# A GENERAL SECURE SUM PROTOCOL

*Van Chung Nguyen*[1]

https://doi.org/10.56651/lqdtu.jst.v11.n01.362.ict

**Abstract**

Secure Multiparty Computation - SMC is one of two main methods for building Privacy - Preserving Data Mining - PPDM. Among SMC techniques, Secure Sum Protocol - SSP is the most basic one. For SMC solutions in general and SSPs in particular, the problem of optimizing three parameters: accuracy, performance, and privacy still poses many challenges for researchers. However, most of the SSP solutions proposed in the past have poor performance or can not preserve privacy. In this paper, I present a new SSP solution that more effectively balances the two above parameters. Besides, in this paper, I build a general mathematical constraint model between performance and privacy.

**Index terms**

Secure Sum, Secure Multiparty Computation, PPDM, Data Mining.

## 1. Introduction

In cryptography field, SMC protocols are methods allowing parties to jointly compute a function over their private inputs without disclosing these values. In 1995, Goldreich developed the framework of SMC [1]. Since then, a lot of SMC protocols have been investigated. Furthermore, many SMC-based solutions have been proposed for practical applications such as secure auction problem [2], [3] where the auctioneer can find out the winner without revealing the bids, privacy-preserving data mining and machine learning techniques [4], [5] that obtain knowledge and valuable information while private/sensitive information in datasets is still securely kept.

One of the most typical SMC techniques is the SSP that enables parties to compute the sum of their private inputs while each participant does not disclose his/her input. The idea of SSP protocol is quite simple, but it has been widely applied to solve many distributed data problems such as secure electronic voting system [6], privacy-preserving recommendation system [7], privacy-preserving multi-party data analytics [8], privacy-preserving classification [9]. For example, a typical application of SSP protocol is secure electronic voting system where the vote counter can obtain the total voting result without knowing each voter's ballot. Another application of SSP is privacy-preserving Naive Bayes classifiers [9], [10] based on the multiple users' data, in which the miner and data owners need to use a SSP protocol to compute frequencies used to determine necessary probabilistic values.

[1]Vinh Phuc Technical Economic College

Up to now, a lot of SSP protocols have been proposed by researchers, as follows:

- Schneier first introduced SSP in 1995 [11]. Then it was improved and republished in [12]. This method has a computation complexity of $O(n)$, but the secret data that $U_i$ holds will be revealed if $U_{i-1}$ and $U_{i+1}$ collude with each other.
- Zhan et al. proposed SSP_HE based on Homomorphic Encryption (HE) in 2008 [13]. This method is not computationally complicated and has low communication costs, but the secret data that $U_i$ holds will be exposed if $U_{i-1}$ and $U_{i+1}$ collude with $U_n$.
- In 2007, Kargupta et al. published research on SSP based on the game theory approach [14]. This protocol cannot resist collusion but also has complex computation and high communication costs. In the same research direction, En and Yongquan also proposed a CFR-SSP solution in 2013 [15]. It can resist $(n-2)$ collusion but the computation complexity is $O(n^2)$ and the communication costs are high.
- Urabe et al. proposed the SSP-CRDM in 2007 [16]. In this solution, the secret data is divided into $(n-i)$ segments and depends on the $(n-1)$ remaining parties. Therefore, this solution can also resist $(n-2)$ collusion but has $O(n^2)$ computation complexity and high communication costs; the installation is quite difficult because the activities at each party are different. We believe that this solution is only suitable when the number of party $n$ is not too big.
- In 2011, Youwen et al. published a CR-SSP solution that was considered very effective so far [17]. In CR-SSP, each party $U_i$ chooses $t$ random integer values $v_1, v_2, \ldots, v_t$ and sends randomly to the other $t$ parties $(0 < t \ll n-3)$. Each party $U_j$ receives an integer $v_j$ from $U_i$ then chooses a random bit $q_{j\_i}$; if $q_{j\_i} = 0$ then the secret data $m_i = m_i + v_j$ and vice versa. Then, $U_j$ sends $q_{j\_i}$ back to $U_i$. When each party $U_i$ receives a bit $q_{j\_i}$, they will compute the secret data $m_i = m_i - v_j$ if $q_{j\_i} = true$ and vice versa. This solution can resist $t+1$ collusion party and has $O(n)$ computation complexity. However, if $U_i$ colludes to learn the secret data of $U_j$ or vice versa, the random bit selection is meaningless. Besides, each party $U_i$ whether $U_i$ or $q_{i,i}$ need to be identified.

