SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks **00**: 1–12 (2010)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.0000

Achieving Privacy and Access Control in Pervasive Computing Environments

Emmanouil Magkos[1]*, Panayiotis Kotzanikolaou[2]

[1]*Department of Informatics, Ionian University, Plateia Tsirigoti 7, Kerkyra, 49100, Greece.*
[2]*Department of Informatics, University of Piraeus, 80, Karaoli-Dimitriou, 18534, Piraeus, Greece*

Summary

This paper focuses on the inherent trade-off between privacy and access control in Pervasive Computing Environments (PCEs). On one hand, service providers require user authentication and authorization for the provision of a service, while at the same time end users require untraceability and unlinkability for their transactions. There are also cases where the anonymity of a specific credential must be revoked and a real identity be traced, in order to establish accountability. We analyze privacy and security requirements for PCEs and we show that existing privacy-preserving access control schemes do not fully satisfy these requirements. Then we propose two approaches towards privacy-preserving access control in PCEs. Our goal is twofold: (a) to enhance privacy by achieving untraceability and unlinkability even against malicious insiders and (b) to enhance security by achieving conditional traceability of user credentials, and if possible, non-repudiation of evidence concerning the user's participating in a transaction. Finally, we analyze and compare the proposed schemes against existing schemes. Copyright © 2010 John Wiley & Sons, Ltd.

KEY WORDS: Pervasive computing environments, Privacy-preserving access control, Unlinkability, Accountability, Conditional traceability

1. Introduction

One of the most exciting aspects of the not so far future will involve the integration of computing and communication into a mobile and dynamic environment. Within a *Pervasive Computing Environment* (PCE), a typical scenario involves mobile users that use low-cost, handheld devices in order to have seamless access to different kinds of value-added services, anytime and anywhere [1], typically by connecting to wireless (local or cellular) access points. Because of the unique characteristics of the dynamic PCE environment, such as user mobility, loose physical boundaries and transient interactions with devices of undefined trust, security will naturally be required as an inherent property by consumers and organizations. Specifically, security issues involve entity authentication and access control, as well as message confidentiality, communication integrity and service availability [2, 3].

On the other hand, the protection of sensitive data such as identity and other context information is also seen as an important criterion for large-scale deployment of PCEs [2, 4, 5]. With the advent of pervasive devices and technologies (*e.g.*, wireless sensors, fusion, RFIDs), individuals are wary of a "Big Brother"-type threat, where their transactions could be monitored without their consent [6, 7]. For

*Correspondence to: Department of Informatics, Ionian University, Plateia Tsirigoti 7, Kerkyra, 49100, Greece. E-mail: emagos@ionio.gr

Prepared using secauth.cls *[Version: 2008/03/18 v1.00]*

example, a supposedly anonymous transaction may be traced back to the identity of an unsuspected user; or, different transactions between the system and the user may be linked in order to build a profile.

Naturally there is a tradeoff between *user privacy* and *access control* in PCEs [8, 9]. From the user's point of view, a *privacy adversary* should not be able to: (a) trace the real identity of the user, (b) link different sessions between the user and the system, (c) obtain context information (*e.g.*, location, time and duration, type of service). On the other hand, the need for access control is threefold. Service providers may need message, entity or context authentication in order to: (a) authorize access to a service (*e.g.*, for billing purposes, for abuse prevention and detection), (b) provide personalized, context-aware services, (c) trace back an identity for accountability or liability (*e.g.*, in case of service abuse, unlawful acts, for privilege revocation). Furthermore, due to the limitations imposed by resource-constrained environments, security and privacy mechanisms should be efficient in terms of storage, communication and computation. As a result, access control with privacy preservation in PCEs is still an open research area [10, 2, 4, 11, 6, 12, 13].

Recent work [8, 13, 14, 15] uses blind signatures [16] and cryptographic hash chains [17], in order to provide users with untraceable credentials for efficient access control. Although the use of hash chains simplifies key management and provides protection against replay attacks in a very efficient way, it fails to provide unlinkability between different transactions of a user, against malicious insiders.

Our Contribution

In this paper we discuss requirements for privacy and access control in PCEs and show how a recent scheme for PCEs [8, 13], proposed by Ren and Lou [13] –hereinafter called the *RL scheme*, does not fully satisfy such requirements against a global passive adversary. We also discuss why the requirements for accountability and non-repudiation of the RL scheme, are weak. In addition, we review a recent attack against the RL scheme, proposed by in [15], and argue that this attack is not practical. To balance the requirements for privacy against malicious insiders (users, front-end and back-end entities), with the security requirements for access control and accountability, we adopt an efficient hybrid approach that combines both public-key and symmetric key credentials. For conditional traceability a trusted entity keeps a record of the pseudonyms and the corresponding real identity of the device. Finally, we analyze and compare the proposed schemes against existing schemes.

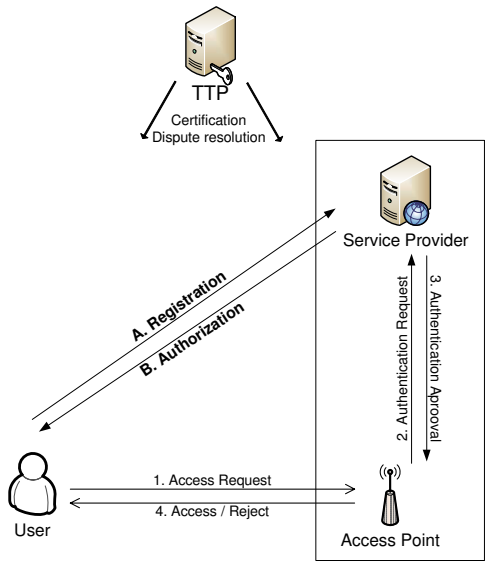

Fig. 1. The system model

2. Privacy and security requirements

In a PCE, a mobile user dynamically accesses a service among a list of available service types. We consider users, front-end entities, back-end authorities, and a Trusted Third Party (TTP) (Figure 1):

- *Users* are equipped with hand-held devices and request access to different kinds of services at anytime and from anywhere.
- *Front-end entities*: Typically these are wireless access points (AP) that handle the communication with the user, collect the service request messages and mediate between the user and a back-end authority.
- *Back-end authorities* involve an application Service Provider (SP) and an authentication server (AS). The task of the SP and the AS is to provide the service data to authorized users. For simplicity we will assume that the SP will also act as an AS, since usually they are controlled by the same entity. However, users may access the same SP through various APs, controlled by different entities (such as network operators).
- *The TTP* is usually an offline authority that is invoked in exceptional circumstances (*e.g.*, certification, dispute resolution, anonymity revocation).

