Privacy-Preserving Profile Matching Protocol Considering Conditions

Yosuke Ishikuro and Kazumasa Omote(✉)

JAIST, Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan
{s1410007,omote}@jaist.ac.jp

Abstract. A social matching service has recently become popular. These services help a user to search friends having common preference or interest. On the other hand, users use their personal information for matching in social matching services, and thus the privacy-preserving profile matching protocols have been well studied. However, although there are various privacy-preserving profile matching protocols, they may cause unwilling matching. In order to solve this problem, it is necessary to achieve a fine-grained matching mechanism considering conditions.

In this paper, we propose a privacy-preserving profile matching protocol embedded with homomorphic encryption considering conditions: matching is established only when the conditions are satisfied. Our protocol reduces computational cost of user's device by using the map-to-prime technique and setting an honest-but-curious server. Furthermore, even if a server is attacked, user's secret key or personal data does not leak since our protocol is designed for a server without such confidential data.

Keywords: Privacy · Profile matching · Homomorphic encryption · Mobile social networks

1 Introduction

With the proliferation of mobile devices such as smart phones and tablets, a social network is becoming an inseparable part of our life. A social matching service has recently become popular. These services help users to search friends having common preference or interest. Users can make new social connections or friends based on matching of their personal profiles. However, a social matching service deals with user's personal profile which includes sensitive information such as name, age, location and preference. Thus, we should protect user's privacy. The service provider needs to reassure users by properly managing personal data and hence it is necessary to prevent leaking personal data. It is also required to safely manage user's personal data for improving quality of service.

A lot of privacy-preserving profile matching protocols have been studied for preventing the leakage of private information in recent years. So, in these protocols, matching is processed using encrypted user's profile. For example, a user

J. Chen et al. (Eds.): NSS 2016, LNCS 9955, pp. 171–183, 2016.
DOI: 10.1007/978-3-319-46298-1_12

A answers some questions to construct her/his profile and then encrypts her/his answers. A's profile is compared with another profile of user B with encrypted. After that, A and B obtain the matching result by decrypting.

Thanks to such cryptographic technology, even if their profiles include sensitive information, a malicious user or a server cannot learn about it except for the matching result. However, the existing privacy-preserving profile matching protocols have a drawback of unwilling matching. In the existing protocols, if two users have at least one common preference or interest, then they output the result "matching is established". Namely, even if a user B has one profile item that another user A cannot accept, the matching between A and B may unwillingly established. For example, we assume that A wants to match to another user who likes baseball but A does not want to match to a smoker. If B likes baseball but B is a smoker, the existing protocols reluctantly output the result "matching is established" based on the attribute "baseball", although A does not fundamentally want to match to B. This result may disappoint A.

In this paper, we propose a privacy-preserving profile matching protocol embedded with homomorphic encryption considering conditions: matching is established only when the conditions are satisfied. As a result, our protocol can prevent the unwilling matching, which occurs in the existing protocols, by setting the conditions. If A wants to match to only a non-smoker, A sets the condition of "non-smoker" against another user. Even if both A and B like "baseball", they are not matched because B is smoker, that is, B does not satisfy the condition of A. Our protocol reduces computational cost of user's device by using the map-to-prime technique and setting an honest-but-curious server. Furthermore, even if a server is attacked, user's secret key or personal data does not leak since our protocol is designed for a server without such confidential data. We assume that every entity has honest-but-curious setting and that secure channel is used among A, B and S. A and B do not directly communicate in order to preserve the fairness and to reduce computational cost on users' devices.

The remainder of the paper is structured as follows: In Sect. 2, we discuss some related works of a privacy-preserving profile matching protocol. Section 3 includes preliminaries. In Sect. 4, we present the privacy-preserving profile matching system. Section 5 gives our proposed protocol in detail. In Sect. 6, performance evaluation is discussed. Finally, Sect. 7 concludes the paper.

