Chinese Journal of Electronics
Vol.20, No.3, July 2011

Efficient Collusion-Resisting Secure Sum Protocol*

ZHU Youwen[1,2], HUANG Liusheng[1,2], YANG Wei[1,2] and YUAN Xing[1]

(1.*National High Performance Computing Center at Hefei, Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China*)

(2.*Suzhou Institute for Advanced Study, University of Science and Technology of China, Suzhou 215123, China*)

Abstract — Secure sum protocol is a significant secure multiparty computation protocol and it has various applications in privacy-preserving distributed multiparty computation. However, most existing secure sum protocols rarely considered how to resist underlying collusion which is a significant practical problem. Urabe *et al.* proposed a collusion-resistant secure sum protocol, but too much cost of communication and computation results in its low performance efficiency. In this paper, we propose security definitions to measure secure multiparty computation protocol's capability of resisting potential collusion. Then, we precisely analyze several previous secure sum protocols' capability of resisting collusion. In addition, considering realistic requirement to resist collusion and performance efficiency needs, we present a novel collusion-resisting secure sum protocol. Theoretical analysis and experimental results confirm that our secure sum protocol is efficient and has strong capability of resisting potential collusion such that it is much superior to previous ones. The communication overheads and computation complexity of our scheme both are linearity of the number of participants. Besides, our protocol's capability of resisting collusion is adjustable according to different security needs.

Key words — Secure multiparty computation, Privacy-preserving, Secure sum, Collusion.

I. Introduction

Secure multi-party computation (SMC) enables a group of participants to perform a cooperative computation based on their private inputs by a special way where each party obtains the cooperative computation's result but no one learns anything about any other party's private inputs. Since the seminal paper[1], SMC has attracted numerous researchers[2−4]. In recent years, SMC has been introduced into various applications, such as data mining, machine learning, computation geometry, statistical analysis, scientific computing, electronic voting, *etc.*, and SMC protocols are employed to protect privacy in practical applications. Nowadays, a great many SMC protocols, including secure sum protocol[5,6], secure comparison protocol[1], scalar product protocol[7] and so on, have severed as the secure building blocks in privacy-preserving distributed multiparty computation protocol.

Secure sum protocol is presented as an example of SMC protocol in Ref.[6]. The goal of secure sum protocol is that three or more parties securely obtain the grand sum of each private number of all participants. In addition, it is a prerequisite that each participating party's private number is not disclosed to anybody else including other participants. Secure sum protocol is a significant secure building block and it has been widely used in privacy preserving multiparty computation over distributed data, including privacy-preserving data mining[5,8,9], privacy-preserving machine learning[10], privacy-preserving collaborative social networks[11] and so on. Kantarcioglu and Clifton[5,9] put forwards a scheme to securely mine association rules over horizontally distributed data. In a horizontally partitioned database, the global support count is equal to the sum of all the local support count. Secure sum protocol is used to compute the global support count of a rule such that participants collaboratively form the global rule and none of local support count is revealed. Similarly, the global confidence of a rule is calculated. A privacy-preserving back-propagation network training protocol is proposed in Ref.[10]. In the main process, the internal network weights are iteratively adapted until the given condition is reached. To ensure privacy, secure sum protocol is used to securely compute the total sum of private coefficient matrix in each iterative process. Privacy-preserving social networks analysis[11−14] is a relatively new research area. Ref.[11] presented a privacy-preserving solution for collaborative social network construction. The key step for constructing a collaborative social network is compare the joint results from different participants with a given threshold and secure sum protocol is employed to securely compute the summation of all the participating parties' private values. More applications of secure sum protocol can be found in Refs.[15-17].

Because of its various applications to serve as a significant secure building block, several secure sum protocols[5,6,9,11,18−20] have been proposed. There are mainly

*Manuscript Received Feb. 2010; Accepted Mar. 2011. This work is supported by the Major Research Plan of the National Natural Science Foundation of China (No.90818005), the National Natural Science Foundation of China (No.60903217), and the Academic New Awards for Ph.D. students in University of Science and Technology of China (No.ZC9850320155).

four existing secure sum protocols which are respectively denoted as simple secure sum protocol[5,6,9], secure sum protocol based on homomorphic encryption[11], secure sum protocol with penalty[18] and UWKT secure sum protocol[20]. The simple secure sum protocol has least communication overheads and lowest computation complexity, but it is incapable of resisting collusion and a collusion containing two parties could violate privacy. The secure sum protocol based on homomorphic encryption, proposed in Ref.[11], has just a little advance in resisting collusion, nevertheless, the secure sum protocol is too computationally intensive owing to employing homomorphic cryptosystem and it is still unsecure while underlying collusion occurs. To prevent from known collusion, Kargupta *et al.*[18] propose a secure sum protocol with penalty through a game theoretic approach. However, collusion, in reality, is uncharted while performing secure sum protocol and some cankered participants may collude after the protocol is end. The secure sum protocol with penalty can't hold back any collusion after the protocol. Another secure sum protocol, denoted as UWKT secure sum protocol, is proposed in Ref.[20]. UWKT secure sum protocol is able to resist underlying collusion, however, too much cost of communication and computation results in its low practicability.

