A Capability-based Privacy-preserving Scheme for Pervasive Computing Environments

Divyan M. Konidala*, Dang N. Duc*, Dongman Lee†, and Kwangjo Kim*
*Cryptology and Information Security Lab, International Research Center for Information Security
†Collaborative Distributed Systems and Networks Lab
Information and Communications University, Daejeon, Republic of Korea
Email: {divyan, nguyenduc, dlee, kkj}@icu.ac.kr

***Abstract*— Pervasive computing is a future technology that provides a user with the capability to compute and communicate from everywhere. In a pervasive computing environment, a user interacts with many smart devices around him to obtain some useful services from them. These smart devices can be either genuine or malicious. Users privacy is at a greater risk, as they are prone to revealing their location, identity and transactions information to such devices. On the other hand, user authentication is also very much required for authorization and service access control. This allows smart devices to provide services to only authorized users. However, authentication and privacy protection are conflicting issues. In order to protect users from invasion of privacy, they must be allowed to interact anonymously with other smart devices. Thus, authenticating and authorizing an anonymous user becomes a challenging task. In this paper, we propose a simple and efficient "capability-based privacy-preserving" scheme for pervasive computing environments. The proposed scheme allows a user to anonymously interact with service providers or other smart devices and also allows the service providers to effectively authenticate and authorize users based on the anonymous information submitted by them.**

I. Introduction

Pervasive computing [22], [20] or ubiquitous computing means the availability of computing and communication resources whenever and wherever we are. A pervasive computing environment is saturated with devices, which compute and communicate "for", "on behalf" and "along with" users in order to provide some useful services. A user should obtain and make use of such services seamlessly and comfortably, but should never be burdened with instructions and interfaces on how to handle those devices. Nowadays mobile phones and PDAs are part and parcel of our lives. Apart from helping us to communicate, these mobile devices would very soon allow us to interact with other smart devices around us, thus supporting a pervasive computing environment. The scenarios provided below would help us to better understand the concept of a pervasive computing environment.

A pervasive computing environment [29], [28], [34] can be a meeting room (Smart Space) [33] that automatically provides a speaker the access to a beam projector and a computer, takes meeting minutes, takes commands from attendees and applies them differently depending on who spoke, *etc*. It can also be an office network [32], where users can easily locate and message their colleagues who are on the move, can use their portable mobile devices to have an on-the-fly access to documents, e-mails, software, security clearance and the nearest available printer, fax machine, and vending machine, *etc*. A user in such an environment need to just concentrate on his work and let the devices do their job seamlessly.

Among many security requirements of a pervasive computing environment, this paper focuses on user authentication, authorization and privacy protection.

- User authentication to corroborate of the identity of a user [15]. In a pervasive computing environment, user authentication establishes trust among communicating users and other smart devices. It allows service providers to interact with only known and trusted users, and prevents malicious users from trying to impersonate genuine ones. User authentication is very much required for providing authorization and service access control to the users.
- User authorization checks whether an authenticated user is indeed permitted to access the service he is currently requesting for. Only when the user is authenticated and authorized, the service provider provides the user the access control to the service he is requesting for. For example, in an office pervasive computing environment, the access rights of staff, manager and president are different. In an airport pervasive computing environment, first class passengers can access some extra special services than economy class passengers.
- User privacy protection is one of the big forthcoming challenges for actually deploying pervasive computing services on a significant scale. In the environments with significant concentration of "invisible" computing devices gathering and collecting users identities, their location and transactions information, the user should rightly be concerned for their privacy. This personal information could allow service providers and eavesdroppers to generate detailed profiles of the user, his buying interests, trace all his actions, and even hack in to users Wireless Personal Area Network (WPAN) [30]. As a result restricted access to users personal data [21] should be provided by all protocols executing in a pervasive computing environment.

In most pervasive computing environments, it is desirable that the user interacts anonymously with the service providers or other smart devices. In anonymous interactions the user

never communicates his real identity to the service providers. This would certainly protect user's privacy. But the catch is, if the user is not revealing his real identity to the service provider, how the service provider can trust the user to be genuine and check whether he is allowed to access that particular service or not. In other words, it would be difficult to accomplish user authentication, authorization, and service access control. Our scheme focuses on resolving this conflicting nature of user authentication, authorization and privacy protection. The scheme provides user anonymity and the service providers would still be able to authenticate, authorize and provide service access control to the users based on the anonymous information submitted by them.