In summary, the proposed SSP solutions either have poor performance or can not guarantee privacy property. Therefore, it is essential to provide a general SSP solution that can balance these two parameters and build a mathematical model that describes the relationship between the privacy and performance of a SSP solution.

In section 2, I will present an effective SSP and evaluate its privacy and performance.

## 2. General secure sum protocol

**Problem definition**: Assume that there are $n$ parties: $U_1, U_2, \ldots, U_n$, each party $U_i$, $(i = \overline{(1,n)})$ holds a secret value $S_i$ $(S_i \in \mathcal{R})$. We need to compute the sum $S = \sum_{i=1}^{n} S_i$, in which each party $U_i$ does not reveal the secret value $S_i$. In other words,

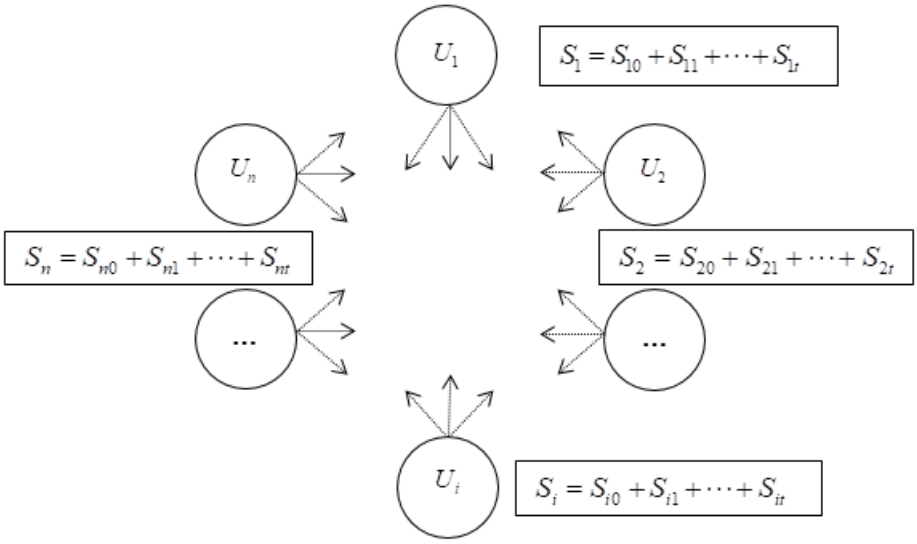
we need an effective secure computational protocol to build the following function:

$$(S_1, S_2, \ldots, S_n) \mapsto S = \sum_{i=1}^{n} S_i$$

Before executing the proposed protocol, it is assumed that the connection between any pair of participants is established by using SSL/TLS that is widely used to make secure communications in current.

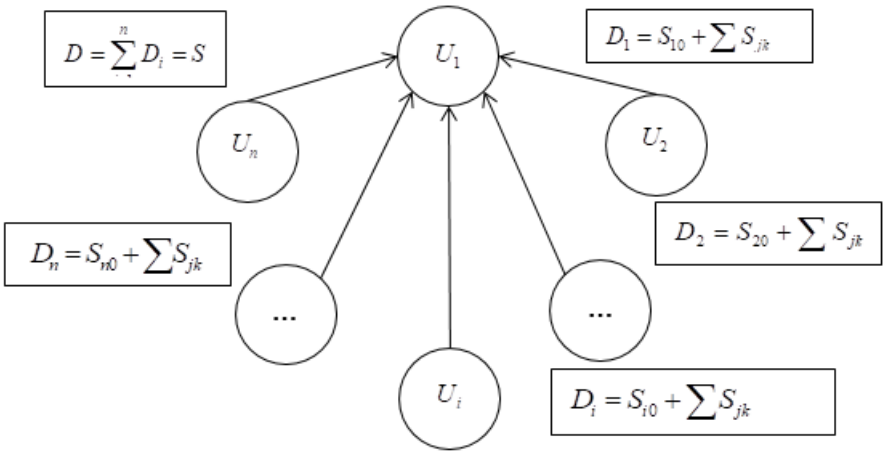
Our general SSP has two main phases as follows:

