Privacy Enhanced Access Control in Pervasive Computing Environments

Kui Ren and Wenjing Lou
Department of Electrical and Computer Engineering
Worcester Polytechnic Institute, Worcester, MA 01609
Email: {kren, wjlou}@ece.wpi.edu

Abstract— **Privacy and security are two important but seemingly contradict objectives in pervasive computing environments (PCEs). On the one hand, service providers want to authenticate service users and make sure they are accessing only authorized services in a legitimate way. On the other hand, users want to maintain necessary privacy without being tracked down for wherever they are and whatever they are doing. In this paper we propose a novel privacy enhanced authentication and access control scheme to secure the interactions between mobile users and services in PCEs. The proposed scheme seamlessly integrates two underlying cryptographic primitives, blind signature and hash chain, into a highly flexible and lightweight authentication and key establishment protocol. It provides explicit mutual authentication between a user and a service, while allowing the user to anonymously interact with the service. Differentiated service access control is also enabled in the proposed scheme by classifying mobile users into different service groups.**

I. Introduction

Pervasive computing environments (PCEs) with their interconnected devices and abundant services promise great integration of digital infrastructure into all aspects of our lives, from our physical selves, to homes, offices, streets and so forth [1], [39]. The huge number of communicating devices will provide seamless access to multiple dynamic networks at any location. Users and their autonomous agents will be able to traverse these networks, coexist with each other and thus create a truly ubiquitous intelligent computing environment.

As networking technologies become commonplace and central to everyday life, companies, organizations and individuals are increasingly depending on electronic means to process information and provide relevant services in order to take advantage of ambient computing intelligence in PCEs [2], [4], [5], [6]. Inevitably, many of these information transactions will be sensitive and critical [19] and thus, it is essential to enforce *access control* to prevent information leakage and service abuse, and to stop malicious attacks. In other words, dynamic access to services should be granted only based on pre-established (direct or indirect) trust between users and service providers. To this end, trust relationship by means of mutual authentication between users and service providers should be established *priori* to the access of services. Traditional authentication which focuses on identity authentication may fail to work in PCEs, partly because it conflicts with the goal of user privacy protection and partly because the assurance achieved by entity authentication will be of diminishing value [19]. For instance, a service provider may only concern whether the accessing user is authorized or not, but has limited interest in who she is in many non-critical scenarios. Meanwhile, services themselves should be authenticated to users. Users will only accept authenticated information from genuine services they intend to interact with to avoid potential deception and other malicious attacks. The importance of authenticating services is discussed in [13].

Another big forthcoming challenge for actually deploying pervasive computing services on a significant scale is how to have adequate provision for handling *user privacy*, which is considered as one of the fundamental security concerns that are explicitly identified by a series of laws [3]. In environments with significant concentration of "invisible" computing devices gathering and collecting the identities, locations and transaction information of users, users should rightly be concerned with their privacy. At the same time, the physical outreach of pervasive computing makes preserving user's privacy a much more difficult task [10], [15], [40].

Some of the user privacy issues that should be treated in PCEs has been pointed out in [8], including location privacy, connection anonymity and confidentiality. We further clarify the scope of privacy in PCEs as follows. **Anonymity:** The real identity of a user should never be revealed from the communications exchanged between the user and a server unless it is intentionally disclosed by the user. Different communication sessions between the same user and service should not be linkable. **Context Privacy:** Neither the service nor other users of the service should be able to learn the exact context information (*e.g.*, location, duration, type of service request, *etc.*) of a user, unless the user decides to disclose such information. Users' context information should be protected against both outsiders and service providers they interact with. **Confidentiality** and **Integrity:** The interactions between a user and a service should have both confidentiality and integrity protections whenever such protections are required.

In reality, the quests for authentication/access control and user privacy protection conflict with each other in many aspects, and the problem is more complex in PCEs. On one hand, the service generally depends on the user identity information and corresponding pre-established trust relationship as well as the service contract between them to accomplish user authentication and conduct access control. On the other hand, the user does not want to be tracked by the service for wherever she is and whatever she does. The trade-off between

0-7803-9277-9/05/$20.00/©2005 IEEE

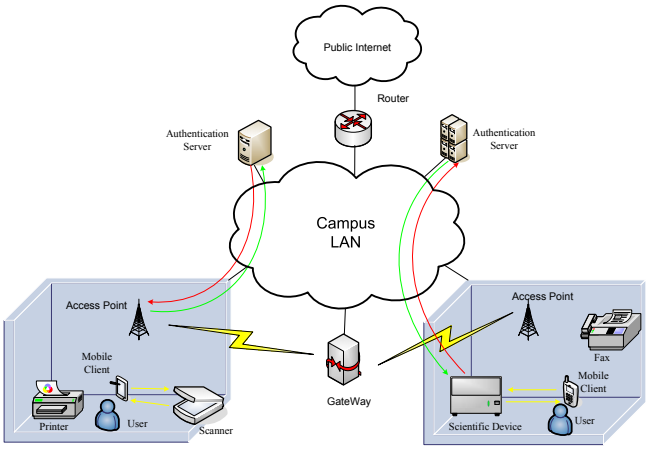

Fig. 1. A sample pervasive computing environment

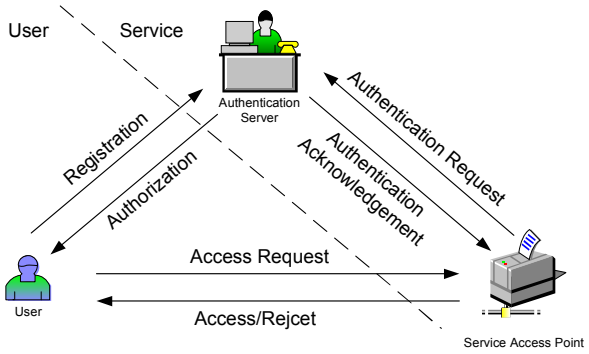

Fig. 2. System architecture

the two thus poses a great challenge to security designers.

