A Lightweight Privacy Preserving Authentication and Access Control Scheme for Ubiquitous Computing Environment

Jangseong Kim, Zeen Kim, and Kwangjo Kim

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

Abstract. In Ubiquitous Computing Environment (UCE), service provider wants to provide its service to only legitimate users. Some users who belong to same service provider do not want to reveal their identities while using some privacy-related services such as location information, printing, browsing web pages, *etc.* In addition, we should consider lightweight cryptographic protocols because UCE can be constructed by lots of resource and energy constrained devices. In this paper we propose a lightweight privacy-preserving authentication and access control scheme for UCE. Compared to the previous schemes [13, 14], our proposed scheme which was designed to reduce the number of public key operations and to improve non-linkability feature is found to be more secure and requires less memory on the user's device. Moreover the proposed scheme provides mutual authentication, accountability and differentiated access control.

1 Introduction

Ubiquitous Computing Environment (UCE) with their interconnected devices and abundant services promise great integration of digital infrastructure into all aspects of our lives [1, 2]. User authentication, authorization and access control are also basic requirements for various services in UCE such as Auction, e-Learning, GPS, accessing wireless LAN, e-Government, *etc.* However we cannot adapt the traditional mechanisms since they do not consider unique characteristics of UCE [3].

Especially user privacy is one of the big challenges due to the limited communication range of ubiquitous computing devices [6, 11]. Also there are many "invisible" computing devices in UCE that can collect and analyze the identities, locations and personal information of users without their prior agreement or recognition. Typical approach for dealing with user privacy protection is to provide anonymity based on blind signature scheme. Double spending problem of an authorized credential [13, 14] can happen if there is no mechanism for verification that the user is actual holder of the authorized credential. In this case a malicious user can reuse a previous credential of a legitimate user on requesting a special service. Therefore we should consider accountability for an authorized credential.

Energy management is also another big challenge in UCE because proactivity and self-tuning for providing adaptation capability increases the energy demand on software of a mobile computer in personal computing space. Consequently we should consider lightweight cryptographic protocol for reducing energy demand while providing proper security level.

There are many approaches to solve user privacy and security challenges in UCE [4, 5, 7–16]. 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 work [4, 5, 7, 9, 10, 15] focused on designing specific security infrastructures to protect user context privacy like location information from service providers. Creese *et al.* [8] and Wu *et al.* [11] revised authentication and privacy requirements and Zugenmaier *et al.* [12] showed that the use of a combination of devices using incompatible anonymizing mechanisms can compromise the anonymity, which is achieved when each device is used seperately. Recent researches [13, 14, 16] focused on designing concrete security protocols. Characteristics and limitations of these protocols are discussed in Section 2.

In this paper we propose a novel scheme for lightweight privacy-preserving authentication and access control in ubiquitous computing environment. The scheme reduces computation overhead and storage overhead on the user side, provides accountability, improves non-linkability, enhances security level by sharing a selected number set between the user and the authentication server, and does not rely on underlying system infrastructure. Also differentiated service access control is feasible by arranging users in different service groups.

The rest of this paper is organized as follows: In Section 2 we review related work, describe the system architecture of a campus UCE and mention requirements of the system. We present our proposed scheme in Section 3. Then we discuss the security features and the performance analysis of the proposed scheme in Section 4. Finally we conclude the paper in Section 5.

2 Background and Related work

2.1 Related work

Jendricke *et al.* [15] introduced an identity management system in UCE. A user can issue multiple identities and use them depending on the applications. Based on these virtual identities the scheme can protect user privacy while providing access control and user authentication. However there is no concrete protocol. He *et al.* [16] presented a simple anonymous ID scheme for UCE but the scheme cannot prevent the double spending problem since it is a direct application of Chaum's blind signature technique [17]. More recently Ren *et al.* [13, 14] proposed new scheme which can satisfy the requirements in UCE and prevent double spending problem by combining two cryptographic primitives, blind signature and hash chain. It reduces the number of signature verifications on an authentication server side. Additionally the scheme provides non-linkability and differentiated service access control, prevents double spending problem of an authorized credential, and does not rely on underlying system infrastructure such as the "lighthouse" or "mist routers" [5]. However a mobile user should store all hash chains of his/her anchor to increase performance aspect and perform public key operation whenever the user sends a service access request message even if the computation can be done off-line.