DOI: 10.1002/sec

2.1. Threat model

In our threat model we consider both passive and active adversaries. The passive adversary monitors all communications within the network to extract private information. This information may be used to link past or future message exchanges in order to track and/or trace users. The adversary may also compromise APs and/or the SP(s), and extract their logs in order to facilitate tracking/tracing.

An active adversary will attempt to modify messages in transit, replay past messages or fabricate messages at will, in order to impersonate system entities or simply disrupt the network.

We assume that the adversary will not exploit any weaknesses in the underlying public-key or symmetric cryptographic algorithms. We also assume the communication channel between the APs and the SP(s) to be secure. Finally, as in [13] we assume that users are capable of manipulating the source address of layer 2 (MAC) frames, or else the unlinkability property is trivially defeated at the access point.

2.2. Privacy requirements

2.2.1. Untraceability

Under this requirement, no unauthorized entity (or a coalition of unauthorized entities) should be able to trace the real identity of the user, unless the user or system policy explicitly permit it.

Untraceability by itself cannot provide adequate privacy protection in PCEs. If a set of distinct credentials can be linked to the same anonymous entity, then a customer profile can be built and this is considered a privacy violation. In this case, and in order to completely undermine privacy, the adversary will only have to trace one particular link of this chain (*e.g.*, after the customer uses a credit card, with use of a camera, physical pursuit etc).

2.2.2. Unlinkability

Under the unlinkability requirement (also known as *tracking protection*), no unauthorized entity, external or internal to the system (*i.e.*, other users, APs, SPs) should be able to link different sessions of the user, unless the user or system policy explicitly permit it.

It is important to note that, as stated in [18], anonymity-protected communications are unlinkable as long as application content does not enable linkage. Another interesting point concerning unlinkability, is that there may be cases where unlikability may be impossible or undesirable, as in reputation-based systems [19]. Or, back-end authorities may need to link information for supporting infotainment or value-added (*e.g.*, location-based) services. In the above cases, the linkage may or may not necessarily require tracing the real identity, *e.g.*, pay services are not always traceable [20].

2.3. Security requirements

2.3.1. Mutual authentication

Communication messages between system entities should be authenticated and integrity-protected. A SP will require user authentication in order to prevent service abuse, while users may also require server authentication, in order to protect themselves from spoofing attacks. Note that user authentication and access control seem to contradict with user privacy, since if a user is completely anonymous, SPs will be concerned with service abuse.

2.3.2. Unforgeability

It should not be easy for outsiders or insiders, not having valid credentials for a particular service, to prove possession of a valid credential.

2.3.3. Conditional traceability

In case of service abuse, *e.g.* illegal actions, abnormal access pattern of the user or when a credential is linked to an unlawful act, it should be possible to have an accountability mechanism for revoking the anonymity of a specific credential and tracing the identity of a real user, in order to establish accountability. Typically, anonymity revocation will be an off-line protocol, where a SP and a TTP, given credential and transaction information, will be able to trace the real identity of a user. The SP can then take appropriate measures, *e.g.*, blacklisting a user. Note however, that it should not be easy to abuse this capability (*e.g.*, in order to impersonate a user).

2.3.4. Non-frameability

It should not be easy for outsiders or insiders to successfully impersonate or frame an honest user. In addition, no entity should be fully trusted: even a TTP in collaboration with the SP, should not be able to create a transaction that opens to an honest user. A special case of non-frameability is exculpability: the TTP and/or the SP, should not be able to give a usable credential of one user to another user.

DOI: 10.1002/sec

2.3.5. Non-repudiation

A related requirement is *non-repudiation*, under which it must be possible to produce evidence regarding an entity participating to a transaction, and then to protect against a user's false denial of having participated to a transaction [21].

2.4. Efficiency requirements

For functional and non-functional requirements that are not strictly security-related, we refer to [3] for a list. For efficiency, we note that any privacy-preserving access control scheme should be efficient in terms of:

1. *Computation.* We require efficient user registration and service access protocols, with as few public operations as possible.
2. *Storage.* Users obtain and store a minimum necessary amount of credential information.
3. *Communication.* The number of passes and bits that are communicated should be kept as low as possible.

3. Related work

Security and privacy preservation in pervasive environments is not a new topic –see for example [3, 6] for general security requirements and challenges, [2] for privacy definitions, [5] for a survey of privacy enhancing technologies. The privacy vs access control tradeoff has also been explored in the literature [22, 23] and particularly in the context of wireless mesh networks (*e.g.*, [24, 25, 26, 27, 28]), where it was suggested that a long list of short-lived pseudonyms is generated during registration in such a way that the pseudonyms cannot be linked to each other by non-authorized entities. In [24, 26] for example, these credentials are anonymous public keys for verifying digital signatures, validated by a trusted entity. In [27] each public key is validated during registration by using blind signatures [16]. Such solutions, if adopted in the PCE context, should be carefully designed to avoid increasing the complexity of the user registration phase and the computational, storage and communication cost for the handheld devices.

Recent work also points out that the use of independent pseudonyms is not a panacea [4, 24, 29]. For example, changing identity does not always guarantee unlinkability in telematics or indoor PCEs, where a global adversary has access to context (*e.g.*, spatiotemporal) information and performs traffic analysis against a mobile user. The preservation of context privacy in pervasive environments has been explored in some recent works (*e.g.*, [4, 30, 5, 7, 31], also refer to [32] for a survey). For example, to deal with adversaries that track moving nodes by using spatiotemporal information, in [24], a pseudonym-changing algorithm based on spatial and temporal criteria was proposed. In [25] the use of a random silent period was suggested to deal with correlation attacks and diffuse the spatiotemporal redundancy when users change pseudonyms. Based on the idea of Chaum [33], the authors in [4] propose that pseudonyms should be changed within a *MIX zone*, in order to hide the identity of a user belonging to a group of users with similar characteristics. In another work [9] it is suggested that user's sensitive (context) data are encrypted by the user and stored at server side to ensure context privacy.