The privacy issue in pervasive computing environments is much more complex than in the traditional environments due to its distributed and heterogeneous nature, and a large variety of different applications running on it. Although the privacy issue in pervasive computing environments received the most extensive concern in recent publications, still limited results have been yielded. Usually, the previous works have provided only frameworks instead of concrete protocol designs, none of them have considered privacy (anonymity), authentication and authorization all together, and are either too complex, or computationally expensive, or assume too much trust in the system, or not at all suitable to heterogenous nature of pervasive computing environments. This paper describes our simple and efficient concrete protocol design, which provides capability/credential based anonymous user authentication and authorization in pervasive computing environments. It has low computation and communication complexity, suitable to both low and high computing smart devices available in pervasive computing environments.

This paper is divided into the following sections. Section II briefly describes the background information pertaining to capability and blind-signature mechanism. This helps the reader to better understand our scheme. Section III provides detailed description of our proposed scheme including design considerations, system architecture, and protocol description. Section IV provides the analysis of our scheme with respect to security, complexity and comparison with related work. Section V mentions our further work and concludes this paper.

II. Background

This section briefly introduces "capability" and "blind-signature" concepts, as they form the basic building blocks of our proposed scheme.

A. Capability-based User Authentication and Authorization

In capability-based approach, a user holds his privileges to access some resources or services in the form of certificates or credentials called capabilities. In other words a capability is something that can be used to prove who you are, or prove that you are authorized to do something. The capabilities are issued to the user by an authorized entity/server, which is trusted both by the user and the service provider.

The authorized server issues capabilities to the user depending on his role in the pervasive computing environment, for *e.g.*, student, professor, staff, manager, visitor, first class passenger, *etc*. Capability is digitally signed (refer "digital signatures" in [16]) by the authorized server and it is unforgeable. It is stored in the user's portable mobile device. The user should present his capability to a service provider whenever he wants to access that service. The service provider first verifies if the capability is indeed issued (signed) by the authorized server and then verifies if the capability includes an entry (made by the authentication server) that authorizes the user to access the particular service he is requesting for. If so, it provides the service, otherwise, it denies service access control to the user. The advantage of this approach is that, when a user requests for a service from a service provider, the service provider need not communicate with the authentication server in order to authenticate and authorize the user, it needs to only verify the capability submitted by the user.

B. Blind-Signature

In 1982, David Chaum invented a new cryptographic primitive called blind-signature [8], which is still being used as a primer tool to design electronic payment and electronic voting schemes with user privacy-protection in mind. Blind-signature is a special kind of digital signature [16], which allows users to get signatures on their messages from authorized entities/signature issuers (*e.g.* banks, trusted third parties) without revealing the message contents to the authorized entity. Furthermore, if malicious signature issuers and verifiers (*e.g.* service providers, merchants) collude, they cannot discover the real identity of the user who actually holds the signatures. The detailed description of a blind signature scheme based on the RSA digital signature scheme is given in [8].

III. Proposed Scheme

This section provides detailed description of our proposed scheme, including design considerations, system architecture, and protocol description.

A. Design Considerations

The following design considerations support our scheme.

1) Targeted Level of User Privacy: The physical outreach of pervasive computing makes preserving user's privacy a much more difficult task [24], [6], [3]. Privacy is being considered as one of the fundamental security concerns that are explicitly identified by a series of laws [31]. Also privacy is a fuzzy term that is often overloaded to mean a large variety of things. Therefore, before proceeding any further, it is important to clarify the scope of user privacy that we strive to achieve in a ubiquitous computing environment. The real identity of a user should never be revealed from the communications exchanged between the user and service providers unless it is intentionally disclosed by the user or the system (with the consent of the user). The linkability among different communication sessions between the same user and service provider should not exist. It is not feasible to provide perfect privacy (anonymity). User

devices can still be traced at the link layer via MAC or IP addresses. As a result, considering link layer anonymity is beyond the scope of this paper. However high computation and communication intensive approaches like mix-networking [7], onion routing [18], crowds [19], and anonymizer [25] do address the link layer anonymity. Our scheme provides user authentication, authorization and privacy protection at the application layer.

2) Data Security Assumptions: Since our main goal is to provide user privacy protection, authentication and authorization, we assume that the pervasive computing environment already has some of the basic data security mechanisms, which provide data confidentiality and integrity for the communication channel among the interacting entities in the environment. We also assume that mechanisms for user to authenticate the service provider are already in place. These assumptions are considered in order to make the protocol explanation simple and to-the-point. Eventually a slight enhancement of our scheme can easily accomplish the above-assumed data security requirements.

3) Targeted Scope of Administrative Domains: Due to the heterogeneous nature of pervasive computing environments and to minimize the overhead involved in dealing with multiple administrative domains, we assume the existence of a medium sized (school campus/office) sized pervasive computing environment under a single administrative domain. Also in this campus sized pervasive computing environment there exist two different network technologies: WLAN (IEEE 802.11a/b/g) and LAN (Local Area Network). The user in this environment uses his mobile device such as a PDA or a mobile phone to access different services available to him through a wireless LAN.

