A Privacy-Preserving Secure Service Discovery Protocol for Ubiquitous Computing Environments*

Jangseong Kim[1], Joonsang Baek[2], Kwangjo Kim[1] and Jianying Zhou[2]

[1] Department of Information and Communications Engineering, KAIST, Daejeon 305-714, Korea {jskim.withkals,kkj}@kaist.ac.kr
[2] Institute for Infocomm Research, Singapore 138632, Singapore {jsbaek,jyzhou}@i2r.a-star.edu.sg

Abstract. Recently, numerous service discovery protocols have been introduced in the open literature. Unfortunately, many of them did not consider security issues, and for those that did, many security and privacy problems still remain. One important issue is to protect the privacy of a service provider while enabling an end-user to search an alternative service using multiple keywords. To deal with this issue, the existing protocols assumed that a directory server should be trusted or owned by each service provider. However, an adversary may compromise the directory server due to its openness property.
In this paper, we suggest an efficient method of membership verification to resolve this issue and analyze its performance. Using this method, we propose a privacy-preserving secure service discovery protocol protecting the privacy of a service provider while providing multiple keywords search to an end-user. Also, we provide performance and security analysis of our protocol.

1 Introduction

Since hundreds of devices and services may surround end-users in ubiquitous computing environment, any prior knowledge about nearby environment (*i.e.*, a list of wireless access points and accessible services) could be useful to satisfy the service requirements of an end-user and choose proper services [1]. Using this knowledge, an end-user can find an alternative service and replace the current service with new one if the current service is no longer available. However, this knowledge cannot be provided to end-users due to their mobility, which may violate their desire for service continuity. This is one of important requirements why we should consider a service discovery protocol for ubiquitous computing environment.

Most existing service discovery protocols consist of three major components:

* This work was done while the first author was visiting I^{2}R, supported by I^{2}R's Postgraduate Attachment Program.

end-user, service provider, and directory server. To obtain the access information of a service, an end-user sends a query message to a directory server. A service provider periodically stores the access information of its own service in the directory server. After obtaining the access information of the target service, the end-user accesses the service. The directory server stores or provides proper service access information for simplified trust management and scalability: each end-user and service provider should identify and trust the directory server only rather than many services and end-users.

As the directory server should be accessible by anyone and placed in public place, the adversary can compromise the directory server to obtain private information of an end-user or service provider. However, the service provider can check whether the directory server gives wrong searching result, which causes the compromised directory server to operate properly against all requests. From this, we consider the directory server to be a semi-honest entity. In short, the directory server may seek to gather any sensitive information of an end-user or service provider, identify which entity submits the given keywords, and impersonate the target entity while giving correct searching result of the given keywords.

Any end-user could be one of the service providers in photo-image sharing or personal music broadcasting applications, and thus the privacy of a service provider should also be protected. To preserve the privacy of an end-user and service provider together, an authentication server, believed to be trusted third party, is required to take a role of signer and verifier in blind signature scheme.

Based on these observations, the system for ubiquitous computing environment should satisfy the following security requirements.

Mutual authentication: Mutual authentication between an end-user and service provider is required to identify whether the communicating party is a legitimate entity or not.

Anonymity and accountability: An end-user wants to preserve the private information (*i.e.*, identity, service usage, and etc) during service discovery while presenting their legitimacy and access permission. Similarly, the service provider wants to protect the private information (*i.e.*, identity, service access information, service presence information, and *etc*) from non-subscribers of the service. Although anonymous authentication can protect the privacy of an end-user and service provider, it also can help a malicious user access several services without permission.

Differentiated access control: Although a service provider wants to provide differentiated services based on access privilege of his/her subscribers, anonymous communication through an authentication server allows the server to access the subscription information of the service provider which should be protected.

Efficient keyword search on encrypted data: As many services may exist in ubiquitous computing environments, keyword search is required to allow an end-user to specify a target service satisfying the service requirements of the end-user. However, it can violate the privacy of a service provider. Anyone can access the stored service information if proper access control is not enforced.

Simple solution is to store the service information and its corresponding keywords in directory server after encrypting them. Still, this approach has three limitations: complexity of key distribution, heavy computational overhead of a directory server, and information leakage by the semi-honest directory server.
Lightweightness: As one of the main characteristics in ubiquitous computing environment is heterogeneity, cryptographic protocols should be lightweight from the view of communication and computation cost.

Let us consider the following example illustrating the convenience of service discovery protocol and its disadvantages: A famous researcher is visiting a university to give a special talk. If the researcher needs to find a place where she will give a talk and to print out some lecture notes, she should access the Internet and find out what type of printers are available. While a service discovery protocol can automatically detect and configure these service, which spares the researcher any manual setup operations, the researcher may have to reveal her sensitive information (*e.g.*, purpose, identity, and access permission) to the service provider and directory server. This disadvantage can happen when the service discovery protocol does not satisfy security requirements (*i.e.*, confidentiality, integrity, and access control). Also, the protocol supporting the security requirements still causes this disadvantage if the compromised directory server (or authentication server) disclose the received information to the adversary. However, as the previous work does not address this problem, we require a privacy-preserving secure service discovery protocol.

