Research on Secure Multi-party Closest String Problem

Liu Wen
School of Computer Science
Communication University of China
Beijing,China
lw8206@gmail.com

Wang Yong-bin
School of Computer Science
Communication University of China
Beijing,China

Cao Yi-zhen
School of Computer Science
Communication University of China
Beijing,China

***Abstract*—A secure two-party hamming distance computing protocol is proposed based on the secure two-party matrix product protocol; a secure four-party addition protocol, a secure three-party addition protocol and a secure two-party number protocol are proposed based on the additive homomorphic encryption and secure two-party compare protocol. A secure multi-party closest string protocol is presented based on the secure two-party hamming distance computing protocol and a secure multi-party addition protocol. These protocols are based on semi-honest model.**

***Keywords*-secure two-party hamming distance computing protocol; secure multi-party addition protocol; secure multi-party closest string protocol;**

I. INTRODUCTION

Secure multi-party computation is that there are n parties, $P_1, P_2, ..., P_n$. They all have a private input $x_1, x_2, ..., x_n$. They want to privately compute a function $f(x_1, x_2, ..., x_n) = (y_1, y_2, ..., y_n)$ and party P_i gets output y_i. The meanings of security is that any party P_i cannot get any other information except the information of y_i, x_i. This problem is introduced by computer scientist Andrew C. Yao in Ref. [1]. Then Glodreich extended the research of secure multi-party computation in Ref. [2][3]. Now the problem of secure multi-party computation is became one of hot problems in cryptography. Many reseachers devote to propose new solutions of different secure multi-party computation problems.

Given n length-l strings $s_1, s_2, ..., s_n$, and a radius d, the closest string problem seeks for a new length-m string s such that $d(s, s_i) \leq d$. Here $d(,)$ is the Hamming distance. This problem has a variety of applications in computational biology and coding theory. The closest string problem has been recently studied in Ref. [4][5][6] in order to improve the computation complexity. In this paper, we study the security of the closest string problem. The secure multi-party closest string problem is described as follows: supposed that there are n parties, $P_1, P_2, ..., P_n$. Each party P_i owns a length-l string $S_i (i = 1, 2, ..., n)$ based on alphabet Σ. n parties want to compute the common closest substring of their n strings. But they don't want to leak their private string $S_i (i = 1, 2, ..., n)$ to others parties. The secure multi-party closest string problem has strong applications.

The contribute in our paper is as follows:

(1)A secure two-party Hamming distance protocol is proposed based on secure two-party matrix multiplication protocol.

(2)A secure four-party, there-party and two-party numbers addition comparing protocol is proposed based on the homomorphic addition encryption and the secure two-party comparing protocol.

(3)A secure multi-party closest string protocol is proposed based on the secure two-party Hamming distance protocol and the secure four-party, there-party and two-party numbers addition comparing protocols.

We supposed that all parties are semi-honest who follow the protocol properly with the exception that it keeps a record of all its intermediate computations. In particular, each party should not gain more information about the others parties than what follows from his secret inputs and his result.

II. PRELIMINARIES

A. Secure Two-party Matrix Multiplication Protocol

The secure two-party matrix multiplication problem is described that Alice has a matrix A, Bob has a matrix B. Alice and Bob want to get the result $A \times B$ but one party doesn't want to leak his matrix information to the other party. In Ref.[7], Du gave a secure two-party matrix multiplication protocol.

B. Semantic Secure Additive Homomorphic Encryption Scheme

We use a semantic secure public-key encryption scheme that has the additive homomorphic property. Supposed that the encryption algorithm is $E(.)$ and the corresponding decryption algorithm is $D(.)$, where encryption key is public and decryption key is private. Plaintext message space is $M \subseteq Z$. The encryption algorithm $E(.)$ has the following properties:

(1)Semantic security: For any two plaintext messages $m_1, m_2 \in M$, there is not any polynomial time algorithm to distinguish $E(m_1)$ and $E(m_2)$;

(2) Additive homomorphism: For any two plaintext messages $m_1, m_2 \in M$ and any $k \in Z$, if $m_1 + m_2 \in M$

978-1-4244-6252-0/11/$26.00 ©2011 IEEE

and $km_1 \in M$, then $D(E(m_1)E(m_2)) = m_1 + m_2$ and $D(E(m_1)^k) = km_1$.