In this paper, we propose a user privacy preserving access control scheme at the application level to address the security and user privacy concerns in PCEs. The proposed scheme is implemented at the application level without relying on any underlying system infrastructure such as the *Lighthouse* or *mist routers* etc. required by many other approaches [2], [7], [8], [15]. The proposed scheme provides explicit mutual authentication between the two parties, while at the same time allowing the mobile user to interact with desired service anonymously without revealing her identity. The scheme seamlessly integrates two underlying cryptographic primitives, *blind signature* and *hash chain*, into a highly flexible and lightweight authentication and key establishment protocol. The scheme possesses many desirable security properties, such as anonymity, non-likability, non-repudiation, accountability, differentiated services access control, etc., with very low protocol complexity (refer to Section IV).

The rest of this paper is organized as follows. In Section II we describe the system architecture of PCEs and introduce the cryptographic primitives used in our scheme. We present in detail the proposed scheme in Section III. Then we discuss the security features and the performance of the proposed scheme in Section IV. Finally, we review the related work in Section V and conclude the paper in Section VI.

II. System Architecture and Cryptographic Primitives

A sample system architecture of a campus PCE is given in Fig. 1. Generally, a PCE consists of three type of entities: mobile users, services and back end authentication servers, in addition to the underlying wired and wireless communication infrastructures. Note that wireless network access is itself a service. User privacy should be protected not only from outsiders but also from network service providers. Our proposed access control scheme is designed to secure the interactions among these three types of entities as shown in Fig. 2. More specific, our scheme aims to provide anonymous mutual authentication between the mobile user and the service, and also possible confidentiality and integrity protection for the communications between the mobile user and the service (e.g., wireless service access point for wireless network access service).

Our scheme is based on two cryptographic techniques, blind signature and hash chain. A brief review of the two techniques is provided as follows.

Blind Signature

Blind signature scheme [16] is a variation of digital signature scheme in which the content of a message is disguised from its signer. Blind signature schemes can be implemented based on a number of well known digital signature schemes, such as RSA [34]. To produce a signature on a message, a user first *blinds* the message with a *blinding function* f, typically by combining it with a random *blinding factor* k, and then forwards the blinded message to the signer A. The signer signs the blinded message using a standard signing algorithm, say $S_A(x)$ which denotes the signature of A on m, and sends the result back to the user, who then *unblinds* it with an *unblinding function* g to obtain the signer's signature on the original message. The algorithm is designed such that $g(S_A(f(m))) = S_A(m)$.

Blind signatures are used to provide non-linkability, which prevents the signer from linking blinded message it signed to the unblinded version that it may be called upon to verify. In this case, the signed, blinded value is unblinded prior to verification in such a way that the signature remains valid for the unblinded message. This can be useful in schemes where anonymity is required. Blind signature schemes find a great deal of use in applications where sender privacy is important. This includes various *digital cash* schemes and voting protocols [17], [18].

Hash Chain

One-way hash function f is a powerful and yet computationally efficient cryptographic tool. By applying $f()$ repeatedly on an initial value m, one can obtain a chain of outputs $f^j(m)$. These outputs can be used in the reverse order of generation for the purpose of authentication: $f^{j-1}(m)$ can be proven to be authentic if $f^j(m)$ has been proven to be authentic due to the one-wayness property of hash function.

Hash chains together with signatures are widely used in micro-payment schemes such as Payword, *i*KP and Netcard [32]. In such schemes, the effect of a digital signature is reused many times over subsequent messages (containing pre-images of a specific hash). The concept of hash chain was first proposed for use in authentication scheme by Lamport [36]. Recently, Weimerskirch *et al.* adopted hash chain technique for efficient user recognition based on weak authentication [38].

III. The Proposed Scheme

This section presents our privacy preserving access control scheme. Consider the scenario that a mobile user wants to be able to dynamically access the wireless or other available services in PCEs. Due to the insecurity of the wireless communication channel, the authorization of the mobile user to the particular service she requests should be verified and the subsequent data traffic should be protected. Moreover, the mobile user should have the full control of her context privacy. That is, the mobile user's context information like location, time, and transaction profiles, *etc.*, should be well protected, and can only be exposed by the mobile user herself; nobody else, including the service she is interacting with, can get the clue regarding the user's context. Therefore, the design considerations of the proposed scheme include: 1) provide explicitly mutual authentication between the mobile user and the service; 2) allow the mobile users to anonymously interact with the service; 3) enable differentiated service access control among different users; 4) provide flexibility and scalability to both user and service sides; 5) generate fresh session keys to secure the interaction if applicable; 6) have high efficiency with respect to both communication & computation costs and management overheads; 7) provide easy accountability.

Conceptually, the proposed scheme works as follows. The mobile user first generates some specific credentials, and then she get these credentials authorized from the services through the *user authorization protocol*. The mobile user then uses the authorized credentials to access the desired services, and performs mutual authentication with the service before actually using it. Note that at this stage, the mobile user identifies herself only by presenting the authorized credentials without disclosing any of her context information. Upon the successful completion of mutual authentication process, both parties will share a fresh session key which will be used to secure the subsequent data traffic of the session. This is done through the *user operational protocol*. A *dispute resolution protocol* is also designed to solve possible disputes that might rise between the mobile users and service providers. Table I lists the notations used throughout the description of the protocols for ease of reference. Note that we assume users are capable of manipulating the source addresses of the outgoing Medium Access Control (MAC) frames. This assumption is prerequisite for anonymous communication otherwise one can easily identify a user based on her unique MAC address.

A. The User Authorization Protocol

The purpose of the user authentication protocol is to establish the a priori security credentials between mobile users and service providers, which can be used as the security anchor in the subsequent mutual authentication processes whenever a mobile user attempts to access a service. In our user authorization protocol, the mobile user and the corresponding service provider (*i.e.*, the authentication server) need to authenticate each other first. This is typically done through some out-of-band non-cryptographic technique. The mobile user needs to register herself as a legal user of some service types the service provider provides. She obtains the public keys of the services of which she is entitled to use. She also needs to obtain a certificate $CertU$ which binds her identity U to her public key $PubK_U$, signed by the private key of the service provider $PriK_S$. Then, the mobile user executes the user authorization protocol to submit her credential and to obtain the signed credential from the authentication server.

