A Lightweight, Privacy Preserving and Secure Service Discovery Protocol in Ubiquitous Computing Environment

Jangseong Kim and Kwangjo Kim

School of Engineering, Information and Communications University
{withkals,kkj}@icu.ac.kr

Abstract. During service discovery, preserving privacy of end users and service providers is one of the challenging research issues in ubiquitous computing environment (UCE). To solve this issue, we define service group and classify it into two cases: public service and private service. End users' privacy is important in public service group while privacy of end users and service providers are important in private service group. Based on this observation, we propose a lightweight, privacy preserving and secure service discovery protocol for UCE. Our protocol needs less computation overhead from the view of user while providing mutual authentication and entity anonymity. In addition, it provides accountability, improves non-linkability, enhances security level by sharing a selected number set, and does not rely on underlying system infrastructure. Differentiated service access control is also feasible by arranging users in different service groups. Since our proposed protocol is designed for providing security services in middleware, the protocol can be used to build a security framework adapting its security service to end users' circumstances while preserving a certain security level.

1 Introduction

In ubiquitous computing environment (UCE) hundreds of devices and services may surround end users. The end users may not find proper service without any prior knowledge about near environment. In this point of view, the service discovery protocol (SDP) is essential to access proper services in UCE. Since most SDPs do not consider privacy and security issues [5], the end users may experience illegal tracking or leakage of their private information while they can get proper access information via SDP. Also, we need to consider privacy for service providers since an end user may be a service provider, for instance, P2P image file sharing and personal music broadcasting. Hence, we should protect not only privacy of the service providers but also privacy of the end users; it means that sensitive information for SP (*e.g.*, service access information, owner's identifier, and presence) should be exposed to legitimate users only and sensitive information for end users (*e.g.*, their identifiers, presence information, and service query information) should be exposed only if necessary.

A typical approach to privacy protection is providing anonymity based on the blind signature scheme. Double spending problem of an authorized credential [9] may happen without verification on the actual holder of the authorized credential. A malicious user can use an inaccessible service using the previous credential. Therefore, we should consider accountability for an authorized credential by limiting usage times.

Energy management is also a big challenge in UCE. The end users want to continue accessing the target service even if their surrounding environment has been changed. It indicates that an alternative service should be provided although the service is not accessible. Proactivity and self-tuning are introduced to support this idea but it causes more energy consumption.Thus, we should consider a lightweight cryptographic protocol for reducing energy demand, while preserving a certain security level.

In this paper, we propose a novel protocol for a lightweight, privacy preserving and secure service discovery for UCE. While providing mutual authentication and entity anonymity, our protocol needs less computation overhead on users' side. In addition, it provides accountability, improves non-linkability, enhances security level by sharing a selected number set between end users and authentication servers, and does not rely on underlying system infrastructure. Differentiated service access control is also feasible by arranging users in different service groups. Since our proposed protocol is designed for providing various security services in the middleware by extending privacy preserving authentication for UCE, this approach is useful for establishing a security framework for UCE. Through secure SDP, the framework adapts its security service to end users' circumstances while preserving a certain security level.

The rest of this paper is organized as follows: In Section 2 we review related work. We present our proposed scheme in Section 3. Then we the discuss security features of the proposed scheme including the performance analysis in Section 4. Finally we conclude the paper in Section 5.

2 Related work

Although several SDPs [2–4, 6] are proposed, these protocols may be roughly classified into two models: client-server and client-server-directory. In the client-server model, clients send a query for searching a proper service and some servers reply to the query only if they provide the service. If clients receive multiple responses, they select one service and contact. Whenever clients send a query, servers should check whether they are providing the target service or not. Since we believe that thousands of computing devices and services are available in ubiquitous computing environment, the client-server model is not a proper choice. It is useful only if a user's communication relies on peer-to-peer architecture. To support thousands of computing devices and services, the client-server-directory model is introduced. Note that directories or directory servers may be used to store service information (*e.g.*, service type, service identifier, service description, network address). In this model clients and servers communicate with directories for receiving proper service information and registering. Therefore, the model can deal with thousands of devices and services in UCE by increasing directories without changing the existing systems. In addition, directory-based approach can simplify trust management since the directory is the only entity which clients need to identify. However, it has some limitations in the security aspect. Since service providers should register their services in the directory, the service information can be exposed to the directory. Also, service providers may not control who should get the service information. The first problem can be solved by providing anonymity for service providers. The second problem can be solved by performing the membership test on the service provider side. In the rest of the paper, we will consider the directory-based approach only.

