Cluster Computing (2019) 22:S1655–S1663
https://doi.org/10.1007/s10586-017-1649-y

Cloud-assisted privacy-preserving profile-matching scheme under multiple keys in mobile social network

Chong-zhi Gao[1,2] · Qiong Cheng[1] · Xuan Li[3] · Shi-bing Xia[1]

Received: 7 November 2017 / Revised: 13 December 2017 / Accepted: 26 December 2017 / Published online: 2 February 2018

Abstract
Making new friends by measuring the proximity of people's profile is a crucial service in mobile social networks. With the rapid development of cloud computing, outsourcing computing and storage to the cloud is now an effective way to relieve the heavy burden on users for managing and processing data. To prevent privacy leakage, data owners tend to encrypt their private data before outsourcing. However, current solutions either have heavy interactions or require users to encrypt private data with a single key. In this paper, we propose a novel cloud-assisted privacy-preserving profile-matching scheme under multiple keys based on a proxy re-encryption scheme with additive homomorphism. Our scheme is secure under the honest-but-curious (HBC) model given two non-colluding cloud servers.

Keywords Mobile social networks · Cloud computing · Privacy-preserving · Multiple keys

1 Introduction

Given the popularity of smart phones, mobile terminals and other mobile devices, the use of mobile devices to access social networks has become mainstream [1]. Some social networks for instance, LinkedIn, Facebook and MySpace have been very prominent and are now the preferred way of communication for many people [2]. Mobile social networks (MSNs) allow mobile users to discover and interact with potential friends. Increasingly more people are beginning to pay attention to the task of looking for a potential new friend with similar interests. Profile matching is the most cost-effective method of measuring the proximity to users' personal profiles. However, one's personal profile may contain sensitive information, and users do not want to reveal their private data. In addition, current profile-matching schemes [3–6] have high computational overhead and are not suitable for mobile devices, which have low computational resources.

One way to address the issue of high computational overhead is to take advantage of cloud computing. The cloud promises to provide massively scalable data storage and powerful computation services to society at a reduced cost [7]. With the rapid development of cloud computing [8], outsourcing computing and storage to the cloud is an effective way to relieve the heavy burden on users of managing and processing data. However, trivially moving the computation to the cloud will lead to privacy leakage [9]. To prevent privacy leakage, data owners tend to encrypt their private data before outsourcing. However, current solutions either have heavy interactions or require users to encrypt private data under a single key. In this paper, we propose a novel cloud-assisted privacy-preserving profile-matching scheme under multiple keys based on a proxy re-encryption scheme with additive homomorphism. The cloud environment is composed of two cloud servers. Our scheme does not require users to be online at the same time. Users only need to encrypt their personal profiles and send them to one cloud server and receive matching results. The two cloud servers perform most

✉ Chong-zhi Gao
czgao@gzhu.edu.cn

Qiong Cheng
chengqiong@e.gzhu.edu.cn

Xuan Li
jessieli24@fjnu.edu.cn

Shi-bing Xia
xsb_summer@163.com

[1] School of Computer Science, Guangzhou University, Guangzhou 510006, China

[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing, China

[3] School of Software, Fujian Normal University, Fuzhou 350117, China

of the computations in our scheme, effectively reducing the user's computational burden and ensuring that the user's private information is not leaked to the clouds. Our scheme is secure under the honest-but-curious (HBC) model assuming that two cloud servers do not collude.

1.1 Related work

With the fast development of cloud computing, Internet of things and smart grids, more and more data are being produced and analyzed, leading to a new big data era [10,11]. However, the risk of the data being revealed or disclosed makes it urgent to enhance the security and privacy of users' data. In the literature, privacy protection issues have been studied in various fields, such as Internet of things [12,13], online social networks [14], smart grids [15], and cloud computing [16]. Negi et al. [16] proposed a modification to the confidence based filtering method (CBF) which is investigated for cloud computing environment based on correlation pattern to mitigate distributed denial of service attacks on cloud. In [12], Stergiou et al. combined the cloud computing and Internet of things in order to examine the common features, and in order to discover the benefits of their integration.

A number of research works have been conducted on protecting a user's privacy during the process of profile matching. Previous work focusing on privacy-preserving profile matching can be broadly divided into two categories. The first approach is the coarse-grained private matching approach. In this approach, social proximity is defined as a set intersection or the cardinality of a set intersection of two users' attribute sets, [4–6,17] are coarse-grained private matching schemes; they cannot further differentiate users with different degrees of attributes. The other approach is the fine-grained private matching approach. In this approach, social proximity is defined as the dot product between two users' vectors, [3,18,19] are fine-grained private matching schemes; they enable finer differentiation among users having different degrees of interest in the same attribute.

With the rapidly increasing ability to store and handle personal data, the problem of protecting privacy in cloud computing has become more important in recent years. To achieve secure data processing in the cloud, many schemes based on various techniques have been proposed, including partially homomorphic encryption (PHE) [20], fully homomorphic encryption (FHE) [21–24] and secure multiparty computation [25]. Fully homomorphic encryption has high computational overhead. Although it supports homomorphic computations over ciphertext, which are encrypted with a single key, it is not suitable for multi-user systems. Secure multiparty computation always requires heavy interactions and is not suitable for outsourcing situations. Compared with schemes employing FHE, PHE has lower computational overhead. To meet certain special security requirements, many schemes based on encryption schemes with certain properties have been proposed such as deduplication schemes [20,26,27], identity-based encryption schemes [28], and attribute-based encryption schemes [29].

Recently, secure data processing in the cloud under multiple keys has become an important area of research. López et al. [30] proposed an FHE under multiple keys, but a large amount of interaction between users is required during the decryption of the final result. Liu et al. [31] presented a distributed public-key cryptosystem with double trapdoors (DT-PKC) to realize privacy-preserving outsourced calculation. DT-PKC is deployed to split a strong private key into different shares. This scheme requires many interactions between the cloud and the server, who provides the computing service. Peter et al. [32] proposed a novel technique based on additively homomorphic encryption. They extensively utilized the BCP cryptosystem [33], which is additively homomorphic and offers two independent decryption mechanisms. However, this scheme requires heavy interactions between servers when transforming the ciphertexts of multiple keys into a single key. In addition, an efficient and secure data sharing framework has been proposed [34] using homomorphic encryption and proxy re-encryption schemes. Rong et al. [35] proposed an outsourced privacy-preserving scalar product protocol that leverages the multiplicatively homomorphic property of a bidirectional proxy re-encryption scheme Wang et al. [36] proposed two privacy-preserving schemes for outsourcing computation over ciphertexts under multiple keys. To the best of our knowledge, there are very few studies [35] on outsourced privacy-preserving dot product computing under multiple keys.