In this paper, we propose an efficient approach of membership verification which evaluates an encrypted polynomial representation of a given set. When the set is a keywords list to specify a service, our approach can support multiple keywords search without leaking sensitive information of the service provider. Also, our approach can remove the privacy concern regarding the abuse of subscription information by the administrator in the authentication server when the set is a subscriber list to ensure access control. Based on these results, we propose a privacy-preserving secure service discovery for ubiquitous computing environment. Moreover, our protocol provides various security-related features and is suitable for establishing a security framework for ubiquitous computing environment.

The rest of this paper is organized as follows: In Section 2, we discuss the related work. Section 3 presents our assumption and notation. In Section 4, we present our membership verification with its performance and security analysis. Then, we explain our privacy-preserving service discovery protocol in Section 5. Also, we provide performance and security analysis of our protocol in Section 6. Finally, we conclude this paper in Section 7.

2 Related work

2.1 Secure service discovery

To address these privacy and security issues, Czerwinski *et al.* [7] proposed a scheme which was called "Secure Service Discovery Service". The scheme con-

sisted of end-user, discovery server, and service provider. Through directory server, regarded as trusted entities, an end-user and service provider authenticated each other. However, they should expose their own identities and service access information during service lookup and service announcement.

In 2006, Zhu *et al.* [6] proposed the PrudentExposure model for a secure service discovery protocol. This approach can preserve the privacy of end-users and service providers based on a Bloom filter. Also, end-users should bind themselves to a nearby agent and transfer all their identities to the agent via a secure channel. However, this approach has several limitations. **First**, additional communication cost is incurred to bind and transfer end-users' identities to an agent. **Second**, privacy leakage occurs among insiders even though the model is designed to preserve sensitive information for an end-user and service provider. **Third**, the end-user should perform two public key encryptions (or decryptions) and one signing operation *whenever* he/she sends a lookup message. Although the agent near the end-user should perform this computation, we need to take into account the cost of removing privacy concern that the nearby agent can identify the user's service selection and obtain all messages between the end-user and service provider as a computational cost of the end-user. **Finally**, each service provider should have his/her own directory server.

2.2 Multiple keywords search on encrypted data

A keyword search on encrypted data is introduced to share audit log and email on a public server while minimizing information leakage. Previous protocols [12–14] have three common entities in their system models: a data provider, public server, and data retriever. The data provider generates shared information and stores it on a public server in an encrypted form. Only an entity having a proper trapdoor (*i.e.*, access permission) can retrieve the stored information. This approach can remove a strong security assumption that a directory server should be trusted. However, no access control is provided [8] and the server can link two different sessions to the same group using the relationship between the stored data and the submitted trapdoor.

To provide access control only, Yau *et al.* [8] proposed an idea to convert the searching of the sets to an evaluation of polynomial representations of a given set [9, 10] using BGN encryption [11]. Interestingly, this approach can address the second problem due to non-deterministic property of BGN encryption. However, the proposed approach is not efficient in view of computational overhead. Denote S_1 and S_2 by a set of access keys and a set of keywords, respectively. Then, the data retriever should compute $(\mid S_1 \mid + \mid S_2 \mid +1)$ exponent multiplications and BGN encryptions [11] per each query.

2.3 BGN encryption [11]

In 2005, Boneh *et al.* proposed a new homomorphic encryption scheme supporting unlimited additive operations and one multiplicative operation on encrypted

data. The proposed encryption scheme enables one entity to evaluate the encrypted data without revealing the content of encrypted data. We review the BGN encryption scheme in brief.

In BGN encryption, all operations are done on two cyclic groups G and G_1 with the same order $n = q_1 q_2$, where q_1 and q_2 are two large prime numbers. The public key PK_{BGN} is g and $h = g^{\mu q_2}$ under the group G, where μ is a random integer. The encryption of m_i, $m_i + m_j$, and $m_i m_j$ can be computed as $g^{m_i} h^{r_i}$, $g^{m_i} h^{r_i} g^{m_j} h^{r_j}$, and $e(g^{m_i} h^{r_i}, g^{m_j} h^{r_j})$ where T is a non-zero random number less than q_2, $m_i \in \mathbb{Z}_T$ be i^{th} message, r_i is i^{th} random number, and e is a bilinear mapping from $G \times G$ to G_1, respectively. The expected decryption time using Pollard's lambda method is $\tilde{O}(\sqrt{|\,T\,|})$ although the authentication server has the private key, $SK_{BGN} = q_1$.

3 Membership verification

We convert membership verification to set search by evaluating of a polynomial representing a given set [9, 10], where the set contains the service subscriber list.

Compared to membership verification cost of the previous work [8], which relies on the number of access keys and subscribers, our membership verification cost is reduced to the number of subscribers. In this point, we argue that our membership verification is an efficient approach. Note that access keys in our scheme are used to derive K_{rk} which is used to encrypt the access information of the target service and can be derived from the list of hidden encryption keys $g^{r-ak_1}, \cdots, g^{r-ak_p}$ using each subscriber's access key ak_i where i is an index of each end-user.