Most SDPs, however, do not provide security and protect entities' privacy [5]. Therefore, any services may be discovered and used by any user via SDPs. To solve privacy and security issues S. E. Czerwinski *et al.* proposed the Secure Service Discovery Service (SSDS) [7]. It has many built-in security features including authentication, authorization, data and service privacy, and integrity. The proposed architecture consists of three major entities: clients or end users, directories or service discovery servers, and service providers. End users and service providers authenticate with directories for lookup and announcement. Note that directories are regarded as trusted entities. However, they should expose private information such as own identities and service access information whenever end users perform lookup operation and service providers register own services.

In 2006 F. Zhu *et al.* suggested the PrudentExposure model for secure SDP [8]. Legitimate end users can discover and access service easily through binding process. Since end users should bind themselves to the target agents and transfer all their identities to the agent via the secure channel, their devices don't require supporting various authentication methods. However, it causes additional communication cost. In addition, privacy leakage may occur among insiders although the model is designed to preserve sensitive information for end users and service providers. For accessing the target service end users only need to suggest their domain ID and authenticate them with the directory; after authentication process end users can access other services. Moreover, end users should perform one public key operation whenever they send a lookup message.

Recently, J. Kim *et al.* proposed the Lightweight Privacy Preserving Authentication and Access Control scheme (LPPA) for UCE [10]. While providing mutual authentication between end users and service providers, the scheme can preserve users' privacy with an enhanced security level. During n service access requests, end users need only one public key operation. Also, it improves non-linkability [11]. Even if an end user generates two sequential credentials, insiders (*i.e.*, the accessed service provider) and outsiders cannot link two different sessions to the same user. But, the authors assumes that all users should know public key of the target service provider and service identifier indicating their access privilege before registering their own credential to the authentication server. Specially, it is not true when an end user should select alternative services due to users' mobility.

Gruteser and Grunwald [12] offered a method for hiding a user' MAC address with an anonymous ID so that the user cannot be tracked in a wireless LAN environment.

3 Our proposed scheme

In this paper, we assume that an end user can control the source addresses of the outgoing Medium Access Control (MAC) frames since it is a prerequisite for anonymous communications. Gruteser *et al.* [12] covered one of the detailed methods for this kind of modification and it is out of scope of this paper. Additionally, we assume that $PubK_{AS}$ and ID_{AS} are known to all entities. Also, a service provider (SP) defines the scope and the meaning of SID, associates each user with a particular service type, assigns a unique public key to each service type and provides this information to the authentication server (AS) for further enforcement of authorization rules. Note that generating a unique public key per each SID is done by the AS and the public key list is sent to the SP. For example, the SP wants to provide three different services: guest, light and premium. The SP requests the AS to issue three public keys corresponding to guest, light and premium. Then the AS issues and sends the public key list only if the SP has a proper privilege. Moreover, the end user receives SID, its corresponding public key ($PubK_{SID}$), and MT from the target service provider and determines n based on their service access frequency.

To preserve privacy of end users and service providers, we introduce a service group consisting of one service provider and its subscribers. We classify the service group into public service group and private service group whether the service information should be protected or not. In public service group we should preserve privacy of users. In private service group, however, we should protect privacy of users and service providers. Note that this classification assumes that privacy of end users should be protected in any case except they want to expose their private information. Table 1 illustrates the notation used in this paper.