2 Related Works

In 2004, PSI protocol using Oblivious Polynomial Evaluation (OPE) was proposed for the first time by Freedman et al. [2]. Then, Kim et al. [4] reduced computational cost of user's device by using the map-to-prime technique instead solution of the polynomial in OPE. Many existing matching protocols need to generate one ciphertext for one question about user profiles, and this means that the large amount of computational cost of encryption is required if the number of questions increases. The map-to-prime technique makes it possible to embed more than one profile inside one ciphertext and hence makes it possible to decrease the computational cost of encryption. We can mainly classify the existing schemes into two types: (1) enhancing the privacy and (2) enhancing the matching function.

Enhancing the privacy restricts the output contents of matching results. Abbas et al. [1] proposes cardinality matching which outputs only the number of matched elements without revealing the matched elements. In [6,7], the private attributes are certified by a trusted third party and these prevent honest-but-curious and malicious users from learning profile information of honest user by choosing their set arbitrarily. In [5,8,9], privacy is enhanced by restricting the information obtained from the matching result as the privacy level rises. For example in [9], in level 1 users can learn the matched elements and their level of interest. In level 2 it outputs the matched elements between users. In level 3 users can learn only if they matched without learning the matched elements.

Enhancing the matching function achieves more detailed matching of user's profiles. Zhu et al. [10] proposes the conditional matching protocol which is established only when the number of matched elements is equivalent to the number a user requires. However, the condition setting of this protocol is not realistic. He et al. [3] proposes more detailed matching protocol in which users can set weights to their profile. Thapam et al. [8] proposes the practical matching protocol which achieves a communication closer to real life, by using not only users' own information but also information of their friends. As explained above, although various matching protocols have been proposed, the fine-grained matching considering user's conditions has not been achieved yet.

3 Preliminaries

3.1 Requirements

Fain-grained profile matching:

The existing matching protocols have a drawback of unwilling match. Even if a user has some profiles that anther user is unacceptable, they may reluctantly match each other as described in Sect. 1. In order to solve this problem, the fine-grained profile matching protocol considering conditions is required.

Safety management of personal information:

Since user's profile includes personal information, it is required to store user's profile to keep a secret. Also, it is required that a server does not have private keys of users or a server and that it does not use them on itself. If a server does not have secret information, the safety management of a server becomes easy.

Reduction of computational cost:

Many existing matching protocols need to generate one ciphertext for one question of profiles, and this means that the large amount of computational cost of encryption is required if the number of questions increases. In order to solve this problem, one ciphertext for multiple profiles is required. This can reduce the computational cost and memory consumption.

3.2 Paillier Encryption

The protocol proposed in this paper is based on Paillier's homomorphic encryption. In the following, we summarize Paillier crypto system.

Key Generation:
The trusted third party chooses two large prime numbers p and q randomly such that $\gcd(pq,(p-1)(q-1)) = 1$ and compute $n = pq$ and $g = (1+\alpha n)\beta^n \bmod n^2$ and $\lambda = \mathrm{lcm}(p-1, q-1)$, where gcd() and lcm() are the functions that computes the greatest common divisor and the least common multiple, respectively. Furthermore, it computes $\mu = (L(g^\lambda \bmod n^2))^{-1} \bmod n$, where $L(u) = (u-1)/n$. The Paillier public and private keys are (n, g) and λ, respectively.

Encryption:
Let $M \in \mathbb{Z}_n$ be a message to be encrypted and $r \in \mathbb{Z}^*_{n^2}$ be a random number. The ciphertext could be given by

$$E(M) = g^M r^n \bmod n^2 \tag{1}$$

Decryption:
Given a ciphertext $c = E(M)$, the corresponding plaintext can be derived as

$$\frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n = M \tag{2}$$

Homomorphic:
Given $m_1, m_2, r_1, r_2 \in \mathbb{Z}_N$, it satisfies the following homomorphic property:

$$E(m_1) \cdot E(m_2) = E(m_1 + m_2) \tag{3}$$

3.3 Adversary Model

We consider an internal attacker that is a malicious user or server. We assume that the adversary model is honest-but-curious setting. Honest-but-curious users or server follow the protocol but they are curious to learn about user's interest. Additionally, we do not consider the collusion among users and server. This model is required to satisfy correctness and privacy as follows.