1.2 Our contribution

We present a cloud-assisted privacy-preserving profile-matching scheme under multiple keys to efficiently compute the social proximity between two users to discover potential friends while ensuring personal privacy. There are three main contributions of our scheme.

- **Non-interactive for users** Our scheme does not require the friend finder and the data provider to be online simultaneously. Before receiving the matching result, users only need to encrypt their personal profiles and send the ciphertext to one cloud server.
- **Efficient** The clouds perform most of the computation in our scheme, thereby effectively reducing the user's computational load and ensuring that the user's personal information is not leaked to the clouds.
- **Allow user and cloud collusion** Our scheme is secure under the HBC model assuming that the two cloud servers do not collude. Even if the participating parties collude with one of the cloud servers, our scheme will still not

reveal any private information about either the inputs, intermediate results or final results.

1.3 Organization

The remainder of this paper is organized as follows. In Sect. 2, we describe our system model and adversary model. In Sect. 3, we give some preliminaries, including the additively homomorphic encryption, proxy re-encryption and additively homomorphic proxy re-encryption schemes. Section 4 presents the cloud-assisted privacy-preserving profile-matching scheme. Section 5 analyses the correctness and security of our scheme. Section 6 compares our scheme with some known privacy-preserving profile-matching or dot product computation schemes. Section 7 concludes this paper.

2 Problem statement

2.1 System model

Figure 1 shows our system model. In our work, we denote Alice as the friend finder, and she wants to find a friend with similar interests from the other users in the mobile social network. The cloud environment consists of two cloud servers: cloud A (CA) and cloud B (CB). The two clouds provide large-scale data storage and computation services to reduce total cost. There are many users (data provider), and they encrypt their own private profile and outsource the computation to the cloud.

Each person in the mobile social network has a profile that is used to measure their personal preference. Every personal profile is defined as a vector $\mathbf{U} = < u_1, u_2, \cdots, u_n >$. The n represents the dimension of the vector. Every attribute corresponds to an interest, such as dancing and traveling, and every attribute value is an integer in [0, 10]. The attribute value represents the degree of the interest and indicates the level of interest, from no interest (0) to extremely high interest (10). In addition, social proximity is defined as the dot product of two users' vectors. Taking the dot product is a popular similarity criterion [3,18]. Before profile matching, we should normalize the dot products to have unit length. To perform the following encryption, every attribute value should be integerized as a member of some mathematic group. To ensure accuracy, we should keep H digits after the binary point. We can multiply every attribute value of users' vectors by 2^H and then integerize the result.

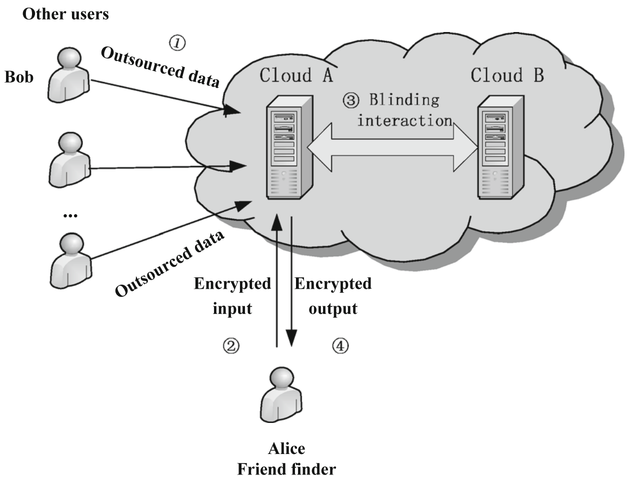


Fig. 1 System Model

2.2 Adversary model

The adversary model considered in this paper is under the honest-but-curious (HBC) framework [37]. In this framework, all participating parties faithfully follow the scheme, but they can collect and infer private information from the protocol and even collude with one of the cloud servers. However, the two cloud servers will never collude with each other. In the HBC model, the users perform the protocol faithfully and do not deliberately attempt to guess the dot product by adjusting the vector multiple times.

3 Preliminaries

3.1 Additive homomorphic

Suppose that $E_{pk}(m_1)$ and $E_{pk}(m_2)$ are two additively homomorphic ciphertexts under the same public key pk. The additively homomorphic cryptosystem has a key property.

- **Homomorphic.**

  1). $E_{pk}(m_1)E_{pk}(m_2) = E_{pk}(m_1 + m_2)$.
  2). $E_{pk}(m_1)^{m_2} = E_{pk}(m_1 \cdot m_2)$.

3.2 Proxy re-encryption

Proxy re-encryption (PRE) [38] allows a semi-trusted proxy to transform the ciphertext from Alice's public key pk_{A} into a ciphertext under CB's public key pk_{CB}. Furthermore, a key pair can be generated to allow the encrypted data to be delivered in a re-encrypted form such that CB can decrypt the data but the proxy cannot. Ultimately, the proxy learns nothing about the corresponding plaintext.

3.3 Additive homomorphic proxy re-encryption

In this work, we adopt the ElGamal-like encryption scheme (EL) in [34]. The EL scheme is a semantically secure proxy re-encryption scheme under the Decisional Bilinear Diffie-Hellman assumption [39,40], which supports additive homomorphism. The EL scheme is based on bilinear mapping and works as follows:

Suppose that G_1 and G_2 are two cyclic groups of prime order q with a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. g is a generator of G_1. The mapping e has three properties. (1) Bilinearity: For any $g \in G_1$ and $a, b \in Z_q$, $e(g^a, g^b) = e(g, g)^{ab}$ is efficiently computable. (2) Non-degeneracy: $e(g, g) \neq 1$. (3) Computability: e can be efficiently computed. Here, G_1, G_2, q, e, g and $Z = e(g, g) \in G_2$ are public parameters.