Our proposed protocol consists of four phases: entity registration, service registration, discovery and service access. End users and service providers register their credentials in the authentication server in the entity registration phase. In service registration phase, service providers register their services in a close directory with their own credential information. Then the directory stores the received service information only if the received credential is authorized by the authentication server. During the discovery phase, end users send a query message to get proper service information including SID and $PubK_{SID}$. After the discovery phase, end users can use the authorized credential for their authentication in n sessions. Using the credential end users can authenticate themselves to the service provider protecting their privacy information. Note that the authentication server is a trusted third party. Figure 1 illustrates our system architecture and these activities.

SID	A service type identifier is identified by a unique public key and it describes a selected subset of the available service pool that can be accessed by a mobile user
SG_{Pub}	A public service group consisting of one SP and its subscribers
SG_{Pri}	A private service group consisting of one SP and its subscribers
MT	A ticket for indicating subscribers of the target SP
K_{AB}	Shared secret key between entities A and B
$\{m\}_{K_A}$	A message m is encrypted by K_A
$\{m\}_{PriK_A}$	A message m is signed by private key of entity A
$H(m)$	Hash message m
n	A user's access frequency
S	A selected number set and its length should be larger than $2n$
ID_A	An identifier of entity A
$C^i, i = 0, 1, \cdots$	A series of authorized credentials
$j^i, i = 1, 2, \cdots$	A series of a user's number selections
$R^i_A, i = 1, 2, \cdots$	A series of nonce generated by entity A and it is usually a 64-bit pseudo random number.
$CertA$	A certificate which binds entity A with A's public key $PubK$
$Credential$	A ticket for authentication
$Anchor$	An initial credential C^0

Table 1. Notation

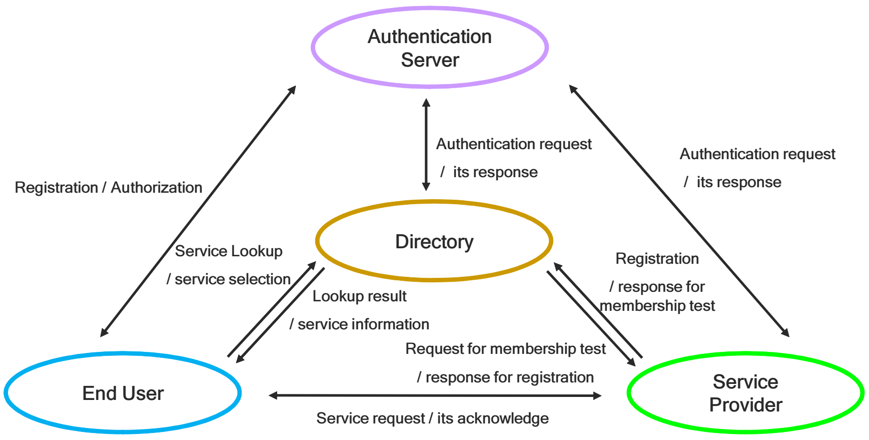
Authentication Server
Authentication request / its response
Authentication request / its response
Registration / Authorization
Directory
Service Lookup / service selection
Lookup result / service information
Registration / response for membership test
Request for membership test / response for registration
End User
Service Provider
Service request / its acknowledge

Fig. 1. System architecture

3.1 Entity registration phase

During the entity registration phase, the end user or SP issue their credential information and receive authorized credential from the AS only if they are legal entities with proper access permission. These activities are illustrated in Figure 2. While the SP receives authorized credential for service registration, the end user receive authorized credential for lookup request or service access. If the entity in Figure 2 is SP, then SID indicates a privilege for providing the target service. If the entity is end user without knowing $PubK_{SID}$, then SID shows a privilege for lookup request. Otherwise, SID indicates a privilege for accessing the target service. After this phase, the SP or the end user can learn about the list of trusted directories.

