A Novel Privacy Preserving Authentication and Access Control Scheme for Pervasive Computing Environments

Kui Ren, *Student Member, IEEE*, Wenjing Lou, *Member, IEEE*, Kwangjo Kim, *Member, IEEE*, and Robert Deng, *Senior Member, IEEE*

***Abstract*—Privacy and security are two important but seemingly contradictory objectives in a pervasive computing environment (PCE). On one hand, service providers want to authenticate legitimate users and make sure they are accessing their authorized services in a legal way. On the other hand, users want to maintain the necessary privacy without being tracked down for wherever they are and whatever they are doing. In this paper, a novel privacy preserving authentication and access control scheme to secure the interactions between mobile users and services in PCEs is proposed. The proposed scheme seamlessly integrates two underlying cryptographic primitives, namely blind signature and hash chain, into a highly flexible and lightweight authentication and key establishment protocol. The scheme 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. The correctness of the proposed authentication and key establishment protocol is formally verified based on Burrows–Abadi–Needham logic.**

***Index Terms*—Access control, authentication, pervasive computing environments (PCEs), security.**

I. Introduction

PERVASIVE computing environments (PCEs) with their interconnected devices and abundant services promise great integration of digital infrastructure into many aspects of our lives, from our physical selves to homes, offices, streets, and so forth [1], [38]. The huge number of communicating devices provides seamless access to multiple dynamic networks any time from 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 intelligence in PCEs [2], [4]–[6], [41]. 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 preestablished (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 "prior" to the access of services. Traditional authentication that 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 noncritical 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 the users' privacy a much more difficult task [10], [15], [39].

Some of the user privacy issues that should be treated in PCEs have 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 divulge such

Manuscript received November 9, 2004; revised July 28, 2005; and November 25, 2005. The review of this paper was coordinated by Dr. K. Martin.

K. Ren and W. Lou are with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA 01609-2280 USA (e-mail: kren@ece.wpi.edu; wjlou@ece.wpi.edu).

K. Kim is with the Information and Communications University, Daejeon 305-732, Korea (e-mail: kkj@icu.ac.kr).

R. Deng is with the School of Information Systems, Singapore Management University, Singapore 188065 (e-mail: robertdeng@smu.edu.sg).

Digital Object Identifier 10.1109/TVT.2006.877704

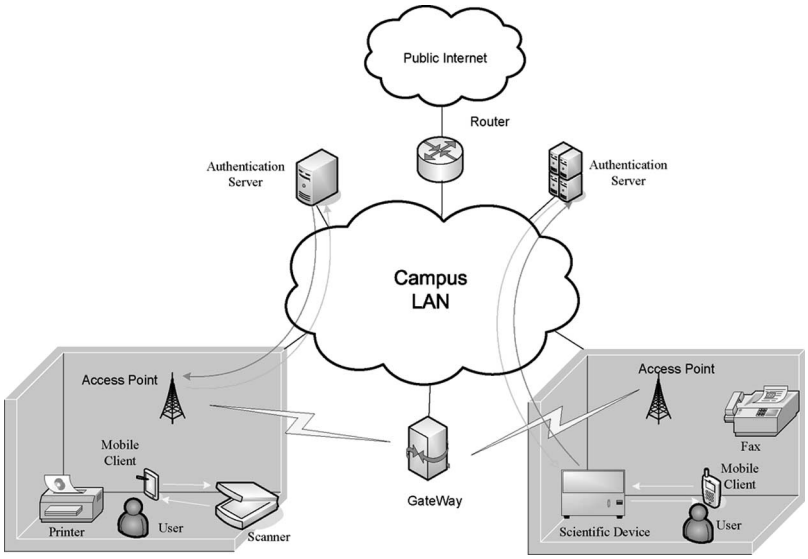


Fig. 1. Sample PCE.

information. Users' context information should be protected against both outsiders and service providers they interact with.

— Confidentialityand 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 highly complex in PCEs as the context information of users is more of a concern. On one hand, the service generally depends on the user identity information and corresponding preestablished 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 tradeoff between the two thus poses a great challenge to security designers.

In this paper, we propose a user privacy preserving authentication and 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., as 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 the desired service anonymously without revealing her identity. The scheme seamlessly integrates two underlying cryptographic primitives, namely 1) "blind signature" and 2) "hash chain," into a highly flexible and lightweight authentication and key establishment protocol. The scheme possesses many desirable security properties, such as anonymity, nonlinkability, nonrepudiation, accountability, differentiated services access control, etc., with very low protocol complexity (refer to Section V). The correctness of the proposed authentication and key establishment protocol is also formally verified based on Burrows–Abadi–Needham (BAN) logic. To the best of our knowledge, this work is the first attempt to an authentication and key establishment protocol with formally verified correctness for privacy preserving access control to differentiated services.

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 and show the formal verification of its correctness in Section IV. Then, we discuss the security features and the performance of the proposed scheme in Section V. Finally, we review the related work in Section VI and conclude the paper in Section VII.

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 types of entities, namely 1) mobile users, 2) services, and 3) 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 specifically, our scheme aims to provide anonymous mutual authentication between the mobile user and the service (e.g., wireless service access point for wireless network access service). It also provides confidentiality and integrity protection for the communications between the mobile user and the service.

Our scheme is based on two cryptographic techniques, namely 1) blind signature and 2) hash chain. A brief review of the two techniques is provided as follows.