- Correctness.
  If two users output the matching result of each profile correctly, this protocol has correctness.
- Privacy.
  If nothing is known about each user's profile which is not existed in the matching result, this protocol has privacy.

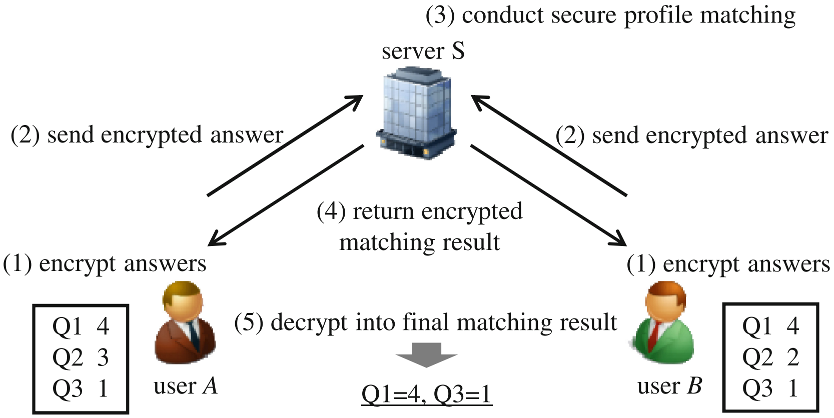

(3) conduct secure profile matching
server S
(2) send encrypted answer
(2) send encrypted answer
(4) return encrypted matching result
(1) encrypt answers
(1) encrypt answers
Q1 4
Q2 3
Q3 1
user *A*
(5) decrypt into final matching result
Q1=4, Q3=1
user *B*
Q1 4
Q2 2
Q3 1

Fig. 1. Privacy-preserving profile matching system.

4 Privacy-Preserving Profile Matching System

A privacy-preserving matching system is that each user answers some questions about user's profile and then obtain only the matching result with another user. If users A and B are matched in some questions, they obtain only the matched items each other. Figure 1 shows an overview of privacy-preserving profile matching system. The basic procedure is as follows.

1. A and B encrypt their own answer about their profiles.
2. A and B send their encrypted answer to a server S.
3. S conducts the secure profile matching with keeping personal information secret.
4. S returns the encrypted matching result to A and B.
5. A and B respectively decrypt the matching result received from S and then they can obtain the final matching result.

In Fig. 1, the final matching result is $\text{Q1} = 4$ and $\text{Q3} = 1$ since A's answers of question 1 and 3 are the same as B. Note that the answer of question 2 is kept secret since the question 2 is not matched between A and B.

We assume that A, B and S are the honest-but-curious entities. In other words, S, A and B are curious to learn about a user's interest but honestly follow the protocol. In addition, we do not assume the collision among A, B and S and assume a secure channel between the server and users.

5 Our Protocol

In the existing protocols, even if a user B has preference or interest that another user A cannot accept, the matching between A and B may unwillingly established. In order to solve this problem, we propose a privacy-preserving profile

Table 1. Notation.

Notation	Description
pk_A, pk_B	Public keys of users A and B
C	Choice set
$\mathcal{S}$	Set of prime numbers to C
t	Size of each prime in C
C_c	Choice set for condition, $C_c \subset C$
γ	Number of questions
$X_A = \{a_1, \dots, a_\gamma\}$	A's answer, $a_i \in S$
$X_B = \{b_1, \dots, b_\gamma\}$	B's answer, $b_i \in S$
$a = \prod_{i=1}^{\gamma}, b = \prod_{i=1}^{\gamma}$	Answers of A and B
a_c	B's condition for user A
b_c	A's condition for user B
r_{aa}, r_{ab}, r_{ba}, r_{bb}	Random numbers
X_{AB}, X_{BA}	Matching results of A and B

matching protocol embedded with homomorphic encryption considering conditions: matching is established only when the conditions are satisfied. Our protocol uses the map-to-prime technique to reduce the computational cost of user's device. Our protocol also has conditions that each user sets to achieve the fine-grained matching. We assume that the secure channel is used among A, B and S and that A and B do not directly communicate in order to preserve the fairness and to reduce computational cost on users' devices.

