Privacy Preserving Aggregate Query of OLAP for Accurate Answers

Youwen Zhu[1,2], Liusheng Huang[1,2], Wei Yang[1,2] and Fan Dong[1,2]
(1. National High Performance Computing Center at Hefei, Department of Computer Science and Technology,
University of Science and Technology of China, Hefei, 230027, P. R. China
2. Suzhou Institute for Advanced Study, University of Science and Technology of China,
Suzhou 215123, China)
(E-mail: zhuyw@mail.ustc.edu.cn, {lshuang, qubit}@ustc.edu.cn,
dongfan@mail.ustc.edu.cn)

Abstract—**In recent years, privacy protection has become an important topic when cooperative computation is performed in distributed environments. This paper puts forward efficient protocols for computing the multi-dimensional aggregates in distributed environments while keeping privacy preserving. We propose a novel model, which contains two crucial stages: local computation and cooperative computation based on secure multiparty computation protocols for privacy-preserving on-line analytical processing. According to the new model, we develop approaches to privacy-preserving *count* aggregate query over both horizontally partitioned data and vertically partitioned data. We, meanwhile, propose an efficient sub-protocol Two-Round Secure Sum Protocol. Theoretical analysis indicates that our solutions are secure and the answers are exactly accurate, that is, they can securely obtain the exact answer to aggregate query without revealing anything about their confidential data to each other. We also analyze detailedly the communication cost and computation complexity of our schemes in the paper and it shows that the new solutions have good linear complexity. No privacy loss and exact accuracy are two main significant advantages of our new schemes.**

Index Terms—**Privacy, OLAP, Homomorphic Encryption, Secure Multiparty Computation, Scalar Product Protocol**

I. Introduction

On-line analytical processing (denoted as OLAP) is a significant data analysis technology. In recent years, OLAP encounters a new problem that it is now required to developed methods to protect the confidential information of individuals when the computation of multi-dimensional aggregates is performed in distributed environments. Privacy preserving OLAP (denoted as PPOLAP) devotes to designing secure distributed OLAP model and developing privacy protection approaches to concrete aggregate query. Agrawal *et al.* [1] first propose the problem of PPOLAP in a distributed scene and define a model for privacy-preserving computation of multidimensional *count* aggregates over data partitioned across multiple clients using the randomization approach. However, the model is not practical, because that its process was extremely complex and the scheme [1] is only able to return the inaccurate answers. Thus, more efficient methods should be developed for the special problems of PPOLAP.

The purpose of Secure Multi-party Computation (denoted as SMC) [2-3] is to allow a group of participants to carry out cooperative computations over their private inputs in a special way that each participant knows the multi-party cooperative computations' result, but nobody learns more other than what could be derived from his outputs of the cooperative computations. Yao [2] introduces the notion of SMC in 1982. Since then, it has been a hot research topic and attracts numerous researchers [2-13]. Basic SMC protocols can be used to construct far more complicated privacy preserving protocols to solve special practical questions, including privacy preserving data mining [14-15], privacy preserving social networks [16], privacy-preserving computation geometry [7-8], etc.

In this paper, we investigate how to achieve PPOLAP using the methods of SMC and propose a novel model for aggregate queries over distributed data in privacy-preserving manner. Based on homomorphous encryption system [17] and scalar product protocol [4-6], we propose two efficient schemes for PPOLAP following the new model, which are respectively application in two kinds of different distributed data structure, horizontally partitioned data and vertically partitioned data. Furthermore, we derive theoretical formulas to analyze the security and performance of the two schemes. It shows that our privacy preserving solutions are secure and the communication overheads and computation cost are reasonable. Another significant advantage of the new schemes is that they will return the accurate result to aggregate query instead of its maximum likelihood estimator which is the answer in Agrawal's model [1].

Our main contributions in this paper are:

(1) We define a model for privacy-preserving computation of multidimensional *count* aggregates based on SMC;

(2) We propose a novel secure sum protocol: Two-Round Secure Sum Protocol (denoted as TSSP), which is quite secure to against underlying collusion;

(3) We present two efficient schemes for PPOLAP following the new model, which are respectively application in two kinds of different distributed data structure, horizontally partitioned data and vertically partitioned data, and we detailedly analyze the security and performance of the two new schemes.

doi:10.4304/jcp.5.11.1678-1685

The rest of this paper is organized as follows. In Section 2, we discuss some related work and give necessary preliminaries. The privacy preserving *count* aggregate query scheme on horizontally partitioned data and Two-Round Secure Sum Protocol are presented in section 3. Section 4 proposes a privacy preserving solution to *count* aggregate query on vertically partitioned data. We conclude our work and present directions for future research in section 5.

II. Related Work And Preliminaries

The related work and some essential preliminaries are introduced in the following content.

A. *Privacy preserving OLAP*

Agrawal *et al.* [1] first propose the problem of PPOLAP in a distributed scene and define a model for privacy preserving computation of multidimensional *count* aggregate query on data partitioned across multiple clients using the randomization approach, where each client contributes perturbed private data to a central server, and then the server evaluates the answers to initial aggregate queries on global tables against perturbed tables by reconstructing original distributions. However, the model in [1] using randomization approach has some drawbacks. The following are several of them:

(1) It can't return the accurate result to aggregate query but just obtains a maximum likelihood estimator of the exact value.

(2) The reconstruction is extremely complex and executes many matrix manipulations which are expensive in computation overhead.

