(Dis)trust Certification Model for Large Access in a Pervasive Environment

RACHID SAADI, JEAN-MARC PIERSON AND LIONEL BRUNIE
LIRIS INSA de Lyon, Lyon, France
Email: {rachid.saadi,jean-marc.pierson,lionel.brunie}@liris.cnrs.fr

Received: October 24 2005

***Abstract*— The challenge of pervasive computing consists in offering access to computing services anywhere and anytime with any devices. However, before it becomes a reality, the problems of access control and authentication have to be solved, among others. Existing solutions are inadequate without adaptation to this specific environment. Among the promising approaches, the trust paradigm seems to be more flexible than others. We base this proposal on this paradigm to implement a distrust model, so-called APC (Access Pass Certificate). The main objective of this model is to enable authorized user to roam and to access trusted sites though they are not known locally. A user can claim two kinds of APCs provided by two kinds of sites: the home site (where the user has an account) and the trusted site (that trusts the user). Using these certificates, the user can progressively extend her access scope. This model implements a decentralized mapping policy, where the correspondence between the user's home profile and her rights in the trusted sites is determined by the trusted site. This distrust model and its implementation are presented in this article where we exhibit its importance for large but controlled access in pervasive environments.**

***Index Terms*— Pervasive environment, security, trust, authentication, access control, certification**

I. INTRODUCTION

Pervasive environments are an innovative field of research which enables large access to information for any user, dynamically, with respect to different constraints of the user's context, such as position, device capabilities and network conditions. The increase in interest for the related technologies (wireless network, light devices) introduces a new security challenge, where the existing security models and mechanisms are inadequate. Among existing solutions, the trust paradigm seems the most flexible approach. Indeed, in the vision of ubiquitous computing [1] especially the pervasive environment, using trust model is the most adequate solution to extend users' access scope.

Classical trust approaches can not be directly adapted to Pervasive Environments, where the unpredictability and unreliability of connections, ownership, locations, context, etc. are the rules, making centralized servers and delegation mechanisms inappropriate. Users may want to access sites where they are not locally known, without bothering to handle complex authentication and authorization schemes. Our approach extends the access scope by using a distrust model which implements a mapping policy. The main objective of our distrust model is to extend user access scope. This access is valuated both by the trust that a site can attribute to other sites in a community and the number of actors building a trust chain. Furthermore, the final decision concerning the access to local resources is decentralized to each site contributing to the trust chain.

This paper is organized as follows. Section 2 presents an overview of existing approaches. Next, in section 3 we introduce our approach that delineates the distrust Model, and show how to implement it in the pervasive environment using a certification model (APC). Then we present our architecture and describe how the APC model is deployed in a site platform in section 4. We illustrate implementation issues of our approach in section 5. Then, in section 6 we discuss benefits and limitations of the approach. Finally, we present the future directions of research and conclude this article.

II. RELATED WORK

There are several projects related to digitally enhanced physical spaces. They provide solutions for particular scenarios and specific types of applications. We identified some relevant works in this domain, and explain how access control is dealt with. The EasyLiving Project [2] at Microsoft Research is concerned with the development of an architecture and technologies for intelligent environments. An intelligent environment is a space that contains myriad devices that work together to provide users access to information and services. CoolTown [3] offers a web model for supporting nomadic users, based on the convergence of web technology, wireless networks and portable devices, with capabilities for contextual authentication and, more specifically, location-aware authentication. In the Centaurus project [4] the goal is the development of a framework for buildings portals to services using various types of mobile devices. It uses a simplified public key infrastructure where all the clients and services managers have a public/private key pair to authenticate themselves.

All these models give users an access to surrounding services, resources, etc. Their weakness is characterized by their predictability: users are local and known or they are close by. Conversely, our approach implements a trust model to extend the access scope of each user to sites where she is not locally known.

In the last decades, trust models such as [5], [6], [7] have defined "trust" as a fundamental aspect for an inter-domain relationship.

To assure security in a ubiquitous computing environment such as pervasive or ad-hoc, a trust model using a certification mechanism (e.g. PGP Pretty good Privacy [8], SPKI [9], [10], [11], [12]) can be implemented.

PGP (Pretty Good Privacy) uses a combination of public-key infrastructure to provide a security based on the trust that gives to the signatory. The largest problem associated with PGP is the key distribution and management.

The SPKI Working Group has developed a standard for digital certificates whose main purpose is authorization rather than authentication. This is the first proposed standard for distributed trust management including a rudimentary notion of delegation.

The access control mechanism Sygn provides a decentralized permission storage and management system in Grid Computing. All permissions in Sygn are encoded in certificates, which are stored by their owners and used when required. Permissions can be created on demand, by the owners of the resources or by administrators to whom this responsibility has been delegated. Sygn allows an efficient decentralized administration of dynamically changing resources and permissions.

Actually, the delegation is considered as a main aspect of Trust in distributed environment. Some certification models use a system of delegation, The delegation gives to users an access for example, through other authorized users [11], [12], or authorized agents [13] etc. This certificate contains the rights of user. The delegation itself is viewed as a right. Only the users with rights to delegate an action can actually delegate that action, and the ability to delegate can itself be delegated.

The delegation mechanism is considered to be efficient, but not sufficient alone to perform a broad access. Indeed, the user's scope is restricted to environments where she is locally known. Consequently, she has an access if there is at least one entity that trusts her.

Therefore, to enhance the access scope, a system based on a "Mapping Policy" is implemented for instance in Grid Computing [15]. Pearlman *et al.* [14] define a virtual Organisation VO that groups some users and resources, e.g., hosts or storage space. It relies on a centralized server CAS. This server is used to map or convert the local user access to a VO access. Applying this mapping concept seems to be inadequate in the pervasive environment. Indeed, the user must have an account in the virtual organisation to obtain a VO certificate, and this certificate does not allow the user to access another VO, but only this one. Moreover, using a centralized server is seen as a drawback.

The concept of mapping is fulfilled between the users' profiles depending on the local access policy. Between different access policies, three models are identified: Discretionary access control (DAC), Mandatory access Control (MAC) and Role based access control (RBAC). In DAC [16], all permission can be represented by an access matrix, where each row corresponds to a user and each column to a resource. The problem of this model is the difficulties of management, because in pervasive environment, it is not able to put all access control list of a user in a mobile device. MAC [17] typically deals with data resources, all of them are assigned a label according to a classification, typically security levels like: top secret, secret, confidential, unclassified. RBAC [18] is based on the concept of roles. A role represents a named collection of permissions. In this manner, users are assigned roles according to the tasks they have to perform.

A distributed system is typically constructed out of different sites using different access policies. For this reason, we define the profile to represent a generic aspect of the user's access: It represents the user's rights, and can be a role or a level depending on the target site policy.