a. **Phase 1**: Each party $U_i$ $(i = \overline{1,n})$ shares a portion of the secret data $S_i$ for $t$ random members $(1 \leq t < n)$.
   - **Step 1**: Each party $U_i$ splits the secret data $S_i$ into $(t+1)$ secret segments: $S_i = S_{i0} + S_{i1} + \cdots + S_{it}$. Each party $U_i$ privately keeps $S_{i0}$.
   - **Step 2**: Each party $U_i$ randomly chooses $t$ different numbers: $a_{i1}, a_{i2}, \ldots, a_{it}$, $(a_{ij} \in \{1, 2, \ldots, n\} \setminus \{i\}; j = \overline{1,t})$ then sends each $Sij$ to $U_{a_{ij}}$ respectively.
     For $i = 1 \rightarrow n$
     {
     $S_i = S_{i0} + S_{i1} + \cdots + S_{it}$;
     Chooses $a_{i1}, a_{i2}, \ldots, a_{it}$; //$a_{ij} \in \{1, 2, \ldots, n\} \setminus \{i\}; j = \overline{1,t}$
     For $j = 1 \rightarrow t$
     Sends $S_{ij}$ to $U_{a_{ij}}$;
     }




b. **Phase 2**: Computes the secret sum $S$.
   - **Step 1**: Each party $U_i$ $(i = \overline{1,n})$ gets $S_{jk}$ from other parties, then party $U_i$ computes: $D_i = S_{i0} + \sum S_{jk}$. Each party $U_i$ $(i = \overline{2,n})$ sends $D_i$ to $U_1$.

- **Step 2**: $U_1$ computes: $D = \sum_{i=1}^{n} D_i = \sum_{i=1}^{n} S_i = S$.
  For $i = 1 \rightarrow n$
  {
  $D_i = S_{i0} + \Sigma S_{jk}$; // where $S_{jk}$ is the values received by $U_i$
  Sends $D_i$ to $U_1$; }
  For $i = 1 \rightarrow n$
  $S = S + D_i$;




## 3. Evaluation of general secure sum protocol

### *3.1. Accuracy*

The GSSP protocol can compute $S = \sum_{i=1}^{n} S_I$ while the requirements of the problem stated in Section 2 are satisfied. Indeed, when $U_1$ receives the values $D_i$ from parties $U_i$ $(i = \overline{2, n})$, $U_1$ computes:

$$D = \sum_{i=1}^{n} D_i = \sum_{i=1}^{n} S_{i0} + \sum_{j=1}^{n} \sum_{k=1}^{t} S_{jk}$$
$$= \sum_{i=1}^{n} \sum_{j=1}^{t} S_{ij} = \sum_{i=1}^{n} S_i = S.$$

### *3.2. Privacy*

For the privacy of the GSSP, we define the probability that the GSSP resist $(n - k)$ colluding parties is $P(n, n - k)$. To do this, we determine the probability of $(n - k)$ parties collude to know the secret data of one party through the function $C(n, n - k) =$

$1 - P(n, n-k)$. This function represents the relationship between the privacy and performance of the SSP solution, which is constructed as follows:

– It is assumed that the SSP model has $(n-k)$ parties colluding to learn the secret value of one of $k$ remaining parties. The notion of the set of $k$ remaining parties is Set_honest $= \{U_{j\_1}, U_{j\_2}, \ldots, U_{j\_k}\}$; the set of $t$ parties received secret segments from $U_i$ is Set_Receiver($U_i$), and the set of parties sending secret segments to $U_i$ is Set_Sender($U_i$).