In a line of works that began with Camenish and Lysyanskaya [34], and extended in [35, 36] an anonymous credential system is built by defining a signature scheme, a commitment scheme and (a) a protocol for obtaining a signature on a committed message, (b) a protocol for proving knowledge that the contents of a commitment have been signed, and (c) a protocol for proving that a pair of commitments commit to the same value. A related class of works consists on anonymous group identification (or, pseudonymous identification) schemes (*e.g.*, [37, 38]), which basically are interactive zero-knowledge (ZK) identification techniques, where a user derives multiple cryptographic pseudonyms from a single master secret. While basic properties such as anonymity (*i.e.*, untraceability and unlinkability) and unforgeability criteria are inherently satisfied in ZK identification protocols, without any trust assumptions, the computation and communication costs of such constructions are high.

In this paper we focus on a recent line of efficient protocols for privacy-preserving access control in PCEs [8, 19, 13, 9, 14, 15]. A representative scheme of this category is the RL scheme [13], which is considered as the first attempt to provide a secure communication model for privacy-preserving access control in PCEs. The RL scheme uses *blind signatures* [16] and cryptographic *hash chains* [17] at the application layer in order to provide mutual authentication while preserving privacy against malicious outsiders. In [14, 15] the RL scheme is tweaked to increase performance while in [15], an impersonation attack against the RL scheme, hereinafter called the *Li et al attack*, was described and addressed (see also section 3.2.3). Finally, the work in

DOI: 10.1002/sec

[9] where non-unique temporal IDs are issued by the access point to system users, achieves authentication and unlikability against back-end authorities, but fails to establish accountability and untraceability against front-end entities.

3.1. The RL scheme

The RL scheme [13] contains a set of protocols executed between a Service Provider (SP) and a mobile user (*e.g.*, Alice) that communicate via an access point (AP). The claimed security services in [13] are: mutual authentication, anonymity, unlinkability, non-repudiation, and accountability. We recall the two distinct phases of the protocol (Figure 2), while avoiding a few (non-crucial) details and slightly changing the notation.

During a *user registration* phase, Alice registers as a legal user, obtains an authentic public-key certificate $Cert_A$ for her real identity, and the public key PK_{SID} of a service she is entitled to use. For a specific service SID, Alice chooses a random seed as the anchor value C^0 of a credential chain, and hashes it n times to compute the end value $C^n = H^n(C_0)$. Then, Alice uses the *blind signature* primitive [16] and prepares a blinded version of C^n as $C_A = \{r'_A\}_{PK_{SID}} \times C^n$, where r'_A is a random nonce and $\{\}_{PK_{SID}}$ denotes encryption with the public key of service SID. Alice submits C_A, her identity A and certificate $CERT_A$, together with SID, to the SP. The SP verifies that Alice is eligible, signs C_A with the private signature key for SID and sends the blindly signed message $[C^n]_{SK_{SID}}$ to Alice, who unblinds it by dividing with r'_A. What Alice finally gets is a valid but untraceable ticket for n future service accesses. In order to access different services, Alice may generate and authorize several different credential chains.

At any time, a *service access* phase can be run between Alice, an Access Point (AP) and the Service Provider (SP). The phase can be viewed as the execution of two different sub-protocols:

- The first sub-protocol achieves mutual entity authentication between Alice and the SP, mediated by the AP who just relays messages. For the j-th service access, $1 \le j \le n-1$, Alice chooses a challenge r_a, concatenates it with a credential C^j (or with $[C^n]_{SK_{SID}}$, if $j = 1$), encrypts them with the public key PK_{SP} of the SP, and sends the result, together with the service SID, to the SP, via the AP. The SP decrypts, uses hash chain verification to check whether C^j is a fresh, valid credential for the specific service type, and returns r_a and C^j to the AP, through a secure channel.
- The second sub-protocol achieves authenticated key agreement between Alice and the AP, with implicit key authentication. The AP chooses a random challenge r_p, and computes two session keys for encryption and message authentication, namely $K_{ap} = H(C^j, r_p, r_a, 0)$ and $K'_{ap} = H(C^j, r_p, r_a, 1)$. The AP encrypts r_a and its identity P with K_{ap} and sends $r_p, [r_a, P]_{K_{ap}}$ to

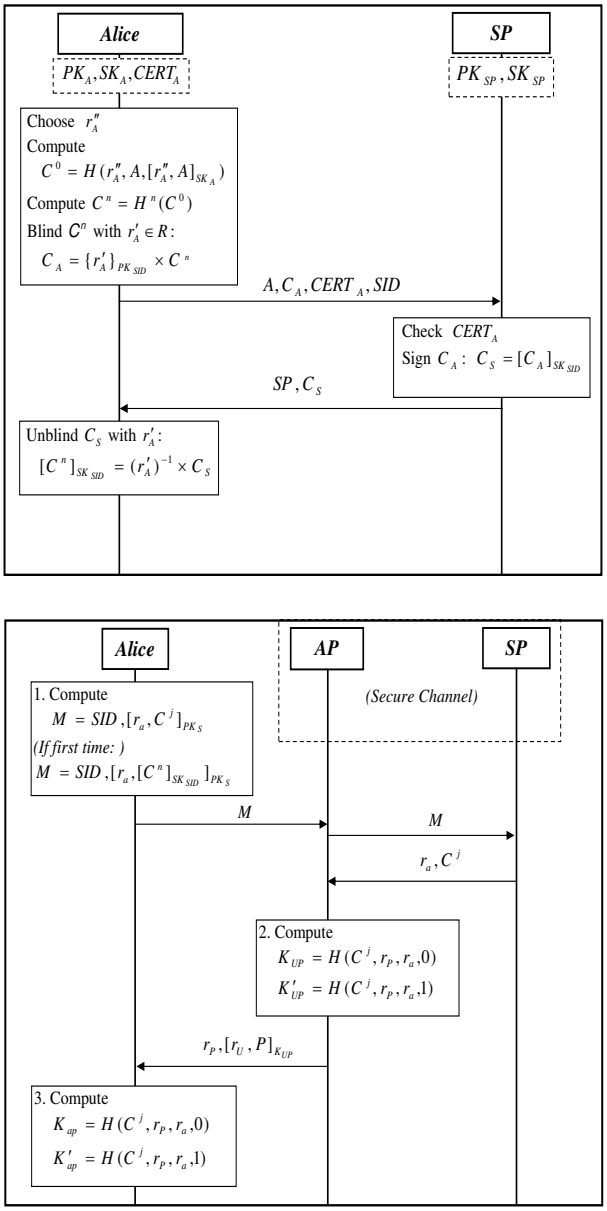

Fig. 2. Registration and user access in the RL scheme

DOI: 10.1002/sec

Alice. From that point, communication between Alice and the AP will be encrypted and authenticated with K_{ap} and K'_{ap}.