On the other hand, to evaluate a trust relationship, various systems are proposed. We illustrated here the most significant ones: Almost all trust models graduate the trust. In [5] the trust value is standardized by using a scale, which is bounded between -1 and 4. This scale valuates the trust with values between the distrust (-1) to the complete trust (4). Another approach [19] defines a trust cloud to describe the uncertainly of the trust relation. A trust cloud is a normal cloud which describes by its behaviour the uncertainly (deterministic or fuzzy) of the trust evaluation between two entities. PTM [20] exploits the cooperation among entities to increase or decrease the relationship degree: actually, the trust value changes according to the entity behaviours by providing feedback about entity performances during their interactions.

All these approaches compute trust value resulting from a trust chain as an average value. We believe that this evaluation not sufficient: For instance, it does not take into account some constraints such as the length of the trust chain.

In the rest of this paper we propose a model of certification using a decentralized mapping policy between the different organisations to extend the access scope of users. This model is based on distrust evaluation for relationship between organisations. This distrust evolution has, as strong constraint, the geographic propagation of the trust.

III. A Vision of the Pervasive Environment

The objective of our research is to extend the access scope for each user inside different sites. These sites are organization, host or domain like universities, restaurants, post offices, airports etc. The challenge is to allow each mobile user to roam and access inside this environment easily and transparently, by exceeding a certain numbers of barriers like the heterogeneity of the different access policies.

Let's consider the following use case, as illustrated in the Fig. 1. We have Pr Bob; he is member of University A. This Professor goes to a conference in University B and goes to meeting in University C. He communicates with the different surrounding "objects" (students, professors and resources: Printer, video projector, etc.). In fact, Bob owns a professional card or conference badge that defines his status and contains a picture or fingerprint to identify its holder. This card or badge allows Bob an access inside these universities according to a convention or shared collaboration (the same work group). These Universities do not know the owner of the card, but trust hir cards.

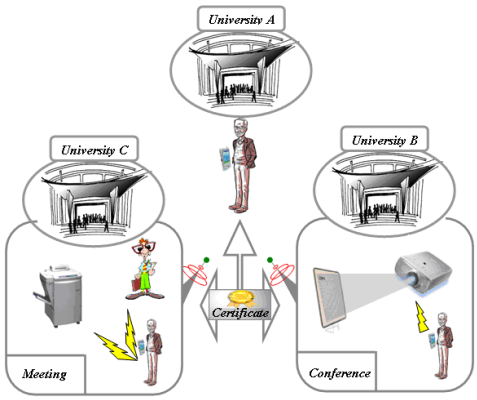

Fig. 1. Pervasive scenario.

If we map this scenario in the pervasive environment, universities correspond to sites. A certificate simulates the professional card and the fingerprint or the picture is replaced by an authentication system embedded in the certificate. In this manner, if Bob has the right to access a conference, according to his certificate, he obtains a new temporary certificate (like a badge in a conference). This certificate allows Bob to communicate inside this new site like all members.

In this paper we use the following terms:

- **Site** represents an organization, domain or host that implements a local security policy and is limited geographically,
- **Profile** each user has a profile, depending on the access policies; according to the access policy, it can represent a role (student, doctor) or an access level (distrust, confidential...), etc.
  Certificate it represents a digital identity of the users. She has some certificates (like professional cards) that prove her membership to organizations.

IV. Proposition

Our solution allows the user to authenticate on remote sites and to assign access in this environment without being locally known.

We propose an architecture based on a distrust model and a certificate system.

A. The Distrust Model

Let S denote a set of sites.

Definition of the Trust Relation
Let A and B two sites, $A \in S, B \in S$. If A trusts B then we say that the relation Trust is verified between A and B and we note ***A Trust B*** A is called the trustor and B the trustee.

Properties of trust relation
Reflexivity $\forall A \in S, A\ Trust\ A$
Trivially, a site trusts itself.

Non-Symmetry The Trust relation is not symmetric. Indeed, a site is fully responsible for its trust policy and there is no obligation of reciprocity, so we can get
$A\ Trust\ B \wedge \neg A\ Trust\ B$

Transitivity The Trust relation is transitive:
$\forall A, B, C \in S, A\ Trust\ B \wedge B\ Trust\ C \Rightarrow A\ Trust\ C$

This property is fundamental for the effectiveness of our proposition. It allows defining "trust chains" between sites that do not know each other (see below).

Based on the *Trust* relation, we introduce *the distrust function* t^0, to estimate the level of (dis)trust between two sites.

Definition: distrust function We call the distrust function and we note t^0, the function defined as:

$$t^0 : S * S \rightarrow N \quad \text{S:set of sites}$$
$$(A, B) \rightarrow d \quad \text{N: set of natural numbers}$$

$$t^0(A,B) = \begin{cases} -1 & if \neg(A\ Trust\ B) \\ 0 \leq d \leq T_A^0 & otherwise \end{cases}$$

where d represents *the distrust degree* and T_A^0 denotes the distrust threshold of the site A.

This function quantifies the degree of distrust that the site A shows wrt the site B. When $t^0(A,B)$ increases, the distrust increases (i.e. the trust decreases). As a consequence:

- $t^0(A,B) = 0$: any site has a complete trust in itself.
- $t^0(A,B) < t^0(A,C)$: means that the trustor A has a higher trust in B than in C.

The distrust threshold represents the maximum level of distrust beyond which A does not trust B (i.e. the relation A Trust B is not verified).

A feature of the distrust function is the use of the value -1 to denote the fact that a site does not trust another site. Indeed, as the distrust degree can range *a priori* from 0 to any positive number, there is not a priori superior limit value. Consequently, it is necessary to introduce and use a symbolic value to state that a site does not trust another one. We could have chosen ∞ or $\perp$ but for easiness of computing reasons, -1 is more convenient.

The distrust function shows properties related to the properties of the Trust relation.

Properties of distrust function
Self trust $\forall A \in S, t^0(A,A) = 0$

Non-commutativity
$\exists A, B \in S / t^0(A,B) = d_1 \wedge t^0(B,A) = d_2 \wedge d_1 \neq d_2$

Composition
Definition: Composition of distrust degrees
Let A, B, C 3 sites. The *composition* of the distrust degrees $t^0(A,B)$ and $t^0(B,C)$, noted $t^0(A,B) \oplus t^0(B,C)$ is defined as:

$$\begin{matrix} t^0(A,B) \\ \oplus \\ t^0(B,C) \end{matrix} = \begin{cases} -1 & if(t^0(A,B) \vee t^0(B,C)) = -1 \\ t^0(A,B) \\ + \\ t^0(B,C) & otherwise \end{cases}$$

Generalization: trust chains

The composition of distrust degrees is generalized to n sites by composing two by two the distrust degrees:
$t^0(A1, ..., An) = t^0(A_1, A_2) \oplus ... \oplus t^0(A_{n-1}, A_n)$
$(A_1, ..., A_n)$ is called a trust chain.

