Privacy-preserving Technology and Its Applications in Statistics Measurements[*]

Yifei Yao
Depart.of Comp. Sci. & Tech., USTC
NHPCC 416, East Campus
USTC, Hefei, 230026, PRC
Tel: 86-551-3602445
yaoyifei@mail.ustc.edu.cn

Liusheng Huang
Depart.of Comp. Sci. & Tech., USTC
NHPCC 416, East Campus
USTC, Hefei, 230026, PRC
Tel: 86-551-3607431
lshuang@ustc.edu.cn

Wei Yang
Depart.of Comp. Sci. & Tech., USTC
NHPCC 416, East Campus
USTC, Hefei, 230026, PRC
Tel: 86-551-3602445
smartyw@mail.ustc.edu.cn

Yonglong Luo
Depart.of Comp. Sci. & Tech., USTC
NHPCC 416, East Campus
USTC, Hefei, 230026, PRC
Tel: 86-551-3602445
ylluo@ustc.edu.cn

Weiwei Jing
Depart.of Comp. Sci. & Tech., USTC
NHPCC 416, East Campus
USTC, Hefei, 230026, PRC
Tel: 86-551-3602445
wwjing@mail.ustc.edu.cn

Weijiang Xu
Depart.of Comp. Sci. & Tech., USTC
NHPCC 416, East Campus
USTC, Hefei, 230026, PRC
Tel: 86-551-3602445
wjxu@mail.ustc.edu.cn

ABSTRACT

Statistics measurements are of great importance in data set description. Although there have been some papers about statistical analysis, little work focused on the flavors of measurements or privacy-preserving property. In this paper, we consider the applications of secure multi-party computation technology in statistics measurements computation to preserve privacy. Secure protocols of harmonic mean, geometric mean and mode are proposed. Detailed analyses about security and complexity of them are also presented.

Categories and Subject Descriptors

G.3 [**PROBABILITY AND STATISTICS**]: Statistical computing; E.m [**MISCELLANEOUS**].

General Terms

Algorithms, Security, Theory.

INFOSCALE 2007, June 6-8, Suzhou, China
Copyright © 2007 ICST 978-1-59593-757-5
DOI 10.4108/infoscale.2007.211

Keywords

secure multi-party computation (SMC), privacy-preserving, statistics measurements, commutative encryption, data perturbation.

1. INTRODUCTION

Nowadays, privacy-preserving is more and more arresting in cooperation networks [1,3~4]. Privacy-preserving provides methods to find important messages correctly in shared data collection. It turns out to be attractively because it can seek more benefit for participants [2~3]. Meanwhile, secure multi-party computation makes cooperative calculation privately, and prevents participants' data from leaking.

Secure two-party computation (STC) was first investigated by A.C.Yao in [1]. He also proposed a general solution for SMC. From then on, many scholars dive into this field, and lots of fruit for special use of SMC come into being [4~5, 8, 10~11].

Statistic is a subject studying the characters of the whole data set, which is important for understanding the essence of things to direct working. Statistics measurements, which include harmonic mean, geometric mean, mode and so on, is basal in statistic. Mean values tell us the big and small of data which are used frequently in normal work. Weighted mean react on calculating the numerical value with weights. Geometric mean and harmonic mean are used in statistical analysis more and more commonly. Although mode and median reflect the middling level as mean values, they are of more stringency when the swatch warps seriously. Mean values, mode and median are statistics measurement values describing the focusing trend of data. Meanwhile, average bias, collectivity deviation and sample deviation are statistics measurements describing the data's discrete degree.

In 2001, Wenliang Du introduced several applications of SMC in [3], and brought forward correlation and regression analysis problem of privacy-preserving statistical analysis firstly. Then he studied privacy-preserving multivariate statistical analysis in 2004

[*] Supported by the National Natural Science Foundation (No.60573171), the Science Foundation of Anhui Province (No.070412043), the Ph.D. Program Foundation of Ministry of Education of China (No.20060358014), and the China Postdoctoral Science Foundation (No.20060390700).

[5]. He gives a solution in two-party instance with his matrix product protocol. Wenjun Luo constructs a protocol to compute mean value recursively for multi-party, however its computational complexity is not desirable [12]. In 2005, Yonglong Luo et al. brings forward a method to calculate mean with secure sum protocol, while its security proof is not sufficiency [9]. Eike Kiltz, Gregor Leander and John Malone-Lee study two-party solution rigorously and analyze other methods' shortage, however few studies focus on their application significance [6].

In this paper, we study how to get statistics measurement values privately and securely, and this research satisfies the increasing need of privacy-preserving in normal days. In this paper, we describe measurement values, including harmonic mean, geometric mean, weighted mean, mode, median, root mean square (RMS), average bias, collectivity deviation and sample deviation, and solve the privacy-preserving statistics measurements computation with the experience of secure multi-party computation. Then, based on the known knowledge, we discuss and verify the problem comprehensively and concretely. We provide a feasible secure protocol for each measurement, discuss their difference and resemblance, and analyze their security and complexity.

The paper is organized as follows. Section 2 contains the material necessary for understanding the protocols of this paper as well as their context. We give protocols for each statistics measurement in Section 3. Then in Section 4 we discuss the protocols' complexity and prove their security theoretically. At last we conclude the paper in Section 5.

2. PRELIMINARIES

In this section, we introduce the preliminary information for the privacy-preserving protocols. Models, definitions and building blocks are also given.