4) Low Computation & Communication Overhead: We assume that the users in this environment carry a low-computing and resource-poor mobile device such as a mobile Phone and a PDA. These portable mobile devices should have the capability to compute in order to communicate anonymously with the system. A pervasive computing environment is heterogeneous in nature with both high and low computing smart devices. As a result, the schemes with cost effective (low computation and communication complexity) cryptographic techniques would best suit both low and high computing smart devices.

B. System Architecture

Considering a medium sized pervasive computing environment, our proposed scheme includes three entities: Authentication server, User and his portable mobile device, and Service Provider. Generally a pervasive computing environment or parts of a huge pervasive computing environment are managed by a Control Server/Main Server. It may mange naming system, service providers, service/resource discovery, context analysis and management, services update, and many more. As a result, it can also take up the job of being an authentication server. The "authentication server" issues capabilities to the user depending on his role in the environment for example student, professor, staff, manager, visitor, *etc*. They are stored in the user's mobile device. The "user using his mobile device" interacts with the service providers anonymously. The "service provider" receives the capability from the user and verifies the genuineness of the capability and accordingly decides whether to provide the service access control to the user.

C. Protocol Description

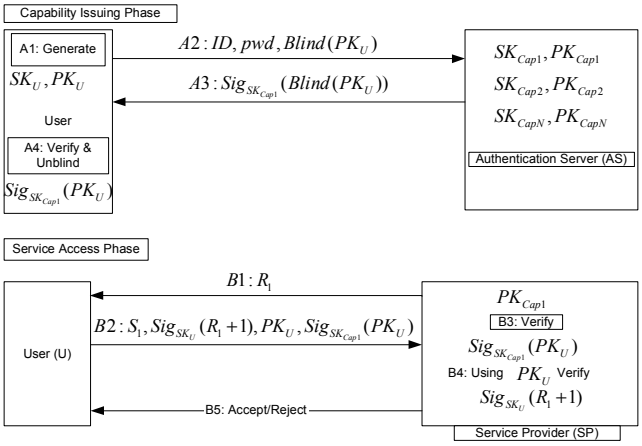

Fig. 1. System architecture

As shown in Figure 1, our proposed scheme involves two phases, "Capability Issuing phase" and "Service Access Phase". Before describing the two phases, we assume that, initially the user "Alice" personally registers with the authentication server via her true ID (for *e.g.* social security number, student ID, employee ID, *etc*.) and a password of her choice, accompanied with officially certified documents (from school administrative dept., company's human resource dept., *etc*.) proving her true ID. Authentication server after verifying the original documents, identifies Alice with a particular role in the organization/pervasive computing environment (for *e.g.* student, professor, staff, manager, visitor, *etc*). The authentication server then stores its public key PK_{AS} in Alice's mobile device. This step is taken, so that Alice can make use of PK_{AS} to verify the future digitally signed communications (using authentication server's secret key SK_{AS}) from the authentication server. The above registration procedure can be carried out securely via online, or offline, or in an entirely different way depending on the environment's security policy. The bottom line is that the authentication server must register the real ID, password and the role of the user in the environment and should be able to transfer its public key to user's mobile device. As a result any secure registration procedure can be employed.

1) Capability Issuing Phase: One fine day, after the registration procedure, Alice decides to make use of certain services in the pervasive computing environment. But in order to make use of services from different service providers in the environment, she needs to have her own "capability" issued by the authentication server. As a result, Alice undergoes the following steps to get a capability from the authentication server.

Step A1: In this phase, Alice generates a public-key PK_U and its corresponding secret-key SK_U. Using a blind-signature scheme, she blinds the PK_U as $Blind(PK_U)$.

Step A2: She sends her real ID, password pwd, and $Blind(PK_U)$ to the authentication server, thus requesting for a capability to be issued. The previously registered ID and password are used for authenticating the user to the authentication server.

Step A3: Authentication server checks the ID and password of the user and verifies if the ID is a member of the organization. If yes, it retrieves the role of the ID in the organization and according to the ID's role, the authentication server signs on $Blind(PK_U)$ with the corresponding capability's secret key. The authentication server stores a set of public and secret-key pairs of different versions of capabilities. For example, $Cap1$ includes permission to access services with service IDs $\{S_1, S_3, S_6, S_{12}\}$ and $Cap1$ may be suitable to all the students in the campus. Similarly $Cap2$ includes permission to access services IDs $\{S_1, S_6, S_{28}, S_{30}\}$, which may be suitable to all the managers in the office. Therefore if the authentication server signs $Blind(PK_U)$ with the secret key SK_{Cap1} of $Cap1$, then the Alice, who happens to be a student, can access the services $\{S_1, S_3, S_6, S_{12}\}$.