(3) The randomization approach only ensures low-lever privacy [1].

(4) Some private data of the clients will be revealed even though other party doesn't know which parts are genuine and which are fictitious.

(5) There is a central server in the model, and all clients send perturbed data to the server; as a result, the server may become a bottle-neck and confidential information of clients will suffer huge leakage when a malicious attacker gains access to the server.

(6) Since each client transports their perturbed data to the central server, the amount of communication between server and clients is great large.

For the reasons above, Agrawal's model [1] is not suitable for practical applications. More efficient methods should be developed for the problems of PPOLAP.

We investigate PPOLAP and propose a new-type model (displayed in Figure 1) using SMC protocols to facilitate it. In our new model, each aggregate query on global data is broken down into some secondary queries for every client; a client performs local computation on his private sub-database. The results of local computation will act as the private inputs of SMC protocols to compute accurate answer to original global aggregate query in privacy-preserving and cooperative way. At last, SMC protocols securely output the exact answer to primordial aggregate query on global data.

According to the model in Figure 1, we propose two approaches to response the privacy-preserving problems in OLAP, which respectively have application in different distributed data structures. We put forward privacy preserving solution to *count* aggregate query on horizontally partitioned data in section 3; communication overheads and computation complexity of each client in our schemes is $O(1)$, therefore, the total communication overheads and computation complexity is $O(h)$ where h denotes the number of participant clients. Then section 4 bends to privacy-preserving *count* aggregate query on vertically partitioned data; cost of each client is $O(n)$ where n denotes the size of global database records. Theoretical analysis gives sufficient evidence of security and correctness of our schemes.

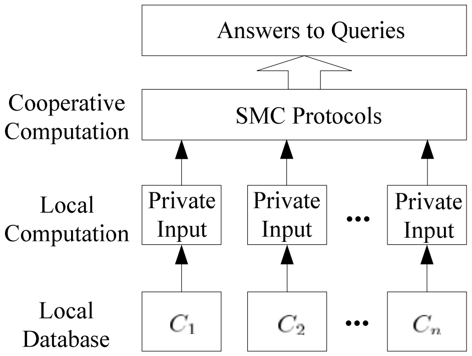


Figure 1. The PPOLAP model based on SMC protocols

B. *Homomorphic Encryption*

A public key encryption scheme (E, D), where E and D are polynomial-time algorithms for encryption and decryption respectively, is homomorphic when it meets the following condition

$$D(E(m_1) \times E(m_2)) = m_1 + m_2.$$

That is $E(m_1) \times E(m_2) \triangleq E(m_1 + m_2)$, where $\triangleq$ denotes that they hide the same plaintext item. As a result, we can employ homomorphic cryptosystem to compute $E(x+y)$ from $E(x)$ and $E(y)$ such that the secret numbers x and y aren't disclosed. Pallier [17] proposed a semantically secure (IND-CPA secure) homomorphic encryption system. Another significant feature of Pallier's homomorphic encryption system is that it doesn't encrypt a plaintext unit into a same ciphertext item everytime. If E is the encryption function which has two inputs a secret message m and a random parameter r, for an arbitrary secret message m, we have $E(m, r_1) \neq E(m, r_2)$ $(r_1 \neq r_2)$. However, the decryption doesn't depend on the random parameter at all and $D(E(m, r_1)) \equiv D(E(m, r_2)) \equiv m$ where D is the corresponding decrypt function. The encryption scheme that satisfies the above property is called a probabilistic encryption scheme which was first introduced in [18]. We briefly describe Pallier's cryptosystem as follows. See [17] for more details.

Key generation: Select two large enough primes p and q. Then the secret private key sk is

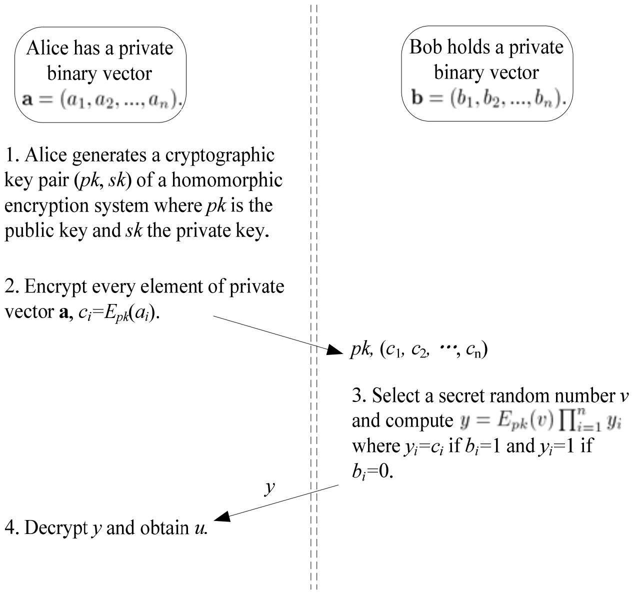
$\lambda = lcm(p-1, q-1)$ which is the least common multiple of $p-1$ and $q-1$. The public key pk is (n, g) , where $n = pq$ and $g \in \mathbb{Z}_{n^2}^*$ such that $gcd(L(g^\lambda \ mod \ n^2), n) = 1$, that is, the maximal common divisor of $L(g^\lambda \ mod \ n^2)$ and n is equivalent to 1 where $L(x) = (x-1)/n$ and the same below.