Our user authorization protocol is based on blind signature [16], which hides the association between the authorized credential and the mobile user's real identity therefore user's context information can be concealed during the access. Further, through a deliberately designed combination with hash chain technique, a series authorized credentials can be obtained by the mobile user through one protocol run, which increases the protocol efficiency and the multiple credentials can be used to avoid the problems of using a single secret for too long.

The proposed user authorization protocol contains two steps: 1) *credential generation*, and 2) *credential authorization*. The mobile user generates her own specific credentials as shown in Table II:

Credential Generation:
1. generate two fresh nonces: r'_U and r''_U.
2. sign her own ID with a fresh nonce r''_U:$\{U, r''_U\}_{PriK_U}$.
2. compute the anchor value C^0 of the credential chain as $C^0 = h(r''_U, U, \{U, r''_U\}_{PriK_U})$,
4. compute the credential chain $C^j = h^j(C^0), j \leq n$, with length n.
5. blind C^n as $C_U = \{r'_U\}_{PubK_{SID}} \times C^n$.

TABLE II
Credential Generation

The mobile user first generates two fresh nonces. Then she signs her own identity together with one fresh nonce using her private key. Next, she computes the anchor value C^0 of the credential chain with the signature. Clearly, the signature contained in C^0 provides non-repudiation property. This is true because only the mobile user herself can generate it and the fresh nonce guarantees its freshness. Then, a credential chain is computed via hash operation. The length n can be adjusted to the proper value depending on the actual frequency of usage and storage capability. In the last step, the mobile user blinds credential chain tail C^n by using blind signature technique. Next, the mobile user sends out the blinded C^n for authorization as shown in Table III. The authentication server signs C^n with the private key of the requested service type and returns the signed credential back to the mobile user.

U	A mobile user that is usually identified by her public key and can belong to some user group(s).
S	Service provider or its authentication server which is used to authenticate the user for the purpose of access control.
M	User group manager that can act as an agent for group members.
TTP	Trusted third party, an entity which is trusted by both the mobile user and the service provider
SID	A service type identifier, which describes a selected subset of the available service pool that can be accessed by a particular mobile user, is identified by a unique public key. Different users may belong to the same service type.
P	Service access point. For wireless networking service, it represents the access point (AP).
$PubK_A, PriK_A$	Public and private key pair of entity A.
K_{AB}	Shared secret key between entities A and B.
m, X_m	Message m and its corresponding ciphertext.
(m_0, m_1)	Concatenation of two messages.
$\{m\}_{PubK_A}$	Encrypt message m with the public key of entity A.
$\{m\}_{PriK_A}$	Sign message m with the private key of entity A. If not otherwise stated, message m is recoverable.
$\{m\}_{K_{AB}}$	Encrypt message m by symmetric key algorithm with the secret key shared between entities A and B.
$h()$	A cryptographic secure one-way hash function, or one-way hash function in short, such as *MD5* [35].
$h_{K_{AB}}(m)$	A cryptographic secure MAC algorithm, computing the message digest of message m with key K_{AB}.
$h^j(m)$	Hash message m j times: $h^j(m) = h(h^{j-1}(m)), j = 1, 2, 3, ...$
r_A	A nonce generated by entity A, usually it is 64-bit pseudo random number.
$C^j, j = 0, 1, 2, \ldots$	A series of authorized credentials used by an entity to obtain service access permission.
$CertA$	A certificate which binds entity A with her public key $PubK_A$.

TABLE I

NOTATION

Credential Authorization:

U $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ S

$U, C_U, CertU, SID$ $\longrightarrow$

authorization request

6. Verifies $CertU$ with $PubK_S$.

7. Signs on C_U: $C_S = \{C_U\}_{PriK_{SID}} = r'_U \times \{C^n\}_{PriK_{SID}}$.

C_S $\longleftarrow$

authorization comfirmation

8. Decrypts and verifies U, S.

9. Computes C_S/r'_U and obtains a valid signature pair $(C^n, \{C^n\}_{PriK_{SID}})$.

TABLE III

CREDENTIAL AUTHORIZATION

- Besides the general public key $PubK_S$ for user authentication purpose, the service provider maintains a pool of public keys corresponding to difference service types. We assume that the mapping between the service type identifier SID and its corresponding public key is clear to the mobile user. The authorized credentials of different service types are actually signed by the different public keys. This allows for differentiated access control in the subsequent stage. If the scope and the meaning of the service type is carefully defined and the services are therefore well classified, the combinational usage of several authorized credentials at the same time can further improve the ability to enable higher level differentiated service access control. This will also improve the flexibility and scalability of the proposed scheme.
- Once the signed credential is returned to the mobile user, the computation of C_S/r'_U indeed results in a valid signature on C^n due to the property of blinded signature. Therefore, after the protocol execution, the mobile user holds a verifiable authenticator - credential C^n and its signature. Although the authentication server doesn't know what the value of C^n is at the time it signs it, the authenticity of C^n can be verified by the signature. Therefore, once the authenticator is submitted to the authentication server, the authentication server will be able to verify and grant the service request. However, it still has no information about who the user is, except for her requested service type.
- Although the authentication server signs only the n-th credential C^n, the remaining hash chain values through C^0 to C^{n-1} are authorized implicitly at the same time, due to the one-wayness nature of the hash function. Note that the values of the credential hash chain should never be revealed to any third party.
- The mobile user can also generate serval different credential hash chains at the same time, and get each C^n signed by the authentication server simultaneously in one protocol run. Hence, the protocol efficiency can be further improved, as well as the flexibility.

The user authorization protocol allows the mobile user to obtain the authorized credentials from service provider. Note that the user authorization protocol runs only when the mobile user's authorized credentials are exhausted or for the first time registration. The user authorization protocol is highly flexible. It can be accomplished via both online and off-line approaches. It can also be accomplished through the agent of the mobile users. We can easily imagine that the network manager or administration staff can acquire the authorized credentials from the service providers on behalf of the users in a company and then distribute them to the user. This delegable feature greatly improves the usage flexibility of the mobile users and allows dynamic authorization. It also significantly simplifies the management overheads at the service side. The authentication server is now able to manage only one certificate for each user group, instead of those of all group members.