Step A4:. Alice receives $Sig_{SK_{Cap1}}(Blind(PK_U))$ and verifies the signature on message A3 using PK_{AS}. Since PK_U is blinded, the authentication server does not know the value of PK_U. But the speciality of the blind signature scheme is that, the authentication server can still sign on $Blind(PK_U)$ as $Sig_{SK_{Cap1}}(Blind(PK_U))$, without even knowing the value of PK_U. User un-blinds $Sig_{SK_{Cap1}}(Blind(PK_U))$ to obtain signature of capability1 directly on PK_U. PK_U is now exposed, and as a result $Cap_U = Sig_{SK_{Cap1}}(PK_U)$ becomes the personalized capability only for Alice.

Authentication server distributes the public-key PK_{Cap1} of $Cap1$ to service providers which are providing services $\{S_1, S_3, S_6, S_{12}\}$. When a user submits Cap_U to a service provider, the service provider uses PK_{Cap1} to verify the authentication server's signature on Cap_U. The capability issued by the authentication server to Alice has Time to Live value, after which the capability expires. Therefore depending on the computational capability of the user's mobile device, this Time to Live value can be at the maximum for a day, or for couple of hours or for that particular session only. After the capability expires, Alice has to restart this capability issuing phase with a new pair of public and secret keys.

2) Service Access Phase: After receiving a capability from the authentication server, in this phase, Alice wants to access a particular service from a service provider. As a result, Alice undergoes the following steps to get service access control from the service provider.

Step B1: Service provider generates a random number R_1 and sends it the user.

Step B2: User signs R_1+1 using SK_U as $Sig_{SK_u}(R_1+1)$. A unique random number is generated by the service provider, every time this phase is initiated by the user. Alice then sends $S_1, Sig_{SK_U}(R_1+1), PK_U, Sig_{SK_{Cap1}}(PK_U)$ to the service provider. Where S_1 is the service ID Alice wants to obtain access control from the service provider.

Step B3: The service provider receives the message B2 from Alice. It first retrieves from its database PK_{Cap1} and verifies the signature of authentication server on $Sig_{SK_{Cap1}}(PK_U)$. If satisfied it proceeds to check if PK_U sent in open equals PK_U in $Sig_{SK_{Cap1}}(PK_U)$.

Step B4: The service provider then verifies the signature on $Sig_{SK_U}(R_1+1)$. Using PK_U sent in open. If equal, the service provider realizes that only the user whose PK_U is signed by the authentication server can only effectively sign R_1+1 using his own SK_U.

Step B5: Thus the service provider without knowing the real identity of the user, concludes that this particular user has the capability $Cap_U = Sig_{SK_{Cap1}}(PK_U)$ issued by the authentication server and hence is authorized to access the service S_1, which is included in the set $\{S_1, S_3, S_6, S_{12}\}$ for $Cap1$. If any one of the above checks fail, Alice is denied access to S_1.

IV. Analysis

A. Security Analysis

The security analysis of our scheme is as follows:

1) User Privacy Protection: In step A4 of capability issuing phase, Alice's capability $Cap_U = Sig_{SK_{Cap1}}(PK_U)$ does not contain her real ID. And also from message B2 in the service access phase, it can be noticed that Alice's real identity (ID) is never sent to the service provider. As a result, the service provider does not know the real identity of Alice. On the other hand the authentication server does not know what services the user has accessed, because the service provider and the authentication server never communicate with each other during the service access phase. The service provider authenticates and authorizes Alice based on her capability Cap_U. This provides complete anonymity and privacy to Alice.

Since the service provider receives PK_U, Alice can still be tracked with her PK_U usage. But even in this situation the real identity of Alice is never revealed, because PK_U acts as a pseudonym for Alice's real ID. And also depending on Alice's mobile device's computational capability and environment's security policy, the Time to Live value of Alice's capability can be at the maximum for a day, or for couple of hours or for that particular session only. After the capability expires, Alice has to restart the capability issuing phase with a new pair of public and secret keys. So the service providers receive different PK_Us/pseudonyms from the same user during different service access phases. The more frequently PK_U and SK_U are changed, the better the anonymity/privacy, but induces high computational overhead on Alice's mobile device. This is a trade-off issue between perfect un-linkable anonymity and performance degradation. However, capabilities issued for only one day are reasonably secure, provide required level of partial un-linkable anonymity/privacy, and at the same time induce less computational overhead on Alice's mobile device.

This consideration especially suits pervasive computing environment, which includes both low and high computing smart devices.

