Pervasive Authentication and Authorization Infrastructures for Mobile Users

Jordi Forné[1], Francisca Hinarejos[1], Andrés Marín[2], Florina Almenárez[2], Javier Lopez[3], Jose A. Montenegro[3], Marc Lacoste[4,] and Daniel Díaz[2]

[1] Telematics Engineering – Technical University of Catalonia
{ jforne,mfcampos}@entel.upc.edu
[2] Telematics Engineering –University Carlos III of Madrid
{ amarin, florina,dds }@it.uc3m.es
[3] Computer Science Department – University of Malaga
{jlm, monte}@lcc.uma.es
[4] Orange Labs
marc.lacoste@orange-ftgroup.com

Abstract

Network and device heterogeneity, nomadic mobility, intermittent connectivity and, more generally, extremely dynamic operating conditions, are major challenges in the design of security infrastructures for pervasive computing. Yet, in a ubiquitous computing environment, limitations of traditional solutions for authentication and authorization can be overcome with a pervasive public key infrastructure (pervasive-PKI). This choice allows the validation of credentials of users roaming between heterogeneous networks, even when global connectivity is lost and some services are temporarily unreachable. Proof-of-concept implementations and testbed validation results demonstrate that strong security can be achieved for users and applications through the combination of traditional PKI services with a number of enhancements like: (i) dynamic and collaborative trust model, (ii) use of attribute certificates for privilege management, and (iii) modular architecture enabling nomadic mobility and enhanced with reconfiguration capabilities.

Keywords. *Ubiquitous Computing, Authentication, Authorization, Trust, Security Architecture*

1. Introduction

Advances in wireless technology and portable computing along with demands for higher user mobility have provided a major impetus towards ubiquitous computing. The promise of this new paradigm is the integration of microprocessors into everyday objects, able to communicate among themselves and with users by means of ad-hoc and wireless networking. Indeed, wireless networks provide mobile users with ubiquitous communication capabilities giving them access to information regardless of their location.

In order to support business applications in ubiquitous networks, trust relationships between users need to be strengthened. Therefore, increasing confidence requires pervasive security services based on strong authentication and authorization mechanisms. In ubiquitous computing, the main security challenges arise from network heterogeneity as well as from a dynamic population of nomadic users with limited devices. The European Project UBISEC aimed at an advanced infrastructure for large-scale mobility and security; more precisely, for context-aware and personalised authorization and authentication services in heterogeneous networks. This requires high-security personalisation and localisation technologies in order to keep privacy and to protect computing devices, their software components, and personal user data such as user profiles.

In this paper, we present a pervasive infrastructure for authentication and authorization services in heterogeneous networks, in the form of a pervasive public key infrastructure (pervasive-PKI). This infrastructure, developed as part of the UBISEC project, is able to provide authentication and access control services for users roaming between different heterogeneous networks. In this sense, the pervasive-PKI fully supports nomadic mobility, enabling secure services for users connecting through many different networking technologies (Wi-fi, UMTS, Bluetooth, etc.), and in multiple network topologies, even when global connectivity is lost and some services are temporarily unreachable.

We clearly differentiate between two modes of operation: in *connected mode*, on-line trusted servers are available and traditional techniques are applicable for validation of user credentials; however, in *disconnected mode*, the information necessary for this validation is not always available. To support the disconnected mode, we combine different solutions: an adapted privilege verifier for authorization, a new trust model for authentication, and a collaborative model to obtain unavailable information. Some of the functions traditionally performed by authentication and authorization infrastructures are integrated into user devices, providing support for credential validation in situations where central authorities are not available, like in peer-to-peer mobile ad-hoc networks (MANETs). Furthermore, the pervasive-PKI is also endowed with reconfiguration capabilities.

The rest of the paper is organized as follows. Section 2 presents the required background, including authentication and authorization infrastructures, evidence-based computational trust management, and component-based reconfigurable architectures. Section 3 points out the requirements of the pervasive-PKI. We then present the proposed architecture for the pervasive-PKI in Section 4, highlighting the components embedded in user devices. Section 5 describes a proof-of-concept implementation developed for the UBISEC project, whereas evaluation results are shown in Section 6. Section 7 is devoted to related work. Finally, Section 8 concludes the paper.

2. Background

2.1. Authentication and Authorization Infrastructures

Authentication solutions like Microsoft .NET Passport and Kerberos depend on user-selected passwords. However, from a security point of view, it is more interesting to use further advanced technologies like digital certificates and PKIs, which combined with some complementary techniques and tools, can also be used as a starting-point to provide authorization. Actually, a X.509v3 *identity certificate* (or *public-key certificate*) can convey authorization information about its 'subject'. For instance, the information can be encoded in one of the X.509v3 standard extension fields.

Nonetheless, the dynamics of the authentication and the authorization information are different. Identity certificates are typically designed to be valid for a relatively long period of time (e.g. 1 or 2 years). Contrarily, the persons authorized to perform a particular function in a company may vary monthly, weekly, or even daily. For that reason, ANSI X9 Committee developed an alternative approach known as *attribute certificate*. This approach has been incorporated into both the ANSI X9.57 standard and the X.509-related standards and recommendations of ITU-T, ISO/IEC, and IETF.

In this sense, the X.509 ITU-T Recommendation [1] specifies the format of an attribute certificate (AC) as a separate data structure from the identity certificate of the subject. The Recommendation proposes that an attribute certificate is issued by an *attribute authority* (AA), rather than by the traditional *certification authorities* (CA) of the identity certificate case. Additionally, ITU proposes the binding of both certificates in such a way that one subject has multiple attribute certificates associated with his identity certificate. Finally, and in a similar way to the PKI case, the chains of attribute certificates can be built recursively and used in a framework supported by a *privilege management infrastructure* (PMI). PMIs also support delegation of rights, which is clearly beyond the capabilities of a traditional PKI.

In ubiquitous scenarios, the need for an integrated authentication-and-authorization service for peers, that is, an *authentication and authorization infrastructure* (AAI), is stronger than ever before. Consequently, the challenge of an AAI is to provide an inter-domain authentication and authorization service. Using an AAI, a user would register only once in his home domain. When he requests a resource in a visited network, he should always be authenticated and authorised by using his home domain credentials. The important issue is that visited networks do not need to register the users by themselves. Instead, they trust the registration process and the credentials provided by the user's home trust domain.

It should be pointed out that PKIs suffer from a restricted and static vision of trust, that is, CAs are organised in strict hierarchies where trust flows from the root to the leaves and certificates denote direct trust relationships, while certification paths capture indirect trust relationships. The same problems have been inherited by PMIs in the few implementations available at this time and, consequently, will be inherited by AAIs because the aforementioned trust model is not suited for P2P networks, ad-hoc networks, and situations where no administrator is available or cannot be afforded. It is precisely that this research work elaborates on a new trust model reflecting the required dynamic nature of trust for roaming users, with little administrative overhead, and which can exploit the communication and collaboration capabilities of new situations.

2.2. Evidence-Based Computational Trust Management

In a PKI, trust is fundamental to establish "certification paths" among entities, either CAs or end users. End user applications usually handle trust relations through "Trusted Certificate Lists". A trusted certificate list is the set of embedded root keys usually found within web browsers like Netscape, Internet Explorer, etc. Such root keys are used to successfully validate a certificate chain. This approach is convenient for systems including a relatively small number of well-known CAs, and is applicable for direct use within companies, and/or to support interactions across a predetermined set of corporate boundaries. Indeed, PKI configuration requires manual intervention, and applying hierarchies through cross-organizational boundaries on a large scale basis could be difficult.