B. The User Operational Protocol

The user operational protocol allows a mobile user to safely enjoy different kinds of services she is authorized to in PCEs at

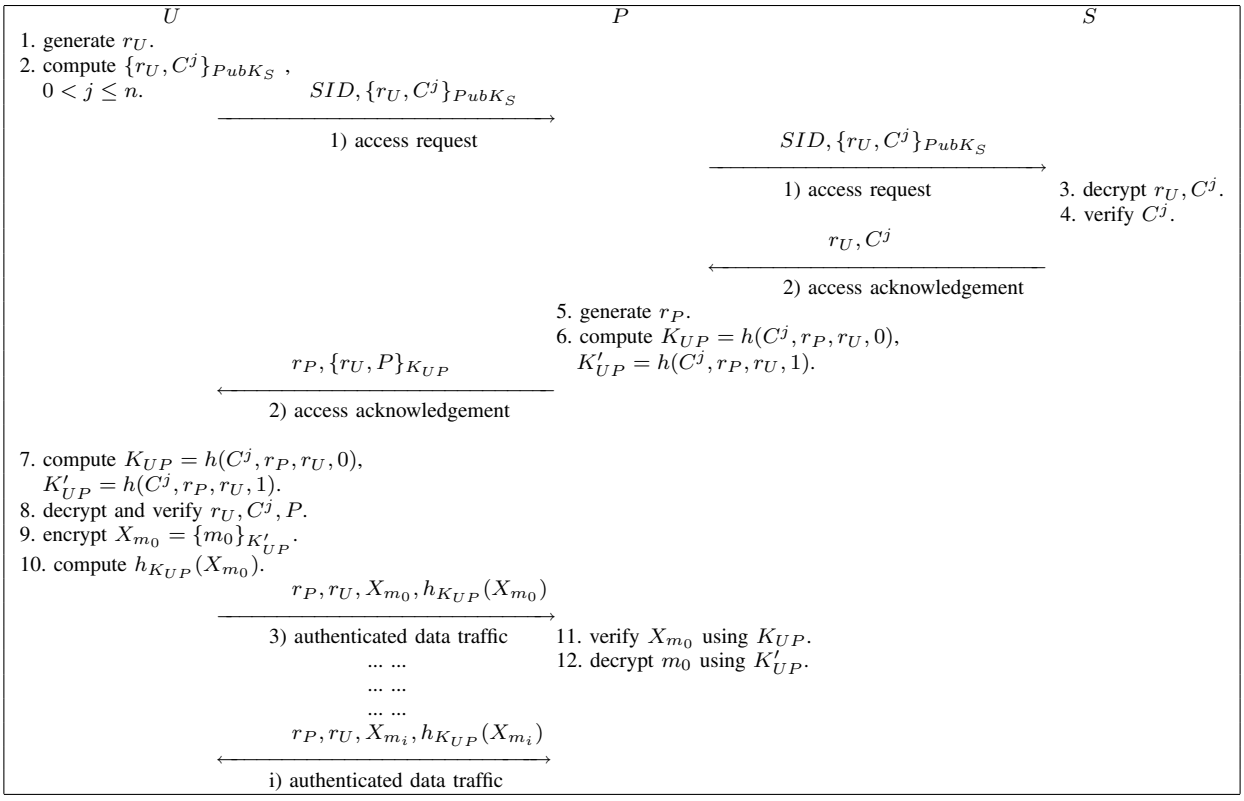

TABLE IV

THE USER OPERATIONAL PROTOCOL

anytime, from anywhere without disclosing any of her context information unless she is willing to do so and it is absolutely necessary (*e.g.*, in case of disputes). Conceptually, the user operational protocol works as follows.

The mobile user first sends an access request, which contains a service access capability claim and an authenticator used to prove her legitimacy to the service requested, including the authorized credential and a fresh nonce, to the service access point such as the wireless network access point or a network printer. The service access point simply forwards this access request message to its back end authentication server for authentication. Upon receiving the forwarded access request, the authentication server decrypts and verifies the authenticity of the contained credential. The authentication server also checks whether the current service access point conforms to the user's service type. If both results are positive, the authentication server ascertains that the mobile user is indeed authorized to access this particular service although it has no idea who the mobile user is, except for the service type the user belongs to. Hence, it replies the service access point with an access grant, which otherwise would be an access deny message. The access grant message contains the decrypted authenticator information of the mobile user it just verified. We assume that there is a secure tunnel (*e.g.*, IPsec ESP mode [26]) between the service access point and its back end authentication server so that the former can securely get this piece of secret information sent by the latter. Note that this process is transparent to the mobile user. The service access point then computes two fresh session keys with the obtained secret information and a fresh nonce of its own. Next the service access point encrypts the obtained secret information and its own identity together with one of the new session keys and finally, replies the mobile user with an access grant message, containing the fresh nonce and the previous encrypted information. Upon receiving this message, the mobile user could compute the same two session keys through the same manner. After decrypting the access grant message with the computed session key, the mobile user now authenticates the service side by checking the validity of the decrypted value with her own. If the result is positive, the mobile user ascertains that the current service is legal. This concludes the mutual authentication, and now both two parties share a fresh session key, which can be used to secure the subsequent data traffic in this session. The user operational protocol is outlined in Table IV, describing a successful protocol run.

- In the access request message, the mobile user encrypts a fresh nonce r_U and authorized credential C^j with service's public key which is used for authentication purpose. The encryption operation has dual purposes: 1) keep the secrecy of r_U and C^j from eavesdropping; 2) service authentication, because only the user's intended legitimate service can decrypt the message correctly. The SID is provided to claim user's capability to access the targeted service.