– Without the loss of generality, assume that the attacked party is $U_{j\_1}$. Since the secret data $S_{j\_1}$ depends on m parties: $U_1$, $t$ parties of Set_Receiver($U_{j\_1}$), and parties of Set_Sender($U_{j\_1}$) so that $(t+1) \leq m \leq n$. We consider the following cases:

+ **Case 1**: If $k = 1$ then $C(n, n-k) = 1$, that means GSSP can not resist $(n-1)$ colluding parties with any $t$. In fact, this is quite understandable.

+ **Case 2**: If $n - k < t + 1$ then $C(n, n-k) = 0$, that means GSSP is secure against $(n-k)$ colluding parties.

+ **Case 3**: The other cases

To find out the secret data $S_{j\_1}, U_1 \notin$ Set_honest; Set_honest $\cap$ Set_Receiver($U_{j\_1}$) $= \emptyset$ and Set_honest $\cap$ Set_Sender($U_{j\_1}$) $= \emptyset$. Because these three conditions are independent, therefore:
$C(n, n-k) = p(U_1 \notin$Set_honest) . p(Set_honest $\cap$ Set_Receiver($U_{j\_1}$) $= \emptyset$). p(Set_honest $\cap$ Set_Sender($U_{j\_1}$) $= \emptyset$)
We have:

- $p(U_1 \notin$ Set_honest) $= \dfrac{C_{n-1}^{k}}{C_{n}^{k}} = \dfrac{n-k}{n}$
- $p($Set_honest $\cap$ Set_Receiver($U_{j\_1}$) $= \emptyset) = p\left(\cap_{m=2}^{k} (U_{j\_m} \notin \text{Set\_Receiver}(U_{j\_1}))\right)$

$\Rightarrow p($Set_honest $\cap$ Set_Receiver($U_{j\_1}$) $= \emptyset) = \dfrac{C_{n-k}^{t}}{C_{n-1}^{t}} = \dfrac{(n-t-k+1)\ldots(n-t-1)}{(n-k+1)\ldots(n-1)}$

- $p($Set_honest $\cap$ Set_Receiver($U_{j\_1}$) $= \emptyset) = p\left(\cap_{m=2}^{k} (U_{j\_1} \notin \text{Set\_Receiver}(U_{j\_m})\right)$
$= \prod_{m=2}^{k} p(U_{j\_1} \notin \text{Set\_Receiver}(U_{j\_m}))$

$\Rightarrow p($Set_honest $\cap$ Set_Sender($U_{j\_1}$) $= \emptyset) = \left(\dfrac{C_{n-2}^{t}}{C_{n-1}^{t}}\right)^{k-1} = \left(1 - \dfrac{t}{n-1}\right)^{k-1}$

Thus, $C(n, n-k) = \dfrac{n-k}{n} \cdot \dfrac{(n-t-k+1)\ldots(n-t-1)}{(n-k+1)\ldots(n-1)} \cdot \left(1 - \dfrac{t}{n-1}\right)^{k-1}$.

Therefore the probability of this solution when it resists $(n-k)$ colluding parties is:

$$P(n, n-k) = 1 - C(n, n-k)$$
$$= 1 - \frac{n-k}{n} \cdot \frac{(n-t-k+1)\ldots(n-t-1)}{(n-k+1)\ldots(n-1)} \cdot \left(1 - \frac{t}{n-1}\right)^{k-1}$$

### *3.3. Performance*

*3.3.1. Computation complexity:* To determine the computation complexity of the GSSP protocol, we take a look at the two phases of the protocol:

- In phase 1, each party $U_i$ $(i = \overline{1,n})$ sends $t$ times of the secret segments $S_{ij}$ $(j = \overline{1,t})$. Therefore, the computation complexity in this phase is $O(tn)$.
- In phase 2, there are only two $n-$step single rounds the computation complexity in this phase is $O(n)$.

So the computation complexity of GSSP is $O(tn)$.