Encryption: Let $m \in \mathbb{Z}_n$ be the plaintext. Select a random number $r \in \mathbb{Z}_n$ as the secret parameter for probabilistic encryption. Then the cryptograph c of m is $c = g^m r^n \ mod \ n^2$.

Decryption: Let $c \in \mathbb{Z}_{n^2}$ be a ciphertext. Then the plaintext m hidden in c is

$$m = \frac{L(c^\lambda \ mod \ n^2)}{L(g^\lambda \ mod \ n^2)} \ mod \ n.$$

The homomorphic encryption system tersely described above is an important tool to be employed in our latter work.



Alice has a private binary vector $\mathbf{a} = (a_1, a_2, ..., a_n)$.
Bob holds a private binary vector $\mathbf{b} = (b_1, b_2, ..., b_n)$.
1. Alice generates a cryptographic key pair (*pk*, *sk*) of a homomorphic encryption system where *pk* is the public key and *sk* the private key.
2. Encrypt every element of private vector **a**, $c_i = E_{pk}(a_i)$.
pk, $(c_1, c_2, \cdots, c_n)$
3. Select a secret random number *v* and compute $y = E_{pk}(v) \prod_{i=1}^n y_i$ where $y_i = c_i$ if $b_i = 1$ and $y_i = 1$ if $b_i = 0$.
y
4. Decrypt *y* and obtain *u*.

Figure 2. Scalar Product Protocol [4]

C. Scalar Product Protocol

Given two vectors $\mathbf{a} = (a_1, a_2, ..., a_n)$ and $\mathbf{b} = (b_1, b_2, ..., b_n)$, their scalar product, which is also called dot product or inner product, is $\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i$. Scalar product protocol has been proposed in [4-6]. When Alice has the vector $\mathbf{a}$ in private and Bob owns the other secret vector $\mathbf{b}$, the objective of scalar product protocol is to securely compute the inner product of two private vectors held by them. That is to say, scalar product protocol permit them to obtain the dot product of their private vectors such that neither party learns anything about partner's secret data apart from what is inferred from the scalar product and his own vector. At the end of scalar product protocol, Alice gets her secret output u and Bob obtains the private number v, which meet the equation $u = \mathbf{a} \cdot \mathbf{b} + v$.

Scalar product protocol is one of significant members in SMC toolkit, and lots of problems can essentially be reduced to computing scalar product [4-6, 19]. While $\mathbf{a}$ and $\mathbf{b}$ are binary vectors (namely $a_i, b_i \in \{0, 1\}$, $i = 1, 2, ..., n$), we call scalar product protocol that privately computes $\mathbf{a} \cdot \mathbf{b}$ as binary scalar product protocol. R. N. Wright and Z. Yang [5] has presented an efficient secure binary scalar product protocol based on homomorphic encryption. In section 4, we will make use of binary scalar product protocol [5], which is illustrated in Figure 2, to privately evaluate *count* aggregate query on heterogeneous data.

D. Security Definition

Many SMC protocols [2-14] are under the semi-honest model. In the paper, we assume that all participants are semi-honest [3], which is also called honest-but-curious, who exactly follow the steps of the protocols and keep a record of all the intermediate results he receives to infer some potentially confidential information contained in them. The formal security definition in the semi-honest model has been presented in [3]. Generally speaking, a SMC protocol is secure if and only if all the data that each participant has or receives during an execution could be deduced from his private input and confidential output. In fact, our approach over horizontally distributed data is stronger than the semi-honest model in some sense (see section 3).

	A_1	A_2	...	A_m
1	a_{11}	a_{12}	...	a_{1m}
2	a_{21}	a_{22}	...	a_{2m}
...	...	...	...	...
n	a_{n1}	a_{n2}	...	a_{nm}

Figure 3. The global table S

E. Problem Definition

There is a relation table $S(A_1, A_2, ..., A_m)$ shown in Figure 3. The table S has m attributes $A_1, A_2, ..., A_m$ and n rows. In our setting, each client only privately owns some rows or a few columns of the global table S. They have an intention of carrying out aggregate query on the table S by means of cooperation and none of private data is leaked simultaneously, i.e., a client gains nothing but the result of aggregate query and what could be deduced from the answer to query. In this paper, we develop approaches to privately perform *count* aggregate query like as the following form on both homogeneous data and heterogeneous data according to the PPOLAP model based on SMC protocols in a decentralized, distributed environments.

$$\begin{aligned} &\textbf{select count}(*) \\ &\textbf{from } S \\ &\textbf{where } P_1 \textbf{ and } P_2 \textbf{ and ... and } P_c \end{aligned}$$

Here P_i $(i = 1, 2, ..., c)$ is the condition for selecting. We propose privacy preserving solution to *count* aggregate query as the form above on horizontally partitioned data in section 3, then section 4 bends to privacy-preserving *count* aggregate query on vertically partitioned data.

III. PRIVACY PRESERVING AGGREGATE QUERY OF OLAP ON HORIZONTALLY PARTITIONED DATA

Horizontally partitioned data is also referred to homogeneous data where each client holds a subset of the rows in the table S including m attribute values. Figure 4 illustrates a horizontally distributed data partitioning over two clients, where the table S is divided into double data blocks and each data block has all the m attribute values.