- When the authorized credential chain is used for the first time, *i.e.*, $C^j = C^n$, the mobile user should send both C^n and its signature. In this case, the access request message would be: $\{r_U, C^n, \{C^n\}_{PriK_{SID}}\}_{PubK_S}$. Each credential is used exactly once, that is, used in only one session and is obsoleted afterwards. Hence, an authorized credential chain of length n can be used to access the services for n sessions before all credentials are exhausted. The use of different credential for each session is necessary to defend against the replay attack and possible double spending problem (e.g., for accountability).
- The authentication of the submitted credential at the service side is as follows: If a credential is submitted together with its signature, that is, $(C^n, \{C^n\}_{PriK_S ID})$, the authentication server verifies the signature using corresponding public key by referring to SID in the same message. A negative result will trigger an access deny message, sent to the service access point. A positive result confirms the validity of the submitted credential. Of course, a duplication check on C^n should be first executed before signature verification to prevent a potential double spending of C^n. Upon success of the verification, the authentication server saves C^n according to its service type. Recall that in the last subsection, we pointed out that each different public key is used to bind a particular service type. Thus, although the authentication server couldn't know who the user is, it does know this user's capability to access the services, that is, whether she is eligible for the requested service or not through the submitted credential. Hence, a differentiated service access control is easily realized.

  If the submitted credential is a single value C^j, its authentication procedure contains two steps: 1) computes $h(C^j)$; 2) finds the matching value of $h(C^j)$ from currently stored credentials of the same service type. If the second step succeeds, the authentication succeeds accordingly. This is due to the one-wayness property of the underlying hash function. The authentication server then updates the currently stored C^{j+1} with C^j. The remaining operation is the same as the previous situation. Note that for each different credential chain, the authentication server stores exactly two values: the signed C^n and the newest (current) C^j. This is for dual purposes: 1) for the ease of credential authentication on $C^j, j < n$; 2) prevent potential double spending of the credentials for all C^js.
- The traffic between the service access point and its back end server is assumed to be protected by private or previously established secure tunnel, which is beyond the design range of this protocol.
- The service access point has no responsibility for user authentication. It simply defers this job to its back end server. The computation and management overheads at the service access point are extremely low and requires little storage capability: i)no public key operations; ii)no long term key and certificate management; iii)session keys are discarded once the session is terminated; iv)hash and symmetric key operations only. Hence, it is very simple and efficient, which could greatly decreases its cost and helps the wide deployments.
- The service access point and the mobile user compute the fresh session keys independently, and authentication server has no control over the computed session keys. The fresh nonce used in key generation guarantees the freshness of the session keys. Two fresh session keys are generated. One is for encryption and the other is for integrity protection, i.e., generating the message authentication code (MAC) [35].
- The fresh nonce r_P, r_U are then used by the mobile user and the service access point to identify the session between them, that is, binding the two communication parties and the exchanged traffic together. We can see that there is no way to identify the session between the two otherwise, because both two parties may interact with many other parties at the same time, especially for service access point.
- The one-time usage feature of the authorized credentials and its linkage with the service type provide effective accountability. Similar to the micro-payment schemes, the accounting mechanism can be easily incorporated into the system through a nearly same manner. We point out that double spending of the authorized credentials actually does not affect the system security as proved in security proof part. Hence, the choice between one-time or multiple usage of the authorized credentials can be a simply policy decision. It also can be dynamically switched between either the way easily according to the real situations.

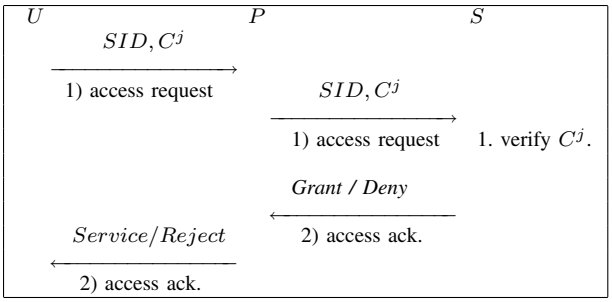

TABLE V
SIMPLIFIED USER OPERATIONAL PROTOCOL

Sometimes a mobile user's targeted service in PCEs has nothing to do with data traffic. (For instance, a vending machine in a company offers free drinks to the company members only. Once the vending machine is assured that the user is indeed a company member, it will offer him a bottle of alcohol after he presses the button. Of course, this guy doesn't want to let the others know that this is the eighth bottle he has drunk since this morning.) In this case, we have a much simplified and extremely efficient protocol as indicated in Table V. Only hash and match operations are needed. Of course, if the credential chain is used for the first time, an

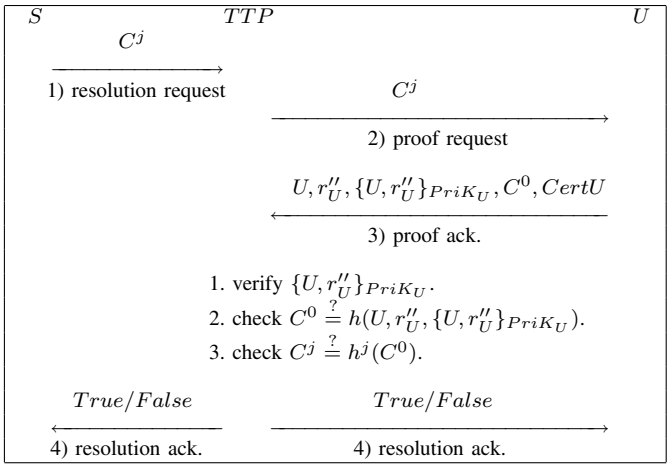

TABLE VI
DISPUTE RESOLUTION PROTOCOL

additional signature verification is required. Also note that in such contexts, the authentication of the service may not be relevant, and thus is skipped in the protocol.

C. The Dispute Resolution Protocol

Since our protocol provides anonymous authentication of the mobile user, one concern from the service provider is that the service might be abused by the users beyond their control. For example, one might try to access the service using some stolen C^j. In the following, we present a dispute resolution protocol, which is used to prove an user's legitimacy while the service provider suspects that the user is potentially illegal by using some stolen credentials (e.g., the service provider has observed some abnormal access pattern of the user). The dispute resolution protocol allows the service provider to challenge the mobile user and the mobile user in response to prove her legitimacy again without exposing her context privacy to the service provider.

