A Double-Private ϵ-fuzzy Matching Protocol

Yuta Urushiyama and Yoshifumi Manabe
Department of Computer Science, Faculty of Informatics
Kogakuin University
1-24-2, Nishi-Shinjuku, Shinjuku, Tokyo, 163-8677 Japan
Email: manabe@cc.kogakuin.ac.jp

***Abstract*—This paper proposes a new double-private protocol for fuzzy matching and ϵ-fuzzy matching. Many works have been done for private database search in which the keyword that a user inputs for the search is concealed to the database owner. In these database searches, the exactly matched data are returned to the user. Fuzzy matching has been proposed in which not exactly matched but nearly matched data are returned to the user. Then the condition to be matched is further relaxed by ϵ-fuzzy matching.**

In fuzzy matching and ϵ-fuzzy matching, a new security requirement, the security of the database can be considered. The database owner just answers the existence of a matched data without showing the matched data itself. This paper first formalizes the problem as the double-private ϵ-fuzzy matching. We show a naive protocol and an efficient protocol for double-private ϵ-fuzzy matching.

Keywords*—*cryptographic protocols, database retrieval, fuzzy matching, ϵ-fuzzy matching, double-private matching

I. Introduction

This paper proposes a new double-private protocol for fuzzy matching and ϵ-fuzzy matching. Many works have been done for private database search in which the keyword that a user inputs for the search is concealed to the database owner [1]–[8]. In these database searches, the exactly matched data are returned to the user. Fuzzy matching has been proposed in which not exactly matched but nearly matched data are returned to the user [9]. Then the condition to be matched is further relaxed by ϵ-fuzzy matching [10].

In fuzzy matching and ϵ-fuzzy matching, a new security requirement, the security of the database can be considered. The database owner just answers the existence of a matched data without showing the matched data itself. In usual database searches, answering the existence of a matched data is just the same as answering the matched data itself, thus the security of the database cannot be achieved anyway. In fuzzy matching and ϵ-fuzzy matching, there are several cases when the database owner wants to hide the information which data is matched to the search. For example, consider a pharmaceutical company's database of patients' genome information to whom a new medicine was effective. A client doctor wants to know the new medicine is effective or not to his current patient by the search using his patient's genome. If his patient's genome is close to an entry in the database, the possibility that the new medicine is effective to his patient is high. The doctor wants to conceal the genome information to the pharmaceutical company, because the genome information is private information of his patient. The pharmaceutical company does not want to give the genome information in the database to the doctor because the genome data is private information. The pharmaceutical company wants to give the information whether there is a genome data that is close to the doctor's input genome data in the database or not. No other information in the database must be given to the doctor. Thus, the client data must not be known to the server and the server data must not be known to the client either. We call this property double-private.

This paper first formalizes the problem as the double-private ϵ-fuzzy matching. The proposed protocols for fuzzy matching [11] and ϵ-fuzzy matching [10] are not double-private, since they answer the matched data itself to the client. We show a naive protocol and an efficient protocol for double-private ϵ-fuzzy matching.

Related works are as follows. Fuzzy matching has been first considered in [9], however, the protocol in [9] was shown to be incorrect [11]. [11] showed a fuzzy private matching protocol. More efficient fuzzy private matching protocols have been shown [12], [13]. [14] showed that branching programs can be used to improve efficiency of fuzzy private matching. ϵ-fuzzy matching protocols have been shown in [10]. Application of fuzzy private matching to fingerprint matching [15] and fuzzy keyword search [16] have been considered.

The rest of the paper is organized as follows. Section II presents the definition of ϵ-fuzzy matching and double-privacy. Section III shows a naive double-private ϵ-fuzzy matching protocol. Section IV presents an improved double-private ϵ-fuzzy matching protocol. Section V summarizes the paper.

