International Journal of Security and Its Applications
Vol. 10, No. 4 (2016), pp.117-134
http://dx.doi.org/10.14257/ijsia.2016.10.4.13

Security of Mobile Agent Platforms using RBAC based on Dynamic Role Assignment

Hind IDRISSI[1,2], Arnaud REVEL[2] and El Mamoun SOUIDI[1]

[1] *Laboratory of Mathematics, Computing and Applications, University of Mohammed-V in Rabat, Morocco*
[2]*L3I Laboratory, University of La Rochelle, France*
hind.idr@gmail.com, revel.arnaud@univ-lr.fr, emsouidi@fsr.ac.ma

Abstract

Mobile agent technology is a new trend in the network computing. It succeeds to capture researchers' and industry's interests long time ago, due to its innovative capabilities and attractive applications. Mobile agents are self-contained data and software modules, able to autonomously move from one host to another across the network, in order to perform their tasks and eventually return to their initiators. Despite their several qualities that make them suitable for various disciplines, where autonomy, dynamism and flexibility are strongly recommended, mobile agents still suffer from some limitations mainly related to the security issues raised by mobility. In this paper, we propose a novel approach to address the security problems evoked by platforms hosting mobile agents, particularly those caused by unauthorized access attacks. Our approach introduces a robust security policy for a Hospital, where a flexible role-based access control model (RBAC) is used and simulated as a set of cooperative agents. We implant a privilege management infrastructure (PMI) charged with issuing attribute certificates based on elliptic curve cryptography, in order to provide this model with a dynamic role assignment. Finally, practical experiments are conducted to evaluate our approach and prove its effectiveness, reliability and security.

Keywords*: Mobile Agent Platform, Security, RBAC, Dynamic Role Assignment, Privilege Management Infrastructure, Attribute Certificate*

1. Introduction

Recently, the systems based on mobile agents are widely emerging, due to their great capacities in designing, implementing and maintaining distributed systems. They are used in open, complex and highly dynamic environments, that are employed by diverse evolving disciplines, such as: Telecommunications [1], Internet with E-commerce [2], optimization of transport systems [3], personal information management [4], computer games [5], industry [6] and many other fields.

According to Ferber [7],"an agent is a physical or virtual entity able to act in an environment; it possesses its own resources and gives services as individual objectives". In other words, an agent is an independent module characterized by its capacity to proceed autonomously in its environment, taking into consideration a specified perception of this latter. When mobility is associated to these agents, they are called "mobile agents", since they become able to physically leave their owner machines, while carrying their own resources (code, data and state), and move from one host to another across the network to execute tasks.

The mobility feature of agents is strongly recommended to provide efficient and elegant solutions for a large variety of problems such as network traffic and latency. However, security is an increasingly concerned issue of this technology, especially

ISSN: 1738-9976 IJSIA

for the platforms hosting the mobile agents. These platforms are vulnerable to potential attacks from foreign agents with malicious behaviors, such as unauthorized access and repudiation.

In this paper, we propose a robust approach to deal with the security issues raised by platforms of mobile agents, especially in terms of unauthorized access. Since the integration of intelligence in the field of health care becomes of a great interest for researchers and scientists, we were motivated in applying our approach on a Hospital, as an organization that includes different, variant and sensible components. Thus, we make use of RBAC model to design a flexible, viable and interoperable architecture, where several cooperative mobile agents are implanted to support the various levels and duties of the organization. Moreover, to provide this latter with a dynamic role assignment, we develop a privilege management infrastructure (PMI) endowed with a role engineering entity and a storage database. Using the elliptic curve cryptography, the PMI delivers X.509 attribute certificates for the legitimate users, where the authorized role is specified. Then, the certificates are verified by an integrated access control policy server in the hospital platform, in order to setup a well defined access control policy for the requesting actors. The proposed approach has been evaluated basing time and security factors, and it has showed promising performances.

The remainder of the paper is structured as following. In Section 2 we give a formulation of the security problems in the systems based on mobile agents. Some related works are discussed in Section 3. In Section 4, we describe in details the proposed approach. Section 5 provides the results of the conducted experiments to evaluate the proposed approach. Further discussions and perspectives are mooted in the conclusion.

2. Problem Formulation

Mobility is an important feature of agents, which is very requested by the distributed systems that aim at performing computations without being permanently connected and ensuring availability to improve the efficiency. However, mobility of agents raises serious security problems, as illustrated in Figure 1.

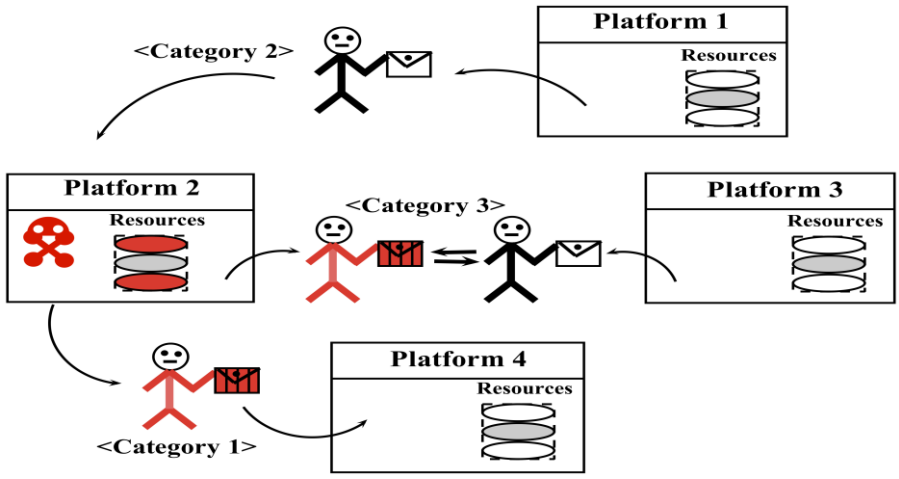

Figure 1. Categories of Security Issues in Mobile Agent Systems

Indeed, there are three categories of vulnerabilities that may be considered along the mobility of the agent, as mentioned in [8], and which are:

- **Agent against Platform**: a mobile agent could have free and unauthorized access to the runtime environment and thus, it could violate its confidentiality, integrity and availability by intercepting or modifying its data, fully exploiting its resources, cloning or migrating indefinitely.

- **Platform against Agent**: when an agent is migrating to a new destination, namely new platform, it has to expose in clear its code, status and data, which makes it susceptible to confidentiality and integrity threats on behalf of the hosting platform, which exploits its information and manipulates its behaviors and results.
- **Agent against Agent**: an agent moving over the network may undergo several attacks from other agents encountered in the itinerary. In this regard, there are two defined types of attacks: passive attacks that just observe and analyze behaviors and results of the agent without changing its code and its data, and active attacks that could alter the code or data of the agent, by changing the variables values or inserting a virus.