Gruteser and Grunwald [18] offered a method for hiding user's MAC address with anonymous IDs so that the user cannot be tracked in a wireless LAN environment.

2.2 System architecture for a campus UCE

Lots of researchers use a campus UCE to illustrate their example scenarios for UCE *i.e.* second scenario in [3] and its system architecture usually consists of three major components, *i.e.*, **U**, SP and AS. For supporting lots of mobile users, database server (DS) is considered as a component for the target environment. Figure 1 illustrates the typical system architecture.

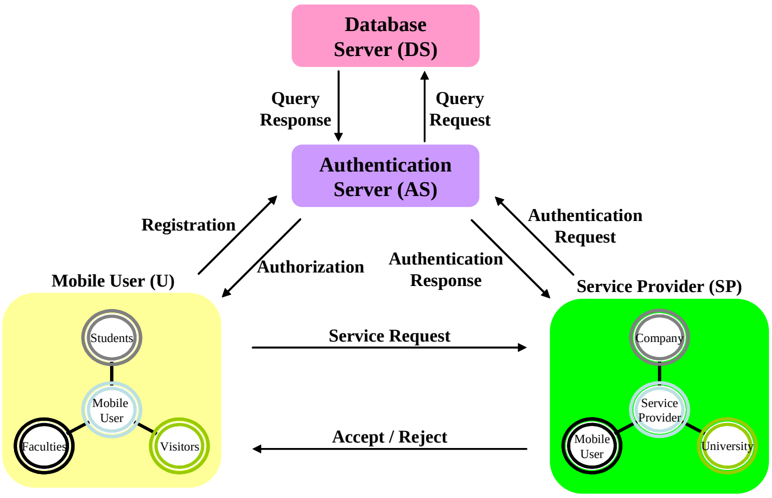

Fig. 1. System architecture

Usually the student is modeled like **U** in the system architecture. Wireless LAN, campus map, a student's time table, a class web page and printer can belong to SP. We assume that DS has all related or partial information for user authentication and has proper security methods to protect the information from the adversaries. However the methods are out of our scope of this paper.

2.3 System requirements

To support second scenario in [3], basically the system should provide user authentication and access control. Since the scenario assumes its target environment as the UCE, we also consider the characteristics, *i.e.*, energy efficiency and user privacy, of the UCE. Typical approach for protecting user privacy is to provide anonymous identity based on blind signature technique [17]. If there is no verification step to check that a **U** is an actual holder of the authorized credential then a malicious user can reuse the authorized credential [13, 14].

Based on these considerations Ren *et al.* states the system requirements for the UCE [13, 14]. The system should 1) provide explicit mutual authentication between **U** and SP; 2) allow the mobile users to anonymously interact with a SP; 3) enable differentiated service access control among different users; 4) provide flexibility, scalability to both **U** and SP; 5) generate fresh session key to secure the interaction if necessary; 6) have high efficiency with respect to communication, computation costs and management overheads; 7) provide easy accountability.

3 Our Proposed Scheme

We assume that a **U** can control the source addresses of the outgoing Medium Access Control (MAC) frames since it is a prerequisite for anonymous communications. Gruteser *et al.* [18] touched one of the detailed methods for this kind of modification and detailed of which is out of scope of this paper. Also all users' public keys, all SID and public key corresponding to each SID are stored in a DS. By sending a query message to the DS, an AS can get proper information and the mobile user knows the mapping between SID and its corresponding public key. Additionally the **U** determines n based on his/her service access frequency. The SP defines the scope and the meaning of service type, associates each user with a particular service type, assigns a unique public key to each service type and provides this information to the AS for further enforcement of authorization rules. Table 1 illustrates the notation used in this paper.

