PAPER Special Section on Trust, Security and Privacy in Computing and Communication Systems

An Efficient and Secure Service Discovery Protocol for Ubiquitous Computing Environments*

Jangseong KIM[†], **Joonsang BAEK**[††], **Jianying ZHOU**[†††], ***Nonmembers***, ***and*** **Taeshik SHON**[††††a)], ***Member***

SUMMARY 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 verification of service subscribers to resolve this issue and analyze its performance and security. Using this method, we propose an efficient and 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.

key words: *privacy protection, service discovery protocol, ubiquitous, privacy-preserving authentication, membership verification*

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.

To illustrate the advantages of service discovery protocol, we present two examples: Alice has reserved a meeting with a doctor in the nearby hospital. As it is first time, she does not know the exact location. Through service discovery protocol, Alice can obtain direction guide from location supporting services (*i.e.*, GPS-based navigation in outdoor environment and RF-based navigation in indoor environment).

As the duty doctors should be ready to deal with emergency case, the corresponding staffs in the emergency room need to identify the current status of the duty doctors (*e.g.*, location, number of mobile phone, and contact method). The duty doctors should notify their status to the corresponding staffs. Through service discovery protocol, the duty doctors can easily notify their status without any background knowledge. These two example present why we should consider a service discovery protocol for ubiquitous computing environment.

Most existing service discovery protocols [2]–[6] consist of three major components: 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. Figure 1 depicts the system architectures of the previous approaches [5], [6].

However, these approaches [5], [6] do not consider the following situation: as the directory server should be accessible by anyone and placed in public place, the adversary may compromise the directory server to obtain private

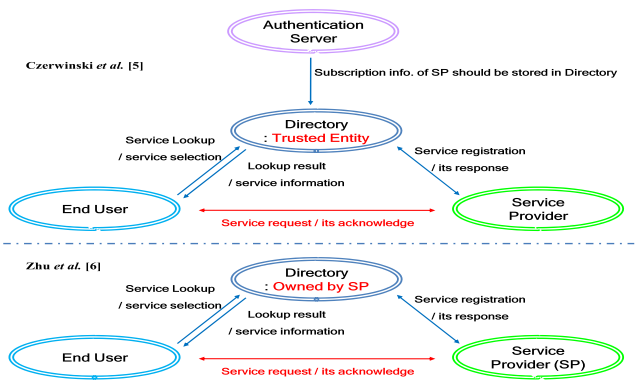


Fig. 1 System architecture of the existing approaches.

Manuscript received March 15, 2011.
Manuscript revised June 30, 2011.

[†]The author is with the Department of Information and Communications Engineering, KAIST, Korea.
[††]The author is with Khalifa University of Science, Technology, and Research (KUSTAR), UAE.
[†††]The author is with Institute for Infocomm Research, Singapore 138632, Singapore.
[††††]The author is with Division of Information and Computer Engineering, College of Information Technology, Ajou University, Suwon 443–749, Korea. Corresponding author.

*This paper is the extended version of the following paper: Jangseong Kim, Joonsang Baek, Kwangjo Kim, and Jianying Zhou, "A Privacy-Preserving Secure Service Discovery Protocol for Ubiquitous Computing Environments", Proc. of EuroPKI 2010, Sep. 23–24, 2010, Athens, Greece.

a) E-mail: tsshon@ajou.ac.kr

DOI: 10.1587/transinf.E95.D.117

information of an end-user or service provider. In addition, 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 should 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. As the end-user should expose his/her identity, access permission on the target service, and location to the directory server for service continuity, the directory server can easily collect the private information of the end-user.

As the end-user (*i.e.*, duty doctor) can be a service provider, the private information of the service provider consists of his/her own private information (*i.e.*, name, number of mobile phone, and home address) and the necessary information providing his/her service only to the legitimate subscribers (*i.e.*, service type, service description, network IP address, location, and the list of the service subscribers). To illustrate the reason why we should protect the privacy of the service provider, let consider the duty doctor case in the above examples: the adversary can obtain the duty doctor's private information (*e.g.*, IP address, number of mobile phone number, name, and location) from the directory server as the duty doctors should register themselves with the directory server. If the adversary monitors the directory server during certain time period, the adversary can identify the private information of all staffs in the hospital.