This scheme is not applicable to mobile users, because these often require peer-to-peer trust relations. In this kind of relationship, each user or domain can be a trust anchor. Although a peer-to-peer trust model is more flexible than traditional PKI, it suffers from scalability and uncertainty problems. For instance, PGP [2] is a well-known example of this type of system. It can be said that PGP manages credential-based trust, which basically allows delegating trust.

For these reasons, evidence-based computational trust management models have been proposed such as PTM [3], Subjective Logic [4], SECURE [15], ENTRAPPED Platform [5], TMF [6], among others. These models consider risk and uncertainty issues, and bring dynamism and flexibility. However, open and peer-to-peer systems are vulnerable to Sybil attacks [7], and computational trust management models do not have well-referenced trust metrics for assessing and reasoning about attack-resistance [8]. Such issues must be taken into account by the several approaches.

Pervasive Trust Management (PTM) has been designed for mobile users, by providing them with autonomy to establish new peer-to-peer trust relations, even with unknown users. PTM models trust relations with continuous function ranging from 0 to 1, where these values represent the extreme cases of complete distrust and complete trust, respectively. The initial trust values are established according to security rules, collaboration among trusted peers, or even user intervention when it is required. After evidences, the trust values change; that is, trust evolves over time in accordance with the user's behaviour. In order to prevent Sybil attacks and to have a high attack-resistance, PTM minimizes possible sources of attacks, taking into account the trustworthiness of entities, establishing a security level and a cooperation threshold according to the perceived risk, and limiting the length of recommendation paths. Likewise, trust evolution is based on the principle: "Trust comes on foot and goes by horse". The increasing factor is proportional to the number of good interactions over time. On the contrary, trust can be easily lost with few negative actions, or an attempted attack. For instance, a forgery user previously require a lot of good interactions to acquire an enough trust level to be recommender in environments with a low security level. Furthermore, a spiteful recommendation can be monitored in order to penalize to the recommender. In the Figure 1, trust evolution is represented with respect to different behaviour patterns. The formulae modelling the dynamic nature of trust can be found in [3].

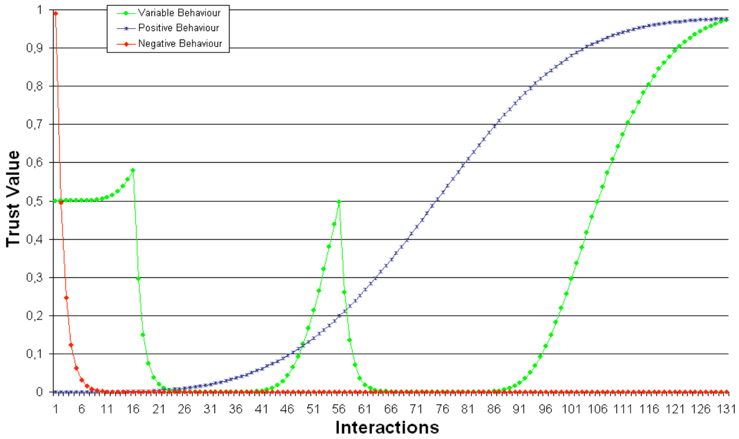

Figure 1. Trust evolution according to different behavior patterns

Trust information obtained from PTM can be also used to take access control decisions, evaluate risk, and calculate historical behaviour.

2.3. Component-Based Reconfigurable Architecture

The dynamicity and heterogeneity of ubiquitous environments require security management to be flexible enough to be easily tailored to different operating conditions such as multiple authorization and authentication policies, variable user preferences, or scarce resources — when only key security services should be included into a featherweight security infrastructure.

Component-based security architectures are currently emerging as a promising solution to reach such flexibility. Components are usually defined as entities encapsulating code and data which appear in software systems as units of execution, configuration, deployment, or administration. The component paradigm enables the security architect to master the complexity of implementation of a software infrastructure: since components can be composed to form higher-level units of code, one can observe and manipulate the infrastructure at the right level of abstraction and granularity, both during design and implementation phases. The resulting infrastructure is thus very modular.

Component-based design is also a simple but powerful manner to achieve flexibility of configuration and reconfiguration: functionalities can be simply adapted or inserted by addition or replacement of components in the system. Indeed, a component model simplifies reconfiguration management by providing control over relationships between components, both in term of containment and interconnection, independently from component functionality. This design approach is thus well suited to the dynamic needs of pervasive computing environments.

To achieve an acceptable trade-off between security and flexibility, we will specify the architecture of the pervasive PKI using the component paradigm. This choice of design facilitates the introduction of hooks to reconfigure authentication and authorization mechanisms. In the following, reconfiguration capabilities will be illustrated using Fractal [9, 10], a generic component model which captures reconfiguration by flexible composition of components using a minimal number of concepts, shown in Figure 2. In Fractal, a component is a run-time entity built from a controller, which supervises the execution of a content possibly including other components (sub-components). A composite component reveals the organization of its content, while a primitive component is a black-box to encapsulate legacy code. A component only interacts with its environment through well-defined interfaces, which may be provided or required. Components interact by establishing bindings between their interfaces.

Fractal manages reconfiguration separately from component functional behaviours, by distinguishing between control interfaces and functional interfaces. The control interfaces of the component model allow to customize properties such as containment and binding relationships between components (BindingController interface), configuration of component properties (AttributeController interface), dynamic reconfiguration, by adding or removing sub-components (ContentController interface), and life-cycle management, for instance, by suspending or resuming component execution (LifeCycleController interface).

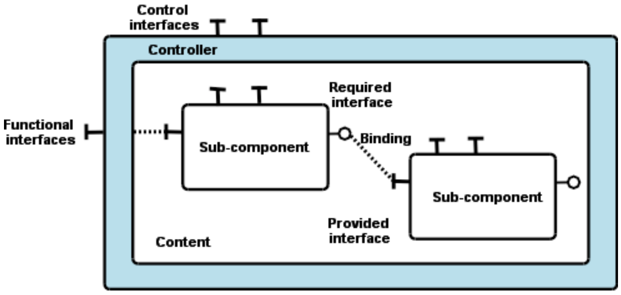

Figure 2. Main concepts of the Fractal component model.

3. Requirements for the Pervasive PKI

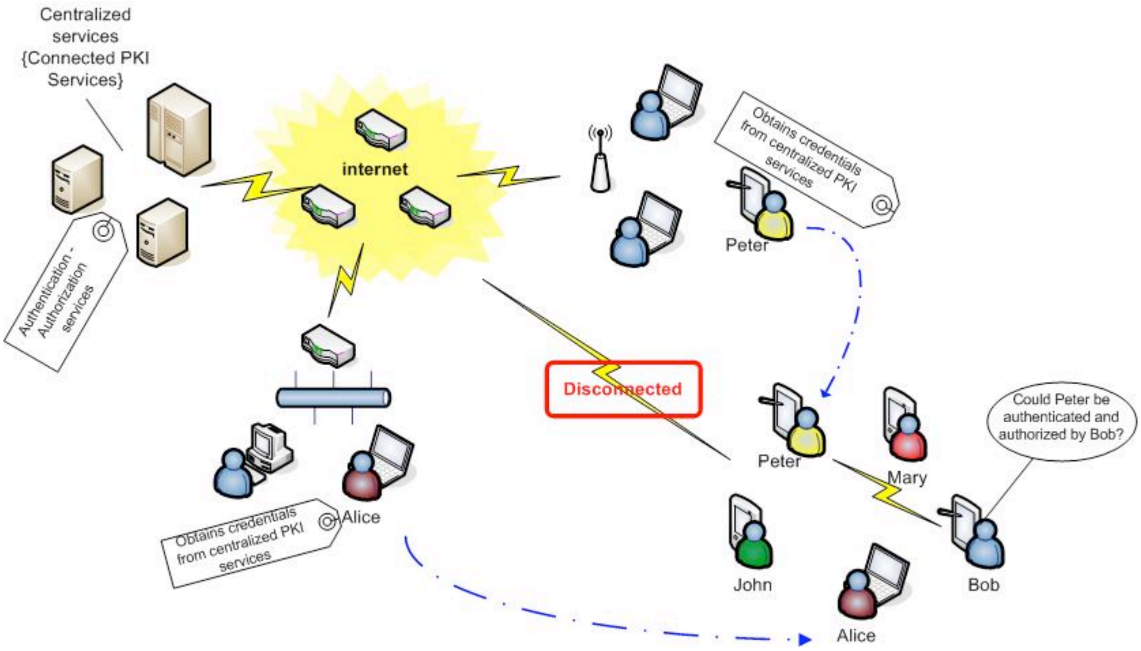
Centralized services {Connected PKI Services}
Authentication - Authorization services
internet
Obtains credentials from centralized PKI services
Peter
Disconnected
Obtains credentials from centralized PKI services
Alice
Peter
Mary
Could Peter be authenticated and authorized by Bob?
John
Bob
Alice