In this paper, we attempt to address the security issues associated to the first category, where an agent platform could be exposed to a variety of attacks aiming at violating its security policy.

3. Related Works

In the context discussed in previous section, many efforts have been devoted to investigate the security issue of platforms hosting mobile agents. Thus, many solutions have been proposed in literature [9], which are mostly related to the authorization and access to the resources of the platform, as well as to the recording of the agent's behaviors during its authorized stay. Among these solutions, Arun and Shunmuganathan [10] have proposed a technique called "Sand-boxing", which allows the execution of the mobile agent's code in a restricted environment (sandbox) that appears similar to the global system, and where restriction affects certain code operations. This ensures that a malicious mobile agent can not cause any harm to the execution environment that is running it. Nevertheless, in addition to the extra charges of such simulated environment, this latter can neither reach the capacities of the origin system, nor provide realistic experiments. Tsiligiridis [11] has introduced the concept of "State Appraisal" that makes use of appraisal functions, to verify the execution of an incoming agent and determine the privileges granted to an agent, based on conditional factors and invariants. However, the major issue of this technique is the difficulty to formulate suitable security properties and obtain appraisal functions that guarantee these properties. Pirzadeh *et al* [12] have proposed a principle called "Proof Carrying Code", which analyses the semantics of the agent's code before being launched, basing its structure and behavior. It makes the author of an agent generates an encoded evidence or proof, that the code adheres to the security policy of the consumer. Then the host can quickly verify and validate the proof, through reliable and automatic procedure. However, this technique shows some limitations related to the size of the proof and the time consumed to validate it. Tuohima *et al* [13] have developed a "Model Carrying Code", where the host forms a model that captures the security relevant behavior of the code, rather than a proof, through using specific information to accompany the untrustworthy code. So that the code consumers are able to know the security needs of untrustworthy code more precisely. Cao and Lu [14] have introduced the approach of "Path Histories", whose principle is to maintain an authenticable recording of the platforms already visited by an agent, since this agent during its travel life may be hosted by multiple platforms with various trust levels. Thus, each platform visited by the mobile agent must append a signed input, according to which, the current hosting platform decides whether to execute the agent or not, and what privileges and services should be granted to him. The main drawback of this technique is that the path verification process becomes increasingly expensive with the increase of visited hosts.

The mentioned approaches still suffering from some restrictions to reach the security level requested, and they usually need to be combined with other mechanisms, which burden the system.

4. The Proposed Approach

During the last few years, the field of telemedicine witnesses quickly and steadily growth. It refers to the delivery of health care using information and communication technologies. Thus, exploring intelligence technologies, such as mobile agent systems, for assisting medical services or transmitting personal information has been widely investigated [15]. Autonomy, proactivity, negotiation strategies, collaborative skills and learning mechanisms are the well-known properties of mobile agents, which may facilitate the efficient deployment of health care systems. However, the security problems raised by mobile agent paradigm, when faced with the high sensibility of the data involved in these systems, represents a brake to its adoption in that discipline. According to this, an agent platform incorporating a medical organization may be attacked by foreign mobile agents that attempt to illegally access the patient information. Moreover, mobile agents owned by the same agent platform could be maliciously affected and manipulated during their migration over the network. For that reason, we were interested to apply our security approach on a Hospital provided with a mobile agent system to ensure collaborative and communicative tasks and services.

Our proposed approach to secure mobile agents platforms is mainly relied on elaborating an access control policy for the platform and its resources, with a dynamic role assignment based on attribute certificates and elliptic curve cryptography (ECC) [16].

We were motivated to use ECC due to its important benefits, which consist in: smaller key sizes, less storage, and faster implementations than traditional cryptographic protocols (e.g. RSA). That is, an ECC system could provide the same level of security requirements afforded by RSA-based systems, by using smaller key sizes. Thus, ECC-based protocols which reduce the storage and transmission requirements are suitable for entities endowed with mobility feature and limited storage capacities

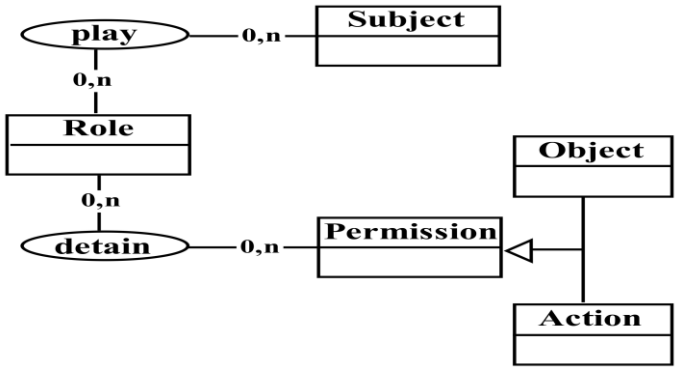

Figure 2. Structure of RBAC Model

Moreover, we make use of RBAC (Role-Based Access Control) model to design our adopted architecture. This model was firstly proposed by Ferrailo and Kuhn [18] in 1992, with the aim to make the administration of access policies easier. RBAC model is mainly based on the "role", which represents a function or a job within an organization, and combines the authority and the responsibility entrusted to the person who plays this role, ex: director, engineer, teacher, etc. Each role detains a set of privileges called "Permissions", which consist of a set of rights corresponding to the tasks that can be performed by this role. In such way, the permissions are not directly associated to subjects, but through roles using two relations as indicated in Figure 2.

A subject may play many roles, and a role can be assigned to multiple subjects. In addition, a role may detain several permissions; as well permission can be associated to several roles.

According to this, and in order to well define and implement our solution, we perform an RBAC modeling for a Hospital, as indicated in Figure 3. This modeling offers administration features that incorporate the activities within the hospital information system. It contains a set of roles, where some of these roles inherit from others. Each role is associated with a set of permissions that allow the execution of activities, with respect to specific constraints (emergencies, diseases, wars...).

