IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 5, 2018, accepted September 10, 2018, date of publication October 5, 2018, date of current version October 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2871447

Privacy Preserving User Based Web Service Recommendations

SHAHRIAR BADSHA[1,2], XUN YI[1], IBRAHIM KHALIL[1], DONGXI LIU[3], SURYA NEPAL[3], AND KWOK-YAN LAM[4], (Senior Member, IEEE)

[1]Computer Science and Software Engineering Department, RMIT, Melbourne, VIC 3001, Australia
[2]Data61, CSIRO, Melbourne, VIC 3008, Australia
[3]CSIRO, Sydney NSW 1710, Australia
[4]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798

Corresponding author: Shahriar Badsha (shahriar.badsha@rmit.edu.au)

ABSTRACT The Quality of Service (QoS)-based personalized web service recommendations have been gaining increasing popularity due to its ability to assist users in finding high quality web services. For this purpose, Collaborative Filtering (CF)-based technique has been a useful approach in that it is able to predict with high accuracy the QoS values of web services which are not invoked by the users. The basic idea behind CF-based techniques is that they identify users with similar QoS experiences and predict their QoS requirements on web services accordingly. However, as the calculation of QoS values and user similarity require parameters which may contain privacy sensitive information, users may not trust the server that provides such third-party recommendations. In general, users are usually not willing to disclose such information to a third-party as it contains their tastes and preferences as well as experiences. Therefore the main challenge is to address the need for providing accurate web service recommendations to users while preserving their privacy from any third party server, as well as to protect the privacy of individual users from one another. To tackle this challenge, we propose a new protocol for privacy preserving web service recommendation where an untrusted recommendation server is able to provide the recommendation without disclosing any private information of individual users, and with negligible loss of accuracy of QoS values. We present both privacy and experimental analysis to verify that our proposed method is secure and efficient in terms of performance.

INDEX TERMS Privacy, search, homomorphic encryption, Web service, recommendation, QoS.

I. INTRODUCTION

Web service, also known as ''Web Inter-operable Machine to Machine Interaction'' in software systems, is a service offered by one electronic processing device to another electronic processing unit to support exchange of information between them over the internet [1]. Different kinds of application functionalities are encapsulated and accessible over the internet via the web service interface. On the other hand, QoS specifies the non-functional characteristics of web services which may be quantified; for example, response time, throughput and reliability. Based on the QoS values of different web services, it is possible to find high quality web services which have been providing good experiences to users.

It is a challenging task to help users to discover the preferred web services over the global internet in that their preferences and experiences tend to vary significantly. For example, it may not provide good service to some users if one decides the best web service by simply averaging the QoS experiences over all users. On the other hand, it is impractical to expect the users to be able to choose a particular web service with good QoS from a huge collection of web services. Recent studies have shown that CF-based techniques can help greatly in recommending suitable web services to users among a set of web services due to their ability to rank web services based on the QoS experiences of individual users.

There are several privacy issues while predicting QoS values or providing web service recommendations to users. The QoS values, which represent users' experiences on different web services, may contain personal information which are private to the users. Leakage of the QoS values might threaten the system's ability to provide creditable recommendations. Indeed, the recommendation system could be compromised by intruders, or by the server itself if it tries to manipulate the recommendation system by learning from users personal information stored in the server. Therefore it is necessary to

hide as much private information as possible from the server while allowing it to use the data needed for computing the recommendations.

To summarize, the privacy objectives of our proposed recommendation system are motivated by the following:

- Web service users usually rely on third-party recommendation systems to perform QoS evaluation of web services and make recommendation for them.
- The calculation of QoS values requires collection of data which are private to users and web service providers, hence it is desirable to preserve the privacy of such data when they are processed by the third party recommendation system.

Therefore the main challenge lies in how to protect these information while allowing the third-arty recommendation system to provide accurate service recommendations to users.

CONTRIBUTION

The main goal of our proposed solution is to predict missing QoS values, in a privacy preserving manner, for any particular user based on their past QoS experiences. We rank web services based on their predicted QoS values followed by recommending the ordered list of web services to any querying user. The process of ranking web services of their missing QoS values consists of two important steps: First, find similar users based on the querying user's QoS experiences; and second, make prediction based on similar users' QoS values. The main contribution of this proposed work is to preserve users privacy by encrypting the QoS values in such a manner that it allows the recommendation server to predict missing QoS values without revealing any private information to the server. Our scheme makes use of homomorphic encryption and search encryption techniques to achieve both requirements at the same time.

Another important contribution of this work is that, unlike other previous works [2], [3], we do not need to rely on a decryption server to decrypt the ciphertexts for querying users. All users in the system are able to share the secret key and, at the end of the protocol, users collaborate to decrypt the ciphertexts for the querying user. To perform secure operations while generating recommendations, we adopt the protocols described in [3] and [4] in our system by modifying them according to our requirements.

The main contributions of this paper can be summarized as follows:

1) We propose a privacy preserving web service recommendation protocol where the recommendation server is able to predict missing QoS values without learning any private information
2) We present a new protocol where the user space can be reduced by eliminating a set of users who are not similar to the querying user.
3) We present a secure protocol for recommendation where the decryption server is not required and users can share their secret keys to decrypt without revealing any private information.

II. RELATED WORK

In this section, we discuss and analyze existing approaches for web service recommendations, privacy issues in the area of service computing and privacy-preserving web service recommendations.

A. WEB SERVICE RECOMMENDATIONS

Substantial research works have been performed to recommend the preferred web service to users based on their personal preferences or previous experiences. Previous studies have used different types of techniques to improve their performances or scalability in web service recommendations. For example [5] and [6] used Content-based Filtering to find similarities among the features of web services. Based on those similar features, the system is able to find suitable web services for active users. In [7]–[9], Collaborative Filtering-based approaches were used for web service recommendations due to their simplicity and effectiveness. Reference [7] used user-based CF. References [8] and [9] used hybrid approaches by combining user and item-based CF. To improve the accuracy of CF-based web service recommendations, [10] and [11] proposed Matrix Factorization-based approaches to predict the missing QoS values in a user-web service matrix.

B. PRIVACY PRESERVING RECOMMENDATIONS USING HOMOMORPHIC ENCRYPTION

The main challenge with privacy preserving CF based approaches is the need to build secure protocol for adding and multiplying two secret values when computing similarity and generating predictions. In this subsection we describe how existing privacy preserving CF based approaches have dealt with this challenge.

Existing privacy preserving CF based approaches with homomorphic encryption can be categorized into two types: item-based CF ([12], [13]) and user-based CF ([3], [14]–[16]). The approach by Erkin *et al.* [12] secures the information by using Paillier encryption before they are used by CF. The approach by Badsha *et al.* [13] uses the ElGamal cryptosystem to secure users' information. This approach shows how individual public and secret keys from different users can be combined to encrypt and decrypt any message without compromising user privacy. Reference [14] proposed a CF-based recommendation system which uses secure multi-party computation and homomorphic encryption. This protocol allows the users and servers to jointly compute similarity and recommendations for target users. However, the main drawback of this approach is that each user gets the ciphertexts of other users' ratings during computation, hence privacy risk increases if some of the users collude with the decryption server in order to learn extra information. Kikuchi *et al.* [15] also proposed a cryptographic protocol for user-based CF where they also addressed the problem of multiplying two private data while generating recommendations. However, according to their protocol, users have to provide additional ciphertexts, which increases

the complexity into $O(n^2)$ by each user. To solve the issues with private multiplication operation in recommendation, [16] introduced the utilization of BGN cryptosystem without having additional ciphertext or losing any amount of privacy.