It is a realistic problem that some cankered participants in SMC may stealingly collude to lay hands on other parties' privacy. In this paper, we propose the security definitions to measure SMC protocol's capability of resisting collusion. Each SMC protocol's capability of resisting collusion totally depends on the probability that each private input is revealed while underlying collusion occurs. Then, we precisely analyze existing secure sum protocols' capability of resisting potential collusion. Secure sum protocol may be invoked repeatedly in a privacy-preserving distributed multiparty computation protocol. Therefore, an efficient collusion-resisting secure sum protocol is a significant secure building block to boost performance of many privacy-preserving distributed multiparty computation protocols. Considering realistic requirement to resist collusion and performance efficiency needs, we propose a novel secure sum protocol in this paper. Theoretical analysis and experimental results confirm that our secure sum protocol is efficient and has strong capability of resisting latent collusion such that it is much superior to previous ones. Besides, the new secure sum protocol's capability of resisting collusion is adjustable according to different security needs. The communication overheads and computation complexity both are linearity of the number of participants.

The main contributions of this paper are: (1) we propose security definitions to quantitatively analyze SMC protocols' capability of resisting collusion; (2) we precisely analyze previous secure sum protocols' capability of resisting collusion in detail; (3) a novel collusion-resisting secure sum protocol is proposed, theoretical analysis and experimental results confirm that our secure sum protocol is efficient and has strong capability of resisting latent collusion such that it is much superior to previous ones.

The rest of the paper is organized as follows. Section II describes security model and security definitions to measure SMC protocol's capability of resisting collusion is proposed in the section. We analyze the performance of four previous secure sum protocols in Section III. In Section IV, we propose a novel collusion-resisting secure sum protocol, explain its correctness and theoretically analyze its capability of resisting potential collusion, communication overheads and computation complexity. Section V confirms performance efficiency and our protocol's capability of resisting underlying collusion by experimental results. Section VI provides conclusion and some directions for future work.

II. Secure Model and Security Definition

Different from conventional cryptography which is concerned with participants who want to communicate privately and authentically over an insecure channel, SMC devotes to preventing from privacy violation caused by potential cankered participants.

In the setting of traditional SMC, there are two different security models: semi-honest model and malicious model[3]. In the semi-honest model, it is assumed that participating parties strictly follow the protocols, no one colludes and each participant could keep a record of the intermediate data which he receives to find out potential confidential information. Each participant may select arbitrary operation or refuse to do anything in the malicious model. However, in reality, participants may follow the protocols to obtain exact output and some of them may well collude after the protocols to work out private data of other party. In this paper, we assume that participating parties correctly follow the protocols but they may try to collude with several participants after the protocol to infer private input of others. In the situation, a significant problem is how to resist potential collusions.

To quantificationally measure SMC protocol's capability of resisting collusion, the following definitions are proposed.

Definition 1 Let $\boldsymbol{P} = \{P_1, P_2, \cdots, P_n\}$ be the set of n participating entities in a SMC protocol f. Suppose P_i, $P_j \in P (i \neq j)$, $\boldsymbol{C} \subset \boldsymbol{P}$, $2 \leq c < n$, $P_i \notin \boldsymbol{C}$ and $|\boldsymbol{C}| = c$ where $\boldsymbol{C}$ is a collusion group to be selected after executing f by P_j according to his view (the view consists of his inputs, outputs and all the intermediate records he received), to deduce P_i's private information. If p is the probability that a collusion containing each party in $\boldsymbol{C}$ can compute the private input of P_i, we say that the private input of P_i is $(n, c, 1-p)$-collusion resisting in the SMC protocol f.

Intuitively, Definition 1 indicates how secure one participant's private input in a SMC protocol is while potential collusion to find out the private data occurs. In a SMC protocol, each private input may be protected on different degree, then, a SMC protocol's capability of resisting collusion is defined as the average collusion-resisting degree of each private input. Based on the view and Definition 1, we propose Definition 2 as follows.

Definition 2 Let $\boldsymbol{P} = \{P_1, P_2, \cdots, P_n\}$ be the collection of n participating parties in a SMC protocol f. If P_i's $(i = 1, 2, \cdots, n)$ private input is $(n, c_0, 1-p_i)$-collusion resisting in the protocol f, we say that the SMC protocol f is $\left(n, c_0, 1 - \frac{1}{n}\sum_{i=1}^{n} p_i\right)$-collusion resisting and $1 - \frac{1}{n}\sum_{i=1}^{n} p_i$ is called f's capability of resisting (n, c_0)-collusion.

Relative to Definition 1, Definition 2 states the overall se-

curity of a SMC protocol while resisting underlying collusion. The capability of resisting collusion is the average effectiveness that a SMC protocol preserves each private input while underlying collusion occurs. In the next section, we will confirm that most exiting secure sum protocols have hardly capability of resisting collusion and a secure sum protocol with strong capability of resisting collusion is inefficient because of its expensive communication overheads and computation complexity.

III. Previous Secure Sum Protocols

Secure sum protocol is a useful SMC basic protocol. In the setting of secure sum protocol, there are n individual participants: $P_1, P_2, \cdots, P_n$ and P_i $(i = 1, 2, \cdots, n)$ has a private natural number x_i. They plan to securely compute the sum of the n confidential numbers while no private data is revealed. That is, each participating party obtains nothing but the summation $\sum_{i=1}^{n} x_j$ via performing the protocol. Secure sum protocol has been achieved by several schemes[5,6,9,11,18−20]. There are mainly four existing secure sum protocols which are respectively denoted as simple secure sum protocol[5,6,9], secure sum protocol based on homomorphic encryption[11], secure sum protocol with penalty[18] and UWKT secure sum protocol[20]. However, prior three secure sum protocols have hardly capability of resisting underlying collusion. UWKT secure sum protocol is expensive in communication overheads and computation complexity. More details are presented as below.