	A_1	A_2	...	A_m
1	a_{11}	a_{12}	...	a_{1m}
2	a_{21}	a_{22}	...	a_{2m}
...	...	...	...	...
k	a_{k1}	a_{k2}	...	a_{km}

(1) $Client\ 1$

	A_1	A_2	...	A_m
$k+1$	$a_{k+1,1}$	$a_{k+1,2}$	...	$a_{k+1,m}$
$k+2$	$a_{k+2,1}$	$a_{k+2,2}$	...	$a_{k+2,m}$
...	...	...	...	...
n	a_{n1}	a_{n2}	...	a_{nm}

(2) $Client\ 2$

Figure 4. Horizontally Partitioned Data $(1 \leqslant k < n)$

In this section, we describe how to securely evaluate *count* aggregate query on horizontally partitioned data. It's set that the table S is horizontally distributed between $h \quad (h \geqslant 3)$ individual clients $C_1, C_2, ..., C_h$ where $C_i \ (1 \leqslant i \leqslant h)$ possesses a private sub-table S_i including m attribute values and $S = \bigcup_{i=1}^{h} S_i$, $S_i \cap S_j = \emptyset (i \neq j, 1 \leqslant i, j \leqslant h)$. They want to jointly conduct *count* aggregate query on the union of all their private data sets without disclosing any private data to anybody else including other participators.

To find out the result of *count* aggregate query on the global table S, $C_1, C_2, ..., C_h$ together implement the following two stages.

1. Local Computation: $C_i \ (i = 1, 2, ..., h)$ carries out the sub-query on its local private sub-table S_i as follows.

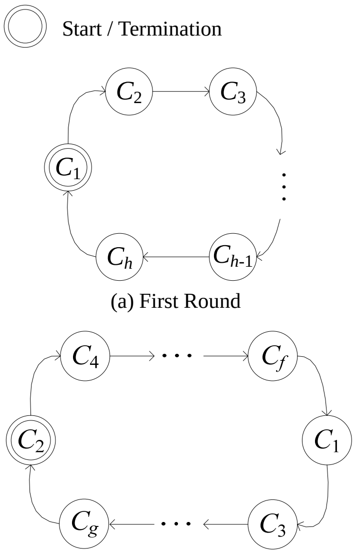
   select count$(*)$
   from S_i
   where P_1 **and** P_2 **and** ... **and** P_c

   We denote the return value of the above SQL statement as n_i.

2. Cooperative Computation: $C_1, C_2, ..., C_h$ cooperate to privately compute the sum $\sum_{i=1}^{h} n_i$ by performing Two-Round Secure Sum Protocol.

Two-Round Secure Sum Protocol

The goal of Two-Round Secure Sum Protocol (denoted as TSSP) is to deal with the following problem that $h \ (h \geqslant 3)$ individual clients $C_1, C_2, ..., C_h$ hold secret private numbers $n_1, n_2, ..., n_h$ separately and they intend to find out the sum $\sum_{i=1}^{h} n_i$ while each party obtains nothing about other client's private information.



(a) First Round

(b) Second Round

Figure 5. Two Different Ring Topologies, g and f are respectively the largest odd number and even integer in the closed interval $[1, h]$.

Highlight of TSSP: In the protocol, there are two rounds to securely compute the sum. In the first round, each client obtains a secret random digit to obscure his private number. They alter the order of transmitting messages in the second round to reduce the information loss caused by underlying collusion and the sum will safely be found out after the second round. In the first round, C_1 randomly generates a key pair (pk_1, sk_1) of Pallier's homomorphic cryptosystem (see section 2.2) and a large integer m_num which is much greater than the total number of records in the table S. Then C_1 sends C_2 the encryption of a random natural number r; C_2 randomly selects a private numeric parameter r_2 which will be used to mask its confidential number n_2 in the second round and computes $E_{pk_1}(r + r_2) = E_{pk_1}(r) \times E_{pk_1}(r_2)$ which is sent to C_3. C_h recursively obtains $E_{pk_1}(r + \sum_{k=2}^{h} r_i)$ along the ring illustrated in Figure 5(a) and sends it to C_1. At last, C_1 decrypts $E_{pk_1}(r + \sum_{k=2}^{h} r_i)$ and sets

$$r_1 = m\_num - (\sum_{k=2}^{h} r_i) \ (mod\ m\_num),$$

such that $0 \equiv (\sum_{k=1}^{h} r_i)(mod\ m\_num)$ holds.

Therefore, each client retains a random integer which is called digital envelope in [14] to hide its private number, the return value of local sub-query, in the next round. Assume g and f are respectively the maximal odd integer and the largest even number in the closed interval $[1, h]$. It is quite clear that

$$g = h, \ f = h - 1 \quad \text{if } h \text{ is odd integer;}$$
$$g = h - 1, \ f = h \quad \text{if } h \text{ is even integer.}$$

The client C_2 generates another key pair (pk_2, sk_2) of homomorphic encryption system in the second round. Then, they implement the following circle where the order of transmitting messages is demonstrated in Figure 5(b). C_2 encrypts $s+r_2+n_2$ where s is a random integral value selected and held by C_2 and transmits $E_{pk_2}(s+r_2+n_2)$ to its immediate successor C_4; then C_4 computes $E_{pk_2}(s+r_2+r_4+n_2+n_4)$ $=E_{pk_2}(s+r_2+n_2)\times E_{pk_2}(r_4+n_4)$ and sends it to C_6. It is repeated on the ring shown in Figure 5(b) until C_2 receives $E_{pk_2}(s+\sum_{k=1}^{h} r_i+\sum_{k=1}^{h} n_i)$ from C_g.