Our scheme also prevents the service provider and the authentication server from maliciously colluding with each other in order to reveal the actions/transactions of the user and expose his privacy. The service provider receives PK_U from Alice via message B3, it can send PK_U to the authentication server with a hope to obtain the real identity ID = Alice. However, at the authentication server's end, there is no match between the real identity (ID) of the user and his/her PK_U. It can be noticed in message A1 of capability request phase, PK_U is blinded as $Blind(PK_U)$ using the blind-signature scheme, as a result the authentication server never knows the value of PK_U. Thus collecting a set of Alice's PK_Us and sending it back to authentication server does not serve any purpose for the malicious service provider to know the real ID of Alice.

2) User Authentication, Authorization and Access Control: Via steps B3 to B5 of the service access phase, it can be noticed that Alice is effectively authenticated, authorized and provided/denied with service access solely based on 1) the verification of authentication server's signature on a capability $Cap_U = Sig_{SK_{Cap1}}(PK_U)$, 2) PK_U sent in open equals PK_U in $Sig_{SK_{Cap1}}(PK_U)$, and 3) verifying Alice's signature: $Sig_{SK_U}(R_1 + 1)$ on $R_1 + 1$ using PK_U. The real identity of Alice user is never used.

In Step B1 of the service access phase, it can be noticed that the service provider sends a unique random number R_1 to Alice's mobile device. Even though Alice's real ID is never included in her capability Cap_U, still the capability can be anonymously linked to Alice, because only the true holder of SK_U can correctly sign on $(R_1 + 1)$ in message B2, which can eventually be verified by the service provider using only PK_U and PK_U is included in the capability $Cap_U = Sig_{SK_{Cap1}}(PK_U)$, which is digitally signed by the authentication server. since the digital signature is unforgeable, no malicious attacker can modify the value of PK_U inside the Alice's capability Cap_U. If an adversary captures message B2: $S_1, Sig_{SK_U}(R_1 + 1), PK_U, Sig_{SK_{Cap1}}(PK_U)$, and tries to impersonate Alice by re-using $Cap_U = Sig_{SK_{Cap1}}(PK_U)$ during a different service access phase, the adversary will not be successful in this impersonation attack. Because during this phase the service provider sends an entirely different unique random number say R_2 and adversary cannot effectively sign on $R_2 + 1$ as he does not know the secret key of Alice SK_U. Thus impersonation attack can be detected and prevented.

3) Replay Attack Detection: If an adversary captures message B2: $S_1, Sig_{SK_U}(R_1 + 1), PK_U, Sig_{SK_{Cap1}}(PK_U)$, and replays it during a different service access phase (in order to illegally access service S_1), the adversary will not be successful in his malicious attempts. Because during this phase the service provider sends an entirely different unique random number say R_3 and the captured message B2, does not contain R_3. Thus replay attack can be detected and prevents adversaries from illegally accessing services.

4) Capability Non-transferability: Our scheme discourages Alice to transfer her capability to another user say "Bob". In step B1 of the service access phase, only Alice who obtained $Cap_U = Sig_{SK_{Cap1}}(PK_U)$ from authentication server can correctly sign on (R_1+1) as $Sig_{SK_U}(R_1+1)$ using her secret key SK_U. As a result, if Alice wants to transfer her capability to Bob, then Alice should give away her secret key to Bob. Alice would not want to give away her secret key to Bob, as Bob may misuse it. This discourages illegal capability transfer.

B. Complexity Analysis

In this section, we analyze the proposed scheme, comparing with the existing approaches in terms of storage requirement and operation complexity:

In a pervasive computing environment, we are mostly interested in storage and computational overhead in user side because they own only small devices such as hand phones, and PDAs. The blind signature scheme is the heart of our proposed scheme. As we mentioned before, the very first secure blind signature scheme is based on RSA and proposed by David Chaum [8]. For our proposed scheme, we recommend 512-bit RSA modulus since normally 512-bit modulus is secure enough for non-critical applications in a pervasive computing environment and also authentication and authorization is a very short and quick process in any transaction demanding short time security. Then, the signature size is 512-bit long. Users also have to generate a public key and secret key pair and elliptic curve cryptography is the most suitable solution as it provides small key size and comparable speed to other public key crypto systems. Commonly, 80-bit key is sufficient in elliptic curve cryptography [27]. So a user has to store totally n*(80 + 80 + 512) = n*672 bits where n is the number of capabilities issued for him. If a user uses common public key and secret key pair for all capabilities (which mean he uses the same public key to interact with different service providers), then the user has to store 80 + 80 + n*512 = 160 + n*512 bits for capabilities.

Regarding computational complexity, we note that, user can generate public key and secret key pair in advance and fetch into its memory. In addition, it will also prefetch a seed value into its memory for generating random value. In capability issuing procedure, according to the blind signature issuing protocol based on RSA [21], user has to perform two modular exponentiation, one modular multiplication and one modular division. In service access procedure, user has to generate a random value and sign that random value using a elliptic curve signature scheme. We refer to [27], [26] for consideration of computational complexity of the two operations.