3.1 Assumption and notation

We assume that an end-user can control the source addresses of the outgoing Medium Access Control (MAC) frames since this assumption is a prerequisite for anonymous communications. A detailed method for this modification is covered by Gruteser *et al.* [15]. Table 1 illustrates the notations used throughout this paper.

The AS (or SP) issues SID, a polynomial $f(x)$ with degree t, access key ak_i, $E[i + r, PK_{BGN}, G_1]$, and ID_i to service provider (or end-user). Using the received information, the end-user or SP can generate their MT.

The SP stores SID list and polynomials for membership verification to the AS in an encrypted form. If there is any change in the stored information, the SP updates the stored information.

Finally, PK_{AS}, ID_{AS} and PK_{BGN} are assumed to be known to all entities.

3.2 Polynomial generation

For a set $S_1 = w_1, w_2, \cdots, w_p$, a polynomial with degree t, $f(x)$ for S_1 is defined as the following:

$$f(x) = \begin{cases} -r & x = w_i \in S_1 \\ -r' & x = w_i \notin S_1, \end{cases}$$

Table 1. Notations

AS / DS / SP	Authentication Server / Directory Server / Service Provider
$Credential$ / ID_A	A ticket for entity authentication / An identifier of entity A
n / U	A user's access frequency / End-user
PK_A / SK_A	A public key of entity A / A private key of entity A
PK_{BGN}	A public key under BGN encryption scheme [11] owned by AS
MT	A ticket for indicating subscribers of the target SP
DMV	A ticket for Discovery Membership Verification
S	A set of selected numbers the length where $\mid S \mid \geq 2n$
SID	A service type identifier describes a selected subset of the available service pool and includes an polynomial identifier for membership test
SK_{BGN}	A private key under BGN encryption scheme [11], which is owned by AS and distributed to DS for membership test
C^i or C_A^i, $i = 0, 1, \cdots$	A series of authorized credentials generated by entity A
$CertA$	A certificate that binds entity A with A's public key
j^i or j_A^i, $i = 1, 2, \cdots$	A series of a user's number selections generated by entity A
$K_{A,B}$	Shared secret key between entities A and B
$E\{m, K_A\}$	A message m is encrypted by a symmetric key K_A
$E[m, PK_A]$	A message m is encrypted by an entity A's public key
$D[m, SK_A]$	A message m is signed by an entity A's private key
$E[m, PK_{BGN}, G$ or $G_1]$	A message m is encrypted by the public key PK_{BGN} on cyclic group G or G_1 and the ciphertext is $g^m h^r$ or $g_1^m h_1^r$ where $g_1 = e(g, g)$ and $h_1 = e(e, h)$
$H(m)$	A hash value of message m using SHA-1
R^i or R_A^i, $i = 1, 2, \cdots$	A series of nonces generated by entity A where $\mid R^i \mid \geq$ 64-bit.

where r and $r'(r' \neq r)$ are random integers. Here, $w_i \in \mathbb{Z}_T$ is a user account of i^{th} subscriber (or keywords) if $f(x)$ is used for membership test in AS (or DS). Figure 1 presents an example of generating a polynomial. A public SP means that

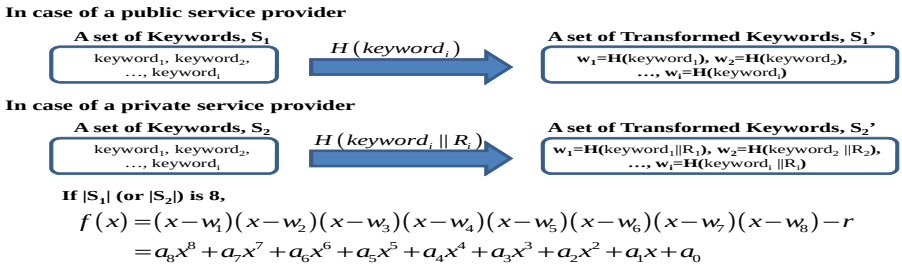

Fig. 1. An example of polynomial generation

the service is well-known to all or some of all entities in a single administration domain and presence information of the service is not important. However, a private SP indicates that the service is known to some selected entities and presence information is important. By concatenating a nonce with the given keywords and distributing the nonce to each subscriber, the private SP can hide a relationship between own service and the given keywords as shown in Figure 1.

3.3 Polynomial evaluation

For membership verification, an end-user submits w_i and $i+r$ to the membership verifier. Then, the membership verifier can check whether the user belongs to one of the service subscribers by computing $i+r+f(w_i)$. Only if $1 \leq i+r+f(w_i) \leq p$, the user is one of the service subscriber where p is the number of subscribers of target service.