C. PRIVACY ISSUES IN SERVICE COMPUTING AND WEB SERVICE RECOMMENDATION

Most of the research work on solving privacy issues in the area of service computing focused on service selection and compositions. Reference [17] presented a privacy-aware composite service where the consumer provides her information to a service provider and the information will be used according to the consumer's privacy preferences. A privacy preserving framework for service matching was proposed by [18] where the querying users are able to send their queries privately and the service provider finds the results by matching search queries with web service attributes without knowing any private information. In [19], Tbahriti *et al.* proposed a privacy formal model in order to extend DaaS (Data as a Service) descriptions with privacy capabilities. The privacy model allows a service to define a privacy policy and a set of privacy requirements. Tbahriti *et al.* extended their privacy-preserving web service composition framework in [20], which deals with privacy not only at the data level (i.e. inputs and outputs) but also at the service level (i.e. service invocation). The first privacy-preserving web service recommendations framework was proposed in [21] where the QoS values were obfuscated before sending them to service providers. However the problem with this approach is that, due to the presence of noise, the service providers are not able to identify information accurately. However, they are still able to predict the missing QoS values and recommend a suitable web service to users. Recently we have developed a privacy preserving protocol for location aware web service recommendation based on homomorphic cryptography [22]. We preserved users QoS and locations privacy by means of homomorphic encryption and, at the same time, the protocol is able to predict missing QoSs by leveraging on past QoS experience and users' locations. Motivating from the above studies in privacy preserving web service recommendation [23], [24], i.e. the privacy issues in service computing especially in web service recommendations and privacy-preserving CF [2], [3], [13], [25], we propose a new framework to preserve users' privacy in web service recommendations.

III. PRELIMINARIES

A. NOTATIONS AND DEFINITIONS

The following defines notations to be used for the rest of the paper.

1) $i = 1, 2, \ldots, n$ is a set of users and $j = 1, 2, \ldots, m$ is a set of web services, where n and m represents total number of users and web services respectively. The Querying User who wants to get recommendations on unobserved web services, represented as u.
2) $r_{i,j}$ represents a QoS value which has been generated by the invocation of user i on web service j. $r_{i,j} = 0$ if the QoS is missing, which means the user i has not invoked the web service j.
3) $r_{u,.}$ and $r_{i,.}$ represents the whole QoS experience vector of user u and i respectively.
4) $l = 1, 2, \ldots, n'$ represents the set of users filtered after DRE based search scheme and n' represents the total number.
5) $P_{u,j}$ represents the prediction of missing QoS values for querying user u on web service j.
6) $E()_{DRE}$ and $E()_{HE}$ represent the encryption function by DRE based scheme and homomorphic scheme respectively.
7) PK represents the common public key of homomorphic encryption. $\{sk_l, pk_l\}$ and $\{sk_u, pk_u\}$ represent the individual secret and public key pairs of user l and u respectively.
8) r_l and r_u represent the random numbers used to encrypt the QoS values by user l and u respectively using homomorphic encryption of ElGamal.

B. CF-BASED RECOMMENDATION

The Recommendation Service Provider (RSP) runs the CF based recommendation once it completes the DRE based similar user search scheme and finds $l = 1, 2, \ldots, n'$ users from total n set of users where $i = 1, 2, \ldots, n$. To facilitate the recommendation process the querying user finds the cosine similarity between her and another user l as $s(u, l)$. To predict an unknown QoS for querying user u, the CF-based recommendation technique computes the QoS prediction according to the following expression, $P_{u,j}$ for user u on web service j

$$P_{u,j} = \frac{\sum_l r_{l,j} \cdot s(u,k)}{\sum_l s(u,l)} \tag{1}$$

where $P_{u,j}$ denotes the QoS prediction for querying user on web service j, $s(u, l)$ denotes the similarity between users u and l. The similarity function $s(u, k)$ between users u and l using Cosine similarity can be represented as follows:

$$\begin{aligned} s(u,l) &= \frac{\sum_{j=1}^{m} r_{u,j} r_{l,j}}{\sqrt{r_{u,1}^2 + \cdots + r_{u,m}^2}\sqrt{r_{l,1}^2 + \cdots + r_{l,m}^2}} \\ &= \sum_{j=1}^{m} R_{u,j} \cdot R_{l,j} \end{aligned} \tag{2}$$

where $R_{u,j} = \frac{r_{u,j}}{\sqrt{r_{u,1}^2+\cdots+r_{u,m}^2}}$, and $R_{l,j} = \frac{r_{l,j}}{\sqrt{r_{l,1}^2+\cdots+r_{l,m}^2}}$.

C. HOMOMORPHIC ENCRYPTION

In our privacy preserving protocol, we use ElGamal cryptosystem [26] to leverage their homomorphic properties while predicting the missing QoS values as well as other computations. The ElGamal encryption scheme is a probabilistic public key encryption algorithm which is composed of key generation, encryption and decryption. Specifically we have used the distributed version [27] of ElGamal cryptosystem

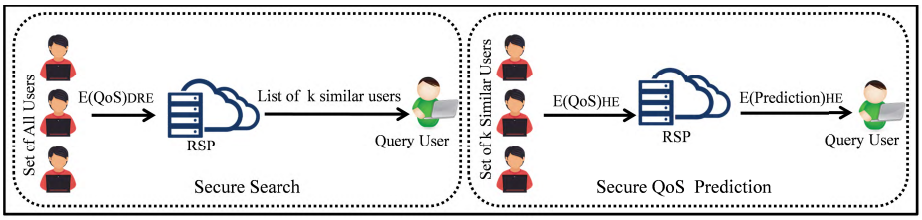


FIGURE 1. System model.

which supports both homomorphic addition and multiplication. For example, given two ciphertexts $C_1 = E_{pk}(m_1)$ and $C_2 = E_{pk}(m_2)$, the homomorphic addition is defined as follows:

$$D_{sk}(C_1 \cdot C_2) = m_1 + m_2 \tag{3}$$

The homomorphic multiplication can be defined as

$$D_{sk}((C_1)^{m_2}) = m_1 \cdot m_2 \tag{4}$$

where pk and sk represent the public and secret keys. It has been shown that ElGamal is semantically secure, i.e., it is computationally infeasible to distinguish between the encryptions of any two given messages, if the decisional Diffie-Hellman problem is intractable [28].

D. THE DISTANCE RECOVERABLE ENCRYPTION (DRE) SCHEME

In this subsection we describe the building blocks of DRE scheme [29] which is useful where a semi-trusted server can find the nearest neighbor of a querying user between two users based on their encrypted QoS experience. The DRE scheme have been using in many applications (searching nearby located users in online social network, secure image search in database, crowd-sourcing etc) to securely search the nearest data points of query point [30]–[32]. The DRE scheme is composed of four functions.

Key Generation $SK_{DRE} \leftarrow KGen(1^{\lambda}, R)$: The symmetric key generation algorithm $KGen$ takes a security parameter $\lambda \in N$ and a dimensional space R and outputs the symmetric key.

Encryption $C \leftarrow E_{DRE}(SK_{DRE}, n)$: The encryption algorithm takes SK_{DRE} and a m dimentional point $n \in R^n$ (R denotes the space), and outputs a ciphertext as an encrypted QoS vector.