Finally, C_2 decrypts it, then obtains $\sum_{i=1}^{h} n_i$ by subtracting the extra value s and modulo m_num and broadcasts the exact sum.

The Correctness Analysis of TSSP: To explain the equation $sum=\sum_{t=1}^{h} n_t$ is correct, the proof is displayed as below.

In the step 5 of TSSP, it is quite clear that

$y_i = E_{pk_2}(s+n_2+n_4+...+n_i+r_2+r_4+...+r_i)$.

Therefore, $y_g = E_{pk_2}(s+\sum_{t=1}^{h} n_t+\sum_{t=1}^{h} r_t)$ holds in the step 8 of TSSP.

Then,

$$\begin{aligned} sum &= (D_{sk_2}(y_g)-s)\ (mod\ m\_num) \\ &= (\sum_{t=1}^{h} n_t+\sum_{t=1}^{h} r_t)\ (mod\ m\_num). \end{aligned}$$

In the step 4 of TSSP, C_1 sets

$$r_1 = max\_num - D_{sk_1}(z_h)+r\ (mod\ m\_num).$$

Thereby, $r_1+D_{sk_1}(z_h)\equiv r\ (mod\ m\_num)$.

Since $z_h=E_{pk_1}(r+\sum_{t=2}^{h} r_t)$, thus,

$$\sum_{t=1}^{h} r_t \equiv 0\ (mod\ m\_num).$$

Therefore, $sum=\sum_{t=1}^{h} n_t$ holds while every client correctly obeys the protocol.

The Security Analysis of TSSP: When each client is semi-honest, that is, they strictly follow the protocol and no clients collude, a single client can't find out any other client's private data since each number is masked and messages in the first round and second round are encrypted by different public key. We present the security analysis of TSSP while potential collusion occurs in the sub-section below. In the process of TSSP, C_1 and C_2 hold larger amount of information than other clients. Specifically speaking, C_1 has sk_1 and m_num; C_2 holds sk_2, m_num and s.

Given that C_j is corrupted, to figure out private data n_i of client C_i, the malignant client C_j has to obtain sk_1, sk_2 and collude with the immediate predecessor and immediate successor of client C_i in first round and second round. There are two cases as below.

1) C_i is one of C_1 and C_2. It's impossible for C_j to determine n_i because C_i privately holds sk_1 or sk_2;

2) C_i is one of $C_3, C_4, ..., C_h$. C_j could collude with C_1 (for sk_1), C_2 (for sk_2), the immediate predecessor and immediate successor of client C_i in first round (for r_i) and second round (for r_i+n_i) to work out n_i. Then C_j has to collude with other 2 (at least) to 4 (at most) clients except C_1 and C_2 to derive the private number n_i of C_i. Therefore, our scheme has

Two-Round Secure Sum Protocol (denoted as TSSP)

Require: There are $h(h \geqslant 3)$ clients $C_1, C_2, ..., C_h$, and $C_i(i=1,2,...,h)$ has a private integer n_i. They intend to securely calculate the sum $\sum_{i=1}^{h} n_i$ and no confidential number is revealed at the same time.
Given m_num, privately owned by C_1 and C_2, is a big enough integer which is much larger than $\sum_{i=1}^{h} n_i$. We assume that g denotes the largest odd number in the closed interval $[1,h]$ and f denotes the maximal even integer in the closed interval $[1,h]$.

Process:

```
// First Round
Step1: $C_1$ generates a cryptographic key pair $(pk_1, sk_1)$
       of a semantically secure homomorphic encryption
       system (see section 2.2) and creates a random
       number $r$. He sends public key $pk_1$ to all the
       participants. Then, he computes $z_1 = E_{pk_1}(r)$
       and sends $z_1$ to $C_2$.
Step2: for $i = 2, 3, ..., h-1$
           $C_i$ selects a private random number $r_i$ and
           computes $z_i = z_{i-1} \times E_{pk_1}(r_i)$, namely,
               $z_i = E_{pk_1}(r + \sum_{j=2}^{i} r_j)$.
           Send $z_i$ to $C_{i+1}$.
       endfor
Step3: $C_h$ generates a private stochastic natural number
       $r_h$ and computes $z_h = z_{h-1} \times E_{pk_1}(r_h)$. Send
       $z_h$ to $C_1$.
Step4: $C_1$ decrypts $z_h$. Then he sets
       $r_1 = max\_num - D_{sk_1}(z_h) + r\ (mod\ m\_num)$
       and keeps $r_1$ secret.

//Second Round
Step5: $C_2$ generates another public and private key pair
       $(pk_2, sk_2)$ of Pallier's homomorphic encryption
       scheme and sends public key $pk_2$ to other
       parties. Then he selects a secret random
       parameter $s$ and sets $y_0 = E_{pk_2}(s)$. //initialize $y_0$
       for $i = 2, 4, ..., f-2$
           $C_i$ calculates $y_i = y_{i-2} \times E_{pk_2}(n_i + r_i)$.
           Send $y_i$ to $C_{i+2}$.
       endfor
Step6: $C_f$ evaluates $y_f = y_{f-2} \times E_{pk_2}(n_f + r_f)$ and
       sends $y_{-1} = y_f$ to $C_1$.
Step7: for $i = 1, 3, 5, ..., g-2$
           $C_i$ computes $y_i = y_{i-2} \times E_{pk_2}(n_i + r_i)$
           and sends $y_i$ to $C_{i+2}$.
       endfor
Step8: $C_g$ evaluates $y_g = y_{g-2} \times E_{pk_2}(n_g + r_g)$ and
           sends $y_g$ to $C_2$.
Step9: $C_2$ computes $sum = (D_{sk_2}(y_g) - s)$
       $= \sum_{t=1}^{h} n_t\ (mod\ m\_num)$ and broadcasts
       $sum$ to $C_1, C_3, C_4, ..., C_h$.
```