1. Simple secure sum protocol

Refs.[5, 6, 9] proposed a secure sum protocol in a simple way, we denote it as Simple secure sum protocol (Simple-SSP). In Simple-SSP, P_i generates a random to distort his private input and the summation is computed along the cycle: $P_1 \to P_2 \to \cdots \to P_n \to P_1$.

In Ref.[5], it has been shown that $\forall i \in \{1, 2, \cdots, n\}$, the collusion of P_i's immediate predecessor and immediate successor can find out the private input: x_i. As a result, Simple-SSP is $(n, 2, 0)$-collusion resisting and it has hardly any capability of resisting collusion.

2. Secure sum protocol based on homomorphic encryption

A public key encryption scheme (Enc, Dec) where Enc and Dec are respectively polynomial-time algorithms for encryption and decryption, is homomorphic if the following condition holds: $Dec(Enc(m_1) * Enc(m_2)) = m_1 + m_2$, where m_1 and m_2 are any plaintext items. That is, $Enc(m_1) * Enc(m_2) \doteq Enc(m_1 + m_2)$, where $\doteq$ denotes that they hide the same plaintext item. As a result, $Enc(m_1 + m_2)$ can be computed from $Enc(m_1)$ and $Enc(m_2)$ such that the secret numbers m_1 and m_2 aren't disclosed in a homomorphic cryptosystem. Based on homomorphic encryption system[21], Zhan *et al.*[11] proposed another secure sum protocol, which is denoted as SSP_HE for short. Suppose integer $X > \sum_{j=1}^{n} x_j$. In SSP_HE, P_n generates a homomorphic cryptosystem (Enc, Dec). Then, P_i $(i = 1, 2, \cdots, n-1)$ generates a random integer R_i, calculates $Enc(x_1 + \cdots + x_i + (R_1 + \cdots + R_i) \times X)$ and sends it to P_{i+1}. At last, P_n obtains

$$Dec(Enc(\textstyle\sum_{j=1}^{n} x_j + X^* \sum_{j=1}^{n} R_j) \bmod X = \sum_{j=1}^{n} x_j.$$

SSP_HE has some advance in security relative to Simple-SSP. However, it has a big problem that the so-called "digital envelope" R_i helps nothing with hiding private integer x_i, such as, x_1 can be computed by $(x_1 + R_1 * X) \bmod X$. As a result, the security of SSP_HE is mainly based on that homomorphic encryption scheme[21] is semantically secure. Privacy could be violated when P_n colludes with some participating parties. Concretely speaking, P_n can find out P_1's (resp. P_{n-1}'s) private data x_1 (resp. x_{n-1}) by colluding with P_2 (resp. P_{n-2}), x_i $(i = 2, 3, \cdots, n-2)$ will be revealed if P_n colludes with P_{i-1} and P_{i+1}, and the private input of P_n will not be disclosed unless other $(n-1)$ participants collude. Then,

$$1 - (1/n) * (100\% + (n-3) * 0 + 100\% + 0) = 1 - 2/n$$

Thus SSP_HE is $(n, 2, 1-2/n)$-collusion resisting, but its capability of resisting $(n, 3)$-collusion is merely $1 - (1/n) * ((n-1) * 100\% + 0) = 1/n$. Another disadvantage of SSP_HE is its high computation cost owing to employing homomorphic cryptosystem.

3. Secure sum protocol with penalty

To prevent from colluding, Ref.[18] proposed a secure sum protocol with penalty through a game theoretic approach. It penalizes known colluding parties by increasing the cost of communication and computation. However, collusion, in reality, is uncharted while performing secure sum protocol and some cankered participants may collude until the protocol is end. Therefore, secure sum protocol with penalty can't hold back any collusion after the protocol. If no collusion is indicated before the protocol, secure sum protocol with penalty will be the same as Simple-SSP[5,6,9].

4. UWKT secure sum protocol

Some works[19,20] have devoted to developing collusion-resistant secure sum protocol by a technique of sharing and masking. Among of them, the secure sum protocol in Ref.[20], denoted as UWKT-SSP for short, has strongest capability of resisting underlying collusion.

Owing to the internal property of secure sum protocol, if any $n-1$ parties collude, the private data of remaining one will be disclosed. Accordingly, a secure sum protocol is $(n, n-2, 100\%)$-collusion resisting at best. It has been shown that P_j can't learn the private data $x_i (i \neq j)$ unless it colludes with all participants except P_i in UWKT-SSP, that is, UWKT-SSP is $(n, n-2, 100\%)$-collusion resisting. However, too much cost of communication and computation results in its low performance efficiency, its communication overheads is $b_0 n(n-1)/2$ where b_0 is the bit length of a private number and its computation complexity reaches up to $O(n^2)$.

IV. Collusion-Resisting Secure Sum Protocol

1. Collusion-Resisting secure sum protocol

In reality, some cankered participants in SMC may stealingly collude to lay hands on other parties' privacy and secure sum protocol may be invoked repeatedly in a privacy-preserving distributed multiparty computation protocol. Therefore, an efficient collusion-resisting secure sum protocol is a significant secure building block to boost performance of many privacy-preserving distributed multiparty computation protocols. Considering realistic requirement to

resist collusion and performance efficiency needs, we propose Collusion-resisting secure sum protocol (CR-SSP).

In a three-party secure sum protocol, if any two participants collude, another party's private input can be computed. As a result, this paper is concerned with secure sum protocol that has four or more participants. The route of previous secure sum protocols is deterministic and public, but the masking process in CR-SSP is probabilistic and secret, with the help of which CR-SSP's capability of resisting collusion has a great advance relative to existing secure sum protocols. In CR-SSP, there are two phases. Each participator's private number is clandestinely masked several uncertain times while the sum of masked values is identically equal to the original sum in phase 1 and the summation of masked numbers will be computed in phase 2. The formal protocol is presented as protocol 1.