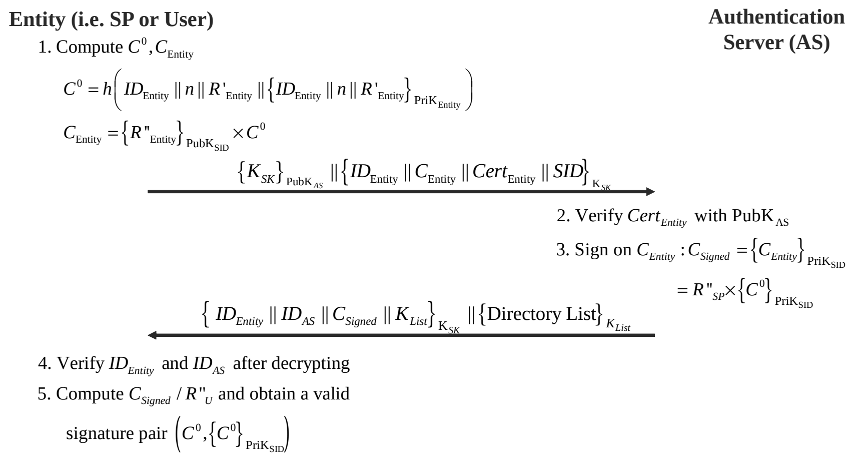
Entity (i.e. SP or User)
Authentication Server (AS)
1. Compute C^0, C_{Entity}
$C^0 = h\left(ID_{\text{Entity}} \| n \| R'_{\text{Entity}} \| \{ID_{\text{Entity}} \| n \| R'_{\text{Entity}}\}_{\text{PriK}_{\text{Entity}}} \right)$
$C_{\text{Entity}} = \{R''_{\text{Entity}}\}_{\text{PubK}_{\text{SID}}} \times C^0$
$\{K_{SK}\}_{\text{PubK}_{AS}} \| \{ID_{\text{Entity}} \| C_{\text{Entity}} \| Cert_{\text{Entity}} \| SID\}_{K_{SK}}$
2. Verify $Cert_{Entity}$ with PubK_{AS}
3. Sign on $C_{Entity} : C_{Signed} = \{C_{Entity}\}_{\text{PriK}_{\text{SID}}}$
$= R''_{SP} \times \{C^0\}_{\text{PriK}_{\text{SID}}}$
$\{ID_{Entity} \| ID_{AS} \| C_{Signed} \| K_{List}\}_{K_{SK}} \| \{\text{Directory List}\}_{K_{List}}$
4. Verify ID_{Entity} and ID_{AS} after decrypting
5. Compute C_{Signed} / R''_U and obtain a valid
signature pair $\left(C^0, \{C^0\}_{\text{PriK}_{\text{SID}}}\right)$

Fig. 2. Entity registration

3.2 Service registration phase

During the service registration phase, the SP registers his/her service access information in a nearby directory by submitting own credential. Then the directory requests the AS to verify whether the received credential is valid or not. This verification method is the same as the verification in the authentication procedure during discovery phase Only if the credential is valid, the directory sends a response including a shared fresh session key. Otherwise, the SP sends the request message for service registration again. Figure 3 illustrates this procedure. Note that C^i is $h(C^0||j^i||R^i)$ and $K_{SP,AS}$ is a shared secret key between SP and AS. It is computed as:

$$K_{SP,AS} = \begin{cases} h(C^0||PubK_{AS}||R^1||j^1||SID) & \text{if } i = 1 \\ h(C^0||C^{i-1}||SID) & \text{otherwise} \end{cases}$$

Since verification for the received credential is done by the AS, SP can hide any relationship between his/her identity and service access information. Also, the SP needs only one public key operation during n service registrations. Using