Query generation $E(r_{u,.})_{DRE} \leftarrow QGen(SK_{DRE}, r_{u,.})$ The query generation algorithm takes the SK_{DRE} and query users QoS values to output encrypted query vector $E(r_{u,.})_{DRE}$

Distance Comparison: $C_l \leftarrow Cmp(E(r_{u,.})_{DRE}, C_i, C_k)$ The comparison algorithm takes an encrypted query vector of QoS $E(r_{u,.})$ and two encrypted QoS vectors from two other users C_i and C_k, outputs C_l for $l \in \{0, 1\}$ if

$$dist(E(r_{u,.}), E(r_{i,.})) \leq dist(E(r_{u,.}), E(r_{k,.})).$$

IV. SYSTEM MODEL

Our proposed system model consists of one server, i.e. the Recommendation Service Provider (RSP), for generating recommendations as well as a set of users. We assume that the RSP and the users in our system are honest but curious, they follow the protocol but try to learn the secret information. The RSP is maintained by a social networking service provider. Figure 1 shows the overall system model of our proposed framework in which the users send their encrypted QoS values to RSP. First the users encrypt the QoS values with DRE scheme and send them to RSP. A querying user who wants to get recommendations will send a query to RSP. The RSP first runs the DRE based search to find the set of users who are similar to the querying user based on their QoS experiences. The RSP does not learn any private information while running this protocol. This way the RSP reduces the user space for predicting the missing QoS values since the prediction protocol depends on the number of users, and recommendation accuracy can be improved by eliminating those users who are not similar to the querying user. After completing the DRE based search protocol, the RSP shares the pseudo identity of users to the querying user in the system. Then the users including the querying user form a group and generates public and secret key pairs of ElGamal cryptosystem. The users share their own public keys to generate common public key and use the common public key to encrypt their QoS values. Any user from this group can get recommendation privately anytime easily by sending query to RSP since they already formed a group and share their keys. They send their encrypted QoS values (homomorphically) to RSP in order to generate recommendation for the querying user. The querying user performs homomorphic computations to calculate encrypted similarity weight between her and other users. Then the querying user sends this encrypted similarity to RSP which holds the encrypted QoS by public key encryption. The RSP performs homomorphic operations to predict the missing QoS values for querying user and sends the encrypted result to her. Finally all users share their secret keys to decrypt the encrypted prediction. During this secret key sharing process and decryption, no user learns the secret key or any private information of other users. Moreover, by sharing the secret key for decryption, we can eliminate the need of another federated server for decryption.

V. DRE BASED SEARCH SCHEME

The users' QoS values are encrypted using DRE scheme to facilitate the encrypted search operations. Once another encrypted QoS is received from the querying user, the RSP determines for any two users, with one of which who is closer to the querying user in terms of encrypted QoS experiences. A DRE scheme is symmetric key encryption scheme with three main functions: key generation, encryption and search operation. In the key generation, the querying user generates the symmetric keys and shares them with other users via secure channel. In the encryption function, the users encrypt their QoS vectors and send them to RSP. In the search function, the RSP finds out which QoS experience vector is more similar to the querying user's experience vector among two other users. Let $i = 1, 2, \ldots, n$ represents the set of users and u represents the querying user in the system initially. The procedures are explained in greater detail below.

A. KEY GENERATION

The querying user generates the secret keys of DRE scheme as S, M_1 and M_2, where S is a vector with the size of $m+1$ bits, M_1 and M_2 are two invertible matrices with the size of $(m+1) \times (m+1)$ dimension. The querying user broadcasts these keys to all other users via any secure channel.

B. ENCRYPTION

Each user i modifies the QoS experience vectors as $r'_{i,.} = r_{i,1}, r_{i,2}, \ldots, r_{i,m}, ||r_{i,.}||^2$ where $||r_{i,.}||$ is the euclidean norm of $r_{i,.}$ of user i. Then the users split $r'_{i,.}$ into two random vectors $\{r_{ia,.}, r_{ib,.}\}$ according to S. The $r_{ia,ja}$ and $r_{ib,jb}$ are set equal to $r'_{i,j}$ if $S_j = 0$, otherwise $r_{ia,ja}$ and $r_{ib,jb}$ are set as two different random values where $r_{ia,ja} + r_{ib,jb} = r'_{i,j}$. Then the users encrypt the QoS experience vector as $E(r'_{i,.})_{DRE} = \{M_1^T\} r_{ia,.}, M_2^T\} r_{ib,.}$. All users send their encrypted QoS values to RSP.

The querying user also modifies the query vector $r'_{u,.} = (-2r_{u,1}, -2\, r_{u,2}, \ldots, -2r_{u,m}, 1)^T$ and split the $r'_{u,.}$ into two random vectors as $\{r_{ua,.}, r_{ub,.}\}$. The querying user sets $r_{ua,ja}$ and $r_{ub,jb}$ equal to $r'_{u,j}$ if $S_j = 1$, otherwise $r_{ua,ua}$ and $r_{ub,ub}$ are set as two different random values if $S_j = 0$ where $r_{ua,ua} + r_{ub,ub} = r'_{u,j}$.

Therefore the query vector is encrypted as $E(r'_{u,.})_{DRE} = \{\gamma M_1^{-1} r_{ua,.}, \gamma M_2^{-1} r_{ub,.}\}$. Finally the encrypted query QoS values are sent to the RSP.

C. SEARCH

The RSP performs search operations on DRE based encrypted QoS values between user u_q and u_i, and u_q and u_k as follows.

$$
\begin{aligned}
&E(r'_{u,.})_{DRE} \cdot E(r'_{i,.})^T_{DRE} \\
&= (\gamma M_1^{-1} r_{ua,.}) \cdot (M_1^T r_{ia,.})^T + (\gamma M_2^{-1} r_{ub,.}) \cdot (M_2^T r_{ib,.})^T \\
&= (\gamma r_{ua,.}) \cdot (r_{ia,.})^T + (\gamma r_{ub,.}) \cdot (r_{ib,.})^T \\
&= \gamma r_{u,.} \cdot r^T_{i,.} \\
&= \gamma(||r_{i,.}||^2 - 2 \sum_{j=1}^{j=m} r_{i,j} \cdot r_{u,j}) \\
&= \gamma(||r_{u,.} - r_{i,.}||^2 - ||r_{u,.}||^2)
\end{aligned}
$$

Note that the distance $d_{u,i} = ||r_{u,.} - r_{i,.}||^2$ is hidden by the secret γ and $||r_{u,.}||^2$ which is unknown. Therefore, we can derive which user among i and k is more similar in terms of QoS values by below equation.

$$E(r'_{u,.})_{DRE} \cdot E(r'_{i,.})^T_{DRE} > E(r'_{u,.})_{DRE} \cdot E(r'_{i,.})^T_{DRE}$$

which means that,

$$||r_{u,.} - r_{i,.}||^2 > ||r_{u,.} - r_{k,.}||^2$$

where i and k are two different users in the set of n users ($i \neq k$, $\{i, k \in n\}$) and user k is more similar to u than the user i.

After completing the search operation, RSP finds the set of users $l = 1, 2, \ldots, n'$ who are similar to the querying user u based on the search results. The querying user can set n' by her choice. In other words, RSP ranks the list of n users based on search results, and the querying user can select first n' users, where $n' < n$.