Protocol 1 Collusion-resisting secure sum protocol (CR-SSP)

Input of node P_i $(i = 1, 2, \cdots, n)$: a private natural number x_i

Output of node P_i $(i = 1, 2, \cdots, n)$: the summation $\sum_{j=1}^{n} x_j$

Remarks Suppose $n > 3$ and there is a large enough number N, $N > \sum_{j=1}^{n} x_j$.

Phase 1 mask the private inputs

Step 1.1 Each party P_i $(i = 1, 2, \cdots, n)$ privately selects, by employing a uniformly random function, t different integers $r_{i\_1}, r_{i\_2}, \cdots, r_{i\_t} \in \{1, 2, \cdots, n\} \backslash \{i, pred(i), succ(i)\}$ where t $(0 < t \ll n - 3)$ is a given constant positive integer, $pred(i) = ((i + n - 2) \bmod n) + 1$ and $succ(i) = (i \bmod n) + 1$. Then, P_i uniformly generates other t_0 different random numbers $v_{i\_1}, v_{i\_2}, \cdots, v_{i\_t} \in \mathbb{Z}_N$. It is emphasized that r_{i_k} and v_{i_k} $(k = 1, 2, \cdots, t)$ are P_i's confidential information. Then, P_i clandestinely sends v_{i_k} $(k = 1, 2, \cdots, t)$ to Pr_{i_k} such that other parties except Pr_{i_k} and P_i itself, learn nothing about v_{i_k} and r_{i_k}. At the end of this sub-step, P_i sets $m_i = x_i$.

Step 1.2 If P_i $(i = 1, 2, \cdots, n)$ receives a integer, denoted as v_j, from P_j, he uniformly generates a random boolean variable q_{i_j}. Then, P_i set $m_i = m_i - v_j \pmod N$ if $q_{i\_j} = true$ and $m_i = m_i + v_j \pmod N$ if $q_{i\_j} = false$. P_i secretly sends the random boolean variable q_{i_j} to P_j.

Step 1.3 When P_i $(i = 1, 2, \cdots, n)$ receives a clandestine boolean variable q_{j_i} from P_j $(j \in \{r_{i\_1}, r_{i\_2}, \cdots, r_{i\_t}\}$, assume $j = r_{i\_k})$, he sets $m_i = m_i + v_{i\_k} \pmod N$ if $q_{j\_i} = true$ and $m_i = m_i - v_{i\_k} \pmod N$ if $q_{j\_i} = false$.

Phase 2 compute the summation

Step 2.1 P_1 uniformly selects a random integer $r \in \mathbb{Z}_N$, sets $s = m_1 + r \pmod N$, and sends s to P_2.

Step 2.2 P_i $(i = 2, 3, \cdots, n)$ computes $s = s + m_i \pmod N$, and sends s to P_{i+1}.

Step 2.3 P_n computes $s = s + m_n \pmod N$ and sends s to P_1.

Step 2.4 P_1 obtains $sum = s - r \pmod N$ and sends sum to $P_2, P_3, \cdots, P_n$.

2. Analysis of CR-SSP

In this sub-section, we will theoretically analyze the correctness, communication cost and computation complexity of CR-SSP. Besides, the new scheme's capability of resisting collusion is presented in detailed.

Correctness CR-SSP is correct if and only if sum is the exact sum of each participant's private input in protocol 1.

Theorem 1 In protocol 1, $sum = \sum_{j=1}^{n} x_j$.

Proof In phase 1 of protocol 1, P_i sets $m_i = x_i$ in the beginning. With a view to each pair (r_{i_k}, v_{i_k}) $(i = 1, 2, \cdots, n; k = 1, 2, \cdots, t)$, Pr_{i_k} sets $m_{r_{i\_k}} = m_{r_{i\_k}} - v_{i\_k} \pmod N$ (resp. $m_{r_{i\_k}} = m_{r_{i\_k}} + v_{i\_k} \pmod N$) if his random boolean variable $q_{r_{i\_k}\_i}$ is $true$ (resp. $false$); P_i sets $m_i = m_i + v_{i\_k} \pmod N$ (resp. $m_i = m_i - v_{i\_k} \pmod N$) if $q_{r_{i\_k}\_i}$ is $true$ (resp. $false$) by contraries. As a result, $\sum_{j=1}^{n} m_j = \sum_{j=1}^{n} x_j$ still holds after phase 1. It is easy to say that sum equals $\sum_{j=1}^{n} m_j$ in phase 2 of protocol 1.

Therefore, $sum = \sum_{j=1}^{n} x_j$ holds.

Communication overheads and computation complexity. If b_0 is the bit length of a private number, the bit cost in phase 1 is $(b_0 + 1)t_0 n$ and the communication overheads of phase 2 is $b_0 n$. Then, the total communication cost in CR-SSP is $(b_0 + 1)t_0 n + b_0 n$ bits.

The computation complexity of both phase 1 and phase 2 are $O(n)$. Therefore, the computation complexity of CR-SSP is $O(n)$.

The communication overheads and computation complexity of CR-SSP and previous secure sum protocols are displayed in Table 1. It is shown that CR-SSP requires a little more communication cost than Simple-SSP, but CR-SSP is much more efficient than UWKT-SSP in both communication overheads and computation cost.