5.1 Notation

Table 1 shows the notation of our protocol. C is a set of choices contained in one ciphertext. Our protocol deals with single answer only from multiple-choice question. $\mathcal{S}$ is the set of prime numbers corresponding to C. Users A and B select prime numbers corresponding to their own answers as $X_A = \{a_1, ..., a_\gamma\} \in \mathcal{S}$ and $X_B = \{b_1, ...b_\gamma\} \in \mathcal{S}$, respectively. User's answer is represented by product of prime numbers. More precisely, the answers of A and B are denoted by $a = \prod_{i=1}^{\gamma} a_i$ and $b = \prod_{i=1}^{\gamma} b_i$, respectively. Each user chooses a condition from $C_c \subset C$. The conditions of A and B are denoted by $b_c \in \mathcal{S}$ and $a_c \in \mathcal{S}$, respectively. If A does not satisfy B's condition a_c or B does not satisfy A's condition b_c, then the matching result is not output. Only if both A and B satisfy conditions each other, the matching is certainly established as usual.

5.2 Protocol Detail

We explain about the procedure that A obtains the matching result since users A and B are in a symmetric position. Figure 2 shows our privacy-preserving profile

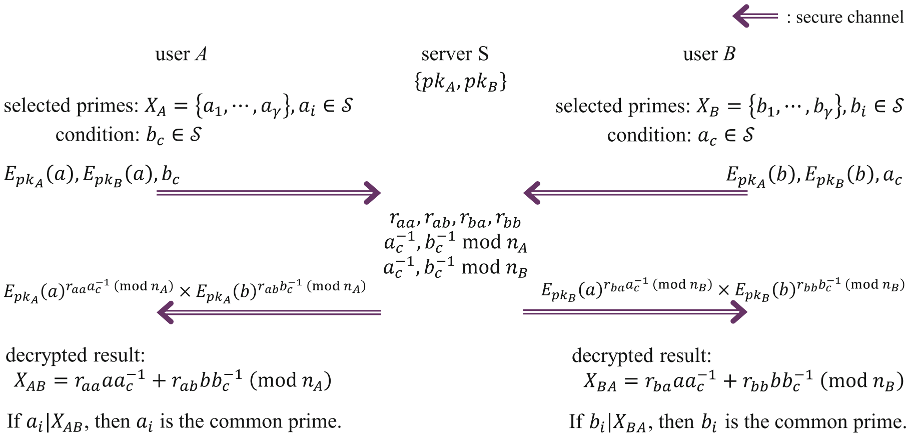

: secure channel
user A
server S
$\{pk_A, pk_B\}$
user B
selected primes: $X_A = \{a_1, \cdots, a_\gamma\}, a_i \in \mathcal{S}$
condition: $b_c \in \mathcal{S}$
selected primes: $X_B = \{b_1, \cdots, b_\gamma\}, b_i \in \mathcal{S}$
condition: $a_c \in \mathcal{S}$
$E_{pk_A}(a), E_{pk_B}(a), b_c$
$E_{pk_A}(b), E_{pk_B}(b), a_c$
$r_{aa}, r_{ab}, r_{ba}, r_{bb}$
a_c^{-1}, b_c^{-1} mod n_A
a_c^{-1}, b_c^{-1} mod n_B
$E_{pk_A}(a)^{r_{aa}a_c^{-1} \pmod{n_A}} \times E_{pk_A}(b)^{r_{ab}b_c^{-1} \pmod{n_A}}$
$E_{pk_B}(a)^{r_{ba}a_c^{-1} \pmod{n_B}} \times E_{pk_B}(b)^{r_{bb}b_c^{-1} \pmod{n_B}}$
decrypted result:
$X_{AB} = r_{aa}aa_c^{-1} + r_{ab}bb_c^{-1} \pmod{n_A}$
If $a_i | X_{AB}$, then a_i is the common prime.
decrypted result:
$X_{BA} = r_{ba}aa_c^{-1} + r_{bb}bb_c^{-1} \pmod{n_B}$
If $b_i | X_{BA}$, then b_i is the common prime.