2.1 Secure Multi-party Computation

In a multi-agent network, secure multi-party computation helps two or more parties complete the synergic calculation without leaking privacy information. Generally speaking, SMC is a distributed cooperation. In this work, each party hold a secret as input, and they want to implement the cooperative computation while getting nothing about other's data except the final result [7].

After [1], the technology of SMC has already come into more and more domains such as data mining [3, 8, 10], private information retrieval (PIR) [10], privacy-preserving computation geometry (PPCG) [3], scientific computation [4], quantum oblivious transfer [11] and so on. Secure multi-party computation for union and join of sets makes SMC useful in data mining. PIR uses the SMC conception for reference to retrieve answer without leaking other information. Privacy-preserving location determinant of two geometry graphics imports SMC into military affairs. With the rapid development of economy, scientific computation and statistical analysis will use SMC technique as one of the basic security tools.

Former methods work on a third-party who is trusted by all parties. A trusted third-party (TTP) can get enough information to complete the calculation and broadcasts the result. But the hypothesis itself is insecure and unpractical. Therefore, an executable protocol which can preserve participants' privacy becomes more and more dramatically. It is known that any secure computation problem can be solved by a circuit protocol, but the size of the corresponding circuit is always too large to realize. So investigators choose to design special protocol for special use.

2.2 Models and Definitions

Computation model: Generally speaking, there exist potential malicious attacks against any multi-party protocol. In this paper, we study the problem under a semi-honest model, in which each semi-honest party follows the protocol with exception that he keeps a record of all its intermediate computations, and he will never try to intermit or disturb with dummy data [7]. The model is practical and useful, because everybody in the cooperation expects the right result but not others' privacy information.

Security model: The classical definition of security is stated in [7] as follows.

Let f be a function that p_i $\left(i=1,\cdots,n\right)$ will compute cooperatively. If there is a protocol Π , for each p_i it can generate a simulator which can get all messages though the process only with its view and output, then it is secure. It is to say:

A protocol Π to compute a function f is secure when it satisfies the conditions as follows:

There exists a probabilistic polynomial-time simulator S_i $(i=1,\cdots,n)$, it holds that

$$\left\{\left(S_i\left(x_i,t_i\right),t_1,t_2,\cdots,t_{i-1},t_{i+1},\cdots,t_n\right)\right\}$$
$$\equiv\left\{view_i^{\Pi}\left(x_1,\cdots x_n\right),v_1,v_2,\cdots,v_{i-1},v_{i+1},\cdots,v_n\right\}$$

where $t_i=f_i\left(x_1,x_2,\cdots,x_n\right)$, $v_i=output_i^{\Pi}\left(x_1,x_2,\cdots,x_n\right)$ .While the party's view consists of its initial input, an auxiliary initial input (which is relevant only for modeling adversarial strategies), its random-tape, and the sequence of messages it has received so far.

In this paper, we denote this security definition equation as $\left(formula*\right)$.

2.3 Building Blocks

Secure_SUM Protocol: Suppose there are $n\left(n\geq 3\right)$ parties $p_1,p_2,\cdots,p_n$ who join in the computation. Each p_i has his private information x_i . They want to calculate the function $\sum_{k=1}^{n}x_k$ together, but no one is willing to leak his secret to others. We can get details about this protocol in [9].

```
Protocol 1 Secure_SUM{
// $n$ consumers compute $X=\sum_{k=1}^{n}x_k$ in security while each has a privacy data $x_i$ .
 $\vee S1$ : Each $p_i$ generates $n$ random shares $x_{i,j}$ for $j=1,2,\cdots,n$, such that $x_i=\sum_{j=1}^{n}x_{i,j}$ ;
//each partner divides his data into $n$ shares at random.
 $\vee S2$ : Each $p_i$ sends $\left(P_i\rightarrow P_j,x_{i,j}\right)$ for $j=1,2,\cdots,n,j\neq i$;
 $\vee S3$ : Each $p_i$ do {
         computes $\hat{x}_i=\sum_{j=1}^{n}x_{j,i}$ ;
//computes the sum of all gathered data.
```

```
        broadcasts ( $x_i$ );
//broadcasts the local sum to other partners.
}
$\vee S4$ : Each $p_i$ computes $X = \sum_{i=1}^{n} \hat{x}_i$ ;
}//end protocol
```

In Protocol 1, $\vee S3$ communicates n^2 times. If the number of messages is 1 at each transfer phase and each data has d bits, then Protocol 1 has a bit complexity of $n^2 d$. However, p_i sends n message in $\vee S2$ and receives n message in $\vee S3$. In $\vee S1$, p_i generates $n-1$ numbers randomly in order to get n random numbers satisfying $x_i = \sum_{j=1}^{n} x_{i,j}$. Then p_i makes $x_{in} = x_i - \sum_{j=1}^{n-1} x_{i,j}$. In $\vee S3$ and $\vee S4$, each partner carries out a sum calculation, so they have the computational complexity of $O(n^2)$ times basic operation totally. We generalize the protocol's performance as theorem 1:

Theorem 1: Protocol 1 has round complexity $2n+3$ for each party, communicational complexity $O(n^2)$ totally, bit complexity $O(n^2 d)$, and time complexity $O(n^2)$ times basic operation.

Data Perturbation: If the input $x \in X$ and r are random and distributed uniformly, then we say $x \times r$ protects x secretly.

Commutative Encryption: A commutative encryption is a pair of encryption functions f and g such that $f(g(v)) = g(f(v))$. Even the encryption is a combination of two functions, each party can apply their function first and still get the same result.