- **Key generation**: Given parameters, output the public key $pk_{\mathrm{a}} = (Z^{a_1}, g^{a_2})$ and the corresponding private key $sk_{\mathrm{a}} = (a_1, a_2)$, where $a_1, a_2 \xleftarrow{R} Z_q$.
- **Re-encryption key generation**: Given $pk_{\mathrm{b}} = (Z^{b_1}, g^{b_2})$ and $sk_{\mathrm{a}} = (a_1, a_2)$, output the proxy re-encryption key using the private key sk_{a} and public key pk_{b}. Specifically, $rk_{pk_{\mathrm{a}} \rightarrow pk_{\mathrm{b}}} = (g^{b_2})^{a_1} = g^{a_1 b_2}$.
- **Encryption**: Given pk_{a} and the message $m \in Z_q$, output $E_{pk_{\mathrm{a}}}(m) = (\beta, \gamma) = (g^r, Z^m Z^{a_1 r})$, where r is a random number from Z_q.
- **Decryption**: Given (β, γ) and sk_{a}, the ciphertext (β, γ) can be decrypted using sk_{a} by computing $\frac{\gamma}{e(g,\beta)^{a_1}} = \frac{Z^m Z^{a_1 r}}{e(g,g^r)^{a_1}} = \frac{Z^m Z^{a_1 r}}{Z^{a_1 r}} = Z^m$.
- **Re-encryption**: Given the ciphertext (β, γ) and $rk_{pk_{\mathrm{a}} \rightarrow pk_{\mathrm{b}}}$, compute $\beta^* = e(rk_{pk_{\mathrm{a}} \rightarrow pk_{\mathrm{b}}}, \beta) = e(g^{a_1 b_2}, g^r) = Z^{a_1 r b_2}$, $(\beta^*, \gamma) = (Z^{a_1 r b_2}, Z^m Z^{a_1 r})$, and output re-encrypted ciphertext (β^*, γ).
- **Re-decryption**: Given the re-encrypted ciphertext (β^*, γ) and sk_{b}, decrypt the re-encrypted ciphertext as $\frac{\gamma}{\beta^{*1/b_2}} = \frac{Z^m Z^{a_1 r}}{Z^{a_1 r}} = Z^m$.

The EL scheme requires computing the discrete logarithm of Z^m in base Z to obtain the plaintext m. If the plaintext size is less than 40 bits, it is efficient to compute the discrete logarithm using Pollard's kangaroo method [41]. The time complexity of Pollard's kangaroo method is $O(\sqrt{M})$, where M is the number of possible values of m.

4 Our construction

In this work, we adopt the EL scheme, which is a semantically secure proxy re-encryption scheme with additive homomorphism. CB and both users in the social network jointly generate the re-encryption keys for transforming the ciphertext from the user's public key into the ciphertext under the CB's public key. CA holds all re-encryption keys.

Let pk_{A} denote Alice's homomorphic public key. Suppose that Alice's vector is $\mathbf{U} =< u_1, u_2, \cdots, u_n >$ and that Bob's vector is $\mathbf{V} =< v_1, v_2, \cdots, v_n >$. The dot product of the vectors $\mathbf{U}$ and $\mathbf{V}$ can be calculated by the following two formulas:

- $\mathbf{U} \circ \mathbf{V} = (u_1 \cdot v_1 + u_2 \cdot v_2 + \cdots + u_n \cdot v_n) = \sum_{i=1}^{n} u_i v_i$
- $2\sum_{i=1}^{n} u_i v_i = \sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} v_i^2 - \sum_{i=1}^{n} (u_i - v_i)^2$

Intuition Users encrypt their vector with their own public key and then outsource the encrypted data to CA. The procedure for the data outsourcing is presented in Fig. 2.

Alice encrypts her vector with her public key pk_{A} and sends the ciphertexts to CA for measuring the proximity to Bob. The EL cryptosystem supports additive homomorphism. However, it can only support additive homomorphism under the same public key. To use the additive homomorphism, CA computes re-encrypted ciphertexts with $rk_{pk_{\mathrm{A}} \rightarrow pk_{\mathrm{CB}}}$ and $rk_{pk_{\mathrm{B}} \rightarrow pk_{\mathrm{CB}}}$. Because EL is an additive homomorphism, the additivity over the two ciphertexts can be performed by CA independently as fol-