VI. COMPUTING USER SIMILARITY WEIGHT

A. SYSTEM MODEL

1) GROUP FORMATION

Once the querying user finds the n' similar users, all of them including the querying user register with a pseudonym and form a group. Let $l = 1, 2, \ldots, n'$ and $n'+1$ is total number of users including querying users in the group. Then the user u can invite the other users with the help of RSP to join the group to participate in the recommendation process. User l can accept or decline the invitation to join in the recommendation process. If user u believes that the total number of users n' is not enough, he/she can request RSP to find more similar users and invite them to join as explained.

2) KEY GENERATION

The members in the group are able to generate the secret and public keys of ElGamal cryptosystem. They generate their own public and secret keys ($\{pk_l, sk_l\}$, $\{pk_u, sk_u\}$)and combine the public keys to form a shared public key ($PK_{n'}$). Then any user l from set n' encrypts his/her QoS experiences values $r_{l,j}$ and $R_{l,j}$ using $PK_{n'}$. and sends $E(R_{l,j})_{HE}$ to user u, where $E(r_{l,j})_{HE}$ is sent to RSP. Then the querying user collaborates to perform homomorphic multiplication and additions to compute the similarity weight between her and another user l. This process is repeated for $l = 1, 2, \ldots, n'$ users to find the similarity between u and other n' users. The encrypted similarity $E(s(u, l))_{HE}$ is sent to RSP to predict the missing QoS values. The scheme for finding similarity between user u and user l homomorphically is shown below.

B. METHODOLOGY

1) STEP 1

Users u and l generate their own secret and public keys as,

$$sk_l = x_l, sk_u = x_u \quad and\ y_l = g^{x_l}, y_u = g^{x_u}$$

They combine their public keys to form a common public key as follows

$$Y = g^{x_l} \cdot g^{x_u} = g^{x_l + x_u}$$

The shared common public key is defined as $PK_{HE} = \{p, g, Y\}$ where p is a prime number and g is group generator. Any operation in the protocol is done with modulus p.

2) STEP 2

For any web service $j = 1, 2, \ldots, m$, user l encrypts using common public key as follows.

$$E(R_{l,j})_{HE} = \{g^{r_l}, g^{R_{l,j}} \cdot Y^{r_l}\}$$

where r_l denotes secret random number used to encrypt by user l.

3) STEP 3

The user l sends the ciphertext $E(R_{l,j})_{HE}$ to querying user. The querying user performs,

$$E(R_{l,j} \cdot R_{u,j})_{HE} = (E(R_{l,j})_{HE})^{R_{u,j}}$$

The user l repeats it for all web services j and sends them to query user. To compute the similarity, the query user computes as

$$E(s(u,l))_{HE} = \prod_{j=1}^{m} (E(R_{l,j})_{HE})^{R_{u,j}} \tag{5}$$

Where $s(u, l)$ denotes the similarity between user u and l. After computing the encrypted similarity, the querying user sends this ciphertext to RSP

Theorem 1: If user u and l follow the protocol we have that $E(s(u,l))_{HE} = \prod_{j=1}^{m} (E(R_{l,j})_{HE})^{R_{u,j}}$

Proof:

$$\begin{aligned}
&\prod_{j=1}^{m} (E(R_{l,j})_{HE})^{R_{u,j}} \\
&= \prod_{j=1}^{m} (g^{r_l}, g^{R_{l,j}} \cdot Y^{r_l})^{R_{u,j}} \\
&= \prod_{j=1}^{m} (g^{r_l (R_{u,j})}, g^{R_{l,j} \cdot R_{u,j}} \cdot Y^{r_l \cdot R_{u,j}}) \\
&= g^{r_l \sum R_{u,j}}, g^{\sum R_{l,j} \cdot R_{u,j}} \cdot Y^{r_l \cdot \sum R_{u,j}} \\
&= E(s(u,l))_{HE}
\end{aligned} \tag{6}$$

□

VII. PREDICTING MISSING QOS AND RANKING

A. SETTING

From the last section, RSP holds the encrypted similarity between querying user u and l as $E(s(u,l))_{HE}$ and the encrypted QoS values of user l as $E(r_{l,j})_{HE}$.

B. PROTOCOL DESCRIPTION

Recall the equation of web service prediction, we are interested in ranking the web services from the set of predicted QoS values for which the web services were not rated by the querying user. Therefore, we can ignore the denominator part of equation 1 and rewrite the equation as

$$P_{u,t} = \sum_{l=1}^{n'} r_{l,j} \cdot s(u,l) \tag{7}$$

where $t = 1, 2, \ldots, m'$ are the target web services for which we want to predict the QoS values. We now show the privacy preserving protocol for QoS prediction below. To facilitate this computation we adopt the protocol described in [3] and [4] where one party holds two homomorphically encrypted values $E(m_1)_{HE}$ and $E(m_2)_{HE}$, and another party who holds the secret key can collaborate to output $E(m_1 \cdot m_2)_{HE}$ without any party learning m_1 or m_2. In our case no party holds the secret key by itself but all users have their own secret keys and all of them need to collaborate to decrypt the ciphertext. Therefore we modify the original protocol [4] to adapt in our scenario so that (1) all parties can collaborate to share their secret keys (no private information or secret keys are disclosed) and (2) unlike the original protocol, we decrypt only one ciphertext among the two: $E(m_1)_{HE}$ and $E(m_2)_{HE}$. This way we can improve both the performance and security.

1) STEP 1

All users in the registered group encrypt and send the QoS values to RSP except the querying user.

$$E(r_{l,j})_{HE} = g^{r_l}, g^{r_{l,j}} \cdot Y^{r_l}$$

2) STEP 2

RSP selects two uniformly distributed random numbers r_{s1} and r_{s2} and performs homomorphic addition as follows.

$$\begin{aligned}
E(r^1_{l,j})_{HE} &= E(r_{l,j})_{HE} \cdot E(-r_{s1})_{HE} \\
&= g^{r_l - r_{s1}}, g^{r_{l,j} - r_{s1}} Y^{r_l - r_{s1}} \\
E(s^1(u,l))_{HE} &= E(s(u,l))_{HE} \cdot E(-r_{s2})_{HE} \\
&= g^{v - r_{s2}}, g^{s(u,l) - r_{s2}} Y^{v - r_{s1}}
\end{aligned}$$

where $v = r_l \sum R_{u,j}$, from equation 6. The RSP sends these ciphertexts to the querying user.

3) STEP 3

The querying user sends a notification request and broadcasts $A = g^{r_l - r_{s1}}$ and $B = g^{v - r_{s2}}$ to user l so that everybody can contribute to decrypt by sharing their secret keys. The protocol for sharing the secret keys privately and followed by decryption is described below.

4) STEP 4

The user l performs A^{x_l}, B^{x_l} and sends them to user u. The user u also performs A^{x_u} and B^{x_u} using his own secret key

and decrypts $E(s^1(u,l))_{HE}$ as follows (detail proof is shown theorem 2).

$$D(E(s^1(u,l)))_{HE} = s^1(u,l)$$

The user u performs $E(s^1(u,l) \cdot r^1_{u,l})_{HE} = E(r^1(l,j))^{s^1(u,l)}$ homomorphically and sends this back to RSP.