strong capability of prevent the disclosure of privacy. For one thing, it's hard to persuade 4 to 6 coconspirators because of high cost and high risk; for another, a malicious client can't learn other clients' private data while C_1 or C_2 does not collude with C_j. Besides, we can adopt measures below in the real world to implicitly preclude collusion.

(A) Set two adversaries as C_1 and C_2 respectively, such that they will not be corrupted simultaneously.

(B) Let a client with high credibility be C_1 or C_2, then he holds high loyalty to refuse collusion with any corrupted party.

(C) Breaking rules and losing reputation will be severely punished, which will hold back potential collusion to a great extent.

Figure 6 indicates that the average number of clients needed to collude with in order to slinkingly infer the private data of C_i increases with the total number of clients and nearly runs up to 8 when the total number of clients is larger than 20; the lower bound of number of clients needed to collude with is 4 (while the total clients are more than 5 clients) or $h-1$ (when there are h $(h=3,4)$ clients in total). Therefore, TSSP, which increases level of difficulty to collude, can effectively reduce the risk from potential collusion.

According to the above fact, our protocol TSSP is quite secure to against underlying collusion. Thus, it has stronger security than the semi-honest assumption [3].

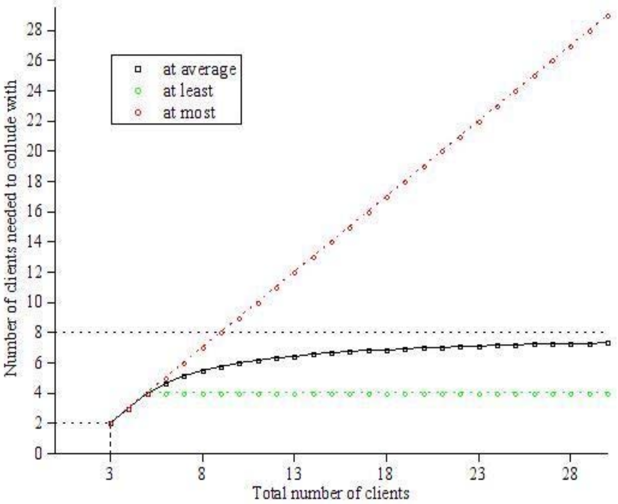


Figure 6. Number of clients needed to collude with in order to slinkingly infer a private data

The Complexity Analysis of TSSP: In the first round, there is the generation of one asymmetric key pair, h encryptions, $h-1$ multiplications, one decryption, two additions (including subtractions) and one modular arithmetic. In the second round, there is the generation of one asymmetric key pair, $h+1$ encryptions, h multiplications, one decryption, one subtraction and one modular arithmetic. Hence, the computation complexity of TSSP is $O(h)$ and the overall computation cost is about $14h$. While a cipher-text is b_0 bits long, the total bit-wise communication overheads are $2b_0h$ (b_0h in the first round and b_0h in the second round).

Based on the model in Figure 1, the novel scheme for privacy-preserving OLAP over horizontally partitioned data contains two stages: Local Computation and Cooperative Computation. In the Local Computation stage, each client separately performs local computation on his private sub-database. Then, TSSP is invoked in the Cooperative Computation stage. Obviously, it will return accurate result *count* aggregate query on the global table S based on the sub-protocol TSSP, instead of the exact value's maximum likelihood estimator in [1]. Besides, in the new scheme, the additional computation cost and communication overheads of cooperatively computing in a privacy-preserving way, which fully come from the Cooperative Computation stage: invoking TSSP, both are $O(h)$ (h is the number of participant clients).

IV. Privacy Preserving Aggregate Query Of OLAP On Vertically Partitioned Data

In this section, we study the *count* aggregate query of OLAP over heterogeneous data in an efficient and privacy preserving way. We consider the setting that the data is vertically distributed between two clients C_1 and C_2, in which C_1 and C_2 respectively hold a private sub-table S_1 and S_2 (displayed in Figure 7) with different attributes of all the records in the global table S. An attribute is held by a single client; we assume that C_1 privately owns the values of former k attributes $A_1, A_2, ..., A_k$ and C_2 has the values of other $m-k$ attributes $A_{k+1}, A_{k+2}, ..., A_m$. The two clients want to collaborate to accurately answer *count* aggregate query without disclosing any individual data in their private sub-table to each other.