Table 1. Comparison of communication overheads and computation complexity

Protocol	Communication overheads (bit)	Computation complexity
Simple-SSP	$b_0 n$	$O(n)$
SSP_HE *	$b_0 n + b_{key}(n - 1)$	$O(H^* n)$
UWKT-SSP	$b_0 n(n - 1)/2$	$O(n^2)$
CR-SSP	$(b_0 + 1)t_0 n + b_0 n (t_0 \ll n - 3)$	$O(n)$

*Suppose the key of homomorphic cryptosystem is b_{key} bits and its computation complexity is $O(H)$.

Capability of resisting collusion In phase 2 of protocol 1, the sum $\sum_{j=1}^{n} m_j$ is computed along the route: $P_1 \rightarrow P_2 \rightarrow \cdots \rightarrow P_n \rightarrow P_1$. Apparently, the immediate predecessor of P_i $(i = 1, 2, \cdots, n)$ is $P_{pred(i)}, pred(i) = ((i + n - 2) \bmod n) + 1$ and the immediate successor of P_i is $P_{succ(i)}, succ(i) = (i \bmod n) + 1$. If $P_{pred(i)}$ and $P_{succ(i)}$ collude, m_i could be computed in phase 2. Nevertheless m_i has been masked in phase 1, and x_i cannot be inferred from m_i unless all parties, which have secretly communicated with P_i in phase 1, collude.

Definition 3 The masking set of P_i is defined as $\mathbf{RCV}(i)$. $\mathbf{RCV}(i) \subset \{i = 1, 2, \cdots, n\}$ and $\forall k \in \mathbf{RCV}(i)$ if and only if $i\{r_{k\_1}, r_{k\_2}, \cdots, r_{k\_t}\}$ or $k\{r_{i\_1}, r_{i\_2}, \cdots, r_{i\_t}\}$ in CR-SSP.

Intuitively, $\mathbf{RCV}(i)$ is the collection of parties that P_i has sent message to each other in phase 1 of CR-SSP. If P_j wants to calculate the private number x_i of P_i by collusion, he has to collude with each party in $\{P_k | k \in \mathrm{RCV}(i) \cup \{pred(i), succ(i)\}\}$. $pred(i)$ and $succ(i)$ are deterministic, but $\mathbf{RCV}(i)$ is probabilistic and confidential, which is much propitious to resist potential collusion. As participants clandestinely communicates with each other in phase 1 of CR-SSP, then $\mathbf{RCV}(i)$ is confidential. Consequently, according to P_j's view, each party of $\{P_k | k \in \{i = 1, 2, \cdots, n\} \backslash \{i, pred(i), succ(i)\}, k \neq j\}$ are in $\{P_k | k \in \mathrm{RCV}(i)\}$ with the same probability. To collude with parties in $\{P_k | k \in \mathbf{RCV}(i)\}$, P_j can just randomly select some participators to collude with from the set $\{P_k | k \in \{i = 1, 2, \cdots, n\} \backslash \{i, pred(i), succ(i)\}\}$.

We denote $T(i) = |\mathrm{RCV}(i)|$. According to protocol 1, $t \le T(i) \le n-3$. Let $\Pr(k)$ $(t \le k \le n-3)$ be the probability that $T(i)$ comes up to k. We have

$$\Pr(k) = \binom{n-3-t}{k-t} \left(\frac{\binom{n-4}{t-1}}{\binom{n-3}{t}} \right)^{k-t} \left(\frac{\binom{n-4}{t}}{\binom{n-3}{t}} \right)^{n-3-k}.$$

That is,

$$\Pr(k) = \binom{n-3-t}{k-t} \left(\frac{t}{n-3} \right)^{k-t} \left(\frac{n-3-t}{n-3} \right)^{n-3-k} \quad (1)$$

According to Eq.(1), the variable $(k-t)$ obeys to binomial distribution.

If P_j randomly selects m participators, $P_{j_1}, P_{j_2}, \cdots, P_{j_m}$, to collude with from $\{P_k | k \in \{i = 1, 2, \cdots, n\} \backslash \{i, pred(i), succ(i)\}\}$ and $p(m, T(i))$ denotes the probability that $\mathbf{RCV}(i) \subset \{j_1, j_2, \cdots, j_m\}$ holds, then

$$p(m, T(i)) = \begin{cases} 0, & \text{if } m < T(i) \\ \dfrac{\binom{n-3-T(i)}{m-T(i)}}{\binom{n-3}{m}}, & \text{if } T(i) \le m \le n-3 \end{cases} \quad (2)$$

When $\mathbf{RCV}(i) \subset \{j_1, j_2, \cdots, j_m\}$ and P_j colludes with all the parties in $\{p_{j_1}, P_{j_2}, \cdots, P_{j_m}\} \cup \{P_{pred(i)}, P_{succ(i)}\}$, P_i's private input x_i will be revealed. Hence, x_i is $(n, m+2, 1-p(m, T(i)))$-collusion resisting in CR-SSP $(0 \le m \le n-3)$.

Theorem 2 Let $P_{cr}(n, m)$ be CR-SSP's capability of resisting $(n, m+2)$-collusion, then, $P_{cr}(n, m) = 100\%$, if $m < t$ and $P_{cr}(n, m) = 1 - \sum_{k=t}^{m} (\Pr(k)^* p(m, k))$, if $t \le m \le n-3$.

