Privacy-Preserving Group Matching Protocol

Takuya Ibaraki*, Yoshifumi Manabe
Faculty of Informatics, Kogakuin University 1-24-2, Nishi-Shinjuku, Shinjuku, Tokyo, 163-8677 Japan.

* Corresponding author. Email: em16001@ns.kogakuin.ac.jp manabe@cc.kogakuin.ac.jp
Manuscript submitted November 10, 2017; accepted January 18, 2018.
doi: 10.17706/jcp.13.9.1037-1041

Abstract: Many works have been done for privacy-preserving matching protocols. Most of them obtain one-to-one privacy-preserving matching. However, when we consider forming a group of people or objects by their similarity, matching can be applied to problems using many data, such as recommendation systems and a lot of similar communities. In this paper, we consider the characteristics of each user as a vector. We obtain the similarity by securely computing the inner product of vectors. Also, we define a group's characteristics by the members' average characteristics. We propose a privacy-preserving group matching protocol. We show computation cost of the proposed protocol and show simulation results.

Key words: Privacy-preserving computation, group matching, homomorpic encryption.

1. Introduction

Many works have been done for privacy-preserving matching protocols. The private matching protocol begins with Freedman *et. al* proposing the Private Set Intersection(PSI) protocol [1]. PSI is realized using additive homomorphic encryption and Oblivious Polynomial Evaluation(OPE). Zhu *et al.* proposed Blind Vector Transforming (BVT) which makes a set corresponds to a vector [2]. In addition, Kim et. al proposed Map To Prime (MTP) which converts each element of a set to a prime number [3]. In these researches, they protect the user's private information by converting the user's profile into a vector or prime number so as to execute the protocol without directly handling the user's profile.

Most of them assume one-to-one privacy-preserving matching. In this paper, we propose a privacy-preserving matching protocol that forms multiple people as a group. When we consider forming a group of people or objects by their similarity, matching can be applied to problems using many data, such as when there are recommendation systems and a lot of similar communities. In this paper, we consider the characteristics of each user as a vector. We obtain the similarity by securely computing an inner product of vectors. Also, we define a group's characteristics by the members' average characteristics.. We propose a privacy-preserving group matching protocol. We show computation cost of the proposed protocol and show simulation results.

2. Definition

2.1. Notation

The notations are used in this paper is as follows.

Set of users : $I = \{i_1, i_2, \ldots, i_n\}$

Characteristics of user : $r_j = (C_{(j,1)}, C_{(j,2)}, \ldots, C_{(j,p)})$ $(0 \leq C_{(j,p)} \leq 100)$

The characteristic norm of user i_j : $||r_j|| = \sqrt{C^2_{(j,1)}+\ldots+C^2_{(j,p)}}$
Tentative groups : $g = \{g_1,g_2,\ldots,g_a\}$
Final groups : $c = \{c_1,c_2,\ldots,c_b\}$
Threshold of similarity : v
Security parameter: ℓ

2.2. Degree of Similarity

The following formula defines the degree of similarity between user i_j and i_k. The degree of similarity is multiplied by a large constant and rounded off to an integer value. In general, the inner product is used to evaluate the similarity between items to recommend goods [4].

The similarity between groups is obtained by a similar calculation using the average value of the characteristics of users belonging to the group.

$$s(i_j, i_k) = \frac{\sum_{m=1}^{p} C_{(j,m)} C_{(k,m)}}{\sqrt{C^2_{(j,1)} + \cdots + C^2_{(j,p)}} \sqrt{C^2_{(k,1)} + \cdots + C^2_{(k,p)}}} \quad (1)$$

2.3. Encryption Scheme

In this paper, we use an encryption scheme that satisfies additive homomorphic[5]. E (x) is the value obtained by encrypting x. Also, the public key and secret key are generated by the server. The server compares the degree of similarity.

2.4. Semi-Honest Model

Throughout this paper, we assume that the players are semi-honest. All players take actions in accordance with the protocol, but try to obtain another player's secret data. However, the personal characteristics must be disclosed among the members of each group. It is assumed that all communication is secure. Also assume the server is semi-honest to prevent collusion with malicious users.

3. Proposed Protocol

3.1. Preparation

Gonda *et al.* proposed a method to compare the magnitude relation of the two numbers while keeping them secret [6]. They use the fact that the computation result X-Y of the two numbers X, Y becomes a negative number on the finite field when $X<Y$. In addition since the value X-Y is camouflaged by the random numbers α and β, no one knows the value X-Y.

The bit length of modulus N is defined t_N. The bit lengths of random numbers α and β are denoted t_α and t_β, respectively. The bit length of n + ℓ is denoted t'.

3.2. Comparison Protocol of Threshold and Similarity

Gonda *et al*. proposed a protocol to compare the number of matches in vectors [6]. In this paper we use this protocol as a subroutine. Suppose, B obtains an encrypted value from A and computes similarity E(s(A,B)) in advance.