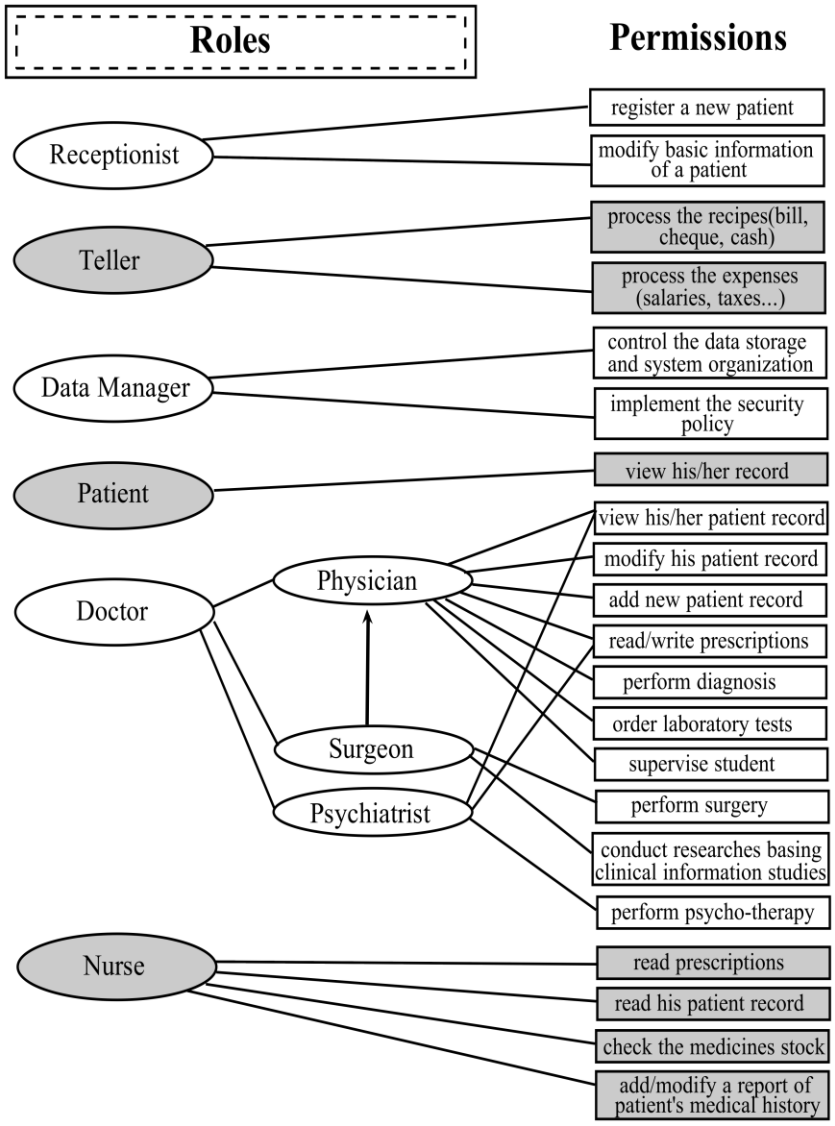

Figure 3. RBAC Model for a Hospital

4.1 The Architecture of the Platform

In order to make our RBAC model flexible and more interactive, we introduce cooperative mobile agents to simulate our architecture. Our basic idea for the construction of the agents platform, is to make each role and each permission allowing an activity be represented by a mobile agent charged with consulting services or executing requested actions.

Figure 4 illustrates the proposed architecture of our platform, which is composed of five main entities:

- ***Manager Agent***: it is responsible for the management of the communications among the agents within the same platform or those of remote platforms. In addition, it is charged with the administration of sensitive data and the monitoring of the different policies (security, storage...).
- ***Container-1***: it contains a set of mobile agents representing the different roles that can be assumed within the Hospital.
- ***Container-2***: it assembles a set of mobile agents representing the permissions that allow the execution of the specific activities, when the associated role is granted the right to do. Every permission is attributed with a private key, which will be used to bind the certificate with the targeted role and permission.
- ***Objects***: this represents the set of objects that the permissions need to deal with. It may concern the patient records, the prescriptions, the database containing the information about the patients and the hospital staff, the applications associated to specific services or materials, *etc*.
- ***Access Control Policy Server***: it is a component related with the application and the monitoring of the restrictions and specifications in the adopted access control policy.

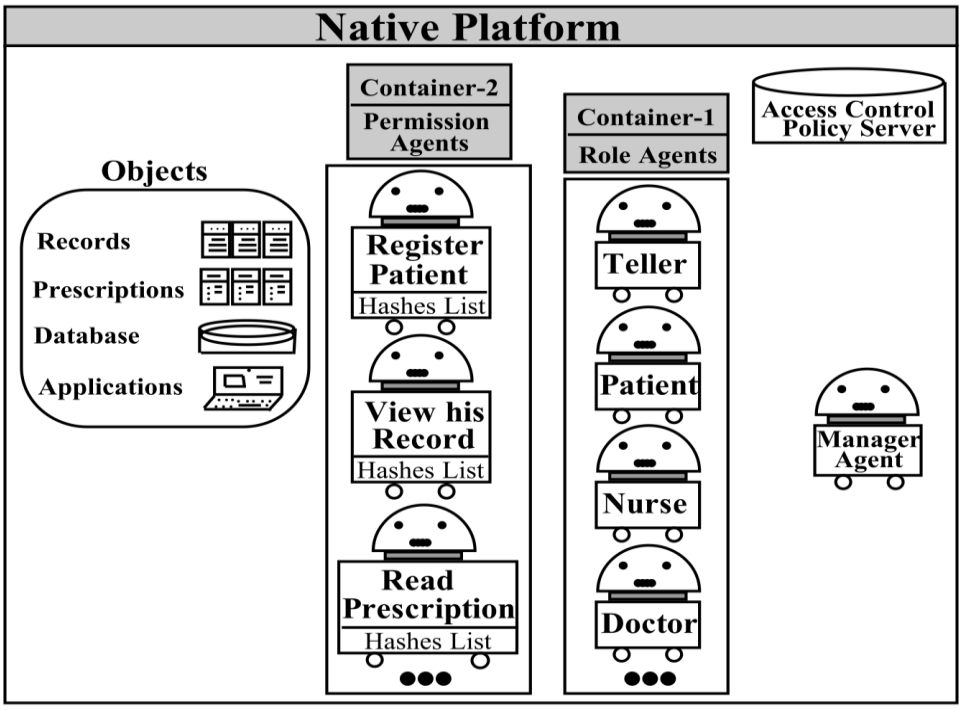

Figure 4. Architecture of the Proposed Agents' Platform

4.2 The Privilege Management Infrastructure (PMI)

The majority of information systems that implement traditional access control policies make use of standard public key infrastructures (PKI), to perform authentication among the components. These infrastructures rely on centralized access policies, based on Certificate Authority (CA), to limit the access to the system resources. However, they suffer from serious drawbacks that make the mission to implant open and scalable systems very difficult, especially for mobile agent systems which require flexibility, scalability and heterogeneity.

A privilege management infrastructure (PMI) [19] provides an alternative to PKI-based resource access control. PMI is the infrastructure needed to manage attribute certificates in an open network. It allows to assign authorizations (permissions or credentials) to concrete entities, as well as to trust delegation among entities. As a major component in privilege management infrastructure (PMI), X.509 attribute certificates allow the construction of a scalable and interoperable authorization infrastructure. These attribute certificates do not employ public keys, but bind entities to attributes, which may be roles, identities, information groups, *etc*. The attribute certificates are signed and issued by an attribute authority (AA).

There are three basic and substantial components in the construction of a PMI:

- ***Privilege Asserter***: it represents the entity holding privileges and asserts them.
- ***Object***: it refers to the resource being protected.
- ***Privilege Verifie***r: it is the entity responsible for performing the necessary verification to decide whether allowing the usage of the object or not.

