An efficient secure sum of multi-scalar products protocol base on elliptic curve

Vu Thi Van, Luong The Dzung, Hoang Van Quan, Tran Thi Luong

***Abstract*—The secure scalar product is one of the most important protocols in PPDM. It is used in many problems such as safe auction, secure voting, ensuring privacy for recommendation systems and statistical data analysis, etc... In this paper, we propose an efficient multi-party secure computation protocol using Elliptic curve cryptography, which allows to compute the sum value of multi-scalar products without revealing the input vectors. Moreover, theoretical and experimental analysis shows that the proposed method is more efficient than others in both computation and communication.**

***Tóm tắt*—Giao thức tính tích vô hướng an toàn được áp dụng rộng rãi để giải quyết các vấn đề thực tế như khai phá dữ liệu có đảm bảo tính riêng tư, đấu giá an toàn, bỏ phiếu điện tử an toàn, hệ gợi ý có đảm bảo tính riêng tư,… Một giao thức tính toán bảo mật nhiều thành viên cải tiến sẽ được đề xuất sử dụng hệ mật mã trên đường cong Elliptic cho phép tính tổng giá trị của các tích vô hướng trong khi không tiết lộ gì về các vectơ đầu vào. Hơn nữa, phân tích lý thuyết và thực nghiệm cho thấy phương pháp đề xuất có hiệu quả cả về tính toán và truyền thông so với các phương pháp khác.**

Keywords— Scalar Product Protocol, Secure Multi-party Computation, Secure Sum Protocol, Elliptic curve cryptosystem, Privacy preserving data mining, Privacy preserving frequency mining.

Từ khóa— Khai phá dữ liệu có đảm bảo tính riêng tư, Tính toán bảo mật nhiều thành viên, Giao thức tính tổng an toàn, Đường cong Elliptic, Tính tổng các tích vô hướng an toàn.

I. INTRODUCTION

Secure Multi-party Computing (SMC for short) evolved from a curious theory in the 1980s into a tool for building practical systems today. Over the past decade, SMC has been one of the most active research areas in theoretical and applied cryptography [1]. SMC allows participants to perform the same calculation without disclosing any participant's private inputs. SMC protocols can be either semi-honest or malicious [2]. SMC protocols are based on a semi-honest model that assumes that the parties follow the protocol rules, but they still try to find out the private information of others by analyzing all the messages which are exchanged. In the malicious model, the parties can perform some arbitrary operations to cause damage to other participants while implementing the protocol. Therefore, the semi-honest model is used more regularly than the malicious model. Effective and practical SMC protocols are a hot topic of interest to many researchers and have yielded many results. The SMC protocol has the following important properties:

- **Correctness:** The outputs of an SMC protocol must be the desired values.
- **Privacy:** The private inputs of each honest party need to be securely protected, even if there are some participants colluding together.
- **Efficiency:** The SMC protocol must be efficient to be applicable in large-scale distributed scenarios.

In this work, we propose an efficient secure sum of multi-scalar product protocol (SSMSP). The problem can be solved as follows: Assume that that there are n users in which each person U_i keeps a secret vector $\vec{u}_i$, and a special party (denoted as a miner) owns his private vector $\vec{v}$. The miner desires to obtain the sum of scalar products $\vec{v}.\vec{u}_i^T$ with $i = \overline{1,n}$. The secure scalar product protocol has been widely applied in many fields such as privacy-preserving data mining [3, 4, 5, 6], privacy-preserving statistical analysis [14, 15], privacy-preserving computational geometry [7, 8], cloud computing [9, 10].

This manuscript is received on October 29, 2021. It is commented on October 29, 2021 and accepted on November 10, 2021 by the first reviewer. It is commented on October 29, 2021 and accepted on November 15, 2021 by the second reviewer.