However, we want to hide the detailed information of membership function from the AS, DS, and non-subscribers of the service. That's why the SP encrypts the polynomial $f(x)$' coefficients with public key PK_{BGN} on cyclic group G and $i+r$ with public key PK_{BGN} on cyclic group G_1. Also, the user encrypts $1, w_i^1, w_i^2, \cdots, w_i^t$ with public key PK_{BGN} since BGN encryption supports only one multiplicative operation on encrypted data. Then, the membership verifier performs the following steps:

1. Set $z = 1$.
2. Compute $C = \prod_{v=1}^{t-1} e(E[a_v, PK_{BGN}, \mathbb{G}], E[(w_j)^v, PK_{BGN}, \mathbb{G}])$.
3. Compute $C' = C \cdot E[a_0, PK_{BGN}, \mathbb{G}_1] \cdot E[(w_j)^t, PK_{BGN}, \mathbb{G}_1] \cdot E[i+r, PK_{BGN}, \mathbb{G}_1]$
4. Repeat the following steps until $z \leq p$.
   (a) If $C'^{(SK_{BGN})} = e(g,g)^{(z \cdot SK_{BGN})}$, return z.
   (b) z = z + 1
5. Return 0.

Using $f(x) = a_t \cdot x^t + a_{t-1} \cdot x^{t-1} + \cdots + a_1 \cdot x + a_0$ and the homomorphic properties of the BGN encryption scheme, we can change $\prod_{v=1}^{t-1} e(E[a_v, PK_{BGN}, \mathbb{G}], E[(w_j)^v, PK_{BGN}, \mathbb{G}])$ to C in the above procedure. By assuming that a_t and a_0 are both 1, C' in the above step (3) is the same as $E[i+r, PK_{BGN}, G_1] \cdot E[f(w_j), PK_{BGN}, G_1] = E[(i+r)+f(w_j), PK_{BGN}, G_1]$.

3.4 Performance analysis

To obtain the processing time of membership verification, we generate a polynomial $f(x)$ with different degree i and implement our polynomial evaluation method using pairing based cryptography library [16] under Intel®Core ™2 2.13GHz CPU, 1GB RAM and Microsoft Windows XP Professional Service Pack 3.

Table 2 shows the processing time of membership verification procedure. Rather than showing membership verification request of all subscribers of the

Table 2. Processing time of membership verification when $|PK_{BGN}| = 512$ bits and $p = 20$

Degree of a polynomial $f(x)$		i=2	i=3	i=4	i=5	i=6	i=7	i=8
Processing time (ms)	1^{st} user	48.5	85.9	121.8	159.4	195.3	232.8	270.3
	11^{th} user	112.5	148.5	184.3	220.3	257.8	296.8	332.8

target service, we present the request of $1st$ (or $11th$) end-user. As increasing the degree of a given polynomial $f(x)$, the number of pairings in step (2) and exponent multiplications in step (4) will be increased. That's why the processing time of membership verification is increased linearly. In addition, the communication cost increases linearly with the degree of a given polynomial $f(x)$. Therefore, we suggest that each SP should divide his own subscribers to several subsets based on access privilege and desired performance of our membership verification.

Let assume that Alice provides a printing service based on two different privileges such as heavy and light user. In each privilege, 30 legitimate users are subscribing the printing service. If 3 degree of polynomial $f(x)$ is sufficient to satisfy Alice's desired performance, for each privilege Alice generates 10 polynomials where the degree of polynomial $f(x)$ is 3 and each polynomial has 10 different subscribers. For membership verification, each end-user should submit $E[(i + r), PK_{BGN}, G_1]||E[(w_1), PK_{BGN}, G]\ ||E[(w_1)^2, PK_{BGN}, G_1]$ where i is the index value of an end-user among the target service subscribers and r is a random number. Then, the message size of submitted membership information is $3 \times |PK_{BGN}|$. However, we can reduce the message size by sending the x-coordinate of membership information to $1.5 \times |PK_{BGN}|$. Since PK_{BGN} is a point on cyclic group G, the encrypted result is also one of points on cyclic group G or G_1 where a point consists of x-coordinate and y-coordinate. Therefore, we can reduce the communication cost and processing delay to the desired performance of the service provider. Moreover, this approach will help us to support differentiated access control.

4 Our protocol

Before describing our protocol in detail, we illustrates our system architecture and activities of each entity in Figure 2. Our protocol consists of four phases: entity registration, service registration, discovery, and service access. Through entity registration phase, an entity (*e.g.*, end-user or SP) obtains his/her authorized credential preserving the privacy of the entity. The AS verifies the legitimacy of the entity using the received MT. In service registration phase, a service provider stores his/her service and proper information, which are required to verify the legitimacy of the end-user, to the DS. Using his/her credential and DMV, the end-user can find the registered service and obtain proper information to access the service in discovery phase.

Note that the DS can authenticate itself and shares a fresh session key with

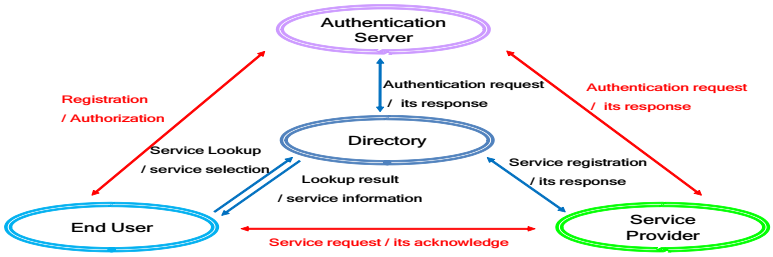

Fig. 2. System architecture for ubiquitous computing environment

the AS though entity registration phase and entity authentication phase. Using the shared session key, the DS can make the communication with the AS to be secure. From now, we assume that the communication between the DS and AS is secure.