5) STEP 5

The server computes as follows.

$$\begin{aligned} &E(s^1(u,l) \cdot r^1_{l,j}) \cdot E(r_{l,j})^{r_{s2}} \cdot E(s_{u,l})^{r_{s1}} \cdot E(-r_{s1} \cdot r_{s2}) \\ &= E(s^1(u,l) \cdot r^1_{l,j} + r_{l,j} \cdot r_{s2} + s_{u,l} \cdot r_{s1} - r_{s1} r_{s2}) \\ &= E(s(u,l) \cdot r_{l,j}) \end{aligned}$$

6) STEP 6

To predict the missing QoS values, RSP performs,

$$E(P_{u,t}) = \prod_{l}^{n'} E(s(u,l) \cdot r_{l,j})$$

7) STEP 7

The server sends the ciphertext to the querying user u. Then querying user sends another notification to user l to share the secret key to decrypt the results (similar to step 4).

$$D(E(P_{u,t})) = \sum_{l}^{n'} s(u,l) \cdot r_{l,j}$$

Theorem 2: If user u and l follow the protocol, we have $D(E(s^1(u,l))) = s^1(u,l)$.

Proof: According to step 3 and 4, the querying user sends $B = g^{v-r_{s2}}$ to user l. User l computes B^{x_l} using its own secret key and sends back to u. At the mean time, user u also computes B^{x_u}. According to the decryption function using ElGamal cryptosystem, the querying user decrypts as follows:

$$\begin{aligned} D(E(s^1(u,l)))_{HE} &= \frac{g^{s(u,l)-r_{s2}} Y^{v-r_{s1}}}{B^{x_l} \cdot B^{x_u}} \\ &= \frac{g^{s(u,l)-r_{s2}} Y^{v-r_{s1}}}{(g^{v-r_{s2}})^{x_l} \cdot (g^{v-r_{s2}})^{x_u}} \\ &= \frac{g^{s(u,l)-r_{s2}} (g^{x_l+x_u})^{v-r_{s1}}}{(g^{v-r_{s2}})^{x_l} \cdot (g^{v-r_{s2}})^{x_u}} \\ &= \frac{g^{s(u,l)-r_{s2}} (g^{x_l+x_u})^{v-r_{s1}}}{(g^{x_l+x_u})^{v-r_{s1}}} \\ &= g^{s(u,l)-r_{s2}} = g^{s^1(u,l)} \end{aligned}$$

□

To retrieve $s^1(u,l)$ from $g^{s^1(u,l)}$, the querying user computes discrete logarithm as $log_g^{(D(E(s^1(u,l)))_{HE}) = s^1(u,l)}$.

*Remark:*The size of $s^1(u,l)$ is not large, therefore computing discrete logarithm is not hard.

Theorem 3: If users u and l follow the protocol, we have $D(E(P_{u,t})) = \sum_{l}^{n'} s(u,l) \cdot r_{l,j}$.

Proof: From homomorphic properties, we have:

$$\begin{aligned} &E(s^1(u,l) \cdot r^1_{u,l})_{HE} \\ &\quad = E(r^1(l,j))^{s^1(u,l)} \\ &\quad = g^{s^1(u,l)(r_l - r_{s1})}, g^{s^1(u,l)(r_{l,j} - r_{s1})} Y^{s^1(u,l)(r_l - r_{s1})} \\ &(E(r^1_{l,j})_{HE})^{r_{s1}} \\ &\quad = g^{r_{s2} \cdot (r_l - r_{s1})}, g^{r_{s2}(r_{l,j} - r_{s1})} Y^{r_{s2}(r_l - r_{s1})} \\ &(E(s^1(u,l))_{HE})^{r_{s2}} \\ &\quad = g^{r_{s2}(p - r_{s2} r_{s2})}, g^{r_{s2}(s(u,l) - r_{s2})} Y^{r_{s2}(p - r_{s1})} \\ &(E(-r_{s1} r_{s2})_{HE}) \\ &\quad = g^{r_{s3}}, g^{-r_{s1} r_{s2}} Y^{r_{s3}} \end{aligned}$$

The encrypted prediction is computed by

$$\begin{aligned} E(P_{u,t}) &= \prod_{l}^{n'} E(s(u,l) \cdot r_{l,j}) \\ &= \prod_{l}^{n'} (E(s^1(u,l) \cdot r^1_{l,j}) \cdot E(r_{l,j})^{r_{s2}} \\ &\quad \cdot E(s_{u,l})^{r_{s1}} \cdot E(-r_{s1} \cdot r_{s2})) \\ &= (g^X, g^{\sum_l s^1(u,l) \cdot r^1_{l,j} + r_{l,j} \cdot r_{s2} + s_{u,l} \cdot r_{s1} - r_{s1} r_{s2}} Y^X) \end{aligned}$$

where $X = \sum_l (s^1(u,l)(r_l - r_{s1}))(r_{s2} \cdot (r_l - r_{s1}))(r_{s2}(v - r_{s2} r_{s2}))(r_{s3})$.

The querying user sends g^X to user l. The querying user and user l perform $(g^X)^{x_u}$ and $(g^X)^{x_l}$ respectively, and $(g^X)^{x_l}$ is received by the querying user. User u decrypts as follows.

$$\begin{aligned} &D(E(P_{u,t})) \\ &= \frac{g^{\sum_l s^1(u,l) \cdot r^1_{l,j} + r_{l,j} \cdot r_{s2} + s_{u,l} \cdot r_{s1} - r_{s1} r_{s2}} Y^X)}{(g^X)^{x_l} \cdot (g^X)^{x_u}} \\ &= \frac{g^{\sum_l s^1(u,l) \cdot r^1_{l,j} + r_{l,j} \cdot r_{s2} + s_{u,l} \cdot r_{s1} - r_{s1} r_{s2}} (g^{x_l + x_u})^X}{(g^X)^{x_l + x_u}} \\ &= g^{\sum_l^{n'} s(u,l) \cdot r_{l,j}} \end{aligned}$$

□

To retrieve $\sum_l^{n'} s(u,l) \cdot r_{l,j}$, the querying user computes discrete logarithm as $log_g^{\sum_l^{n'} s(u,l) \cdot r_{l,j}} = \sum_l^{n'} s(u,l) \cdot r_{l,j}$.

VIII. PRIVACY ANALYSIS

In this section we analyze the privacy of secure search scheme using DRE as well as secure similarity and web service ranking protocols. The goal is to preserve users' privacy from server and any third party intruder.

According to our system the m has to be sufficiently large. For example RSA keys are required to be at least 1024 bits. It is shown that 1024 bits RSA keys are as strong as 80-bit symmetric keys [29]. In our protocol we set m to 100 which is sufficiently large. Moreover as analysed by [29] and [31] we also split the QoS rating vectors. We choose a secret configuration of vector bits $S_1, S_2, \ldots, S_j, \ldots, S_m$, where $S_j = \{0, 1\}$. If $S_{i,j} = 1$, we split r'_{uj} into $r_{ua,ja}$ and $r_{ub,jb}$ and set $r_{ua,ja} = r_{ub,jb}$. Otherwise we set two random values

such that $r_{ua,ja} + r_{ub,jb} = r'_{uj}$. This configuration is secretly shared among the users in the group. Since the configuration is unknown to server, there are in total 2^m possible choices and therefore the scheme is 2^m costly for the server to break the system.

The similarity and QoS prediction protocol depends on the semantic security of homomorphic encryptions. At the beginning all users encrypt their QoS vectors using common public key Y. Note that, each user shares their own g^x to form the Y without disclosing the secret key x. The ciphertexts are sent and stored in the server. By using the common public key, it is not possible to decrypt the ciphertext by any single user unless they all collude with one another. That means we need $g^{x_l+x_u}$ for both users u and l to decrypt the ciphertext and it is secure as long as at least one user is honest amongst the n users. Therefore it is not possible for the server or any user to decrypt the ciphertext without sharing all of their secret keys. In the similarity protocol, the querying user received the ciphertext of $R_{l,j}$ from other user l to calculate the similarity homomorphically. This protocol does not disclose any plaintext or private information of other users to querying user while computing similarity.

