Comparison over Encrypted Data for Privacy Preservation Based on Homomorphic Encryption

[1] Vijayendra S. Gaikwad, [2]Amit A. Kadam, [3]Jitendra C. Musale, [4]Rahul B. Diwate
[1]Assistant Professor, [2]Assistant Professor, [3]Assistant Professor, [4]Assistant Professor
Department of Computer Engineering,
APCOER, Pune, India

Abstract: Now a days, with data mining computation being performed by cloud servers it is a problem to securely determining whether x>y, given two input values x, y, which are held as private inputs by two parties, respectively. The output which is result of comparison becomes known to both parties. In this paper we consider a variant of comparison problem in which the inputs x, y are encrypted and the actual values are not known to the parties. Our solution deals with single comparison; however, in many applications, we encounter situations where it is necessary to make multiple comparisons to find the maximum among several encrypted data, so we make a modification to our protocol to solve the multiple comparisons problem. Such a secure comparison is an important building block for applications like privacy preserving data mining and secure business. Also our protocols can be performed in constant rounds and do not use general circuit evaluation techniques so they are more efficient than circuit based ones but not general. Implementation is easy and fast.

***Index Terms*–Secure two party computation; multiple comparisons, Encrypted data**

I. INTRODUCTION

About two decades ago, Yao introduced the “millionaire problem” [1]: Two parties want to determine who is richer without disclosing anything else about their wealth. Several solutions have been proposed for this problem; however, none of them consider our issue. In addition there are some limitations with Yao's Millionaire Problem; we call this problem YMP in short in this paper.

YMP is applicable with just one comparison, i.e. a comparison between two numbers. However, we may face lots of situations where there are several numbers (e.g n numbers) to be compared. This problem cannot be solved by just using the basic Yao’s Millionaire solution n-1 times, once for each pairs. For two reasons this approach is not suggested. Applying Yao's solution several times not only reduces the performance but also affects on privacy, because that would inappropriately reveal the relative ordering between pairs which causes information leakage.

Another limitation; YMP does not considered situation where encrypted data should be compared and actual values are not known to parties.

To explain the issue which is addressed in this paper, consider the following scenario:

Alice and Bob owning private databases wish to run a decision tree algorithm on the union of their databases. Since the databases are confidential, neither of them is willing to disclose any of the contents to the other. Lindell and Pinkas have shown that how to construct decision tree by cooperation without disclosing of private data in [10]. As we know the important step to build a decision tree is choosing best attribute for branching, for example in ID3 algorithm the best attribute is chosen based on Information gain. After two parties compute information gain for each attribute in privacy preserving manner, it is time to determine the maximum information gain.

Alice acts as server and works over his data and Bob's encrypted data and computes information gain in encrypted form while Bob owns the decryption key. Due to privacy considerations Alice would not send all encrypted values to Bob. So they should run a protocol to find the attribute with maximum information gain.

Another requirement is that the actual values of Information gain should not be clear to any parties and should not going to be revealed at the end of protocol, only the result (selected attribute) is important for decision.

We call this problem, "comparison over encrypted data" in this paper. "Comparison over encrypted data" can be extended into three issues that we propose solutions for each one.

1. Alice and Bob want to compute a function on their private inputs; the initial output is two cipher texts (C1 and C2). The problem is to securely determine whether actual value of C1 or C2 is greater. Our solution to this problem is general.
2. According to mentioned scenario; sometimes we have more than two numbers (n) to be compared. The requirement of not allowing each party to know any partial information about the individual comparisons immediately rules out using our first solution n-1 times. A new protocol is proposed to solve such "comparison *over encrypted data"* problem by extending the first solution.

"Comparison over encrypted data" is animportant building block for applications such as privacy preserving data mining and secures business. For example two companies (A , B) determine to aware of the product with highest sale. A has an n-dimensional vector A= (a_1, …, a_n) and B has another n-dimensional vector B=(b_1, , b_n). a_i and b_i show the number of sales for a specific product. B encrypts its elements and sends to A then A adds his values to received values via Homomorphic encryption. After that

they should run the *"comparison over encrypted data"* protocol. Due to privacy the actual values should not be revealed even the maximum. Only the relevant product with highest sale isknown.

Another typical example would be secure distributed database mining. The setting is as follows: several parties, each having a private database, wish to determine some properties of, or perform computations on their joint database. Many interesting properties and computations, such as classification by decision tree (mentioned above), needs comparing severalvalues.

Our protocols solve the "*comparison over encrypted data"* problem directly by analyzing the special properties of the problem and do not use Boolean circuit gates; so the solutions are more efficient but not general, only "Greater Than" function (GT) is considered. Because of the large size of the databases, even a minor efficiency gain in computing GT, results in significant performance improvements.