II. ϵ-fuzzy matching

This section gives the definition of ϵ-fuzzy matching problem. The set of data in client C be $X = \{X_1, X_2, \ldots, X_m\}$ and the set of data in server S be $Y = \{Y_1, Y_2, \ldots, Y_n\}$. Each data $X_i = (x_i^1, x_i^2, \ldots, x_i^T)$ and $Y_j = (y_j^1, y_j^2, \ldots, y_j^T)$ is a T-dimensional vector whose elements $x_i^k, y_j^k \in \mathbb{Z}_p (1 \leq k \leq T)$. Note that for a set X, the number of elements in X is denoted by $|X|$.

Fuzzy matching is defined as follows in [9].

Definition 1: For some threshold $t \leq T$, X_i and Y_j are fuzzy matching if $t \leq |\{k | x_i^k = y_j^k\}|$ is satisfied and denoted as $X_i \sim_t Y_j$.

Definition 2: For a data X_i and a set Y, $X_i \sim_t Y$ if there is an element $Y_j \in Y$ that satisfies $X_i \sim_t Y_j$.

Definition 3: For two sets X and Y, $X \sim_t Y$ if there is a pair of elementw $X_i \in X$ and $Y_j \in Y$ that satisfy $X_i \sim_t Y_j$. For given X and Y, the set $\{Y_j \in Y | \exists X_i \in X, X_i \sim_t Y_j\}$ is denoted as $Sim_t(X; Y)$.

Then, ϵ-fuzzy matching is defined as follows [10].

Definition 4: For some threshold $t \leq T$ and some $\epsilon \geq 0$, X_i and Y_j are ϵ-fuzzy matching if $t \leq |\{k||x_i^k - y_j^k| \leq \epsilon\}|$ is satisfied and denoted as $X_i \sim_{\epsilon,t} Y_j$.

Next, ϵ-fuzzy matching between a data and a set is defined as follows.

Definition 5: For a data X_i and a set Y, $X_i \sim_{\epsilon,t} Y$ if there is an element $Y_j \in Y$ that satisfies $X_i \sim_{\epsilon,t} Y_j$.

Last, ϵ-fuzzy matching between two sets is defined as follows.

Definition 6: For two sets X and Y, $X \sim_{\epsilon,t} Y$ if there is a pair of elements $X_i \in X$ and $Y_j \in Y$ that satisfy $X_i \sim_{\epsilon,t} Y_j$. For given X and Y, the set $\{Y_j \in Y | \exists X_i \in X, X_i \sim_{\epsilon,t} Y_j\}$ is denoted as $Sim_{\epsilon,t}(X;Y)$.

A private fuzzy matching protocol has been proposed in [9], in which the input and output of the protocol are defined as follows.

- (Input) Client C: data set X
  Server S: data set Y
- (Output) C: $Sim_t(X;Y)$
  S: (none)

As the generalization to ϵ-fuzzy matching, a private ϵ-fuzzy matching protocol has been proposed in [10], whose inputs and outputs are as follows.

- (Input) Client C: data set X
  Server S: data set Y
- (Output) C: $Sim_{\epsilon,t}(X;Y)$
  S: (none)

As shown in the introduction, there are several cases when the server wants to hide the information which data is matched to X. Thus, we propose a double-private ϵ-fuzzy matching protocol whose inputs and outputs are as follows.

- (Input) Client C: data set X
  Server S: data set Y
- (Output) C: Whether $X \sim_{\epsilon,t} Y$ or not (that is, whether $|Sim_{\epsilon,t}(X;Y)| > 0$ or not)
  S: (none)

The security of double-private ϵ-fuzzy matching protocol is defined as follows. This paper assumes that all parties are honest but curious, that is, all parties act according to the protocol, but they try to obtain extra information using all the information obtained during the execution of the protocol.