*3.3.2. Communication cost:* To determine the communication cost of the GSSP protocol, we take a look at the two phases of the protocol:

- In phase 1, each party $U_i$ $(i = \overline{1,n})$ sends $t$ secret segments $S_{ij}$ $(j = \overline{1,t})$ to $t$ randomly selected members. Therefore in this phase, there are $tn$ messages sent.
- In phase 2, each party $U_i$ $(i = \overline{1,n})$ needs to compute the value $D_i$, then all parties (excepting $U_1$) send this value to $U_1$. Thus in this phase, there are $(n-1)$ messages to be sent.

Therefore the communication cost of the GSSP protocol is the need to send $(t+1)n-1$ messages.

### *3.4. Comparison between GSSP and other SSP solutions*

The comparison between GSSP protocol and some typical SSP solutions express in Table 1.

Because both the supposed solution and the CR-SSP solution [17] share the same idea of building a mathematical model that describes the relationship between privacy and performance. Therefore, we compare the collusion resistance of these two solutions by selecting the parameter sets $(n, k, P(n, n-k))$ to determine the number of parties. Experimental results are presented in Table 2. The above results show that the collusion resistance of the GSSP and CR-SSP solutions is similar.

## 4. Conclusions

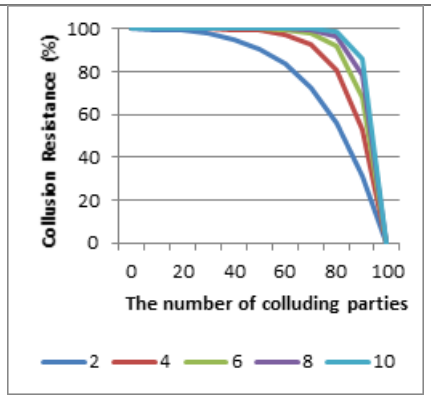
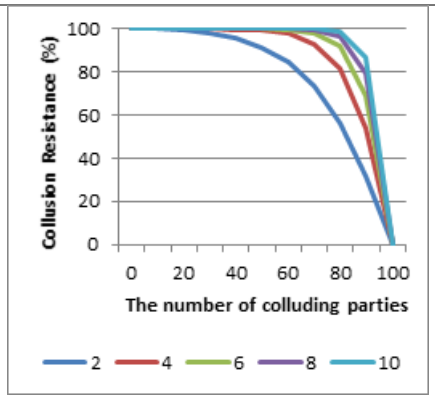
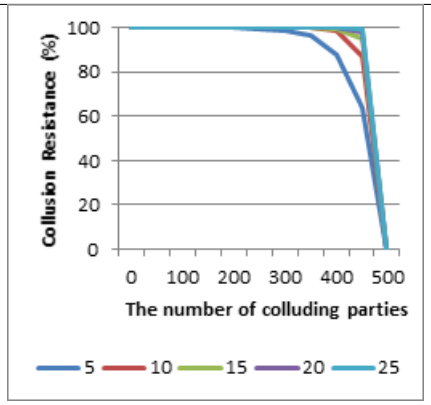
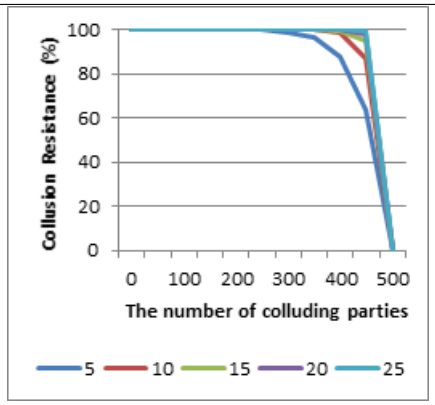
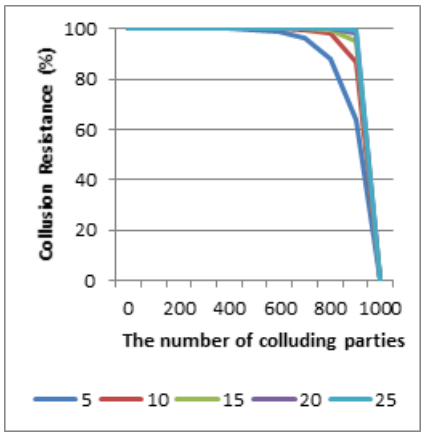
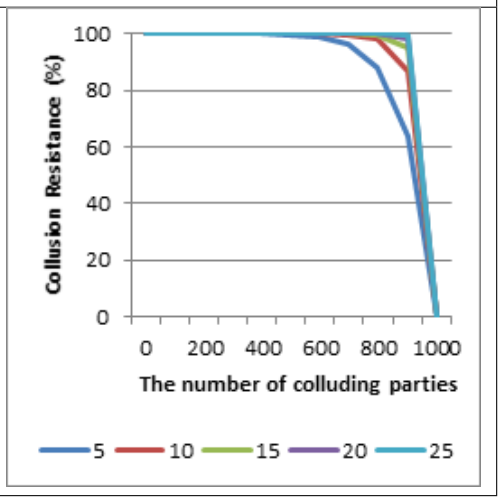
It is nearly impossible to find out an SSP solution that can optimize both parameters, i.e. privacy and performance, so I do not set this goal. In the paper, I have proposed a general SSP solution and built a mathematical model describing the relationship between the privacy and performance of the SSP solution. Based on this model, the parameters can been set to suit the goal and scale of practical problems.

