SYMMETRIC-KEY BASED PRIVACY-PRESERVING SCHEME FOR MINING SUPPORT COUNTS

Yu Li[1] and Sheng Zhong

[1]Department of Computer Science and Engineering, University at Buffalo, NY, USA

yli32@buffalo.edu

ABSTRACT

In this paper we study the problem of mining support counts using symmetric-key crypto which is more efficient than previous work. Consider a scenario that each user has an option (like or unlike) of the specified product, and a third party wants to obtain the popularity of this product. We design a much more efficient privacy-preserving scheme for users to prevent the loss of the personal interests. Unlike most previous works, we do not use any exponential or modular algorithms, but we provide a symmetric-key based method which can also protect the information. Specifically, our protocol uses a third party that generates a number of matrixes as each user's key. Then user uses these key to encrypt their data which is more efficient to obtain the support counts of a given pattern.

KEYWORDS

Privacy-Preserving, Symmetric-key encryption, Support Counts

1. INTRODUCTION

With the rapid development of information technology, personal data should be protected in order to respect the people's private information and prevent the inappropriate usage. Most Internet users will be asked to provide an opinion that whether they support some pattern or not, then a data miner can count the number of this support. However, some websites collect, store and share these personal private data in order to benefit themselves despite users' desire. Privacy-preserving data mining has received a lot of attention in the past few years. This kind problem can be mainly divided into two categories: (a) the data is divided among more than one different party and the privacy preserving method needs to protect each party's private data; (b) the data is to be published for research, but due to its sensitivity the data should be modified [1].

A lot of typical privacy-preserving data mining problems include the scenario that mentioned above belongs to the first category. This kind of privacy-preserving data mining problem is called secure multiparty computation [1]. That means different party wants to jointly compute the correct result without revealing any private data. The most common way to provide privacy-preserving method is using cryptographic technology. Huge amounts of research have been investigated in this field after [2] provided an elegant privacy preserving data mining method using cryptographic tools.

DOI : 10.5121/acij.2013.4201

In this paper, we propose a symmetric-key crypto based privacy-preserving scheme for mining support counts using symmetric-key encryption. Our scheme is more efficient than previous work [3]. The rest of this paper is organized as follows. Section II describes the related work. We introduce in symmetric-key cryptosystem and block cipher in Section III. In Section IV, we describe our symmetric-key based privacy preserving support counts protocol and analyse our correctness of our protocol. Section V gives the discussion of work. Last, Section VI gives the conclusion of this paper.

2. RELATED WORK

Providing privacy-preserving for Internet data is a longstanding goal of the computer research community. It is has received considerable attention with the development of data mining and network technology. Cryptograph based privacy-preserving method can provide a better guarantee of the privacy when different institute want to cooperate in a common goal. In [4], Lea Kissner and Dawn Song propose efficient techniques for privacy-preserving operations on multisets using cryptosystem. In [5], the authors proposed a security protocol for the IVC applications based on group signature and ID-based signature schemes. In [3] the authors propose an identity-based protocol for support count that preserves privacy using public key cryptosystem. Different from the above reported schemes, we propose a symmetric-key based privacy-preserving protocol, which cannot only guarantee the requirements of security and privacy but can also provide a very good efficiency. Furthermore, the protocol and infrastructure are easy to be implemented. Thus, the proposed protocol can practically be launched for enabling the application of Internet website.

3. TECHNICAL PRELIMINARIES

In this section, we give a brief review of symmetric-key cryptosystem and block cipher that used in our privacy-preserving scheme for mining support counts.

3.1. Symmetric-key Cryptosystem

A cryptosystem [6] can be defined as a tuple (M, C, K, G, E, D, G). The denotations are shown in table 1. The cryptosystem must be efficient and must be sufficiently random so that it is hard to guess. The encryption and decryption process $(E(k_e, m) and\ (D(k_e, c))$ must be efficient. We must have $D(k_d, E(k_e, m)) = m$. We have a symmetric-key cryptosystem when encryption key is same with decryption key which means $k_e = k_d$.Without the decryption key, there is no efficient algorithm that can decrypt ciphertexts. There are two requirements for a symmetric cryptosystem. The first one is that it is impractical to get a plaintext only known the ciphertext and encryption scheme. Another one is that all users must obtain the key in a safe way and keep their keys secure.