To deal with situations where a credential C^j may be compromised or stolen, the authors in [13] suggest that the anchor value is the output of a cryptographic hash function whose pre-image also contains a message that is digitally signed with Alice's private key SK_A. That is, $C_0 = H(r''_a, A, \{A, r''_a\}_{SK_A})$, where r''_a is a nonce selected by Alice during registration. Then, in case of a dispute, a *dispute resolution* protocol is run between a TTP, the SP and Alice. The SP presents the TTP with the disputed credential and the real Alice will reveal to the TTP a valid pre-image value of C_0, *i.e.*, her authentic signature on the nonce r''_a. The claimed property for the above enhancement is *non-repudiation* for the system users [13].

3.2. Vulnerabilities of the RL scheme

In the following we summarize several privacy and security vulnerabilities of the RL scheme. Specifically we will show how the RL scheme fails to satisfy the unlinkability and untraceability criteria within our threat model, while its accountability and non-repudiation assurances are also weak.

3.2.1. Privacy vulnerabilities

Weak unlinkabilty against the APs. The remarks below also apply to [8, 14, 15]. Assume that an AP "sees"[†] an authorized credential C^j of Alice for the epoch j, and that at some future epoch i, where $i > j$, Alice uses AP to access the same service using the anonymous credential C^i. If AP keeps a database of authorized credentials, it will effectively link the two transactions by performing $j - i$ hashing operations. Similarly, two or more cooperating APs will be able to jointly link some or all of Alice's transactions.

Weak unlinkabilty against the SPs. A SP will be able to link all transactions for all services that it offers, by efficiently performing hashing operations on the authorized credentials and checking whether there is a match with any value stored in its database. Similarly, two or more cooperating SPs will be able to jointly link some or all of Alice's transactions.

[†]This attack assumes that the AP has application-level capabilities. Typically, a compromised AP will forward traffic data to a capable adversary.

Strong trust assumptions for the TTP. During dispute resolution for a challenged credential, Alice proves her identity by digitally signing a unique message, and also reveals the anchor value C_0 for her credential chain. The side-effect is that the TTP will be able not only to trace Alice's identity but also to re-construct all components C^j, $0 \leq j \leq n-1$, of the hash chain of credentials. These may be used to link and trace Alice's past and future transactions for a given service, not only the disputed one.

3.2.2. Security vulnerabilities

Weak accountability. Accountability is not well supported in the RL scheme. There is not a way for the SP and/or the TTP, when they are given a specific credential and transaction information, to work off-line and trace the real identity of the owner of the credential.

Weak non-repudiation. The dispute resolution protocol of [13] will expose an adversary that has stolen a credential from a valid user, as he will not be able to demonstrate knowledge of a valid pre-image of the anchor value C^0. However, a repudiation attack may also be possible, where for example a valid user who used an authentic credential during a transaction chooses for some reason to falsely deny participation in this transaction [39].

3.2.3. The Li et al (impersonation) attack [15]

We review an impersonation attack, described in [15], against the RL scheme. The attack is summarized as follows: In Step 1 of the user registration phase (Fig. 2), an adversary (hereinafter called Mallory) computes a fake credential $C'^0 = H(r''_M, A', \{A', r''_M\}_{PK_{A'}})$, where A' is some valid user's identity and r''_M is a nonce. Mallory computes the end value $C'^n = H^n(C'^0)$ and the blinded message $C_{A'} = \{r'_M\}_{PK_{SID}} \times C'^n$. He then sends $C_{A'}$, the identity A' and the certificate $CERT_{A'}$, together with SID, to the SP. In [15] it is said that the SP will only check the validity of the certificate $CERT_{A'}$ and if it is correct the SP will authorize the submitted credential. Then, Mallory will unblind $C_{A'}$ as normally and use the credential chain for n service accesses, by impersonating the user A'.

We argue that the Li et al attack cannot be seen as a practical attack against the RL scheme. Although not explicitly stated in [13], the attack is trivially defeated if, during the user registration phase, the

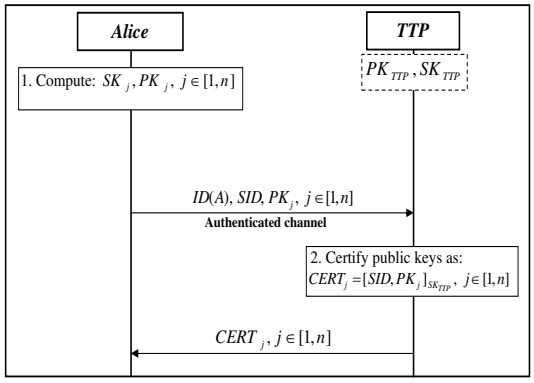

Fig. 3. Basic scheme: User registration phase

user-SP channel is assumed authenticated. This is an assumption made by many security models (*e.g.*, for applications such as e-payment [40] or e-elections [41]) that are based on blind signatures to authorize a set of anonymous credentials. Under this assumption, the attack would not be applicable. A trivial way to explicitly turn the user-SP channel of the RL scheme into an authenticated one, is by combining the use of certified signature keys with the SSL/TLS protocol. This will not violate the privacy of the authorized credentials, as C^n is blindly signed by the SP.

4. Enhancing privacy and access control in PCE environments

In this section we propose two approaches for privacy-preserving access control in PCEs. Our motive is twofold: (a) to enhance privacy by achieving untraceability and unlinkability even against malicious insiders and (b) to enhance security by achieving conditional traceability of user credentials, and if possible, non-repudiation of evidence concerning the user's participating in a transaction. The solutions presented below are based on existing protocols proposed in the literature for various applications and technologies. Therefore we do not focus on formal security and efficiency analysis, but we rather discuss how these solutions can be applied in order to provide the required privacy and security properties.

The first approach is trivial and it is based on uncorrelated public keys. Our second approach, which is the actual proposed scheme, extends the basic scheme into a hybrid variation, in which uncorrelated public keys are balanced, for efficiency, with the RL scheme.