3. PRIVACY-PRESERVING PROTOCOLS FOR STATISTICS MEASUREMENTS

In this section, we design the secure protocol for each statistics measurement value which can preserve partners' privacy.

Let there be n partners $p_1, p_2, \cdots, p_n$ in the computation, p_i has n_i privacy data which we saved as $D_i = \{x_{ik} \mid k = 1,2,...,n_i\}$. They want to have a statistical analysis on the data set $D = \bigcup_{i=1}^{n} D_i$, but each p_i wants to have the guarantee that his D_i will not be obtained by other parties.

3.1 Schemes Using Secure_SUM as Sub-protocol

In this section, we describe protocols using Secure_SUM as a building block.

In statistics measurements, many kinds of calculation are similar to mean where summation is the basic. Harmonic mean, weighted mean, RMS and discrete measurement such as average bias, collectivity deviation and sample deviation all use summation operation. Then, Secure_SUM is important in their security protocols. The tip of this kind of calculation is how to construct the local computation before and after Secure_SUM. It must ensure that all message exchanged is happened along Secure_SUM process. Meanwhile, this kind of protocols has the same security proof as Secure_H-Mean. In conclusion, we can take Secure_SUM into account when there is $\sum g(x_i)$ in the statistics measurements.

3.1.1 Secure_H-Mean Protocol

The harmonic mean of n positive numbers $x_1, x_2, \cdots, x_n$ equals to the inverse of mean value of their reciprocal. We denote it as

$$H = \frac{1}{\frac{1}{n}\sum_{k=1}^{n}\frac{1}{x_k}} = \frac{n}{\sum \frac{1}{X}} .$$

To compute the harmonic mean of partners in D , each p_i sums up his privacy reciprocal firstly. Then they carry out Secure_SUM protocol together to get the sum. At last, everyone calculates the harmonic mean value according to the sum locally.

```
$\vee S1$ : Each $p_i$ computes $H_i = \sum_{k=1}^{n_i} \frac{1}{x_{ik}}$ ;
//each one calculates the sum of all his privacy reciprocal locally.
$\wedge S2$ : $H' \leftarrow Secure\_SUM(H_1, H_2, ..., H_n)$ ;
// $n$ partners carry out Secure_SUM protocol to get the sum.
$\vee S3$ : Each $p_i$ computes $H = \frac{n}{H'} = \frac{1}{\frac{1}{n}\sum_{k=1}^{n}\frac{1}{x_k}}$ ;
}//end protocol
```

In Protocol 2, the total computational cost of $\vee S1$ is $O(n)$ and of $\vee S3$ is $O(n)$. Because protocol 2 calls the Secure_SUM protocol only once in $\wedge S2$, we can get corollary 2 form theorem 1 as follow:

Corollary 2: Protocol 2 has round complexity $2n+5$ for each partner, communicational complexity $O(n^2)$ totally, bit complexity $O(n^2 d)$, and time complexity $O(n^2 + n)$ times basic operation.

3.1.2 Secure_W-Mean Protocol

There are n numbers $x_1, x_2, \cdots, x_n$, and $w_1, w_2, \cdots, w_n$ are their corresponding weight. Then their weighted mean is

$$\overline{x} = \frac{w_1 x_1 + w_2 x_2 + \ldots w_n x_n}{w_1 + w_2 + \ldots + w_n} = \frac{\sum wX}{\sum w} .$$

To compute the weighted mean of partners in D , each p_i sums up his privacy weighted mean firstly. Then they carry out Secure_SUM protocol together to get the sum. At last, everyone calculates the weighted mean value according to the sum locally.

```
**Protocol 3** Secure_W-Mean{
$\vee S1$ : Each $p_i$ computes $W_i = \sum_{k=1}^{n_i} w_{ik} x_{ik}$ and $W_i' = \sum_{k=1}^{n_i} w_{ik}$ ;
//each one calculates the sum of all his privacy weighted mean.
$\wedge S2$ : $W \leftarrow Secure\_SUM(W_1, W_2, ..., W_n)$ ;
        $W' \leftarrow Secure\_SUM(W_1', W_2', ..., W_n')$ ;
// $n$ partners carry out Secure_SUM protocol together to get the sum and the weight sum.
$\vee S3$ : Each $p_i$ computes $\bar{X}=\frac{W}{W'}=\frac{\sum_{i=1}^{n}W_i}{\sum_{i=1}^{n}W_i'}$;
}//end protocol
```

The total computational cost of $\vee S1$ is $O(n)$ and of $\vee S3$ is $O(n)$. Because Protocol 3 invokes the Secure_SUM protocol twice in $\wedge S2$, we can get corollary 3 form theorem 1 as follow:

Corollary 3: Protocol 3 has round complexity $4n+8$ for each partner, communicational complexity $O(n^2)$ totally, bit complexity $O(n^2 d)$, and time complexity $O(n^2+n)$ times basic operation.

3.1.3 RMS and Discrete Measurement

RMS is $\sqrt{\bar{x}^2}=\sqrt{\frac{\sum_{k=1}^{n}x_k^2}{n}}=\sqrt{\frac{\sum X^2}{n}}$, average bias is $\frac{\sum_{k=1}^{n}|x_k-\bar{x}|}{n}$, collectivity deviation is $\sigma^2=\frac{\sum_{k=1}^{n}x_k^2-\frac{\left(\sum_{k=1}^{n}x_k^2\right)^2}{n}}{n}=\frac{\sum_{k=1}^{n}x_k^2}{n}-\mu^2$, and sample deviation is $s^2=\frac{\sum_{k=1}^{n}(x_k-\bar{x})^2}{n-1}$.