Notation: Distrust propagation function
Let A and C 2 sites of S; let B_1 B_n n sites of S. Let us note T = $(B_1, ..., B_0)$. We note $P_T^0(A, C)$ and we call distrust propagation degree between A and C based on T the value:
$P_T^0(A, C) = t^0(A, B_1, ..., B_n, C)$.

Property
$P_\phi^0(A, C) = t^0(A, C)$

Theorem
$P_\phi^0(A, C) = -1 \Leftrightarrow$
$.....\exists F, G \in (A, B_1, ..., B_n, C)/t^0(F, G) = -1.$

Proof Trivial by application of the definition of t^0: the composition of distrust degrees equals -1 if and only if one at least of the distrust degrees equals -1.

When several chains exist between two sites, the trustor site chooses a reference chain according to its local policy (for instance the chain that returns the maximum or the minimum distrust propagation degree).

B. The Local Policy

1) The access policy: This distrust model allows one to construct a *Trust graph*, noted $T_g(S, E)$.

Definition: Trust graph
Let S be a set of sites. Let E the set of trust relations verified over the nodes of S. We call Trust graph, and we note $T_g(S, E)$ a valued and directed graph such that:

- The nodes of the graph represent the sites of S.
- Each Trust relation between two sites is represented by a directed edge e. The set of edges is consequently identified with the set of relations, E.
- Each edge is valued by the distrust degree between the sites represented by the source and destination nodes of this edge (use of the t^0 function).

This trust graph is used to build trust sets.

Definitions: Trust-In set / Trust-Out set / Trusted site
Let A be a site of S. We note:

$$TS_I(A) = \{X \in S/(X\ Trust\ A)\}$$

$TS_I(A)$ represents the set of the sites that trust A. It is called the ***Trust-In set of A*** (see Fig. 2).

$$TS_O(A) = \{X \in S/(A\ Trust\ X)\}$$

$TS_O(A)$ represents the set of sites that A trusts. It is called ***the Trust-Out set of A*** (see Fig. 2). The sites of the Trust-Out set are called *Trustees* or *trusted sites*.

This distrust model is used to decide if a "foreign" user can be allowed to access to a protected site (i.e. to decide if a user who does not own an account of the system can get logged in the system).

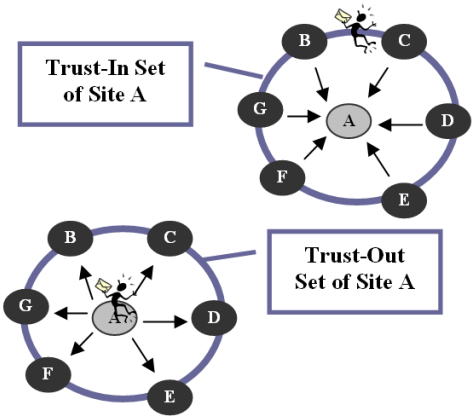

Fig. 2. Trust sets.

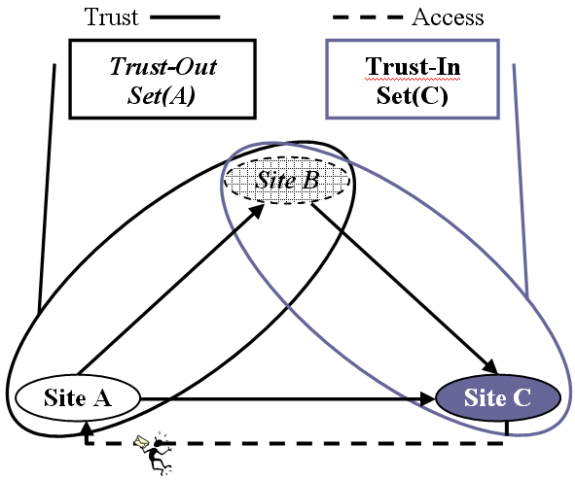

Fig. 3. The trust propagation.

We will consider two different types of access: direct access and transitive access.

Definitions: Home sites / Direct and Transitive access

- *A user's home site* is a site on which the user is registered as an authorized user (i.e. on which she/he has an account and she/he can authenticate her/himself).
- *A direct access* is provided by a trustor to all users registered by its trustees. This direct access is valued by the distrust degree between the trustor and the trustee.
- *A Transitive access* can be provided by a trustor to a user who does not belong to its trustees on condition that it exists a (positive) distrust chain between one of the user's home sites and the trustor (see figure 3). This transitive access is valued by the distrust propagation degree between these two sites (as before, in case of the existence of several possible chains, the trustor is responsible for choosing the reference chain).

To manage the users' access, each site has to define thresholds beyond which access is not allowed.

We distinguish three thresholds:

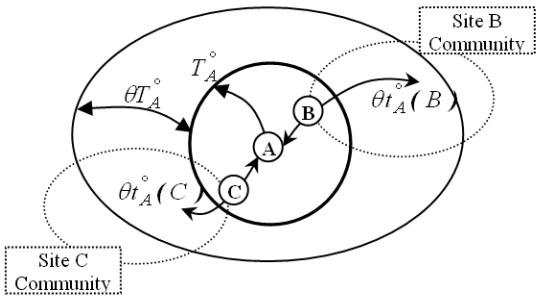

Fig. 4. The Threshold Access.

- Distrust threshold corresponding to a direct access.
- Global and Maximum thresholds corresponding to a transitive access.

Distrust threshold Each trustor A can define a specific *distrust threshold* T_A^0 for each of its trusted sites.

Global (distrust) threshold Each trustor A has to define a *global threshold* θT_A^0, corresponding to the maximum tolerated degree for a transitive access. This value is proportional to the distrust threshold and to the maximum authorized chain length L_A starting from A.

Maximum (distrust) threshold A trustee X is connected to a community of trusted sites (Trust-Out set of X). When X is used as intermediary between a foreign site F and a trustor A (i.e. when X is at the end of a trust chain from F to A), this latter can decide what trust threshold to apply to decide if it can allow F enter the system. This threshold may depend on the community of X (i.e. A can trust X but nor the sites trusted by X).

Definition: Trust coefficient / community
A trustor A attributes for each trustee site X a *trust coefficient* $\alpha_A(X)$ that corresponds to the trust A has in the trustees of X (called the community of X). This coefficient ranges between 0 and 1.