Our proposed scheme consists of two main phases. The first phase is to generate and authorize a user's credential information. Second phase is to establish a fresh session key based on the user's authenticated credential information. Our proposed scheme can hide the relationship between the authorized credential and the mobile user's real identity through blind signature technique based on the first phase. Moreover our scheme can provide non-repudiation because an anchor value contains a user's signature which consists of access frequency n, his/her identity and a fresh nonce. To provide accountability of the authorized credential we adopt selected number set. The selected number set is expressed as l-bit array. **U** only once generates it randomly during the first access request. For example if the i-th element of the array is 1, it means that i is already selected.

U	A mobile user
AS	Authentication server
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
SP	Service provider or service access point
K_{AB}	Shared secret key between entities A and B
m, X_m	A message m and its corresponding ciphertext
(m_0, m_1)	Concatenation of two messages m_0 and m_1
$\{m\}_{K_A}$	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

3.1 Credential generation

The **U** generates two fresh nonces and signs his/her identity together with one fresh nonce R'_U using own private key $PriK_U$. Then the **U** computes an anchor value C^0 with the signature. Note that the procedure can be done off-line. We summarize it as:

1. Generate two fresh nonces: R'_U and R''_U
2. Sign user's own ID with a fresh nonce $R'_U andn$:

$$\{ID_U, n, R'_U\}_{PriK_U}$$

3. Compute the anchor value C^0 of credential chain as:

$$C^0 = h(ID_U, n, R'_U, \{ID_U, n, R'_U\}_{PriK_U})$$

4. Blind C^0 as $C_U = \{R''_U\}_{PubK_{SID}} \times C^0$

3.2 Credential authorization

The **U** sends own identity, blinded credential C_U and SID with own certificate. Next the AS checks validation of the received certification and verifies whether the **U** has access permission on the service. Note that the proposed scheme can provide the differentiated service access control through this verification procedure. If the result is valid and the requestor has access permission, the AS signs on C_U and sends C_S, ID_{AS} and the received identity to the **U**. Then the **U** verifies ID_U and ID_{AS}. Only if the information is valid, the **U** can get valid credential information by unblinding the received C_S. Otherwise the **U** discards it and retries. We illustrate this procedure in Figure 2.

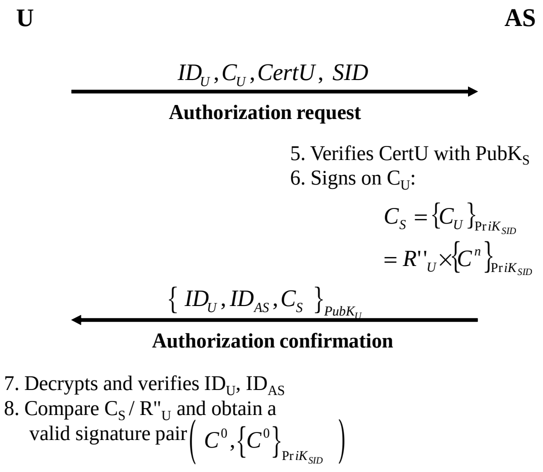
U
AS
$ID_U, C_U, CertU, SID$
Authorization request
5. Verifies CertU with PubK_S
6. Signs on C_U:
$C_S = \{C_U\}_{\mathrm{Pri}K_{SID}}$
$= R''_U \times \{C^n\}_{\mathrm{Pri}K_{SID}}$
$\{ ID_U, ID_{AS}, C_S \}_{PubK_U}$
Authorization confirmation
7. Decrypts and verifies ID_U, ID_{AS}
8. Compare C_S / R''_U and obtain a
valid signature pair $\left(C^0, \{C^0\}_{\mathrm{Pri}K_{SID}} \right)$

Fig. 2. Authorization of credential information

3.3 Verification of credential and session key establishment

Only if the **U** has the legitimacy of the target service, the **U** sends correct C^0, C^{i-1} and S^i to the AS. Also the AS store C^0, C^{i-1} and S^i to the DS only if it verifies that the credential is authorized. The AS authenticates **U** based on these facts. Moreover both entities **U** and AS can easily generate a fresh session key $K_{U,AS}$ since they shared the anchor value and S. Also the **U** discloses her previous credential information C^{i-1}.