Figure 3. Typical operation scenario for the pervasive-PKI.

Figure 3 shows a typical operation scenario that should be supported by the pervasive-PKI. Let Peter and Alice be two mobile users that have temporal connectivity to centralised PKI services (for example, when they have a stable Internet connection). During this period of connectivity they work in *connected mode*, and they have full access to all PKI services offered by the infrastructure: they can register, obtain certificates, download certificate revocation lists (CRLs), contact to online certificate status protocol (OCSP) responders, etc. Later, users move and lose global connectivity. However, they can still communicate among each other by forming a peer-to-peer MANET but they cannot accede to centralized PKI services. In this situation, which we call *disconnected mode*, users should be able to establish secure communications. In particular they should be able to perform authentication and access control decisions. The main objective of this research is focused on providing mechanisms that extend the PKI functionalities to mobile users when they work in the disconnected mode. To achieve this objective, we propose a new architecture for the pervasive-PKI, where functionality traditionally performed by a centralized infrastructure is moved to user devices, making certificate validation also possible in the disconnected mode.

Below, we state the main requirements for an infrastructure providing authentication and authorization services in ubiquitous scenarios, which lead to the design of the pervasive-PKI:

Operation over heterogeneous networks.- Pervasive computing environments include many different network topologies and technologies (Wi-fi, UMTS, Bluetooth, etc.). This heterogeneity implies multiple disconnected trust domains, where each domain applies its own policies and mechanisms for authentication and authorization. Consequently, an important challenge for the pervasive-PKI is to provide an inter-domain authentication and authorization service.

Support for the authorization service.- PKI-based mechanisms are suitable for authentication in heterogeneous trust domains, and also facilitate mobility over heterogeneous networks and temporal disconnection of services: users carry their credentials to authenticate themselves anywhere at anytime. Similarly, credential-based authorization also allows supporting more scalable decentralised authorization policies.

Support for mobile users.- A major challenge to implement mobility is the free roaming of users across different administrative domains. In the past, users have been demanding roaming in homogeneous GSM networks. However, in the near future, users will require context-aware computing involving an increasing number of heterogeneous networks and mobile devices. One key enabler of future mobile systems will therefore be the support of roaming over heterogeneous networks. Therefore, a main requirement is to provide an infrastructure supporting secure services for mobile users *anywhere* and *anytime*.

Single sing-on (SSO).- Using an AAI, a user would register only once in his home domain. When he requests a resource in a visited network, he should always be authenticated and authorised by using his home domain credentials. The important point to note is that visited networks themselves do not need to register the users; instead they trust the registration process and the credentials provided by the user home trust domain. They only focus on local authorization and access control decisions.

Support for temporal disconnections.- When users move across different networks, global connectivity may be lost and some PKI services may be temporarily unreachable. Figure 3 shows the main modes of operation for the pervasive-PKI: *connected* and *disconnected* modes. When working in the connected mode, users have full access to all PKI services, including certificate issuing and validation. On the other hand, in the disconnected mode, the infrastructure cannot be reached and alternative mechanisms for certificate validation are required.

A dynamic trust model for disconnected modes.- It should be pointed out that existing PKIs suffer from a restricted and static vision of trust, that is, CAs are organised in strict hierarchies where trust flows from the root to the leaves and certificates denote direct trust relationships, while certification paths capture indirect trust relationships. The same problems have been inherited by PMIs in the few implementations available at this time and, consequently, will be inherited by AAIs because the aforementioned trust model is not suited for P2P networks, ad-hoc networks, and situations where no administrator is available or cannot be afforded. This is precisely why this research work elaborates on a new trust model reflecting the required dynamic nature of trust for mobile users, with little administrative overhead, and which can exploit the communication and collaboration capabilities of new situations. Several typical certificate validation processes, such as path processing and revocation status checking cannot be guaranteed when working in disconnected mode. Even more, unrelated users possibly certified by unknown CAs may want to interact. Therefore, we require a new trust model for situations where a certificate cannot be verified or new trust relations cannot be established by traditional PKI mechanisms.

Operation into limited devices.- Mobile users typically used lightweight devices such as PDAs or mobile phones. Although these devices are easily portable they have much more limited capabilities than PCs or laptops. In particular these constrained devices have limited computational, communication and storage capabilities, and power consummation is also an important issue. We have to take these limitations into account, providing mechanisms that can be performed by very constrained devices.

Reconfigurability.- The operating environment is constantly changing due mainly to: dynamic trust relationships between users and devices; download of platform updates; installation of new security components and policies; and personalization of existing services. These conditions require an adaptable security infrastructure supporting both dynamic configuration to change some security parameters, and reconfiguration to introduce new protection mechanisms.

4. Architecture

Authentication and authorization services for Internet users can be based on AAIs as presented in section 2.1. In the *connected mode*, Public Key Infrastructures and Privilege Management Infrastructures (PMIs) can efficiently support both services, respectively. However, in the *disconnected mode* these infrastructures are not reachable and we require new solutions. This section proposes an architecture that extends the functionality of several PKI services to the disconnected mode.

Figure 4 shows the proposed architecture for the pervasive-PKI. Mobile users obtain their credentials from a X.509 AAI and they store the certificates in their devices or smart cards. Further authentication and authorization will imply the validation of these credentials. In connected mode, part of the AAI can be used to help credential validation, whereas in disconnected mode the infrastructure is not reachable and several functionalities such as certificate path processing and revocation status checking are not available. Therefore, several cooperating software components installed in the user device have to be used to support credential validation.