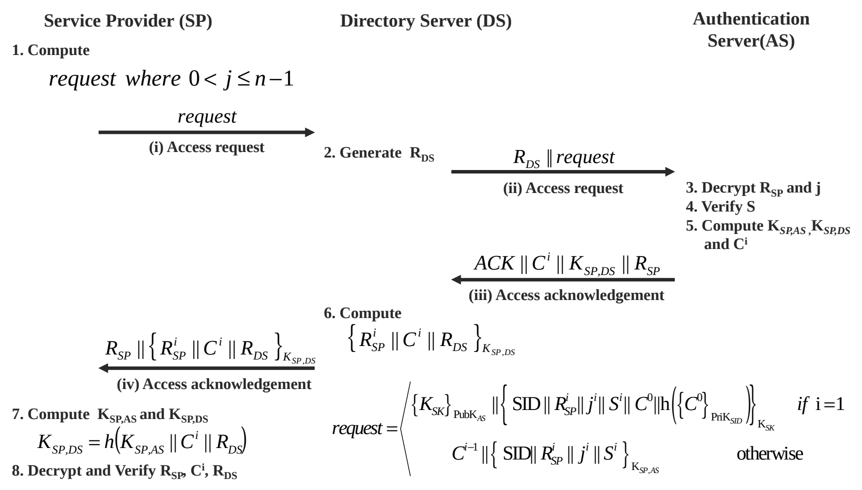
Service Provider (SP)
Directory Server (DS)
Authentication Server(AS)
1. Compute
$request\ where\ 0 < j \le n-1$
$request$
(i) Access request
2. Generate R_{DS}
$R_{DS} \| request$
(ii) Access request
3. Decrypt R_{SP} and j
4. Verify S
5. Compute $K_{SP,AS}$, $K_{SP,DS}$ and C^i
$ACK \| C^i \| K_{SP,DS} \| R_{SP}$
(iii) Access acknowledgement
6. Compute
$\{R_{SP}^i \| C^i \| R_{DS}\}_{K_{SP,DS}}$
$R_{SP} \| \{R_{SP}^i \| C^i \| R_{DS}\}_{K_{SP,DS}}$
(iv) Access acknowledgement
7. Compute $K_{SP,AS}$ and $K_{SP,DS}$
$K_{SP,DS} = h(K_{SP,AS} \| C^i \| R_{DS})$
8. Decrypt and Verify R_{SP}, C^i, R_{DS}
$request = \begin{cases} \{K_{SK}\}_{\mathrm{PubK}_{AS}} \| \{\mathrm{SID} \| R_{SP}^i \| j^i \| S^i \| C^0 \| \mathrm{h}(\{C^0\}_{\mathrm{PriK}_{SID}})\}_{\mathrm{K}_{SK}} & if\ \mathrm{i}=1 \\ C^{i-1} \| \{\mathrm{SID} \| R_{SP}^i \| j^i \| S^i\}_{\mathrm{K}_{SP,AS}} & \text{otherwise} \end{cases}$

Fig. 3. Authentication in service registration

the shared session key the SP registers his/her service access information (*e.g.*, service type, service identifier, service description, SID list, public key list, network address)in the directory. Note that communication between the directory and the authentication server is secure since it is easily achieved by sharing a key or performing the registration phase.

3.3 Discovery phase

After the entity registration phase, the end user obtains authorized credential implying the user can send n lookup requests to the directory. The end user in SG_{Pub} receives a possible service list during the discovery phase. From the service list she selects one service and obtains service access information. However, the end user in SG_{Pri} should send a service discovery query with service access privilege. Next, the directory forwards the received message to the service provider for a service membership test. Only if the end user is one of the legitimate subscribers, then the directory forwards service access information. The detailed procedure is as follows:

Service lookup : There are two methods for service lookup. One is to find an accessible service list with SID. From the service list, the end user may select one service or retry. The other is to find a specific service with SID and *query*. While the first method can be used in SG_{Pub}, the second method can be used in both SG_{Pub} and SG_{Pri}.

The end user generates a lookup message and sends it to the directory. If the user wants to find an accessible service list, the lookup message is *request*. Otherwise, the message is $request||query$. If the lookup is first request message, then $request$ message is $\{K_{SK}\}_{PubK_{AS}}||\{SID||DSID||R^1||j^1||C^0||h(\{C^0\}_{PriK_{DSID}})\}_{K_{SK}}$. Otherwise, the message is $C^{i-1}||\{SID||DSID||R^i||j^i||S^i\}_{K_{U,AS}}$. Note that $DSID$ shows a privilege for lookup request, i indicates the number of request, $query$ is $R'||\{UID||R'\ ||AUTH_{NUM}\}_{K_{token}}$ and K_{token} is $h(UID||R'||MT||SID)$. Also, UID is the user ID in the service and $AUTH_{NUM}$ is the authentication number issued when the user becomes a subscriber. After one public key operation in the first request, the end user and the authentication server can share a fresh session key. We use service type finding a proper service in service discovery since it is simplest. A more sophisticated way to build query message is out of scope of this paper.