A. Blind Signature

The blind signature scheme [16] is a variation of the digital signature scheme in which the content of a message is disguised

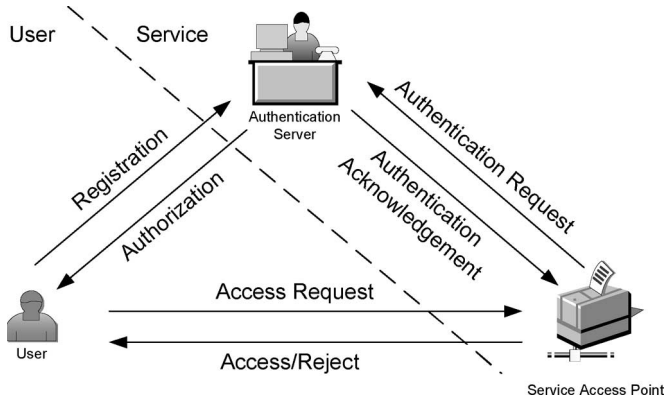


Fig. 2. System architecture.

from its signer. Blind signature schemes can be implemented based on a number of well-known digital signature schemes, such as the Rivest–Shamir–Adleman (RSA) scheme [33]. 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. The signer signs the blinded message using a standard signing algorithm, say $S_A(m)$, 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 nonlinkability, which prevents the signer from linking a 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].

B. Hash Chain

One-way hash function h is a powerful yet computationally efficient cryptographic tool, which takes a message of arbitrary size as its input and outputs a fixed string of digits. "One way" means that it is computationally infeasible to derive the original input from the output. By applying $h()$ repeatedly on an initial value m, one can obtain a chain of outputs $h^j(m)$. These outputs can be used in the reverse order of generation for the purpose of authentication: $h^{j-1}(m)$ can be proven to be authentic if $h^j(m)$ has been proven to be authentic due to the one-wayness property of hash function. Hash chains together with signatures are widely used in micropayment schemes such as Payword, *i*KP, and Netcard [31]. 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 a hash chain was first proposed for use in an authentication scheme by Lamport [35]. Recently, Weimerskirch and Westhoff adopted a hash chain technique for efficient user recognition based on weak authentication [37].

III. Proposed Scheme

This section presents our privacy preserving authentication and access control scheme. Consider the scenario that a mobile user wants to be able to dynamically access 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 full control of her context privacy. That is, the mobile user's context information like location, time, transaction profiles, etc., should be known only 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) providing explicit mutual authentication between the mobile user and the service; 2) allowing mobile users to anonymously interact with the service; 3) enabling differentiated service access control among different users; 4) providing flexibility and scalability to both user and service sides; 5) generating fresh session keys to secure the interaction if applicable; 6) having high efficiency in terms of communication, computation, and management overheads; 7) providing easy accountability; and 8) providing formally verified correctness.

Conceptually, the proposed scheme works as follows: The mobile user first generates some specific credentials (as will be described in Section III-A), and then she gets 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 the mutual authentication process, both parties will share fresh session keys that 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 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)[1] frames. This assumption is a prerequisite for anonymous communication; otherwise, one can easily identify a user based on her unique MAC address. Detailed technique on this can be found, for example, in [20] and is out of the scope of this paper.

A. User Authorization Protocol

The purpose of the user authentication protocol is to establish 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

[1]Note that MAC is used to stand for both message authentication code and medium access control.

TABLE I
NOTATION

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* [34].
$h_{K_{AB}}(m)$	A cryptographic secure Message Authentication Code (MAC) algorithm, computing the message digest of message m with key K_{AB}.
$h^j(m)$	Hash message m j times: $h^1(m) = h(m)$, $h^j(m) = h(h^{j-1}(m)), j = 2, 3, 4, ...$
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 II
CREDENTIAL GENERATION

Credential Generation:
1. generate two fresh nonces: r'_U and r''_U.
2. sign her own ID with a fresh nonce r''_U:$\{U, r''_U\}_{PriK_U}$.
3. 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 III
CREDENTIAL AUTHORIZATION

Credential Authorization:

U		S
	$U, C_U, CertU, SID$ →	
	authorization request	
		6. Verify $CertU$ with $PubK_S$.
		7. Sign C_U: $C_S = \{C_U\}_{PriK_{SID}}$ $= r'_U \times \{C^n\}_{PriK_{SID}}$.
	C_S ←	
	authorization confirmation	

8. Decrypt and verify U, S.
9. Compute C_S/r'_U and obtain $(C^n, \{C^n\}_{PriK_{SID}})$.

protocol, the mobile user and the corresponding service (i.e., the authentication server) need to authenticate each other first. This is typically done through some out-of-band noncryptographic technique. The mobile user needs to register herself as a legal user of some service type the service 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 $PriK_S$. 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. The user's context information can therefore be concealed from the service. Further, through a deliberately designed combination with hash chain technique, a series of authorized credentials, i.e., a credential chain[2], can be obtained by the mobile user in one protocol run, which increases the protocol efficiency.

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:

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 a nonrepudiation 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 the credential chain tail C^n by using the 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.

- Besides the general public key $PubK_S$ for user authentication purposes, the service maintains a pool of public keys corresponding to different 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 different private keys. This allows for differentiated access control in the subsequent stage. If the scope and the meaning of service types are 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.

[2]The same notion is used in [43] but with a different definition. In [43], a credential chain means a delegation chain from the source of authority to the requester.