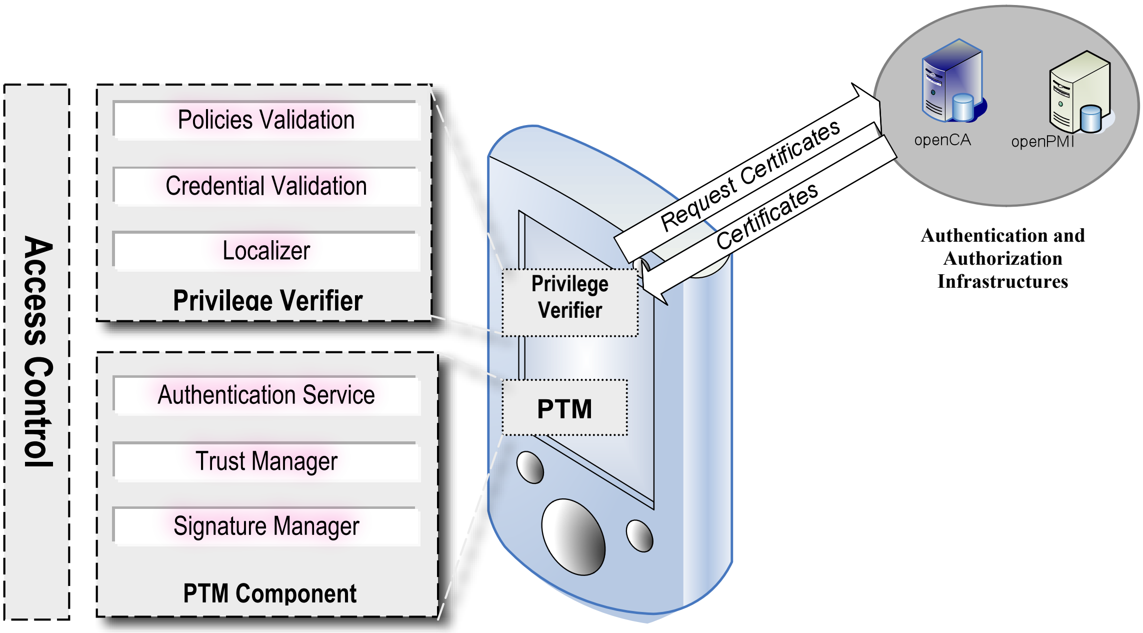

Figure 4. Proposed architecture for the pervasive-PKI

Mobile users can form self-organizing networks in isolation, in order to share services, or to participate in peer-to-peer applications. Thus, some devices have to behave as servers, making certain decisions about access control, establishment of new trust relationships, or validation of credentials, etc. In this section, we present the three software components that have to be included in user devices that allow implementing secure authentication and access control, even in disconnected mode: the PTM component, the Privilege Verifier (PV) and the Access Control Engine (ACE).

4.1. PTM Component

PTM component manages trust information about users, validates public key certificates (PKCs), signs messages and verifies digital signatures. These functionalities are provided by three components:

(i) The *Authentication Service* manages PKC validation, using the "trusted certificate list" holds in the user device. In the disconnected mode case, this component has been adapted by implementing our own algorithm for certification path validation. The algorithm uses recommendation information, trusted certificate list, and a revocation service if available [11]. Here is how our algorithm works:

   a. Firstly, PTM receives a PKC supplied by the user or by another way to the pervasive-PKI system.

   b. PTM checks PKC validity (syntax, signature, and period) and then it verifies the certification path. It allows trusted auto-signed certificates or certificates issued by directly trusted users.

   c. If revocation information is available, PTM requests revocation information about certificates in the certification path. If not, PTM could request recommendation information to close trusted peers. Finally, if additional information can not be obtained, the certificate is validated with a low authentication level.

   This component also provides information, e.g. identity certificates, to other components.

(ii) The *Trust Manager* boots and manages trust information about users. Trust information includes trust values, and trustworthiness level according to a threshold. This information allows to handle the trusted certificate list in a semi-automatic way. Likewise, this component maintains a black list of untrustworthy users. The pervasive trust model underlying allows dynamic modification of the trust value assigned to an entity. This is achieved by incorporating an "Action Monitor" module which monitors the evidences, i.e. interactions from different entities. The interactions are classified by the service designer in order to assign them weights, in order to recalculate the trust values assigned to those entities according to the Pervasive Trust Model formulae.

(iii) Finally, the *Signature Manager* can act in two ways:

   a. For message signing, using the user's private key that can be placed in tamper-proof storage like a smart card.

   b. For signature verification, this uses the public keys bound to the certificates stored in the trusted certificate list.

Summarising, the PTM component has the following outcome interfaces:

- The result of the PKC validation and any error information.
- The information contained in the PKC.
- The trust information about user, for instance, trustworthiness, trust value, and user's behaviour.
- The signature of a message.
- The result of the digital signature verification.

4.2. Privilege Verifier (PV)

This component manages the validation of Attribute Certificates. The Privilege Verifier (PV) is divided into three components:

(i) The *Authentication Manager* manages PKC validation. This component was adapted to disconnected mode to work in connection with the PTM component. When it is needed to validate any PKC linked to the AC target, the *Authentication Manager* delegates this functionality to the PTM component.

(ii) The *Localizer* component provides information (certificates, policies…) to other components of the PV. For example, the AA's PKC must be validated in order to verify the AC's signature. The *Localizer* can act in to ways:

a. It can provide the necessary PKCs to the validation of an AC from its local cache,

b. or it can request this information to the PTM module.

(iii) The *Attribute Certificate Verifier (AC Verifier)* component is in charge of validating the user AC and to get the privileges or role assigned to the user. Therefore, this component supplies to the rest of modules with the information contained in the CA of the user.

Attribute Certificate validation is performed as follows:

- Firstly, the PV receives an AC supplied by the user or by another way to the pervasive-PKI system. The PKC linked to the AC can be supplied in the same way as the AC. If the PKC is not supplied, the *AC Verifier* can obtain it from the *Localizer*. If the *Localizer* stores the PKC into his local cache, send it to the *AC Verifier*, if not, requests the PKC to the PTM component.
- The validation of PKCs is then delegated to the PTM component and is managed by the *Authentication Manager*. If the PKCs are valid, the authentication result is a

boolean value. Otherwise, the outcome is a trust level calculated by the PTM component.

- The *AC Verifier* component uses the information supplied by the user and the PTM component to validate the user privileges based on policies and environment variables.

Therefore the outcomes of the PV are:

- The outcome of the AC validation.
- The information contained in the AC, for example, the user attributes, the validity period, and so on.

4.3. Access Control Engine (ACE)

This component is in charge of decision-making when controlling access to resources. It relies on the PTM to authenticate users requesting access, and on the PV to validate the credentials presented by the requester. Access is then granted or not depending on the current authorization policy. The ACE implementation mostly follows the XACML access control framework, clearly separating logic for policy enforcement and decision.

4.4. Reconfigurability

Following the discussion of section 2.3, we now show the benefits of adopting a component-oriented architecture for the pervasive PKI to achieve adaptability in the security services it provides. We illustrate this approach on the case of authentication.

A flexible authentication service should allow adapting the authentication method to the security context. This operation can range from fine-tuning some configuration parameters to changing the authentication algorithm. For instance, the strength of authentication may be tuned by selecting the threshold T for PTM trust values above which user entities are authenticated: T=1 for boolean authentication if a CA is available on-line as in a traditional PKI; and T<1 for disconnected mode, where trust is managed in a P2P manner between entities. At the other end, if different authentication algorithms are supported, a Pluggable Authentication Modules (PAM) type of architecture for the PTM component allows to install different Authentication Service Providers (ASP), and to select the provider best matching the security context (selectASP method). A component-based architecture for the authentication service captures both situations.

We consider two modes of operations for authentication. In connected mode, TTP (Trusted Third Party) servers are available on-line to validate credentials. Traditional PKI-based schemes are therefore applicable to manage trust. A TTP may not be accessible, the lack of a stable backbone in infrastructure-less networks resulting in intermittent connectivity. In disconnected mode, due to missing information, validation operations cannot be performed so simply, and require new models of trust. A distributed trust management system such as [3] is then more suitable to handle trust relationships between devices to take decisions without central servers. These two modes are managed by separate sets of components, shown in Figure 5.