Regarding performance of the proposed scheme, since in pervasive computing environment, users and probably service providers only own very low computational power devices. Therefore, we are primarily interested in performance of cryptographic algorithms in handheld devices like smartcard and PDA. As we mentioned, blind signature scheme based on RSA and digital signature scheme based on elliptic cure are recommended. The Table 1 gives timing information of

several relevant cryptographic operations in smartcard and Palm Pilot [23], [10], [11]. From this table we can infer that our scheme has medium computational complexity and induces less overhead on user's mobile device.

TABLE I
PERFORMANCE TABLE

	Siemens Crypto Smartcard *		Palm V Dragonball **	
	Perf.	Remark	Perf.	Remark
1024-bit RSA Key Gen.	n/a		15 min.	SSL library
1024-bit RSA Sig. Gen.	230 ms	CRT method	27.8 sec.	SSL library
1024-bit RSA Sig. Verif.	24 ms		1.8 sec.	SSL library
1024-bit Mod. Expo.	n/a		96.91 ms.	512-bit exponent
512-bit Mod. Multi.	n/a		410 ms.	
512-bit Mod. Inverse	n/a		1381 ms.	
EC-DSA Key Gen.	n/a		514 ms.	
EC-DSA Sig. Gen.	185 ms	135-bit sig.	713 ms.	163-bit sig.
EC-DSA Sig. Verif.	360 ms	135-bit sig.	1740 ms.	163-bit sig.
* 5MHz SLE66CX160S, **16.6 MHz				

C. Comparison With Related Work

This section compares our scheme with previously proposed user privacy protection schemes in pervasive computing environments.

1) Traditional Capability-based User Authentication and Authorization: A simple and cheap capability-based user authentication and authorization can be accomplished by using message authentication code (MAC) [15]. More specifically, an authentication server agrees with a service provider on a secret key, say K_{AS}. The authentication server, upon receiving a request from the user to issue him a capability, validates the user and issues the capability in the following form: $< UserID, ServiceID, AccessRightID, Sign_{K_{AS}}(UserID, ServiceID, AccessRightID >$, where UserID represents user identification, ServiceID does service identification and AccessRightID does the type of access to the service. Because MAC provides data integrity and data authentication, only parties aware of the secret key K_{AS} (i.e., authentication server and service provider) can produce and verify such capabilities. In fact, we expect that only the authentication server can issue capabilities. However, by using MAC, the secret key is known to a service provider therefore it can issue capability as well. To avoid such drawback, we can use digital signature instead of MAC. Because digital signature employs the public key cryptography, then the only information available for service providers (as well as other parties) is the authentication server's public key, say PK_{AS}, and a service provider can solely verify the validity of issued capabilities only when it knows PK_{AS}. A capability now has the following form: $< UserID, ServiceID, AccessRightID, Sign_{SK_{AS}}(UserID, ServiceID, AccessRightID >$, where SK_{AS} a secret key of the authentication server. The unforgeability of capability is straightforward since digital signature is unforgeable.

Note that we intend to protect user privacy as well. Above approaches cannot support user privacy since UserID is included in capability to prevent transferability, impersonation and misuse of capability. Even if we do not include UserID in capability, there is possibility that service provider and authentication server collude to discover user activity since service provider knows which service an user access while the authentication server knows the real identification of that user (to validate the user). Since each issued digital signature is likely unique due to the randomized nature of digital signature issuing procedure. Therefore, the authentication server can make a table matching real user identification and corresponding signature issued for each user. Later on, the service provider can forward capability which contains the digital signature to the authentication server. Thereby the two colluded parties can discover who actually used which services.

As explained in the above security analysis section, in our scheme, even though user's real ID is never included in his capability, the capability can still be anonymously linked to user, because only the user can correctly sign on a random number (sent by the authentication server) using his secret key, which can eventually be verified by the service provider using user's public-key included in his capability. This provides complete privacy to the user. Also the scheme prevents service providers from maliciously colluding with the authentication server in order to reveal the true identity of the user and expose his privacy.

2) Identity Management Approach: Changing pseudonym every time when using services is a way to protect the real identity of users. [13] used a similar method to define multiple identities for a user, and the user uses them depending on situations. Hence user privacy is only revealed upon users'decisions. The identity management approach in [13] has presented a general framework without a concrete architecture, protocol flows, and mechanisms to generate virtual identities (VIDs). In addition, the user has to choose carefully, towards which party he uses which VID and when he has to change this VID, on too much disclosed information in the VID's context. Handling VIDs in this way is certainly not user friendly as it involves a great deal of user presetting and user intrusion to resolve a situation not considered in the presetting. It also creates a burden on the user's mobile device to decide and choose the appropriate VID depending on the interacting service. Moreover, the linkable problem among VIDs must be considered,otherwise the real identity of a user may be revealed. Also this approach does not consider the user authorization aspect. [12] describes further drawbacks of this approach.