Fig. 2 Protocol 1

Protocol 1 Data outsourcing

Input: a user (say, Bob) wishes to outsource his personal profile. He holds a private vector $\mathbf{V} =< v_1, v_2, \cdots, v_n >$ and his own pair of public key and private key $(pk_{\mathrm{B}}, sk_{\mathrm{B}})$
Output: output the $E_{pk_{\mathrm{B}}}(\mathbf{V})$ and $E_{pk_{\mathrm{B}}}(\sum_{i=1}^{n} v_i^2)$ to CA

```
1: for $i = 1$ to $n$ do
       computes $E_{pk_{\mathrm{B}}}(v_i)$
   end for

2: $E_{pk_{\mathrm{B}}}(\mathbf{V}) \longleftarrow < E_{pk_{\mathrm{B}}}(v_1), E_{pk_{\mathrm{B}}}(v_2) \cdots E_{pk_{\mathrm{B}}}(v_n) >$

3: Bob uploads $E_{pk_{\mathrm{B}}}(\mathbf{V})$ and $E_{pk_{\mathrm{B}}}(\sum_{i=1}^{n} v_i^2)$ to CA
```

```
Protocol 2 Privacy-preserving profile matching

Input Alice: a private vector $\mathbf{U} =< u_1, u_2, \cdots, u_n >$, public key $pk_{\mathrm{A}}$ and private key $sk_{\mathrm{A}}$
Input CA: re-encryption keys and outsourced data
Input CB: public key $pk_{\mathrm{CB}}$ and private key $sk_{\mathrm{CB}}$
Output Alice: $\mathbf{U} \circ \mathbf{V}$

1: Alice sends $E_{pk_{\mathrm{A}}}(u_i)$ and $E_{pk_{\mathrm{A}}}(\sum_{i=1}^{n} u_i^2)$ to CA for querying social proximity with Bob, who is in the mobile social network
2: CA computes re-encrypted ciphertexts with $rk_{pk_{\mathrm{A}} \to pk_{\mathrm{CB}}}$ and $rk_{pk_{\mathrm{B}} \to pk_{\mathrm{CB}}}$
   $E_{pk_{\mathrm{CB}}}(u_i) \xleftarrow{rk_{pk_{\mathrm{A}} \to pk_{\mathrm{CB}}}} E_{pk_A}(u_i)$
   $E_{pk_{\mathrm{CB}}}(\sum_{i=1}^{n} u_i^2) \xleftarrow{rk_{pk_{\mathrm{A}} \to pk_{\mathrm{CB}}}} E_{pk_{\mathrm{A}}}(\sum_{i=1}^{n} u_i^2)$
   $E_{pk_{\mathrm{CB}}}(v_i) \xleftarrow{rk_{pk_{\mathrm{B}} \to pk_{\mathrm{CB}}}} E_{pk_{\mathrm{B}}}(v_i)$
   $E_{pk_{\mathrm{CB}}}(\sum_{i=1}^{n} v_i^2) \xleftarrow{rk_{pk_{\mathrm{B}} \to pk_{\mathrm{CB}}}} E_{pk_{\mathrm{B}}}(\sum_{i=1}^{n} v_i^2)$
3: CA generates random integers $\delta_i$ and computes the following two terms:
     (i). $E_{pk_{\mathrm{CB}}}(u_i) \cdot E_{pk_{\mathrm{CB}}}(v_i)^{-1} = E_{pk_{\mathrm{CB}}}(u_i - v_i)$
     (ii). $E_{pk_{\mathrm{CB}}}(u_i - v_i)^{\delta_i} = E_{pk_{\mathrm{CB}}}(\delta_i(u_i - v_i))$
   Then, CA sends $E_{pk_{\mathrm{CB}}}(\delta_i(u_i - v_i))$ to CB
4: CB decrypts $E_{pk_{\mathrm{CB}}}(\delta_i(u_i - v_i))$ and computes $\delta_i(u_i - v_i) \cdot \delta_i(u_i - v_i)$. Next, CB sends $E_{pk_{\mathrm{CB}}}(\delta_i^2(u_i - v_i)^2)$ to CA
5: CA computes
     (i). $E_{pk_{\mathrm{CB}}}(\delta_i^2(u_i - v_i)^2)^{\delta_i^{-2}} = E_{pk_{\mathrm{CB}}}((u_i - v_i)^2)$
     (ii). $E_{pk_{\mathrm{CB}}}(\sum_{i=1}^{n} v_i^2) \cdot E_{pk_{\mathrm{CB}}}(\sum_{i=1}^{n} u_i^2) \cdot (E_{pk_{\mathrm{CB}}}(\sum\limits_{i=1}^{n} (u_i - v_i)^2))^{-1} = E_{pk_{\mathrm{CB}}}(2\mathbf{U} \circ \mathbf{V})$
     (iii). $E_{pk_{\mathrm{CB}}}(2\mathbf{U} \circ \mathbf{V})^{2^{-1}} = E_{pk_{\mathrm{CB}}}(\mathbf{U} \circ \mathbf{V})$
6: CA generates a random integer $w$ and sends $E_{pk_{\mathrm{CB}}}(w\mathbf{U} \circ \mathbf{V})$ to CB
7: CB decrypts $E_{pk_{\mathrm{CB}}}(w\mathbf{U} \circ \mathbf{V})$, computes $E_{pk_{\mathrm{A}}}(w\mathbf{U} \circ \mathbf{V})$ and sends it to CA
8: CA computes $E_{pk_{\mathrm{A}}}(w\mathbf{U} \circ \mathbf{V})^{w^{-1}} = E_{pk_{\mathrm{A}}}(\mathbf{U} \circ \mathbf{V})$. Finally, CA sends $E_{pk_{\mathrm{A}}}(\mathbf{U} \circ \mathbf{V})$ to Alice
9: Alice decrypts $E_{pk_{\mathrm{A}}}(\mathbf{U} \circ \mathbf{V})$ and outputs $\mathbf{U} \circ \mathbf{V}$
```

Fig. 3 Protocol 2

lows: $E_{pk_{\mathrm{CB}}}(m_1)E_{pk_{\mathrm{CB}}}(m_2) = E_{pk_{\mathrm{CB}}}(m_1 + m_2)$. Because $2\sum_{i=1}^{n} u_i v_i = \sum_{i=1}^{n} u_i^2 + \sum_{i=1}^{n} v_i^2 - \sum_{i=1}^{n}(u_i - v_i)^2$, we need to compute the product of ciphertexts $(u_i - v_i)$ by the two cloud servers. CA sends a blinded version of $E_{pk_{\mathrm{CB}}}(u_i - v_i)$ to CB. Then, CB decrypts the ciphertexts, performs the multiplication and encrypts the result with pk_{CB}. The encrypted result is sent to CA. CA computes $E_{pk_{\mathrm{CB}}}(\mathbf{U} \circ \mathbf{V})$ and sends the blinded ciphertext $E_{pk_{\mathrm{CB}}}(w\mathbf{U} \circ \mathbf{V})$ to CB. CB decrypts the ciphertext and encrypts it with Alice's public key. Then, CB sends $E_{pk_{\mathrm{A}}}(w\mathbf{U} \circ \mathbf{V})$ to CA. CA removes the blinding value w and sends the final result to Alice. The details of our scheme are given in Fig. 3.

The privacy of our scheme can be further enhanced by only letting Alice get a 1-bit matching result, i.e., a result of whether the dot product is above or below some threshold. We don't present the details of the improved scheme here but instead include them in the full version of this paper.