- 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 blind signature. Therefore, after protocol execution, the mobile user holds a verifiable authenticator—credential C^n and its signature. Although the authentication server does not 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 nth credential C^n, the remaining hash chain values C^0 through 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 several 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 authorized credentials from the service provider. Note that the user authorization protocol runs only when the mobile user's authorized credentials are exhausted or for first-time registration. The user authorization protocol is highly flexible. It can be accomplished via both online and offline approaches (i.e., in-person interaction and so on). 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 authorized credentials from 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.

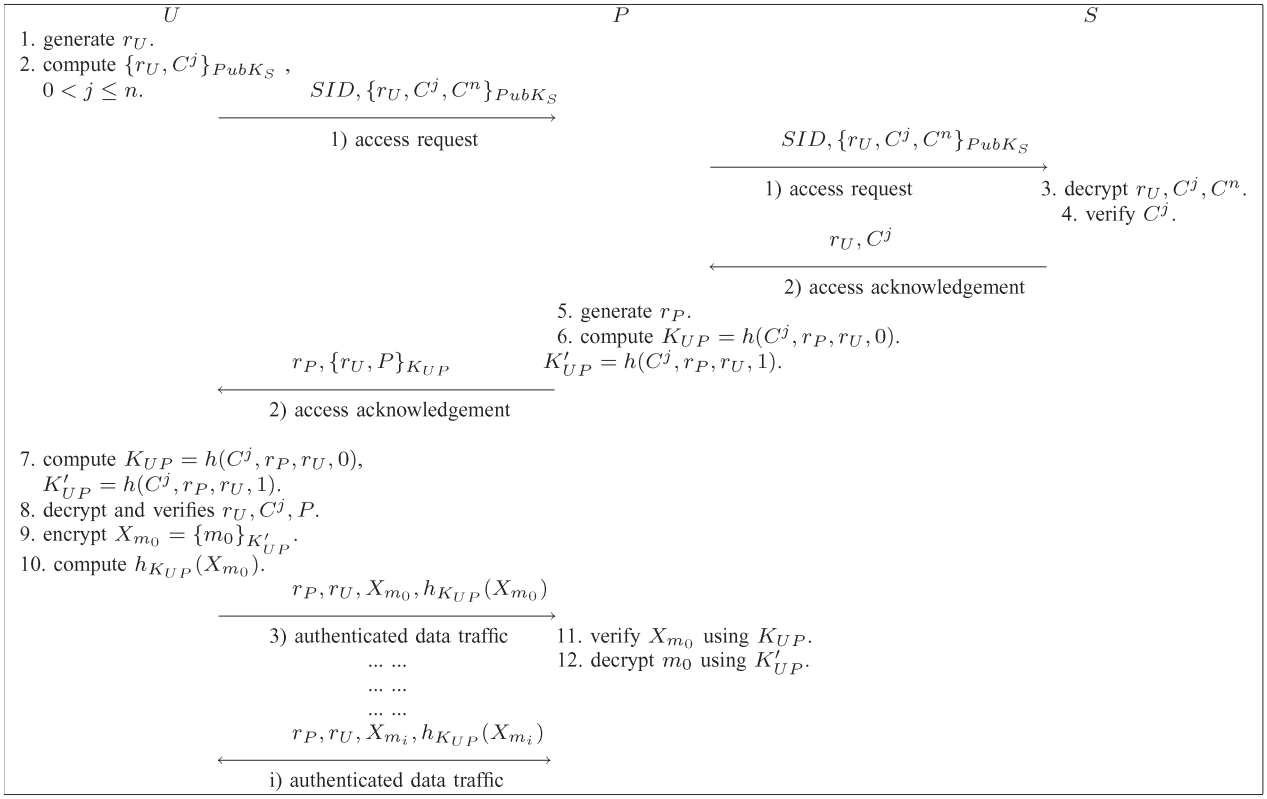
B. User Operational Protocol

The user operational protocol allows a mobile user to safely enjoy different kinds of services she is authorized to in PCEs 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, to the service access point such as the wireless network access point or a network printer. The authenticator includes an authorized credential and a fresh nonce. 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 to 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 (i.e., encryption key and integrity key) with the obtained secret information and generates a fresh nonce of its own. Next, the service access point encrypts the obtained secret information and its own identity with one of the new session keys (i.e., encryption key) and finally replies to 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 two session keys in the same manner. After decrypting the access grant message with the shared encryption 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 parties share two fresh session keys, 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 1), the mobile user encrypts a fresh nonce r_U and an authorized credential C^j with the service's public key that is used for authentication purposes. 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 the 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 for authentication. In this case, the access request message 1) 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 afterward. 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 a 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, $(C^n, \{C^n\}_{PriK_{SID}})$,

TABLE IV
USER OPERATIONAL PROTOCOL



the authentication server verifies the signature using the corresponding public key by referring to SID 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. 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 could not 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 through the submitted credential. Hence, a differentiated service access control is easily realized. If the submitted credential is a single value C^j, the back-end server simply verifies whether $h(C^j)$ matches the currently stored credential whose belonging hash chain is indexed by C^n. The authentication server then updates the currently stored C^{j+1} with C^j. The remaining operation is the same as above. 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 a 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 minimized, and little storage capability is required: 1) no public key operations; 2) no long-term key and certificate management; 3) session keys are discarded once the session is terminated; 4) hash and symmetric key operations only. Hence, it is very simple and efficient, which could greatly decrease its cost and help wide deployments.
- The service access point and the mobile user compute the fresh session keys independently, and the 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) [34].
- The fresh nonces 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 micropayment schemes, the accounting mechanism can be easily incorporated into the system in nearly the same manner. We point out that double spending of the authorized credentials actually does not affect the system security as proved in Section V. Hence, the choice between one-time or multiple usage of the authorized credentials can be a simple policy decision. It can also be dynamically switched according to real situations.