1. B generates random numbers α_b of t_α bits and β_b of t_β bits.
2. B computes $E(\alpha_b \cdot (s(A,B)-v)+ \beta_b)$ to compare the similarity value and the threshold value while keeping it encrypted. This result is sent to the server.
3. The server receives $E(\alpha_b \cdot (s(A,B)-v)+ \beta_b)$ sent from B. The server decrypts it to obtain $\alpha_b \cdot (s(A,B)-v)+$

β_b. The server verifies the bit length. If the result t_N bit, the server sends false to B otherwise sends true to *B*.

The result is determined by the bit length, so $s(A,B)$ is never known to anyone.

3.3. Comparison Protocol of Similarities

In 3.2, we compared the threshold and the similarity. It is possible to compare similarities between each users by a similar procedure. C obtains encrypted values from A and B, and computes each the similarities E(s(A,C)) and E(s(B,C)) in advance.

1. C generates random numbers α_c of t_α bits and β_c of t_β bits.
2. C computes $E(\alpha_c \cdot (s(A,C)-s(B,C))+ \beta_c)$ to compare the similarity values while keeping the similarity values encrypted. This result is sent to the server.
3. The server receives $E(\alpha_c \cdot (s(A,C)-s(B,C))+ \beta_c)$ sent from C. The server decrypts it to obtain $\alpha_c \cdot (s(A,C)-s(B,C))+ \beta_c$. The server verifies the bit length. If the result t_N bit, the server sends false to C otherwise sends true to C. True means that s(A,C) is larger than s(B,C). False means that s(B,C) is larger than s(A,C).

   The result is determined by the bit length, so s(A,C) and s(B,C) are never known to anyone.

Property 1[6]. The protocol correctly outputs the result when the bit length of each parameter satisfies the following conditions.

$$\begin{aligned} t_\alpha \cdot t' + t_\beta &< t_N - 1 \\ t_\alpha &> t_\beta \end{aligned} \tag{2}$$

3.4. Our Main Protocol

1. Each user i_n computes $||r_n||$. Let a be the number of tentative groups. Initially, set a=1.
2. Let a new tentative group $g_a=\{i_k\}$ (k is the minimum index of the user who does not belong to any tentative group.) . User i_k computes $E(C_{(k.1)}/||r_k||),\ldots,E(C_{(k,p)}/||r_k||)$and sends them to the users who do not belong to any tentative group. Note that $C_{(k,p)}/||r_k||$ are integer values.
3. User i_m computes $E(C_{(k.1)}/||r_k||)\times(C_{(m,1)}/||r_m||)+\ldots+E(C_{(k,p)}/||r_k||)\times(C_{(m,p)}/||r_m||)$ using the homomorphism of the encryption scheme. This result is $E(s(i_k,i_m))$.
4. i_m compares the similarity and the threshold v using the algorithm in 3.2
5. If the similarity is more than the threshold v, i_m belongs to the tentative group g_a.
6. The tentative group g_a updates the average value of the characteristics. Set it the representative value of the group.
7. If all users do not belong to some tentative group, then set $a = a+1$ and return to step 2.
8. The first b tentative groups $g_1,\ldots,g_b$ obtained in steps 1 to 7 are denoted $c_1,\ldots,c_b$.
9. Repeat the following step 10-15 from $q=b+1$ to $q=a$.
10. Each group $c_i \in\{c_1,\ldots,c_b\}$ computes $||r_i||$ as in step1.
11. Group $c_i \in\{c_1,\ldots,c_b\}$ computes $E(C_{(i.1)}/||r_i||),\ldots,E(C_{(i,p)}/||r_i||)$and sends them to the users who do not belong to any tentative group.
12. Group g_q computes $E(C_{(i.1)}/||r_i||)\times(C_{(q,1)}/||r_q||)+\ldots+E(C_{(i,p)}/||r_i||)\times(C_{(q,p)}/||r_q||)$. This reslult is $E(s(c_i,g_q))$.
13. g_q finds the group with the highest similarity from $s(c_1,g_q),\ldots,s(c_b,g_q)$ by repeatedly using the algorithm of 3.3.
14. g_q belongs to the group c_b having the highest similarity.
15. c_b updates the average value of the characteristics.

4. Discussion

In this section, we evaluate the proposed protocol using simulation. The simulation was made in C language. Set the number of users n = 100. Set the dimension of the user's characteristics p = 10. Also, the characteristics of each user are generated by random numbers. We show the average result of 1000 executions each with changing the threshold from 0.85 to 0.95 and the group number from 3 to 7.