5 Correctness and security

5.1 Correctness

In our scheme, CA possesses $E_{pk_{\mathrm{A}}}(u_i)$, $E_{pk_{\mathrm{B}}}(v_i)$, $E_{pk_{\mathrm{A}}}(\sum_{i=1}^{n} u_i^2)$, $E_{pk_{\mathrm{B}}}(\sum_{i=1}^{n} v_i^2)$, $rk_{pk_{\mathrm{A}} \to pk_{\mathrm{CB}}}$, and $rk_{pk_{\mathrm{B}} \to pk_{\mathrm{CB}}}$; CB possesses sk_{CB}. Recall that CA computes re-encrypted ciphertexts with $rk_{pk_{\mathrm{A}} \to pk_{\mathrm{CB}}}$ and $rk_{pk_{\mathrm{B}} \to pk_{\mathrm{CB}}}$. Then, CA generates random integers δ_i, and using additive homomorphism and blinding with δ_i, CA obtains $E_{pk_{\mathrm{CB}}}(\delta_i(u_i - v_i))$. Then, CB decrypts $E_{pk_{\mathrm{CB}}}(\delta_i(u_i - v_i))$ and computes the products $(\delta_i(u_i - v_i))^2$. The products are later encrypted under

pk_{CB} by CB. After that, CA removes the blinding value δ_i^2 by

$$E_{pk_{\text{CB}}}\left(\delta_i^2(u_i - v_i)^2\right)^{\delta_i^{-2}} = E_{pk_{\text{CB}}}\left((u_i - v_i)^2\right),$$

and computes

$$\begin{aligned} &E_{pk_{\text{CB}}}\left((u_1 - v_1)^2\right) \cdot E_{pk_{\text{CB}}}\left((u_2 - v_2)^2\right) \cdots \cdot \\ &\quad E_{pk_{\text{CB}}}\left((u_n - v_n)^2\right) \\ &\quad = E_{pk_{\text{CB}}}\left(\sum\nolimits_{i=1}^{n} (u_i - v_i)^2\right), \\ &\quad E_{pk_{\text{CB}}}\left(\sum\nolimits_{i=1}^{n} v_i^2\right) \cdot E_{pk_{\text{CB}}}\left(\sum\nolimits_{i=1}^{n} u_i^2\right)\left(E_{pk_{\text{CB}}}\left(\sum\nolimits_{i=1}^{n} (u_i - v_i)^2\right)\right)^{-1} \\ &\quad = E_{pk_{\text{CB}}}\left(\sum\nolimits_{i=1}^{n} v_i^2 + \sum\nolimits_{i=1}^{n} u_i^2 - \sum\nolimits_{i=1}^{n} (u_i - v_i)^2\right) \\ &\quad = E_{pk_{\text{CB}}}(2\mathbf{U} \circ \mathbf{V}), \end{aligned}$$

and

$$E_{pk_{\text{CB}}}(2\mathbf{U} \circ \mathbf{V})^{2^{-1}} = E_{pk_{\text{CB}}}(\mathbf{U} \circ \mathbf{V}).$$

Then, CA obtains $E_{pk_{\text{CB}}}(w\mathbf{U} \circ \mathbf{V})$ via blinding with w. CB decrypts the ciphertext and encrypts it with Alice's public key. Finally, Alice removes the blinding value w and decrypts $E_{pk_{\text{A}}}(\mathbf{U} \circ \mathbf{V})$ to yield the desired output $\mathbf{U} \circ \mathbf{V}$.

5.2 Security

We now analyse the security of our scheme under the semi-honest model using a real and ideal paradigm [42]. For any adversary attacking a real protocol execution, there exists an adversary attacking an idea execution (with a trusted party) such that the input/output distributions of the adversary and the participating parties in the real and ideal executions are essentially the same.

Theorem 1 *Our scheme described in Sect. 4 can securely obtain the matching result via computations on ciphertexts in the presence of semi-honest (non-colluding) adversaries.*

Proof Our scheme involves four types of parties: Alice, Bob, CA and CB. We construct four simulators $Sim = (Sim_{\text{A}}, Sim_{\text{B}}, Sim_{\text{CA}}, Sim_{\text{CB}})$ against four types of adversaries $(\mathcal{A}_{\text{A}}, \mathcal{A}_{\text{B}}, \mathcal{A}_{\text{CA}}, \mathcal{A}_{\text{CB}})$ that corrupt Alice, Bob, CA, and CB, respectively.

$Sim_{\mathbf{A}}$ **simulates** $\mathcal{A}_{\mathbf{A}}$ **as follows**: After receiving the input of $\mathbf{U} =< u_1, u_2, \cdots, u_n >$, it encrypts data u_i as $E_{pk_{\text{A}}}(u_i)$ and encrypts data $\sum_{i=1}^{n} u_i^2$ as $E_{pk_{\text{A}}}(\sum_{i=1}^{n} u_i^2)$. Then, it randomly chooses data $\hat{\mathbf{V}} =< \hat{v}_1, \hat{v}_2, \cdots, \hat{v}_n >$, encrypts $E_{pk_{\text{A}}}(\mathbf{U} \circ \hat{\mathbf{V}})$ and sends it to $\mathcal{A}_{\text{A}}$. The view of $\mathcal{A}_{\text{A}}$ includes the input $\{u_i\}$, where $i \in [1, n]$, the encrypted data $\{E_{pk_{\text{A}}}(u_i), E_{pk_{\text{A}}}(\sum_{i=1}^{n} u_i^2)), E_{pk_{\text{A}}}(\mathbf{U} \circ \hat{\mathbf{V}})\}$ and the decrypted result $\{\mathbf{U} \circ \hat{\mathbf{V}}\}$. The views of $\mathcal{A}_{\text{A}}$ in the real and ideal executions are indistinguishable because of the security of the EL scheme mentioned above.

$Sim_{\mathbf{B}}$ **simulates** $\mathcal{A}_{\mathbf{B}}$ **as follows**: After receiving the input of $\mathbf{V} =< v_1, v_2, \cdots, v_n >$, it encrypts data v_i as $E_{pk_{\text{B}}}(v_i)$ and encrypts data $\sum_{i=1}^{n} v_i^2$ as $E_{pk_{\text{B}}}(\sum_{i=1}^{n} v_i^2)$. Finally, it returns $E_{pk_{\text{B}}}(v_i)$ and $E_{pk_{\text{B}}}(\sum_{i=1}^{n} v_i^2)$ to $\mathcal{A}_{\text{B}}$ and outputs $\mathcal{A}_{\text{B}}$'s entire view. The view of $\mathcal{A}_{\text{B}}$ includes input $\{v_i\}$, where $i \in [1, n]$, and the encrypted data $\{E_{pk_{\text{B}}}(v_i), E_{pk_{\text{B}}}(\sum_{i=1}^{n} v_i^2)\}$. The views of $\mathcal{A}_{\text{B}}$ in the real and ideal executions are indistinguishable because of the security of the EL scheme mentioned above.