C. Extension for Out-of-Order Requests

Sometimes, a mobile user might want to launch multiple sessions simultaneously. Note that if the multiple sessions are with respect to different service types, or if the multiple sessions are of the same service type but come to the authentication server in the same order as they were originated, the proposed protocol can handle them well. However, if the multiple concurrent sessions are with respect to a single service, but for some reason (e.g., unexpected network problems, DoS attacks) the access request messages arrive out of order at the back-end authentication server, one or more legitimate requests will be deemed illegal by the user operational protocol we just described. In this subsection, we present a simple extension to the user operational protocol to deal with such out-of-order arrival of access requests at the authentication server.

Our solution is a sliding-window-based extension to the credential authentication procedure at the authentication server. Recall that in the previous setting, each hash chain stores two values: C_n, used as the index of the hash chain, and C_j, the most recently used hash value. A submitted credential is hashed only once and compared with C_j. In the extension, the back-end authentication server is allowed to store up to $k+1$ hash values for each hash chain: C_n and a window of up to k hash values. A submitted credential may be hashed up to $k-1$ times; once a hashed value is found equal to a value already in the window, the credential may be accepted. The extension works as follows: Obviously, at the beginning, when all the requests come in order, there is only one value kept in the window—the most recently used value C_j. At this time, when the next credential C^{j_1} comes, 1) if it comes in order, namely $j_1 = j-1$ or $C^j = h(C^{j_1})$, the window will move forward by one place, which makes C^{j_1} the first value in the window while the rest of the places in the window remains empty; 2) if $j-k < j_1 < j-1$, which means that C^{j_1} is hashed more than once to be equal to C^j, the window remains where it is and the value C^{j_1} is saved at the corresponding location, i.e., $(j-j_1+1)$th place in the window; 3) otherwise, if a match is not found after k hashes, the credential is rejected. Similarly, when there are multiple credentials in the window, the following process applies for the next arrived credential C^{j_2}: 1) If C^{j_2} equals any of the existing credentials, it is a double submission and the corresponding session should be rejected. 2) If $j_2 = j-1$[3], the credential is valid and the window moves forward by one place, which makes C^{j_2} the first credential in the window. Further, if its next position (j_2-1) is not empty, the window will continue to move forward until it reaches a value C^{j_3} whose next position (j_3-1) is empty. This operation will ensure that the second location in the window is always empty. 3) If $j-k < j_2 < j-1$, the window remains unmoved, and the value C^{j_2} is saved at the corresponding empty location in the window. 4) Otherwise, the credential is simply rejected. Fig. 3 shows an example of how an out-of-order arrival of requests is handled by this extension.

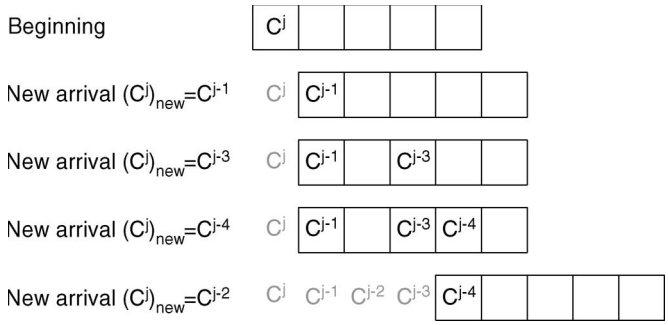


Fig. 3. Sliding-window-based credential authentication procedure (for $k=5$).

Note that in this extension, k $(k \ll n)$ is the maximum distance between two access requests that go out of order. The requests are of the same type and from the same mobile user. Since the time needed to complete the whole authentication is in a scale of milliseconds or at worst seconds, it is very unlikely that a mobile user can launch a large number of sessions and have them go out of order. Therefore, this number could be very small. Note that the number of concurrent sessions the user could have is not limited by k. A user may still have a large number of concurrent sessions as long as she initializes each of them with a small time separation.

In the proposed user operational protocol, we assume that an incomplete session will always be due to the failure of the mobile user. The packets lost in the network can be resent by reliable link/transport protocols or a failure notification will be returned to the source. If a mobile user gets no reply after sending an access request message, it would resend the same message again until obtaining a corresponding reply. However, to make the scheme highly robust to unexpected network problems, we may allow the window to move forward when the last position of the window is filled while the second position has yet not filled. On the other hand, as long as an access request message is successfully received and processed (replied) by the service access point (i.e., as long as an access acknowledgement message is sent back to the mobile user by the service side), the submitted credential is void thereafter even if the mobile user may fail to continue the subsequent session. Note that the access acknowledgement message may be resent many times before the session is aborted by the service access point. Since we generally consider a micropayment case, dropping a single credential occasionally does not affect much on accountability.

IV. Correctness Verification of the Proposed Scheme

In this section, we formally verify the correctness of the proposed user operational protocol based on the BAN logic [12],

[3]We still assume that C^j is the first credential in the window.