Although our problem can be solved by using the available SMC protocols such as secure scalar product [3, 6, 9, 4, 11], secure multi-party sum [12, 5, 13], and privacy-preserving frequency mining in 2PFD setting (2-Party Fully Distributed setting) [14, 15] protocols, but these cryptographic tools either do not protect the privacy of each party or reduce the solution's performance significantly. The SSMSP protocol in [16] can solve this problem without communication channels between different users. Moreover, this protocol also provides strong privacy for each user without loss of accuracy. However, this protocol is based on the ElGamal cryptosystem with low performance. Therefore, the main contribution of this paper is to develop an efficient protocol for the secure sum of multi-scalar product computation by optimizing the protocol in [16] with the use of Elliptic curve cryptography. The proposed protocol inherits the advantages of the original protocol such as:

- The proposed protocol has the capacity to ensure the correctness of the output result while privately protecting the parties' inputs and outputs.
- The proposed protocol only requires one flow of communication between the miner with each data user. This advantage especially makes the protocol appropriate in web applications as the data providers only need to submit and finish their task.
- The proposed protocol does not require any communication between each tuple of the data providers. As a result, it is suitable for multi-party computation models.

The experimental results and performance analysis find that the proposed protocol has better performance than others [16].

To illustrate the effectiveness of our solution, we implemented protocols to calculate the sum of dot products for different numbers of users from 10,000 to 50,000 and different numbers of vector dimensions of each user from 10 to 50.

The rest of the paper is organized as follows. Section 2 reviews some technical preliminaries used in this work. Our protocol is described in Section 3. Finally, Section 4 discusses the conclusion of the paper.

II. PRELIMINARIES

A. Shanks' baby-step giant-step algorithm

In this section, we present Shanks' baby-step giant-step algorithm [17] that is used to solve the discrete logarithm problems on Elliptic curve.

Input: An Elliptic curve $E(F_p)$ with base point G has order q, and a point Q.

Output: Value x satisfies $x.G = Q$.

$m \leftarrow [\sqrt{q}] + 1$

forall j *where* $0 \leq j < m$ ***do***

  Compute $j.G$ and store the pair $(j, j.G)$ in a hash table

Compute $-m.G$

$\beta \leftarrow Q$

forall i *where* $0 \leq i < m$ ***do***

  if β *is the second component* $(j.G)$ *of any pair in the hash table* **then**

    return $x = i.m + j$

  else $\beta \leftarrow \beta - m.G$

Fig. 1. Shanks' baby-step giant-step algorithm.

B. Elliptic curve cryptography

In this section, we review elliptic curve analog of the ElGamal system [18] that is the main fundamental to construct our solution.

Let E(F_p) be an Elliptic curve over a finite field F_p with a point O at infinity and p be a large prime, in which elliptic curve discrete logarithm problem is hard. In addition, G is a base point of the elliptic curve E with order q (i.e., $q.G = O$). The private key is the random number $d \in [1, q-1]$, and the corresponding public key curve point is $Q = d.G$. To encrypt the plaintext m, the sender uses the public key Q to compute the ciphertext C from the plaintext m as follows: he randomly chooses k from $[1, q-1]$ and computes the ciphertext $C = (C_1 = P_m + k.Q, C_2 = k.G)$ where P_m is a point of E with

$x_{P_m} = m$. To decrypt the ciphertext C using the private key d, the receiver may compute $m = x_M$, in which $M = C_1 + (-d.C_2)$.

Under the decisional Diffie-Hellman assumption for the curve E, elliptic curve analog of the ElGamal system is semantically secure.

III. SSMSP PROTOCOL BASE ON ELLIPTIC CURVE

A. Problem statement

In this paper, it is assumed that there are n users $\{U_1, \dots, U_n\}$ in which each user U_i owns his private k-dimensions vector $\overrightarrow{u^{(i)}} = (u_1^{(i)}, \dots, u_k^{(i)})$ and a special party U_0 (called the miner) who keeps a secrete k-dimensions vector $\overrightarrow{u^{(0)}} = \vec{v} = (v_1, \dots, v_k)$ with $u_j^{(i)}, v_j \in \{0, 1\}$. While executing a data analytic process on n users' data, the miner needs to compute the value $S = \sum_{i=1}^{n} \vec{v}.\overrightarrow{u^{(i)}}^T$ without knowing each data user's vector. Simultaneously, he also reveals nothing about his private vector and the output value S.