$Sim_{\mathbf{CA}}$ **simulates** $\mathcal{A}_{\mathbf{CA}}$ **as follows**: It randomly chooses numbers $\hat{\mathbf{U}} =< \hat{u}_1, \hat{u}_2, \cdots, \hat{u}_n >$, $\hat{\mathbf{V}} =< \hat{v}_1, \hat{v}_2, \cdots, \hat{v}_n >$ and encrypts them as $E_{pk_{\text{A}}}(\hat{u}_i)$, $E_{pk_{\text{B}}}(\hat{v}_i)$, $E_{pk_{\text{A}}}(\sum_{i=1}^{n} \hat{u}_i^2)$, and $E_{pk_{\text{B}}}(\sum_{i=1}^{n} \hat{v}_i^2)$. It re-encrypts them as $E_{pk_{\text{CB}}}(\hat{u}_i)$, $E_{pk_{\text{CB}}}(\hat{v}_i)$, $E_{pk_{\text{CB}}}(\sum_{i=1}^{n} \hat{u}_i^2)$, and $E_{pk_{\text{CB}}}(\sum_{i=1}^{n} \hat{v}_i^2)$. Then, it generates random integers $\hat{\delta}_i$ and $\hat{w}$ and computes $E_{pk_{\text{CB}}}(\hat{\delta}_i^2 (u_i - v_i)^2)^{\delta_i^{-2}}$, $E_{pk_{\text{CB}}}(\hat{w}\hat{\mathbf{U}} \circ \hat{\mathbf{V}})$ and $E_{pk_{\text{A}}}(\hat{\mathbf{U}} \circ \hat{\mathbf{V}})$. The view of $\mathcal{A}_{\text{CA}}$ is the encrypted data. The views of $\mathcal{A}_{\text{CA}}$ in the real and ideal executions are indistinguishable because of the security of the EL scheme mentioned above.

$Sim_{\mathbf{CB}}$ **simulates** $\mathcal{A}_{\mathbf{CB}}$ **as follows**: It randomly chooses numbers $\hat{m}_i$ and encrypts them as $E_{pk_{\text{CB}}}(\hat{m}_i)$. Then, it computes $\hat{m}_i \cdot \hat{m}_i$ and encrypts $\hat{m}_i \cdot \hat{m}_i$ with CB's public key. Then, it randomly chooses a number $\hat{s}$, encrypts it as $E_{pk_{\text{CB}}}(\hat{s})$ and encrypts $\hat{s}$ with Alice's public key. Finally, it returns $\{E_{pk_{\text{CB}}}(\hat{m}_1), E_{pk_{\text{CB}}}(\hat{m}_2), \ldots, E_{pk_{\text{CB}}}(\hat{m}_n), \hat{m}_1, \hat{m}_2, \ldots, \hat{m}_n, E_{pk_{\text{CB}}}(\hat{m}_1^2), E_{pk_{\text{CB}}}(\hat{m}_2^2), \ldots, E_{pk_{\text{CB}}}(\hat{m}_n^2), E_{pk_{\text{CB}}}(\hat{s}), E_{pk_{\text{A}}}(\hat{s})\}$ to $\mathcal{A}_{\text{CB}}$. $\mathcal{A}_{\text{CB}}$ is able to decrypt the ciphertexts with its private key, but the decrypted messages are all blinded. Because of the random numbers, the decrypted messages are randomly distributed. The view of $\mathcal{A}_{\text{CB}}$ is also the encrypted data and blinded data. Security in the real world can be guaranteed by the security of the EL scheme. The views of $\mathcal{A}_{\text{CB}}$ in the real and ideal executions are indistinguishable.

For the case of a user colluding with CA or CB, the security can be proven in a similar manner. □

6 Comparison

We compare our new scheme with some known privacy-preserving profile-matching schemes in Table 1. In [3,5,43], multiple rounds of interactions between users are required to perform the profile matching, which causes high communication and computation cost for users. Our scheme does not require users to be online at the same time. Users only need to encrypt their personal profiles and send them to one cloud server and receive matching results. The two cloud servers perform most of the computations in our scheme, effectively reducing the user's computational burden and ensuring that the user's private information is not leaked to the clouds.

Table 1 Comparison of our scheme with some known privacy-preserving profile-matching schemes

Scheme	Our scheme	Zhang et al.'s scheme [3]	Dong et al.'s scheme [43]	Zhang et al.'s scheme [5]
Non-interactive for users	Yes	No	No	No
Fine-grained private matching	Yes	Yes	Yes	No

Table 2 The comparison of interaction quantities

Scheme	Our scheme	Sheng et al.'s scheme [44] and Vaidya et al.'s scheme [45]	PTKb [32]	OPPSP* [35]
S2S interactions in Re-Enc	0	N/ A	n	0
S2S interactions in dot product computing	n	N/ A	n	$2n - 2$
Support multi-key	Yes	No	Yes	Yes

The n represents the dimension of the vector
S2S server to server, *Re-Enc.* re-encryption, *N/ A* not applicable

The comparison of interaction quantities among some known privacy- preserving schemes for dot product calculation is shown in Table 2. [35,44,45] were constructed to achieve secure dot product computation. [32] was constructed to achieve secure addition and multiplication under multi-key. Because schemes [44,45] do not use multiple servers, there is no interaction between servers. Goethals et al. [46] showed that two of the private scalar product protocols, one of which [45] adopting the matrix multiplication operation was proved to be insecure. The approach of [31] differs from [32] in the sense that [31] randomly separates the strong trapdoor into two shares, and distributes the shares to two different servers. Only when both servers work together can the ciphertext be successfully decrypted. This decreases the risk of privacy leakage caused by single point attack. Current solutions for privacy-preserving profile matching either have heavy interactions or require users to encrypt private data under one key. In our scheme, the computation is non-interactive to users, and the scheme is secure under the honest-but-curious model, assuming that the two cloud servers do not collude. Furthermore, our scheme is collusion resistant, i.e., collusion between a user and a cloud server will not reveal any privacy information.

