Anti-Collusion Privacy-Preserving Data Mining

楊 斌[†] 中川 裕志[‡]

東京大学 情報理工学研究科[†] 東京大学 情報基盤センター[‡]

Abstract

Up to now, numerous methods of privacy-preserving data mining have been proposed. Most of them are under the assumption of semi-honest and without collusion, which means that every party follows the protocol properly with the exception that it keeps a record of all its intermediate computations but does not share the record to any others. In this study, we consider the problem of multiparty secure computation of some functions on secure summations of data spread around multiple parties. In particular, we focus our attention on the problem of collusions, in which some parties may collude to deduce the privacy of other parties. We propose a new method satisfying a high level security, full-privacy. By this method, nothing in a party is revealed even though all other parties collude. This method is also an efficient one with running time of $O(m)$, and it can solve a large number of problems in privacy-preserving data mining.

1. Introduction

In recent years, with the increased concerns on privacy protection, a number of techniques have been suggested in order to perform data mining with the private information preserved. The goal of privacy-preserving data mining with m parties is to collaborate to compute some function of their inputs without any party in the system learning any additional information except the result functions and its own inputs.

Many conclusions in privacy-preserving data mining have been obtained by researchers, and most of them are under an assumption that each party is semi-honest which means the party follows the protocol properly with the exception that it keeps a record of all its intermediate computation results and tries to deduce additional information from them other than the protocol result. Moreover, they also assume that every party does not collude to share its record with any other party. However, in many practical problems, a party may not trust any other one. For example, consider a case in which all members securely compute some function of values spread out them, and all the members can freely join in or quit. Obviously, in such a case, no one can guarantee that each party will not collude to other parties. Accordingly, we consider the collusion problem and propose a symmetric protocol called SPoS protocol, which satisfies (m-1)-private. Here, a protocol is called t-private, if no coalition containing at most t parties can get any additional information from its execution.

1.1 Related Work

Recently, there have been numerous propositions in privacy-preserving data mining, such as [1] (2005), [2] (2007), [3] (2008) and [4] (2003). Similar to our protocol, all of them are to solve the problem of secure computation of some function on secure summations of values spreading around multiparty. Unfortunately, in these methods, if some of the parties collude, they may get some private information of other parties.

† Bin Yang, Graduate School of Information Science and Technology, The University of Tokyo
‡ Hiroshi Nakagawa, Information Technology Center, The University of Tokyo

1.2 Our Contributions

The problem of computation of function on secure summations is a general problem in distributed computation and privacy-preserving data mining. Our proposed method enhances the security against collusions of parties. It satisfies full-privacy and hence a secure communication channel is unnecessary. It is also an efficient one whose running time is just $O(m)$. Therefore, it can be used as a common tool to solve a large number of problems of privacy-preserving data mining.

2. Problem Formulation

From many examples of privacy-preserving data mining, we notice that the secure computation of a function of summations of the data spread around m parties is a classical problem in privacy-preserving data mining. Without loss of generality, we just describe the case in which the number of parties is m=3.

Since the goal of our main protocol is to compute the Secure Product of (two) Summations, we abbreviate it as "SPoS".

Problem 1 Secure Product of Summation Protocol

Input:
Each party has two numbers:
Party *1*: ${}^{1}\chi$, ${}^{1}y \in (0,1)$;
Party *2*: ${}^{2}\chi$, ${}^{2}y \in (0,1)$;
Party *3*: ${}^{3}\chi$, ${}^{3}y \in (0,1)$.
Output:
These *3* parties collaborate to securely compute

$$p = \chi y = ({}^{1}\chi+{}^{2}\chi+{}^{3}\chi)({}^{1}y+{}^{2}y+{}^{3}y)$$

Condition: (The case of party 1)
Party 2 and 3 cannot get any information about ${}^{1}\chi$, ${}^{1}y$, χ and y, even though they collude.
(The cases of party 2 and 3 are similar.)