Based on these observations, the system for ubiquitous computing environment should satisfy the following security requirements: Mutual authentication, Anonymity and accountability, Differentiated access control, Efficient keyword search on encrypted data, and Lightweightness.

In this paper, we propose an efficient and secure service discovery protocol which satisfies the above security requirements. To address the problems of the existing approaches [5], [6], we present an efficient verification of service subscribers 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. Compared to the previous approaches [5], [6], our approach is efficient in terms of computation cost and communication overhead while preventing the semi-honest directory from obtaining any private information. Figure 2 shows our system architecture and activities of each entity. We assume that the communication between the DS and AS is secure.

The rest of this paper is organized as follows: In Sect. 2, we discuss the related work. Section 3 presents our assumption and notation. Also, we present our membership verification with its performance and security analysis

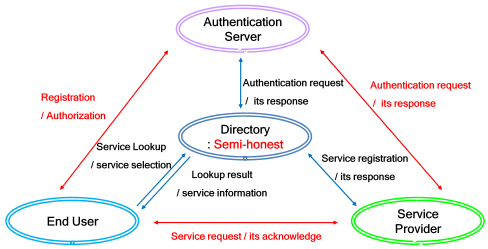


Fig. 2 Our system architecture for ubiquitous computing environment.

in Sect. 3. Then, we explain our privacy-preserving service discovery protocol in Sect. 4. Also, we provide performance and security analysis of our protocol in Sect. 5. Finally, we conclude this paper in Sect. 6.

2. Related Work

2.1 Secure Service Discovery

To address these privacy and security issues, Czerwinski *et al.* [5] proposed a scheme which was called "Secure Service Discovery Service". The scheme consisted 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 protect sensitive data for end-users and service providers. 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 [9]–[11] 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 [7] 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.* [7] proposed an idea to convert the searching of the sets to an evaluation of polynomial representations of a given set [17], [18] using BGN encryption [8]. Interestingly, this approach can address the second problem due to non-deterministic property of BGN encryption. However, the proposed approach is not efficient in terms of computational overhead. Let S_1 and S_2 denote a set of access keys and a set of keywords, respectively. Then, the data retriever should compute $(|S_1|+|S_2|+1)$ exponent multiplications and BGN encryptions [8] per each query. Also, the server should compute $|S_1|+|S_2|+1$ pairing operations and $2 \cdot |S_1| + |S_2| + 1$ exponent multiplications.

However, as these approaches only support exact keyword search, end-users should input only correct keywords. When the end-users input invalid keywords due to minor typos and format inconsistencies, they will get wrong searching results. To address this problem, Li *et al.* proposed fuzzy keyword search over encrypted data providing tolerance to minor typos and format inconsistencies [12]. In order to address the above problems, edit distance [13], used to measure the string similarity, and wildcard-based fuzzy set construction, enforcing each service provider to build a set of the possible keywords based on wildcard, were suggested.

In order to allow end-users to capture the relevance of the encrypted data, Wang *et al.* proposed secure ranked keyword search over encrypted data [14]. When the relevance of the encrypted data, which can be obtained by the various scoring mechanisms [15], is available to the end-users, they can get more correct result. "Through an order-preserving symmetric encryption scheme [16] which maintains the relevance of the plaintext data even after encryption, the proposed search scheme can support ranked search for end-users. When the data provider registers the data to be shared, the data provider should build proper relevance of the encrypted data. However, this approach cannot provide differentiated access control. Since the specification of a service can be effectively done by the keywords describing the properties of the service, the relevance information in service discovery is less important. That's why we consider exact keyword search in this paper.

2.3 BGN Encryption [8]

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. Verification of the Service Subscribers

We convert verification of the service subscribers to set search by evaluating of a polynomial representing a given set [17], [18], where the set contains the service subscriber list. From now, we call verification of the service subscribers as membership verification.

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.* [19]. 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:

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

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 3 presents an example of generating a polynomial. A public SP means that 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 Fig. 3.

Table 1 Notations.