Definition: Maximum (distrust) threshold
Let A a set and X Trust-Out-Set(A) (X is a trustee of A). The *Maximum threshold* between A and the trustees of X, noted $\theta t_A^0(X)$, is defined as (see Fig. 4):
$\theta t_A^0(X) = T_A^0 + \theta T_A^0 * \alpha_A(X)$

Property
$T_A^0 \leq \theta t_A^0(X) \leq T_A^0 + \theta T_A^0$

This distrust model is decentralized. Each site can evaluate its distrust threshold differently from other sites. This can lead to a divergence in the evolution of the transitive access. For example: one trustor can value its trustees up to 20 and another can value its own trustees up to 500. To smooth these differences, the Composition of distrust degrees is redefined to evaluate, for each site, its distrust degree relatively to its distrust threshold as follows:

TABLE I
THE TRSUT-OUT TABLE

ID	Identifier of the trusted site
Type	Type of the trusted site (close or distant)
t^0	Distrust degree
T^0	Distrust threshold
θt^0	Maximum threshold

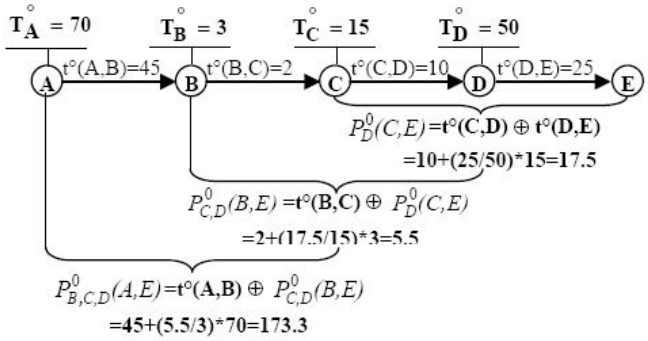

Fig. 5. Compute a Trust Chain.

Redefinition
Let A, B, C be three sites. The composition of the distrust degrees $t^0(A,B)$ and $t^0(B,C)$, noted $t^0(A,B) \oplus t^0(B,C)$ is redefined as:

$$\begin{matrix} t^0(A,B) \\ \oplus \\ t^0(B,C) \end{matrix} = \begin{cases} -1 & if(t^0(A,B) \vee t^0(B,C)) = -1 \\ \left[\begin{matrix} t^0(A,B) \\ + \\ \frac{t^0(B,C)}{T_B^0} * T_A^0 \end{matrix} \right] & otherwise \end{cases}$$

Consequently, a site A can allow a foreign user U (registered in C) an access through an intermediary trusted site B:

$$\text{iif } 0 \leq P_B^0(A,C) \leq \theta t_A^0(B)$$

To implement this model, each trustor maintains a trust-out table that contains the following information about the trusted sites (see Table 1).

Example
Let five sites that build a trust chain (A, B, C, D, E). Suppose a user of the site E wants to access to the site A. To decide if this user can be granted an access, A computes $P_{B,C,D}^0(A,E)$ progressively (see Fig. 5).

If $\alpha_A(B) = 0.5 \wedge P_A = 3$

$$\text{then } \theta t_A^0(B) = 70 + (70 + 0 * 3) * 0.5 = 175$$

As a consequence, the user of the site E will be allowed to access to the site A since:

$$0 \leq (P_{B,C,D}^0(A,E) = 173.3) \leq (\theta t_A^0(B) = 175)$$

2) The Mapping Policy: When a user is allowed to access a site X, it attributes to her/him a new profile using a mapping policy. This profile defines the user's right inside the site X. This profile is called a A-profile (analogue profile).

TABLE II
APC HEADER

IDC: ID Certificate	Identifier which permits to identify the APC in Site A.
Type of certificate	Identifies if it is an APC-M or APC-T
IDS : ID of Site	Identifier which permits to identify Site A.
IDO : ID of Owner	Identifier which permits to identify the APC's holder.
Lifespan	This field defines the validity period.

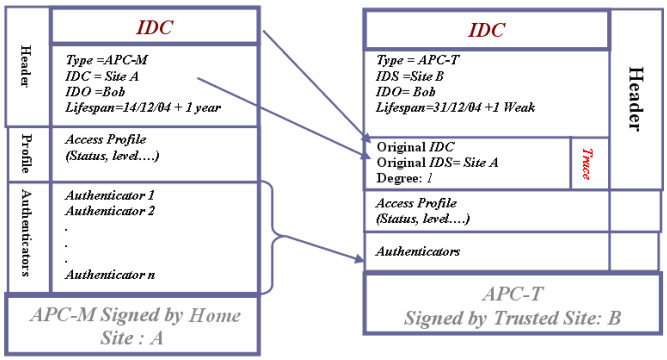

Fig. 6. The APC Certificate.

A mapping policy can be simply implemented by creating a mapping table to store the correspondence between foreign profiles of trusted sites and its analogue profiles.

C. The Certificate Model: (APC: Access Pass Certificate)

To implement the distrust model we use a model of certificate called APC: Access Pass Certificate. This "APC" works as a pass, allowing its owner to roam between different sites.

The APC testifies the user's profile (status or access level) in a Home or Trusted site. If the user wants to access a particular site, she uses one of her certificates that is recognized by the target site. Using this APC, the local site applies its mapping policy between the local policy and the policy held in the APC.

An APC could be obtained in two different ways:

- Each site gives a Home Certificate or APC-M, to all its members.
- Each site gives a Trust certificate or APC-T, to a guest, when it trusts in the user's Home Site, or when the user presents an APC-T from one of its trustees.

1) APC-M: Each site delivers to all its members an APC-M that denotes members' profile. The target site authenticates the user and attributes him a corresponding analogous profile. The APC contains three parts: the header, the profile and the authenticators, and like all certificates, it is signed with the private key of a Home site (see Fig. 6)

The header Identifies the certificate, and is composed of the fields depicted in Table 2.

The profile It is a variable part of a certificate, depending on the site's policy. This part contains information about user's profile, such as status or access level in a Home/Trusted Site (certifying site). The use of this profile is original. Indeed, unlike other systems of certification that certify an access to particular resources, this one certifies the profile that represents all authorized accesses to site's resources.

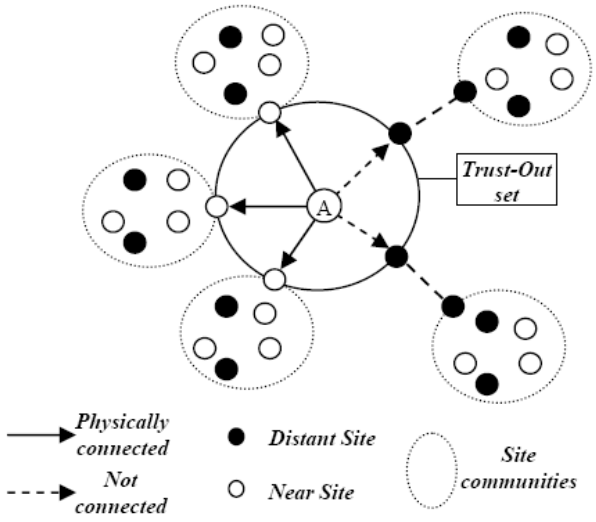

Fig. 7. The community connection.

Authenticators This part permits one to identify the owner of APC. Authenticators are numerous, and related to the variety of devices used in the pervasive environment (PDA, mobile phone, terminals). Facilitating certificates management could be fulfilled by embedding some authenticators according to device authentication capabilities and the site's security policy.