Proof It has been shown that P_i's $(i = 1, 2, \cdots, n)$ private input x_i is $(n, m+2, 1-p(m, T(i)))$-collusion resisting in CR-SSP $(0 \le m \le n-3)$, therefore,

$$P_{cr}(n, m) = 1 - \frac{1}{n} \sum_{i=1}^{n} p(m, T(i)).$$

If $m < t$, then, $\forall i$, $m < t \le T(i)$. According to Eq.(2), $p(m, T(i)) = 0$. Therefore,

$$P_{cr}(n, m) = 1 - \frac{1}{n} \sum_{i=1}^{n} p(m, T(i)) = 100\%.$$

If $t \le m \le n-3$, then, $P_{cr}(n, m) = 1 - \frac{1}{n} \sum_{i=1}^{n} p(m, T(i))$.

According to the Eq.(1),

$$\begin{aligned} P_{cr}(n, m) &= 1 - \frac{1}{n} \sum_{T(i)=t}^{n-3} ((\Pr(T(i)) * n) * p(m, T(i))) \\ &= 1 - \frac{1}{n} \sum_{T(i)=t}^{m} ((\Pr(T(i)) * n) * p(m, T(i))) \\ &= 1 - \sum_{T(i)=t}^{m} (\Pr(T(i)) * p(m, T(i))). \end{aligned}$$

That is, $P_{cr}(n, m) = 1 - \sum_{k=t}^{m} (\Pr(k) * p(m, k))$, which completes the proof.

To intuitively illustrate CR-SSP's capability of resisting collusion, we calculate $P_{cr}(n, m)$ when the pair (n, t) are some specialized values in Fig.1. Theorem 2 and Fig.1 illustrate that CR-SSP has much stronger capability of resisting collusion than Simple-SSP and SSP_HE. Besides, CR-SSP's capability of resisting collusion is practicable and adjustable. The bigger t is, the larger CR-SSP's capability of resisting $(n, m+2)$-collusion $P_{cr}(n, m)$ is. By setting t as an appropriate value, CR-SSP can meet given security needs while requiring as low cost of communication and computation as possible.

According to the number of participants n and request to capability of resisting collusion, the optimal values of t can be computed. For example, there are 200 participants and the request to capability of resisting (200, 162)-collusion is 99.9%. Then, we can find out that $t = 16$ is the minimal value enabling P_{cr} (200, 162)$\ge$99.9% to hold. That is, CR-SSP can meet the above security needs while setting $t = 16$. Accordingly, the communication overhead of CR-SSP is $3400b_0 + 3200$ bits. However, Simple-SSP and SSP_HE cannot achieve the requirements to resist collusion and UWKT-SSP's communication cost is as high as $19900b_0$ bits (nearly 5.8 times as much as $3400b_0 + 3200$ bits) though its capability of resisting collusion is also qualified.

V. Experimental Results

We implement CR-SSP and previous used-extensively secure sum protocols and the experimental results are displayed in this section. All experiments are performed on the Windows XP operating system with Intel Pentium Dual 2.00GHz CPU and 1GB memory.

1. Runtime

As we have analyzed the exact communication bit overheads in Section IV.2 and different network performance will make a great deal of difference in communication time, each participant is simply action as a thread and the exchange data is directly shared in memory. That is, the runtime in the experimental results are only computation time and different participants are in parallel as much as possible.

In the experiment, we set the summation is not more than 3×10^9 and each private input is randomly generated. The key length of homomorphic cryptosystem in SSP_HE was set to 512 bits. The experimental results are shown in Table 2. Again, n is the number of parti-

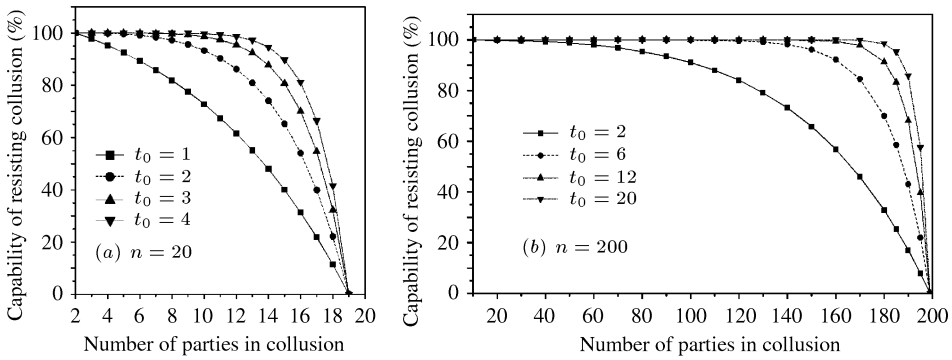


Fig. 1. CR-SSP's capability of resisting collusion when (n, t) are some specialized value