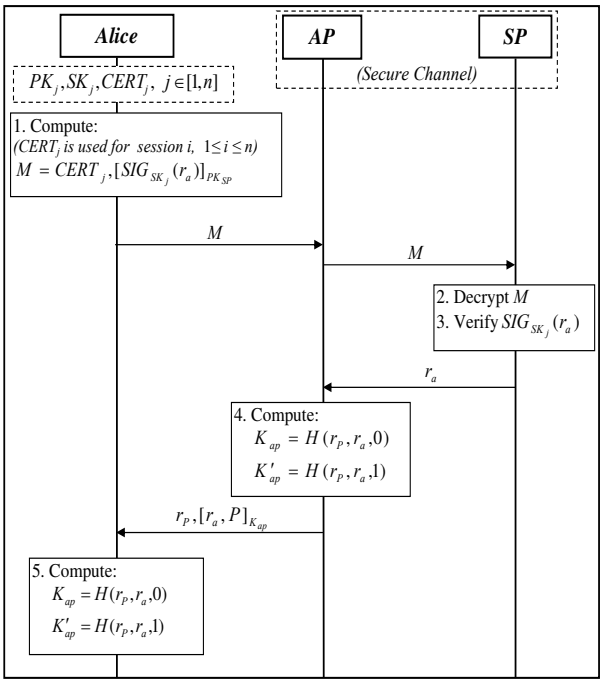

Fig. 4. Basic scheme: User access phase

4.1. A basic scheme with unlinkable pseudonyms

We propose a basic scheme that is based on lists of short-lived, uncorrelated pseudonyms, in order to achieve full user unlinkability. Specifically any handheld device A is installed with a list of anonymous public/private signature key pairs (PK_j, SK_j) and their corresponding certificates $CERT_j$, where $1 \le j \le n$. Each public key will be used for a different service access. For conditional traceability and accountability, a TTP is involved, during the user registration. Below we describe the registration and the user access protocols.

We assume that all the communication during the user registration protocol is performed through an encrypted and authenticated channel. Otherwise, it is possible for an attacker to intercept the communication and/or impersonate a user, in order to steal the credentials belonging to the user.

4.1.1. User registration

Each user, say Alice, participates in the registration protocol with the TTP as shown in Figure 3, in order to issue its anonymous public key certificates. The user will generate n independent anonymous public/private signature key pairs (PK_j, SK_j), $j \in$

DOI: 10.1002/sec

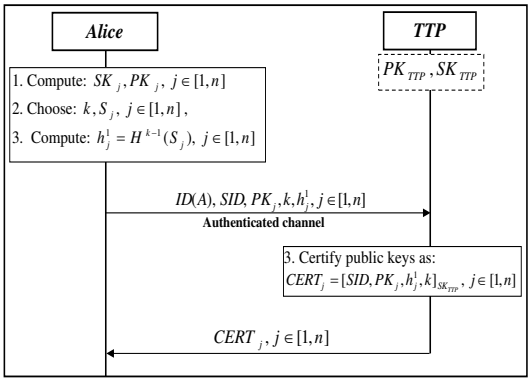

Fig. 5. Hybrid scheme: User registration phase

$[1, n]$. Then, the user will send a certificate request to the TTP, along with its real identifier A, the service id SID and the independent public keys $PK_j, j \in [1, n]$. After the TTP has authenticated the user, it will certify the anonymous public keys and will generate n user certificates $CERT_j = [SID, PK_j]_{SK_{TTP}}$ for n anonymous service accesses.

4.1.2. User access

In order to access a service with identifier SID, (Figure 4) Alice will use an anonymous certificate issued by the TTP for that service, say $CERT_j$. For full unlikability against access points and service providers, each certificate is used for a single service access. The user signs a random nonce r_a with the private key SK_j, encrypts the signature with the public key of the SP and sends the result message M to the SP via the access point AP, along with the corresponding certificate. The SP will decrypt the message and verify the signature. If the verification is successful, the SP will forward r_a to the AP. Then the user and the AP will compute the common keys K_{ap} and K'_{ap} as in the RL scheme.

Remark: reducing the trust assumptions

In order to reduce the trust assumptions for the TTP, the basic scheme could be trivially modified so that the TTP blindly signs the public keys of the user, as in the user registration phase of the RL scheme. In this case, the obtained key pairs and the corresponding certificates would be completely anonymous and it would not be easy even for a coalition between the TTP and the SP to undermine the privacy of the user. Under this approach however, conditional traceability would be difficult to achieve, or it would require inefficient escrow approaches (*e.g.*, [27]).

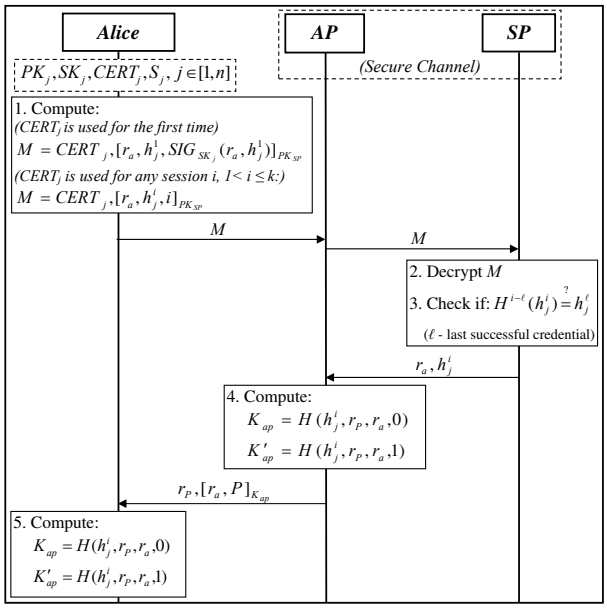

Fig. 6. Hybrid scheme: User access phase

4.2. A hybrid scheme

In view of the costs associated with registration, key storage and signature generation, we extend the basic and adopt a hybrid approach, where both public-key and symmetric-key credentials are combined in order to balance unlinkability with efficiency. For conditional traceability, the TTP keeps a record of the pseudonyms and the corresponding real identity of the device. Below we describe the registration and the user access phases.