According to this, four models of PMI are considered: general model, control model, delegation model, and roles model. In our approach, we make use of the roles model as illustrated in Figure 5. In this model, the privilege asserter is the subject which may be a human user, a robot, an application or a system. This subject interacts with a PMI attribute authority to request and issue attribute certificates, which allow him to get access to the protected resources. These later are owned by our native platform, which plays the role of the privilege verifier charged with checking the privileges asserted by the subject, that forward its attribute certificate along with his access request.

It is worth to mention that within the PMI role model, it is necessary to associate the privilege asserters with roles; this is called "*Role Assignment*". Similarly, once the roles are assigned, it is necessary to associate them with permissions; this is called "*Role Specification*". The two mechanisms can be expressed and contained in the attribute certificate.

Our PMI attribute authority (AA) is composed of three main entities:

- *Attribute Certificate Server (ACS)*: it is responsible for the elaboration, signature and delivery of the attribute certificates, where the assigned role and the specified permission are provided.
- *Certificate Storage (CS)*: once an attribute certificate is conceived and signed, a copy of this certificate is stored in a database of certificates, which is accessible for public.
- *Role Engineering (RE)*: it is the entity charged with the application of the role-based policy. It is supposed to retain and manipulate secret information about the definitions of roles and their associated permissions.

In our approach, the privilege verifier is at the same time our native platform that hosts the data and resources to be protected. Thus, we integrate in our platform an "*access control policy server"*, which makes decisions about access rights according to the binding information in the attribute certificates.

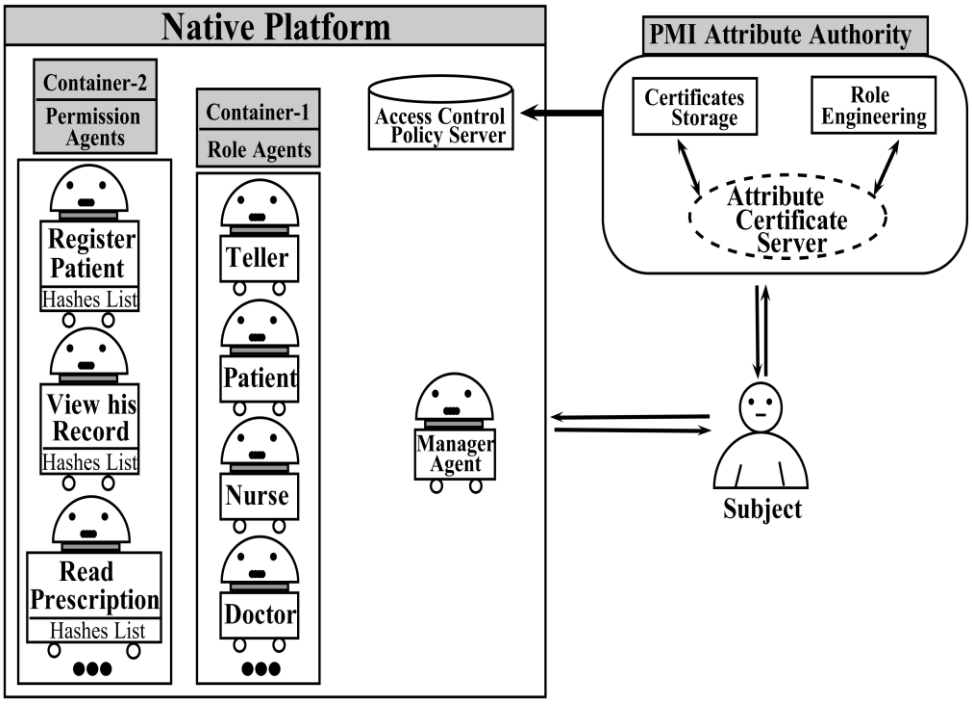

Figure 5. Platform Architecture using PMI-based Role Model

4.3 Dynamic Role Assignment using ECC-Attribute Certificate

Our approach attempts to implant an interoperable authorization strategy, which relies on an agents-roles model based on attribute certificates. For that purpose, we make use of ECC-based certificates [20] that have been standardized by IETE as PKIX-X.509, which is almost similar to X.509 with a main difference of using the elliptic curve digital signature algorithm (ECDSA) [21]. Figure 6 illustrates the format of the ECC-based X.509 certificate, such that we replace the public key of the user (the subject) by the corresponding attributes for the access control policy, which consist mainly of the role and the permission associated.

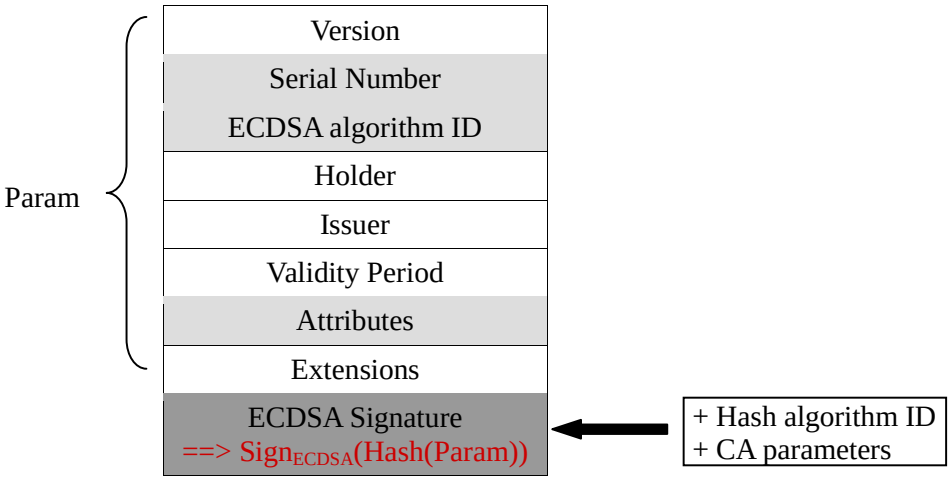

Figure 6. The Basic Format of the ECC-based X.509 Certificates

At this stage, in order to understand the processing of our approach, it will be interesting to consider a scenario. Thus, let us suppose that a subject wants to get the role of a "*nurse*" and the permission to "*modify the report of his patient's medical history*", which we abbreviate as "*MRPMH*". Both, the subject and the native platform own a pair of public and private keys, derived using elliptic curve cryptography, such that each interaction between them is encrypted using the public key of the recipient, while the secret key of the receiver is used to decrypt it. Figure 7 shows the interactions occurred during the process of this scenario.