When the **U** sends an i-th access request to the SP, the **U** generates a fresh nonce R_U^i and selects one random number j between 0 to $l-1$. Next the **U** verifies j is not in the list S. If j is in S then the **U** should select unused random number. Then she generates one time credential as $C^i = h(C^0, j^i, R_U^i)$. Also both entities **U** and AS share a secret key $K_{U,AS}$ by computing as:

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

The SP forwards the request message to the AS with a fresh nonce. After decrypting the request message, the AS checks duplication and validation of the secret information, C^{i-1} and S. There are two cases in the verification procedure:

1. When type is 0: It means that the received message is the user's "first access request". After decrypting it, the AS checks whether the requestor has an authorized credential. So the AS signs C^0 with the private key of the SID and compares the result $\{C^0\}_{PriK_{SID}}$ with the received signature. Only if the result is same, then the AS computes $C^1 = h(C^0, j^1, R_U^1)$ and stores SID, S^1, C^0 and C^1 in the DS. Otherwise the AS discards it.
2. When type is 1: To get proper C^0 and S^{i-1}, the AS sends a query message to the DS by setting C^{i-1} and SID as searching condition. If the AS can finds C^0 and S^{i-1}, then the received message can be decrypted by $K_{U,AS}$. After decrypting the request message, the AS verifies that the j-th index of the stored S^{i-1} is 0 and the stored S^{i-1} is the same as S^i except the j-th index. Only the verification result is correct, then the AS believes the **U** has legitimacy of the requested service and stores C^i and S^i in the DS. Otherwise the AS discards it. If there are several verification failure on series of the authorized credentials, the AS can request the mobile user to change his/her credential or notify that there is an impersonation attack on the **U**. Note that C^i and S^i are stored as the authorized credential and the selected number list respectively.

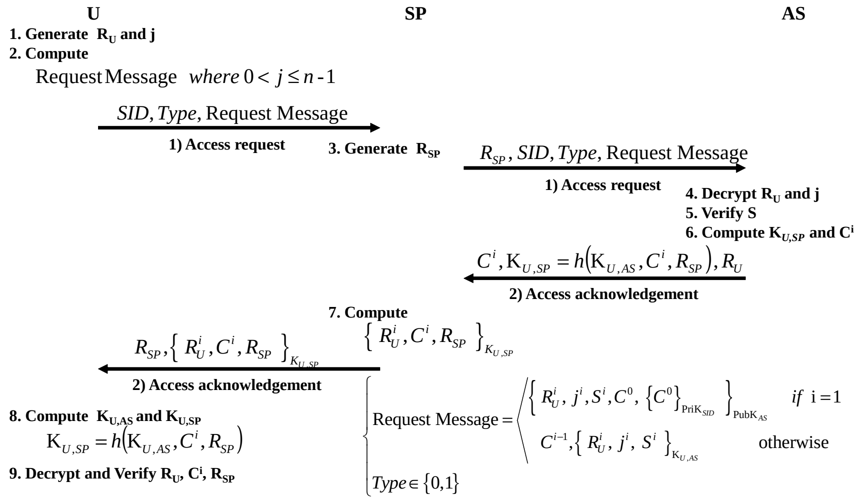
U
SP
AS
1. Generate R_U and j
2. Compute
RequestMessage $where\ 0 < j \le n-1$
$SID, Type,$ Request Message
1) Access request
3. Generate R_{SP}
$R_{SP}, SID, Type,$ Request Message
1) Access request
4. Decrypt R_U and j
5. Verify S
6. Compute $K_{U,SP}$ and C^i
$C^i, K_{U,SP} = h(K_{U,AS}, C^i, R_{SP}), R_U$
2) Access acknowledgement
7. Compute
$\{R_U^i, C^i, R_{SP}\}_{K_{U,SP}}$
$R_{SP}, \{R_U^i, C^i, R_{SP}\}_{K_{U,SP}}$
2) Access acknowledgement
8. Compute $K_{U,AS}$ and $K_{U,SP}$
$K_{U,SP} = h(K_{U,AS}, C^i, R_{SP})$
9. Decrypt and Verify R_U, C^i, R_{SP}
Request Message $= \begin{cases} \{\{R_U^i, j^i, S^i, C^0, \{C^0\}_{PriK_{SID}}\}_{PubK_{AS}} & if\ i = 1 \\ C^{i-1}, \{R_U^i, j^i, S^i\}_{K_{U,AS}} & \text{otherwise} \end{cases}$
$Type \in \{0,1\}$