RMS, average bias, collectivity deviation and sample deviation are similar to Secure_H-Mean protocol. They can be achieved with the help of Secure_SUM protocol in the same way. So, after modifying protocol 2, we can get corresponding secure protocols for RMS, average bias, collectivity deviation and sample deviation, and we omit them here for concision.

3.2 Protocol Using Data Perturbation

In this section, we describe protocols using data perturbation as a building block.

Secure_G-Mean protocol: The geometric mean G of n positive numbers $x_1,x_2,\cdots,x_n$ equals to the n th root of their production, i.e. $G=\sqrt[n]{x_1\cdot x_2\cdot x_3\cdot\ldots\cdot x_n}$.

To compute the geometric mean of partners in D, it is necessary to get the product of all data firstly. Each p_i generates his auxiliary random number r_i locally for privacy. Then, they calculate cooperatively for $\prod_{i=1}^{n}x_i\cdot r_i$ on the distributed network. At last, they pass the product in turns to divide r_i $(i=1,\cdots,n)$ and get the final $\prod_{i=1}^{n}x_i$.

```
**Protocol 4** Secure_G-Mean{
//notice that $p_{i+k}$ means $p_{(i+k)\bmod n}$.
$\vee S1$ : Each $p_i$ generates $r_i$ at random, sets $leader=0$ and $round=0$;
//each $p_i$ generates $r_i$ at random locally, and sets $leader=0$ as the beginner on the ring, sets $round=0$ to take count of round.
$\wedge S2$ : Run a distributed algorithm as follows:
        upon receiving no message:
if $i=leader$ then
        {$round$++; send $\left\{\left(r_i\cdot\prod_{k=1}^{n_i}x_{ik}\right),round\right\}$ to $p_{i+1}$; }
        upon receiving $M$ from $p_{i-1}$:
if $round<n$ then
        {$round$++; send $\left\{\left(M\cdot r_i\cdot\prod_{k=1}^{n_i}x_{ik}\right),round\right\}$ to $p_{i+1}$;}
if $n\le round<2n$ then
        {$round++$; send $\left\{\left(\frac{M}{r_i}\right),round\right\}$ to $p_{i+1}$;}
if $2n\le round<3n$ then
        {$round++$;send $\{M,round\}$ to $p_{i+1}$;}
if $round=3n$ then terminate.
//all the $n$ partners run the distributed protocol as above to compute their private product.
$\vee S3$ : Each $p_i$ computes $G=\sqrt[n]{\left(\prod_{k=1}^{n_1}x_{1k}\right)\cdot\left(\prod_{k=1}^{n_2}x_{2k}\right)\cdot\ldots\cdot\left(\prod_{k=1}^{n_n}x_{nk}\right)}$;
}//end protocol
```

In Protocol 4, each participant products twice in phase $\wedge S2$, so $\wedge S2$ costs $O(2n)$ product operation. $\vee S1$ could be finished in $O(n)$ and $\vee S3$ computes n th root. Then, we can get corollary 4 as follow:

Corollary 4: Protocol 4 has round complexity $2n+2$ for each partner, communicational complexity $O(n)$ totally, bit complexity $O(nd)$, and time complexity $O(n)$ times basic operation.

3.3 Protocols Using Commutative Encryption

3.3.1 Secure_Mode Protocol

The value or item occurring most frequently in a series of observations or statistical data is called a *mode*. Particularly worth a mention is that the mode does not always exist, and is probable not the unique one if it exists.

To compute the mode of partners in D, commutative encryption is useful. Firstly, each p_i encrypts his owned data with private key then pass to the next partner. After all data are encrypted by all the partners, p_n picks the cipher appears most frequently as the *mode*'s cipher. At last, the participants get *mode*'s plain after decrypting its cipher in turn along the ring.

```
**Protocol 5** Secure_Mode{
//notice that $p_{i+k}$ means $p_{(i+k)\bmod n}$.
$\wedge S1$ : Form the players into a ring structure.
Mark $p_i$ 's data as " $M_i$ "; set $leader = 0$ ; set $round = 0$ ;
//form a directional ring structure for passing message, and set $leader = 0$ as the beginner on the ring, set $round = 0$ to take count of round.
$\wedge S2$ : Run a distributed algorithm as follows:
    1. upon receiving no message:
if $i = leader$ then
  { $round++$ ; send $\{(E_i(M_i),1), round\}$ to $p_{i+1}$ };
    2. upon receiving $M$ from $p_{i-1}$ :
if $round < n$ then
  { $round++$ ; $M' = \Phi$ ;
    for each message pair $(m_j, mark_j)$in $M$
    { $mark_j++$ ; append $(E_i(m_j), mark_j)$ to $M'$ ;
      append $(E_i(x_i),1)$ to $M'$ ;
      send $\{M', round\}$ to $p_{i+1}$ ;}
  }
if $n \le round < 2n$ then
  { $round++$ ; $M' = \Phi$ ;
    for each message pair $(m_j, mark_j)$ in $M$
    {if $mark_j < n$ then
        append $(E_i(m_j), mark_j++)$ to $M'$ ;
      if $mark_j = n$ then
        append $(m_j, n)$ to $M'$ ;
      send $\{M', round\}$ to $p_{i+1}$ ;}
  }
//everyone encrypts data passed form previous.
if$round = 2n$ then
  { $round++$ ;
   $p_i$ chooses the mode of encrypted data set $M$ and mark it as " $M_0$ ";
  send $\{D_i(M_0), round\}$ to $p_{i+1}$ ;}
  if $2n \le round < 3n$ then
      { $round++$ ; send $\{D_i(M), round\}$ to $p_{i+1}$ ;}
  if $3n \le round < 4n$ then
      { $round++$ ; send $M$ to $p_{i+1}$ ;}
  if $round = 4n$ then terminate.
//all participants decrypt the mode's cipher in turn to get the corresponding plain.
}//end protocol
```