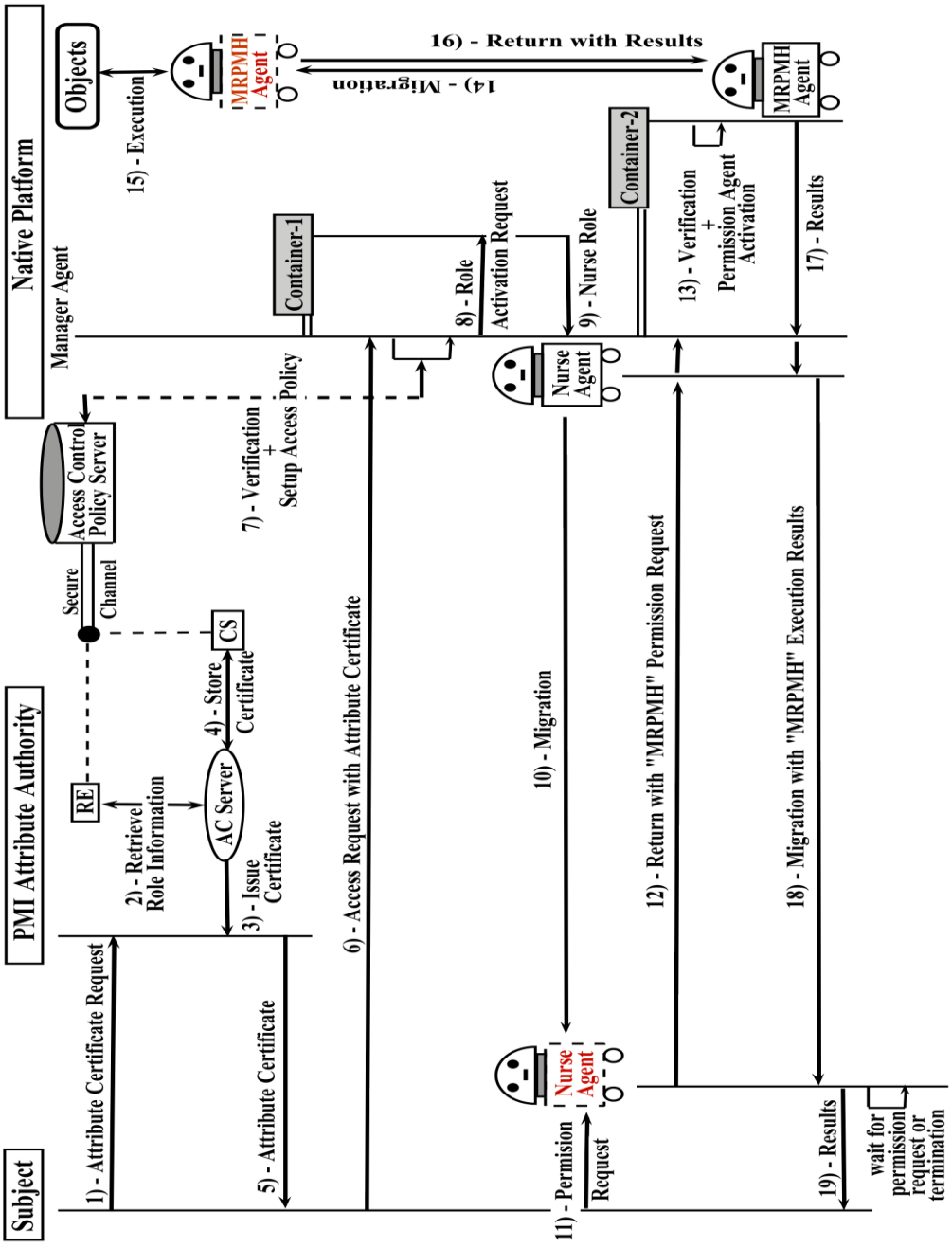

Figure 7. The Process to Assign the Role "Nurse" with the Permission "MRPMH"

According to this, the subject sends a request along with its identification credentials (subject_id, subject_name, ECC-public key, *etc*) to the PMI Attribute Authority (AA), in order to issue an attribute certificate for him. Then, the "AA" retrieves the necessary

information about the corresponding role from the "RE" entity, that involves a database of roles with log files to designate the transactions and actions made by this subject within the native platform. These information are directed to the "AC Server" that generates and issues the corresponding attribute certificate to the named subject, while a copy of this certificate is stored in the "CS" entity.

In possession of its attribute certificate, the subject uses it to send an access request to the native platform, namely to the "Manager Agent". This later takes in charge to verify the certificate and ask the "*Access Control Policy Server*" (ACPS) about the subject and the privileges it may have access to. According to this and since the "ACPS" is connected to the "RE" and the "CS" through a secure channel, an access policy is implanted by this server, where the access control decisions depend on the roles in the attribute certificate. A simple example of such role-access policy is as the following:

```
<RoleAssignmentPolicy>
   <RoleAssignment>
      <Subject ID="Sub-008",
               PublicKey="04:49:8D:1A:12:8A:CE:00:5F:18:..." />
      <Role>
         <ID="C1-NS10-371-230216"/>
         <Type="CNT-1 Role"/>
         <Value="Nurse"/>
      </Role>
      <Delegate Depth="0"/>
      <PolicyOwner ID="ACPS-9Z01"/>
      <Validity>
         <Absolute Start="2016-02-20T12:00:00"/>
      </Validity>
   </RoleAssignment>
</RoleAssignmentPolicy>
```

Once the verification is successfully processed and the access policy is implanted, such that the role in question is well defined, the "Manager Agent" sends a request to the "Container-1" in order to activate the mobile agent which corresponds to the named role. Thus, a "Nurse Role-Agent" is activated; encrypted using the public key of the subject, and it migrates to the location of the subject. Once there, it is decrypted using the private key of the subject, and it waits for services requests in the context of the defined role. To perform these services, permissions to the associated objects are needed. According to this, the subject sends a request for the "MRPMH" permission to the "Nurse Role-Agent", which returns back to the native platform after being encrypted using the public key of this latter. Within the native platform, the "Manager Agent" forwards the permission request to the "Container-2", which verifies whether the "Nurse Role" can be granted the permission "MRPMH" or not.

In our approach, each permission agent in the "Container-2" holds necessarily a list of hashed values, which correspond to the identities of the roles, which may be granted this permission. Thus, the "Nurse Role-Agent" has to hash its "Role_ID" using SHA-3 (256 bits) [22] and send it to the "Container-2" that verifies the existence of this hashed value in the list owned by the "MRPMH Permission Agent", as shown in Figure 8. If so, then the specified permission can be granted to the requesting role agent. Else, an acquittal of denied permission is sent to the "Nurse Role-Agent" in order to be forwarded to the subject.