- The client's security: The server obtains no information from the protocol, that is, the server cannot distinguish the two cases when the client has two different inputs.
  Formally, the security is defined by the following game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$.
  Adversary $\mathcal{A}$ selects two client inputs X_0, X_1 and server input Y where $|X_0| = |X_1|$. $\mathcal{A}$ sends X_0 and X_1 to challenger $\mathcal{C}$.
  $\mathcal{C}$ randomly selects bit $b \in \{0,1\}$. X_b is given to an honest client and Y is given to $\mathcal{A}$. The honest client and server $\mathcal{A}$ execute the protocol.
  After the protocol execution, $\mathcal{A}$ outputs bit b'. $\mathcal{A}$'s advantage
  $$Adv_c(\mathcal{A}) = |Pr[b = b'] - 1/2|.$$
  $\mathcal{A}$ wins the game if the advantage is not negligible.
- The server's security: The client obtains no information from the protocol other than whether $X \sim_{\epsilon,t} Y$ or not.
  Formally, the security is defined by the following game between adversary $\mathcal{A}$ and challenger $\mathcal{C}$.
  Adversary $\mathcal{A}$ selects two data sets Y_0, Y_1 and input X, where $|Y_0| = |Y_1|$ and
  ($X \sim_{\epsilon,t} Y_0$ and $X \sim_{\epsilon,t} Y_1$) or
  ($X \not\sim_{\epsilon,t} Y_0$ and $X \not\sim_{\epsilon,t} Y_1$) holds.
  $\mathcal{A}$ sends Y_0 and Y_1 to challenger $\mathcal{C}$.
  $\mathcal{C}$ randomly selects bit $b \in \{0,1\}$. Y_b is given to an honest server and X is given to $\mathcal{A}$. Client $\mathcal{A}$ and the honest server S execute the protocol.
  After the protocol execution, $\mathcal{A}$ outputs bit b'. $\mathcal{A}$'s advantage
  $$Adv_s(\mathcal{A}) = |Pr[b = b'] - 1/2|.$$
  $\mathcal{A}$ wins the game if the advantage is not negligible.

An ϵ-fuzzy matching protocol is double-private if both of $Adv_c(\mathcal{A})$ and $Adv_s(\mathcal{A})$ are negligible.

To the best of our knowledge, no fuzzy matching protocols or ϵ-fuzzy matching protocols are double-private. The fuzzy matching protocol [11] which outputs $Sim_{\epsilon,t}(X;Y)$ uses an additively homomorphic public-key encryption, a symmetric key encryption, and a t-out of-T secret sharing scheme.

Additively homomorphic public key encryption consists of the following algorithms.

- Key generation Gen : $(pk, sk) \leftarrow Gen(1^l)$
- Encryption Enc: $c \leftarrow Enc(pk, m)$
- Decryption Dec: $Dec(sk, c) = m$
  if $c \leftarrow Enc(pk, m)$.
- Additive homomorphism: three is an operation $+$ on two ciphertexts. When $c = c_1 + c_2$,
  $$Dec(sk, c) = m_1 + m_2$$
  if $c_i \leftarrow Enc(pk, m_i)(i = 1, 2)$.

Using the homomorphism for multiple times, it is possible to calculate $Enc(pk, a \cdot m)$ for given $c \leftarrow Enc(pk, m)$ and an integer $a > 0$.

t-out of-T secret sharing scheme is as follows. For a secret data s, a set of shares $U = \{v_1, v_2, \ldots, v_T\}$ is calculated such that from any set $U' \subset U$ that satisfies $|U'| \geq t$, the secret data s can be recovered. In addition, no information about s can be obtained from any set $U'' \subset U$ that satisfies $|U''| < t$.

The outline of the simple fuzzy matching protocol in [11] is as follows. Note that the ϵ-fuzzy matching protocol in [10] is based on the same idea.

1) Client C generates $(pk, sk) \leftarrow Gen(1^l)$ of an additive homomorphic public key encryption. C gives pk to server S.