4.2.1. User registration

Each user, say Alice, participates in the registration protocol with the TTP, in order to issue her pseudonymous certificates. Specifically, Alice generates n independent private/public key pairs $(PK_j, SK_j), j \in [1, n]$. For each pseudonym j, Alice also chooses a random seed S_j and generates a hash chain $h_j^1, h_j^2, ..., h_j^k$, where k is a previously agreed parameter, $h_j^k = S_j$, and $h_j^i = H^{\ell-i}(h_j^\ell)$ with $i < \ell$. Each element of the chain can be used in reverse order for up to k service access sessions. Then, Alice sends a certificate request to the TTP, along with her real identifier $ID(A)$, the service id SID, the public keys PK_j, the end values h_j^1 and the parameter k. The TTP authenticates Alice, and issues n pseudonymous certificates $CERT_j, j \in [1, n]$. The TTP also keeps records of those certificates and Alice's identity $ID(A)$, for conditional traceability.

At the end of user registration, Alice obtains n unlinkable pseudonyms, to be used for up to $n \times k$ sessions with the SP, since each pseudonym can be used for up to k sessions.

4.2.2. User access

When engaging to a new transaction with a service SID, Alice uses a pseudonymous certificate issued by the TTP for that service, say $CERT_j$. If this is the first session where $CERT_j$ is used, Alice chooses a random nonce r_a and computes a signature[‡] on r_a and the end value h_j^1 of the hash chain, using the private key SK_j. Or, if $CERT_j$ is used for the i-th session ($1 < i \leq k$), Alice encrypts r_a, the i-th chain element h_j^i and the index i with the public key of the SP. Alice will send the encrypted message, along with the corresponding pseudonymous certificate to the SP via the AP. The SP will decrypt the message, verify the signature (in case this is the first session) and perform hash chain verification on h_j^i. If the verification is successful, the SP will forward h_j^i and r_a to the AP. Then the AP and Alice will establish the common keys K_{ap} and K'_{ap}, as in the RL scheme. For the next sessions (up to a total of k sessions), and until Alice chooses to update her pseudonym, message authentication from Alice to the SP is performed by using the chain elements in reverse order, $h_j^{i+1}, ..., h_j^k$.

4.2.3. Key management

The credentials obtained during the user registration phase, can be distributed to the handheld devices either *statically* or *dynamically*. In the static mode [24] a number of pseudonymous certificates are pre-loaded to the device offline *e.g.*, by the TTP. In the dynamic mode, the certificates are updated by executing an online protocol between the device and a TTP periodically or when needed [43].

At first glance, the lifetime of a certificate should be (relatively) short and Alice could use a different pseudonymous public key for each interaction with the SP (as in the basic scheme). In reality, some transactions can be linked anyhow at the application layer [18], because of context-related information; for example, when the user repeatedly connects via the same (non-trusted) AP to access services, privacy can be difficult to achieve. For this reason, it is possible to stretch the use of each of the n pseudonyms up to k linkable user-SP sessions. During that time, the more efficient hash chain authentication can be used, as described earlier in the user access phase. Typically, the mobile user will change a pseudonym every time it connects to the SP through a different AP.

Ultimately, a best strategy for a mobile node to avoid being tracked, in view of a global adversary that performs traffic analysis and has access to spatiotemporal information, is to make itself indistinguishable from other nodes by hiding in the crowd: this means that pseudonyms should be changed within a *MIX zone* that hides the identity of a user belonging to a group of users with similar characteristics [4].

5. Protocol analysis

We examine the proposed schemes against the privacy, security and efficiency requirements which were set in Section 2. Since the hybrid scheme extends the RL scheme, we also compare the proposed schemes against the RL scheme. A summary of the results is shown in table I.

5.1. Privacy analysis

5.1.1. Untraceability

Untraceability in the RL scheme is based on the blind signature primitive, whose security is underlied by the RSA function [16]. In the basic and hybrid schemes, the user obtains a validated list of pseudonyms, that can not be traced by (a coalition of) the AP and SP. Untraceability is conditional in the sense that, under well-defined conditions, the TTP is able to trace the identity of the user, if the SP provides the messages sent by a user, during the user access phase.

5.1.2. Unlinkability

In the RL scheme user transaction linkability is protected only against outsiders. Furthermore, during dispute resolution the TTP is able to link all user transactions simply by performing hash operations to the anchor value (see also Section 3.2.1). From the proposed schemes, the basic version provides the strongest unlinkability assurances against a malicious SP, since during registration the user obtains from a trusted center a validated list of information-theoretically unlinkable pseudonyms and uses a different anonymous certificate for each access to a service. Unlinkability also holds against an AP or a coalition of APs for the same reasons.

[‡] *e.g.*, using a conventional digital signature (*e.g.* ECDSA [42]).

DOI: 10.1002/sec

The hybrid scheme is a trade-off between privacy and efficiency, since it provides unlinkability against SPs and APs, only if a different anonymous public key is used for a different transaction with the same back-end entity. Within the use of a particular public key PK_j, a malicious SP or AP is able to link transactions performed with the credentials h_j^1, h_j^2, ..., h_j^k, in the same way as in the RL scheme. However, it is possible for a user to preserve transaction unlinkability against a particular AP or SP, by accepting the extra cost of public key operations and by using a different anonymous certificate.

5.2. Security analysis

5.2.1. *Mutual authentication*

During user access, all protocols (RL, basic and hybrid) achieve mutual authentication between the user and the SP, mediated by the AP. In all user access protocols (*e.g.*, Fig. 6), the user encrypt a random number with the public key of the SP, and verify that this random number is included in the message that receives later from the AP. Furthermore, in the basic and hybrid schemes, user access requests are digitally signed with a private key and the SP verifies that the accompanying pseudonymous certificate is valid and that it contains the correct public key. In the RL and hybrid schemes, symmetric authentication of the user to the SP is also based on the security of the cryptographic hash function. Finally, all schemes also achieve authenticated key agreement between Alice and the AP, with implicit key authentication.

5.2.2. *Conditional traceability*

Accountability is not supported in the RL scheme. Indeed, since the TTP signs blinded user credentials during the registration phase, even if the SP and the TTP cooperate, it is not possible to set a user accountable for a possible illegal action, since there is no way to trace the real identity of the user. In both the proposed schemes conditional traceability is achieved. If a user must be set accountable for a transaction, the TTP can be employed, under assumingly well-defined conditions, in order to reveal the real identity of the user. Typically when there is a dispute, the SP will submit a set of transaction data and misuse evidence to the TTP, who will look up in the database for a matching between an pseudonymous certificate and the real identity of a user.