Fig. 2. Privacy-preserving profile matching protocol considering conditions between two users.

matching protocol considering conditions between two users. We need only one ciphertext for plural questions of profiles owing to the map-to-prime technique. Note that two or more ciphertext is required when $t\gamma > |n_A|$. The protocol detail is shown in Fig. 2.

1. S chooses a set $\mathcal{S}$ corresponding to a set C for the map-to-prime technique.
2. A and B respectively generate their own public key pk_A and pk_B and send them to S.
3. A receives pk_B from S. A computes $E_{pk_A}(a)$ and $E_{pk_B}(a)$ by encrypting her/his own answer, and then selects the condition b_c from C_c. A sends $E_{pk_A}(a)$, $E_{pk_B}(a)$ and b_c to S. B processes in a similar way.
4. S generates four random numbers r_{aa}, r_{ab}, r_{ba} and r_{bb} where $|r_{aa}| = |r_{ab}| = |n_A| - t\gamma + t - 1$ and $|r_{ba}| = |r_{bb}| = |n_B| - t\gamma + t - 1$. These random numbers are used to pad message space. S obtains the prime number corresponding to each condition received from A and B. Then, S computes the inverse elements of b_c and a_c on n_A and n_B, i.e., $b_c^{-1} (\text{mod } n_A)$, $a_c^{-1} (\text{mod } n_A)$, $b_c^{-1} (\text{mod } n_B)$ and $a_c^{-1} (\text{mod } n_B)$. Finally, S computes the following Eq. (4) and returns it to A.
$$\begin{aligned} &E_{pk_A}(a)^{r_{aa}a_c^{-1} \pmod{n_A}} \times E_{pk_A}(b)^{r_{ab}b_c^{-1} \pmod{n_A}} \\ &= E_{pk_A}(r_{aa}aa_c^{-1} + r_{ab}bb_c^{-1}) \\ &= E_{pk_A}(X_{AB}), \end{aligned} \tag{4}$$
where $X_{AB} = r_{aa}aa_c^{-1} + r_{ab}bb_c^{-1}$.
5. A decrypts $E_{pk_A}(X_{AB})$ to obtain X_{AB}.
6. A verifies the matching result, that is, A conducts $a_i | X_{AB}$ $(i = 1, ..., \gamma)$. If it is true, a_i is the common prime between A and B, otherwise, a_i is not common. If A and B satisfy their conditions each other, both users can obtain the matching result. Otherwise, neither A nor B can obtain any matching result.

Note that we can easily construct the protocol among m users by operating our protocol between two users in parallel.

5.3 Matching Mechanism Considering Conditions

In this section, we explain the matching mechanism considering conditions in our protocol. Only if two users satisfy their conditions each other, they can obtain their final matching result. More precisely, only if A selects a_c and B selects b_c in their answers, then they can obtain their matching result, when the conditions of A and B are b_c and a_c, respectively. If neither A nor B is satisfied, they cannot obtain any matching result.

We explain an example of our matching mechanism considering conditions. We assume that two users A and B respectively have $a = xdh$ $(x \in C_c)$ and $b = yeh$ $(y \in C_c)$, where x, y, d, e and h are the prime numbers corresponding to answers. Additionally, A and B respectively select $a_c = x \in C_c$ and $b_c = y \in C_c$ as a condition. In this case, since A and B respectively have a_c and b_c in their answers (i.e., they satisfy their conditions each other.), they can obtain the matching result except for conditions a_c and b_c. Users can derive h as their common prime number as follows.

$$\begin{aligned} &E_{pk_A}(a)^{r_{aa}a_c^{-1} \pmod{n_A}} \times E_{pk_A}(b)^{r_{ab}b_c^{-1} \pmod{n_A}} \\ &= E_{pk_A}(r_{aa}xdhx^{-1} + r_{ab}yehy^{-1}) \\ &= E_{pk_A}(r_{aa}dh + r_{ab}eh) \\ &= E_{pk_A}(h(r_{aa}d + r_{ab}e)) \end{aligned} \tag{5}$$

The most important point of this computation is the cancel process of conditions. In Eq. (5), the inverse elements x^{-1} and y^{-1} are canceled by the primes $a_c = x$ and $b_c = y$ for conditions, respectively. From this computation, A and B can know that h is matched between them. If the inverse element of condition is not canceled, the matching result is randomized and hence two users cannot obtain any result.