The major drawback of this scheme is the management of pseudonyms and high degree of user involvement. Our scheme provides user anonymity, authentication and authorization with out using pseudonyms. User just needs to store his secret and public-key until the capability issued by the authentication server (on a daily, hourly, session basis) expires and hence there is no overhead in managing different identities by the user's mobile device. User's involvement is also very low. Our scheme is based on capability approach and it provides user authorization facility to the service providers.

3) Pseudonym Systems: When considering authentication and privacy issues at the same time, David Chaum proposed "pseudonym systems" [9]. In a pseudonym system, a user interacts with multiple organizations in an anonymous manner using so called pseudonyms (different pseudonym to interact with different organization). These pseudonyms are unlinkable to prevent two organizations from combining their database to discover user's activities. Pseudonym system provides a way to user to get credential from one organization and demonstrate to other ones (authentication) and that is why it is also called an anonymous credential system. One of famous example of the anonymous credential system is idemix [5], which is based on the protocols proposed in [4]. However, in current anonymous credential systems like [9], [4], while employing high computational complexity for number-theoretic operations, the computational complexity at the user's end is very high. This fact is not appropriate for pervasive computing environment where users own computing devices with limited computational power. Furthermore, it is required that every service provider maintain its own user database (remember that a user uses different pseudonyms to interact with different organizations).

Our scheme is simple and easy to implement. As mentioned in the above "complexity analysis" section, our scheme induces less computational burden on the user's mobile device.

4) Mix-Network: Other notion proposed by David Chaum to protect user privacy is "mix net" [7]. A mix net consists of several servers, called mixes. Each server receives a batch of input messages and produces a batch of output in a permuted (mixed) order. An observer cannot know how the output messages of a mix net correspond to input messages. Mix net is also used in electronic voting to provide voter privacy.

Mist [1], [2], [29] concentrated on communication privacy by building an overlay network with Mist Routers. Routing through Mist Routers protects authorized users' location privacy. But users have to trust a "Lighthouse". The Lighthouse keeps all information of users registered with it. If the Lighthouse is not honest, the user's privacy will be exposed. In addition, performance aspect is always a limitation of the systems, which utilize Mix network style as it involves computationally intensive procedures. It also assumes high degree of trust in the mix network. As a result Installing and maintaining trusted mix networks in a pervasive computing environment is very difficult and expensive.

Our scheme is very simple, easy to implement. The computational complexity at the user's end is is also very low as described in the "complexity analysis" section. In our scheme the authentication server can never know the actions/transactions of the user.

V. Conclusion

Our scheme can be easily ported on to a public space or large scale pervasive computing environments, for example airports, train stations, streets, highways, *etc*. Nowadays, in our society we can experience the ubiquitous presence of services being offered by credit card companies (VISA, MASTER, *etc*) and mobile operators (AT&T, BT, Vodafone, KT, *etc*). Consumers and service providers have already put in a great deal of trust in such big organizations. Therefore these organizations can take up the role of authentication servers mentioned in our scheme and issue capabilities to their esteemed customers or subscribers. In pervasive computing environments, such organizations may want their customers, silver/gold/platinum card holders and VIP members to access different types of services depending on their privileges.

Our proposed scheme is first of its kind to introduce capability based privacy preserving user authentication and authorization scheme for pervasive computing environments. The scheme is very simple, efficient and cost effective with respect to storage and computation complexity. It provides complete privacy and anonymity to the user. The service providers authenticate and authorize the users based on the anonymous information submitted by them. The service provider does not know the user's real identity and the authentication server does not know what services the user is accessing. Our scheme also achieves capability non-transferability.

Our further work includes providing capability revocation feature. This option would terminate a capability once issued to a user, who is later caught indulging in malicious acts in the pervasive computing environment.

References

[1] J. Al-Muhtadi, R. Campbell, A. Kapadia, D. Mickunas, and S. Yi, "Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments", *International Conference of Distributed Computing Systems (ICDCS 2002)*, Vienna,Austria, 2002.

[2] 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.

[3] J. Al-Muhtadi, A. Ranganathan, R. H. Campbell, M. D. Mickunas, "Cerberus: A Context-Aware Security Scheme for Smart Spaces", *PerCom 2003*.

[4] J. Camenisch and A. Lysyanskaya, "Effcient Non-transferable Anonymous Multi-show Credential System with Optional Anonymity Revocation", *Advances in Cryptology, EUROCRYPT 2001*, LNCS 2045, pp. 93-118.