4.1 Entity registration phase

Only if the entity is a legal entity having proper permission to access a service or provide his/her service, the AS authorizes the received credential. To hide the relationship between the authorized credential and the entity's real identity we apply blind signature technique. Figure 3 depicts entity registration phase.

To verify whether the entity is a legitimate one and has proper permission

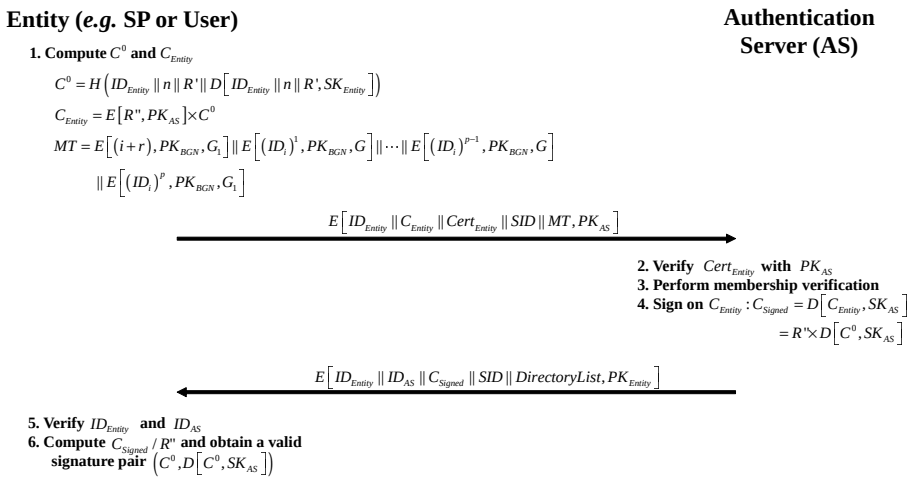

Fig. 3. Entity registration

to access or provide a service, the AS verifies the received certificate with PK_{AS}

and performs our membership verification (i.e., polynomial evaluation discussed in Section 3). If the result of the entity's membership verification is non-zero integer less than p, the AS sends proper information to the entity. Otherwise, the AS discards the received message. Note that $DirectoryList$ indicates a list of the accessible and legitimate DS.

4.2 Authentication phase

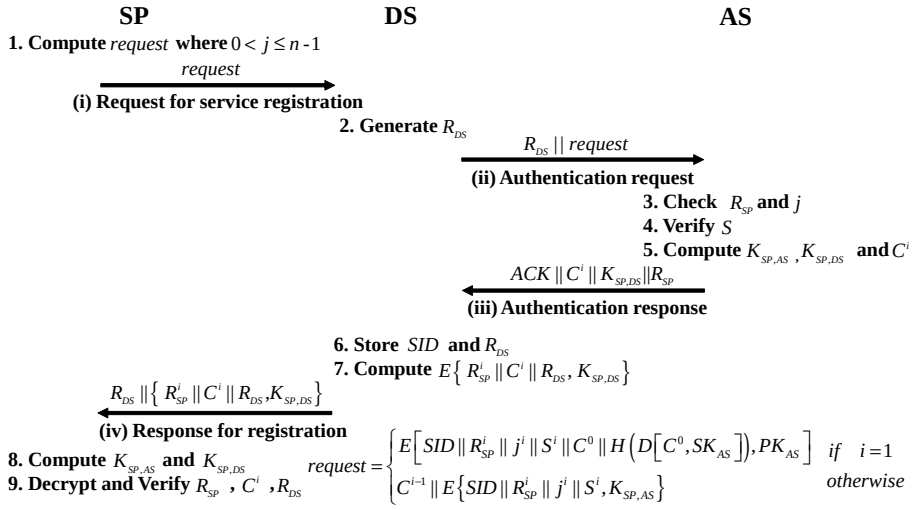

Fig. 4. Entity authentication in service registration

In the entity authentication phase, each entity establishes $K_{Entity,AS}$ and $K_{Entity,DS} = H(K_{Entity,AS}||C^i||R_{DS})$.