2) APC-T: The trust-out set of each site A contains two kinds of trusted sites (see Fig. 7):

- *Near trusted site:* it represents the trusted sites that can communicate with A (directly or indirectly).
- *Distant Trusted site:* represents the trusted sites that cannot communicate with A.

If a site does not recognize a user, it asks the near trusted sites about the Home site of this user, which forwards the request if they are unable to answer. This way, each user obtains a direct access (if she belongs to a trusted site) or a transitive access (if she belongs to communities of the near trusted sites). Consequently, if the user belongs to communities of distant trusted sites, she can't obtain any access.

For this reason, each user can obtain an access (in another manner) by presenting a trust APC (APC-T). Therefore each site delivers an APC-T if it trusts the user's APC. Thus, the users use this one to roam inside the distance sites.

In general, the APC-T is obtained from a main APC that is used to authenticate user on the target site. The APC-T has a short period of validity, e.g. two or three days, as compared to APC-M which are valid as long as they are not revoked explicitly by its issuer.

As illustrated in the Fig. 6, the header of the trust certificate contains two parts:

- *The standard part* that identifies the APC-T.
- *The Trace parts* that contain information about the original certificate (IDC, IDS) and the propagation degree

(degree) that corresponds to the distrust degree between the home site and the local site.

The profile part contains the new analogous profile provided by the target site to the user, as outcome of the mapping policy. Concerning the authenticator part, the trust certificate embeds the authenticators of the original APC.

D. APC Authentication system

The certificate is a proof of the owner's status inside a community, but the problem is: How can the user assure that she owns the APC? Our approach embeds in APC a number of authenticators using, for example: a password authentication or a public key mechanism. Two ways of authentication have been identified, remote and local authentication.

1) Remote authentication: A challenge response authentication is the most important system of authentication; it uses a key-pair mechanism. The APC embeds one or several keys depending on the user's devices capabilities (storage for instance).

2) Local authentication: This authentication is represented mostly by a hash function. This is a one-way function, meaning that it does not permit one to retrieve the original information from the hash. Like a credit card system, the hash result (the residual) is calculated from ciphering several parameters (credit card) using a password. This residual does not permit one to retrieve the password nor the parameters. Therefore it can be embedded safely in APC.

V. Our Architecture

The proposed approach detailed in section 3 is mapped in the architecture depicted herein. This architecture is composed of Databases, Modules and Protocols (see Fig. 8).

A. Databases

Our approach uses three databases:

Site DB contains information about the trusted sites. This DB is composed of two tables:

- *Trust-in Table* contains all information about sites that belong to the trust-in set.
- *Trust-out Table* contains all information about sites that belong to the trust-out set.

APC DB gathers all required information about APC. This DB contains two tables:

- *APC table* each tuple represents an APC that is created by the local site, and contains relationship between this APC and its owner.
- *Revocation table* contains the information concerning APCs that have been revoked by the local site or the Trust-out set sites.
- *Mapping DB* contains information about the Mapping Process: The mapping profile, as well as the traceability along the mapping process.

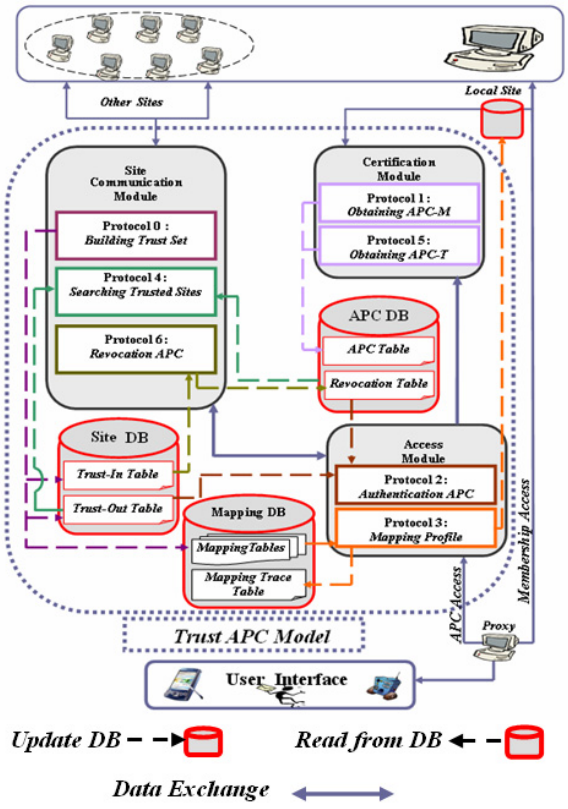

Fig. 8. The Trust APC Architecture.

B. The Modules

To distinguish between the different functionalities of the APC Model, we group all protocols in three modules:

The Site Communication Module groups three communication protocols, used exclusively between sites, for:

- Building trust sets
- Searching trusted sites
- Revoking users.

The Certification Module contains two protocols used between the user and the site. This module allows the user to claim an APC-M or APC-T.

The Access Module groups two protocols, which permit the user to authenticate and obtain access in some site using an APC model.

C. Protocols

The APC Model implements seven communications Protocols, which are used principally

- To create and maintain the APC Model.
- To perform the APC access.

Protocol 0: Building Trust Sets

This is a basic protocol, it represents two phases:

Public Key exchange: if the different sites want to communicate, they must exchange their site's public key (Pub). This key identifies a site and enables to use the certification system. The exchange process depends on the set's policy such as:

TABLE III
NOTATION USED IN THE FOLLOWING

AC	Authority of certification
Pub_x	Represent a public key of x
ID_x	Identifier of site/user : x
LS	Local Site
PS	Previous site in the trust chain
Id_x	Identifier of packet : x
$< m > Sig(x)$	m is signed with x's private key
A-Profile	Analogue profile
Path(A,B)	All sites in the distrust chain between A and B
TTL	Time to live of packet

- Use of an authority of certification (AC) (X509 model) [21]
- Phone, Fax, Email, etc.
- Other trusted exchange systems.

Mapping: the second phase develops a correspondence between a local policy and all the trusted sites' policies. Therefore the local site can build a Mapping Table and gives a local profile for each trusted guest.

When A wants to communicate with B (A and B are neighbour sites), there are two scenarios:

1. A trusts B: A gives an access to users from B.

1) A sends Trust Request(TR) Packet to B :
   $TR =< ID_A, Type = Trust, (Pub_A)Sig(AC) > Sig(A)$
   - If B accepts this request
     - B updates Trust-In Table: $< ID_A, Pub_A >$.
     - B sends 2 packets to A:
       Acceptance response :
       $TR =< ID_B, Type = Trust\_Ok, (Pub_B)Sig(AC) > Sig(B)$.
       Shared Profiles :
       $TR =< ID_B, Type = Prof, Profiles > Sig(B)$.
2) A updates Trust-Out Table $< ID_B, Pub_B >$.
3) A gives a corresponding profile to A's profiles by applying the mapping's policy and updates the Mapping DB.