[5] J. Camenisch and E. Herreweghen, "Design and Implementation of the Idemix Anonymous Credential System", *9th ACM Conference on Computer and Communications Security, ACM*, p. 21-30 in 2002.

[6] R. H. Campbell, J. Al-Muhtadi, P. Naldurg, G. Sampemane, M. D. Mickunas, "Towards Security and Privacy for Pervasive Computing", *ISSS 2002*, p.1-15.

[7] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms", *Communications of the ACM*, 24(2):84-88, 1981.

[8] D. Chaum, "Blind Signatures for Untraceable Payments", *Advances in Cryptology*, Proceedings of Crypto 82, Plenum, pp. 199-203.

[9] D. Chaum "Security without identification: transaction systems to make Big Brother obsolete", *Communications of the ACM*, 28(10), 1985.

[10] N. Daswani, “Cryptographic Execution Time for WTLS Handshakes on Palm OS Devices”, http://www-db.stanford.edu/~daswani/papers/WTLSPerformancePaper3.pdf
[11] H. Handschuh and P. Paillier, “Smart Card Crypto-Coprocessors for Public-Key Cryptography”, “Smart Card Research and Applications”, Springer-Verlag, 2000, LNCS vol. 1820, pp 386 - 394.
[12] C. Hauser, “Privacy and Security in Location-Based Systems With Spatial Models”, *PAMPAS’02*.
[13] U. Jendricke, M. Kreutzer, and A. Zugenmaier, “Pervasive Privacy with Identity Management”, *UBICOMP ’02*.
[14] M. Langheinrich, “A Privacy Awareness System for Ubiquitous Computing Environments”, *UbiComp 2002*,Springer-Verlag, LNCS 2498, p.237-245, 2002.
[15] A.J. Menezes, P.C. vaz Oorschot, and S.A. Vanstone, “Handbook of Applied Cryptography”, ISBN 0-8493-8523-7, 1997 CRC Press LLC.
[16] A.J. Menezes, P.C. vaz Oorschot, and S.A. Vanstone, “Handbook of Applied Cryptography”, Chapter 11: Digital Signatures, ISBN 0-8493-8523-7, 1997 CRC Press LLC.
[17] G. Myles, A. Friday, and N. Davies, “Preserving Privacy in Environments with Location-Based Applications”, *IEEE Pervasive Computing journal - Security and Privacy*, Jan-Mar 2003.
[18] M. Reed, P. Syverson, and D. Goldschag, “Anonymous Connections and Onion Routing”, IEEE J. Selected Areas in Commun, Vol. 16, No. 4, May 1998, pp.482-494
[19] M. Reiter and A. Rubin, “Crowds: Anonymity for Web Trasactions”, ACM Transactions on Information System Security, Vol. 1, No. 1, November 1998, pp. 66-92
[20] M. Satyanarayanan, “Pervasive Computing: Vision and Challenges”, *IEEE Personal Communications*,August, 2001.
[21] H.T. Tavani, and J.H. Moor, “Privacy Protection, Control of Information, and Privacy-Enhancing Technologies”, *ACM SIGCAS Newsletter ’01*.
[22] M. Weiser, “The Computer for the 21st Century”, *Scientific American*,Sep, 1991.
[23] D. S. Wong, H. H. Fuentes and A. H. Chan, “The Performance Measurement of Cryptographic Primitives on Palm Devices” - *17th Annual Computer Security Applications Conference*, New Orleans, Louisiana, 2001.
[24] M. Wu, A. Friday, “Integrating Privacy Enhancing Services in Ubiquitous Computing Environments”, *Workshop on Security in Ubiquitous Computing, 4th International UBICOMP, 2002.*
[25] Anonymizer.com, http://www.anonymizer.com
[26] “Computer Security Resource Center (CSRC), NIST”, http://csrc.nist.gov/csrc/fedstandards.html.
[27] “Elliptic Curve Cryptography”, http://csrc.nist.gov/CryptoToolkit/dss/ecdsa/NISTReCur.pdf.
[28] “Easy Living”, Microsoft Research, http://research.microsoft.com/easyliving/.
[29] “GAIA - Active Spaces for Ubiquitous Computing”, University of Illinois at Urbana-Champaign, http://choices.cs.uiuc.edu/gaia/.
[30] “IEEE 802.15 Working Group for WPAN”, http://grouper.ieee.org/groups/802/15/.
[31] “Location Privacy Proteaction Act and other privacy related law”, http://www.techlawjournal.com/cong107/Privacy.
[32] “MIT Project Oxygen”, http://oxygen.lcs.mit.edu/.
[33] National Institute of Standards and Technology (NIST),“Pervasive Computing SmartSpace Laboratory”, http://www.nist.gov/smartspace/.
[34] “The Aware Home”, Georgia Institute of Technology, http://www.cc.gatech.edu/fce/ahri/.