Authentication : The directory (DS) forwards the received message with nonce, R_{DS}, to the AS. Then the AS decrypts and verifies whether the requestor has an authorized credential or not. If it is first request, then the AS signs C^0 with $PriK_{SID}$ and compares the result with the received signature. Only if the result is the same, the AS computes C^1 and $K_{U,AS}$. Also the AS stores SID, S^1, C^0 and C^1. Otherwise, the AS verifies that the j-th value of the stored S^{i-1} is zero and the stored S^{i-1} is the same as S^i except the j-th value. Only if the verification result is true, the AS computes C^i and $K_{U,AS}$, S^i by flipping j-th value of S^{i-1} and stores the results since the AS believes that the requestor has legitimacy of the requested service. Otherwise, the AS discards the message. If there are several verification failures on series of the authorized credential, the AS can request the end user to change his/her credential to notify that there is an impersonation attack.

$$K_{U,AS} = \begin{cases} h(C^0||PubK_{AS}||R^1||j^1||SID) & \text{if } i = 1 \\ h(C^0||C^{i-1}||SID) & \text{otherwise} \end{cases}$$

After verification, the AS computes $K_{U,DS}$ as $h(K_{U,AS}||C^i||R_{DS})$ and sends the acknowledge message to the directory with C^i, SID, $K_{U,DS}$ and R_U. Now, there are two cases:

1. ***Public service group***: the directory computes response message, $response$, and sends it to the end user. After computing $K_{U,AS}$ and $K_{U,DS}$, the end user decrypts the received message and verifies the received R^i_U, C^i, and R_{DS}. Only if all the verification results are correct, then the end user continues. Otherwise, the user sends lookup request again. Note that $response$ is $R_{DS}||\{R^i_U||C^i||R_{DS}||ServiceList\}_{K_{U,DS}}$. Also, $ServiceList$ indicates the service list matching the submitted SID.
2. ***Private service group***: the directory computes a request message for membership test, $R_{DS}||\{R_{DS}\ ||query\}_{K_{SP,DS}}$, and sends it to the service provider. Next, the service provider decrypts the message and verifies R_{DS}. Only if the verification result is correct, then the service provider performs membership test using the received $query$. Based on the result of membership test, the service provider computes response message, $\{ACK||R_{DS}||R_{SP}\}_{K_{SP,DS}}$, and

sends it to the directory. Then the directory computes *response*, $R_{DS}||\{R_{DS}$ $||C^i||R_U||INFO_{SP}\}_{K_{U,DS}}$, and sends it to the user only if the user is a subscriber of the service. After computing $K_{U,AS}$ and $K_{U,DS}$, the end user decrypts the received message and verifies the received R^i, C^i, and R_{DS}. Only if all the verification results are correct, then the end user continues. Otherwise, the user sends another lookup request. Note that $INFO_{SP}$ is the information how to access the selected service.

After authentication in the discovery phase, the end user can access the target service. If the purpose of discovery is finding an alternative service to handle the user's mobility, the user may not have an authorized credential to access the service. Then, the user should perform the entity registration phase. If the user knows his/her $PubK_{SID}$ and access information without the authorized credential, the user only needs to perform the entity registration step without the discovery phase. Note that end user and authentication server share a fresh session key, $K_{U,AS}$ through this step. Also end user and directory share a fresh session key, $K_{U,DS}$.

Service selection : The end user selects one service among the received service list and notifies his/her selection. Then, the directory forwards $INFO_{SP}$ to the user. Note that the communications between both entities are protected by the shared session key. This step will be used only if the end user generates a lookup request without *query*.