In Protocol 5, each participant encrypts all data from others, so $\wedge S2$ costs $O(n)$ times of encryption operation, and costs $O(n)$ times for decrypting mode's cipher. Meanwhile, comparison appears $O(n)$ times in $\wedge S2$. In order to form a ring structure in $\wedge S1$, we need another $O(n)$ communication.

Corollary 5: Protocol 5 has round complexity $2n+2$ for each partner, communicational complexity $O(n^2)$ totally, bit complexity $O(n^2 d)$, and time complexity $O(n)$ times of encryption operation and $O(n)$ times of other basic operation.

3.3.2 Secure_Median Protocol

When a group of data is in ascending order or descending order, the midst one or the average mean of midst two is called median.

There is an experiential formula between mean, median and mode:
1.For a dissymmetrical frequency curve which has only one peak but tiny slope, it holds that $mean - mode = 3 \times (mean - median)$.
2.For a symmetrical frequency curve which has only on peak, mean is the same as median and mode.

```
**Protocol 6** Secure_Median{
$\wedge S1$ : $W_{mean} \leftarrow Secure\_Mean(p_1, p_2, ..., p_n)$ ;
//all the $n$ partners carry out the Secure_Mean protocol to get the mean in security.
$\wedge S2$ : $W_{mode} \leftarrow Secure\_Mode(p_1, p_2, ..., p_n)$ ;
//all the $n$ partners carry out the Secure_Mode protocol to get the mode in security.
$\vee S3$ : Each $p_i$ computes $W = \frac{1}{3}(2 \times W_{mean} - W_{mode})$;
}//end protocol
```

We have corollary 6 form corollary 2 and corollary 5 as follow:
Corollary 6: Protocol 6 has round complexity $4n+8$ for each partner, communicational complexity $O(n^2)$ totally, bit complexity $O(n^2 d)$, and time complexity $O(n)$ times of encryption operation and $O(n)$ times basic operation.

4. ANALYSIS

4.1 Complexity Analysis

There are $n(n \ge 3)$ consumers $p_1, p_2, \cdots, p_n$ who join in the computation. Each p_i has n_i private data. Suppose each transfer phase and each data has d bits. Then, we summarize the complexity of protocols as shown in Table 1.

Table 1. Complexity summary

	Round	Communication	Bit	Time
SUM	$2n+3$	$O(n^2)$	$O(n^2 d)$	$O(n^2)$
H-Mean	$2n+5$	$O(n^2)$	$O(n^2 d)$	$O(n^2+n)$
W-Mean	$4n+8$	$O(n^2)$	$O(n^2 d)$	$O(n^2+n)$
G-Mean	$2n+2$	$O(n)$	$O(nd)$	$O(n)$
Mode	$2n+2$	$O(n^2)$	$O(n^2 d)$	$O(n) Encry + O(n)$
Median	$4n+8$	$O(n^2)$	$O(n^2 d)$	$O(n) Encry + O(n)$

4.2 Security Analysis

An algorithm or protocol is said to be secure if there exists a simulator such that we can simulate the views of all parties on the known input and output, i.e. the output of the simulator is computationally indistinguishable from the real views of the party in the algorithm or protocol. A detailed discussion on security can be found in [7].

First, let us consider the security of protocols using Secure_SUM as building block.

Theorem 7: Secure_H-Mean protocol is privacy-preserving.

Proof: Suppose Secure_SUM protocol (denote it as Π') is privacy-preserving. Then there exists a probabilistic polynomial-time simulator $S_i{}'$ $(i=1,\cdots;n)$ such that for function f', it holds that

$$\left\{\left(S_i{}'(x_i{}',t_i{}'),t_1{}',t_2{}',\cdots,t_{i-1}{}',t_{i+1}{}',\cdots,t_n{}'\right)\right\}$$
$$\equiv\left\{view_i^{\Pi'}{}'(x_1{}',\cdots x_n{}'),v_1{}',v_2{}',\cdots,v_{i-1}{}',v_{i+1}{}',\cdots,v_n{}'\right\}$$

where $t_i{}'=f_i{}'(x_1{}',x_2{}',\cdots;x_n{}')$, $v_i{}'=output_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}')$.

Now, we will construct a probabilistic polynomial-time simulator S_i by dint of $S_i{}'$ in order to simulate $view_i^{\Pi}(x_1,x_2,\cdots,x_n)$ for $(formula*)$ where we note Π as Secure_H-Mean protocol.