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

To provide accountability of the authorized credential, we adopt a set of selected numbers S, which is l-bit array. In the first access request, each entity generates the set randomly. Whenever sending an i^{th} authentication request, each entity generates a fresh nonce R^i_{Entity} and selects one random number j between 0 to $l-1$ until $j-th$ value of S is 0. Since the set is only known to the entity and AS, the adversary without knowing S cannot generate the authentication request. Therefore, we believe that our protocol can enhance security level. Note that $C^i = H(C^0||j^i||R^i)$. For entity authentication, the AS performs the following verification procedure:

1. 1^{st} request: After decrypting the request message, the AS computes $H(D[C^0, SK_{AS}])$ and compares the result with the received $H(D[C^0, SK_{AS}])$. Only if the result is same, the AS believes that the entity has an authorized credential and computes $C1 = H(C^0||j^1||R^1)$ and stores SID, S^1, C^0, and C^1 in the database. Otherwise, the AS discards the request.
2. i^{th} request: The AS finds C^0, $S^(i-1)$ and SID in the database using the received $C^(i-1)$ and decrypts the received message with $K_{Entity,AS}$. Next, the AS verifies that the entity has the same set of selected numbers and j^i is not in the set. Only if the result is correct, the AS stores the received C^i and S^i. Otherwise, the AS discards it. If the entity is a legal one with proper access permission, C^i and S^i are stored in the database. As a result, the AS can verify whether the entity has an authorized credential using the received $C^(i-1)$.

After this verification, the AS sends a response message to the DS. Then, the DS stores proper information (*i.e.*, SID and R_{DS}) and gives a response for entity authentication request to the entity. After verifying the response message, the entity computes $K_{Entity,AS}$ and $K_{Entity,DS}$. Figure 4 illustrates this phase when the entity is a SP.

4.3 Service registration phase

The service registration phase consists of entity authentication and registration. Through the entity authentication phase, an SP can anonymously authenticate himself/herself to a DS and establish the shared keys, $K_{SP,AS}$ and $K_{SP,DS}$. Using $K_{SP,DS}$, the SP registers an encrypted service access information (*e.g.*, service type, service name, service description, SID list, and network address) by $K_{rk} = H(g^r)$, encrypted coefficients of $f(x)$ with PK_{BGN}, and a list of hidden encryption keys $\{g^{r_m/ak_1}, \cdots, g^{r_m/ak_p}\}$ with the directory, where r, g, and p are a random number, a generator of cyclic group G with order $n = q_1q_2$, and the number of access keys, respectively.

Also, the SP may expose polynomial identifiers, SID list, and service type to the DS when the SP wants to serve all or partial end-users enrolled in the AS. Because the exposed access information allows an end-user without any prior knowledge about nearby environment to obtain an accessible service list, this approach can support an end-user's mobility.

4.4 Discovery phase

The discovery phase consists of three sub-steps, entity authentication, service lookup, and service selection. As the entity authentication phase has already been explained, we will skip the explanation.

Service lookup Although our keyword search is an efficient method compared to the previous approach [8], we should reduce a searching space to address scalability issue by reducing the processing time of an end-user's service lookup request in a DS or entity authentication request in the AS. Also, the end-user

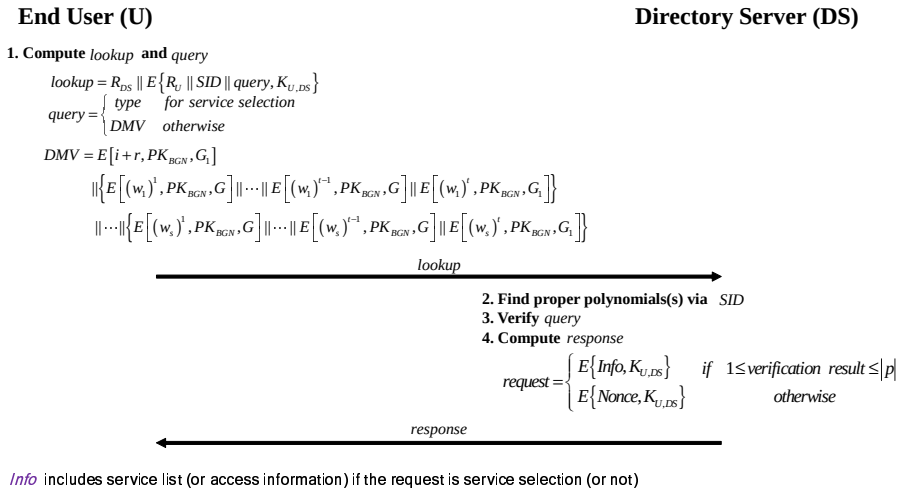

Fig. 5. Service lookup and its response in discovery phase

without having any prior knowledge may know that the same or similar type of the alternative services. That's why we use the service type as searching condition in service lookup request.

When the lookup is used to find alternative services, query is type indicating a type of the alternative services. If lookup is used to obtain service access information, query is DMV. Then, the DS finds the shared key using R_{DS}, decrypts the lookup message, and checks whether the stored SID is the same as the received SID. If the comparison is correct, the directory server performs the following steps:

1. To find alternative services: Using the type, the DS can search alternative services as some service providers expose partial access information of their services in the service registration. If any matched services exist, the DS encrypts the stored service list using $K_{U,DS}$ and sends the resulting ciphertext to the user.
2. To obtain the service access information: The DS performs membership verification procedure by evaluating the given DMV. When the verification result is not zero, the DS sends the stored access information, Em, K_{rk}, and the matched hidden key g^{r-ak_i}.