This protocol can be also used to solve the problem of secure computation of the secure ratios of summations and the problem of secure comparison of summations.

3. Protocol

3.1 Secure Linear Function Evaluation Protocol

We firstly introduce a sub-protocol, Secure Linear Function Evaluation or SLFE, which would be used in our main protocols. By this protocol, the value of a linear function is securely computed. Figure 1 shows this protocol in detail.

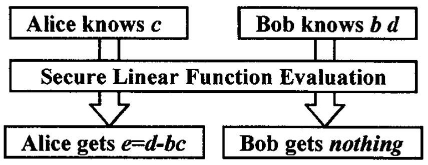

Figure 1. Secure Linear Function Evaluation Protocol.

3.2 Secure Product of Summations Protocol

In the following, we use the index in the left-up of a variable to express the party where the variable generated.

3.2.1 Stage 1-Generating Parameters

Firstly, each party randomly generates three δs satisfying the following conditions.

$$\begin{gathered} {}^1\delta_1+{}^1\delta_2+{}^1\delta_3=0 \\ {}^2\delta_1+{}^2\delta_2+{}^2\delta_3=0 \\ {}^3\delta_1+{}^3\delta_2+{}^3\delta_3=0 \end{gathered}$$

Then using SLFE protocol, each party randomly generates three βs satisfying the following conditions.

$$\begin{gathered} ({}^1\beta_1+{}^2\beta_1+{}^3\beta_1)\,{}^1\gamma={}^1\delta_1+{}^2\delta_1+{}^3\delta_1 \\ ({}^1\beta_2+{}^2\beta_2+{}^3\beta_2)\,{}^2\gamma={}^1\delta_2+{}^2\delta_2+{}^3\delta_2 \\ ({}^1\beta_3+{}^2\beta_3+{}^3\beta_3)\,{}^3\gamma={}^1\delta_3+{}^2\delta_3+{}^3\delta_3 \end{gathered}$$

3.2.2 Stage 2-Computing the Result

Each party computes three αs using the random numbers βs as following. Since βs are random numbers, αs do not contain any information about the inputs χs.

$$\begin{array}{llll} \textbf{Party 1:} & {}^1\chi+{}^1\beta_1\rightarrow{}^1\alpha_1, & {}^1\chi+{}^1\beta_2\rightarrow{}^1\alpha_2, & {}^1\chi+{}^1\beta_3\rightarrow{}^1\alpha_3 \\ \textbf{Party 2:} & {}^2\chi+{}^2\beta_1\rightarrow{}^2\alpha_1, & {}^2\chi+{}^2\beta_2\rightarrow{}^2\alpha_2, & {}^2\chi+{}^2\beta_3\rightarrow{}^2\alpha_3 \\ \textbf{Party 3:} & {}^3\chi+{}^3\beta_1\rightarrow{}^3\alpha_1, & {}^3\chi+{}^3\beta_2\rightarrow{}^3\alpha_2, & {}^3\chi+{}^3\beta_3\rightarrow{}^3\alpha_3 \end{array}$$

Each party sends its three αs to each party respectively, and then each party computes p by received numbers as following.

$$\begin{array}{llcl} \textbf{Party 1:} & {}^1\alpha_1,\ {}^2\alpha_1,\ {}^3\alpha_1 & & ({}^1\alpha_1+{}^2\alpha_1+{}^3\alpha_1)\,{}^1\gamma\rightarrow{}^1p \\ \textbf{Party 2:} & {}^1\alpha_2,\ {}^2\alpha_2,\ {}^3\alpha_2 & \Rightarrow & ({}^1\alpha_2+{}^2\alpha_2+{}^3\alpha_2)\,{}^2\gamma\rightarrow{}^2p \\ \textbf{Party 3:} & {}^1\alpha_3,\ {}^2\alpha_3,\ {}^3\alpha_3 & & ({}^1\alpha_3+{}^2\alpha_3+{}^3\alpha_3)\,{}^3\gamma\rightarrow{}^3p \end{array}$$

In the end, every party publics the value of its own p, then anyone can compute the summation.

$$p={}^1p+{}^2p+{}^3p$$

In fact this p is just what we want to compute.