Then the server performs $E(s^1(u,l) \cdot r^1_{l,j}) \cdot E(r_{l,j})^{r_{s2}} \cdot E(s_{u,l})^{r_{s1}} \cdot E(-r_{s1} \cdot r_{s2})$ to get $E(s(u,l) \cdot r_{l,j})$ and finally it performs $\prod_l^{n'} E(s(u,l) \cdot r_{l,j})$ to get the encrypted prediction or missing QoS. Note that, the server uses homomorphic operations and none of the value was in plaintext or disclosed to server or any other users. Once the querying user receives the encrypted prediction, all other users are notified and share their secret keys as $(g^X)^{x_l}$. Since x_l is of 1024 bits, its very hard to perform discrete log to find x_l from this value. This way the querying user combines $(g^X)^{x_u}$ with $(g^X)^{x_l}$ as $(g^X)^{x_l+x_u}$ by multiplying these two values and decrypts the ciphertext of predicted QoS value. Therefore, the decryption is done without revealing any private information of any user.

IX. PERFORMANCE ANALYSIS

We have conducted a set of experiments to evaluate the performance of our protocol. Our experimental analysis is organized as follows. First, we analyze the computation complexity for each stage of our protocol. Second, we present the performance of our method on publicly available dataset WSDREAM-datatset-1 [33], which contains the QoS records of service invocations on 5825 web services from 339 users. Note that, in our experiment, we only use RTT (response time) to test the performance of our protocol. More specifically, we address and analyze the following points.

- The computation complexity of the overall protocol.
- Performance evaluation of the protocol in terms of secure symmetric DRE scheme to search similar users, privacy preserving similarity and recommendation.
- Scalability of different scheme in terms of increasing the number of users and web services.

To conduct the experiments, we use Java 2 SE 8 including cryptographic libraries on a hardware platform with OS windows 7, 64 bit and 3.6 GHz- core i7 and 8GB CPU unit. For the performance measurement, the metrics we considered in our experiment are shown in table 1.

TABLE 1. Notations and descriptions.

Notations	Description
e	Modular exponentiation
m_d	Modular multiplication
d_e	Decryption using Paillier
m'	Number of web services that user has not invoked before
m_d	Modular multiplication
d_l	Discrete log operation

A. COMPUTATION COMPLEXITY

We present the computational complexities of three different protocols below: DRE based similar user search, user similarity computation and web service ranking. Table 2 shows the overall complexities of each protocols.

1) DRE BASED SEARCH

This protocol consists of mainly two stages: key generation and secure search operation. The overall complexity of key generation is $O(m+1) \times O(m+1)$ for generating two $m+1$ invertible matrices, $O(m+1)$ for generating a vector S and $O(m+1)$ for generating random numbers, where m is the total number of web services. The DRE search scheme search through $O(n)$ users to rank the users based on the euclidean distance between the querying user and other users' QoS experiences.

2) SIMILARITY COMPUTATION

To compute the similarities between the querying user and other users, the protocol allows all users to encrypt their QoS values $r_{u,j}$ and $R_{u,j}$, and send the encrypted results to RSP. As the ElGamal cryptosystem consists of two ciphertexts created by performing two modular exponentiation operations and one multiplication, the total time to encrypt $r_{u,j}$ and $R_{u,j}$ for m web services becomes $4e + 2m_d$. To compute the similarities, the querying user performs m modular exponentiation operations for m services, and finally performs $m-1$ modular multiplications to homomorphically add the resultant ciphertexts to get the additions in plaintexts. This computations takes total $mne + (m-1)m_d$ seconds.

3) WEB SERVICE RANKING

To get the web service recommendations, RSP predicts the missing QoS values and sends the encrypted results to the querying user. The RSP holds the ciphertexts of the QoS values and similarity between u and l as $s(u,l)$.According to step 2, the RSP computes $E(r^1_{l,j})$ and $E(s^1(u,l))$ which take two modular multiplications i.e, $2m_d$ seconds.

The user l performs one modular exponentiation to perform A^{x_l} and B^{x_l} which takes $2e$ seconds in step 3. To perform the decryption of $E(s^1(u,l))$ in step 4, the querying user performs $(B)^{x_l} \cdot (B)^{x_u}$ (one modular multiplication) and a discrete logarithm which takes total $m_d + \sqrt{T}e$ seconds, since one discrete

TABLE 2. Computation complexity and time.

	DRE Scheme	Similarity Computations	Web Service Ranking
		Computation Complexity	
User l	$(m+1)E_{dre}$	$2(2e+m_d)m$	e
User u	$(m+1)E_{dre}$	$mn'e+(m-1)m_d$	$e+(m_d+\sqrt{T}e)n'$
Server	n		$mn'm_d+n'm_d+(3m_d+2e)n'+(n'-1)m_d$
		Required Time ($n=300, m=100, n'=30$)	
User l	$.001s$	$.009s$	$2\times10^{-5}s$
User u	$.001s$	$.06s$	$.012s$
Server	0.48		$.019s$

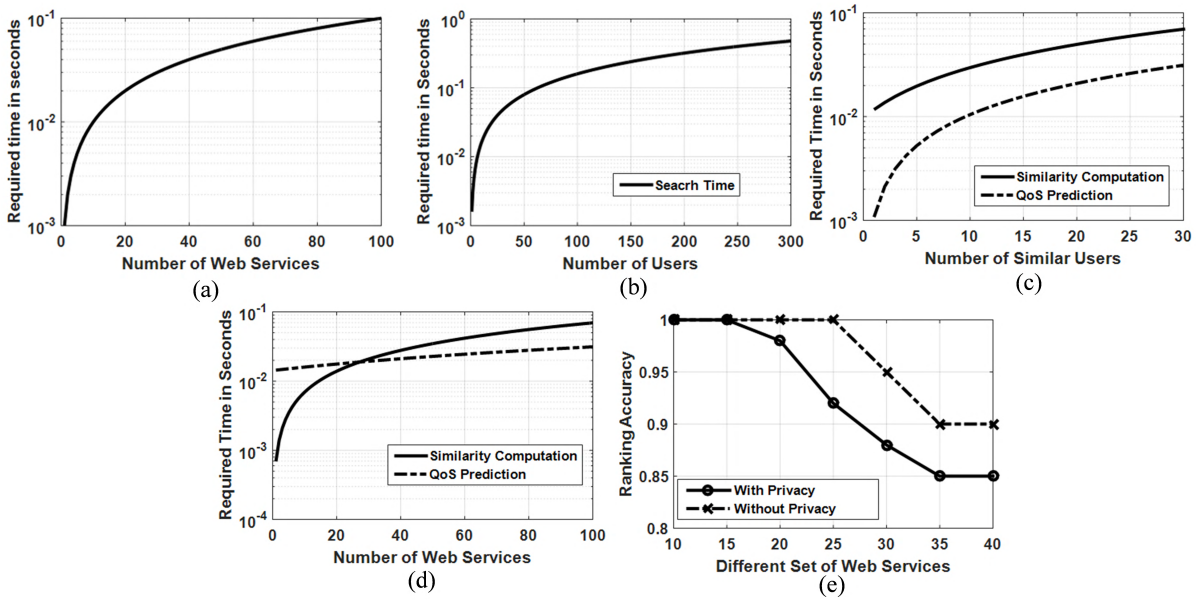


FIGURE 2. (a)Required time of DRE based encryption and its effect on number of web services for each user ($n=300$) (b) Required time of nearest neighbor search by DRE scheme and its effect on increasing number of users ($m=100$) (c)Required time for Similarity Computation and QoS Prediction Scheme by increasing the number of web services ($n'=30$) (d) Required time for Similarity Computation and QoS Prediction Scheme by increasing the number of similar users ($n'=30$) (e) Web service ranking accuracy in terms of different set of web services ($t=\{10,15,20,25,30,35,40\}$).

logarithm takes $\sqrt{T}e$ seconds. Also the user performs one modular exponentiation e and sends the ciphertexts to RSP. In this step the querying user takes $m_d+\sqrt{T}e+e$ seconds in total.