Service selection After the service lookup phase, the end-user may obtain a service list having the submitted service type. Then, the end-user selects one service from the list and notifies the selection to the DS. If a proper access control is enforced against the selected service, he/she should submit the proper DMV to the DS. As the detailed procedure is the same as the service lookup phase for obtaining service access information, we do not explain the procedure.

Table 3. Performance and security analysis

(a) Computational overhead in each phase

	Entity registration		**Service registration**			**Discovery**		
	U or SP	AS	SP	DS	AS	U	DS	AS
Public key Oper.	$(1)^{\dagger}+1$	1	$(1/N)^{\dagger}$	0	1/N	$(1/N)^{\dagger}$	0	1/N
Signature Oper.	1	1	0	0	1/N	0	0	1/N
Hash Oper.	1	0	4	0	4	4	0	4
Secret Key Oper.	0	0	$(1)^{\dagger}+2$	1	1	$(1)^{\dagger}+3$	1	1
BGN Enc. [11]	$(p+1)^{\dagger}$	0	$(t+1)^{\dagger}$	0	0	$(s\cdot(t+1))^{\dagger}$	0	0
Pairing Oper.	0	p+1	0	0	0	0	s·(t+1)	0
Exponent Oper.	0	p+2	$(t+1)^{\dagger}$	0	0	s·(t+1)	s·(t+1)+1	0

† : Precomputation Oper. : Operation Enc. : Encryption

(b) Security-related features comparison

	Our scheme	Zhu *et al.* [6]	Czerwinski *et al.* [7]
Mutual authentication	Yes	Yes	Yes
Confidentiality & Integrity	Yes	Yes	Yes
Anonymity & non-linkability	Yes	Yes to outsiders	No
Accountability	Yes	No	No
Directory server	Not trusted	Trusted	Trusted
Access control	Yes	Easy to obtain	Yes
Scalability	Good	Not too bad	Good
Enhanced security of level	Yes	No	No
The abuse of subscription information	None	Yes	Yes

4.5 Service access phase

To preserve an end-user's privacy during the service access phase, an anonymous authentication protocol is required. Our protocol can support this without additional computational cost since the end-user already has the authorized credentials after the entity registration phase. The detailed procedure is similar to the entity authentication phase. The difference is that the entities participating in the authentication process are the end-user, SP, and AS.

5 Analysis of our protocol

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

5.1 Performance analysis

Storage overhead Each end-user should store $E[(i+r), PK_{BGN}, G_1]$, access key, PK_{BGN}, and one 5-tuples (C^0, R^i, j^i, N, S) for service discovery and service access request. The service provider needs to save a set of polynomials

$f_1(x), f_2(x), \cdots, f_k(x)$ presenting a subset of own subscriber and one 5-tuple (C^0, R^i, j^i, N, S) for service registration.

Computational overhead Table 3(a) shows the computational overhead of our protocol. Note that p is the number of access keys in S_1, t is the number of keywords in S_2, s is the number of keywords specifying the target service and $1/N$ indicates that one operation is needed during N sessions. During discovery phase, the user should compute (one secret key encryption, two secret key decryptions, and four hash operations) per each discovery request while computing ($1/N$ public key encryption, one secret key encryption, $s \times (t+1)$ BGN encryptions, and $s \times (t+1)$ exponent multiplications) before the session. Compared with the previous protocol in [6], where the end-user should compute two public key encryptions and one signature generation, our protocol needs less computational cost.

Communication overhead Our protocol needs four rounds during service discovery. Previous protocol in [6] also requires four rounds. To discuss with the size of communication message, let assume that SHA-1, AES-128, and ECC-160 are used as hash function, symmetric encryption scheme, and asymmetric encryption scheme, respectively. Also, service identifier is 8 bit, N is 80, the given polynomial identifier is 24 bit, and degree of polynomial is 4. Although the previous protocol [6] requires 1104 bits, which is less than our protocol (*i.e.*, 1920 bits in i^{th} request or 2016 bits in 1^{st} request), the previous protocol does not consider that communication cost of agree on the hash functions to be used service match in advance. Also, certificate exchange between an end-user and directory server is not considered. To this point, our protocol is reasonable in communication overhead.

5.2 Security analysis

Our protocol provides the following security-related features. In Table 3(b), we compare the security-related features of our protocol with previous work.

Mutual authentication The end-user authenticates the AS through a public key of the AS and knowledge of the corresponding private key. Also, the AS authenticates the end-user using an authorized credential of the end-user.

Anonymity Our protocol protects privacy of an end-user against insiders and outsiders. As the each user authenticates herself to insiders (*i.e.*, SP and DS) using authorized credentials, the insiders cannot predict who sends the service access request or lookup request. Here, insiders other than the user cannot find any relationship among the authorized credentials, in that the credentials are derived from initial credential C^0 using one-way hash function. Outsiders also cannot identify who sends the messages since all communication messages are encrypted using a shared session key.