Though the protocol, what p_i observes and outputs are as follows:

$$view_i^{\Pi}(x_1,x_2,\cdots,x_n)=\left\{x_i,r^i,m_1^i,\cdots,m_n^i\right\}$$
$$=\left\{\left(x_{i1},x_{i2},\cdots,x_{in_i}\right),view_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}')\right\}$$

where $x_i{}'=\sum_{k=1}^{n_i}\frac{1}{x_{ik}}$,

$$output_i^{\Pi}(x_1,x_2,\cdots,x_n)=f(x_1,x_2,\cdots,x_n)$$
$$=\left\{\frac{N}{output_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}')}\right\}.$$

S_i is formed by the use of $S_i{}'$ and we discuss the process now.

Each p_i has n_i data, and there are n parties, so $N=\sum_{i=1}^{n}n_i$ and $x_i=\sum_{k=1}^{n_i}\frac{1}{x_{in_k}}$. For p_i to get S_i, we sum the known $\frac{1}{x_{i1}},\frac{1}{x_{i2}},\cdots,\frac{1}{x_{in_i}}$ and denote it as x_i. Firstly, $S_i{}'$'s $x_i{}'$ is simulated by x_i. Then, we use $S_i{}'$ by simulating Secure_SUM to get $output_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}')$. At last, $output_i^{\Pi}(x_1,x_2,\cdots,x_n)$ is posed by $\frac{N}{output_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}')}$.

Note that

$$S_i\left(x_i,f_i\left(x_1,x_2,\cdots,x_n\right)\right)$$
$$=\left\{\left(x_{i1},x_{i2},\cdots,x_{in_i}\right),view_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}')\right\},$$

and

$$\left\{S_i\left(x_i,f_i\left(x_1,x_2,\cdots,x_n\right)\right),t_1,t_2,\cdots,t_{i-1},t_{i+1},\cdots,t_n\right\}$$
$$=\left\{\left(x_{i1},x_{i2},\cdots,x_{in_i}\right),view_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}'),H\right\},$$

$$\left\{view_i^{\Pi}\left(x_1,x_2,\cdots,x_n\right),v_1,v_2,\cdots,v_{i-1},v_{i+1},\cdots,v_n\right\}$$
$$=\left\{\left(x_{i1},x_{i2},\cdots,x_{in_i}\right),view_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}'),H\right\},$$

So

$$\left\{S_i\left(x_i,f_i\left(x_1,x_2,\cdots,x_n\right)\right),t_1,t_2,\cdots,t_n\right\}$$
$$\equiv\left\{view_i^{\Pi}\left(x_1,x_2,\cdots,x_n\right),v_1,v_2,\cdots,v_{i-1},v_{i+1},\cdots,v_n\right\},$$

where $t_i=f_i(x_1,x_2,\cdots;x_n)$, $v_i=output_i^{\Pi}(x_1,x_2,\cdots,x_n)$.

And this completes the proof of theorem 7. ■

For Secure_W-mean, Secure_RMS protocol, the corresponding security analysis is similar:

Secure_W-Mean(denote as Π_2):

$$view_i^{\Pi_2}(x_1,x_2,\cdots,x_n)$$
$$=\left\{\left(x_{i1},x_{i2},\cdots,x_{in_i},w_{i1},w_{i2},\cdots,w_{in_i}\right),view_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}')\right\}$$

where $x_i{}'=\sum_{k=1}^{n_i}w_{ik}\cdot x_{ik}$, and

$$output_i^{\Pi_2}(x_1,x_2,\cdots,x_n)=\left\{\frac{1}{N}\times output_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}')\right\}.$$ ■

Secure_RMS(denote as Π_3):

$$view_i^{\Pi_3}(x_1,x_2,\cdots,x_n)$$
$$=\left\{\left(x_{i1},x_{i2},\cdots,x_{in_i}\right),view_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}')\right\}$$

where $x_i{}'=\sum_{k=1}^{n_i}x_{ik}{}^2$, and

$$output_i^{\Pi_3}(x_1,x_2,\cdots,x_n)=\left\{\sqrt{\frac{1}{N}\times output_i^{\Pi'}{}'(x_1{}',x_2{}',\cdots,x_n{}')}\right\}.$$ ■

Second, let us consider the security of protocols using data perturbation.

Theorem 8: Secure_G-Mean protocol is privacy-preserving.

Proof: We prove the Secure_G-Mean (denote as Π)'s privacy-preserving property by constructing a simulator. Though the protocol, what p_i observes and outputs are as follow:

$$view_i^{\Pi}(x_1,x_2,\cdots,x_n)$$
$$=\left\{\left(x_{i1},x_{i2},\cdots,x_{in_i}\right),r_i,M_{i-1}\cdot r_i\cdot\prod_{k=1}^{n_i}x_{ik},M_n\cdot Q_{i-1}\cdot\frac{1}{r_i},M_n\cdot Q_n\right\}$$

where $M_{i-1}=\prod_{j=1}^{i-1}\left(r_j\cdot\prod_{k=1}^{n_i}x_{jk}\right)$, $Q_{i-1}=\frac{1}{\prod_{j=1}^{i-1}r_j}$.

$$output_i^{\Pi}(x_1,x_2,\cdots,x_n)$$
$$=f(x_1,x_2,\cdots,x_n)=\left|\sqrt[n]{M_n\cdot Q_n}\right|.$$

Now, we begin to discuss the process how S_i simulates the protocol.