TABLE V
GOALS OF CORRECTNESS VERIFICATION

Verification Goals:	
1. $A \equiv A \overset{K_{ab}}{\longleftrightarrow} B$	2. $B \equiv A \overset{K_{ab}}{\longleftrightarrow} B$
3. $A \equiv B \equiv A \overset{K_{ab}}{\longleftrightarrow} B$	4. $B \equiv A \equiv A \overset{K_{ab}}{\longleftrightarrow} B$

which is a formal logic widely used to reason about beliefs, encryption, and protocols. Although BAN logic does have its own limitations, it is simple and has been successfully applied to many protocols. Protocol correctness means that, after protocol execution, both of the communication parties ascertain that they are sharing a fresh session key, and both are sure that the same belief is held by the other side too. Table V expresses this verification goal following the BAN logic. In order to eliminate the expression complexity, we simplify the two protocols into their generic types. In the following description in this section, we use the following notations by convention: A and B are two entities, K_{ab} is the fresh session key shared between A and B, and (K_a, K_a^{-1}) and (K_b, K_b^{-1}) are the public/private key pairs of entities A and B, respectively; other notations follow those of BAN logic [12].

In the user operational protocol, the service access point and the back-end authentication server trust the authenticity and integrity of the messages exchanged between them because a secure communication channel is assumed between them. Therefore, without loss of generality, we simplify the user operational protocol into the following generic type shown in Table VI, which is further idealized in the same table. We omit the unnecessary steps. We focus on the messages exchanged between the mobile user and the service access point and verify whether both parties can ascertain that they share a fresh session key K_{ab} with each other.

The assumptions we make before the verification are

1) $A \equiv \overset{K_b}{\mapsto} B$;
2) $B \equiv \overset{K_b}{\mapsto} B$;
3) $A \equiv \sharp(N_a)$;
4) $B \equiv \sharp(N_b)$;
5) $A \equiv A \overset{C^j}{\rightleftharpoons} B$;
6) $B \equiv A \overset{C^j}{\rightleftharpoons} B$;
7) $A \equiv B \equiv A \overset{C^j}{\rightleftharpoons} B$;
8) $B \equiv A \equiv A \overset{C^j}{\rightleftharpoons} B$;
9) $A \equiv \sharp(A \overset{C^j}{\rightleftharpoons} B)$;
10) $B \equiv \sharp(A \overset{C^j}{\rightleftharpoons} B)$;
11) $A \equiv B \Rightarrow A \overset{K_{ab}}{\longleftrightarrow} B$;
12) $B \equiv A \overset{K_{ab}}{\longleftrightarrow} B$;
13) $B \equiv A \sim (A \overset{K_{ab}}{\longleftrightarrow} B)$.

The first two assumptions state that both entities A and B believe that B possesses a public key K_b. Assumptions (3) and (4) mean that A and B generate two fresh nonces N_a and N_b, and therefore, assure their freshness. Assumptions (5)–(10) are about the authorized credentials shared between the two parties. When $j = n$, the authentication server believes that C^n is the secret shared between an authorized user and itself because it can easily verify the authenticity of C^n through the attached signature. So although the authentication server has no information about who the mobile user is, it still believes that C^n is an authentic secret shared between the two. The blind signature technique and our user authorization protocol ensure this very nice property. When $j < n$, the authentication server holds the same belief because of the one-wayness property of the hash chain. The C^n values, after verification, are stored in the authentication server to prevent the possible reuse of the same value[4], which ensures the freshness of the C^n values. Each $C^j \sim (j < n)$ is guaranteed to be fresh, e.g., used only once, because the matching window is constantly moving whenever a C^j value has been used. Formal verification on these beliefs on the hash chain scheme can be found in [37] and is omitted here due to space limitations. Assumption (11) tells that A believes B has jurisdiction right over K_{ab}, because once A generates N_a and sends it to B together with shared secret C^j, the value of the final K_{ab} is determined by the nonce N_b generated by B from the viewpoint of A. Assumptions (12) and (13) hold because B invents the fresh session key K_{ab} with a shared secret between A and B and a fresh nonce chosen by itself. The verification is outlined in Table VII.

TABLE VI
GENERIC TYPE OF THE USER OPERATIONAL PROTOCOL

Protocol Generic Type:	
Message 1	$A \longrightarrow B : \{N_a, C^j\}_{K_b}$
Message 2	$B \longrightarrow A : N_b, \{N_a, C^j, B\}_{K_{ab}}$
Message 3	$A \longrightarrow B : N_b, \{m\}_{K_{ab}}$
Session key:	$K_{ab} = h(N_a, N_b, C^j)$
Idealized protocol:	
Message 1	$A \longrightarrow B : \{N_a, A \overset{C^j}{\rightleftharpoons} B\}_{K_b}$
Message 2	$B \longrightarrow A : \{N_a, A \overset{C^j}{\rightleftharpoons} B, A \overset{K_{ab}}{\longleftrightarrow} B\}_{K_{ab}}$
Message 3	$A \longrightarrow B : \{A \overset{K_{ab}}{\longleftrightarrow} B\}_{K_{ab}}$

Equations (18), (19), (21), and (24) together accomplish the verification. Note that assumptions (9) and (10) are not used in the verification, which means that even if the authorized credential is not fresh, the correctness of the protocol still holds. This property implies that the authorized credentials can be reused without affecting the system security.

V. ANALYSIS OF THE PROPOSED SCHEME

A. Security-Related Properties of the Proposed Scheme

The proposed scheme exhibits 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

[4]In this case, a mechanism to obsolete old C^n values is necessary. For example, the authentication server can change the public/private key pairs for the service types periodically.