2. A is trusted by B: A permits its users to access B.

1) A sends to B
   $TR =< ID_A, Type = Trust - In, (Pub_A)Sig(AC) > Sig(A)$
2) If B accepts this request
   - B updates Trust-Out Table $< ID_A, Pub_A >$.
   - B sends an acceptance response to A:
     $TR =< ID_B, Type = Trust_Ok, (Pub_B)AC > Sig(B)$
3) A updates Trust-In Table $< ID_B, Pub_B >$
   A sends its shared profiles to B:
   $TR =< ID_A, Type = Prof, Profiles > Sig(A)$.
4) B gives a corresponding profile to A's profiles by applying the mapping policy and updates the Mapping DB

Protocol 1: Obtaining APC-M

Each user accesses its Home site, and this protocol permits her to obtain certificate (APC-M), which certifies her access profile (see Fig. 9.A1).

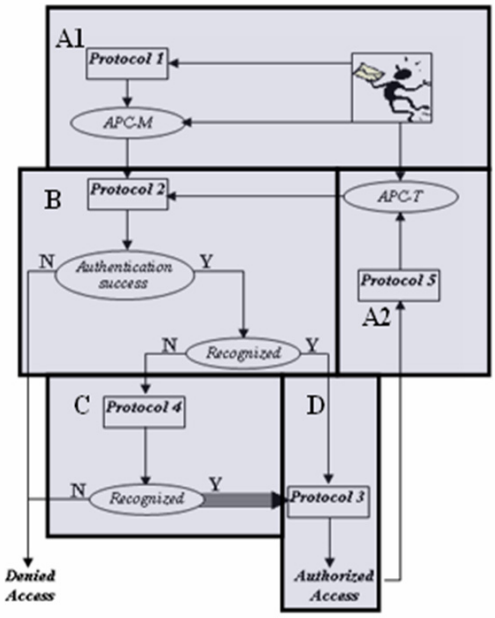

Fig. 9. Schema for performing an APC access.

1) User Bob accesses his Home site A as usually, for example: using Username and Password.
2) Bob claims an APC-M
3) According to Bob's devices, A chooses the appropriate authenticators.
4) A creates the APC-M to Bob:
   {The header
   $< IDC >$
   $< Type = APC\_M, IDS = Site_A, IDO = Bob, Lifespan >$
   Bob Profile
   Authenticators $< Auth_1, Auth_n >$
   } Sig(A)

When the user has an APC, she can access the target site using Protocol 2 (see Fig. 9.B).

Protocol 2: Authentication APC

This protocol permits the user Bob to authenticate himself and allow access to the target site B in two steps:

- *The authentication step:* Bob must authenticate the APC to attest that this certificate has not been altered. Then, she chooses one of the authentications that are embedded in this APC, which is enabled by the local policy of the target site.
- *The recognize step:* if the authentication succeeds, B checks if the certifying site APC.IDS belongs to its "Trust-out Set".

1) Bob sends his APC
2) Site B chooses from Bob's APC only the trust authenticators according to B's security policy and Bob's devices.
3) Bob chooses one from the list of authenticators, and proceeds to authentication.
4) B interrogates Site DB if the APC.IDS exists in "Trust-out Table"
5) B checks if:
   $0 \leq (t^0(B, APC.IDS) \oplus APC.degree) \leq \theta t_B^0(APC.IDS)$

Then execute the Protocol 3
Else Refuse the access to Bob ***(Distrust user)***

If the user was authenticated and recognized by the local site as a trusted user, the local policy will give him an analogous profile using the Protocol 3 (see Fig. 9.D).

Protocol 3: Mapping Profile

When the user Bob was authenticated and recognized, the Site B applies the mapping policies and gives an analogous profile (A-Profile) to Bob's profile, which is denoted by the last APC (APC.Profile).

Protocol 3-a: Searches and creates an analogue profile A-Profile

1) Site B checks in APC DB if the APC is not revoked.
   If the APC is revoked refuse the access to Bob (***Revoked user)***

   Else
2) Site B checks in Mapping DB if this profile exists **If** the profile does not exist refuse the access to user ***(Unknown user)***

   Else gives an analogous profile to Bob (A-Prof)

Protocol 3-b: B creates a short life access with the new profile.

1) Updates the Mapping trace table
   $< APC.IDC, APC.IDS, APC.IDO, APC.Profile, A-Prof >$
2) Creates a short life account (two days, or one week)
   Update local host DB
   $< Username, Password, A-Prof, Validity >$

Protocol 4: Search trusted Site

If the user is authenticated, but not recognized, the system executes the protocol 4 to extend and facilitate user access without continually soliciting the user's device. Local site asks surrounding sites about this guest (see Fig. 9.C).

---

1) Creates the Recognizing Request Packet (RR)
   $< Id_{RR}, Type = SS, ID_{LS}, APC.IDC, APC.IDS, ------$
   $----------\triangleright APC.Profile, APC.degree, TTL > Sig(LS)$
   Id_{RR}= identifier of RR packets, SS= Searching Site,
   ID_{LS}: ID of Local Site.

2) Local Site multicasts RR packet to all trusted sites in the trust-out set.

3) If the trusted site receives RR packet type SS, and does not recognize APC.IDS and TTL<>0 :
   a- Save in a temporary Trace Table (TT)the trace of this packet
   $< Id = RR.Id_{RR}, ID_S = RR.ID_{LS} >$
   b- Modify and Propagate the RR Packet to all near trusted sites
   $< Id_{RR}, Type = SS, ID_{LS}, APC.IDC, APC.IDS, ------$
   $----------\triangleright APC.Profile, APC.degree, TTL-1 > Sig(LS)$

4) If the trusted site receives again an already received RR packet, it destroys it.