Based on the efficient scalar product protocol [5], we present a solution to privacy-preserving *count* aggregate query on vertically partitioned data. Given that the conditions $P_1, P_2, ..., P_{c_a}$ are over the attributes of client C_1 and $P_{c_a+1}, P_{c_a+2}, ..., P_c$ over the attributes of client C_2, they conduct sub-query from respective private sub-table and then work out the answer to the primordial aggregate query on global table S by some secure multiparty computation protocol. First, we definite two binary vectors $\mathbf{a}=(a_1, a_2, ..., a_n)$ (privately owned by C_1) and $\mathbf{b}=(b_1, b_2, ..., b_n)$ (privately held by C_2) where the value of a_i (resp. b_i, $i=1,2,...,n$) is 0 or 1. The value of a_i (resp. b_i, $i=1,2,...,n$) is set to 1 if the *i-th* record in sub-table S_1 (resp. S_2), " $a_{i1}, a_{i2}, ...a_{ik}$ " (resp. " $a_{i,k+1}, a_{i,k+2}, ...a_{i,m}$ "), meets the condition P_1 **and** P_2 **and** ... **and** P_{c_a} (resp. P_{c_a+1} **and** P_{c_a+2} **and** ... **and** P_c) and a_i (resp. b_i, $i=1,2,...,n$) is set to 0 otherwise. In this way, the accurate answer to aggregate query on global table S is the scalar product $\mathbf{a}\cdot\mathbf{b}$ which can be computed by employing scalar product protocol (see section 2.3) on private binary vector $\mathbf{a}$ of C_1 and secret binary vector $\mathbf{b}$ from client C_2.

	A_1	A_2	...	A_k
1	a_{11}	a_{12}	...	a_{1k}
2	a_{21}	a_{22}	...	a_{2k}
...	...	...	...	...
n	a_{n1}	a_{n2}	...	a_{nk}

(1) Sub-table S_1

	A_{k+1}	A_{k+2}	...	A_m
1	$a_{1,k+1}$	$a_{1,k+2}$	...	a_{1m}
2	$a_{2,k+1}$	$a_{2,k+2}$	...	a_{2m}
...	...	...	...	...
n	$a_{n,k+1}$	$a_{n,k+2}$	...	a_{nm}

(2) Sub-table S_2

Figure 7. Vertically Partitioned Data $(1 \leqslant k < m)$

According to the above-mentioned, our method of dealing with privacy-preserving *count* aggregate query on heterogeneous data is illustrated as following.

1. Local Computation: C_1 and C_2 separately compute private binary vectors $\mathbf{a}$ and $\mathbf{b}$ based on their confidential sub-table S_1 and S_2 at respective local site.
2. Cooperative Computation: They cooperatively perform scalar product protocol (see section 2.3) to compute the scalar product of confidential binary vectors $\mathbf{a}$ and $\mathbf{b}$, such that C_1 obtains u and C_2 receives v, which meet $u = \mathbf{a} \cdot \mathbf{b} + v$. Then they publish $u - v$ which is the accurate answer to original *count* aggregate query on the table S.

The Correctness Analysis: To illustrate $u - v$ is the exact answer to *count* aggregate query on the table S, we need to consider the following factors.

As can be seen from Table I, the value of $a_i b_i$ is 1 if and only if $a_i \wedge b_i$ is true, as a result, $\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i$ is the exact count of rows of the table S which satisfy the global selecting condition

$$P_1 \textbf{ and } P_2 \textbf{ and } ... \textbf{ and } P_c.$$

Therefore, $u - v$, equal to $\mathbf{a} \cdot \mathbf{b}$, is the accurate answer to *count* aggregate query on the table S when each party exactly follows the steps. That is, the scheme for privacy-preserving *count* aggregate query of OLAP over heterogeneous data is correct and it is able to return the accurate result rather than the exact value's maximum likelihood estimator in [1].

TABLE I. $a_i \wedge b_i$ AND $a_i b_i$

a_i	0	0	1	1
b_i	0	1	1	0
$a_i \wedge b_i$	false	false	true	false
$a_i b_i$	0	0	1	0

The Security Analysis: The novel scheme is secure if and only if each participant's privacy will be well preserved and no one can find out other party's confidential information.

Since scalar product protocol [5] we employ is secure based on the security of Paillier's homomorphic encryption scheme [17], therefore, the private data of neither of participants will be disclosed and our approach to privacy-preserving *count* aggregate query on vertically partitioned data is secure.

The Complexity Analysis: The communication overhead of the above scheme is about nb_0 where b_0 is the bit length of an encrypted item and the computation cost in the cooperative computation is less than $7n$.

To sum up, it is clear that the novel scheme for privacy-preserving *count* aggregate query of OLAP over vertically partitioned data will return accurate result based on scalar product protocol [5], which is superior to the exact value's maximum likelihood estimator in [1]. Besides, the foregoing analysis of complexity shows our new scheme is efficient in computation cost and communication overheads.

V. CONCLUSION AND FUTURE WORK

In the paper, we have proposed a new model based on SMC protocols for PPOLAP. According to the new model, we put forward approaches to privacy preserving *count* aggregate query of OLAP over both horizontally partitioned data and vertically partitioned data. Additionally, we proved the security and correctness by theoretical analysis and we also analyzed communication overheads and computation complexity of our methods. No privacy loss and exact accuracy are two main advantages of our schemes.

For the future work, we will develop other privacy preserving aggregate operations except *count* according to the PPOLAP model based on SMC protocols.

ACKNOWLEDGMENT

This work was supported by the Major Research Plan of the National Natural Science Foundation of China (No. 90818005), the National Natural Science Foundation of China (Nos. 60903217 and 60773032), and the China Postdoctoral Science Foundation funded project (No. 20090450701).