3.4 Service access phase

To access the target service an end user sends a service request to the service provider. Next, the service provider forwards the request to the authentication server. After user authentication the authentication server replies the authentication response to the SP. Using the response the SP determines whether SP should provide his/her service or not. The detailed procedure is similar to the authentication in the service registration phase. The main difference is that the initiator of the authentication process is not the service provider but the end user. Therefore, the term 'SP' and 'DS' in Figure 3 are replaced with end user (U) and SP, respectively. It means that $K_{SP,AS}$ and $K_{SP,DS}$ are replaced with $K_{U,AS}$ and $K_{U,SP}$, respectively.

4 Analysis of our proposed scheme

In this section we analyze the performance and security related-features of our proposed scheme.

4.1 Performance

Storage overhead : End user should store two 5-turples (C^0, R^i, j^i, n, S) for lookup request and service access request. Service provider needs to save one 5-turple (C^0, R^i, j^i, n, S) for service registration.

Computation overhead : Before comparing computation overheads between ours and the protocol in [8], it is required to describe the differences. The protocol in [8] does not have the AS and the end user delegates his/her sensitive information to the user agent. Then the user agent performs authentication tasks instead of the end user. As it is a similar concept to the middleware approach, we consider the end user and its user agent as same entity. Table 2 shows a comparison of the computation overheads between ours with SG_{Pri} and [8]. Our proposed protocol only needs three public key operations during n times discovery phase while the protocol in [8] requires n public key operations. Note that in Table 2 the communication entity (*i.e.*, User, DS, SP and AS) needs online computation if there is no the term "off-line".

		# of Pub. Key	Sig.Oper.	Nonce Gen.	Hash Oper.	# of Sym. Key
	User	1	0	3	3	5
[8]	DS	1	3	1	3	5
	AS	0	0	0	0	0
	SP	0	0	0	0	1
	User	$1/n$(off-line)+$2/n$	$1/n$	3	6	2(off-line)+2
Ours	DS	0	0	1	0	2
	AS	$2/n$	$2/n$	0	3	2
	SP	0	0	0	1	2

Table 2. Computation overheads comparison

4.2 Security

Mutual authentication : In the proposed protocol, the end user authenticates himself/herself to the AS using his/her own authorized credential, so that the AS knows that the user is legal and authorized. The AS also authenticates itself to the user through its public key and by showing its knowledge of the corresponding private key.

User context privacy : Our proposed protocol protects the end user's context privacy against insiders and outsiders. Note that all communication channels are well protected. The SP and DS cannot imagine who sends the service access request and lookup request.

Non-linkability : Non-linkability means that, for insiders(*i.e.*, SP) and outsiders, 1) neither of them could ascribe any session to a particular end user, and 2) neither of them could link two different sessions to the same user [11]. In the proposed protocol non-linkability is achieved with respect to both insiders and outsiders. Firstly, the authorized credential combined with the fresh nonce is never transmitted in the plaintext form. Hence, the outsiders cannot associate a session with a particular user and ascribe two sessions to the same user. Secondly, the end user derives all the authorized credentials from the anchor value, and it is only known to the authentication server and the end user. Even if the SP and DS can obtain all the authorized credentials except the anchor value, they cannot link two different sessions to the same user. Moreover, the authorized credential combined with the fresh nonce is never transmitted in the plaintext form. Therefore, the insiders cannot associate a session with a particular user. Note that the AS is regarded as a trusted third party.

Accountability and non-transferability equivalency : In the proposed protocol the credentials are authorized only when the end user is explicitly authenticated. By adopting a select number set, the proposed protocol can provide one-time usage of the authorized credentials. Hence, it can prevent double spending problems. By incorporating an accounting function the proposed protocol can provide good accounting capability. Furthermore, the proposed scheme provides equivalent to non-transferability from the service point of view. Because the credentials are delegated among users, no harm is done to the SP in the sense that the authorized user is responsible for all the received services via his/her own credentials. This property greatly reduces the problem of service abuse about which the SPs are worried.

Data confidentiality and integrity : All communications except between DS and AS are protected by the shared session key or the receiver's public key. Hence, data confidentiality and integrity can be easily achieved using symmetric cryptography.