5) If the trusted site receives RR packet type SS, and recognizes APC.IDS :
   If the distrust degree = -1 ***(Distrust user)*** then
   Creates a new RR Packet and returns it to the previous site as follow:
   $< Id_{RR}, Type = SF_E, Path = ID_{LS}, degree = -1 > Sig(LS)$
   SF_E = Site Found Error (distrust degree=-1)

   Else
   Executes Protocol 3-a with APC.Profile to recover the Analogue Profile (A-Profile)

   If the APC is revoked ***(Revoked user)***
   Creates a new RR Packet and returns it to the previous site as follow:
   $< Id_{RR}, Type = SF_R, Path = ID_{LS}, degree = -1 > Sig(LS)$
   SF_R= Site Found Revoke

   Else If the A-Profile has not been found: ***(Unknown user)***
   Goto step 3-a (to ask near trusted sites about this unknown APC.Profile)

   Else Creates the new RR Packet and returns it to the previous site as follow:
   $< Id_R R, Type = SF_{OK}, Path = ID_{LS}, A-Profile, degree = t^0(LS, APC.IDS) \oplus APC.degree > Sig(LS)$
   SF_{OK} = Site Found OK

6) If the trusted site receives RR packet of type SF, it modifies this packet and returns it to the site corresponding to the Id_{RR} saved in the TT Table
   If $(SF = SF_R/SF_E)$ then
   $RR =< Id_{RR}, Type = SF_R/SF_E, Path = Path+ID_{LS} ------$
   $----------\triangleright, Degree = -1 > Sig(LS)$
   Else
   $RR =< Id_{RR}, Type = SF_{OK}, Path = Path + ID_{LS} -----$
   $----\triangleright, A-Profile, degree = t^0(LS, PS) \oplus degree > Sig(LS)$

7) If the packet's author receives his packet (type SF),
   If $0 \leq t^0(B, PS) \oplus RR.degree \leq \theta t^0_{LS}(PS)$ then
   it executes the protocol 3 and updates the mapping trace table
   Else Refuse the access to Bob ***(Distrust user)***

When the user is authenticated and recognized, she accesses the trusted site following the analogous profile, and can now claim an APC-T using the Protocol 5 (see Fig. 9.A2).

Protocol 5: Obtaining APC-T

---

1) The user Bob accesses target site B using one of his APC: Cert.
2) Bob claims an APC-T
3) Following the local security policy, B proposes to Bob only the trusted authenticators from the original APC.
4) B Create the APC-T:
   $\{Header$
   $< IDC >$
   $< Type = APC-T, IDS = ID_B, IDO = ID_Bob, Lifespan >$
   $Trace = Cert.trace + < Cert.IDC@path(Cert.IDS, B) >$
   $Degree = t^0(B, Cert.IDS) \oplus Cert.degree$
   $Analogue\ Bob\ Profile\ :\ A-Profile.$
   $Authenticators < Auth_1, Auth_p >$
   $\}Sig(B)$

Protocol 6: Revocation APC

In our architecture, a site can deny access to a particular user (for incompetence or illegal access). Like all other certification systems, there exists a revocation protocol: Protocol 6. This one characterizes the exchange of periodic information between all the trusted sites within a community.

---

1) Site A creates the Revocation Request Packet "VR"
   $VR =< ID_A, APC.IDC, APC.Lifespan > Sig(A)$
2) Local site multicasts VR packet to all near trusted site in the trust-In set.
3) When a trusted site receives the VR Packet, it updates its Revocation Table $< ID_A, APC.IDC, APC.Lifespan >$

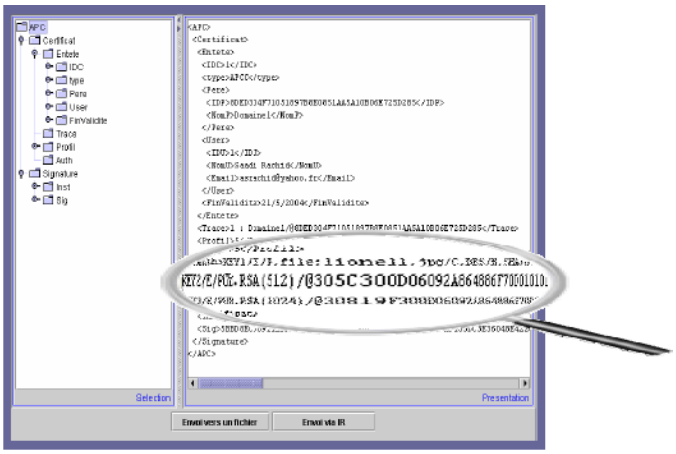

Fig. 10. The XML representation of APC.

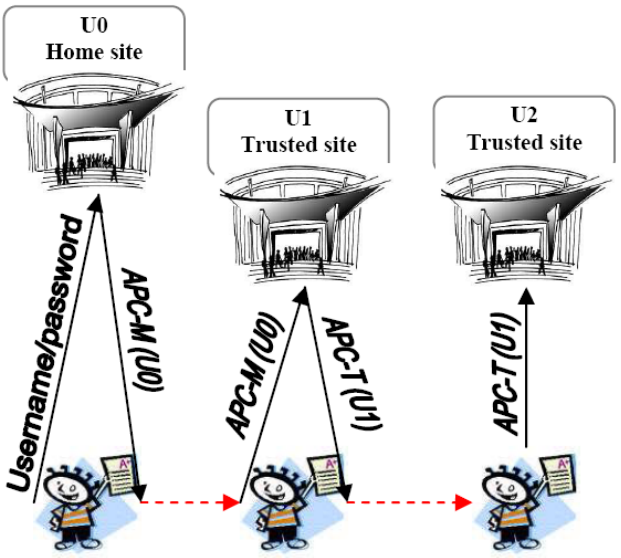

Fig. 11. The demonstrator.

VI. Implementation

A demonstrator has been implemented to illustrate the use of the APC architecture. APC are stored as XML representations (see Fig. 10).

This demonstrator allows the user to roam inside three universities, her Home university (using Username and Password), and two other universities (using APC-M and APC-T).

The user enters her Home university U0 and claims an APC-M. She uses this APC-M and accesses university U1, who trusts U0. When the user is allowed to access U1, she can claim an APC-T. Finally, this APC-T provide user with an access to U2, thanks to the trust that is given by U2 to U1 (see Fig. 11).

The generated APC embeds three authentications: Two remote (Public keys 512 and 1024) and one local (using an Infrared connection with a mobile phone)

The authentication system uses the challenge response mechanism for remote authentication. Each user is authenticated by signing the challenge with corresponding private key to one of public keys in the APC. However the local authentication is fulfilled in the following process: The user captures a picture with her mobile phone, then sends it trough infrared connexion. Afterwards, she attached a password to this picture. Finally, the site embeds the hash function generated by this authentication as an authenticator. In the same way, when the user wants to authenticate her certificate in the trusted site, she sends, by infrared connection, the photo and introduces an associated password to authenticate it.

VII. Discussion

Our approach allows the user, to roam transparently in the pervasive environment simply by using her APCs. The APC model presents a number of advantages: It is decentralised, since each site, having knowledge only about its neighbors can perform a large but controlled access to some user's communities; it reduces the human interaction since most security management functions can be processed dynamically; it does not modify the local site's policy; and finally, having a unique certificate it can manage many devices.

All protocols are automated except protocol 0 because it is subject to constraint. This constraint is illustrated mainly by difficulties arisen while managing relationship among organizations (sites) and applying the mapping policies. In fact, an organization, having a trust relationship with other organizations, must validate and value relations manually by the administrator. However, each organization has a trust relationship with only a few other organizations, and it builds this relationship only once. When the relationship is validated and the Mapping DB created, the system becomes standalone.