An off-line Trusted Third Party (TTP) is introduced for this purpose. The idea is that when a dispute happens, the service provider submits the suspected credential to TTP, and the mobile user is requested to prove to TTP her ownership of the submitted credential. Technically, the proof of the ownership is done through showing the knowledge about the whole credential chain corresponding to the questioned credential. From our user authorization protocol, we know that the anchor value C^0 of the credential chain is computed over the mobile user's real identity and a corresponding signature signed over its identity and a fresh nonce. Therefore, only the mobile user herself can generate a valid pre-image value of C^0. Obviously, it is computationally impossible to forge such a message as long as the underlying cryptosystems is secure. So, if the mobile user proves that she could compute C^0 from the above identity information, it is indeed a proof of her ownership about the credential. TTP is therefore informs the service provider that the mobile user is indeed the authorized one, which otherwise would be a failure information. This accomplishes our protocol goal. We outline our dispute resolution protocol in Table VI.

Note that in our dispute resolution protocol, the mobile user's privacy is well protected. The service provider only gets to know whether the questioned mobile user is an authorized one or not. Still the service provider has no clue about who the user is. At the same time, although TTP knows who the mobile user is, it has no information about user's service usage profile. We exclude the colluding possibility of TTP and the service provider, which is beyond the scope of this paper. The ability to resolve the disputes between the mobile user and the service provider also helps prevent potential illegal credential transfer among the users in some sense, as we will discuss further in Section IV.

IV. ANALYSIS OF THE PROPOSED SCHEME

A. Security Related Properties of the Proposed Scheme

The proposed scheme exhibit many nice security related properties as discussed below:

- **Mutual Authentication:** In the proposed scheme, the mobile user is authenticated based on her authorized credential, in the sense that the service knows the user is indeed legal and authorized. The service authenticates itself to the user through its public key certificate and by showing its knowledge of the corresponding private key. The mutual authentication is highly necessary in the PCEs as discussed before to prevent potential malicious attacks from the both sides.
- **User Context Privacy:** The users' context privacy is well protected by the proposed scheme, only absolutely necessary information is known to the service, i.e., users' service type, in order to grant appropriate access. Through the blind signature technique, the mobile users could be authenticated anonymously without disclosing any other information. All the service side knows is that some legal users are accessing some particular services. The information is also protected against the outsiders. No third party has the ability to acquire the user's context information, as all the interaction traffic are well protected.
- **Dispute Resolution (Non-repudiation):** By carefully integrating the mobile user's fresh identity signature into the pre-image of the anchor value of the authorized credential chain, dispute resolution function is enabled in our proposed scheme. Because only the given mobile user can generate its fresh identity signature, and therefore, the authorized credential chain, explicit non-repudiation is also provided. This is a good resort to resolve the potential disputes arising between the two parties.
- **Non-Linkability:** Because the authorized credential is never transmitted in plaintext, and is always combined with fresh nonce in the message, there is no relationship between different sessions from the same user. In other words, sessions from the same user can not be identified by the attackers. Hence, users' transaction profiles are well protected, which further enhances the user privacy.
- **Accountability & Non-Transferability Equivalency:** In the proposed scheme, the credentials are authorized only

	This paper	[21]	[24]
Concrete Protocol	Yes	Yes	No
Mutual Authentication	Yes	Yes	No
User Context Privacy	Yes	Yes	Not to the services
Dispute Resolution	Yes	No	Yes
Non-Linkability	Yes	No	Not to the services
Non-Transferability Equivalency	Yes	No	N/A
Data Confidentiality	Yes	Easy to obtain	No
Message Integrity	Yes	Yes	No
Accounting Capability	Yes	No	Yes
Differentiated Service Access Control	Yes	No	Yes
Provable Security	Yes	No	N/A

TABLE VII
PROTOCOL SECURITY FEATURES COMPARISON

when the mobile user is explicitly authenticated. The one-time usage property of the authorized credentials prevents double spending problem and further provides good accounting capability which allows the accounting function be easily incorporated[1]. Furthermore, from the service point of view, the proposed scheme provides equivalent Non-Transferability, which means that even the credentials are delegated among users, no harm is done to the service provider in the sense that the authorized user is responsible for all the service received by her authorized credentials. This novel feature greatly reduces the service abuse problem worried by the service providers. Using blind signature [21] alone can not provide this property because there's no way to prevent the double spending problem and hence, no way to prevent service abuse problem.

- **Data Traffic Protection:** The user operation protocol generates fresh session keys to protect the interaction data traffic between the mobile user and the service. Data **confidentiality** and **integrity** can be provided based on the symmetric cryptography.
- **Differentiated Service Access Control:** By classifying the mobile users into different service types, differentiated service access control is enabled in our scheme. Different mobile users are authorized accordingly based on their belonged service types. **User authorization** is therefore, accomplished in a differentiated way. Moreover, the combinational usage of the different credentials may help to provide high level differentiated service access control, which is beyond the scope of this paper.

We compare our scheme to other similar approaches that are intended to provide anonymous interactions between the users and the services in Table VII. The advantage of our scheme is obvious.

[1]For example, we can limit the amount of service one credential is entitled thus making the amount of service measurable.

B. Performance of the Proposed Scheme

Despite the number of desirable security properties provided, the proposed scheme is extremely lightweight. We analyze the overheads introduced in this subsection.