AS / DS / SP / U	Authentication Server / Directory Server / Service Provider / End-user
$Credential$ / ID_A / n	A ticket for entity authentication / An identifier of entity A / A user's access frequency
$CertA$ / VSS	A certificate that binds entity A with A's public key / Verification of the service subscriber
MT / DMT	A ticket for VSS which indicates subscribers of the target SP / A ticket for VSS in discovery phase
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 [8] owned by AS
S	A set of selected numbers the length where $\|S\| \geq 2n$
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 [8], which is owned by AS and distributed to DS for membership test
C^i or C^i_A, $i = 0, 1, \cdots$	A series of authorized credentials generated by entity A
j^i or j^i_A, $i = 1, 2, \cdots$	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
$E[m, PK_A]$ or $D[m, SK_A]$	A message m is encrypted by an entity A's public key or 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^i_A, $i = 1, 2, \cdots$	A series of nonces generated by entity A where $\|R^i\| \geq 64$-bit.

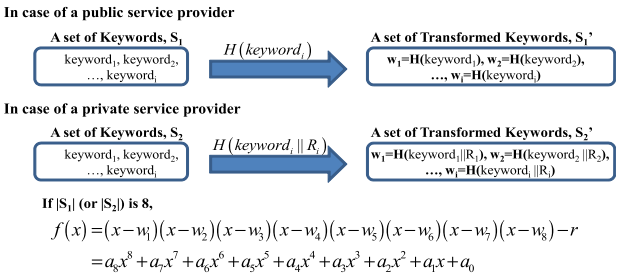


Fig. 3 An example of polynomial generation.

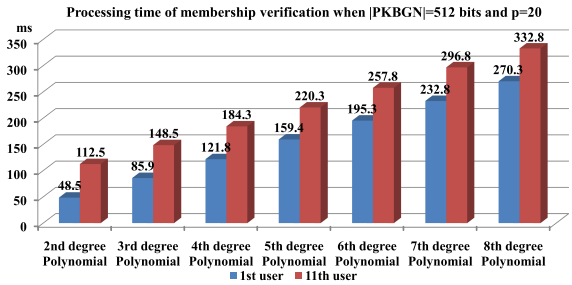


Fig. 4 Processing time of membership verification when $|PK_{BGN}| = 512$ bits, $s = 10$, and $p = 20$.

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 [20] under Intel® Core™2 2.13 GHz CPU, 1 GB RAM and Microsoft Windows XP Professional Service Pack 3.

Figure 4 shows the processing time of membership verification procedure. 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. As 'z' in step (4) starts from 1, the request of $1st$ end-user requires only one exponent multiplication. However, the request of $11th$ end-user requires 11 exponent multiplications. That's why the time difference in Fig. 4 is a constant at all degree. By comparing the request of 1st end-user and that of 11th end-user, we can easily compute the processing time of one exponent multiplication. The processing time of one exponent multiplication (or pairing operation) is 6.4 ms (or

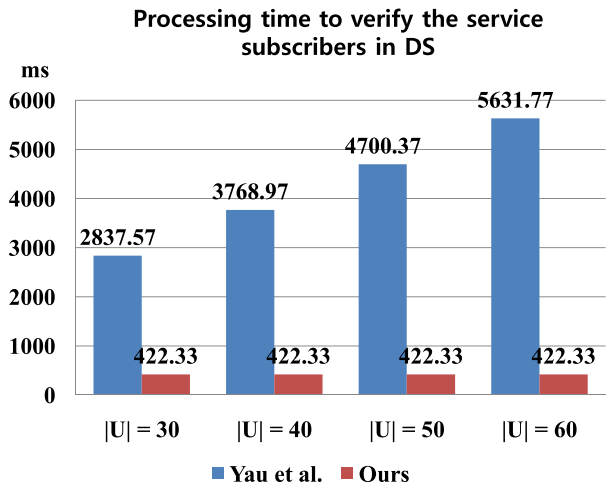


Fig. 5 Performance comparison between Yau *et al.*'s approach and ours when $|PK_{BGN}| = 512$ bits and $|t| = 10$.