C. Secure Two-party Comparing Protocol

The secure two-party comparing problem is described that Alice has a private number m and Bob has a private number n. Alice and Bob try to compare m and n privately without leaking their private number. In Ref.[1][8][9], some secure two-party comparing protocols were proposed.

III. Basic Protocol

A. Secure Two-party Hamming Distance Protocol

Supposed that Alice has a private length-l string $S_a = s_{a1}s_{a2}...s_{al}$ and Bob has a private length-l string $S_b = s_{b1}s_{b2}...s_{bl}$ based on a alphabet Σ, where the alphabet Σ includes n letters $\{letter_1, letter_2, ..., letter_n\}$. Two parties want to compute the Hamming distance $d_H(S_a, S_b)$ of S_a, S_b without leaking the information about S_a, S_b.

The protocol is as follows:

(1)Alice transforms the length-l string S_a into a $n \times l$ matrix $A_{n\times l} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1l} \\ a_{21} & \cdots & \cdots & a_{2l-1} & a_{2l} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & \cdots & a_{nl} \end{bmatrix}$, where if $s_{ai} = letter_j$, then $a_{ji} = 1$; otherwise $a_{ji} = 0$.

Bob transforms the length-l string S_b into a $n \times l$ matrix $B_{n\times l} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1l} \\ b_{21} & \cdots & \cdots & b_{2l-1} & b_{2l} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{n1} & b_{n2} & \cdots & \cdots & b_{nl} \end{bmatrix}$, where if $s_{bi} = letter_j$, then $b_{ji} = 1$; otherwise $b_{ji} = 0$.

(2)Alice and Bob use secure two-party matrix multiplication protocol to calculate $A \times B^T$. Alice gets a n-dimension vector $R_a = (r_{a1}, r_{a2}, ..., r_{an})$ and Bob gets a n-dimension vector $R_b = (r_{b1}, r_{b2}, ..., r_{bn})$, where $R_a + R_b = A \times B^T$.

(3)Alice calculates $r_a = \frac{l}{2} - \sum_{i=1}^{n} r_{ai}$ and Bob calculates $r_b = \frac{l}{2} - \sum_{i=1}^{n} r_{bi}$, where $d_H(S_a, S_b) = r_a + r_b$.

In this protocol, Alice and Bob transform their private strings into matrixes A and B and the problem transform into the problem of $A \times B^T$. Alice gets n-dimension vector $R_a = (r_{a1}, r_{a2}, ..., r_{an})$ and Bob gets n-dimension vector $R_b = (r_{b1}, r_{b2}, ..., r_{bn})$. $r_{ai}+r_{bi}$ denotes the number of equal numbers in the ith row of A, B. $\sum_{i=1}^{n} r_{ai} + \sum_{i=1}^{n} r_{bi}$ denotes the number of equal letters in strings of S_a, S_b. The Hamming distance of S_a, S_b is $d_H(S_a, S_b) = l - (\sum_{i=1}^{n} r_{ai} + \sum_{i=1}^{n} r_{bi})$. The correctness of this protocol is analyzed.

The security of this protocol id depend on the security of secure two-party matrix multiplication protocol.

B. Secure Multi-party Numbers Addition Comparing Protocol

1) Secure Four-party Numbers Addition Comparing Protocol: Supposed that there are four parties P_1, P_2, P_3, P_4. They own a private number $R_{P_1a_1}, R_{P_1a_2}, R_{P_1b_1}, R_{P_1b_2}$. They want to compare $R_{P_1a_1} + R_{P_1b_1}$ and $R_{P_1a_2} + R_{P_1b_2}$ without leaking their private numbers.

P_1, P_3, P_4 know a public key pk and own a part of private key sk of a additive homomorphic encryption scheme .