Table 2. Performance of CR-SSP and previous secure sum protocols

Protocol			Runtime (10^{-6}s)	Communication overheads (bit)	Security
Simple-SSP	$n=10$		1	$10b_0$	$(n,2,0)$-collusion resisting
	$n=20$		2	$20b_0$	
	$n=200$		5	$200b_0$	
	$n=1000$		16	$1000b_0$	
SSP_HE	$n=10$		2971023	$10b_0+4608$	$(n,2,1-1/n)$-collusion resisting, $(n,3,1/n)$-collusion resisting
	$n=20$		5631102	$20b_0+9728$	
	$n=200$		53511051	$200b_0+101888$	
	$n=1000$		266312478	$1000b_0+511488$	
UWKT-SSP	$n=10$		4	$45b_0$	$(n,n-2,100\%)$-collusion resisting
	$n=20$		7	$190b_0$	
	$n=200$		33	$19900b_0$	
	$n=1000$		107	$499500b_0$	
CR-SSP	$n=10$	$t=1$	1	$20b_0+10$	$(10,m+2,P_{cr}(10,m))$-collusion resisting
		$t=2$	1	$20b_0+20$	
		$t=3$	2	$20b_0+30$	
	$n=20$	$t=1$	2	$40b_0+20$	$(20,m+2,P_{cr}(20,m))$-collusion resisting
		$t=2$	3	$40b_0+40$	
		$t=3$	3	$40b_0+60$	
		$t=4$	4	$40b_0+80$	
	$n=200$	$t=2$	6	$400b_0+40$	$(200,m+2,P_{cr}(200,m))$-collusion resisting
		$t=6$	8	$400b_0+1200$	
		$t=12$	11	$400b_0+2400$	
		$t=20$	15	$400b_0+4000$	
	$n=1000$	$t=10$	21	$2000b_0+10000$	$(1000,m+2,P_{cr}(1000,m))$-collusion resisting
		$t=30$	33	$2000b_0+30000$	
		$t=50$	41	$2000b_0+50000$	
		$t=100$	63	$2000b_0+100000$	

cipants in secure sum protocol, t is the same as stated in CR-SSP and b_0 is the bit length of a private input.

As can be seen from Table 2, CR-SSP's runtime and communication overheads are close to Simple-SSP, but CR-SSP's capability of resisting collusion verges on UWKT-SSP's. The communication cost of CR-SSP is about $2(t+1)/(n-1)$ of UWKT-SSP's and the superiority of CR-SSP is more distinct when the number of participants n is larger. Because different participants are in parallel as much as possible, the difference between Simple-SSP, CR-SSP and UWKT-SSP is inapparent, but UWKT-SSP still requires more runtime than CR-SSP. In SSP_HE, each private data will be encrypted, as a result, it requires a great much more computation time than others.

2. Security performance

To experimentally illustrate the security of CR-SSP, we simulate the distribution of the cardinality of each party's masking set (see Definition 3 in Section IV.2). As the distribution of the cardinality of masking set is probabilistic, the statistical results of 10000 independent experiments are presented in Fig.2.

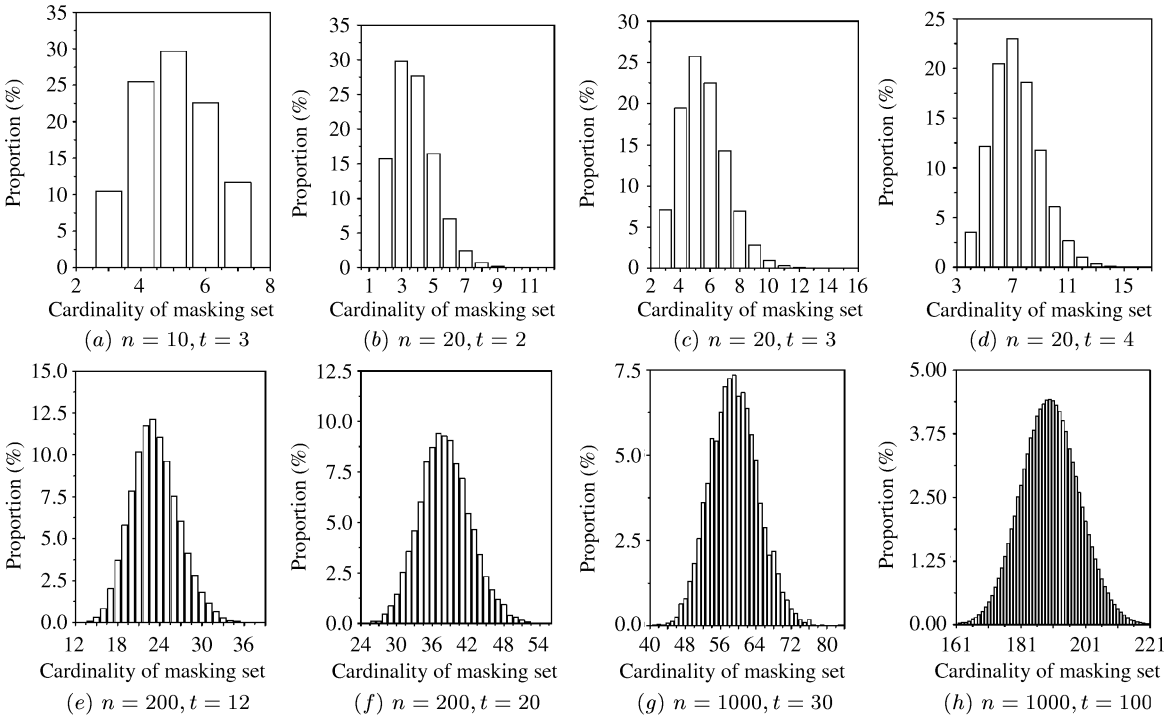


Fig. 2. Probability distribution of the cardinality of masking set

Fig.2 shows that the distribution of cardinality of masking set obeys to the distribution in Eq.(1). Theorem 2 is based on the Eqs.(1) and (2), and the Eq.(2) stands to reason. As a result, Fig.2 experimentally confirms Theorem 2 which has shown that CR-SSP's capability of resisting underlying collusion is quite strong.

As stated above, the communication overheads and computation complexity of CR-SSP both are linearity of the number of participants, simultaneously, its capability of resisting collusion is on the verge of the best possible capability of any secure sum protocol. Therefore, CR-SSP can efficiently and effectively resist potential collusion.