36.97 ms). As the communication cost increases linearly with the degree of a given polynomial $f(x)$, we suggest that each SP should divide his own subscribers to several subsets based on access privilege and desired performance. Compared to verification cost of service subscribers in the previous work [7], our membership verification is more efficient approach. Figure 5 depicts the performance comparison between Yau *et al.*'s approach and ours. Note that $|t|$ is the number of access keys in a subset of the service subscribers. While Yau *et al.*'s approach cannot allow each SP to divide his own subscribers to several subsets based on access privilege, our approach can allow this. Using this idea, the SP can divide the service subscribers to several subsets which have 10 degree of polynomial $f(x)$. Although the size of the service subscribers increases, the processing time to verify the service subscribers in DS is constant value. The constant value is the same as the processing time when 10 degree of polynomial $f(x)$ is given to DS for verification of the service subscribers.

3.5 Security Analysis

Based on the following observations, we can hide subscription information of a target service from a computationally-bounded adversary having a ciphertext, PK_{BGN}, and SK_{BGN}. First, the expected decryption time of the given ciphertext, encrypted by PK_{BGN}, using Pollard's lambda method is $\tilde{O}(\sqrt{|T|})$ when the adversary has the private key $SK_{BGN} = q_1$ [8]. As a result, the adversary cannot obtain any useful information in polynomial time.

Second, when the size of public key is larger than 512 bits, our membership verification is resilient against the MOV or Frey-Ruck attack [21], [22], which is a method to break the discrete log problem in $\mathbb{F}_q$ after breaking the discrete log problem in $\mathbb{F}_{q^k}$ using finite field discrete logarithm algorithms such as index calculus. Since the size of public key is larger than 512 bits and k in BGN encryption is 2, the ciphertext in $\mathbb{F}_{q^k}$ is large enough, index calculus method is infeasible in $\mathbb{F}_{q^k}$.

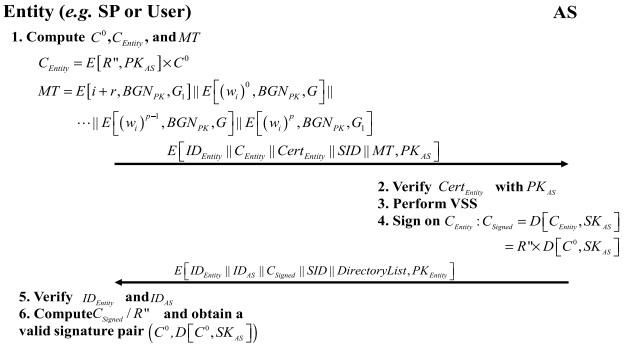


Fig. 6 Entity registration.

Hence, here comes our third observation. Even if the adversary can decrypt the given ciphertext in polynomial time, the adversary obtains keywords specifying the target service and index of the target user. However, the information is only useful to identify presence information of the target service and the service type since an end-user, who passed membership verification, can obtain the encrypted service access information only. Moreover, the service access information is encrypted with a random key $K_{rk} = H(g^{r_m})$ that can be derived from the list of hidden encryption keys $\{g^{r_m/ak_1}, \cdots, g^{r_m/ak_p}\}$ using each subscriber's access key ak_i where i is an index of each end-user. As a result, the adversary without a random key, K_{rk}, can only obtain presence information of the target service. Whether the service provider is public or private, leakage of presence information is not a problem. Since presence information of the target service provider (*e.g.*, the reserved doctor by the patient in hospital) is not important, the decryption of given ciphertexts is not valuable. In the case of private service provider implying that presence information is important, he/she can hide a relationship between his/her service and the given keywords by concatenating a nonce like the idea in Fig. 3. Note that the private service provider should distribute the nonce, R^i, when each end-user becomes a subscriber of the target service.

4. Our Protocol

To preserve the privacy of an end-user, we introduce an authentication server, believed to be trusted third party, in our system model to take a role of signer and verifier in blind signature scheme.

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 6 depicts entity registration phase.