On the other hand, when the conditions of A and B are respectively $b_c = z \in C_c$ and $a_c = x \in C_c$, the computation by S for a user A is as follows.

$$\begin{aligned} &E_{pk_A}(a)^{r_{aa}a_c^{-1} \pmod{n_A}} \times E_{pk_A}(b)^{r_{ab}b_c^{-1} \pmod{n_A}} \\ &= E_{pk_A}(r_{aa}xdhx^{-1} + r_{ab}yehz^{-1}) \\ &= E_{pk_A}(r_{aa}dh^{-1} + r_{ab}yehz^{-1}) \\ &= E_{pk_A}(\text{random}) \end{aligned} \tag{6}$$

In this case, the inverse element of A's condition z is not canceled since B does not select z. As a result, an overflow occurs with a high probability on message space n_A and thus users cannot obtain the common prime number. Unless the conditions are satisfied, the matching result becomes random.

We can regard our protocol as two-step matching by setting the conditions. At the first step our protocol conducts the matching of conditions, and also at the second step it conducts the matching of the profiles.

Table 2. Comparison of efficiency.

	Computation	Communication
KLC'11 [4]	$\mathcal{O}(\|C\|)$	$\mathcal{O}(\|C\|)$
TLSL'14 [8]	$\mathcal{O}(\|C\|^2)$	$\mathcal{O}(\|C\|)$
ZZSY'12 [9]	$\mathcal{O}(\|C\|)$	$\mathcal{O}(\|C\|)$
Our protocol	$\mathcal{O}(\|C\|)$	$\mathcal{O}(\|C\|)$

6 Evaluation

6.1 Security Analysis

In an honest-but-curious model, we have only to prove correctness and privacy as follows.

Theorem 1 (Correctness). *Our protocol outputs a matching result correctly.*

Proof. When we assume $x \in X_A \cap X_B$, both a and b are divided by x and hence both $X_{AB} = r_{aa}aa_c^{-1} + r_{ab}bb_c^{-1} \pmod{n_A}$ and $X_{BA} = r_{ba}aa_c^{-1} + r_{bb}bb_c^{-1} \pmod{n_B}$ are also divided by x. As a result, each user knows that x is a common answer. On the other hand, when we assume $x \notin X_A \cap X_B$, we can consider two cases: (1) x is included in X_A or X_B, and (2) x is included in neither X_A nor X_B. However, in both cases, x is accidentally existed as a common prime in $X_A \cap X_B$ with a probability of P (see P in Subsect. 6.3). Therefore, our protocol can guarantee the correctness with a failure probability of P. □

We show the following lemma of indeterminate equation.

Lemma 1. *If* $gcd(a, b) = 1$*, then solution* (x, y) *of* $ax + by = 1$ *is existed certainly.*

We will not prove Lemma 1 since this is a famous theorem of indeterminate equation. Using this Lemma, we show that our protocol has privacy as follows.

Theorem 2 (Privacy). *An attacker cannot obtain any information about answers of honest user except for common elements between users.*

Proof. We assume that A is an honest user and another user B is an honest-but-curious attacker. B wishes to know A's answer. B can obtain b, $E_{pk_A}(b)$, $E_{pk_B}(b)$ and $X_{BA} = r_{ba}aa_c^{-1} + r_{bb}bb_c^{-1} \pmod{n_B}$ in the protocol. In order to know the result of A's selection, B needs to know the prime number selected by only A from $X_{BA} = r_{ba}aa_c^{-1} + r_{bb}bb_c^{-1} = \pi(r_{ba}a'a_c^{-1} + r_{bb}b'b_c^{-1}) \pmod{n_B}$, where $a = \pi a'$ and $b = \pi b'$. Note that π is the common prime(s) between A and B. Since B knows X_{BA}, b' and π, B needs a' from following Eq. (7).