TABLE VII
VERIFICATION OF THE USER OPERATIONAL PROTOCOL

Verification:

Message 1 $A \longrightarrow B : \{N_a, A \overset{C^j}{\rightleftharpoons} B\}_{K_b}$

14. $B \lhd (N_a, A \overset{C^j}{\rightleftharpoons} B)$ //(2), Seeing rule

Message 2 $B \longrightarrow A : \{N_a, A \overset{C^j}{\rightleftharpoons} B, A \overset{K_{ab}}{\longleftrightarrow} B\}_{K_{ab}}$

15. $A \lhd \{N_a, A \overset{C^j}{\rightleftharpoons} B, A \overset{K_{ab}}{\longleftrightarrow} B\}_{K_{ab}}$

16. $A \models B \sim (N_a, A \overset{C^j}{\rightleftharpoons} B, A \overset{K_{ab}}{\longleftrightarrow} B)$ //(5),(15), Msg.-meaning rule

17. $A \models B \models (N_a, A \overset{C^j}{\rightleftharpoons} B, A \overset{K_{ab}}{\longleftrightarrow} B)$ //(3),(16), Nonce-veri. rule and Freshness rule

18. $A \models B \models A \overset{K_{ab}}{\longleftrightarrow} B$ //(17), Belief rule

19. $A \models A \overset{K_{ab}}{\longleftrightarrow} B$ //(11),(18), Jurisdiction rule

Message 3 $A \longrightarrow B : \{A \overset{K_{ab}}{\longleftrightarrow} B\}_{K_{ab}}$

20. $B \lhd \{A \overset{K_{ab}}{\longleftrightarrow} B\}_{K_{ab}}$

21. $B \models A \overset{K_{ab}}{\longleftrightarrow} B$ //(12)

22. $B \models \sharp(A \overset{K_{ab}}{\longleftrightarrow} B)$ //(13)

23. $B \models A \sim (A \overset{K_{ab}}{\longleftrightarrow} B)$ //(20),(21), Msg.-meaning rule

24. $B \models A \models A \overset{K_{ab}}{\longleftrightarrow} B$ //(22),(23), Nonce-veri. rule

corresponding private key. The mutual authentication is highly necessary in the PCEs as discussed before to prevent potential malicious attacks from 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, 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 outsiders. No third party has the ability to acquire the user's context information as all the interaction traffic is well protected.

Nonlinkability: Ideally, nonlinkability means that, for both insiders (i.e., service) and outsiders, 1) neither of them could ascribe any session to a particular user, and 2) neither of them could link two different sessions to the same user [42]. In the proposed scheme, ideal nonlinkability is achieved with respect to outsiders. Because the authorized credential is never transmitted in plain text and is always combined with fresh nonce in the message, an outsider cannot ascribe a session to a particular user, neither can he ascribe two sessions to the same user. Hence, users' transaction profiles are well protected. On the other hand, using the hashing chain could allow the service provider to link up to n sessions using the hash values from the same chain to the same user, where n is the length of the hash chain. However, the service provider cannot ascribe such information to a particular user due to the underlying blind signature technique used. In addition, such a linkage is limited to n sessions only; there is no relationship among different hash chains. Therefore, there is no interhash chain information that can be accumulated by the service provider. Hence, users' transaction profile can still be well protected from the service provider.

Accountability and Nontransferability Equivalency: In the proposed scheme, the credentials are authorized only when the mobile user is explicitly authenticated. The one-time usage property of the authorized credentials prevents the double spending problem and further provides good accounting capability, which allows the accounting function be easily incorporated[5]. Furthermore, from the service point of view, the proposed scheme provides equivalent nontransferability. That is, 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 cannot provide this property because there is no way to prevent the double spending problem and hence no way to prevent service abuse problem.

TABLE VIII
PROTOCOL SECURITY FEATURES COMPARISON

	This paper	[21]	[24]
Concrete Protocol	Yes	Yes	No
Mutual Authentication	Yes	Yes	No
User Context Privacy	Yes	Yes	Not to services
Non-Linkability	Yes to outsiders, partially yes to services	No	Not to services
Non-Transferability Equivalency	Almost 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

Data Traffic Protection: The user operational 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 symmetric cryptography.

Differentiated Service Access Control: By classifying mobile users into different service types, differentiated service access control is enabled in our scheme. Different mobile users are authorized accordingly based on the service types to which they belong. "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.

Formally Verified Correctness: The proposed scheme is secure as sound as the underlying cryptosystem against both passive and active attacks. We verified the correctness of the proposed scheme in the above section based on the well-known BAN logic.

We compare our scheme to other similar approaches that are intended to provide anonymous interactions between the users and the services (Table VIII). The advantage of our scheme is shown clearly.

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

TABLE IX
PROTOCOL COMPUTATION OVERHEADS COMPARISON

		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

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 minimal 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 tradeoff. 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 rounds to accomplish mutual authentication and session key establishment between the user and the service. Note that two rounds 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 offline. 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 IX 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 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 could therefore be the bottleneck of the whole system due to the potentially 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 great trouble to perform public key operations for every user in every session due to its constrained computation capability.

VI. RELATED WORK

Recently, quite a few papers have been published to address the new security and privacy challenges in PCEs [7]–[9], [11], [15], [19], [21], [24], [25], [27], [30], [39], [40]. 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 the MIX network style approach [14]. A proxy-based scheme can be found in [11]. Another recent infrastructure-based approach, LEXP, can be found in [11]. Some efforts try to maximize user privacy by restricting access to users' context information. Hengartner and Steenkiste suggested an architecture to filter out user context information [22].