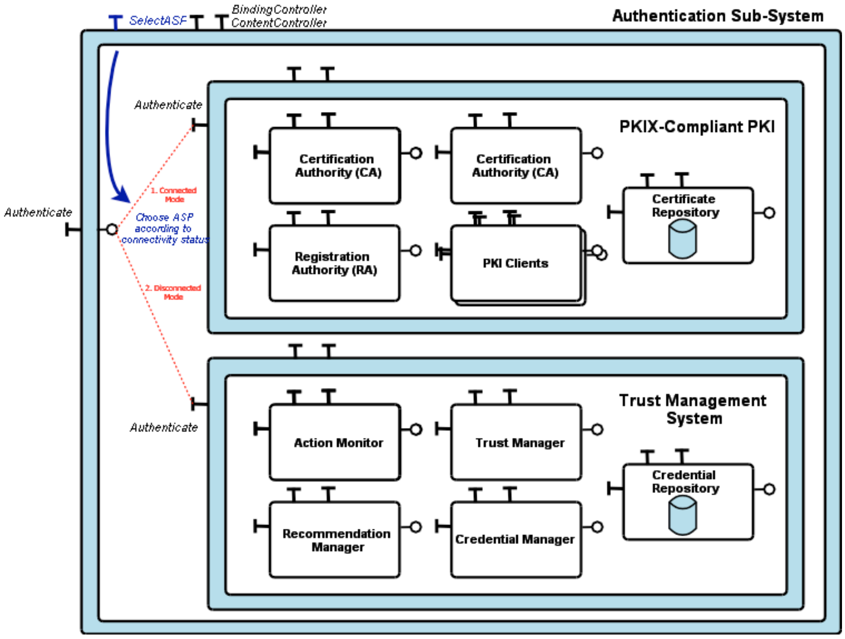

Figure 5. Components of an adaptable authentication service in the pervasive PKI.

In connected mode, authentication and trust management are based on certificates using a PKI. The PKI includes typically the following components which may be distributed: the PKI Clients may initiate Certificate Signing Requests (CSRs) or certificate revocation requests, and verify the validity of certificates; the Registration Authority (RA) may approve CSRs, or revoke identity certificates; the Certification Authorities (CAs) may issue new certificates, sign CSRs, verify the validity of a certificate, or revoke certificates, for instance when a public key was compromised; finally, the Certificate Repository allows storing and retrieving certificates and Certificate Revocation Lists (CRLs).

In disconnected mode, authentication is based on reputations maintained by a P2P trust management system (TMS), including the following components: an Action Monitor keeps track of behaviours (normal or malicious) of other devices; a Trust Manager combines this information with recommendations received from other devices to update the reputation of each device according to the chosen trust model; a Recommendation Manager implements the recommendation protocol between devices; the reputation values are then converted by the Credential Manager into credentials for authentication, stored in the Credential Repository.

Authentication functionalities may be adapted to the security context at several levels, starting with the choice of the trust management strategy (certificates vs. trust values) depending on the connectivity status to a TTP. Changes are performed simply using the BindingController control interface to bind the Authenticate functional interface of the authentication service, either to the PKI or the TMS sub-components.

The previous design also provides adaptability in the deployment architecture, the provided security services, and the supporting security protocols to meet the current protection requirements. As a rule, the functional components of the authentication service are distributed, deployed in different network topologies. For instance, the PKI CAs can be organized in hierarchies, optionally federated using cross-certification, or in mesh networks, where certificate holders are both CAs and clients. Independently from the network architecture, the PKI is expected to provide a list of security services which may need to be extended. Further, for each security service such as certificate validation, the interactions between the components can be described with several protocols [9]. The proposed component-based architecture supports these different types of PKI design by providing full control over the deployment of functional components, their relationship with security services, and the interaction protocols between components.

When mapping the security services onto the functional components, one obtains a set of technical components which can be arranged flexibly to realize several types of PKI architectures. The component interfaces can be specified with an ADL (Architecture Description Language) such as the Fractal ADL 10. The CA component is shown in Figure 6, the other components being similar. Using the ContentController and BindingController interfaces, these components may be distributed on the network nodes according to a chosen topology. The topology may also be reconfigured according to the context, e.g., by creating a new CA, closer to clients, to optimize communications. The PKI security services can thus also be customized to the execution environment (security objectives, available resources), by adding/removing specific security components in the architecture. Finally, new interaction protocols (e.g., a more efficient certificate validation protocol) can be introduced by implementing specific bindings between components.

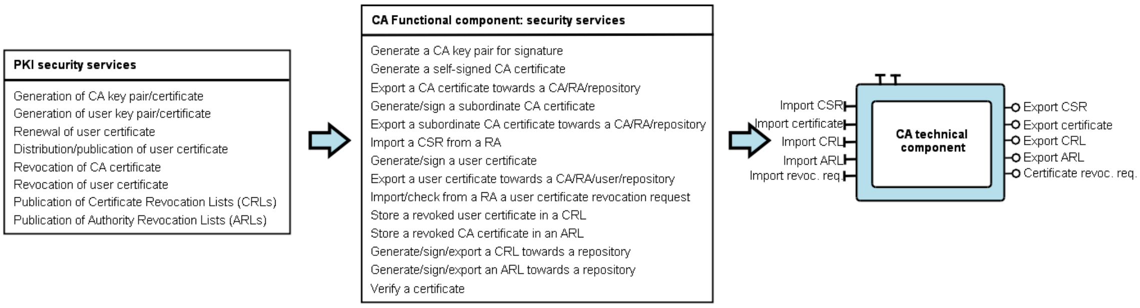
PKI security services
Generation of CA key pair/certificate
Generation of user key pair/certificate
Renewal of user certificate
Distribution/publication of user certificate
Revocation of CA certificate
Revocation of user certificate
Publication of Certificate Revocation Lists (CRLs)
Publication of Authority Revocation Lists (ARLs)
CA Functional component: security services
Generate a CA key pair for signature
Generate a self-signed CA certificate
Export a CA certificate towards a CA/RA/repository
Generate/sign a subordinate CA certificate
Export a subordinate CA certificate towards a CA/RA/repository
Import a CSR from a RA
Generate/sign a user certificate
Export a user certificate towards a CA/RA/user/repository
Import/check from a RA a user certificate revocation request
Store a revoked user certificate in a CRL
Store a revoked CA certificate in an ARL
Generate/sign/export a CRL towards a repository
Generate/sign/export an ARL towards a repository
Verify a certificate
Import CSR
Import certificate
Import CRL
Import ARL
Import revoc. req.
CA technical component
Export CSR
Export certificate
Export CRL
Export ARL
Certificate revoc. req.

Figure 6. From PKI security services to technical components.

The technical components share a number of finer-grained sub-components which allow tuning several PKI features. For instance: cryptographic algorithms; format of certificates; initialization procedures of the PKI entities; certificate life-cycle; local storage policies for keys and certificates; certificate validation protocols; or CSR management. Thus, the protocols governing interactions between the components of the PKI can be implemented and adapted very flexibly. Similarly, in disconnected mode, one can change the trust model, the action monitoring policy, the recommendation protocol, or the type of exchanged credentials by replacing the corresponding components of the TMS.

5. Proof-of-Concept Implementation

We considered the following scenario to demonstrate the functionalities of the pervasive-PKI in the UBISEC project. Let Peter, Alice and Bob be three users that previously do not know each other. Each user device (PDA) has the following software installed:

(i) A Photo Album Service (PAS) application, which allows users to store, view and organize their digital pictures.

(ii) The pervasive-PKI providing access control to both shared and private pictures.

Peter sometimes tries to break the security of all the devices that he can find. He doesn't have any picture in his album yet. Alice has some private pictures that she doesn't want to share. But, she has given permission to fans of the Pervasive club to access some of them. Bob has several pictures divided into two categories: private and free access. He is a fan of the Pervasive club.