Differentiated service access control : Our proposed protocol can provide differentiated service access control by classifying users into different service types. Different users are authorized based on the service types to which they belong. Hence, "User authorization" is accomplished in a different way. Moreover, it is possible to combine usage of the different credentials for high-level differentiated service access control. But, it is beyond the scope of this paper.

Enhanced security level : Every access request message contains S proving the actual holder of the message since it is randomly generated by the end user and is only delivered to the AS. To impersonate the target user, the adversary is required to present S even if the adversary knows the user's anchor value. Therefore, the proposed scheme enhances the security level.

No additional key management : Two entities among end user, SP, DS, and AS generate a shared symmetric key that is used only once (*e.g.*,$K_{U,AS}$, $K_{U,DS}$, $K_{U,SP}$, $K_{SP,AS}$, $K_{SP,DS}$). Also, the shared key can derive from the stored 5-turple (C^0, R^i, j^i, n, S). Thus, there is no additional key management overhead by replacing the reduced public key operations with the symmetric key operations.

5 Conclusion

In this paper, we have proposed a lightweight, privacy preserving and secure service discovery protocol for UCE. Our proposed protocol provides mutual authentication while preserving privacy. Additionally, the proposed protocol provides non-linkability, security margin, accountability and differentiated access control. While providing these novel features, the proposed protocol needs less public key operations than the others. In the near future, we want to extend our proposed protocol in the multiple administration domain. End users in UCE can temporarily visit the other administration domains. Issuing and registering them in the domains may cause unnecessary management cost.

References

1. M. Satyanarayanan, "Pervasive computing: Vision and Challenges", IEEE Personal Communications, Aug., vol. 8. no. 4, pp.10-17, 2001
2. C. Ellison, "Home Network Security", Intel Technology J., vol. 6, pp. 37-48, 2002.
3. Sun Microsystems, "Jini Technology Core Platform Specification", Version 2.0, `http://www.sun.com/software/jini/specs/`, 2003.
4. C. Lee and S. Helal, "Protocols for Service Discovery in Dynamic and Mobile Networks", International Journal of Computer Research, vol. 11, no. 1, pp.1-12, 2002.
5. F. Zhu, M. Mutka and L. Ni, "Service Discovery in Pervasive Computing Environments", IEEE Pervasive Computing, vol. 4, pp.81-90, 2005.
6. R. M. Perianu, P. Hartel and H. Scholten, "A Classification of Service Discovery Protocols", `http://eprints.eemcs.utwente.nl/735/01/0000012d.pdf`, 2005.
7. S. Czerwinski, B. Y. Zhao, T. Hodes, A. Joseph, and R. Katz, "An Architecture for a Secure Service Discovery Service", in Proc. Fifth annual. Intl. conference. Mobile Computing and Networks (MobiCom '99), pp.24-35, 1999.
8. F. Zhu, M. Mutka and L. Ni, "A Private, Secure, and User-Centric Information Exposure Model for Service Discovery Protocols", IEEE Transactions on Mobile Computing, vol. 5, no. 4, pp.418-429, Apr. 2006.
9. K. Ren, W. Lou, K. Kim and R. Deng, "A Novel Privacy Preserving Authentication and Access Control Scheme for Pervasive Computing Environments", IEEE Transactions on Vehicular technology, vol. 55, no. 4, pp.1373-1384, Jul. 2006.
10. J. Kim, Z. Kim and K. Kim, "A Lightweight Privacy Preserving Authentication and Access Control Scheme for Ubiquitous Computing Environment", in Proc. 10th International Conference on Information Security and Cryptology, LNCS4817, pp.37-48, Nov. 2007.
11. S. Xu and M. Yung, "K-anonymous Secret Handshakes with Reusable Credentials", in Proc. ACM Conf. CCS, pp. 158-167, 2004.
12. M. Gruteser and D. Grunwald, "Enhancing Location Privacy in Wireless LAN Through Disposable Interface Identifiers: A Quantitative Analysis", Mobile Networks and Applications, vol. 10, no.3, pp. 315–325, 2003.