The remaining efforts mostly focused on identity manipulation approaches, 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 cannot 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 a credential. An in-depth description and analysis of different pseudonym systems can be found in [28].

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 did not give any concrete protocols. 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 V.

Henrici and Muller [23] utilized hash functions to recompute identifiers of a radio frequency identification (RFID) device every time it sends a request to service providers. Their intention was to protect the location privacy of RFID devices. Another approach proposed by Weimerskirch *et al.* uses hash functions to realize efficient weak authentication [37]. In order to avoid leakage of user's MAC address or IP address at lower levels, Gruteser and Grunwald [20] came up with a method to hide user's MAC address with anonymous IDs so that the user cannot be tracked in a wireless LAN environment.

VII. Conclusion

In pervasive computing, mobile users interact seamlessly with abundant services anytime, anywhere. However, user privacy is at great risk in PCEs because of its inherent pervasive and heterogeneous nature. Legitimate service providers may also suffer from abuse from unauthorized and malicious users. The conflict between user privacy protection and user authentication makes 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 one hand, the proposed scheme provides explicit mutual authentication between a mobile user and a service; on the other hand, 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, namely 1) blind signature and 2) 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. The correctness of the proposed authentication and key establishment protocol was formally verified based on BAN logic.

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 context-aware services without affecting user privacy is our future work.

Acknowledgment

The authors would like to thank anonymous referees for their constructive comments that helped improve the manuscript.

References

[1] "Easy living," *Microsoft Research*. [Online]. Available: http://research.microsoft.com/easyliving/

[2] *GAIA—Active Spaces for Ubiquitous Computing*, Univ. Illinois at Urbana-Champaign. [Online]. Available: http://choices.cs.uiuc.edu/gaia/

[3] *Location Privacy Protection Act And Other Privacy Related Law*. [Online]. Available: http://www.techlawjournal.com/cong107/Privacy

[4] *MIT Project Oxygen*. [Online]. Available: http://oxygen.lcs.mit.edu/

[5] National Institute of Standards and Technology (NIST), *Pervasive Computing SmartSpace Laboratory*. [Online]. Available: http://www.nist.gov/smartspace/

[6] *The Aware Home*, Georgia Inst. Technol. [Online]. Available: 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," in *Proc. ICDCS*, Vienna, Austria, 2002, pp. 65–74.

[8] ——, "Routing through the mist: Design and implementation," Mar. 2002 UIUCDCS-R-2002-2267.

[9] J. Al-Muhtadi, A. Ranganathan, R. Campbell, and M. Mickunas, "A flexible, privacy-preserving authentication framework for ubiquitous computing environments," in *Proc. ICDCS Workshops*, 2002, pp. 771–776.

[10] ——, "Cerberus: A context-aware security scheme for smart spaces," in *Proc. PerCom*, 2003, p. 489.

[11] M. Burnside *et al.*, "Proxy-based security protocols in networked mobile devices," in *Proc. ACM SAC*, Madrid, Spain, 2002, pp. 265–272.

[12] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," in *Proc. Royal Soc. London A*, 1989, vol. 426, pp. 233–271.

[13] L. Bussard and Y. Roudier, "Authentication in ubiquitous computing," in *Proc. Workshop Security UBICOMP*, Goteborg, Sweden, 2002.

[14] J. Camenisch and A. Lysyanskaya, "Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation," in *Proc. Advances Cryptology, EUROCRYPT*, 2001, vol. 2045, pp. 93–118.

[15] R. Campbell, J. Al-Muhtadi, P. Naldurg, G. Sampemane, and M. Mickunas, "Towards security and privacy for pervasive computing," in *Proc. ISSS*, 2002, pp. 1–15.

[16] D. Chaum, "Blind signatures for untraceable payments," in *Proc. Crypto—Advances Cryptology*, D. Chaum, R. L. Rivest, and A. T. Sherman, Eds. New York: Plenum, 1982, pp. 199–203.

[17] ——, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–88, Feb. 1981.

[18] ——, "Security without identification: transaction systems to make big brother obsolete," *Commun. ACM*, vol. 28, no. 10, pp. 1030–1044, Oct. 1985.

[19] S. Creese, M. Goldsmith, B. Roscoe, and I. Zakiuddin, "Authentication for pervasive computing," in *Proc. Security in Pervasive Computing 2003*, 2004, vol. 2802, pp. 116–129.

[20] M. Gruteser and D. Grunwald, "Enhancing location privacy in wireless lan through disposable interface identifiers: A quantitiative analysis," in *Proc. WMASH*, San Diego, CA, 2003.

[21] Q. He *et al.*, "The quest for personal control over mobile location privacy," *IEEE Commun. Mag.*, vol. 42, no. 5, pp. 130–136, May 2004.

[22] U. Hengartner and P. Steenkiste, "Access control to information in pervasive computing environments," in *Proc. 9th Workshop HotOS IX*, Lihue, HI, May 2003.

[23] D. Henrici and P. Muller, "Tackling security and privacy issues in radio frequency identification devices," in *Proc. PERVASIVE*. New York: Springer-Verlag, 2004, vol. 3001, pp. 219–224.

[24] U. Jendricke, M. Kreutzer, and A. Zugenmaier, "Pervasive privacy with identity management," in *Proc. 1st Workshop Security, UbiComp*, 2002.

[25] ——, "Mobile identity management," in *Proc. 1st Security Workshop, UBICOMP*, 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," in *Proc. UbiComp*, 2002, vol. 2498, pp. 237–245.