2) For each $X_i = (x_i^1, x_i^2, \ldots, x_i^T) \in X$, C generates ciphertexts

$$c_i^w = Enc(pk, x_i^w)(1 \leq w \leq T).$$

C sends $c_i^w (1 \leq w \leq T)$ to S.

3) For each $Y_j \in Y$ and $X_i \in X$, execute the following steps.

a) S generates a key k_j of a symmetric key encryption (Enc_S, Dec_S), calculates

$$C_j = Enc_S(k_j, Y_j),$$

and sends C_j to S.

b) For the secret $0^k||k_j$, S generates a set of shares $\{v_j^1, v_j^2, \ldots, v_j^T\}$ by a t-out of-T secret sharing scheme, where $||$ is concatenation.
S then calculates the following ciphertexts using homomorphism of the encryption.

$$c_w = Enc\left(pk, r_w \cdot (x_i^w - y_j^w) + v_j^w\right)$$

$(1 \leq w \leq T)$, where r_w are random values generated by S. S sends all ciphertexts to C.

c) C executes

$$p_w = Dec(sk, c_w)(1 \leq w \leq T).$$

C can recover the secret $0^k||k_j$ from the plaintexts if $X_i \sim_t Y_j$. C obtains Y_j by

$$Dec_S(k_j, C_j).$$

Note that the protocol is not double-private since C obtains the matched data.

III. A NAIVE PROTOCOL

A naive double-private ϵ-fuzzy matching protocol is as follows, whose main idea is the same as the (not double-private) ϵ-fuzzy matching protocol in [11].

1) Client C generates $(pk, sk) \leftarrow Gen(1^l)$ of an additive homomorphic public key encryption. C gives pk to server S.

2) Server S generates the set

$$Z = \{Z_k | Z_k \sim_{\epsilon,t} Y_j \in Y\}.$$

Note that if the pair of parameters (ϵ, t) is the same for all requests, S needs to calculate the set Z just once in advance.

3) For each $X_i = (x_i^1, x_i^2, \ldots, x_i^T) \in X$, C generates ciphertexts

$$Enc\left(pk, (x_i^1)^{\gamma_1} \cdot (x_i^2)^{\gamma_2} \cdot \ldots \cdot (x_i^T)^{\gamma_T}\right),$$

for every tuple $(\gamma_1, \gamma_2, \ldots, \gamma_T)$ that satisfies

$$1 \leq \sum_{w=1}^{T} \gamma_w \leq mn\binom{T}{t}(2\epsilon + 1)^t$$

and $\gamma_w \geq 0 (1 \leq w \leq T)$.
C sends all ciphertexts to S.

4) S calculates the following ciphertext using homomorphism of the encryption.

$$C_Z = Enc\left(pk, \prod_{X_i \in X} \prod_{Z_k \in Z} \left(\sum_{w=1}^{T} r_{z_{k,w}}(x_i^w - z_k^w)\right)\right),$$

where $X_i = (x_i^1, x_i^2, \ldots, x_i^T)$, $Z_k = (z_k^1, z_k^2, \ldots, z_k^T)$, and $r_{z_{k,w}}$ are random values generated by S.
S sends C_Z to C.

5) C executes $P_Z = Dec(sk, C_Z)$. C obtains 0 if $X \sim_{\epsilon,t} Y$, otherwise C obtains a random value.

If $(x_i^1, x_i^2, \ldots, x_i^T) = (z_k^1, z_k^2, \ldots, z_k^T)$ then

$$\sum_{w=1}^{T} r_{z_{k,w}}(x_i^w - z_k^w) = 0,$$

otherwise the sum is a random value. Thus, C_Z is an encryption of 0 if there is a pair $X_i \in X$ and $Z_k \in Z$ such that $X_i = Z_k$. Otherwise, C_Z is an encryption of a random value. Since $Z = \{Z_k | Z_k \sim_{\epsilon,t} Y_j \in Y\}$, double-private ϵ-fuzzy matching is achieved.