*Table 1. Comparison between GSSP protocol and some typical SSP solutions*

| Protocol | | Privacy Collusion Resistance | Performance | | |
|---|---|---|---|---|---|
| | | | Computation complexity | Communication costs | |
| Basic SSP | | | $O(n)$ | $nM^*$ | $O(nM)$ |
| SSP-HE | | | $O(H \times n)^\dagger$ | $nM + (n-1)M_{key}{}^\ddagger$ | $O(nM)$ |
| SSP-CRDM | | $n-2$ | $O(n^2)$ | $\frac{n(n-1)}{2}M$ | $O(n^2M)$ |
| CFR-SSP | | $n-2$ | $O(n^2)$ | $n(n+2)M$ | $O(n^2M)$ |
| CR-SSP | | $t_0+1$<br>$P(n,m) = 1 - \sum_{k=t_0}^{m}(Pr(k) * p(m,k))$ | $O(n)$ | $(t_0n+n)M + t_0n$ | $O(nM)$ |
| GSSP | $t \ll n$ | $P(n,n-k) = 1 - \frac{n-k}{n} \cdot \frac{C_{n-k}^t}{C_{n-1}^t} \cdot \left(1 - \frac{t}{n-1}\right)^{k-1}$ | $O(n)$ | $(tn+n)M - M$ | $O(nM)$ |
| | $t = n-1$ | $n-2$ | $O(n^2)$ | $n^2M$ | $O(n^2M)$ |

$^*M$ is the message length
$^\dagger$O(H) is the computation complexity of $HE$
$^\ddagger M_{key}$ is the key length of $HE, t_0 \ll n-3$

*Table 2. Experimental results on the collusion resistance of GSSP and CR-SSP*

| $n$SSP | GSSP | CR-SSR |
|---|---|---|
| $n = 100$ |  |  |
| $n = 500$ |  |  |
| $n = 1000$ |  |  |

# References

[1] O. Goldreich, *Foundations of Cryptography*. Cambridge University Press, 2004, vol. 2.