S_i holds $\left(x_{i1}, x_{i2}, \cdots, x_{in_i}\right)$, then it tosses a coin to decide the r_i ' and chooses R_1, R_2 randomly, sets $R_3 = M_n \cdot Q_n$. For P_i, M_{i-1} is the product of former $i-1$ partners' privacy data and random assistant. So, M_{i-1} is computational indistinguishable from R_1 by the use of data perturbation theorem. Similarly, R_2 replaces $M_n \cdot Q_{i-1}$, R_3 equals to $M_n \cdot Q_n$ .Because R_1, R_2 and R_3 contains P_j $(j \neq i)$'s random assistant, so this replacement can preserve other's privacy.

From the design above, we get that

$$S_i\left(x_i, t_i\right) = \left\{\left(x_{i1}, x_{i2}, \cdots, x_{in_i}\right), r_i, R_1 \cdot r_i \cdot \prod_{k=1}^{n_i} x_{ik}, R_2 \cdot \frac{1}{r_i}, R_3\right\},$$

so

$$\left\{S_i\left(x_i, t_i\right), t_1, t_2, \cdots, t_{i-1}, t_{i+1}, \cdots, t_n\right\}$$
$$= \left\{\left(x_{i1}, x_{i2}, \cdots, x_{in_i}\right), r_i, R_1 \cdot r_i \cdot \prod_{k=1}^{n_i} x_{ik}, R_2 \cdot \frac{1}{r_i}, R_3, \sqrt[N]{R_3}\right\}.$$

Meanwhile,

$$\left\{view_i^{\Pi}\left(x_1, x_2, \cdots, x_n\right), v_1, v_2, \cdots v_{i-1}, v_{i+1}, \cdots, v_n\right\}$$
$$= \{\left(x_{i1}, x_{i2}, \cdots, x_{in_i}\right), r_i, M_{i-1} \cdot r_i \cdot \prod_{k=1}^{n_i} x_{ik},$$
$$M_n \cdot Q_{i-1} \cdot \frac{1}{r_i}, M_n \cdot Q_n, \sqrt[n]{M_n \cdot Q_n}\}.$$

Now, we see that

$$\left\{S_i\left(x_i, t_i\right), t_1, t_2, \cdots, t_{i-1}, t_{i+1}, \cdots, t_n\right\}$$
$$= \left\{view_i^{\Pi}\left(x_1, x_2, \cdots, x_n\right), v_1, v_2, \cdots, v_{i-1}, v_{i+1}, \cdots, v_n\right\}.$$

And this completes the proof of the theorem. ■

At last, let us consider the security of protocols using commutative encryption.

Theorem 9: Secure_Mode protocol is privacy-preserving.

Proof: We prove the Secure_Mode (denote as Π)'s privacy-preserving property by constructing a simulator. Though the protocol, what p_i observes and outputs are as follow:

$$view_i^{\Pi}\left(x_1, x_2, \cdots, x_n\right) = \{$$
$$x_i, E_i\left(x_i\right), E_{i-1}\left(E_{i-2}\left(\cdots E_1\left(x_1\right)\right)\right), \cdots, E_{i-1}\left(x_{i-1}\right),$$
$$E_n\left(E_{n-1}\left(\cdots E_1\left(x_1\right)\right)\right), \cdots, E_n\left(E_{n-1}\left(\cdots E_1\left(x_i\right)\right)\right),$$
$$E_{i+k}\left(E_{i+k+1}\left(\cdots E_{i+k+i+1}\left(x_{i+k}\right)\right)\right)_{k=1,\cdots,n-k},$$
$$E_n\left(E_{n-1}\left(\cdots E_{i-1}\left(x_0\right)\right)\right), mode \qquad \}$$

$$output_i^{\Pi}\left(x_1, x_2, \cdots, x_n\right) = f\left(x_1, x_2, \cdots, x_n\right) = mode.$$

We discuss the process how S_i simulates the protocol below.

Firstly, S_i chooses a commutative encryption algorithm E and encryption key a_i, chooses $n-1$ another key $a_k\left(k \neq i\right)$ randomly. Secondly, according to *mode*, S_i generates a random set $\left\{x_1', x_2', \cdots, x_{i-1}', x_{i+1}', \cdots, x_n\right\}$ (holding two of the $x_k'\left(k \neq i\right)$ equals to *mode*, and each other else is different). However, we need to supply a specific. p_n knows how many times *mode* and other candidates appears. So, S_n generates the random set $\left\{x_1', x_2', \cdots, x_{i-1}', x_{i+1}', \cdots, x_n\right\}$ according to the times each candidates present (except for *mode*, other candidate's value is random).

Then, S_i encrypts x_i with a_i to get $E_i\left(x_i\right)$, and has an encryption to form

$$E_{i-1}\left(E_{i-2}\left(\cdots E_1\left(x_1\right)\right)\right), \cdots, E_{i-1}\left(x_{i-1}\right),$$
$$E_n\left(E_{n-1}\left(\cdots E_1\left(x_1\right)\right)\right), \cdots, E_n\left(E_{n-1}\left(\cdots E_1\left(x_i\right)\right)\right)$$

and $E_{i+k}\left(E_{i+k+1}\left(\cdots E_{i+k+i+1}\left(x_{i+k}\right)\right)\right)_{k=1,\cdots,n-k}$

with $a_k\left(k \neq i\right)$ and $\left\{x_1', x_2', \cdots, x_{i-1}', x_{i+1}', \cdots, x_n\right\}$, where $a_k\left(k \neq i\right)$ is encryption key of E_k and x_k ' is plaintext in stead of x_k .