The users form ad-hoc networks in order to exchange pictures. The available pictures can be viewed or stored into the local photo album repository. The access policy for these pictures should be inherited or defined to allow other users to get them.

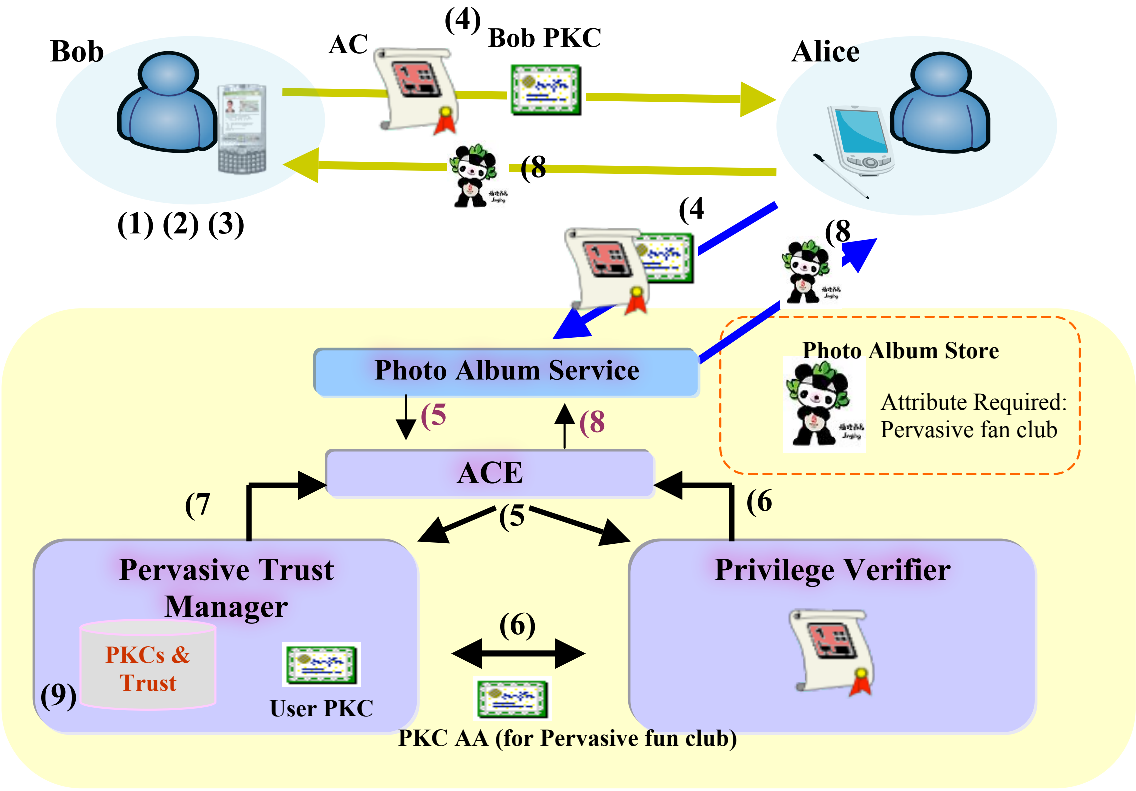

Figure 7. Testbed scenario for the pervasive-PKI

We tested two cases. In the first one, Bob act as client and Alice as server. Figure 7 shows the steps performed for a successful access to photo:

1. Bob starts the PAS application and joins the MANET.
2. Bob's PAS application starts a service discovery process to find other reachable PAS in the network.
3. Bob's PAS application shows all other available PAS. Alice's PDA offers pictures about the Beijing 2008 Olympic Games.
4. Bob asks for the picture of Beijing 2008 Olympics Jingjing Mascot, therefore, the identification and authorization process starts. Bob's credentials (PKC and AC) are sent to Alice to prove he has the rights to see the picture.
5. This request is delivered to Alice's pervasive-PKI software. Thus, the ACE component requests AC validation to the PV and PKC validation to the PTM.
6. The PV requests to the PTM the AA (for Pervasive fun club) PKC validation in order to validate the AC signature.
   a. If the AA PKC is valid and trusted, the PV gets the attribute bound to Bob's AC.
   b. The PV sends the response to the ACE component.
7. The PTM sends the response to the ACE concerning the user PKC validation.
8. The ACE sends the result to the PAS application to start sending Beijing 2008 Olympics Jingjing Mascot picture to Bob.
9. The PTM monitors Bob's behaviour to update his trust level.

In the second case, Bob acts as server and Peter as client: Bob is also offering a few pictures, and Peter tries to get a new picture from Bob's PDA. Bob's PDA then performs the same steps to validate Peter credentials. However, Peter is an untrustworthy user. Moreover, he doesn't have enough privileges to obtain that picture since Peter is not a member of the Pervasive fun club. Then, the ACE component denies access to Peter and his trust level is updated.

The PAS defines 28 different error codes arising from the validation process, i.e. output of the PV-PTM invocation. These error codes besides some other general patterns (like a DoS attack) are assigned weights by the PAS and continuously traced by the "Action Monitor" in the log files to recalculate the trust values.

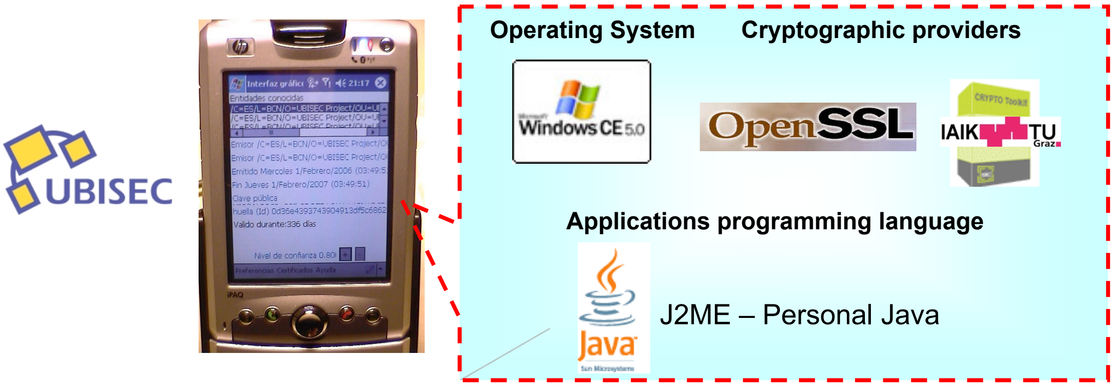

Figure 8. Software for proof-of-concept implementation

The implementation of the pervasive-PKI prototype has been developed in the J2ME Personal Profile. Different cryptographic providers have been tested: (i) OpenSSL (www.openssl.org) as open-source cryptographic provider; and (ii) IAIK (jcewww.iaik.tu-graz.ac.at) as cryptographic libraries based on Java. This implementation has been tested in Linux, Windows, and Windows CE. The developed software is publicly available as open-source, and can be downloaded from the main Web page of the UBISEC project.

The AAI is also based on freely-available software. The PKI components are based the widely-known OpenCA (www.openca.org) project, while the PMI is based on the OpenPMI project. The OpenPMI is an open-source prototype developed by the University of Malaga (http://openpmi.sourceforge.net). It is based on the ITU-T X509 recommendation, although is also influenced by the European Telecommunications Standards Institute (ETSI) and PKIX reports. The main component of the infrastructure is the Attribute Authority (AA). The AA performs the following services for Attribute Certificates: certificate generation, revocation management, and revocation status checking.