In step 5, the server performs three modular multiplications $3m_d$ and two modular exponentiation $2e$, which takes total $3m_d+2e$ seconds. Finally, to predict the missing QoS values, the RSP has to perform $n'-1$ modular multiplications which takes $m_d(n'-1)$ seconds.

B. EFFICIENCY

In this section we present the required time of DRE scheme in terms of encryption and secret search operations. To test the performance of our proposed scheme we reduce the dataset into 300 users and 100 web services. That means initially we have $n=300$ and $m=100$. Figure 2(a) shows the scalability of encryption in terms of the number of web services. Since the users can perform the encryptions in parallel, in our protocol the encryption does not depend on the number of users, but the number of web services. This figure shows the scalability for up to 100 web services. Figure 2(b) shows the effect of nearest neighbor search in terms of increasing total number of users up to 300, where the number of web services is fixed at 100. The search process uses a simple linear algorithm by using a heap structure to maintain the n' closest points to the query. In our experiment we choose $n'=30$. From figure 2(a) and 2(b) it is clear that the DRE scheme performs linearly to the number of web services and users in terms of encryption and nearest neighbor search respectively.

Figure 2(c) shows the required time for similarity computation and QoS prediction schemes by varying the number of web services and fixing the number of users to 30. On the other side, the figure 2(d) shows the required time of both QoS prediction and similarity computation schemes while varying the number of users and fixing the number of web services. For figures 2(c) and 2(d) it is clear that similarity computation takes longer time than QoS predictions except in the case of varying the number of web services. For very small number of web services, the similarity computation is

faster than QoS prediction scheme but it gradually increases with increasing the number of web services. The reason of consuming higher computation cost of similarity scheme is that user u has to perform m modular exponentiation and the results are being multiplied as modular multiplication which is of $m-1$ times. Therefore it is clear that the protocol depends on the number of web services m and in our case the m is higher than the total number of similar users $n' = 30$.

Table 2 shows required time of our protocol in terms of different schemes, users and the server. Initially the user has to perform very little computations in DRE scheme which takes 0.001 seconds for each user. The server takes only 0.48 seconds to complete the DRE scheme and to find 30 nearest neighbor users. The users are involved in similarity computations which take 0.009 seconds and 0.06 seconds for user l and user u respectively. The main reason to consume more time for user u than user l is that, most of the computation is done by the querying user side. For QoS prediction scheme the server takes on 0.019 seconds to predict one QoS values where the number of nearest users is 30.

C. RECOMMENDATION ACCURACY

We test the accuracy of proposed privacy preserving web service recommendation protocol using spearman correlation coefficient. Randomly 10 querying users are chosen to predict missing QoS for them and we run the prediction on different set of web services. Based on the prediction result, we rank the web services as recommendation. We repeat the process for 10 querying users and take the average result. We run same experiment two times: as privacy preserving protocol and without privacy preserving to check how the privacy preserving system differs from the original one. We choose the web services which are already invoked by the querying users, that means the test web services have the QoS values corresponding to the querying user. Concretely, We randomly select 10, 15, 20, 25, 30, 35, 40 web services for one querying user and rank them first based on available QoS. Then we run the proposed privacy preserving recommendation and rank the web services for those 6 different sets. Then we compare the ranking with the available QoS based ranking. We also run the protocol without privacy functionalities to get the rankings and compare it with available one. Figure 2(e) shows the such comparison where we found that for both cases (with and without privacy functionalities) the system gives almost same accuracy and the loss of accuracy of privacy preserving protocol is very low. It starts to degrade when the number of web services is more than 20. Note that we did not contribute to modify or to improve the recommendation system itself but only privacy preservation of existing recommendation technique. However, our analysis of recommendation accuracy shows that there was no significant loss of accuracy while preserving the privacy of recommendations.

X. CONCLUSION

In this work, we proposed a secure web service recommendation protocol which is composed of three sub protocols: secure DRE based search, secure similarity computation and secure QoS prediction followed by recommendations. Finding the similar user based a threshold value in a privacy preserving manner is time consuming and inefficient using homomorphic encryption. To overcome this situation we have introduced the DRE based search scheme into web service recommendation to find n' nearest users based on their QoS experience which reduces the total user space from n to n', where $n' < n$. Then we perform privacy preserving cosine similarity computations on the reduced space of users n'. Therefore we do not need to perform secure comparison protocol but at the same time we are able to find similarity weight of the similar users. The system output of DRE scheme which gives the set of similar users can be used in other different services rather than only web service recommendations where we do not need to calculate similarity weight. For example if any querying user wants to find the places which were visited by other users with similar preferences. In this kind of scenario we do not need to calculate the cosine similarity and the protocol would be more efficient in terms of computation complexities. Therefore our protocol has lots of potentiality to use in different privacy preserving online systems. The main limitation of this work is the RSP needs to search over all users to find the similar set of users based on their QoS experiences. In future, we intend to develop encrypted searchable index tree so that the search space is reduced.

REFERENCES

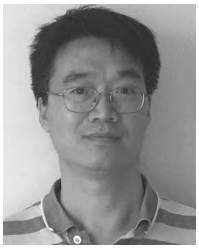
[1] L.-J. Zhang, H. Cai, and J. Zhang, *Services Computing*. Springer, 2007.
[2] S. Badsha, X. Yi, I. Khalil, D. Liu, S. Nepal, and E. Bertino, "Privacy preserving location recommendations," in *Proc. 18th Int. Conf. Web Inf. Syst. Eng. (WISE)*, Puschino, Russia, Oct. 2017, pp. 502–516.
[3] Z. Erkin, T. Veugen, T. Toft, and R. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1053–1066, Jun. 2012.
[4] R. Cramer, I. Damgård, and J. B. Nielsen, "Multiparty computation from threshold homomorphic encryption," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. Adv. Cryptol. (EUROCRYPT)*, Innsbruck, Austria, May 2001, pp. 280–299.
[5] G. Kang, J. Liu, M. Tang, X. Liu, B. Cao, and Y. Xu, "AWSR: Active Web service recommendation based on usage history," in *Proc. IEEE 19th Int. Conf. Web Services (ICWS)*, Jun. 2012, pp. 186–193.
[6] L. Liu, F. Lecue, and N. Mehandjiev, "Semantic content-based recommendation of software services using context," *ACM Trans. Web*, vol. 7, no. 3, p. 17, 2013.
[7] L. Shao, J. Zhang, Y. Wei, J. Zhao, B. Xie, and H. Mei, "Personalized QoS prediction for Web services via collaborative filtering," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2007, pp. 439–446.
[8] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "WSRec: A collaborative filtering based Web service recommender system," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2009, pp. 437–444.
[9] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware Web service recommendation by collaborative filtering," *IEEE Trans. Services Comput.*, vol. 4, no. 2, pp. 140–152, Apr./Jun. 2011.
[10] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "Collaborative Web service QoS prediction via neighborhood integrated matrix factorization," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 289–299, Jul. 2012.
[11] Q. Yu, Z. Zheng, and H. Wang, "Trace norm regularized matrix factorization for service recommendation," in *Proc. IEEE 20th Int. Conf. Web Services (ICWS)*, Jun./Jul. 2013, pp. 34–41.
[12] Z. Erkin, M. Beye, T. Veugen, and R. Lagendijk, "Privacy-preserving content-based recommendations through homomorphic encryption," in *Proc. Inf. Theory Benelux, 2nd Joint WIC/IEEE Symp. Inf. Theory Signal Process. Benelux*, 2012, p. 71.