Fig. 3. Verification of credential information and Key generation when the **U** sends the i-th access request

After verifying validation and duplication of the request message, the AS computes $K_{U,SP}$ which is used to secure communication between SP and **U**. Next the AS sends $C^i, K_{U,SP}$ and R_U^i to the SP. The SP encrypts received information with a fresh nonce R_{SP} by using $K_{U,SP}$. Through this activity our proposed scheme provides explicit key authentication between SP and **U**. This processes are shown in Figure 3. Since to provide a secure tunnel is not our interesting point we simply assume that there is a secure tunnel (e.g., IPsec ESP mode [19]) between SP and AS.

After computing $K_{U,AS}$ and $K_{U,SP}$, the **U** decrypts the received access acknowledgement and verifies C^i,R^i_U and R_{SP}. If the verification result is correct, the **U** can access the target service of the SP. Otherwise the **U** resend the access request to the SP.

3.4 Extension for out-of-order requests

Sometimes the **U** might want to request multiple services simultaneously. If the multiple concurrent sessions are handled by a single server, it is possible to happen that the access request messages arrive out of order at the AS due to unexpected network problems. To deal with this problem we adapt a sliding-window-based extension to the credential verification and key generation procedure on the AS side. We assume that the DS has the stored credential list, the selected number list, the nonce list and the encrypted message to deal with our-of-order requests. There are two cases to deal with out-of-order requests:

1. When the AS cannot find C^{i-1} in the authorized credential and the stored credential list
   (a) Store C^{i-1} and $\{R^i_U, j^i, C^0\}_{K_{U,AS}}$ to the stored credential list and the encrypted message respectively.
2. When the AS find C^{i-1} in the authorized credential and the stored credential
   (a) Send a query message to the DS by setting C^{i-1} and SID as searching condition for getting proper S^{i-1} and C^0.
   (b) Compute $K_{U,AS}$ and decrypt the received message.
   (c) Flip the j-th index of the stored S^{i-1} only if the index is set as 0. Otherwise, discard it.
   (d) Update C^i in the authorized credential and generate C^{i+1}. Next search the generated credential in the stored credential list. If the C^{i+1} are found in the stored credential list then repeat 2.(a)- 2.(d) steps until the searching has failed or the stored credential list has empty.

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:*** **U** is only required to save C^0, R^i, j^i, n and S. Since all credential information except the anchor value can be generated directly from the anchor value, it means that our proposed scheme does not require to store all credential information. Although **U** in [13, 14] should store C^0, to avoid repeated hash operation all credential information should be stored. In this point our proposed scheme requires less storage capability. Additionally our proposed scheme is more flexible in the view of access frequency since the information which should be stored is fixed even if the user's access frequency is increased.

- ***Computation overhead:*** Except first access request encrypted with a public key of an AS, all messages between **U** and AS are encrypted using a shared symmetric key. Therefore our proposed scheme is computationally efficient since symmetric key operation is lightweight than public key operation, We compare computation overhead of the proposed scheme with the scheme in [13, 14] in Table 2. Note that in Table 2 if we do not append the term "off-line", then the communication entity such as **U**, AS and SP, needs online computation.