A detailed explanation can be found in Montenegro and Moya [12].

6. Evaluation

In this section we explain some measures we have obtained with our pervasive-PKI prototype and PAS application running in a PDA with a Intel PXA270 processor running at 520MHz, equipped with a IEEE 802.11b wireless interface. We have runned 40 times the experiment consisting of Peter downloading a picture from Alice which requires a valid Attribute Certificate with a given role. The amount of time required to download the picture is in mean 0.467 seconds. The pictures below show the overhead introduced by the validation process. The validation time includes the time consumed by the PTM component (1.25s in mean) and the time consumed to validate the attribute certificate (0.73s in mean). We have used a native method using JNI to get the value of a counter with precision of 1ms in the PDA, which is more precise than System.currentTimeMillis(). We have taken values of the different cryptographic operations required to validate the attribute certificate, as shown in Figure 10. The PAS implementation fires the PV component which in terms requests different validation methods from the PTM component. That is the reason why we subtract PV time from PTM time in Figure 9b, since PV includes the whole time consumed in the validation, and we wanted to be able to separate the contributions of the different validation steps.

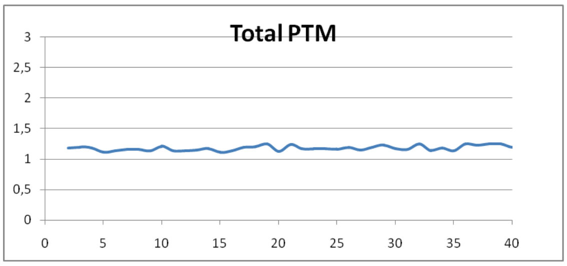

Figure 9a. PTM performance data

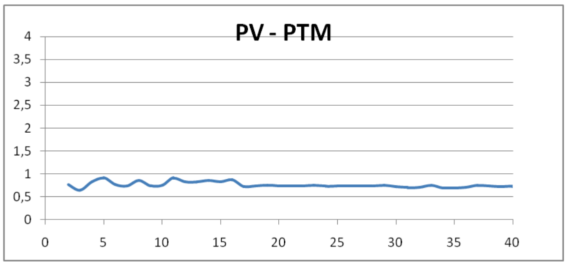

Figure 9b. PV performance data

We have used the cryptographic support of AIK for the PV component and a native implementation of a wrapper module of OpenSSL for the PTM component. We have also performed the same experiment in a PC with similar results, with roughly a magnitude (10x) improvement in time. Though in the PDA the result shows the time is not negligible for the user, this prototype is just a proof of concept not optimized for throughput, and we foresee that future implementations may reduce this time by a factor of 2, not taking into account the improvement of hardware.

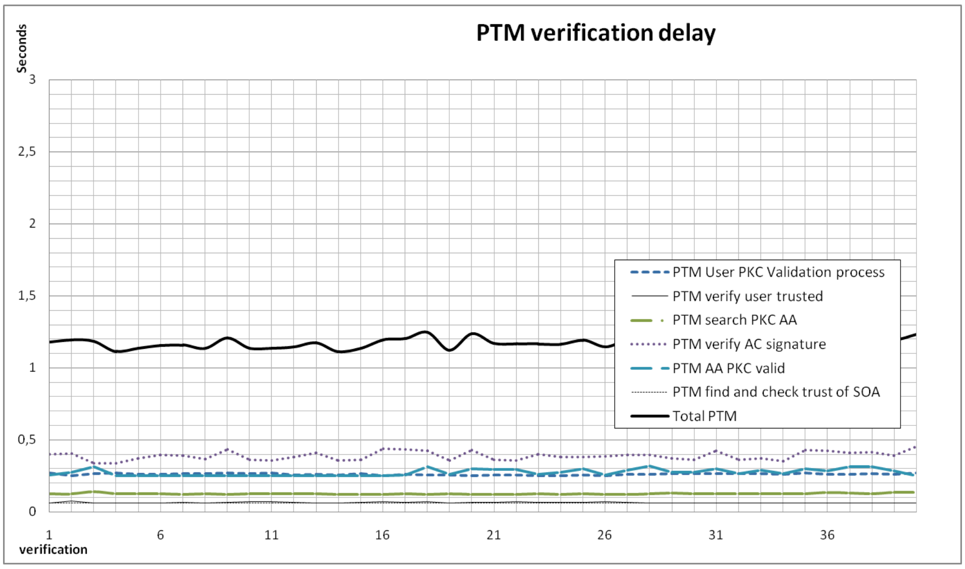

Figure 10. PTM verification delay

7. Related Work

The evolution of distribution systems and the notable progress of hand held device technology has constituted the base to the creation of the concept of pervasive computing. Security was one of the more active areas of distribution systems and consequently has been becoming essential in pervasive computing. At the beginning, the security solutions were inherited and applied without significant contributions from distributed to pervasive systems, mainly due to lack of infrastructure support and knowledge between devices. Nowadays, several proposals have been working and improving security concerns about pervasive computing, covering their particulars security requirements.

By way of illustration, three IST European funded projects, MobiLife [14], SECURE (Secure Environments for Collaborations among Ubiquitous Roaming Entities [15]) and TrustCoM [16] have tackled issues on trust and privacy applied to pervasive technologies. These solutions are close related to our work and the security objectives of our project UBISEC.

In MobiLife, the notion of trust is based on the Trust Engine concept, which is composed by User Trust Engine (Policy Decision Policy, PDP) and Context Provider Trust Engine (Policy Enforcement Point, PEP). The specification details the policies can be accessible off-line and can be modified in real time according the preferences of the users.

SECURE project is mainly focused on trust and risk. The trust decisions are established based on a cost probability density function. The inputs of this function are the values provided by a trust calculator and a risk evaluator, broadly speaking, trust values are setting using context and user data. No information about offline access to policies is provided.

TrustCoM proposes an access control model that combines a membership list with the RBAC model. The list contains the mappings between all participants' public keys to their roles and an entry in the registry of management data. The membership list is maintained by a coordinator and distributed to all the participants in the community. In addition, a recent column in [17] references some proposals that address the topic of individual's privacy related to context aware information.

In brief, authentication services in disconnected networks such as ad-hoc networks can be broadly divided into two categories: solutions based on threshold cryptography where the private key of the service is shared among all or a subset of network nodes, like [21] or [22] which propose the construction of a trusted network similarly to PGP where users build trust paths by issuing, storing and distributing certificates. In other hand, there is no unique proposal to provide ad-hoc networks with authorization services. Among some solutions [23-28]: Loong et al. [29] specify authorization policies grouped according to the roles of users, whereas the work [30] proposes to hold in each user device a valid policy certificate and an enforcement module that ensures fulfillment of the policy. These solutions do not guarantee both authentication and authorization for peer-to-peer applications. Furthermore, in general they do not provide support for user mobility and re configurability.

The work [31] introduces the concept of delegation in pervasive environments. In centralized solutions, an individual has the role of the leader of the group where its main task is handling the admission control in ad-hoc group. Delegation sentences must be used to obtain a scalable solution. Therefore, the leader can delegate to other individuals the capability to admin the inclusion of the participants in a group. The mentioned delegation is in fact an identity delegation, that is to say, the leader delegates his identity to selected individuals instead of the appropriated authorization sentences.

Some of the co-authors of this work establish in [32] the concept of controlled delegation of authorization to enhance the drawbacks of identity delegation. Basically, the controlled delegation is possible due to the separation of the authentication and authorization sentences. For this purpose, among others, X509 identity and attribute certificate [1, 13, 33] respectively are employed, as this work suggests too.