- **Management Overhead:** The proposed scheme involves minimum management overheads (e.g., human interaction). The service provider needs to manage one certificate per user and the corresponding user profile. Due to the delegation property, this number can be significantly reduced to that of the user groups (i.e., one user per group). On the other side, each mobile user needs to manage the certificates of the service provider and the different service types she belongs to.
- **Storage Overhead:** While the protocol is running, the back end authentication server stores two values (C^j, C^n) for each currently in-use credential chain and one value (C^n) for each of the used but unexpired chain. The service access point maintains no permanent user information or key information. Each service access point only stores two session keys per session, besides two nonces to identify that session. The mobile user stores the two random nonces (r'_U, r''_U), and the credential chains authorized to her (e.g., $C^0, \ldots, C^n$ and signature of C^n). In addition, the mobile user should store two nonces and two session keys for each ongoing session. The method to store a hash chain can have a computation and storage trade-off. The mobile user can also choose to store the anchor value and the current value of the hash chain only and compute the needed value on-the-fly.
- **Communication Overhead:** The user operational protocol requires two-way and four messages to accomplish mutual authentication and session key establishment between the user and the service. Note that two-way is the minimum number for any authenticated key establishment protocol to fulfill its goal. Therefore, the proposed scheme is highly efficient in the sense of communication overhead.
- **Computation Overhead:** The mobile user performs one public key operation per session and all the remaining are hash and symmetric cryptographic operations. The public key operation can be done off-line. The authentication server also needs to do one public key operation per session, and one additional signature verification for each authorized credential chain (which could be used for n sessions). The service access point is completely exempted from performing public key operations. We compare in Table VIII the computation overhead of the proposed scheme with the scheme proposed in [21]. It is observed that the proposed user operational protocol is extremely lightweight.

Notice that our protocol is much more efficient than [21], despite of so many additional security features as discussed above. In [21], the authentication server needs to perform one signature verification every session in addition to one public key decryption. The server therefore, could be the bottleneck

		Public Key Oper.	Sig. Veri.	Nonce Gen.	Hash Oper.	Sym. Key Oper.
Ours	U	1(off-line)	0	1	2	3
	P	0	0	1	2	3
	S	1(online)	1/n	0	0	0
[21]	U	1(off-line)	0	0	1	1
	P	0	1(online)	0	1	1
	S	1(online)	1(online)	0	1	1

TABLE VIII
PROTOCOL COMPUTATION OVERHEADS COMPARISON

of the whole system, due to the potential large amount of concurrent transactions. Moreover, the service access point is required to perform one public key operation for each session, which is also a heavy burden to it. For instance, a wireless access point will have a great trouble to perform public key operations for every user in every session due to its constrained computation capability.

V. RELATED WORK

Recently, quite a few papers have been published to address the new security and privacy challenges in PCEs [7], [8], [9], [11], [15], [19], [21], [24], [25], [27], [31], [40], [41]. However, most of these results fall in the scope of establishing general security framework and identifying general security requirements, without providing concrete security protocols.

Some of these efforts focused on designing specific security infrastructures to protect user context privacy like location information from service providers. The MIST system [7], [8] provides user anonymity through an overlay network assuming the existence of a *Lighthouse*, which keeps all information of all the users. In addition, performance degradation is unavoidable for systems that utilize MIX-network style approach [14]. A proxy-based scheme can be found in [11]. Another recent infrastructure based approach, LEXP, can be found in [31]. Some efforts try to maximize user privacy by restricting the access to users' context information. Hengartner *et al.* suggested an architecture to filter out user context information [22].

The remaining efforts mostly focused on identity manipulation approaches, with most of which originated from Chaum's anonymous ID based scheme in 1985 [18]. This general scheme allows users to interact with different services anonymously, using pseudonyms. Pseudonyms can not be linked, but are formed in such a way that a user can prove to one service about his relationship with another using a "statement". Such a statement is called credential. An in-depth description and analysis of different pseudonym systems can be found in [29].

Jendricke *et al.* [24] introduced an identity management system in PCEs where a user is issued multiple identities, and the user uses them depending on applications. The paper only presented a general framework of using virtual identities to protect user privacy while performing access control and authentication, but didn't give any concrete protocol. More recently, He *et al.* [21] presented a simple anonymous ID scheme for PCEs, which is a direct application of Chaum's blind signature technique [16]. However, the scheme suffers from several drawbacks as discussed in Section IV.

Henrici and Muller [23] utilized hash functions to recompute identifiers of a RFID device every time it sends a request to service providers. Their intention were to protect the location privacy of RFID devices. Another approach proposed by Weimerskirch *et al.* uses hash function to realize efficient weak authentication [38]. In order to avoid leakage of user's MAC address or IP address at the lower levels, Gruteser *et al.* [20] came up with a method to hide user's MAC address with anonymous IDs so that the user can not be tracked in a wireless LAN environment.

VI. CONCLUSION

In pervasive computing, mobile users interact seamlessly with abundant services anytime, any where. However, user privacy is at great risk in the pervasive computing environment (PCE) because of its inherent pervasive and heterogenous nature. Legitimate service providers may also suffer from abuse from unauthorized or malicious users. The conflict between user privacy protection and user authentication make security design in PCEs a very challenging task.

In this paper, we proposed a privacy preserving authentication and access control scheme to secure interactions between mobile users and services in PCEs. On the one side, the proposed scheme provides explicit mutual authentication between a mobile user and a service; on the other side, it allows the mobile user to anonymously interact with the service. Hence, the proposed scheme successfully satisfies concerns of both parties - security and privacy. The scheme integrates two cryptographic primitives, blind signature and hash chain, into a highly flexible and lightweight authentication and session key establishment protocol. Differentiated service access control is also enabled in the proposed scheme by classifying mobile users into different service types.

In the future, we would like to extend our scheme to context-aware services in PCEs. In this scenario, certain aspects of user context information should be authenticated and presented to the services. How to disclose necessary authenticated user context information to the context-aware services without affecting user privacy is our future work.

REFERENCES

[1] "Easy Living", Microsoft Research, http://research.microsoft.com/easyliving/.
[2] "GAIA - Active Spaces for Ubiquitous Computing", University of Illinois at Urbana-Champaign, http://choices.cs.uiuc.edu/gaia/
[3] Location Privacy Proteaction Act and other privacy related law, http://www.techlawjournal.com/cong107/Privacy.
[4] "MIT Project Oxygen", http://oxygen.lcs.mit.edu/.
[5] National Institute of Standards and Technology (NIST),"Pervasive Computing SmartSpace Laboratory", http://www.nist.gov/smartspace/
[6] "The Aware Home", Georgia Institute of Technology, http://www.cc.gatech.edu/fce/ahri/.
[7] J. Al-Muhtadi, R. Campbell, A. Kapadia, D. Mickunas, and S. Yi, "Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments," presented at International Conference of Distributed Computing Systems (ICDCS 2002), Vienna, Austria, 2002.
[8] J. Al-Muhtadi, R. Campbell, A. Kapadia, D. Mickunas, and S. Yi, "Routing through the Mist: Design and Implementation," UIUCDCS-R-2002-2267, March 2002.