To verify whether the entity is a legitimate one and has

proper permission 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 Sect. 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

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 ith 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 $C^1 = H(C^0||j^1||R^1)$ 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 7 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

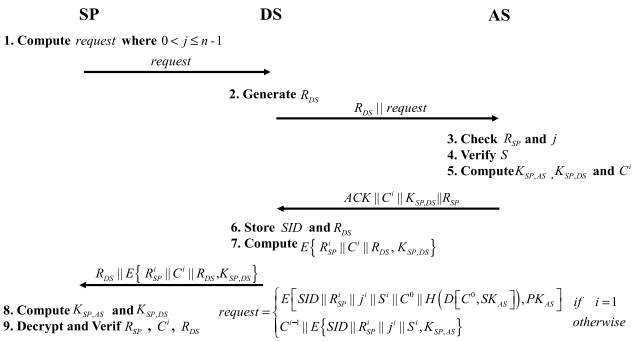


Fig. 7 Entity authentication in service registration.

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 [7], 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 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. Figure 8 depicts this phase.

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 DMT. 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

Table 2 Computational overhead in each phase.

	Entity registration		Service registration			Discovery		
	U or SP	AS	SP	DS	AS	U	DS	AS
Public key Operation	$(1)^{\dagger}$ + 1	1	$(1/N)^{\dagger}$	0	1/N	$(1/N)^{\dagger}$	0	1/N
Signature Operation	1	1	0	0	1/N	0	0	1/N
Hash Operation	1	0	4	0	4	4	0	4
Secret Key Operation	0	0	$(1)^{\dagger}$ + 2	1	1	$(1)^{\dagger}$ + 3	1	1
BGN Encryption [8]	$(p+1)^{\dagger}$	0	$(t+1)^{\dagger}$	0	0	$(s\cdot(t+1))^{\dagger}$	0	0
Pairing Operation	0	p+1	0	0	0	0	s·(t+1)	0
Exponent Operation	0	p+2	$(t+1)^{\dagger}$	0	0	s·(t+1)	s·(t+1)+1	0

† : Precomputation

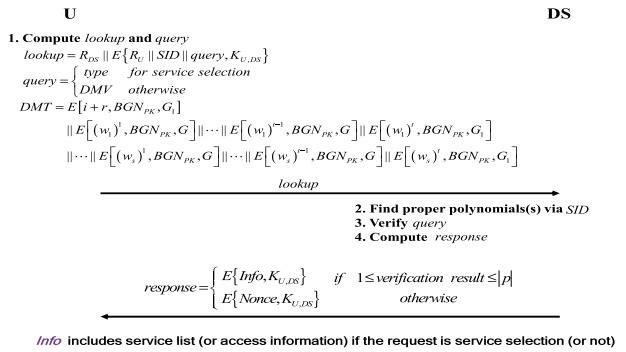


Fig. 8 Service lookup and its response in discovery phase.

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 DMT. 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 DMT 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.

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 2 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 for cryptography primitives. 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, we compare the security-related features of our protocol with previous work.

Mutual authentication: The end-user authenticates the AS

Table 3 Security-related features comparison.

	Our scheme	Zhu *et al.* [6]	Czerwinski *et al.* [5]
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
Abuse of subscription information	None	Yes	Yes

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 [23]. 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.

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.

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 an efficient and 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 performance and security analysis of our membership verification procedure, we show that our protocol is practical approach with proper security level. Also, we suggest that each service provider should divide all the subscribers to several subsets. According our performance analysis in Sect. 3.4, the proper size of subset indicating the service subscribers should be 3 or 4 in order to guarantee that the processing delay is less than a few hundred milliseconds.

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

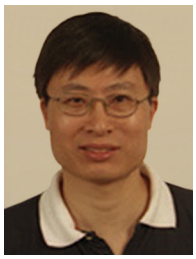
References

[1] M. Satyanarayanan, "Pervasive computing: Vision and challenges," IEEE Pers. Commun., vol.8. no.4, pp.10–17, Aug. 2001.
[2] C. Ellison, "Home network security," Intel Technology J., vol.6, pp.37–48, 2002.
[3] 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.
[4] Sun Microsystems, "Jini technology core platform specification," Version 2.0, http://www.sun.com/software/jini/specs/, 2003.
[5] S. Czerwinski, B.Y. Zhao, T. Hodes, A. Joseph, and R. Katz, "An architecture for a secure service discovery service," Proc. Fifth Annual International Conf. on Mobile Computing and Networks (MobiCom '99), pp.24–35, Aug. 1999.
[6] F. Zhu, M. Mutka, and L. Ni, "A private, secure, and user-centric information exposure model for service discovery protocols," IEEE Trans. Mobile Computing, vol.5, no.4, pp.418–429, April 2006.
[7] S.S. Yau and Y. Yin, "Controlled privacy preserving keyword search," Proc. ACM Symposium on Information, Computer & Communication Security (ASIACCS '08), pp.321–324, March 2008.
[8] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," Theory of Cryptography (TCC '05), LNCS 3378, pp.325–341, Feb. 2005.
[9] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," Advances in Cryptology - EUROCRYPT '04, LNCS 3027, pp.506–522, May 2004.
[10] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive search over