In the case that the permission verification is successfully processed, a mobile agent charged with the "MRPMH" permission is activated, and moved to the area of the objects concerned by the action in the permission. Once being executed, the "MRPMH Agent" returns back to the "Container-2" along with the results, which are directed to the "Nurse Role Agent". This latter, is encrypted before moving, with the results of "MRPMH" execution, to the subject location. When receiving the "Nurse Role Agent", the subject decrypts it and extracts the carried results. At the end, this process can be partially

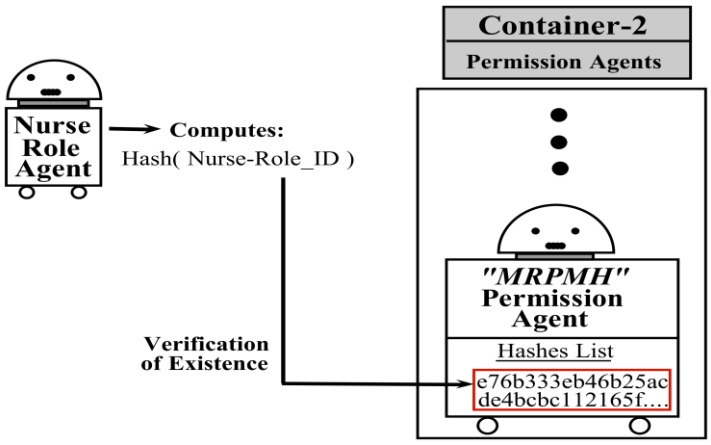

Figure 8. The Verification of "MRPMH" Permission Eligibility for the "Nurse" Role

repeated as much times as the subject needs permissions under the defined "Nurse Role", which stays activated waiting for more requests or for termination.

Our approach is conceived in such way to approve a security strategy, that aims to achieve the following essential concepts:

- Instead of hosting outer and foreign subjects within the native platform, the communications are more controlled and secured from far, through creating mobile agents charged with the execution of the requested actions.
- Every transaction across the network is encrypted before being addressed and decrypted once arrived.
- The adoption of the separation of duties "SOD" principle allows reducing the probability of intrusions and accidental damages. Our approach integrates two level of SOD: "Static" which is illustrated in the subject-role assignment, and "Dynamic" that is based on the role-agent activation.
- Least Privileges principle is also respected, since each activity or action is assigned to a specific agent, such as: the "Manager Agent" responsible for managing communications and performing necessary verification, the "Role Agent" which supports the knowledge of the services and activities allowed by this role, the "Permission Agent" that is charged with the executing the assigned permission. This allows reducing the lifetime of each agent, in order to be unable to adopt any malicious behaviors or eavesdropping the sensitive data and the secure exchanges of the native platform.

5. Evaluation

In this section, we proceed to an evaluation to prove the feasibility, effectiveness and security of the proposed approach, through investigating practical experiments. Thus, we have implemented a hospital application hosted by a mobile agent platform, such that the proposed architecture and its relevant components are considered:

- **Privilege Asserter**: we develop a simple platform using Microsoft Internet Explorer (version 8.0), and which is based on ActiveX Control. This allows the identified subjects to download the attribute certificates encoded by the PMI-AA. The platform is also equipped with JADE (4.3.3) [23] FIPA-compliant agent framework, that allows to receive mobile agents, communicate with them and dispatch them.

- **PMI Attribute Authority**: in order to produce attribute certificates, we conceive an attribute certificate server (ACS) in Eclipse using JAVA. This is achieved by the programming library called "*Bouncy Castle Cryptography Library 1.54*", which includes a class "X509AttributeCertificateHolder" that supports the functionality related to the generation of the attribute certificates. An example of such attribute certificates is provided in Figure 9. Concerning the certificate storage (CS) and the role engineering (RE) entity, they are developed using the Java SDK of Netscape Directory Service.

```
AttributeCertificate ::= SEQUENCE {
  acinfo                  AttributeCertificateInfo,
  signatureAlgorithm      SHA256withECDSA,
  signatureValue          cd:8d:d6:95:af:f1:d2:8d:85:df:e8:6d:a7:ed:b6:
                          48:a4:05:ed:b6:b5:65:97:75:a0:66:84:b5:a1:38:
                          bd:62:74:33:ef:be:43:25:06:b9:92:a2:0d:43:2d:
                          59:92:b0:ba:e9:d2:ed:84:e1:dc:de:40:b8:5d:f9:
                          ..... (256 bit) }

AttributeCertificateInfo ::= SEQUENCE {
  version                 3 (0x2),
  serialNumber            1 (0x1),
  issuer                  CN=root-PMI-AA/Email:sub-008@Hospital.com,
                          O=Hospital.com, L=LaRochelle, ST=CM, C=FR
  subject                 CN=User1/Email:sub-008@Hospital.com,
                          O=Hospital.com, L=LaRochelle, ST=CM, C=FR
  acValidityPeriod
                          Not Before: Feb 20 09:00:00 2016 GMT
                          Not After: Jul 01 08:59:59 2016 GMT
  attributes              Role-ID=C1-NS10-371-230216
                          Role-value=Nurse
                          Policy=RBAC-PMI
  issuerUniqueID          PMI-AA-NTP1100
  X.509v3 extensions
                          Netscape Comment:
                               OpenSSL Generated Certificate
                          Subject ECC-PublicKey (256 bit):
                               04:49:8D:1A:12:8A:CE:00:5F:18:2A:3C:37:
                               0E:A9:CC:71:B6:4B:A5:...
                          PMI-AA ECC-PublicKey (256 bit):
                               04:A9:13:B6:BE:C4:5A:22:C0:BB:E7:34:15:
                               FF:91:2B:54:ED:26:7E:....

-----BEGIN CERTIFICATE-----
QKEw5TcGVubmViZXJnLmNvbecgruQkua0yfTtvvplcxEjAQBgNVBAcTCVN0
EBBAUAA4GBAG+JKAc3Blbm5lYmVyZy5bXcqzBO37WTzFJT7z/aW5mdXJ0MR
cwFQYDVQM++KckxnWOp9CZ6qfttYJKoZIhvcNAQkBFhNyYWUEBhMCREUxsd
OU3Blbm5lYmVyZy5jb20xFDASBgNVBELMAkGA1UEBhMCREUxDDAKBgNVBAg
TA05ZWluZnVydDEXMBUGA1UEChMOU3Blbm5lYmVyZy5jb20xFDASBgNVBAM
TC1Jvb3RDQSAyMDAzMSIwIAY.........

-----END CERTIFICATE-----
               }
```

Figure 9. Example of Attribute Certificate Generated during the Process of our Approach

- ***Privilege Verifier (Native Platform)***: it is a platform equipped with JADE framework to create and monitor agents, and it is connected to internet via an HTTP server. The platform also includes an engine working as access control policy server (ACPS). This latter is developed using java, and allows verifying the validity of the attribute certificate, *i.e.*, the signature and the period of the certificate, if it is not valid, the access request is denied. Otherwise, the "ACPS"

checks if there is a previously received and valid certificate relating to the same subject in the policy database. If it is not the case, then the "ACPS" extracts the role from the attribute certificate and setups the associated role-assignment policy.