REFERENCE

[1] R. Agrawal and R. Srikant, D. Thomas. Privacy Preserving OLAP. *Proc. of SIGMOD* 2005, Baltimore, Maryland, USA, 14-16 June, 2005, pp. 251-262.

[2] A. C. Yao. Protocols for secure computations. *Proc. of 23rd Annual IEEE Symposium on Foundations of Computer Science*. Los Alamitos: IEEE Computer Society Press, 1982, pp. 160-164.

[3] O. Goldreich. *Fotmdations of Cryptography: Volume II, Basic Applications*. Cambridge: Cambridge University Press, 2004.

[4] Bart Goethals, Sven Laur, *et al.* On private scalar product computation for privacy-preserving data mining. *Proc. of the Seventh Annual International Conference in Information Security and Cryptology*, LNCS. Springer-Verlag, 2004, pp. 104-120.

[5] R. Wright and Z. Yang. Privacy-preserving Bayesian network structure computation on distributed heterogeneous data. *Proc. of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and*

Data Mining, ACM Press, 2004, pp. 713-718.

[6] A. Amirbekyan, V. Estivill-Castro, A New Efficient Privacy-Preserving Scalar Product Protocol, *Proc. of 6th Australasian Data Mining Conference*, Gold Coast, Australia, 2007, pp. 209-214.

[7] Youwen Zhu, Liusheng Huang, *et al*. Privacy-preserving Practical Convex Hulls Protocol, *Proc. of 2008 Japan-China Joint Workshop on Frontier of Computer Science and Technology*, Nagasaki, Japan, 27-28 Dec. 2008, pp. 10-16.

[8] Youwen Zhu, Liusheng Huang, *et al*. Privacy-Preserving Approximate Convex Hulls Protocol, *Proc. of International Workshop on Education Technology and Computer Science*, Wuhan, China, 7-8 Mar. 2009, pp. 208-214.

[9] Youwen Zhu, Liusheng Huang, *et al*. Relation of PPAtMP and Scalar Product Protocol and Their Applications, *The 15th IEEE symposium on Computers and Communications*, Riccione, Italy, 2010.

[10] Wenliang Du and Mikhail J. Atallah. Privacy-Preserving Statistical Analysis. *Proc. of the 17th Annual Computer Security Applications Conference*, New Orleans, Louisiana, USA, December 10-14 2001, pp. 102-110.

[11] Wenliang Du and Mikhail J. Atallah. Protocols for Secure Remote Database Access with Approximate Matching, volume 2 of *Advances in Information Security*, page 192. Kluwer Academic Publishers, Boston, 2001.

[12] Jaideep Vaidya and Chris Clifton. Privacy Preserving Association Rule Mining in Vertically Partitioned Data. *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, July 23-26 2002, pp. 639-644.

[13] Wenliang Du and Zhijun Zhan. A Practical Approach to Solve Secure Multi-party Computation Problems. *Proc. of New Security Paradigms Workshop*, ACM Press, Virginia Beach, Virginia, USA, Sep. 23-26 2002, pp. 127-135.

[14] Y. Lindell and B. Pinkas. Privacy Preserving Data Mining. *Proc. of the 20th Annual International Cryptology Conference on Advances in Cryptology*, Springer-Verlag London, UK, 2000, pp. 36-54.

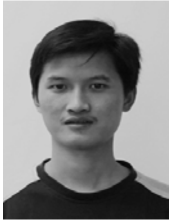
[15] M. Barni, C. Orlandi, and A. Piva. A privacy-preserving protocol for neural-network-based computation. *Proc. of the 8th workshop on Multimedia and security*, ACM, 2006.

[16] F. Kerschbaum and A. Schaad. Privacy-preserving social network analysis for criminal investigations. *Proc. of the 7th ACM workshop on Privacy in the electronic society*, ACM New York, NY, USA, 2008, pp. 9-14.

[17] P. Paillier, Public key cryptosystems based on composite degree residuosity classes, *Advances in Cryptology-Eurocrypt* 1999, LNCS 1592, Springer, 1999, pp.223-238.

[18] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270-299, 1984.

[19] I. Ioannidis, A. Grama, M. Atallah, A secure protocol for computing dot-products in clustered and distributed environments, *Proc. of the 2002 International Conference on Parallel Processing*, Vancouver, British Columbia, 2002, pp. 379-384.



Youwen Zhu, born in 1986, received his B.Sc. in both Computer Science and Sci-Tech Policy and Communication from University of Science and Technology of China in 2007. Currently, he is a Ph. D. candidate in School of Computer Science and Technology from University of Science and Technology of China. His main research interests include information security and wireless senor network.
E-mail: zhuwy@mail.ustc.edu.cn

Liusheng Huang, born in 1957, is the professor in School of Computer Science and Technology from University of Science and Technology of China. His main research interests include information security and wireless senor network. Prof. Huang has been involved in many academic activities including reviewing articles for journals and conferences, serving as a member of program committee for information security and wireless senor network related conferences, and supervising dozens of students.

Wei Yang, born in 1978, received his Ph.D. in Computer Science and Technology from University of Science and Technology of China in 2007. At present, he is a post-doctor in School of Computer Science and Technology from University of Science and Technology of China. His main research interests include information security and Quantum Information.

Fan Dong, born in 1983, received his B.Sc. in information security from University of Science and Technology of China in 2005. Currently, he is a Ph. D. candidate in School of Computer Science and Technology from University of Science and Technology of China. His research interests include information security and Trustworthy Computing.