		# of Pub. Key	Sig.Veri.	Nonce Gen.	Hash Oper.	# of Sym. Key
	User	1(off-line)	0	1	2	3
[13, 14]	SP	0	0	1	2	3
	AS	1	1/n	0	0	0
	User	1/n(off-line)	0	1	1	1(off-line)+1
Ours	SP	0	0	1	0	1
	AS	1/n	1/n	0	2	1

Table 2. Computation overheads comparison

- ***Communication overhead:*** The proposed scheme only requires two rounds to achieve the authenticated key establishment. Note that two rounds in authenticated key establishment protocol are minimum rounds to satisfy its goal. Let's compare the message size in the proposed scheme with the scheme in [13, 14]. When we assume that the nonce in the both scheme is 64 bits and the hash function in the both scheme is SHA-1, we can calculate the increased size of the message during n sessions:

$$\text{Increased size} = (n-1) \times (2 \times n + \log n - 159) + (2 \times n + \log n + 1).$$

If the **U**'s access frequency is 80, then the increased size of the message is 392.247. It means that 4.903 bits is increased per each session. However the message size is not critical factor in the campus UCE, the proposed scheme is efficient from the point of communication overhead.

4.2 Security

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

- ***User context privacy:*** Our proposed scheme protects the **U**'s context privacy against insiders and outsiders. Note that all communication channels are well protected. The AS can only know the **U**'s SID. Also the SP can not imagine who sends the service access 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 **U**, and 2) neither of them could link two different sessions to the same **U** [20]. In the proposed scheme non-linkability is achieved with respect to both of insiders and outsiders. Firstly the authorized credential combined with the fresh nonce is never transmitted in plaintext form. Hence the outsiders can't associate a session with a particular user and ascribe two sessions to the same user. Secondly the **U**'s all authorized credentials are derived from an anchor value and it is only known to AS and **U**. Even if the SP can get all authorized credentials except the anchor value, the SP can't link two different sessions to the same user. Moreover the authorized credential combined with the fresh nonce is never transmitted in plaintext form. Therefore the insiders can't associate a session with a particular user. Note that the AS is regarded as trusted third party.
- ***Accountability and nontransferability equivalency:*** In the proposed scheme the credentials are authorized only when the **U** is explicitly authenticated. By adapting selected set the proposed scheme can provide one-time usage of the authorized credentials. Hence the proposed scheme can prevent double spending problems. Also the proposed scheme can provide good accounting capability feature by incorporating accounting function. Furthermore the proposed scheme provides equivalent nontransferability 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 service received by own authorized credentials. This property greatly reduces the service abuse problem worried by the SPs.
- ***Data confidentiality and integrity:*** Both entities **U** and SP generate a fresh session key to protect their communications during verification and session key establishment process . Hence data confidentiality and integrity can be easily achieved using symmetric cryptography.
- ***Differentiated service access control:*** Our proposed scheme can provide differentiated service access control by classifying users into different service types. Different users are authorized accordingly based on the service types to which they belong. Hence "User authorization" is accomplished in a differentiated 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:*** The **U**'s every access request message contains S used to prove the actual holder of the message since it is randomly generated by the **U** and only known to **U** and AS. To impersonate the target user, the adversary should present S even if the adversary knows the user's anchor value. Therefore the proposed scheme enhances security level.

- ***No additional key management:*** **U** and AS can generate a shared symmetric key based on the anchor value. Also it is used only one-time. So there is no additional key management overhead by replacing the reduced public key operations with the symmetric key operations.

In table 3 we compare our proposed scheme with other similar approaches whose goal is to provide anonymous interaction between **U** and SP. Note that the SP in our proposed scheme can't link two different sessions to the same user even if the SP can get all authorized credentials except the anchor value. However the SP in the scheme which was proposed by Ren *et al.* [13, 14] can link two different sessions to the same user if the SP can get all authorized credentials except the anchor value.

	Our scheme	Ren *et al.*[13, 14]	He *et al.*[16]
Concrete protocol	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes
User context privacy	Yes	Yes	Yes
Non-linkability	Yes to outsiders, yes to SP	Yes to outsiders partially yes to SP	No
Non-transferability	Almost yes	Almost yes	No
Data confidentiality	Yes	Yes	Easy to obtain
Message integrity	Easy to obtain	Yes	Yes
Differentiated service access control	Yes	Yes	No

Table 3. Security-related features comparison

5 Conclusion

In this paper we have proposed a lightweight privacy-preserving access control for UCE which can be used as a component in middleware. Our proposed scheme is efficient in solving the conflict between user privacy and user authentication. Because user authentication requires the user identity information while user privacy needs to hide the user identity information. Additionally the proposed scheme improves non-linkability on SP's side, enhances security level and consumes less storage burden on the user's device. Moreover the proposed scheme also provides mutual authentication, accountability and differentiated access control.