7 Conclusion

In this paper, we propose a novel cloud-assisted privacy-preserving profile-matching scheme under multiple keys based on a proxy re-encryption scheme with additive homomorphism. Current solutions either have heavy interactions or require users to encrypt private data under one key. In our scheme, the computation is non-interactive to users. Users only need to encrypt their personal profiles and send them to one cloud server and receive matching results. In addition, the two cloud servers perform most of the computations, effectively reducing the user's computational burden and ensuring that the user's private information is not leaked to the clouds. Furthermore, our scheme is proven secure under the honest-but-curious model, assuming that the two cloud servers do not collude. Even if participating parties collude with one of the cloud servers, our scheme will still not reveal any user's private information.

Acknowledgements We would like to thank anonymous reviewers for giving valuable suggestions to this paper. This work is supported by the State Key Laboratory of Cryptology, Beijing, China, the Innovation Team Project of Guangdong Province Universities (No. 2015KCXTD014, 2016KCXTD017), and the Natural Science Foundation of China (61772148).

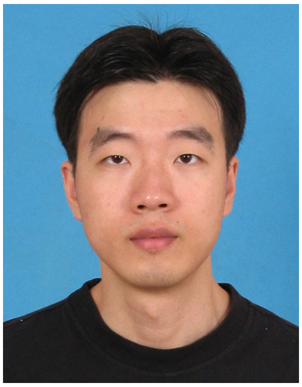
References

1. Atat, R., Liu, L., Chen, H., Wu, J., Li, H., Yi, Y.: Enabling cyber-physical communication in 5g cellular networks: challenges, spatial spectrum sensing, and cyber-security. IET Cyber-Phys. Syst. Theory Appl. **2**, 49–54 (2017)
2. Zhang, Z., Gupta, B.B.: Social media security and trustworthiness: overview and new direction. Future Gener. Comput, Syst (2016)
3. Zhang, R., Zhang, J., Zhang, Y., Sun, J., Yan, G.: Privacy-preserving profile matching for proximity-based mobile social networking. IEEE J. Sel. Areas Commun. **31**, 656–668 (2013)
4. Li, M., Yu, S., Cao, N., Lou, W.: Privacy-preserving distributed profile matching in proximity-based mobile social networks. IEEE Trans. Wirel. Commun. **12**, 2024–2033 (2013)
5. Zhang, L., Li, X.Y., Liu, K., Jung, T., Liu, Y.: Message in a sealed bottle: privacy preserving friending in mobile social networks. IEEE Trans. Mob. Comput. **14**, 1888–1902 (2015)
6. Li, M., Cao, N., Yu, S., Lou, W.: Findu: Privacy-preserving personal profile matching in mobile social networks. In: INFOCOM, 2011 Proceedings IEEE, IEEE, pp. 2435–2443 (2011)
7. Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, F., Patterson, D., Rabkin, A., Stoica, I., et al.: Above the

clouds: a berkeley view of cloud computing, UC Berkeley EECS, 10 February (2013)

8. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R.H., Konwinski, A., Lee, G., Patterson, D.A., Rabkin, A., Stoica, I., Zaharia, M.: A view of cloud computing. Commun. ACM **53**, 50–58 (2010)
9. Wei, L., Zhu, H., Cao, Z., Dong, X., Jia, W., Chen, Y., Vasilakos, A.V.: Security and privacy for storage and computation in cloud computing. Inf. Sci. **258**, 371–386 (2014)
10. Wu, J., Guo, S., Li, J., Zeng, D.: Big data meet green challenges: Big data toward green applications. IEEE Syst. J. **10**, 888–900 (2016)
11. Wu, J., Guo, S., Li, J., Zeng, D.: Big data meet green challenges: greening big data. IEEE Syst. J. **10**, 873–887 (2016)
12. Stergiou, C., Psannis, K.E., Kim, B., Gupta, B.B.: Secure integration of iot and cloud computing. Future Gener. Comput. Syst. **78**, 964–975 (2018)
13. Alsmirat, M.A., Jararweh, Y., Obaidat, I., Gupta, B.B.: Internet of surveillance: a cloud supported large-scale wireless surveillance system. J. Supercomput. **73**, 973–992 (2017)
14. Gupta, S., Gupta, B.B.: XSS-Secure as a Service for the Platforms of Online Social Network-Based Multimedia Web Applications in Cloud, Multimedia Tools and Applications, pp. 1–33. Springer, New York (2016)
15. Hamedani, K., Liu, L., Rachad, A., Wu, J., Yi, Y.: Reservoir computing meets smart grids: attack detection using delayed feedback networks. IEEE Trans. Ind. Inf. (2017). https://doi.org/10.1109/TII.2017.2769106
16. Negi, P., Mishra, A., Gupta, B.B.: Enhanced CBF packet filtering method to detect ddos attack in cloud computing environment. CoRR **abs/1304.7073** (2013)
17. Von Arb, M., Bader, M., Kuhn, M., Wattenhofer, R.: Veneta: Serverless friend-of-friend detection in mobile social networking. In: Networking and Communications, 2008. WIMOB'08. IEEE International Conference on Wireless and Mobile Computing, IEEE 184–189 (2008)
18. Ioannidis, I., Grama, A., Atallah, M.: A secure protocol for computing dot-products in clustered and distributed environments. In: International Conference on Parallel Processing, 2002. Proceedings, IEEE, pp. 379–384 (2002)
19. Zhang, L., Li, X.Y., Liu, Y., Jung, T.: Verifiable private multi-party computation: ranging and ranking. In: INFOCOM, 2013 Proceedings IEEE, IEEE, pp. 605–609 (2013)
20. Boneh, D., Gentry, C., Halevi, S., Wang, F., Wu, D.J.: Private database queries using somewhat homomorphic encryption. In: Jr., M.J.J., Locasto, M.E., Mohassel, P., Safavi-Naini, R., (eds).: Applied Cryptography and Network Security—11th International Conference, ACNS 2013, Banff, AB, Canada, June 25–28, 2013. Proceedings. Volume 7954 of Lecture Notes in Computer Science., Springer 102–118 (2013)
21. Wang, W., Hu, Y., Chen, L., Huang, X., Sunar, B.: Exploring the feasibility of fully homomorphic encryption. IEEE Trans. Computers **64**, 698–706 (2015)
22. Gentry, C.: Computing arbitrary functions of encrypted data. Commun. ACM **53**, 97–105 (2010)
23. Li, P., Li, J., Huang, Z., Li, T., Gao, C., Yiu, S., Chen, K.: Multi-key privacy-preserving deep learning in cloud computing. Future Gener. Comput. Syst. **74**, 76–85 (2017)
24. Li, P., Li, J., Huang, Z., Gao, C.Z., Chen, W.B., Chen, K.: Privacy-preserving outsourced classification in cloud computing. Clust, Comput. pp. 1–10 (2017). https://doi.org/10.1007/s10586-017-0849-9
25. Choi, S.G., Elbaz, A., Juels, A., Malkin, T., Yung, M.: Two-party computing with encrypted data. In: Kurosawa, K., (ed.) Advances in Cryptology—ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, 2–6 December 2007, Proceedings. Volume 4833 of Lecture Notes in Computer Science., Springer 298–314 (2007)
26. Li, J., Chen, X., Huang, X., Tang, S., Xiang, Y., Hassan, M.M., Alelaiwi, A.: Secure distributed deduplication systems with improved reliability. IEEE Trans. Computers **64**, 3569–3579 (2015)
27. Li, J., Li, Y.K., Chen, X., Lee, P.P.C., Lou, W.: A hybrid cloud approach for secure authorized deduplication. IEEE Trans. Parallel Distrib. Syst. **26**, 1206–1216 (2015)
28. Li, J., Li, J., Chen, X., Jia, C., Lou, W.: Identity-based encryption with outsourced revocation in cloud computing. IEEE Trans. Computers **64**, 425–437 (2015)
29. Li, J., Zhang, Y., Chen, X., Xiang, Y., Li, J., Zhang, Y., Chen, X., Xiang, Y.: Secure attribute-based data sharing for resource-limited users in cloud computing. Comput. Secur. **72**, 1–12 (2017)
30. López-Alt, A., Tromer, E., Vaikuntanathan, V.: On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption. In Karloff, H.J., Pitassi, T. (eds.) Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, 19–22 May 2012, ACM, pp. 1219–1234 (2012)
31. Liu, X., Deng, R.H., Choo, K.R., Weng, J.: An efficient privacy-preserving outsourced calculation toolkit with multiple keys. IEEE Trans. Inf. Forensics Secur. **11**, 2401–2414 (2016)
32. Peter, A., Tews, E., Katzenbeisser, S.: Efficiently outsourcing multiparty computation under multiple keys. IEEE Trans. Inf. Forensics Secur. **8**, 2046–2058 (2013)
33. Bresson, E., Catalano, D., Pointcheval, D.: A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In Laih, C. (ed.) Advances in Cryptology—ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30–December 4, 2003, Proceedings. Volume 2894 of Lecture Notes in Computer Science, Springer 37–54 (2003)
34. Samanthula, B.K., Elmehdwi, Y., Howser, G., Madria, S.K.: A secure data sharing and query processing framework via federation of cloud computing. Inf. Syst. **48**, 196–212 (2015)
35. Rong, H., Wang, H., Huang, K., Liu, J., Xian, M.: Privacy-preserving scalar product computation in cloud environments under multiple keys. In: Yin, H., Gao, Y., Li, B., Zhang, D., Yang, M., Li, Y., Klawonn, F., Tallón-Ballesteros, A.J., (eds.) Intelligent Data Engineering and Automated Learning—IDEAL 2016—17th International Conference, Yangzhou, China, October 12-14, 2016, Proceedings. Volume 9937 of Lecture Notes in Computer Science., Springer 248–258 (2016)
36. Wang, B., Li, M., Chow, S.S., Li, H.: A tale of two clouds: computing on data encrypted under multiple keys. In: IEEE Conference on Communications and Network Security, IEEE, pp. 337–345 (2014)
37. Goldreich, O.: The Foundations of Cryptography. Basic Applications, vol. 2. Cambridge University Press, New York (2004)
38. Blaze, M., Bleumer, G., Strauss, M.: Divertible protocols and atomic proxy cryptography. In Nyberg, K. (ed.) Advances in Cryptology—EUROCRYPT '98, International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May 31–June 4, 1998, Proceeding. Volume 1403 of Lecture Notes in Computer Science., Springer 127–144 (1998)
39. Boneh, D., Franklin, M.K.: Identity-based encryption from the weil pairing. SIAM J. Comput. **32**, 586–615 (2003)
40. Cheon, J.H., Lee, D.H.: Diffie-hellman problems and bilinear maps. IACR Cryptol. ePrint Archive **2002**, 117 (2002)
41. Pollard, J.M.: Monte carlo methods for index computation (mod p). Math. Comput. **32**, 918–924 (1978)
42. Lindell, Y., Pinkas, B.: Secure multiparty computation for privacy-preserving data mining. IACR Cryptol. ePrint Arch. **2008**, 197 (2008)

43. Dong, W., Dave, V., Qiu, L., Zhang, Y.: Secure friend discovery in mobile social networks. In: INFOCOM, 2011 Proceedings IEEE, IEEE, pp. 1647–1655 (2011)
44. Sheng, G., Wen, T., Guo, Q., Yin, Y.: Privacy preserving inner product of vectors in cloud computing. Int. J. Distrib. Sensor Netw. **10**(5), 537252 (2014)
45. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 639–644 (2002)
46. Goethals, B., Laur, S., Lipmaa, H., Mielikäinen, T.: On private scalar product computation for privacy-preserving data mining. In: International Conference on Information Security and Cryptology, Springer 104–120 (2004)



Chong-zhi Gao is a professor at the School of Computer Science of Guangzhou University. He received his Ph.D. (2004) from Sun Yat-sen University in applied mathematics. His research interests include cryptography and privacy in machine learning.

Qiong Cheng received her Bachelor's degree from Guangzhou University, in 2016. Currently, she is a master student of computer science at Guangzhou University.

Xuan Li received her BS degree in applied mathematics and PhD degree in computer science from south china university of technology. She is currently an associate professor in college of mathematics and informatics, Fujian Normal University, China. Her recent research interests include applied cryptography and privacy protection in cloud computing.

Shi-bing Xia received his Bachelor's degree from Tianjin University of Technology, in 2014. Currently, he is a master student of computer science at Guangzhou University.