a suitable form. The interface contains three components:

1) Communication used to send messages to other agents and receive messages from other agents.

2) Sensor is used for the perception of the environment and the acquisition of data on the visited platform. These data are useful for authentication and the adaptation of the mobile agent.

3) Actuator is used to execute a sequence of actions on a selected component. The choice of executing component is determined by the controller.

3.3.2. Meta Level

The meta level represents the non-functional code of the mobile agent describing non-functional services and performing transformations on the basic level, the meta level contains components created by the agent owner and components added by the MSAS.

1) Policy assembly is a non-functional component added by the owner's agent, used to control the assembly process of components used by the agent, since the outbreak of an assembly to its realization. This policy also initializes the components with default settings.

2) Adaptation policy: represents a very important factor to support the protection strategy we propose for our mobile agent. Indeed, the adaptation policy is a set of rules that

defines the relationship between the security components of the agent and the degree of confidence associated with changes of the execution context.

3) Intelligent itinerary: consists of a set of nodes. A node is a connection point between the component to be executed by the agent, the identity and the confidence degree on the platform of execution. This section also contains the estimated time of the execution of each component in order to avoid replay attacks.

4) Security components contain security components replaceable that the mobile agent can use during adaptation. The components are added by the MSAS based on the security needs of the mobile agent. Each component is a piece of code selected by the controller According to the degree of confidence of each platform.

5) Controller plays the role of coordinator between the components of the mobile agent. Provides the proper functioning of the whole agent system and can also detect any poor execution of the mobile agent in the platform in question. The controller compares the actual execution time of each task with the estimated time. If the controller discovers an exceeding in the prescribed time, it can assume this as a misuse of the mobile agent and therefore, the agent must stop its execution in the concerned host in order to migrate to the next host in its itinerary. The controller sends notifications to MSAS, following a detected attack.

3.3.3. Basic Level

The basic level contains a functional component library of mobile agent, each component with a unique identifier and carries out a specific task, the assembly of components that implement the functionality of our mobile agent, combination and sequencing of these components to first be studied and specified by the assembly policy.

3.3.4. Agent Memory

Agent memory contains static and dynamic data transported from the original host or hosts visited by the mobile agent.

1) Static data: this is information that is transported from the original host. This information contains data that do not change such as the identity of the creator and the digital signature.

2) Dynamic data: this is information collected from the runtime environment after each migration. Such As the partial results of calculation of each platform, the data obtained during the execution of the agent.

3.4. Trust Estimation

The quantitative dimensions of trust are based on the quantitative dimension of its cognitive constituents. To determine the host trust, we must identify the following parameters:

1) Parameters that make a transaction trustworthy.
2) Parameters that determine a level of trust of every costumer.
3) Parameters that determine a set of costumers to which the host belongs.
4) Software and hardware parameters that may affect perception of trust and transaction fulfillment.
5) Reputation (may not exist) of hosts provided by the agent's owner or a third party and witch related to the history of host's transactions.

The trust degree estimation of the visited host is calculated using the collection of the values of certain parameters starting from the environment. The trust degree T is calculated according to importance Ij, weight Wj of the parameter j and factor Sj which is equal to 1 in the case of success (conformity of information), and equal to 0 in the case of a failure (non-conformity of information). The trust degree estimation is performed according to the following formula (1):

$$T(Trust) = \sum_{j=1}^{k} (W_j I_j S_j) \quad (1)$$

Each parameter J has predefined values of its importance and its weight. These values are stored into the mobile agent memory. With an aim of deciding on an adequate reaction, the value of T is compared with the limits of the various trust estimation intervals (see Table 1). If the trust value belongs to the good interval (e.g., [81-100]) the host is trusted, the agent can

execute safely. On the other hand, if the obtained trust value is considered to be low, it is possible to find the exact cause of this failure by seeking among the parameters which had a factor S equal to zero, and then the mobile agent selects and adapts security components to the tasks to be performed by the host. Given the importance of some parameters, they can (in case of failure) largely influence the choice of the action to be undertaken. The values assigned to the attributes (the weight and the importance) of each parameter define its impact in the final decision.

Table 1. Example of estimation intervals with their related feedback

Interval of trust estimation	Feedback	commentary
0→20	Stopping	The mobile agent will stop and exit the current host after notifying the reason for which he took the decision and go to the next host.
21→80	Reducing	The mobile agent selects and adapts security components to the tasks to be performed by this host.
81→100	Performing	The agent can run safely

4. Conclusion

Our study showed that the security of the mobile agent requires the addition of a dynamic assessment of the credibility of the host visited and determination of the sensitivity of the services requested by the mobile agent. Accordingto our knowledge, no approach has used the service sensitivity and adaptability to solve this problem. We have proposed a new architecture for mobile agents and identified the functions of different system components. This proposal is based on two strategies of adaptation. The first is a static adaptation performed by the MSAS. The second is a reflexive dynamic structural adaptation performed by the mobile agent itself.

References

[1] M Alvaro, R Sergi. *Component-Based Development of Secure Mobile Agents Application*. In Proceedings of Multi-Agent Systems and Applications V. 2007: 113-122.
[2] G Samaras. Mobile agents: What about them? Did they deliver what they promised? Are they here to stay?. University of Cyprus. 2004: 294-295.
[3] G Karjoth, DB Lange, M Oshima. *A Security Model for Aglets*. In IEEE Internet Computing. 2009; 1(4): 68-77.
[4] S Zhidong, W Xiaoping. *A Trusted Computing Technology Enabled Mobile Agent System*. In Proceedings of International Conference on Computer Science and Software Engineering. 2008: 567-570.
[5] R WANG, T HU, XX. Research in to mobile agent security. *Journal of Chongqing University of Posts and Telecommunications.* 2004; 16(3): 81-86.
[6] N Lukasz, P Marcin, R Michal. Mobile agent security. 1st Thomas edition. Information assurance and computer security. 2006; 102-123.
[7] P Bella vista, A Corradi, C Frederici, R Montanari, D Tibaldi. Security for Mobile Agents: Issues and Challenges. In: I Mahgoub, M Ilyas. *Editors*. Invited Chapter in the Book Handbook of Mobile Computing. CRC Press. 2004.
[8] X Zhang, F Parisi-Presicce, R Sandhu. *Towards Remote Policy Enforcement for Runtime Protection of Mobile Code Using Trusted Computing*. International Workshop on Security (IWSEC). 2006.
[9] FT Ibharalu, AB Sofoluwe, AT Akinwale. A reliable protection architecture for mobile agents in open network system. *International journal of computer applications.* 2011; 17(7): 6-14.
[10] P Nisha, K Sunil, B Ashu. Security on mobile agent based crawler. *International journal of computer applications.* 2010; I(IS): 5-11.
[11] A Shibli, I Yousaf, S Muftic. *MagicNET: Security system for protection of mobile agents.* In Proceedings of IEEE International Conference on Advanced Information Networking and Applications. 2010: 1233-1240.
[12] MA Shibli, S Muftic, A Giambruno, A Lioy. *MagicNET: Security system for development, validation and adoption of mobile agents.* In Proceedings of 3d International Conference on Network and system security. 2009: 389-396.

[13] S Leriche, J Arcangeli. Flexible architectures of adaptive agents: the agent approach. *International journal of grid computing and multi agent systems (IJGCMAS).* 2010; 1(1): 55-75.
[14] T Bellwood, et al. UDDI Version 3.0. UDDI Spec Technical Committee Specification. 2002. web: http://uddi.org/pubs/uddi-v3.00-published-20020719.htm.