The rest of this paper is organized as follows. In section 2,wereviewrelatedwork.In section3weexplain thedetails of our solutions. Proposed protocols are evaluated insection 4. Finally we conclude the in section 5 and bring up some futurework.

II. RELATED WORK

Secure two party computations was first introduced by Yao [8], and then it was generalized to multi-party computation.

Authors in [10] states that most of secure two party computation protocols for a function *f*use a similar methodology where the function *f* to be computed is first represented as a combinatorial circuit, and then the parties run a short protocol for every gate in the circuit. Although circuit based methods are appealing due to their generality, the complexity protocols are dependent on the size of the circuit.

This size depends on the size of the input and on the complexityof *f* as a circuit. As we know, input data might be huge as in data mining applications. For example, a naïve multiplication circuit is quadratic in the size of its inputs [11].Following [10] let's call this general approach as "generic protocol".

Secure two-party computation of small circuits with small inputs may be practical using protocol introduced in [8]. As the significant point of our paper is the encrypted input, the previous solutions are not applicable here.

Due to inefficiency of generic protocols, some researchers have focused on finding efficient protocols for specific problems in secure computation (e.g. *comparison over encrypted data* problem in this paper). We continue this trend for the problem of *comparison over encrypted data.*

There appear to be a few works in the literature which consider encrypted inputs for comparison, which no solution is available for solving "*comparison over encrypted data*" problem yet.

A variant of YMP, which takes as input two lists of encrypted bits representing x and y, respectively is considered in [2, 3]. Our problem is to compare two encrypted scalar numbers. In addition their protocols are based on Boolean circuit; while our problem setting is different from using circuit such that makes it easy and fast for implementation. {Computation complexity in worst case is O (n×N) which encryption system is not additively Homomorphic.}

There are many other protocols with improvement to YMP basic solution, which their challenge is to securely determine either x>y or x<=y, where x is Alice's private input and y is Bob's private input. Examples of such approaches are presented in [4, 5, 6, 7]. In contrast to our solutions, the actual encrypted inputs are known to the parties running the protocol, and in our described scenario we wish to find the maximum over multiple encrypted data.

Avariant of oblivious transfer technique is used in [4, 6]. Protocol presented in [5] is based on quadratic residuosity assumption. In [7] YMP is solved through reducing the problem to the intersection problem of two sets applying a special coding for the private inputs. Their technique is based on Homomorphic encryption.

III. OUR SOLUTIONS

We are working in the semi honest model that means the parties correctly follow the protocol specification, yet they may attempt to learn additional information by analyzing received messages during the execution.

For definiteness, suppose Alice has E(x) and E(y), which are encrypted with Bob's public key. We need a protocol to decide whether x<y, such that this is the only thing they know in the end and the actual values are not revealed.

Let $E_{pk\text{-}Bob}$ be the encryption function under public key of Bob and $D_{sk\text{-}Bob}$be the decryption function under secret key ofBob.

In the following, we present a protocol to solve the *"comparison over encrypted data"* problem. It is applicable to all public key encryption schemes.

The protocols proceed as follows:

A. Protocol1:General"ComparisonoverEncrypted Data"

1) Alice has two encrypted values ($E_{pk\text{-}Bob}(x)$,$E_{pk\text{-}Bob}(y)$).

2) Alice picks a random integer (1<i<100) and compute thevalues:
   - $k= E_{pk\text{-}Bob}(x)-i$
   - $k'=E_{pk\text{-}Bob}(y)-i$.

3) Alice sends Bob k andk'.

4) Bob computes privately the values of $Z_u= D_{sk\text{-}Bob}(E_{pk\text{-}Bob}(x)- i+u)$ for u=1,2,….,100.
   When u=i then $Z_i= D_{sk\text{-}Bob}(E_{pk\text{-}Bob}(x)- i+u)= x$

5) Bobcomputesprivatelythevaluesof$Z'_u=D_{sk\text{-}Bob}(E_{pk\text{-}Bob}(y)- i+u)$ for u=1,2,….,100.

When u=i then $Z_i= D_{sk\text{-}Bob}(E_{pk\text{-}Bob}(y)- i+u)= y$

6) Bob generate a random number R and computes $Z_u= Z_u+R$, $Z'_{u+}R$ forallu;In other words Bob adds noise to values to hide the actual values of x and y from Alice.

7) Bob sends the following two arrays with size of 100 to Alice: Z_u and$Z'_{u.}$

8) Alice looks at i-th number of each arrays sent by Bob, and compares two values without knowing the real values of x and y then decide the greaterone.

9) Alice tells the Bob what the conclusionis.

B. Protocol 1': General "Multiple Comparison over EncryptedData"

The second issue of *Comparison over Encrypted Data* problem introduced at the introduction of paper was when we wish to compare multiple encrypted data (n numbers). Here we apply a modification to the first protocol to solve multiple comparison issue asfollowing:

1) Alice has n encrypted values ($E_{pk\text{-}Bob}(x_1)$, ….,$E_{pk\text{-}Bob}(x_n)$).