Non-linkability Non-linkability means that, for insiders (*i.e.*, SP and DS) and outsiders, 1) neither of them can ascribe any session to a particular end-user, and 2) neither of them can link two different sessions to the same user [17]. More precisely, non-linkability needs to prevent insiders and outsiders from obtaining an end-user's private information. Our protocol can achieve non-linkability with

respect to both insiders and outsiders. First, the information to distinguish each user is never transmitted in a plaintext form. As a result, outsiders cannot associate a session with a particular user and ascribe two sessions to the same user. Second, outsiders and insiders cannot find any relationship between the exposed credentials due to the one-way hash function. Third, as the given DMV is non-deterministic, the DS cannot link two different sessions to the same user. Finally, all communications are protected by a fresh session key.

Accountability The credentials are authorized only when the end-user is explicitly authenticated and has proper access permission on the service. By adopting a set of selected numbers, our protocol can provide a one-time usage of the authorized credentials to prevent an attacker from reusing the authorized credentials. Also, our protocol can provide good accounting capability by incorporating an accounting function.

Data confidentiality and integrity All communications are protected by a shared session key or the receiver's public key. In this point, our protocol supports data confidentiality. Although we do not explain explicitly how to generate a key for integrity check, end-user, SP, DS, and AS can derive the key using the shared information such as a fresh session key (or the receiver's public key) and exchanged nonce. By applying HMAC with the derived key, our protocol can support data integrity.

Enhanced level of security Every access request message contains S, randomly generated by the end-user and delivered only to the AS, to prove the actual holder of the message. Thus, an adversary is required to present S even if the adversary knows the target user's initial credential. In this point, the proposed scheme enhances the level of security.

Less the abuse of subscription information In our protocol, the administrator can only identify the subscribers of the target service provider when he/she monitors all registration requests. However, a proper operation policy for AS can prevent illegal tracking of all registration requests. In this way, our approach reduces the privacy concern of each service provider regarding the abuse of subscription information.

6 Conclusion

In this paper, we proposed a privacy-preserving secure service discovery protocol for ubiquitous computing environment. Our main contribution is to provide an efficient membership verification procedure while preventing information leakage regarding privacy from a semi-honest directory server. Through actual performance on our membership verification procedure, we show that our protocol is practical. However, the communication cost for membership verification increases linearly as the degree of polynomial $f(x)$. To relieve this limitation we suggest that each service provider should divide all the subscribers to several subsets, which is also useful to reduce processing delay of service discovery and support differentiated access control by assigning a different privilege to each subset.

In addition, our protocol requires fewer computations compared with the previous approach in [6], while providing more useful features. Finally, our protocol is effective in establishing a security framework since the protocol can support anonymous authentication in a service access phase without additional computation cost.

References

1. M. Satyanarayanan, "Pervasive computing: Vision and Challenges", IEEE Personal Communications, Aug., vol. 8. no. 4, 2001, pp.10-17.
2. C. Ellison, "Home Network Security", Intel Technology J., vol. 6, 2002, pp. 37-48.
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, 2002, pp.1-12.
5. F. Zhu, M. Mutka and L. Ni, "Service Discovery in Pervasive Computing Environments", IEEE Pervasive Computing, vol. 4, 2005, pp.81-90.
6. 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, Apr. 2006, pp.418-429.
7. S. Czerwinski, B. Y. Zhao, T. Hodes, A. Joseph, and R. Katz, "An Architecture for a Secure Service Discovery Service", Proc. of Fifth annual International Conf. on Mobile Computing and Networks (MobiCom '99), Aug. 1999, pp.24-35.
8. S. S. Yau and Y. Yin, "Controlled Privacy Preserving Keyword Search", Proc. of ACM Symposium on Information, Computer & Communication Security (ASIACCS '08), Mar. 2008, pp. 321-324.
9. M. J. Freedman, K. Nissim and B. Pinkas, "Efficient Private Matching and Set Intersection", Advances in Cryptography - EUROCRYPT '04, LNCS 3027, May 2004, pp. 1-19.
10. L. Kissner and D. X. Song, "Privacy-preserving Set Operations", Advances in Cryptology - CRYPTO '05, LNCS 3621, Aug. 2005, pp. 241-257.
11. D. Boneh, E.-J. Goh and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts", Theory of Cryptography (TCC '05), LNCS 3378, Feb. 2005, pp. 325-341.
12. D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search", Advances in Cryptology - EUROCRYPT '04, LNCS 3027, May 2004, pp. 506-522.
13. P. Golle, J. Staddon and B. Waters, "Secure Conjunctive Search over Encrypted Data", Proc. of Applied Cryptography and Network Security (ACNS '05), LNCS 3089, Jun. 2005, pp. 31-45.
14. J. Baek, R. Safavi-Naini and W. Susilo, "Public Key Encryption with Keyword Search Revisited", Cryptology ePrint Archive, Report 2005/191.
15. 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, 2003, pp. 315-325.
16. B. Lynn, Pairing Based Cryptography, http://crypto.stanford.edu/pbc/.
17. S. Xu and M. Yung, "K-anonymous Secret Handshakes with Reusable Credentials", in Proc. of the 11th ACM Conf. on Computer and communications security (CCS '08), Oct. 2004, pp. 158-167.