The protocol is described as follows:

(1)P_1, P_3, P_4 use their public key pk and get $E_{pk}(R_{P_1a_1}), E_{pk}(R_{P_1b_1}), E_{pk}(R_{P_1b_2})$. P_3, P_4 send $E_{pk}(R_{P_1b_1}), E_{pk}(R_{P_1b_2})$ to P_1;

(2)P_1 calculates $E_{pk}(R_{P_1a_1}) + E_{pk}(R_{P_1b_1}) - E_{pk}(R_{P_1b_2})$ and sends it to P_4;

(3)P_4 uses sk_4 to calculate $D_{sk_4}(E_{pk}(R_{P_1a_1} + R_{P_1b_1} - R_{P_1b_2}))$ and sends it to P_3; P_3 uses sk_3 to calculate $D_{sk_3sk_4}(E_{pk}(R_{P_1a_1} + R_{P_1b_1} - R_{P_1b_2}))$ and sends it to P_1; P_1 uses sk_1 to calculate $D_{sk_1sk_3sk_4}(E_{pk}(R_{P_1a_1} + R_{P_1b_1} - R_{P_1b_2}))$ and gets $R_{P_1a_1} + R_{P_1b_1} - R_{P_1b_2}$;

(4)P_1, P_2 uses secure two-party comparing protocol to compare $R_{P_1a_1} + R_{P_1b_1} - R_{P_1b_2}$ and $R_{P_1a_2}$: if $R_{P_1a_1} + R_{P_1b_1} - R_{P_1b_2} > R_{P_1a_2}$, then they get $R_{P_1a_1} + R_{P_1b_1} > R_{P_1b_2} + R_{P_1a_2}$; otherwise $R_{P_1a_1} + R_{P_1b_1} \leq R_{P_1b_2} + R_{P_1a_2}$. P_1, P_2 tell others parties the result.

In this protocol, P_1 gets $E_{pk}(R_{P_1b_1}), E_{pk}(R_{P_1b_2}), R_{P_1a_1} + R_{P_1b_1} - R_{P_1b_2}$. Because of the semantic security of additive homomorphic encryption scheme, P_1 cannot get any information $R_{P_1b_1}, R_{P_1b_2}$ from $E_{pk}(R_{P_1b_1}), E_{pk}(R_{P_1b_2})$; P_1 only knows $R_{P_1b_1} - R_{P_1b_2}$ and cannot get exact information about $R_{P_1b_1}, R_{P_1b_2}$ in $R_{P_1a_1} + R_{P_1b_1} - R_{P_1b_2}$; P_2 cannot get the private information of P_1, P_3, P_4 because of the security of secure two-party comparing protocol; Because of the semantic security of additive homomorphic encryption scheme, P_3, P_4 also cannot get information from $E_{pk}(R_{P_1a_1} + R_{P_1b_1} - R_{P_1b_2})$ and $D_{sk_4}(E_{pk}(R_{P_1a_1} + R_{P_1b_1} - R_{P_1b_2}))$. So our protocol is secure.

2) Secure Three-party Numbers Addition Comparing Protocol: Supposed that P_1 has two private numbers $R_{P_1a_1}$ and $R_{P_1a_2}$, P_2 has a private number $R_{P_1b_1}$ and P_3 has a private number $R_{P_1b_2}$. They want to compare $R_{P_1a_1} + R_{P_1b_1}$ and $R_{P_1a_2} + R_{P_1b_2}$ without leaking their private numbers.

P_1, P_2 know a public key pk and P_2 owns the private key sk of a additive homomorphic encryption scheme .

The protocol is described as follows:

(1)P_1 uses pk and gets $E_{pk}(R_{P_1a_1} - R_{P_1a_2})$ and sends it to P_2;

(2)P_2 calculates $D_{sk}(E_{pk}(R_{P_1a_1} - R_{P_1a_2}))$ and gets $R_{P_1b_1} + R_{P_1a_1} - R_{P_1a_2}$;

(3)P_2, P_3 uses secure two-party comparing protocol to compare $R_{P_1b_1} + R_{P_1a_1} - R_{P_1a_2}$ and $R_{P_1b_2}$: if $R_{P_1b_1} + R_{P_1a_1} - R_{P_1a_2} > R_{P_1b_2}$, then they get $R_{P_1a_1} + R_{P_1b_1} >$

$R_{P_1b_2} + R_{P_1a_2}$; otherwise $R_{P_1a_1} + R_{P_1b_1} \leq R_{P_1b_2} + R_{P_1a_2}$. P_2, P_3 tell P_1 the result .