encrypted data," Proc. Applied Cryptography and Network Security (ACNS '05), LNCS 3089, pp.31–45, June 2005.
[11] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," Cryptology ePrint Archive, Report 2005/191.
[12] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," Proc. IEEE INFOCOM 2010, San Diego, CA, USA, pp.1–5, March 2010.
[13] V. Levenshtein, "Binary codes capable of correcting spurious insertions and deletion of ones," Problems of Information Transmission, vol.1, no.1, pp.8–17, 1965.
[14] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure ranked keyword search over encrypted cloud data," Proc. 2010 International Conference on Distributed Computing Systems (ICDCS 2010), pp.253–262, Genova, Italy, June 2010.
[15] A. Singhai, "Modern information retrieval: A brief overview," IEEE Data Engineering Bulletin, vol.24, no.4, pp.35–43, 2001.
[16] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Nelli, "Order-preserving symmetric encryption," Proc. Eurocrypt '09, LNCS 5479, pp.224–241, 2009.
[17] M.J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," Advances in Cryptography - EUROCRYPT '04, LNCS 3027, pp.1–19, May 2004.
[18] L. Kissner and D.X. Song, "Privacy-preserving set operations," Advances in Cryptology - CRYPTO '05, LNCS 3621, pp.241–257, Aug. 2005.
[19] 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.
[20] B. Lynn, Pairing Based Cryptography, http://crypto.stanford.edu/pbc/
[21] A. Menezes, T. Okamoto, and S. Vanstone, "Reducing elliptic curve logarithms to logarithms in a finite field," STOC '91: Proc. Twenty-Third Annual ACM Symposium on Theory of Computing, pp.80–89, New York, NY, USA, 1991.
[22] G. Frey and H. Ruck. "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves," Math. of Computations, vol.62, pp.865–874, 1994.
[23] S. Xu and M. Yung, "K-anonymous secret handshakes with reusable credentials," Proc. 11th ACM Conf. on Computer and Communications Security (CCS '04), pp.158–167, Oct. 2004.

Jangseong Kim received the B.S. degree in Computer Engineering from Kyungpook University, Korea and the M.S. degree in Computer Science from Information and Communications University, Korea, respectively 2004 and 2006. He is presently Ph.D. degree in Information and Communications Engineering, Korea Advanced Institute of Science and Technology, Korea. His interests are in network security and security for ubiquitous computing environment.

Joonsang Baek received his Ph.D. in Computer Science from Monash University, Australia. Prior to his Ph.D., he obtained B.S. degree in Mathematics from Pohang University of Science and Technology, Korea, and M.S. degree in Computer Science from Information and Communications University, Korea. He worked as a scientist at Institute for Infocomm Research, Singapore. He is now Assistant Professor at the Khalifa University of Science, Technology, and Research (KUSTAR), UAE. His research interests are in the area of applied cryptography, information and network security.



Jianying Zhou is a senior scientist at Institute for Infocomm Research (I2R), and heads the Network Security Group. He worked in China, Singapore, and USA before joining I2R. He received Ph.D. degree in Information Security from University of London in 1997. His research interests are in computer and network security, cryptographic protocol, digital signature and non-repudiation, mobile and wireless communications security, and secure electronic commerce.

Taeshik Shon (M'10) is an assistant professor in Ajou University. He received his Ph.D. degree in Information Security from Korea University, Seoul, Korea, 2005 and his M.S. and B.S. degree in computer engineering from Ajou University, Suwon, Korea, 2000 and 2002, respectively. He worked as a senior engineer in the Convergence Solution Team, DMC R&D Center of Samsung Electronics Co., Ltd.