Table 1. Denotations of Cryptosystem

Summary of Denotations
M: cleartext space
C: ciphertext space
K: key space
G: key generating algorithm
E: encryption algorithm
D: decryption algorithm
G: generate a pair of encryption/decryption keys for a given length

3.2. Blockcipher

A block cipher [7] is a function $E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$. There are two inputs in this algorithm. The first input is the key whose length is k . The second input is the plaintext whose length is n, and the output is the ciphertext whose length is same as the length of plaintext. The length of the key and the length of the block are parameters associated to the blockcipher.
When using block cipher, a random key K is assigned and kept secret between users. The function E_k is then used by a user to process data before she sends it to other user. Typically, we will assume the adversary will be able to obtain some input-output examples for E_k which are pairs of $\{M, C\}$ where $C = E_k(M)$. But the adversary will not be shown the key K because the security relies on the secrecy of the key. So the goal of the adversary is recovering the key by given some input-output examples of E_k. However, a good blockcipher scheme should be designed to make this task computationally difficult.

4. SYMMETRIC-KEY CRYPTO BASED SUPPORT COUNT PROTOCOL

In this section, we go into the detail of our symmetric-key crypto based support counts mining protocol and prove its correctness.

4.1. Protocol

In our symmetric-key crypto based support count protocol, we use a third party key-generator. First, let the generator generate two matrixes. The dimensions are based on the number of users. Assume there are n users in total, the dimension of the two matrixes should be $n \times n$. Then the generator fills the first matrix with random numbers. After that, generator fills the second matrix with the negative values of first one. Note that the filling order of the second matrix should be permuted. Then we define the symmetric-key for each user U_i as the i_{th} row of M_1 and i_{th} column of M_2.

$$k_{i1} = \sum_{j=1}^{n} M_1[i,j], \quad k_{i2} = \sum_{j=1}^{n} M_2[j,i]$$

Then each user performs our protocol shown in algorithm 1.

Algorithm 1 Symmetric-key based privacy preserving scheme

Initialize

Step 1

User 1 will be asked whether he support one given pattern or not. Define support as 1 and not support as 0. Then he uses the two keys distributed to him to encrypt his answer. He will get the value of $S_1 + k_{11} + k_{12}$ and sends the value to User 2.

Step 2

For each user i, $2 \leq i \leq n$, User i receives $\sum_{i=1}^{i-1} S_i + \sum_{i=1}^{i-1} k_{i1} + \sum_{i=1}^{i-1} k_{i2}$ User i computes $\sum_{i=1}^{i-1} S_i + \sum_{i=1}^{i-1} k_{i1} + \sum_{i=1}^{i-1} k_{i2} + S_i + k_{i1} + k_{i2}$ and sends it to user $i+1$

Step 3

User n receives $\sum_{i=1}^{n-1} S_i + \sum_{i=1}^{n-1} k_{i1} + \sum_{i=1}^{n-1} k_{i2}$. User n adds $S_n + k_{n1} + k_{n2}$ to the received number, and sends the results to all other parties.

Step 4

Each user i uses the block cipher to update their keys.

Here, we specify that how to update user U_i's keys using blockcipher in step 4. Since k_{i1} is the sum of elements in i_{th} row in matrix M_1 which is $e^1_{icol_1} + e^1_{icol_2} + \cdots + e^1_{icol_n}$. Therefore we use algorithm 2 to update the $\{e^1_{icol_1} + e^1_{icol_2} + \cdots + e^1_{icol_n}\}$.

Algorithm 2 Keys updating algorithm

Initialize

Step 1

Each user i will be assigned a blockcipher "E" as well as a key "x" during the initialization stage.