In this protocol, P_1 only knows the result; P_2 only knows $E_{pk}(R_{P_1a_1} - R_{P_1a_2})$ and cannot get exact information about $R_{P_1b_1}, R_{P_1b_2}$; P_3 cannot get the private information of P_1, P_2 because of the security of secure two-party comparing protocol. So our protocol is secure.

3) Secure Tow-party Numbers Addition Comparing Protocol: Supposed that P_1 has two private numbers $R_{P_1a_1}$ and $R_{P_1b_2}$, P_2 has a private number $R_{P_1b_1}$ and $R_{P_1a_2}$. They want to compare $R_{P_1a_1} + R_{P_1b_1}$ and $R_{P_1a_2} + R_{P_1b_2}$ without leaking their private numbers.

The protocol is described as follows:

P_1 calculates $R_{P_1a_1} - R_{P_1b_2}$ and P_2 calculates $R_{P_1a_2} - R_{P_1b_1}$. P_1, P_2 uses secure two-party comparing protocol to compare $R_{P_1a_1} - R_{P_1b_2}$ and $R_{P_1a_2} - R_{P_1b_1}$: if $R_{P_1a_1} - R_{P_1b_2} > R_{P_1a_2} - R_{P_1b_1}$, then they get $R_{P_1a_1} + R_{P_1b_1} > R_{P_1b_2} + R_{P_1a_2}$; otherwise $R_{P_1a_1} + R_{P_1b_1} \leq R_{P_1b_2} + R_{P_1a_2}$.

The security of this protocol is depend on the security of secure two-party comparing protocol.

IV. Secure Multi-party Closest String Protocol

In this section, we first present the definition of secure multi-party closest string problem, then use the secure two-party Hamming distance protocol and the secure multi-party numbers addition comparing protocol to solve the problem.

A. Problem Definition

Supposed that there are n parties, $P_1, P_2, ..., P_n$. Each party P_i owns a length-l string $S_i (i = 1, 2, ..., n)$ based on alphabet Σ. n parties want to compute the common closest substring of their n strings, where the Hamming distance $d_H(S, S_i)$ of S and S_i satisfies that $d_H(S, S_i) \leq d, i = 1, ..., n$. But they don't want to leak their private string $S_i (i = 1, 2, ..., n)$ to others parties.

B. Secure Multi-party Closest String Protocol

The secure multi-party closest string protocol can be divided into three parts: (1)$P_i (i = 1, 2, ..., n)$ calculates the Hamming distance $d_H(S_i, S_j)$ of S_i and $S_j, j = 1, 2, ..., i-1, i+1, ..., n$; (2)$P_i (i = 1, 2, ..., n)$ calculates the maximum $d_H max(i)$of $d_H(S_i, S_j), j = 1, 2, ..., i-1, i+1, ..., n$; (3) $P_1, P_2, ..., P_n$ calculates the minimum $d_H min$ of $d_H max(1), d_H max(2), ..., d_H max(n)$.

The protocol is described as follows:

(1)For $i = 1, 2, ..., n$,

(1.1)P_i and $P_j (j = 1, 2, ..., i_1, i+1, ..., n)$ use secure two-party Hamming distance protocol to calculate $d_H(S_i, S_j)$. P_i gets $R_{P_ia_1}, R_{P_ia_2}, ..., R_{P_ia_{i-1}}, R_{P_ia_{i+1}}, ..., R_{P_ia_n}$ and $P_j (j = 1, ..., i-1, i+1, ..., n)$ gets $R_{P_ib_j}$, where $d_H(S_i, S_j) = R_{P_ia_j} + R_{P_ib_j}$.

(1.2)$P_i (i = 1, 2, ..., n)$ uses secure three-party numbers addition comparing protocol to get $\max\{d_H(S_i, S_1), d_H(S_i, S_2), ..., d_H(S_i, S_{i-1}), d_H(S_i, S_{i+1}), ..., d_H(S_i, S_n)\}$.