[9] J. Al-Muhtadi, A. Ranganathan, R. Campbell and M. Mickunas, "A Flexible, Privacy-Preserving Authentication Framework for Ubiquitous Computing Environments", ICDCS Workshops 2002: 771-776, 2002.
[10] J. Al-Muhtadi, A. Ranganathan, R. Campbell and M. Mickunas, "Cerberus: A Context-Aware Security Scheme for Smart Spaces", PerCom 2003, 489.
[11] M. Burnside, *et al.*, "Proxy-based Security protocols in Networked Mobile Devices", ACM SAC 2002, Madrid, Spain, 2002.
[12] M. Burrows, M. Abadi and R. Needham, "A logic of authentication", Proceedings of the Royal Society of London A, 426:233-271, 1989.
[13] L. Bussard and Y. Roudier, "Authentication in Ubiquitous Computing", Workshop on Security in Ubiquitous Computing, UBICOMP'2002, Goteborg, Sweden, 2002.
[14] J. Camenisch and A. Lysyanskaya, "Efficient Non-transferable Anonymous Multi-show Credential System with Optional Anonymity Revocation", In Advances in Cryptology, EUROCRYPT 2001, LNCS 2045, pp. 93-118.
[15] R. Campbell, J. Al-Muhtadi, P. Naldurg, G. Sampemane and M. Mickunas: "Towards Security and Privacy for Pervasive Computing", ISSS 2002: 1-15.
[16] D. Chaum, "Blind Signatures for Untraceable Payments", Advances in Cryptology Proceedings of Crypto 82, D. Chaum, R.L. Rivest, & A.T. Sherman (Eds.), Plenum, pp. 199-203, 1982.
[17] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms", Communications of the ACM, 24(2):84-88,1981.
[18] D. Chaum, "Security without identification: transaction systems to make Big Brother obsolete", Communications of the ACM, 28(10), 1985.
[19] S. Creese, *et al.*, "Authentication for Pervasive Computing", In Security in Pervasive Computing 2003, LNCS 2803, pp. 116-129, 2004.
[20] M. Gruteser and D. Grunwald, "Enhancing Location Privacy in Wireless LAN through Disposable Interface Identifiers: A Quantitiative Analysis, WMASH'03, San Diego, USA, 2003
[21] Q. He, *et al.*, "The Quest for personal control over mobile location privacy", IEEE Communications Magzine, pp.130-136, May 2004.
[22] U. Hengartner and P. Steenkiste, "Access Control to Information in Pervasive Computing Environments", In Proc. of 9th Workshop on Hot Topics in Operating Systems (HotOS IX), Lihue, HI, May 2003.
[23] D. Henrici and P. Muller "Tackling Security and Privacy Issues in Radio Frequency Identification Devices", PERVASIVE 2004, Springer-Verlag, LNCS 3001, pp. 219-224, 2004.
[24] U. Jendricke, M. Kreutzer, and A. Zugenmaier, "Pervasive Privacy with Identity Management", The 1 st Workshop on Security, UbiComp 2002, 2002.
[25] U. Jendricke, M. Kreutzer and A. Zugenmaier, "Mobile Identity Management", The 1st Security Workshop, UBICOMP 2002, Sep. 2002.
[26] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol", IETF RFC 2401, 1998.
[27] M. Langheinrich, "A Privacy Awareness System for Ubiquitous Computing Environments", UbiComp 2002, Springer-Verlag, LNCS 2498, pp.237-245, 2002.
[28] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf, "Pseudonym Systems", Selected Areas in Cryptography '99, Springer Verlag Lecture Notes in Computer Science No. 1758; edited by H. Heys and C. Adams, 2000, pp. 184-199.
[29] A. Lysyanskaya, R. Rivest, A. Sahai and S. Wolf, "Pseudonym Systems", In Proc. of Selected Areas in Cryptography 1999, pp.184-199, 1999.
[30] A. Menezes *et al.*, "*Handbook of Applied Cryptography*", CRC Press, 1997
[31] K. Nakanishi, J. Nakazawa and H. Tokuda,"LEXP: Preserving User Privacy and Certifying Location Information", The 2nd Workshop on Security (Ubicomp2003), 2003.
[32] D. Park, "Cryptographic Protocols for Third Generation Mobile Communication Systems", PhD Thesis, Queensland University of Technology, Australia, 2001.
[33] Ron Rivest, "Electronic Voting", available at http://theory.lcs.mit.edu/ rivest/Rivest-ElectronicVoting.pdf.
[34] R. Rivest, A. Shamir and L. Adleman, A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Commun. ACM, Vol. 21, pp. 120-126, 1978.
[35] R. Rivest, The MD5 Message Digest Algorithms, IETF RFC 1321,1992.
[36] L. Lamport, "Password authentication with insecure communication", Communications of the ACM, 24(11), pp.770-771, Nov., 1981.
[37] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges", *IEEE Personal Communications*, Aug., 2001.
[38] A. Weimerskirch and D. Westhoff, "Zero common-knowledge authentication for pervasive networks", In Proceedings of Selected Areas of Cryprotgraphy 2003 (SAC 2003), 2003.
[39] M. Weiser, "The Computer for the 21st Century", *Scientific American*, Sep, 1991.
[40] M. Wu and A. Friday, "Integrating Privacy Enhancing Services in Ubiquitous Computing Environments", Workshop on Security in Ubiquitous Computing, 4th International UBICOMP, 2002.
[41] A. Zugenmaier, A. Hohl, "Anonymity for Users of Ubiquitous Computing", Security Workshop, UbiComp 2003, Seattle, October 2003.