2) Alice picks a random integer (1<i<100) and compute the valuesof:
  - $k_1 = E_{pk\text{-}Bob}(x_1)$-i
  - ….
  - $k_n=E_{pk\text{-}Bob}(x_n)$-i.

3) Alice sends Bob k_1, …,k_n.

4) In contrast the first solution that Bob computes two arrays for each value, here he should compute an array for every value (n arrays); Z_{1u}, …., Z_{nu}

5) Bob generates random number R and computes $Z_{1u}= Z_{1u}+R$, …,$Z_{nu=}Z_{nu}+ R$ for allu.

6) Bob sends arrays in size of 100 toAlice.

7) Alice looks at i-thnumber of each array sent by Bob, and compares values without knowing the real values of numbers then decide the greaterone.

8) Alice tells the Bob what the conclusionis.

IV. PROTOCOL EVALUATION

Thefirstprotocolisdoneintworoundswhilethe second one complete in only one round. In first round of protocol 1, Alice sends Bob two values which random number of iis added to them. In second round Bob sends Alice two arrays with size of about 100 foreach.

The significant point in this two rounds protocol is the number of computations done at Bob's side. Number of computations depends on the range which we are authorized to choose i (i is used in step2). We assume size of the range equal to N.Note that, if we determine N a large number so the number of computations becomes more and has side effect on performance.

Ontheotherhand,Bob canguesstheactualvalueofxand ywithprobabilityof1/N,becausehedecryptsreceivedvalue for all u=1,….,N and in the u=i the decryption result is exact value of x or y. according to this if we determine N small to reduce the number of computations, the information leakage occurs with high probability, so we should considera tradeoff between performance and privacy. In this paper we assume N=100.As the number of computations depends on N the computation complexity of first protocol is O(n×N), n is the number of encrypted data. If we have only two encrypted data for comparison n=2. Totaly, the first protocol is said to take lineartime.

V. CONCLUSION AND FUTURE WORK

In this paper we proposed new and practical protocols for "*comparison over encrypted data*" problem. The first protocol is general and is not depend on properties of public key cryptosystem, so we need two rounds of data communication between parties. The computation complexity is O(k×n) that algorithm is said to take linear time.

The proposed protocols are simple and fast in implementation; more important is that, they are easy to understand and do not need complex background of cryptography and number theory.

As a part of our future work, is to apply protocols in real systems e.g. in privacy preserving data mining applications where only the result of comparison is important.

REFERENCES

[1] A. C. Yao, "Protocols for secure computations," presented at the Proceedings of the 23rd Annual Symposium on Foundations of Computer Science,1982.

[2] J. Garay, et al., "Practical and secure solutions for integer comparison," presented at the Proceedings of the 10th international conference on Practice and theory in public-key cryptography, Beijing, China,2007.

[3] B. Schoenmakers and P. Tuyls. Practical two-party computation based on the conditional gate. In Advances in Cryptology— ASIACRYPT ’04, volume 3329 of Lecture Notes in Computer Science, pages 119–136, Berlin, 2004.Springer-Verlag.

[4] G. Di Crescenzo. Private Selective Payment Protocols. In FC ’00: Proc. 4th International Conference on Financial Cryptography, Lecture Notes in Computer Science, pages 72–89, London, 2001, Springer-Verlag.

[5] M. Fischlin. A cost-effective pay-per-multiplication comparison method for millionaires. In Progress in Cryptology – CT-RSA 2001, volume 2020 of Lecture Notes in Computer Science, pages 457–471, Berlin, 2001. SpringerVerlag.

[6] I. Blake and V. Kolesnikov. Strong conditional oblivious transfer and computing on intervals. In Advances in Cryptology—ASIACRYPT '04, volume 3329 of Lecture Notes in Computer Science, pages 515– 529, Berlin, 2004.Springer-Verlag.

[7] H. Lin and W. Tzeng. An efficient solution to the millionaires' problem based on homomorphic encryption. In ACNS 2005, volume 3531 of Lecture Notes in Computer Science, pages 456–466. Springer-Verlag,2005.

[8] A. C. Yao, How to generate and exchange secrets, Proceedings 27th Symposium on Foundations of Computer Science (FOCS), IEEE, 1986, pp.162–167.

[9] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Advances in Cryptography EUROCRYPT '99, pp 223-238, Prague, Czech Republic, 1999.

[10] Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining," presented at the Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology,2000.

[11] V. Kapoor, et al., "Privacy preserving sequential pattern mining in distributed databases," presented at the Proceedings of the 15th ACM international conference on Information and knowledge management, Arlington, Virginia, USA,2006.