VI. Conclusion

In this paper, we proposed the security definitions to measure the SMC protocols' capability of resisting collusion, analyzed previous secure sum protocols' capability of resisting collusion and then presented a novel secure sum protocol, CR-SSP. Theoretical analysis and experimental result confirm that CR-SSP is efficient and has strong capability of resisting underlying collusion such that it is much superior to previous ones. The communication overheads and computation complexity of CR-SSP both are linearity of the number of participants. Additionally, CR-SSP's capability of resisting collusion is adjustable according to different security needs. For the future work, we will analyze other SMC protocols' capability of resisting collusion and develop some new efficient SMC protocols with practical capability of resisting collusion.

References

[1] A.C. Yao, "Protocols for secure computations", In *23rd Annual IEEE Symposium on Foundations of Computer Science* 1982.

[2] V. Goyal, P. Mohasssel, A. Smith, "Efficient two party and multiparty computation against covert adversaries", *Advances in Cryptology-EUROCRYPT 2008, LNCS*, Vol.4965, pp.289–306, Springer Berlin / Heidelberg, 2008.

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

[4] Y. Lindell, B. Pinkas, "Privacy preserving data mining", *Advances in Cryptology-CRYPTO'00, LNCS*, Vol.1880, pp.36–53, Springer-Verlag, 2000.

[5] C. Clifton, M. Kantarcioglu, X. Lin, J. Vaidya, M. Zhu, "Tools for privacy preserving distributed data mining", *SIGKDD Explorations*, Vol.4, No.2, pp.28–34, 2003.

[6] B. Schneier, *Applied Cryptography* (2nd edition), John Wiley & Sons, 1995.

[7] B. Goethals, S. Laur, H. Lipmaa *et al.*, "On private scalar product computation for privacy-preserving data mining", In *7th Annual International Conference on Information Security and Cryptology*, Seoul, Korea, LNCS, Vol.3506, pp.104–120, Springer-Verlag, 2004.

[8] C. Aggarwal, S. Yu *et al.*, *Privacy-Preserving Data Mining*, Springer US, 2008.

[9] M. Kantarcioglu, C. Clifton, Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data, *IEEE Transactions on Knowledge and Data Engineering*, Vol.16, No.9, pp.1026–1037, 2004.

[10] N. Schlitter, "A protocol for privacy preserving neural network learning on horizontally partitioned data", In *Privacy Statistics in Databases*, Istanbul, Turkey, 2008.

[11] J. Zhan, G. Blosser, C. Yang, L. Singh, "Privacy-preserving collaborative social networks", In *ISI 2008 International Workshops*, Taipei, Taiwan, LNCS, Vol.5075, pp.114–125, Springer-Verlag, 2008.

[12] G. Blosser, J. Zhan, "Privacy-preserving collaborative social network", In *IEEE International Conference on Information Security and Assurance*, Busan, Korea, 2008.

[13] L. Singh, J. Zhan, "Measuring topological in social networks", In *IEEE International Conference on Granular Computing*, Silicon Valley, USA, 2007.

[14] D. Wang, C. Liau, T. Hsu, "Privacy protection in social network data disclosure based on granular computing", In *IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada, 2006.

[15] G. Blosser, J. Zhan, "Privacy-preserving collaborative E-voting", In *ISI 2008 International Workshops*, Taipei, Taiwan, LNCS, Vol.5075, pp.508–513, Springer-Verlag, 2008.

[16] S. Han, W. Ng, "Privacy-preserving linear fisher discriminant analysis", In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Osaka, Japan, LANI, Vol.5012, pp.136–147, Springer-Verlag, 2008.

[17] M. Shaneck, Y. Kim, V. Kumar, "Privacy-preserving nearest neighbor search", University of Minnesota, TR 06-014, 2006.

[18] H. Kargupta, K. Das, K. Liu, "A game theoretic approach toward multi-party privacy-preserving distributed data mining", In *11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Warsaw, Polland, 2007.

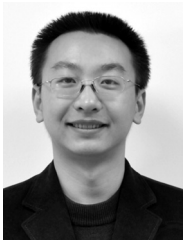
[19] S. Shepard *et al.,* "Data mining and collusion resistance", In *World Congress on Engineering, London*, U.K., 2009.

[20] S. Urabe *et al.,* "A high collusion-resistant approach to distributed privacy-preserving data mining", *IJSP Transactions on Databases*, Vol.48, No.11, pp.104–117, 2007.

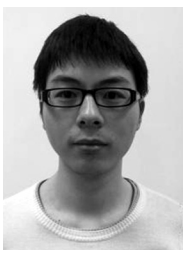
[21] P. Paillier, "Public key cryptosystems based on composite degree residuosity classes", *Advances in Cryptology-EUROCRYPT 1999*, LNCS, Vol.1592, pp.223–238, Springer-Verlag, 1999.

Zhu Youwen was born in 1986, received B.S. degree in 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 of School of Computer Science and Technology at University of Science and Technology of China. His main research interests include information security and wireless senor network. (Email: zhuwy@mail.ustc.edu.cn)

HUANG Liusheng was born in 1957, is currently a professor and Ph.D. supervisor of School of Computer Science and Technology at 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.



YANG Wei was born in 1978, received Ph.D. degree in computer science and technology from University of Science and Technology of China in 2007. At present, he is a post-doctor of School of Computer Science and Technology at University of Science and Technology of China. His main research interests include information security and quantum information.



YUAN Xing was born in 1986, received B.S. degree in information security from Hefei University of Technology in 2009. Currently, he is a master student of School of Computer Science and Technology at University of Science and Technology of China. His research interests include information security and trustworthy computing.