4. Performance

4.1 Security

The security of SPoS is guaranteed by the following theorem. We omitted the proof of it.

Theorem 1 (*Security of SPoS*) *SPoS protocol is fully private ((m-1)-private), where m is the number of parties.*

Here $(m-1)$-private means that any party does not reveal its information, even though all other parties collude.

4.2 Efficiency

Our proposed protocol contains many operations performed by each two parties. And at any one time, each party can only perform such an operation with only one party. The following theorem implies that although the running cost of these operations is $O(m^2)$, the running time is just $O(m)$.

Theorem 2 *m parties collaborate to achieve a process in which each party achieves the same operation with any other party. Suppose that the running time of such an operation is T and any party cannot achieve that operation to more than one party simultaneously. Then the total cost is m(m-1)T/2, and the running time is (m-1)T when m is even, mT when m is odd.*

5. Related Applications

Each party has two inputs, ${}^ix$ and ${}^iy$. Consider the problem of secure computation of the functions on x and y, where

$$x={}^1x+{}^2x+{}^3x \quad \text{and} \quad y={}^1y+{}^2y+{}^3y$$

5.1 Secure Ratios of Summations Protocol

To compute the ratio of x and y, each party generates a secure number ${}^ic$, they use ${}^ix$ and ${}^ic$ to securely compute

$$xc=({}^1x+{}^2x+{}^3x)({}^1c+{}^2c+{}^3c),$$

and similarly,

$$yc=({}^1y+{}^2y+{}^3y)({}^1c+{}^2c+{}^3c).$$

The ratio of these two values is what we want to compute:

$$r=xc/yc=({}^1x+{}^2x+{}^3x)/({}^1y+{}^2y+{}^3y).$$

By this protocol, we can solve the following problems in privacy-preserving data mining more securely.

1) Probability Distribution Estimating from Distributed Data;
2) Naïve Bayes classify on Horizontally Partitioned Data ([3]);
3) K-Means on Horizontally Partitioned Data ([1]).

5.2 Secure Product of Summations Protocol

To compare the values of x and y, each party generates a secure number ${}^ic$, they use ${}^is={}^ix-{}^iy$ and ${}^ic$ to securely compute:

$$sc=({}^1s+{}^2s+{}^3s)({}^1c+{}^2c+{}^3c).$$

Comparison of these two values can be judged from this value.

By this protocol, we can solve the following problems in privacy-preserving data mining more securely.

1) K-Means over Vertically Partitioned Data([4]);
2) Sorting of Secure Summations.

6. Conclusion

We proposed a protocol to securely compute the value of the function on secure summations. It satisfies a high private level, full-privacy, hence it is more secure than other related methods. It is also an efficient one, for its running time is proportional to the number of parties. Since it is a general problem in privacy-preserving data mining, our proposed protocol can be applied to solve many problems in this field.

Reference

[1] S. Jha, L. Kruger, and P. McDamiel, *Privacy Preserving Clustering*, 10th European Symposium On Research In Computer Security, Milan Italy, 2005.

[2] Mert Ozarar and Attila Ozgit, *Secure Multiparty Overall Mean Computation via Oblivious Polynomial Evaluation*, International Conference on Security of Information and Networks, 2007.

[3]. Vaidya J., Murat Kantarcıoglu, Clifton C., *Privacy-preserving Naïve Bayes classification*, The VLDB Journal (2008), pp. 17:879-898.

[4] Vaidya J., Clifton C (2003) *Privacy-preserving k-means clustering over vertically partitioned data.* In: The 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, pp. 206–215 (2003).