In [34] the authors proposed the use of attribute certificate, although they design their own format using the XML language. The main drawback of this solution is XML parsers are computationally more costly to pervasive devices than ASN.1 [35] (the format of X509 certificates). Moreover, the authors employ the identity structure to perform the delegation sentences instead of making use of the defined authorization elements, therefore this could be considered other example of misuse of how establish delegation of duties.

8. Conclusions

In this paper, we presented a ubiquitous authentication and authorization infrastructure, which allows the validation of user credentials in heterogeneous networks where global connectivity can be lost and some services can become temporarily unreachable. Authentication and authorization are provided to users and applications through the combination of traditional PKI and new PMI services, notably thanks to a new trust model and the use of attribute certificates. Several software components are proposed for the users' devices, in order to extend several PKI functionalities to the disconnected mode. This modular infrastructure supports free roaming of users across different administrative domains and network technologies, and it is endowed with reconfiguration capabilities.

We also described a proof-of-concept implementation of the pervasive-PKI developed in the UBISEC project. In the validation testbed we showed the functionality of the pervasive-PKI in the disconnected mode, computing the performance of our implementation.

References

1. ITU-T Recommendation X.509. "Information Technology - Open systems interconnection- The Directory: Public-key and attribute certificate frameworks", 2000.
2. The International PGP Home Page. Avalaible at http://www.pgpi.org/
3. F. Almenárez, A. Marín, D. Díaz, and J. Sánchez. Developing a Model for Trust Management in Pervasive Devices. IEEE Workshop on Pervasive Computing and Communication Security, 2006.
4. A. Josang, "An Algebra for Assessing Trust in Certification Chains,". Proc.. *Network and Distributed Systems Security Symposium (NDSS)*, 1999.
5. D. Ingram. An Evidence Based Architecture for Efficient, Attack-Resistant Computational Trust Dissemination in Peer-to-Peer Networks. Third International Conference on Trust Management (iTrust'05). 2005
6. M. Waseen, R. McClatchey, I. Willers. "A Scalable Evidence Based Self-Managing Framework for Trust Management". *Electronic Notes in Theoretical Computer Science (ENTCS)*. 179:59-73. 2007.

7. J. Doucer. The Sybil Attack. First International Workshop on Peer-to-Peer Systems. Vol. 2429. Pages: 251 - 260. LNCS. Springer-Verlag. 2002
8. A. Twigg, N. Dimmock. Attack-Resistance of Computational Trust Models. In Proceedings of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'03). 2003
9. E. Bruneton, T. Coupaye, M. Leclerc, V. Quéma, and J.-B. Stéfani. The Fractal Component Model and its Support in Java. *Software - Practice and Experience (SP&E), special issue on Experiences with Auto-adaptive and Reconfigurable Systems*, 36(11-12): 1257-1284, 2006.
10. ObjectWeb Consortium. The Fractal Component Framework. Open source software available for download at http://fractal.objectweb.org/
11. J. Forné, J. L. Muñoz, F. Hinarejos, O. Esparza. "Certificate Status Validation in Mobile Ad-Hoc Networks". *IEEE Wireless Communications,* 16(1):55-62. 2009.
12. J. A. Montenegro, F. Moya, "A Practical Approach of X509 Attribute Certificate Framework as Support to Obtain Privilege Delegation,". Proc. *1st European PKI Workshop: Research and Applications. Samos Island, Greece, 2004. Lecture Notes in Computer Science (LNCS) 3093, Springer-Verlag,* pp. 160-172
13. R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 3280, April 2002. http://www.ietf.org/rfc/rfc3280.txt.
14. IST Mobilife project. Avalaible at http://www.ist-mobilife.org/
15. IST SECURE project. http://www.dsg.cs.tcd.ie/dynamic/?category_id=206
16. IST TrustCoM project. Avalaible at http://www.eu-trustcom.com/
17. Péter Boda, et al. "Privacy and trust initial model and interfaces". IST-2004-511607 MobiLife. D24 (D3.4) v1.0.
18. V. Cahill et. al. "Using trust for secure collaboration in uncertain environments." *IEEE Pervasive Computing*, 2(3). 2003.
19. Sye Keoh and Emil Lupu. "An Efficient Access Control Model for Mobile Ad-hoc Communities". 2nd International Conference on Security in Pervasive Computing, Boppard, Germany, 6 - 8 April 2005 Springer-Verlag Berlin.
20. A. Joseph Ed. "Security and Privacy in Pervasive Computing", *IEEE Pervasive Computing*, 6(4):73-75. 2007.
21. L. Zhou and Z.J. Haas, "Securing Ad Hoc Networks". *IEEE Networks*, 13(6):24-30, 1999.
22. S. Capkun, L. Buttyan, and J.P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks". *IEEE Transactions on Mobile Computing*, 2(1):52-64, 2003.
23. J. Al-Muhtadi, A. Ranganathan, R. Campbell, and M. D. Mickunas. "Cerberus: A Context-Aware Security Scheme for Smart Spaces". In Proceedings of IEEE Internacional Conference on Pervasive Computing and Communications (PerCom 2003), pages 489–496, March 2003.
24. M. J. Covington, P. Fogla, Z. Zhan, and M. Ahamad. "A Context-Aware Security Architecture for Emerging Applications". In Proceedings of 18th Annual Computer 190 Security Applications Conference (ACSAC 2002), December 2002.
25. T. Kagal, L. Finin and A. Josh. "Trust-Based Security in Pervasive Computing Environments". IEEE Computer, pages 154–157, December 2001.
26. G. Myles, A. Friday, and N. Davies. "Preserving Privacy in Environments with Location-Based Applications". Pervasive Computing, 2(1):56–64. 2003.
27. A. Tripathi, T. Ahmed, D. Kulkarni, R. Kumar, and K. Kashiramka. "Context-Based Secure Resource Access in Pervasive Computing Environments". In Proceedings of

First IEEE International Workshop on Pervasive Computing and Communications Security(PerSec'04), pages 159–163, March 2004.

28. C. Wullems, M. Looi, and A. Clark. "Towards Context-aware Security: An Authorization Architecture for Intranet Environments". In Proceedings of First IEEE International Workshop on Pervasive Computing and Communications Security(PerSec'04), pages 132–137, March 2004.
29. S. Loong, E. Lupu, M. Sloman, "PEACE : A Policy-based Establishment of Ad-hoc Communities". Proc. *Computer Security Applications Conference*, 2004.
30. W. Jansen, T. Karygiannis, S. Gravila, and V. Korolev, "Assigning and Enforcing Security Policies on Handheld Devices", Proc. *Canadian Information Technology Security Symposium*, 2002.
31. S. Maki, T. Aura, and M. Hietalahti. "Robust Membership Management for Adhoc Groups". In Proceedings of the 5th Nordic Workshop on Secure IT Systems (NORDSEC 2000), Reykjavik, Iceland, 2000.
32. I. Agudo, J. Lopez and J. A. Montenegro. "Enabling Attribute Delegation in Ubiquitous Environments", Mobile Networks and Applications, Vol. 13, n.3, August 2008.
33. S. Farrell and R. Housley. "An Internet Attribute Certificate Profile for Authorization". IETF PKIX Working Group, April 2002. Request for Comments 3281.
34. L Bussard, Y Roudier, R Kilian-Kehr, S Crosta. "Trust and Authorization in Pervasive B2E Scenarios" Information Security: 6th International Conference, ISC 2003. LNCS 2851, pp. 295 – 309, 2003.
35. D. Mundy and D. W. Chadwick, "An XML Alternative for Performance and Security: ASN.1*". IEEE IT Professiona,l* 6:(1):30-36. 2004.