5.2.3. *Non-frameability*

In contrast with [39], it is not easy even for the TTP to impersonate system users: The TTP does not know the private signature keys of the users, nor is able to invert the cryptographic hash function and compute the next element of the credential chain. The same arguments also apply to the unforgeability criterion.

5.2.4. *Non-repudiation*

The proposed schemes, in contrast to the RL scheme, achieve non-repudiation of evidence. Since in each transaction, the user signs an access request message, a user cannot later deny participation to a specific transaction.

5.3. Efficiency analysis

The tradeoff for the privacy and security enhancements offered by the proposed schemes is the increased storage, processing and communication cost for the user. We examine both the registration and user access phases. For the computation costs, we compare the number of required (expensive) modular exponentiations. For the communication costs, we measure the number of exchanged messages, as well as the expected message length (in bytes). Finally for the storage costs we measure the number of bytes required for the storage of the protocol parameters. In order to calculate message and storage size, we assume that the used identifiers and nonces are 64 bit, the hash functions provide output of 128 bit, the public key modulo is 768 bit[§] and each certificate requires 1024 bit.

The RL scheme is the most efficient in both registration and user access phases. During the registration phase, the RL scheme involves one modular exponentiation, in order to blind the credential C^n. The proposed schemes are more expensive, since the basic scheme requires n exponentiations and the hybrid scheme n/k exponentiations (assuming that credentials are registered for n different user transactions). All three schemes require the exchange of two messages during the registration, but with different message length, as shown in table I. In terms of storage costs, the RL scheme outperforms the proposed schemes, since it requires the storage of a single public/private key and certificate, in comparison with n and n/k.

[§]Since the public keys have short time life and are used for a single or for a small number of transactions, for better performance we avoid using larger public key parameters.

DOI: 10.1002/sec

		RL scheme	Our scheme (basic)	Our scheme (hybrid)
Privacy Requirements	*Untraceability*	✓	✓	✓
	Unlinkability	APs and SPs can link users by performing hashing operations (Section 3.2.1)	✓	✓
	TTP implied level	The TTP is able to link all user transactions by hashing the anchor value (it is trusted not to)	A trusted TTP is assumed	A trusted TTP is assumed
Security Requirements	*Mutual Authentication*	✓	✓	✓
	Conditional traceability	Not possible for the SP and/or the TTP to resolve a credential to a user (Section 3.2.2)	✓	✓
	Non-repudiation	Possible for a user to deny participation to a transaction. (Section 3.2.2)	✓	✓
	Non-frameability	✓	✓	✓
Efficiency (Registration)	*Computation*	1 EXP	n EXP	n/k EXP
	Communication	2 messages (2048 bit)	(128 + n * 1792) bit	(192 + n/k * 1856) bit
	Storage	1 public/private keys and certificates	n public/private keys and certificates	n/k public/private keys and certificates
Efficiency (User Access)	*Computation*	1 EXP	2 EXP	1 EXP
	Communication	2 messages (1088 bit)	2 messages (2048 bit)	2 messages (1408 bit)
	Storage	256 bit	256 bit	256 bit

Table I. A comparison of the schemes

During the user access phase, the hybrid scheme has almost the same computation cost per user transaction by requiring one modular exponentiation (public key encryption), while the basic scheme requires two exponentiations, considering the cost for the signature required in each transaction. The communication cost for the three schemes is 1088 bit (RL scheme), 2048 bit (basic scheme) and 1408 bit (hybrid scheme). The storage cost per user access is the same in all schemes, since only the keys K_{ap} and K'_{ap} need to be stored.

Since in Pervasive Computing Environments the mobile devices may have very limited computation and storage capabilities, it may be required to further decrease the computation and storage costs on the mobile device side. For this reason, optimized implementation scenarios may be considered, especially in the case of the basic scheme. A possible optimization is that the user registers with the Service Provider through a full functional device (e.g. a laptop), which is capable of performing expensive exponentiations and has no storage limitations. Then, the user may load her low-capabilities pervasive device with credentials for a limited number of use from the full functional device, through a secure local connection. For example, in the basic scheme, the user may have pre-computed step 1 of figure 3 for k user accesses and load the device with the corresponding values M, r_a for k unlinkable transactions with SP. In this way, the user may balance the computation and storage requirements, even for lower capabilities mobile equipment.

6. Conclusions

Privacy and access control in PCEs pose some interesting challenges. In this paper we defined a threat model as well as requirements for privacy and security in pervasive computing environments, reviewed the related work on the subject and shown that a recent scheme, the RL scheme has privacy and security vulnerabilities under our threat model. Finally we presented two alternative schemes, a basic and a hybrid scheme, for privacy-preserving access control in PCEs and discussed their security and efficiency.

References

1. Weiser M. The computer for the 21st century. *ACM SIGMOBILE Mobile Computing and Communications Review* 1999; **3**(3):3–11.
2. Al-Muhtadi J, Campbell R, Kapadia A, Mickunas M, Yi S. Routing through the mist: privacy preserving communication in ubiquitous computing environments. *22nd International Conference on Distributed Computing Systems*, IEEE, 2002; 74–83.
3. Campbell RH, Al-Muhtadi J, Naldurg P, Sampemane G, Mickunas MD. Towards security and privacy for pervasive computing. *ISSS International Symposium on Software Security*, 2002; 1–15.
4. Beresford A, Stajano F. Location privacy in pervasive computing. *Pervasive Computing, IEEE* Jan-Mar 2003; **2**(1):46–55.