Our evaluation will focus on two important factors: time performance and security, which are described in details in the forthcoming sections.

5.1 Time Performance

In this section, we measure the time spent during the process of our approach. Considering the same scenario adopted in Section 6, let *Trp* be the total time cost of performing the nurse role assignment with granting the permission "MRPMH". *Trp* involves many periods related to the manipulation performed, which are listed in Table 1. According to this table, *Trp* can be calculated by the Equation 1:

$$\boldsymbol{Trp} = Tsr + Tre + Tacs + Tv + Tpr + Traa + 3Trae + 3Tram + 3Trad + Tpv + Tpaa + 2Tpam + Tpae \quad (1)$$

Table 1. The Sub-Periods Related to the Process of our Approach

Period	Description
Tsr	The time of the requests submitted by the subject
Tre	The time of retrieving role information from the "RE"
Tacs	The time of generating and issuing the attribute certificate
Tv	The time of verifying the attribute certificate and setting up the access policy
Tpr	The time of the requests performed within the native platform
Traa	The time of the role-agent activation
Trae	The time of the role-agent ECC-encryption
Tram	The time of one role-agent migration
Trad	The time of the role-agent ECC-decryption
Tpv	The time of the permission verification
Tpaa	The time of the permission-agent activation
Tpam	The time of one permission-agent migration
Tpae	The time of one permission-agent execution

Since the migration of the permission agent is performed internally without any network configuration or charges, and also because this agent carries a small amount of information, we can neglect the time cost dedicated for this operation. Thus, the Equation (1) becomes:

$$\boldsymbol{Trp} = Tsr + Tre + Tacs + Tv + Tpr + Traa + 3Trae + 3Tram + 3Trad + Tpv + Tpaa + Tpae \quad (2)$$

Table 2 presents the measurements of these sub-periods and the total time *Trp*:

Table 2. Time Cost of the Different Operations in our Approach

	Tsr	*Tre*	*Tacs*	*Tv*	*Tpr*	*Traa*	*Trae*	*Tram*	*Trad*	*Tpv*	*Tpaa*	*Tpae*
Time (ms)	*1,3*	*28,6*	*17,59*	*9,22*	*0,67*	*3,31*	*3,52*	*122*	*1,89*	*2,17*	*2,83*	*41,2*
Trp	*245,12 ms*											

In order to prove the efficiency of our approach regarding large and complex conditions, we have conducted an experiment to show the scalability of our approach face to three situations:

- **Situation A**: when multiple subjects are requesting different roles and different permissions;
- **Situation B**: when multiple subjects are requesting the same role with different permissions;

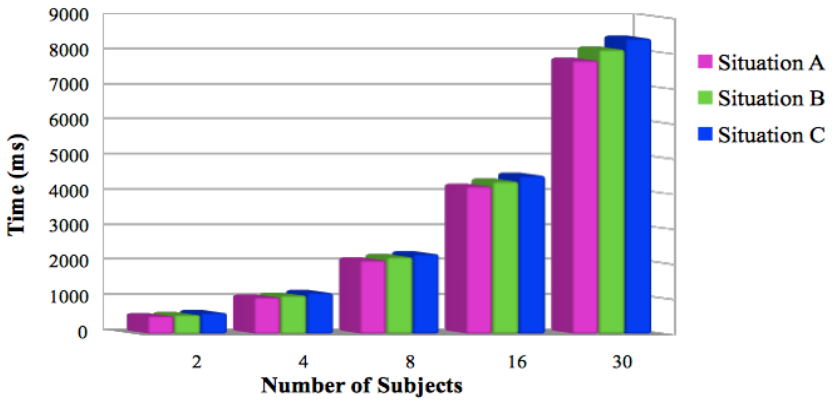

Figure 10. The Scalability Evaluation of the Proposed Approach under the Situations (A, B, C)

- **Situation C**: when multiple subjects are requesting the same role and the same permission;

Figure 10 illustrates the results of this experiment, which clearly attests of the scalability of our approach, in terms of time performance, face to the increase of subjects under the three mentioned situations.

5.2 Security

In order to compute the overhead of the security added in our approach, we develop a baseline experiment, where the role and the permission requested by the subject are verified according to its login and password, and referring to centralized databases. In this baseline test, no PMI Attribute Authority is provided and no interactive and cooperative agents are integrated. Thus, Table 3 presents the measurements that are considered to calculate the overall time cost, named ***Tbt***.

According to Table 2 and Table 3, the security time overhead, noted as D, is calculated as follows:

$D = Trp - Tbt = 245,12 - 175,88 = 69,24\ ms$

Table 3. The Sub-Periods Related to the Process of the Baseline Test

Period	Description	Time (in ms)
Tsr	The time of the requests submitted by the subject	0,46
Tv	The time of verifying the credentials of the subject	6,88
Trd	The time of retrieving role information from the role database	43,57
Tpr	The time of the requests performed within the native platform	1,7
Tra	The time of the role activation	14,39
Tpd	The time of retrieving permission information from the permission database	39,83
Tpa	The time of the permission activation	9,45
Tpe	The time of the permission execution	59,6
Tbt		175,88 ms

This overhead represents 28% of the overall cost of our approach, which appears admissible, credible and not compromising the performances of our agent platform, which benefit from a security feature against any vulnerabilities or damages.

Moreover, in order to evaluate the ability of our approach to detect malicious subjects that attempt to gain unauthorized accesses, we have conducted an experiment based on integrating a set of subjects with masqueraded identities or hacked attribute certificates. The number of these malicious subjects is continuously increased to observe the behavior of the platform in case of burdening. For comparison purposes, the same experiment is performed for the baseline architecture.

Figure 11 illustrates the results of this evaluation, which shows that our approach is more efficient and flexible in depicting more attempts for unauthorized access than the baseline architecture. In general, the detection rate of our approach is higher than the baseline architecture, since among 40 unauthorized subjects, our approach was able to detect the totality while the baseline architecture only detects 27 subjects, and among 100 unauthorized subjects, our approach comes to detect 93 subjects face to 66 subjects detected by the baseline architecture.