Note that the encryption is addively homorphic (fully homomorphic encryption is not practical yet), thus S needs to calculate the expanded form of the equation at step 4). Therefore, C needs to send

$$Enc\left(pk, (x_i^1)^{\gamma_1} \cdot (x_i^2)^{\gamma_2} \cdot \ldots \cdot (x_i^T)^{\gamma_T}\right)$$

to S. Since $|X| = m$ and $|Z| \leq n\binom{T}{t}(2\epsilon + 1)^t$, the maximum degree of the equation is at most $mn\binom{T}{t}(2\epsilon + 1)^t$, and the number of terms is at most $(T+1)^{mn\binom{T}{t}(2\epsilon+1)^t}$.

IV. AN IMPROVED PROTOCOL

The maximum degree of the polynomial of the above naive protocol is very large. Thus, a simplication of the protocol is necessary. We show another protocol below.

1) Let $\alpha = \{1, 2, \ldots, T\}$. Client C generates $(pk, sk) \leftarrow Gen(1^l)$ of an additive homomorphic public key encryption. C gives pk to server S.

2) For each $X_i = (x_i^1, x_i^2, \ldots, x_i^T) \in X$, C generates ciphertexts

$$Enc\left(pk, (x_i^{\beta_1})^{\delta_{\beta_1}} \cdot (x_i^{\beta_2})^{\delta_{\beta_2}} \cdot \ldots \cdot (x_i^{\beta_{T-t+1}})^{\delta_{\beta_{T-t+1}}}\right)$$

for every $\beta = \{\beta_1, \ldots, \beta_{T-t+1}\} \subseteq \alpha$ and
$0 \leq \delta_{\beta_i} \leq 2\epsilon + 1 (1 \leq i \leq T - t + 1)$.
C sends all ciphertexts to S.

3) Server S calculates the following ciphertext using homomorphism of the encryption.

$$C'_Z = Enc(pk,$$

$$\prod_{\substack{X_i \in X \\ Y_j \in Y}} \left(\sum_{\substack{\forall \beta \subseteq \alpha \\ |\beta| = T-t+1}} r_{y_{j,\beta}} \prod_{w \in \beta} \prod_{c=-\varepsilon}^{\varepsilon} (x_i^w - (y_j^w + c))\right)\Bigg),$$

where $r_{y_{j,\beta}}$ are random values generated by S.
S sends C'_Z to C.

4) C executes

$$P'_Z = Dec(sk, C'_Z).$$

C obtains 0 if $X \sim_{\epsilon,t} Y$, otherwise C obtains a random value.

If $x_i^{w_k} = y_j^{w_k}$ is satisfied for at least t different w_k's among T elements, any term

$$(x_i^{w_1} - y_j^{w_1})(x_i^{w_2} - y_j^{w_2}) \cdots (x_i^{w_{T-t+1}} - y_j^{w_{T-t+1}})$$

is 0, thus the sum of these terms is also 0. If $x_i^{w_k} = y_j^{w_k}$ is satisfied for less than t different w_k's among T elements, at least one term

$$(x_i^{w_1} - y_j^{w_1})(x_i^{w_2} - y_j^{w_2}) \cdots (x_i^{w_{T-t+1}} - y_j^{w_{T-t+1}})$$

among $\binom{T}{T-t+1}$ terms is not 0, thus the sum of these terms is not 0. Each term $(x_i^{w_k} - y_j^{w_k})$ in the above equation is replaced with

$$(x_i^{w_k} - (y_j^{w_k} + \epsilon))(x_i^{w_k} - (y_j^{w_k} + \epsilon - 1)) \cdots (x_i^{w_k} - (y_j^{w_k} - \epsilon))$$

to allow the difference at most ϵ, then the equation in the protocol is derived.

The maximum degree of the polynomials of this protocol is at most $mn(2\varepsilon+1)(T-t+1)$ and the number of terms is at most $((2\epsilon+2)\binom{T}{T-t+1})^{mn}$. The maximum degree is smaller than $mn\binom{T}{t}(2\epsilon+1)^t$ by the naive protocol.