Step 2

Using blockcipher method, we define a function F which makes

$E_x\{e^1_{icol_1} + e^1_{icol_2} + \cdots + e^1_{icol_n}\} \rightarrow \{e^{1'}_{icol_1} + e^{1'}_{icol_2} + \cdots + e^{1'}_{icol_n}\}$

Step 3

User i uses the same method to update the second key k_{i2}. First the user converts the value of e to absolute value. Then make them as negative after update the value use step 2.

Step 4

After $n/10$ mining rounds, the server reassigns another "E" and "x" to each user.

In keys updating process, blockcipher "E" and key "x" are chosen randomly by third party key-generator and make known to all users. In order to enhance the security level of our protocol, we let the generator create a new blockcipher "E" and key "x" and send them to all users after $n/10$ rounds of support counts mining.

Unlike most previous works, we do not use any exponential or modular algorithms, we provide a symmetric-key based method which can also protect the information and the execution time is faster than public-key based algorithm. Since in modern network service, the number of user can be as large as millions. Therefore the efficiency is most essential when designing of the privacy-preserving algorithm.

4.2. Correctness

In this section, we prove the correctness of our symmetric-key crypto based support counts mining protocol.

Theorem 1. Our symmetric-key crypto based support counts mining protocol correctly computes the sum of all users' outputs.

Proof. Now, we prove that the user can obtain the final result by following our protocol.

$$s_{total} = \sum_{i=1}^{n} s_i + \sum_{i=1}^{n} k_{i1} + \sum_{i=1}^{n} k_{i2}$$

Since we have

$$k_{i1} = e^1_{icol_1} + e^1_{icol_2} + \cdots + e^1_{icol_n}$$
$$k_{i2} = e^2_{irow_1} + e^2_{irow_2} + \cdots + e^2_{irow_n}$$

Therefor we can get

$$\sum_{i=1}^{n} k_{i1} + \sum_{i=1}^{n} k_{i2} = 0$$

It is proved that $s_{total} = \sum_{i=1}^{n} s_i$.

5. DISCUSSION

Our proposed symmetric-key crypto based protocol for mining support counts has the following characteristics.

It preserves the data privacy for each user without revealing it to others in the process of mining support counts. Our privacy-preserving protocol is accurate which means the result obtained by our protocol is the same as those obtained by having all data collected without privacy-preserving.

We are utilizing a symmetric-key based method in our models. The protocol is much more efficient compared with public-key based approached. An existing work [3] in mining support counts proposed the privacy-preserving was based on public-key based approach. Different from their protocol, we focus on another cryptographic method---symmetric-key cryptosystem. Furthermore, our protocol is more efficient than previous work.

6. CONCLUSION

In this paper, we proposed a symmetric-key crypto based privacy preserving protocol for mining support counts, in order to obtain the result without revealing users' personal private information. We provided a very efficient method based on symmetric-key cryptosystem and cipher block to encrypt each user's data and proved the correctness of our protocol.

REFERENCES

[1] Y. Lindell & B. Pinkas, (2009) Secure multiparty computation for privacy-preserving data mining. Journal of Privacy and Confidentiality, Vol. 1, No.1, pp5.

[2] Y. Lindell & B. Pinkas , (2000) Privacy preserving data mining. In Advances in Cryptology CRYPTO, Springer, pp36-54.

[3] F.Wu, J. Liu & S. Zhong. (2009) An efficient protocol for private and accurate mining of support counts. Pattern Recognition Letters, Vol. 30, No.1, pp80-86.

[4] L.Kissner & D.Song. (2005) Privacy-preserving set operations. In Advances in Cryptology CRYPTO, Springer, pp241-257.

[5] X.Lin, X.Sun, P.H. Ho & X.Shen. (2007) Gsis: a secure and privacy-preserving protocol for vehicular communications. Transactions on Vehicular Technology, IEEE, Vol. 56, No.6, pp3442-3456.

[6] H.Delfs & H.Knebl. (2007) Symmetric-key encryption. Introduction to Cryptography, Springer , pp11-31.

[7] T.W. Cusick & P.Stanica. (2009) Cryptographic Boolean functions and applications. Academic Press.