[2] J. Howlader and A. K. Mal, "Sealed−bid auction: a cryptographic solution to bid−rigging attack in the collusive environment," *Secur. Commun.*, vol. 8, p. 3415–3440, May 2015.
[3] T. Jung, X. Y. Li, and M. Wan, "Collusion−tolerable privacy−preserving sum and product calculation without secure channel," *IEEE Trans. Dependable Secure Comput.*, vol. 12, p. 45–57, Jan 2015.
[4] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," *Annual Network and Distributed System Security Symposium*, p. 1–14, 2015.
[5] S. Urabe, J. Wang, E. Kodama, and T. Takata, "A high collusion−resistant approach to distributed privacy−preserving data mining," *IPSJ Digit. Courier*, vol. 3, p. 442–455, 2007.
[6] F. Hao, M. N. Kreeger, B. Randell, D. Clarke, S. F. Shahandashti, and P. H. J. Lee, "Every vote counts: ensuring integrity in large−scale electronic voting," *USENIX J. Election Technol.Syst.*, vol. 2(3), p. 1–25, Jul 2014.
[7] D. Li, C. Chen, Q. Lv, L. Shang, Y. Zhao, T. Lu, and N. Gu, "An algorithm for efficient privacy−preserving item−based collaborative filtering, future gener," *Comput. Syst.*, vol. 55, p. 311–320, 2016.
[8] S. Mehnaz, G. Bellala, and E. Bertino, "A secure sum protocol and its application to privacy−preserving multi−party analytics," in *Proc. of the 22nd ACM on Symposium on Access Control Models and Technologies, ACM*, 2017.
[9] T. Li, J. Li, Z. Liu, P. Li, and C. Jia, "Differentially private naive bayes learning over multiple data sources," *Inf. Sci.*, vol. 444, pp. 89–104, 2018.
[10] Z. Yang, S. Zhong, and R. N. Wright, "Privacy−preserving classification of customer data without loss of accuracy," in *Proc. of the 2005 SIAM Int. Conf. on Data Mining, SIAM*, 2005.
[11] B. Schneier, *Applied Cryptography*. John Wiley & Sons, 1995.
[12] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu, "Tools for privacy preserving distributed data mining," *J. SIGKDD Explorations Newsletter*, vol. 4(2), 2002.
[13] J. Zhan, G. Blosser, C. Yang, and L. Singh, "Privacy−preserving collaborative social networks," in *In ISI 2008 International Workshops, Taipei, Taiwan*, 2008.
[14] H. Kargupta, K. Das, and 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.
[15] Z. En and C. Yongquan, "Collusion−free rational secure sum protocol," *Chinese Journal of Electronics*, vol. 12, pp. 563–566, 2013.
[16] S. Urabe, "A high collusion−resistant approach to distributed privacy−preserving data mining," *IJSP Transactions on Database*, vol. 48, p. 104–117, 2007.
[17] Z. Youwen, H. Liusheng, Y. Wei, and Y. Xing, "Efficient collusion−resisting secure sum protocol," *Chinese Journal of Electronics*, vol. 20, pp. 407–413, Jul 2011.

*Manuscript received 15-1-2022; Accepted 15-4-2022.*

■

**Van Chung Nguyen** received a master's Computer Science, 2013, from Thainguyen of Information and Communication Technology. Currently working at the Department of Information Technology, Vinh Phuc Technical Economic College. Research fields: cryptography, information security. E-mail:nguyenvanchung.vtec@gmail.com

# MỘT GIAO THỨC TÍNH TOÁN TỔNG BẢO MẬT TỔNG QUÁT

*Nguyễn Văn Chung*

**Tóm tắt**

Tính toán bảo mật nhiều thành viên (Secure Multiparty Computation - SMC) là một trong những phương pháp chính để xây dựng các giải pháp đảm bảo tính riêng tư trong quá trình khai phá dữ liệu (Privacy Preserving Data Mining - PPDM). Trong các kỹ thuật của SMC, tổng bảo mật (Secure Sum Protocol - SSP) là kỹ thuật cơ bản nhất. Đối với các giải pháp SMC nói chung và SSP nói riêng, vấn đề tối ưu ba tham số: độ chính xác, hiệu năng và tính riêng tư vẫn luôn đặt ra nhiều thách thức cho các nhà nghiên cứu. Tuy nhiên, hầu hết các giải pháp SSP được đề xuất trước đây có hiệu năng không cao hoặc chưa đảm bảo tính riêng tư. Trong bài báo này, tác giả trình bày một giải pháp SSP mới giúp cân bằng hiệu quả hơn hai tham số kể trên. Bên cạnh đó, trong bài báo này tác giả xây dựng mô hình ràng buộc toán học tổng quát giữa hiệu năng và tính riêng tư.