$$r_{ba}a' + r_{bb}b' = X_{BA}/\pi \tag{7}$$

$\gcd(a', b') = 1$ holds since the common prime number is not existed in a' and b'. Even if a' has any value in Eq. (7), both r_{ba} and r_{bb} certainly exists from Lemma 1. As a result, it is difficult for an attacker to compute a'. □

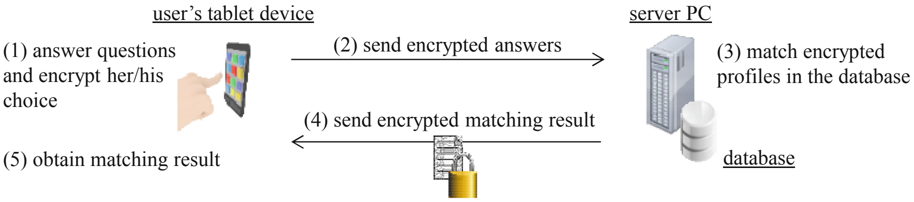

user's tablet device
server PC
(1) answer questions and encrypt her/his choice
(2) send encrypted answers
(3) match encrypted profiles in the database
(4) send encrypted matching result
(5) obtain matching result
database

Fig. 3. Flow of our demonstration.

6.2 Efficiency

We evaluate computational/communication complexities of our protocol on user's tablet device and a server PC. We also evaluate the computation process time of our implemented system.

Complexity of Computation and Communication. Users send and receive two ciphertexts in our protocol. The communication complexity of ciphertext, which each user sends and receives, is $4|C|$. Therefore, the communication complexity of our proposed protocol is denoted as $\mathcal{O}(|C|)$.

We evaluate the computational complexity with the number of modulo exponentiation. In Paillier crypto system, it is required two modulo exponentiations in encryption and one modulo exponentiation in decryption. In our protocol, the number of each user's modulo exponentiation is $5|C|$ since it needs two encryptions and one decryption. Therefore, the computational complexity of our protocol is denoted as $\mathcal{O}(|C|)$.

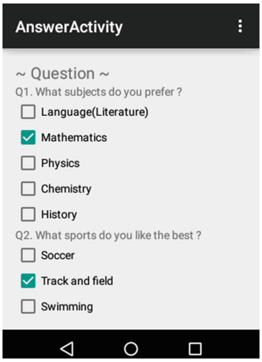
Table 2 shows the comparison the efficiency of the existing schemes and our protocol. We employ the existing schemes that users can know which elements are matched, which is similar to our proposed protocol. The result show that our protocol has lower computational/communication complexities.

Implementation Evaluation. We implemented our proposed protocol in JAVA and evaluated it on a laptop with Intel Core i5 (1.4 GHz) and 8 GB RAM and a tablet device NEXUS 7 with Qualcomm Snapdragon S4 Pro (1.5 GHz) and 2 GB RAM. We evaluated the running time of our protocol on a laptop as a server and a tablet as a user's mobile device. Figure 3 shows the environment of our implementation evaluation. A user constructs her/his own profile by answering some questions. Figure 4 shows the answer window of our implemented application on the tablet device.

We evaluated the running time of two encryptions, one decryption and the verification of matching on a table device, and the matching process on the server PC. Table 3 shows ten times average of running time in each processing.

Table 3. Running time in each processing (10 times average).

Tablet			Server PC
Encryptions (two times)	Decryption (one time)	Matching verification	Matching processing
165 ms	51.3 ms	7.92 ms	30.6 ms



AnswerActivity

~ Question ~

Q1. What subjects do you prefer ?

□ Language(Literature)

☑ Mathematics

□ Physics

□ Chemistry

□ History

Q2. What sports do you like the best ?

□ Soccer

☑ Track and field

□ Swimming

Fig. 4. The answer window of our application on the tablet device NEXUS 7.