[28] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," in *Selected Areas in Cryptography*, H. Heys and C. Adams, Eds. New York: Springer Verlag, 2000, pp. 184–199.

[29] A. Menezes *et al.*, *Handbook of Applied Cryptography*. Boca Raton, FL: CRC, 1997.

[30] K. Nakanishi, J. Nakazawa, and H. Tokuda, "LEXP: Preserving user privacy and certifying location information," in *Proc. 2nd Workshop Security Ubicomp*, 2003.

[31] D. Park, "Cryptographic protocols for third generation mobile communication systems," Ph.D dissertation, Queensland Univ. Technol., Brisbane, Australia, 2001.

[32] R. Rivest, *Electronic Voting*. [Online]. Available: http://theory.lcs.mit.edu/~rivest/Rivest-ElectronicVoting.pdf

[33] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[34] R. Rivest, "The MD5 message digest algorithms," *IETF RFC 1321*, 1992.

[35] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–771, Nov. 1981.

[36] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 10–17, Aug. 2001.

[37] A. Weimerskirch and D. Westhoff, "Zero common-knowledge authentication for pervasive networks," in *Proc. SAC*, 2003, pp. 73–87.

[38] M. Weiser, "The computer for the 21st century," *Sci. Amer.*, vol. 265, no. 3, pp. 94–104, Sep 1991.

[39] M. Wu and A. Friday, "Integrating privacy enhancing services in ubiquitous computing environments," in *Proc. 4th Int. UBICOMP Workshop Security Ubiquitous Comput.*, 2002.

[40] A. Zugenmaier and A. Hohl, "Anonymity for users of ubiquitous computing," in *Proc. Security Workshop—UbiComp*, Seattle, WA, Oct. 2003.

[41] M. Raisinghani, A. Benoit, J. Ding, M. Gomez, K. Gupta, V. Gusila, D. Power, and O. Schmedding, "Ambient intelligence: Changing forms of human-computer interaction and their social implications," *J. Digital Inf.*, vol. 5, no. 4, p. 271, Aug. 24, 2004.

[42] S. Xu and M. Yung, "K-anonymous secret handshakes with reusable credentials," in *Proc. ACM Conf. CCS*, 2004, pp. 158–167.

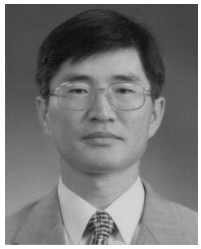
[43] N. Li, W. Winsborough, and J. Mitchell, "Distributed credential chain discovery in trust management," in *Proc. ACM Conf. CCS*, 2001, pp. 156–165.

Kui Ren (S'04) received the B.Eng. and M.Eng. degrees from Zhejiang University, Zhejiang Province, China, in 1998 and 2001, respectively. He is currently working toward the Ph.D. degree at the Electronics and Communications Engineering Department, Worcester Polytechnic Institute, Worcester, MA.

He was a Research Assistant with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China, from March 2001 to January 2003; the Institute for Infocomm Research, Singapore, from January 2003 to August 2003; and the Information and Communications University, Daejeon, Korea, from September 2003 to June 2004. His research interests include *ad hoc*/sensor network security, wireless mesh network security, Internet security, and security and privacy in ubiquitous computing environments.

Wenjing Lou (M'03) received the B.E. and M.E. degrees in computer science and engineering from Xi'an Jiaotong University, Xi'an, China, in 1993 and 1996, respectively, the M.A.Sc. degree from the Nanyang Technological University, Singapore, in 1998, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in 2003.

From December 1997 to July 1999, she was a Research Engineer in the Network Technology Research Center, Nanyang Technological University. She is currently an Assistant Professor in the Electrical and Computer Engineering Department, Worcester Polytechnic Institute, Worcester, MA. Her current research interests are in the areas of *ad hoc* and sensor networks with emphases on network security and routing issues.



Kwangjo Kim (M'93) received the B.Eng. and M.Sc. degrees from Yonsei University, Seoul, Korea, in 1980 and 1983, respectively, and the Ph.D. degree from Yokohama National University, Yokohama, Japan, in 1991.

He is currently a Full Professor at the Information and Communications University (ICU), Daejeon, Korea, the Director of the International Research Center for Information Security, ICU, and the Chair of the Asiacrypt Steering Committee. He has led Coding Technique Section #1 with the Electronics and Telecommunication Research Institute and served as the Director of the International Association for Cryptologic Research and Institute for IT-Gifted Youth at the ICU. He has more than 20 patents and more than 150 technical publications in international conferences and journals in the areas of crytography and information security and has edited three volumes of *Lecture Notes in Computer Science* by Springer Verlag.

Dr. Kim has served as General Chair, Program Chair, and Program Committee Member of numerous international conferences.

Robert Deng (SM'04) received the B.Eng. degree from the National University of Defense Technology, Hunan, China, and the M.Sc. and Ph.D. degrees from the Illinois Institute of Technology, Chicago.

He was a Principal Scientist and the Manager of the Infocomm Security Department, Institute for Infocomm Research. He is currently a Professor and the Director of the SIS Research Center, School of Information Systems, Singapore Management University, Singapore. He has more than 20 patents and more than 140 technical publications in international conferences and journals in the areas of digital communications, network and distributed system security, and information security.

Dr. Deng has served as General Chair, Program Chair, and Program Committee Member of numerous international conferences. He received the University Outstanding Researcher Award from the National University of Singapore in 1999.