DOI: 10.1002/sec

5. Ackerman MS. Privacy in pervasive environments: next generation labeling protocols. *Personal Ubiquitous Comput.* 2004; **8**(6):430–439.
6. Ranganathan K. Trustworthy pervasive computing: The hard security problems. *PERCOMW '04: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, IEEE Computer Society: Washington, DC, USA, 2004; 117.
7. Gorlach A, Heinemann A, Terpstra WW. Survey on location privacy in pervasive computing. *Privacy, Security and Trust within the Context of Pervasive Computing*, Robinson P, Vogt H, Wagealla W (eds.), The Kluwer International Series in Engineering and Computer Science, 2004.
8. Ren K, Lou W, Kim K, Deng R. A novel privacy preserving authentication and access control scheme for pervasive computing environments. *Vehicular Technology, IEEE Transactions on* July 2006; **55**(4):1373–1384.
9. Diep NN, Lee S, Lee YK, Lee H. A privacy preserving access control scheme using anonymous identification for ubiquitous environments. *RTCSA '07: Proceedings of the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, IEEE Computer Society: Washington, DC, USA, 2007; 482–487.
10. Langheinrich M. Privacy by design - principles of privacy-aware ubiquitous systems. *UbiComp '01: Proceedings of the 3rd international conference on Ubiquitous Computing*, Springer-Verlag: London, UK, 2001; 273–291.
11. Chan H, Perrig A. Security and privacy in sensor networks. *Computer* 2003; **36**(10):103–105.
12. Juels A. Rfid security and privacy: a research survey. *Selected Areas in Communications, IEEE Journal on* Feb 2006; **24**(2):381–394.
13. Ren K, Lou W. Privacy-enhanced, attack-resilient access control in pervasive computing environments with optional context authentication capability. *Mobile Networks and Applications* 2007; **12**(1):79–92.
14. Kim J, Kim Z, Kim K. A lightweight privacy preserving authentication and access control scheme for ubiquitous computing environment. *ICISC*, 2007; 37–48.
15. Li CT, Hwang MS, Chu YP. Further improvement on a novel privacy preserving authentication and access control scheme for pervasive computing environments. *Computer Communications* 2008; **31**(18):4255–4258.
16. Chaum D. Blind signatures for untraceable payments. *Advances in Cryptology Proceedings of Crypto 82*, Chaum D, Rivest R, Sherman A (eds.), 1983; 199–203.
17. Lamport L. Password authentication with insecure communication. *Commun. ACM* 1981; **24**(11):770–772.
18. Stubblebine SG, Syverson PF, Goldschlag DM. Unlinkable serial transactions: protocols and applications. *ACM Trans. Inf. Syst. Secur.* 1999; **2**(4):354–389.
19. Wakeman I, Chalmers D, Fry M. Reconciling privacy and security in pervasive computing: the case for pseudonymous group membership. *MPAC '07: Proceedings of the 5th international workshop on Middleware for pervasive and ad-hoc computing*, ACM: New York, NY, USA, 2007; 7–12.
20. Choi JY, Jakobsson M, Wetzel S. Balancing auditability and privacy in vehicular networks. *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, ACM: New York, NY, USA, 2005; 79–87.
21. Zhou J, Gollman D. A fair non-repudiation protocol. *Security and Privacy, IEEE Symposium on* 1996; **0**:0055.
22. Jakobsson BM. Privacy vs. authenticity. PhD Thesis, La Jolla, CA, USA 1998.
23. Bangerter E, Camenisch J, Lysyanskaya A. A cryptographic framework for the controlled release of certified data. *Security Protocols Workshop*, 2004; 20–42.
24. Raya M, Hubaux JP. The security of vehicular ad hoc networks. *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, ACM: New York, NY, USA, 2005; 11–21.
25. Sampigethaya K, Huang L, Matsuura K, Poovendran R, Sezaki K. Caravan: Providing location privacy for vanet. *Escar 2005: 3rd Embedded Security in Cars Workshop*, 2005.
26. Raya M, Hubaux JP. Securing vehicular ad hoc networks. *Journal of Computer Security* 2007; **15**(1):39–68.
27. Rahman S, Hengartner U. Secure crash reporting in vehicular ad hoc networks. *Third International Conference on Security and Privacy in Communication Networks (SecureComm 2007)*, To appear: New York, NY, USA, 2007.
28. Sun J, Zhang C, Fang Y. An id-based framework achieving privacy and non-repudiation in vehicular ad hoc networks. *Military Communications Conference, 2007. MILCOM 2007. IEEE* 29-31 Oct 2007; :1–7.
29. Buttyán L, Holczer T, Vajda I. On the effectiveness of changing pseudonyms to provide location privacy in vanets. *ESAS*, 2007; 129–141.
30. Myles G, Friday A, Davies N. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing* 2003; **2**(1):56–64.
31. Kapadia A, Henderson T, Fielding JJ, Kotz D. Virtual walls: Protecting digital privacy in pervasive environments. *Pervasive*, 2007; 162–179.
32. Liu L. From data privacy to location privacy: models and algorithms. *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, VLDB Endowment, 2007; 1429–1430.
33. Chaum D. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM* 1981; **24**(2):84–88.
34. Camenisch J, Lysyanskaya A. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. *Advances in CryptologyEUROCRYPT 2001* 2001; :93–118.
35. Camenisch J, Lysyanskaya A. Signature schemes and anonymous credentials from bilinear maps. *Advances in Cryptology–CRYPTO 2004*, Springer, 2004; 1–6.
36. Belenkiy M, Chase M, Kohlweiss M, Lysyanskaya A. P-signatures and noninteractive anonymous credentials. *Proceedings of the 5th conference on Theory of cryptography*, Springer-Verlag, 2008; 356–374.
37. Lee C, Deng X, Zhu H. Design and security analysis of anonymous group identification protocols. *Public Key Cryptography*, Springer, 2002; 399–402.
38. Nguyen L, Safavi-Naini R. Dynamic k-times anonymous authentication. *Applied Cryptography and Network Security*, Springer, 2005; 318–333.
39. Magkos E, Kotzanikolaou P. Enhancing Privacy-Preserving Access Control for Pervasive Computing Environments. *MobiSec 2010: Proceedings of the 2nd International ICST Conference on Security and Privacy in Mobile Information and Communication Systems*, LNICST, Springer, to be published, 2010.
40. Chaum D, Fiat A, Naor M. Untraceable electronic cash. *CRYPTO '88: Proceedings on Advances in cryptology*, Springer-Verlag New York, Inc.: New York, NY, USA, 1990; 319–327.
41. Fujioka A, Okamoto T, Ohta K. A practical secret voting scheme for large scale elections. *ASIACRYPT '92: Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, Springer-Verlag: London, UK, 1993; 244–251.
42. SECG. Standards for efficient cryptography group. SEC 1: Elliptic curve cryptography. Available at: http://www.secg.org/download/aid-385/sec1_final.pdf 2005.
43. Parno B, Perrig A. Challenges in securing vehicular networks. *Workshop on Hot Topics in Networks (HotNets-IV)* 2005; .

DOI: 10.1002/sec