[13] S. Badsha, X. Yi, and I. Khalil, "A practical privacy-preserving recommender system," *Data Sci. Eng.*, vol. 1, no. 3, pp. 161–177, 2016.
[14] Z. Erkin, M. Beye, T. Veugen, and R. L. Lagendijk, "Privacy enhanced recommender system," in *Proc. 31st WIC Symp. Inf. Theory Benelux*, 2010, pp. 35–42.
[15] H. Kikuchi, H. Kizawa, and M. Tada, "Privacy-preserving collaborative filtering schemes," in *Proc. Int. Conf. Availability, Rel. Secur.*, 2009, pp. 911–916.
[16] S. Badsha, X. Yi, I. Khalil, and E. Bertino, "Privacy preserving user-based recommender system," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 1074–1083.
[17] W. Xu, V. N. Venkatakrishnan, R. Sekar, and I. V. Ramakrishnan, "A framework for building privacy-conscious composite Web services," in *Proc. Int. Conf. Web Services (ICWS)*, 2006, pp. 655–662.
[18] A. Squicciarini, B. Carminati, and S. Karumanchi, "A privacy-preserving approach for Web service selection and provisioning," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2011, pp. 33–40.
[19] S.-E. Tbahriti, M. Mrissa, B. Medjahed, C. Ghedira, M. Barhamgi, and J. Fayn, "Privacy-aware DaaS services composition," in *Proc. 22nd Int. Conf. Database Expert Syst. Appl. (DEXA)*, Toulouse, France, Aug./Sep. 2011, pp. 202–216.
[20] S.-E. Tbahriti, C. Ghedira, B. Medjahed, and M. Mrissa, "Privacy-enhanced Web service composition," *IEEE Trans. Services Comput.*, vol. 7, no. 2, pp. 210–222, Apr. 2014.
[21] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "A privacy-preserving QoS prediction framework for Web service recommendation," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun./Jul. 2015, pp. 241–248.
[22] S. Badsha *et al.*, "Privacy preserving location-aware personalized Web service recommendations," *IEEE Trans. Services Comput.*, 2018, 10.1109/TSC.2018.2839587.
[23] C. Yan, X. Cui, L. Qi, X. Xu, and X. Zhang, "Privacy-aware data publishing and integration for collaborative service recommendation," *IEEE Access*, vol. 6, pp. 43021–43028, 2018.
[24] L. Qi, X. Zhang, W. Dou, and Q. Ni, "A distributed locality-sensitive hashing-based approach for cloud service recommendation from multi-source data," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2616–2624, Nov. 2017.
[25] S. Badsha, X. Yi, I. Khalil, and A. Kelarev, "Private recommendations generation for vertically partitioned datasets," in *Proc. 21st Pacific Asia Conf. Inf. Syst. (PACIS)*, Langkawi, Malaysia, Jul. 2017, p. 163.
[26] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.
[27] F. Brandt, "Efficient cryptographic protocol design based on distributed El Gamal encryption," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Springer, 2005, pp. 32–47.
[28] X. Yi, R. Paulet, and E. Bertino, *Homomorphic Encryption and Applications*. Springer, 2014.
[29] W. K. Wong, D. W.-L. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2009, pp. 139–152.
[30] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2594–2608, Nov. 2016.
[31] R. Schlegel, C.-Y. Chow, Q. Huang, and D. S. Wong, "Privacy-preserving location sharing services for social networks," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 811–825, Sep./Oct. 2017.
[32] J. Shu, X. Jia, K. Yang, and H. Wang, "Privacy-preserving task recommendation services for crowdsourcing," *IEEE Trans. Services Comput.*, 2018.
[33] Z. Zheng, Y. Zhang, and M. R. Lyu, "Distributed QoS evaluation for real-world Web services," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2010, pp. 83–90.

SHAHRIAR BADSHA received the M.Eng.Sc degree from the University of Malaya, Kuala Lumpur, Malaysia, in 2014. He is currently pursuing the Ph.D. degree with the Computer Science and Software Engineering Department, School of Science, RMIT University, Australia. He is also with Data61, CSIRO, Melbourne, Australia. His research interests include privacy-preserving applications and applied cryptography.

XUN YI is currently a Professor with the Computer Science and Software Engineering Department, School of Science, RMIT University, Australia. He has published more than 160 research papers in international journals and conference proceedings. Recently, he has led some Australia Research Council Discovery Projects in data privacy protection. His research interests include applied cryptography, computer and network security, mobile and wireless communication security, and data privacy protection. He has ever undertaken program committee members for more than 30 international conferences. Since 2014, he has been an Associate Editor of the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING.

IBRAHIM KHALIL received the Ph.D. degree from the University of Berne, Switzerland, in 2003. He is currently an Associate Professor with the Computer Science and Software Engineering Department, RMIT University, Melbourne, Australia. He has several years of experience in Silicon Valley-based companies working on large network provisioning and management software. His research interests are in scalable efficient computing in distributed systems, network and data security, and secure data analysis, including big data security.



DONGXI LIU received the B.E. and M.E. degrees from the Taiyuan University of Technology in 1996 and 1999, respectively, and the Ph.D. degree in computer science and engineering from Shanghai Jiao Tong University in 2003. He is currently a Research Scientist at the CSIRO ICT Center, Australia. His research interests include trusted computing and security and privacy in collaborative environments.

SURYA NEPAL is currently a Principal Research Scientist at Data61, CSIRO, Australia. At CSIRO, he undertook research in the area of multimedia databases, Web services, and service-oriented architectures, security, privacy, and trust in collaborative environment. He has more than 150 publications to his credit. His main research interests include the development and implementation of technologies in the area of distributed systems and social networks, with a specific focus on security, privacy, and trust.

KWOK-YAN LAM received the B.Sc. degree (Hons.) from the University of London, London, U.K., in 1987, and the Ph.D. degree from the University of Cambridge, Cambridge, U.K., in 1990. In 1997, he founded PrivyLink International Ltd., a spinoff company of the National University of Singapore, specializing in e-security technologies for homeland security and financial systems. In 2012, he co-founded Soda Pte Ltd., which received the Most Innovative Start Up Award at the RSA 2015 Conference. He is currently a Professor of computer science with Nanyang Technological University, Singapore. He is also the Lead PI of the SPIRIT Programme, a $11 000 000 programme on smart nation research funded by NRF.