In the near future we would like to extend our scheme to deal with privacy and security in the service discovery protocol which is an essential element to access network services. Also we try to show the correctness of the proposed scheme by formal verification method.

References

1. “Easy Living”, Microsoft Research, `http://research.microsoft.com/easyliving`.
2. M. Weiser, “The Computer for the 21st Century”, Scientific of American, vol. 265, Sep., 1991.
3. M. Satyanarayanan, “Pervasive computing: Vision and Challenges”, IEEE Personal Communications, Aug., vol. 8. no. 4, pp.10–17, 2001.
4. J. Al-Muhtadi, A. Ranganathan, R. Campbell and M. Mickunas, “A Flexible, Privacy-Preserving Authentication Framework for Ubiquitous Computing Environments”, Proc. 22nd International Conference on Distributed Computing Systems (ICDCS), 2002, pp. 771–776.
5. J. Al-Muhtadi, R. Campbell, A. Kapadia, D. Mickunas, and S. Yi, “Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing”, Proc. ICDCS, Vienna, Austria, 2002, pp. 65–74.
6. J. Al-Muhtadi, A. Ranganathan, R. Campbell and M. Mickunas, “Cerberus: A Context-Aware Security Scheme for Smart Spaces”, Proc. the First IEEE International Conference on Pervasive Computing and Communications (PerCom), 2003, pp. 489–496.
7. M. Burnside et al., “Proxy-Based Security Protocols in Networked Mobile Devices”, Proc. ACM SAC, Madrid, Spain, 2002, pp. 265–272.
8. S. Creese, M. Goldsmith, B. Roscoe, and I. Zakiuddin, “Authentication for Pervasive Computing”, Proc. Security in Pervasive Computing 2003, 2004, vol. 2802, pp.116–129.
9. M. Langheinrich, “A Privacy Awareness System for Ubiquitous Computing Environments”, Proc. UbiComp, 2002, vol. 2498, pp. 237–245.
10. K. Nahanishi, J. Nakazawa, and H. Tokuda, ”LEXP: Preserving User Privacy and Certifying Location Information”, Proc. 2nd Workshop Security Ubicomp, 2003.
11. M. Wu and A. Friday, “Integrating Privacy Enhancing Services in Ubiquitous Computing Environments”, Workshop on Security in Ubiquitous Computing, 4th International Ubicomp, 2002.
12. A. Zugenmaier and A. Hohl, “Anonymity for Users of Ubiquitous Computing”, Proc. Security Workshop in Ubicomp, Seattle, Washington, Oct. 2003.
13. K. Ren, W. Lou, “Privacy Enhanced Access Control in Ubiquitous computing Environments”, 2nd International Conference of Broadband Networks 2005, Vol. 1, pp. 356–365, 3-7 Oct. 2005.
14. 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, July 2006.
15. U. Jendricke, M. Kreutzer and A. Zugenmaier, “Pervasive Privacy with Identity Management”, in Proc. 1st Workshop Security, Ubicomp, 2002.
16. Q. He, D. Wu and P. Khosla, “Quest for Personal Control over Mobile Location Privacy”, IEEE Commun. Mag., vol. 42, no. 5, pp. 130–136, May 2004.
17. D. Chaum, “Untraceable Electronic Mail, Return Address, and Digital Pseudonyms”, Communications of the ACM, vol. 24, no. 2, pp. 84–88, 1981.
18. 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.
19. S. Kent and R. Atkinson, “Security Architecture for the Internet Protocol”, IETF RFC 2401, 1998
20. S. Xu and M. Yung, “K-anonymous Secret Handshakes with Reusable Credentials”, in Proc. ACM Conf. CCS, pp. 158–167, 2004.