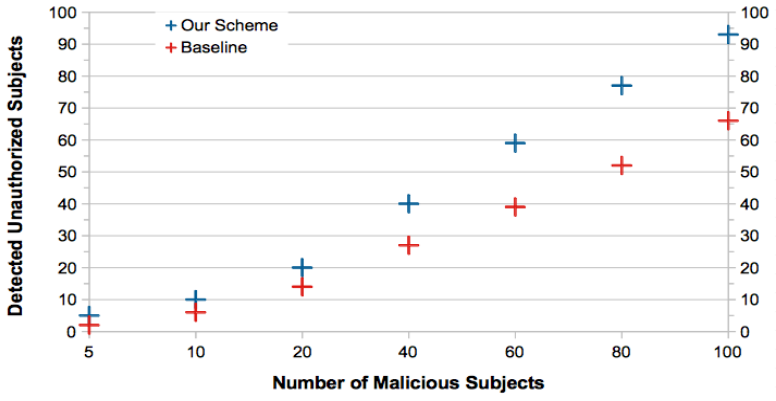

Figure 11. Evaluation of the Unauthorized Access Detection of our Approach Compared to the Baseline Architecture

6. Conclusion

In this paper, we have proposed a novel contribution to secure mobile agent platforms against unauthorized access attacks. Our approach, which is applied on an example of hospital providing tele-medical services, is conceived to dynamically assign roles within an RBAC architecture, where roles and permissions are simulated as cooperative and communicative mobile agents. For that purpose, we make use of attribute certificates issued by a privilege management infrastructure, taking advantage of elliptic curve primitives and referring to a database of role engineering. From the conducted experiments, our approach shows high performances in terms of time and robustness in detecting unauthorized accesses from malicious requesters.

Health care is not the only field concerned by the security of mobile agent platforms. Practical application of this work could be as significant for the new trend of technologies, such as: cloud and grid computing, where mobility, autonomy and interoperability are strongly requested, without compromising the security aspect. As perspectives, it will be interesting to extend the application of this contribution to other models based on roles or other attributes, and integrate temporal and behavioral contexts.

References

[1] D. Gavalas, G. E. Tsekouras, and C. Anagnostopoulos, "A mobile agent platform for distributed network and systems management". Journal of Systems and Software, vol. 82, no. 2, **(2009)**, pp. 355-371.
[2] M. Fasli. "On agent technology for e-commerce: trust, security and legal issues". The Knowledge Engineering Review, vol. 22, no.1, **(2007)**, pp. 3-35,
[3] B. Chen and H. Cheng. "A review of the applications of agent technology in traffic and transportation systems". IEEE Transactions on Intelligent Transport Systems, vol. 11, no. 2, **(2010)**, pp. 485-497.
[4] L. Zhou, A. S. Mohammed and D. Zhang. "Mobile personal information management agent: Supporting natural language interface and application integration". Information Processing and Management, vol. 48, no. 1, **(2012)**, pp. 23-31.
[5] F. Dignum. "Agents for games and simulations". Autonomous Agents and Multi-Agent Systems, vol. 24, no. 2, **(2012)**, pp. 217-220.
[6] M. Metzger and G. Polakow. "A survey on applications of agent technology in industrial process control". IEEE Transactions on Industrial Informatics, vol. 7, no. 4, **(2011)**, pp. 570-581.
[7] J. Ferber, "Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence". Addison-Wesley, **(1999)** Reading, MA.
[8] P. Ahuja and V. Sharma, "A Review on Mobile Agent Security". International Journal of Recent Technology and Engineering (IJRTE). 2, 1, 2277–3878. **(2012)**.
[9] Y. Jung, M. Kim, A. Masoumzadeh and J. B. Joshi. "A survey of security issue in multi-agent systems". Artificial Intelligence Review, vol. 37, no. 3, **(2012)**, pp. 239-260.
[10] V. Arun and K. L. Shunmuganathan. "Secure Sand-box for Mobile Computing Host with Shielded Mobile Agent". Indian Journal of Applied Research, vol. 3, no. 9, **(2013)**, pp.296-297.
[11] T. A. Tsiligiridis. "Security for Mobile Agents: Privileges and State Appraisal Mechanism". Neural Parallel and Scientific Computations, vol. 12, no. 2, **(2004)**, pp. 153-162.
[12] H. Pirzadeh, D. Dub and A. Hamou-Lhadj. "An extended proof-carrying code framework for security enforcement". In Transactions on computational science XI, Springer Berlin Heidelberg, **(2010)**, pp. 249-269.
[13] S. Tuohimaa, M. Laine and V. Leppnen. "Dynamic rights in model-carrying code". In Proceedings of the International Conference on Computer Systems and Technologies, **(2006)**, pp. 1-7.
[14] C. Cao and J. Lu. "Path-history-based Access Control for Mobile Agents". International Journal of Parallel, Emergent and Distributed Systems, vol. 21, no. 3, **(2006)**, pp. 215-225.
[15] W. S. Hsu and J. I. Pan. "Secure mobile agent for telemedicine based on P2P networks". Journal of medical systems, vol. 37, no. 3, **(2013)**, pp. 1-6.
[16] V. Kapoor,V. S. Abraham and R.Singh. "Elliptic curve cryptography". ACM Ubiquity, vol. 9, no. 20, **(2008)**, pp. 20-26.
[17] P. Samarati and S. de Vimercati. "Access Control: Policies, Models, and Mechanisms". In Riccardo Focardi and Roberto Gorrieri, editeurs, Foundations of Security Analysis and Design, LNCS Springer Berlin Heidelberg, vol. 2171, **(2001)**, pp. 137-196.
[18] D. Ferraiolo, D. R. Kuhn and R. Chandramouli." Role-based access control". Artech House. **(2003)**.
[19] B. Jerman-Blazic. "The X. 509 privilege management infrastructure." Security and Privacy in Advanced Networking Technologies, vol.193, **(2004)**, pp. 15.
[20] Q. Dang, S. Santesson, K. Moriarty, D.Brown and T. Polk. "Internet X.509 public key infrastructure: additional algorithms and identifiers for DSA and ECDSA". RFC 5758, **(2010)** January.
[21] D. Johnson, A. Menezes and S. Vanstone. "The elliptic curve digital signature algorithm (ECDSA)". International Journal of Information Security, vol. 1, no. 1, **(2001)**, pp. 36-63.

[22] A. Jaffar and C. J. Martinez."Detail Power Analysis of the SHA-3 Hashing Algorithm Candidates on Xilinx Spartan-3E". International Journal of Computer and Electrical Engineering, vol. 5, no. 4, **(2013)**, pp. 410-413.
[23] F. Bellifemine, A. Poggi and G. Rimassa. "JADE: A FIPA2000-compliant agent development environment", Proceedings of the 5th International Conference on Autonomous Agents. Montreal: ACM, **(2001)**, pp. 216-217.

Authors

Hind IDRISSI, She received her Master's degree in Codes, Cryptography and Information Security from University of Mohammed-V in Rabat, Morocco. She is actually a PhD student in cotutelle between the laboratory of Mathematics, Computing and Applications (LabMIA) in Rabat-Morocco, and the L3I laboratory in La Rochelle-France. Her research interests include: information security, cryptography, multi-agent systems, distributed computing.

Arnaud REVEL is a full professor at the University of La Rochelle, France, and a member of the L3I laboratory. His research interests include: multi-agent systems, robotic, interactivity, learning.

El Mamoun SOUIDI is a full professor at the University of Mohammed-V in Rabat-Morocco, and a member of the laboratory of Mathematics, Computing and Applications (LabMIA). His research interests include: information security, cryptography, error correcting codes.