V. Conclusion

This paper proposed a new protocol for ϵ-fuzzy matching. The proposed protocol is double-private, that is, the client obtains no information other than whether there is a ϵ-fuzzy matching. A further study includes considering malicious parties.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 26330019.

References

[1] W. Gasarch, "A survey on private information retrieval," *Bulletin of the EATCS*, vol. 82, pp. 72–107, 2004.

[2] S. Yekhanin, "Private information retrieval," *Commun. ACM*, vol. 53, no. 4, pp. 68–73, Apr. 2010. [Online]. Available: http://doi.acm.org/10.1145/1721654.1721674

[3] R. Ostrovsky and W. E. Skeith III, "A survey of single-database private information retrieval: Techniques and applications," in *Public Key Cryptography–PKC 2007*. Springer, 2007, pp. 393–411.

[4] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, no. 6, pp. 965–981, Nov. 1998. [Online]. Available: http://doi.acm.org/10.1145/293347.293350

[5] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylogarithmic communication," in *Advances in Cryptology EUROCRYPT 99*, ser. Lecture Notes in Computer Science, J. Stern, Ed. Springer Berlin Heidelberg, 1999, vol. 1592, pp. 402–414. [Online]. Available: http://dx.doi.org/10.1007/3-540-48910-X_28

[6] Y. Gertner, Y. Ishai, E. Kushilevitz, and T. Malkin, "Protecting data privacy in private information retrieval schemes," in *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, ser. STOC '98. New York, NY, USA: ACM, 1998, pp. 151–160. [Online]. Available: http://doi.acm.org/10.1145/276698.276723

[7] G. Di Crescenzo, T. Malkin, and R. Ostrovsky, "Single database private information retrieval implies oblivious transfer," in *Advances in Cryptology EUROCRYPT 2000*, ser. Lecture Notes in Computer Science, B. Preneel, Ed. Springer Berlin Heidelberg, 2000, vol. 1807, pp. 122–138. [Online]. Available: http://dx.doi.org/10.1007/3-540-45539-6_10

[8] A. Beimel, Y. Ishai, and T. Malkin, "Reducing the servers computation in private information retrieval: Pir with preprocessing," in *Advances in Cryptology CRYPTO 2000*, ser. Lecture Notes in Computer Science, M. Bellare, Ed. Springer Berlin Heidelberg, 2000, vol. 1880, pp. 55–73. [Online]. Available: http://dx.doi.org/10.1007/3-540-44598-6_4

[9] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology-EUROCRYPT 2004*. Springer, 2004, pp. 1–19.

[10] S. Ogawa and T. Satoh, "Private fuzzy matching protocols," in *30th Symposium on Cryptography and Information Security*. IEICE, 2013, pp. 1C–1(In Japanese).

[11] L. Chmielewski and J.-H. Hoepman, "Fuzzy private matching," in *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*. IEEE, 2008, pp. 327–334.

[12] J. M. Doumen, "Non-interactive fuzzy private matching," http://eprints.eemcs.utwente.nl/10738/, Centre for Telematics and Information Technology University of Twente, Enschede, Technical Report TR-CTIT-07-45, June 2007.

[13] Q. Ye, R. Steinfeld, J. Pieprzyk, and H. Wang, "Efficient fuzzy matching and intersection on private datasets," in *Information, Security and Cryptology–ICISC 2009*. Springer, 2010, pp. 211–228.

[14] H. Lipmaa, "Private branching programs: On communication-efficient cryptocomputing." *IACR Cryptology ePrint Archive*, vol. 2008, p. 107, 2008.

[15] S. F. Shahandashti, R. Safavi-Naini, and P. Ogunbona, "Private fingerprint matching," in *Information Security and Privacy*. Springer, 2012, pp. 426–433.

[16] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–5.