For $k = 1, 2, ..., i-1, i+1, ..., n-2$, P_i uses $R_{P_ia_k}, R_{P_ia_{k+1}}$, P_k uses $R_{P_ib_k}$ and P_{k+1} uses $R_{P_ib_{k+1}}$ to execute secure three-party numbers addition comparing protocol to compare $R_{P_ia_k} + R_{P_ib_k}$ and $R_{P_ia_{k+1}} + R_{P_ib_{k+1}}$. Then P_i continues to compare the maximum and $R_{P_ib_{k+2}}$.

(2)After step(1), P_1 gets the maximum Hamming distance $d_H(S_1, S_{k_1}) = R_{P_1a_{k_1}} + R_{P_1b_{k_1}}$, where P_1 owns $R_{P_1a_{k_1}}$ and P_{k_1} owns $R_{P_1b_{k_1}}$;...;P_n gets the maximum Hamming distance $d_H(S_n, S_{k_n}) = R_{P_na_{k_n}} + R_{P_nb_{k_n}}$, where P_n owns $R_{P_na_{k_n}}$ and P_{k_n} owns $R_{P_nb_{k_n}}$. $P_1, P_2, ..., P_n$ use secure multi-party numbers addition comparing protocol to calculate $\min\{d_H(S_1, S_{k_1}) = R_{P_1a_{k_1}} + R_{P_1b_{k_1}}, d_H(S_2, S_{k_2}) = R_{P_2a_{k_2}} + R_{P_2b_{k_2}}, ..., d_H(S_n, S_{k_n}) = R_{P_na_{k_n}} + R_{P_nb_{k_n}}\}$. P_i, P_{i+1} get the maximum Hamming distance according to step(1). They choose the maximum to compare with the maximum Hamming distance of P_{i+2} until $i = n-2$.

The correctness and security of our protocol depends on the correctness and security of secure two-party Hamming distance protocol and secure multi-party numbers addition comparing protocol.

V. Conclusion

In this paper, we propose a secure multi-party closest string protocol based on the secure two-party Hamming distance protocol and the secure four-party, there-party and two-party numbers addition comparing protocols. The secure multi-party closest string protocol can be applied in computational biology. And it's necessary to research this problem.

We have to point out that our protocol is not effective. We will try to study the protocol with high efficiency.

Acknowledgment

This paper is supported by Beijing Municipal Special Fund for Cultural and Creative Industries(2009); the Engineering Course Programming Project of Communication University of China, Grant No.XNG0925; the National "211" Development Fund for Key Engineering Programs; and the Beijing Municipal Natural Science Foundation (4112052).

References

[1] A Yao. Protocols for secure computation [A]. Proceeding of the 23th IEEE Symposium on Foundations of Computer Science [C]. Los Alamitos, CA: IEEE Computer Society Press, 1982. 160-164.

[2] O. Goldreich, S. Micali, A. Wigderson, How to play any mental game, in: Proc. the 19th Annual ACM Symposium on Theory ofComputing, New York, 1987, pp. 218-229.

[3] O. Goldreich, Foundations of Cryptography: Basic Applications, Cambridge University Press, London, 2004.

[4] M.Franees, A.Litman. On covering problems of codes. Theory of Computing Systems, 1997,30:113-119

[5] J.K.Lanctot, M.Li, B.Ma, S.Wang, L.Zhang. Distinguishing string selection problems. In:Proc.of the 10th ACM-SIAM Syposium on Discrete Algorithms(SODA), ACM Press, 1999. 633-642

[6] F.Nicolas, E.Rivals. Complexities of the centre and median string problems. In Proc.of the Fourteenth Annual Syposium on Combinatorial Pattern Matching (CPM, 03), volume2676 of LNCS, 2003. 315-327

[7] Du Wenliang. A Study of Several Specific Secure Two-party Computation Problems. Department of Computer Sciences, Purdue University, 2001.

[8] Ioannidis I, Grama A. An efficient protocol for Yao's millionaires'problem. In: Proceedings of the 36th Hawaii International Conference on System Science. Los Alamitos: IEEE Computer Society Press, 2003. 205

[9] Li S D, Dai Y Q, You Q Y. Efficient secure multiparty computation solution to Yao's millionaires' problem based on set inclusion. Prog Nat Sci, 2005, 15(9): 851-856