Table 1. Average Tentative Group Number

Threshold	0.85	0.86	0.87	0.88	0.89	0.90	0.91	0.92	0.93	0.94	0.95
Tentative group number	20.69	23.58	26.82	30.93	35.73	41.61	48.40	56.29	65.39	74.78	83.92

Table 2. Average Variance

Threshold	0.85	0.86	0.87	0.88	0.89	0.90	0.91	0.92	0.93	0.94	0.95
Final group number b=3	6868	6835	6875	6870	6875	6930	6974	6975	7017	7012	7017
Final group number b=4	6396	6383	6341	6351	6360	6413	6465	6491	6520	6535	6557
Final group number b=5	5981	5980	5921	5952	5959	5959	6011	6059	6099	6135	6133
Final group number b=6	5622	5423	5579	5555	5566	5576	5642	5664	5732	5771	5812
Final group number b=7	5314	5298	5220	5275	5251	5257	5278	5355	5401	5471	5500

Table 1 shows the average of the number of tentative groups. It can be seen that the tentative group number increases with the threshold value. Table 2 shows the average variance of the final groups. From Table 2, it can be seen that the average variance value decreases as the final group number increases. Also, it can be seen that the average variance increases slightly as the threshold increases.

First, we consider the computation cost. The proposed protocol performs matching in two stages. The first stage depends on the number of users. In the worst case that all tentative groups consist of one person, the computation cost is $O(n^2)$. For the second stage, computation cost is determined by the number of tentative groups. Each tentative group computes the similarity value with each final group. The computation cost is $O(n)$. Thus, it is better to have fewer tentative groups, since the amount of computation cost can be reduced by decreasing the number of tentative groups. However, if the threshold is set too small, the average variance becomes large. It is necessary to set the threshold appropriately.

Second, we consider the relationship between the threshold and average variance. In the simulation results, the average variance is the lowest for the cases when the number of in the final groups is 4-7 if the threshold value is 0.87. Therefore, it is necessary to appropriately set the threshold in order to reduce the average variance. Random networks often become scale-free networks, thus randomly-generated users' similarity (and users' similarity in the real world) can be modeled as a scale-free network[7]. The scale-free networks have a structure in which similar users are formed as clusters and the connections between clusters are sparse. Therefore, we consider that it is better to merge groups of moderate sizes than merging groups whose sizes are too small. Thus, if the threshold is too large, the variance of the final groups becomes large. On the other hand, if the threshold is too small, tentative groups consist of not so similar users, thus the variance increases. Thus the threshold value must be set appropriately. In the simulation results, when the threshold value was 0.86-0.87, the average variance was the minimum value in each of the final groups 3-7.

5. Conclusion

In this paper, we considered the user's average characteristics as group characteristics and proposed a privacy-preserving group matching protocol. We showed that computation cost of the propose protocol is $O(n^2)$. We then showed the relationship between the threshold value, the number of tentative groups and

the average variance using simulation. The threshold value was 0.86-0.87, the average variance was the minimum value in each of the final groups 3-7. Therefore, it is necessary to appropriately set the threshold in order to reduce the average variance.

Future challenges is the propose of the protocol that is reduced computation cost, and the method of a better similarity evaluation.

References

[1] Freedman, M. J., Nissim, K., & Pinkas, B. (2004). Efficient private matching and set intersection. *Eurocrypt, 3027*, 1-19.

[2] Zhu, H. J., Du, S. G., Li, M. Y., & Gao, Z. Y. (2013). Fairness-aware and privacy-preserving friend matching protocol in mobile social networks. *IEEE Transactions on Emerging Topic in Computing, 1(1),* 192-200.

[3] Kim, M., Tae Lee, H., & Hee Cheon, J. (2011). Mutual private set intersection with linear complexity. *WISA 2011,* 219-231.

[4] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J., (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web(WWW10)* (pp. 285-295). Hong Kong.

[5] Paillier, P. (1999). Public-key cryptosystems based on composite degree RESIDUOSITY classes. *Advances in Cryptology-EUROCRYPT 1999*, 223-238.

[6] Gonda, A., & Omote, K. (2016). Consideration of privacy-preserving matching protocol with threshold. *Proceedings of Symposium on Cryptography and Information Security 2016.*

[7] Barabási, A. L., Albert, R., & Jeong, H. (2000). Scale-free characteristics of random networks: The topology of the world-wide web. *Physica A, 281,* 2115.

Takuya Ibaraki was born in Tokyo, Japan in 1993. He received his bachelor of informatics from Kogakuin University, Japan, in 2016. Currently, he is a master course student in the Faculty of Informatics, Kogakuin University. His research interests include privacy-preserving computation.

Yoshifumi Manabe was born in Osaka, Japan in 1960. He received B.E., M.E. and Dr. E. degrees from Osaka University, Osaka, Japan in 1983, 1985, and 1993, respectivery.

From 1985 to 2013, he worked for Nippon Telegraph and Telephone Corporation. He was a guest associate professor of Kyoto University in 2001-2013. His research interest includes cryptography, distributed algorithms, and game theory. Dr. Manabe is a member of ACM, IEEE, IPSJ, JSIAM and IEICE.