Because *mode* is the mode of

$$\left\{x_1', x_2', \cdots, x_{i-1}', x_i, x_{i+1}', \cdots, x_n\right\}$$

under our design, then

$E_n'\left(E_{n-1}'\left(\cdots E_{i-1}'\left(x_0\right)\right)\right) = E_n'\left(E_{n-1}'\left(\cdots E_{i-1}'\left(mode\right)\right)\right)$ holds. Meanwhile, for the property of commutative encryption, the mode of $\left\{E_n'\left(E_{n-1}'\left(\cdots E_{i-1}'\left(x_k\right)\right)\right)\right\}_{k=1,\cdots,n}$ equals to the encryption in turn of $\left\{x_1', x_2', \cdots, x_{i-1}', x_i, x_{i+1}', \cdots, x_n\right\}$'s mode.

So, we get that $output_i^{\Pi}'\left(x_1, x_2, \cdots, x_n\right) = mode$.

Now,

$$\left\{S_i\left(x_i, t_i\right), t_1, t_2, \cdots t_{i-1}, t_{i+1}, \cdots, t_n\right\} = \{$$
$$x_i, E_i\left(x_i\right), E_{i-1}'\left(E_{i-2}'\left(\cdots E_1'\left(x_1'\right)\right)\right), \cdots, E_{i-1}'\left(x_{i-1}'\right),$$
$$E_n'\left(E_{n-1}'\left(\cdots E_1'\left(x_1'\right)\right)\right), \cdots, E_n'\left(E_{n-1}'\left(\cdots E_1'\left(x_i\right)\right)\right),$$
$$E_{i+k}'\left(E_{i+k+1}'\left(\cdots E_{i+k+i+1}'\left(x_{i+k}'\right)\right)\right)_{k=1,\cdots,n-k},$$
$$E_n'\left(E_{n-1}'\left(\cdots E_{i-1}'\left(x_0\right)\right)\right), mode \qquad \}$$

Obviously,

$$\left\{S_i\left(x_i, f_i\left(x_1, x_2, \cdots, x_n\right)\right), t_1, t_2, \cdots, t_n\right\}$$
$$\equiv \left\{view_i^{\Pi}\left(x_1, x_2, \cdots, x_n\right), v_1, v_2, \cdots, v_{i-1}, v_{i+1}, \cdots, v_n\right\}.$$

And this completes the proof of the theorem. ■

The proof of theorem 9 has a simple manner that each p_i has only one data x_i but not a series $x_{i1}, x_{i2}, \cdots, x_{in_i}$. In fact, protocol 5 can be performed safely in the latter instance. Even in two-party cooperation it will work well. But the protocols which use Secure_SUM or data perturbation as subprogram must have $n > 3$ participants.

5. CONCLUSION

Privacy-preserving statistics measurements computation is important for secure multi-party statistical analysis, it offers basic tool to calculate conveniently. It is useful in science research and engineering technology.

We have designed several protocols to gain a few kinds of concentrative and discrete measurements such as harmonic mean, geometric mean, mode, average bias and so on. Then, we analyze their complexity and prove their security. Although there have already been some previous works in this domain, our work is more comprehensive and rigorous.

6. REFERENCES

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

[2] A.C.Yao. How to generate and exchange secrets. In *Proc. 27th IEEE Symposium on Foundations of Computer Science*. Los Alamitos: IEEE Computer Society Press, 1986.162-167.

[3] Du Wenliang, and Mikhail J.Atallah. Secure multi-party computation problems and their applications: a review and open problems. In *New Security Paradigms Workshop*. Cloudcroft, New Mexico, USA, September 11-13, 2001, 11-20.

[4] Du Wenliang, and Mikhail J.Atallah. Privacy-preserving cooperative scientific computation. In *Fourteenth IEEE Computer Security Foundations Workshop*. Nova Scotia, Canada, 2001, June 11-13, 273- 282.

[5] Du Wenliang, Yunghsiang S. Han, and Shigang Chen. Privacy-preserving multivariate statistical analysis: linear regression and classification. In *Proceedings of the 4th SIAM International Conference on Data Mining*. Lake Buena Vista, Florida, April 22-24, 2004, 222- 233.

[6] Eike Kiltz, Gregor Leander, and John Malone-Lee. Secure computation of the mean and related statistics. In *Theory of Cryptography Conference (TCC 2005)*. 283- 302.

[7] Goldreich. *Secure multi-party computation(manuscript version1.3)*. http://theory.lsc.mit.edu/~oded, 2002.

[8] Luo Yonglong, Huang Liusheng, Jin Weiwei, and Yao Yifei, An algorithm for privacy-preserving boolean association rule mining. *Acta Electronica Sinica*, 2005 Vol.33 No.5, 900- 903.

[9] Luo Yonglong, Xu zhiyun, and Huang Liusheng. Secure multi-party statistical analysis problems and their applications. *COMPUTER ENGINEERING AND APPLICATIONS*. 2005 Vol.41 No.24, 141- 143.

[10] Rakesh Agrawal, Alexandre Evfimievski, and Ramakrishnan Srikant. Information sharing across private databases. In *Proceedings of the 2003 ACM SIGMOD international conference*. 2003, 96- 97.

[11] Wei Yang, Liusheng Huang, Yifei Yao and Yonglong Luo. Quantum Chosen m out of n Oblivious Transfer. In *Proceedings of CHINACRYPT' 2006*. 37- 45.

[12] Wenjun Luo, and Xiang Li. A study of secure multi-party statistical analysis. In *Proceedings of International Conference on Computer Networks and Mobile Computing*. Shanghai, 2003, 377- 382.