6.3 Probability of Failure Matching

We note that our protocol does not deterministically output the matching result. For example, in spite of $x \notin X_A \cap X_B, x \in \mathcal{S}$, if x becomes accidentally the common prime number of A and B, then the matching result becomes wrong. Therefore, it is important that the failure probability is negligible. P is the failure probability that a common t-bit prime number in $X_A \cap X_B$ may be accidentally included in $\mathcal{S}$ as follows.

$$P = 1 - \left(1 - \frac{1}{2^t}\right)^{|\mathcal{S}|} \tag{8}$$

We assume that the message space of E is 1024 bits, i.e., $|n_A| = |n_B| = 1024$. Since the message space is fixed, t and C have the relation of tradeoff. As long

Table 4. The probability of failure matching when t and $|C|$ are changed.

t	$\|C\|$	P
25	34	5.07×10^{-6}
26	33	2.46×10^{-6}
27	31	1.15×10^{-6}
28	30	5.59×10^{-7}
29	29	2.70×10^{-7}
30	28	1.30×10^{-7}
31	27	6.29×10^{-8}

as the probability of failure matching is less than 2^{-20} ($\simeq 10^{-7}$), we assume that the correctness is guaranteed. Table 4 shows the probability of failure matching when t and $|C|$ are changed. In order to satisfy the above condition (i.e., less than 10^{-7}), we set $t = 28$ bits prime numbers and $|C| = 30$ from Table 4. In this implementation, a user selects a single answer from five items assigned to each question.

7 Conclusion

We have proposed a privacy-preserving profile matching protocol considering conditions. In the existing protocols, the unwilling matching may occur, that is, a user may match to another unacceptable user. In our protocol, matching is established only when the conditions are satisfied, and hence our protocol can prevent such unwilling matching, which occurs in the existing protocols, by setting conditions. Additionally, we have reduced computational cost and memory consumption by using the map-to-prime technique and an honest-but-curious server. As a future work, we try to enhance the privacy such as condition hiding and configuration of privacy level, which restrict the matching result.

Acknowledgements. This work was partly supported by Grant-in-Aid for Scientific Research (C) (16K00183).

References

1. Abbas, F., Rajput, U., Hussain, R., Eun, H., Oh, H.: A trustless broker based protocol to discover friends in proximity-based mobile social networks. In: Rhee, K.-H., Yi, J.H. (eds.) WISA 2014. LNCS, vol. 8909, pp. 216–227. Springer, Heidelberg (2015). doi:10.1007/978-3-319-15087-1_17
2. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 1–19. Springer, Heidelberg (2004). doi:10.1007/978-3-540-24676-3_1
3. He, D., Cao, Z., Dong, X., Shen, J.: User self-controllable profile matching for privacy-preserving mobile social networks. IEEE ICCS **2014**, 248–252 (2014)
4. Kim, M., Lee, H.T., Cheon, J.H.: Mutual private set intersection with linear complexity. In: Jung, S., Yung, M. (eds.) WISA 2011. LNCS, vol. 7115, pp. 219–231. Springer, Heidelberg (2012). doi:10.1007/978-3-642-27890-7_18
5. Li, M., Yu, S., Cao, N., Lou, W.: Privacy-preserving distributed profile matching in proximity-based mobile social networks. IEEE Trans. Wirel. Commun. **12**(5), 2024–2033 (2013)
6. Sarpong, S., Xu, C., Zhang, X.: An authenticated privacy-preserving attribute matchmaking protocol for mobile social networks. Int. J. Netw. Secur. **17**(3), 357–364 (2015)
7. Sarpong, S., Xu, C., Zhang, X.: PPAM: privacy-preserving attributes matchmaking protocol for mobile social networks secure against malicious users. Int. J. Netw. Secur. **18**(4), 625–632 (2016)
8. Thapam, A., Li, M., Salinas, S., Li, P.: Asymmetric social proximity based private matching protocols for online social networks. IEEE Trans. Parallel Distrib. Syst. **26**(6), 1547–1559 (2014)

9. Zhang, R., Zhang, Y., Sun, J.S., Yan, G.: Fine-grained private matching for proximity-based mobile social networking. IEEE INFOCOM **2012**, 1–9 (2012)
10. Zhu, H., Du, S., Li, M., Gao, Z.: Fairness-aware and privacy-preserving friend matching protocol in mobile social networks. IEEE Trans. Emerg. Top. Comput. **1**(1), 192–200 (2013)