The mapping policy is applied in the site set which generally uses a similar policy e.g., RBAC, MAC, DAC. For example: in a medical community, it is probable that roles such as "Doctor", "Nurse" or "Patient" exist in all organizations, allowing for an easy mapping through the community.

The chain of APC might be seen as a problem due to the uncertainly chained of certificates. Nevertheless, the APC contains a traceability of trust chain (Trace part: Id of sites and the distrust degree). Thus, the target site checks the Trace part of APC, and denies access whether it does not trust any site in the trust chain, or the degree becomes greater than a given value (leading to high distrust in the guest).

VIII. Conclusion and Future Work

Trust is basis of the inter-domain relationships in ubiquitous computing. In this paper, we reviewed trust models and have showed difficulties to apply these models. We have presented a certificate model which aims at enabling a broad access to a user when this latter is roaming in an environment where she is not locally known. Besides, the user can enter unknown sites with various user interfaces (devices) using a single certificate. We also validate the feasibility of our approach by the demonstrator.

We are currently developing an Inference Engine that derives the distrust value. This engine computes this distrust value according to new constraints like geographic position, type of user interface, type of authentication, but also the

history of site recommendations. Indeed, the level of distrust can be adjusted to capture the past interaction between the trustor and its trustees.

Finally we investigate how the distrust model presented here can be integrated to treat the security requirements of our team project PerSe [22], which implements a pervasive service environment.

REFERENCES

[1] Shankar, N. and Arbaugh, W. On Trust for Ubiquitous Computing. *Workshop on Security in Ubiquitous Computing*, September 2004.

[2] Brummit, B. *et al.* EasyLiving: Technologies for Intelligent Environments. *2nd International Symposium on Handheld and Ubiquitous Computing*, September 2000.

[3] Kindberg, T. and Barton, J. A Web-based Nomadic Computing System. *Computer Networks*, **35**(4): 443–456, 2001.

[4] Kagal, L., Korolev, V., Avencha, S., Joshi, A., Finin, T. and Yesha, Y. *Highly Infrastructure for Service Discovery and Management in Ubiquitous Computing*, Technical report, TR CS-01-06, Department of Computer Science and Electrical Engineering, University of Maryland Baltimore Country, Baltimore, 2001.

[5] Abdul-Rahman, A. and Hailes, S. A distributed Trust Model. *ACM Workshop on New Security Paradigms*, pp. 48–60, September 1997.

[6] Marsh, S. P. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, April 1994.

[7] Beth, T., Borcherding, M. and Klein, B. Valuation of Trust in Open Networks. *The European Symposium on Research in Computer Security*, November 1994.

[8] Zimmermann, P. R. *The Official PGP User's Guide*. MIT Press, Cambridge, MA, 1995.

[9] *ITU-T Simple public key infrastructure* (SPKI) charter, http://www.ietf.org/html.charters/OLD/spki-charter.html.

[10] Seitz, L., Pierson, J. M. and Brunie, L. Semantic Access Control for Medical Applications in Grid Environments. *International Conference on Parallel and Distributed Computing*, pp. 374–383, August 2003.

[11] Kagal, L., Finin, T. and Joshi, A. Trust-Based Security in Pervasive Computing Environments. *IEEE Computer*, **34**(12): 154–157, December 2001.

[12] Bussard, L., Roudier, Y., Kilian-Kehr, R. and Crosta, S. Trust and Authorization in Pervasive B2E Scenarios. *6th Information Security Conference*, October 2003.

[13] Kagal, L., Finin, T. and Peng, Y. A Delegation Based Model for Distributed Trust. *Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents*, pp. 73–80, August 2001.

[14] Pearlman, L., Welch, V., Foster, I., Kesselman, C. and Tuecke, S. A Community Authorization Service for Group Collaboration. *IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, June 2002.

[15] Welch, V., Siebenlist, F., Foster, I., Bresnahan, J., Czajkowski, K., Gawor, J., Kesselman, C., Meder, S., Pearlman, L. and Tuecke, S. Security for Grid Services. *Twelfth International Symposium on High Performance Distributed Computing*, June 2003.

[16] Harrison, M. H., Ruzzo, W. L. and Ullman, J. D. Protection in Operating Systems. *Communications of the ACM*, **19**(8): 461–471, 1976.

[17] Bell, D. E. *A Refinement of the Mathematical Model*. Technical Report ESD-TR-278 vol. 3, The Mitre Corp., Bedford, MA, 1973.

[18] Sandhu, R., Coyne, E. J., Feinstein, H. L. *et al.* Role-Based Access Control Models. *IEEE Computer*, **29**(2): 38–47, 1996.

[19] He, R., Niu, J., Yuan, M. and Hu, J. A Novel Cloud-Based Trust Model for Pervasive Computing. *The Fourth International Conference on Computer and Information Technology*, pp. 693–700, September 2004.

[20] Almenarez, F., Marn, A., Campo, C. and Garcia, C. R. PTM: A Pervasive Trust Management Model for Dynamic Open Environments. *Workshop on Pervasive Security, Privacy and Trust*, August 2004.

[21] *ITU-T Rec. X.509 (2000)*. ISO/IEC 9594-8 The Directory: Authentication Framework

[22] Bihler, B., Brunie, L. and Scuturici, V. Modeling User Intention in Pervasive Service Environments. *The International Conference on Embedded and Ubiquitous Computing*, December 2005.

Rachid Saadi is currently a PhD Student in LIRIS laboratory at the National Institute of Applied Sciences (INSA) of Lyon. He received his Master and engineering degree in computer science respectively in 2003 at INSA of Lyon and 2000 at University of Sciences and Technologies Houari Boudiene(USTHB), Algeria. His research interests include pervasive systems, ubiquitous computing, trust and security. His current work concerns the management of trust in pervasive environments.

Jean-Marc Pierson received his PhD from the ENS-Lyon, in France in 1996. Since 2001, he is a member of the LIRIS laboratory and works as an Associate Professor at INSA-Lyon. His main interests are related to Data Management in large scale distributed systems, with several project in Grids and Pervasive environments. His researches focuses on security, cache and replica management, and data mediation.

Prof. Dr. Lionel Brunie is the director of the doctoral school of computer science of Lyon (EDIIS – Lyon). After he received his PhD in computer science at the Joseph Fourier University, Grenoble, Lionel Brunie joined Ecole Normale Suprieure (LIP lab) of Lyon as assistant professor. Then he took a University Professor position in computer science at the National Institute of Applied Sciences (INSA) of Lyon in October 1998 where he co-founded the LIRIS laboratory in 2002. Lionel Brunie leads a research team of 15 researchers and coordinates the Health informatics research action (40+ researchers). His main topics of interest include: collaborative multimedia information systems, multimedia databases, distributed systems, medical informatics. Lionel Brunie is the (co-)author of over 100 research papers; he has been member of over 30 scientific conference and workshop committees.