B. Definition of privacy

The SSMSP protocol is designed to follow a semi-honest model in which each user must abide by the protocol's rules. Therefore, all parties involved are only semi-honest and anyone can become corrupted. So we have the following security definition:

Definition 1 [16]**:** Assume that each party U_i keeps a private vector $\overrightarrow{u^{(i)}}$, and he has the set of private keys $PriK_i$ with the corresponding set of public keys $PubK_i$. An SSMSP protocol protects each user's privacy against t corrupted parties in the semi-honest model if, for all $I \subseteq \{0,1,2,\dots,n\}$ such that $||I|| = t$, there exists a probabilistic polynomial-time algorithm M such that:

$$\left\{M\left(S, \left[\overrightarrow{u^{(i)}}, PriK_i\right]_{i\in I}, \left[PubK_j\right]_{j\notin I}\right)\right\} \stackrel{c}{\equiv} \left\{view_{\{P_i\}_{i\in I}}\left(\left[\overrightarrow{u^{(i)}}, PriK_i\right]_{i=1}^{n}\right)\right\} \quad (1)$$

where $\stackrel{c}{\equiv}$ is computational indistinguishability.

C. Protocol

Before the protocol starts, the participants prepare the parameters to enhance the efficiency of the proposed protocol as follows:

- Let $E(Z_d)$ be an elliptic curve with a point O at infinity and d be a large prime, in which elliptic curve discrete logarithm problem is hard. In addition, G is a base point of the elliptic curve E with order d (i.e., $d.G = O$).
- The miner keeps the private m and corresponding public key $M = m.G$.
- Each user U_i has already owned private keys $x_i, y_i \in [1, d-1]$, and the corresponding public keys is $X_i = x_i.G, Y_i = y_i.G$. These public keys are sent to the miner before the protocol starts.

Then because of the efficiency, the miner pre-computes:

$$X = \sum_{i=1}^{n} X_i = x.G$$

$$Y = \sum_{i=1}^{n} Y_i = y.G$$

Where: $x = \sum_{i=1}^{n} x_i$, $y = \sum_{i=1}^{n} y_i$.

Subsequently, the miner sends the message $(M||X||Y)$ to all users.

Our proposed protocol has three main stages as summarized in Fig. 2.

The miner	**User U_i**
Phase 1:	
• Chooses k random numbers $t_j \in [1, \dots, d-1]$.	
• Computes: $C_1^{(j)} = v_j.G + t_j.M$; $C_2^{(j)} = t_j.G$.	
$\xrightarrow{C_1^{(i)}\|\|C_2^{(i)}}$	

Phase 2:

- Chooses a random number $r_i \in [1, \dots, d-1]$.
- Computes:

$$R_1^{(i)} = \sum_{j=1}^{k} \Big(u_j^{(i)} C_1^{(j)} + r_i.M + y_i.X - x_i.Y \Big)$$

$$R_2^{(i)} = \sum_{j=1}^{k} (u_j^{(i)} C_2^{(j)} + r_i.G)$$

$R_1^{(i)} || R_2^{(i)}$ ⟵

Phase 3:

- Computes:

$$K = \sum_{i=1}^{n} R_1^{(i)} - m. \sum_{i=1}^{n} R_2^{(i)}$$

- Executes Shanks' algorithm to find out S that satisfies $S.G = K$.

Fig. 2. An efficient and secure protocol for computing the dot product value.

D. Proof of correctness

In this section, to show the correctness of the proposed protocol, we prove the following theorem:

Theorem 1. *The protocol for secure n-parties sum presented in Fig. 2 can exactly compute the sum value* $S = \sum_{i=1}^{n} \vec{v}.\overrightarrow{u^{(i)}}^T$.

Proof. Indeed,

$$\begin{aligned} S.G &= K \\ &= \sum_{i=1}^{n} R_1^{(i)} - m. \sum_{i=1}^{n} R_2^{(i)} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} (u_j^{(i)} C_1^{(j)} + r_i.M + y_i.X - x_i.Y) \\ &\quad - m. \sum_{i=1}^{n} \sum_{j=1}^{k} (u_j^{(i)} C_2^{(j)} + r_i.G) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} (u_j^{(i)}.v_j.G + u_j^{(i)}.t_j.M + r_i.M \\ &\quad + y_i.X - x_i.Y - m.(u_j^{(i)}.t_j.G \\ &\quad + r_i.G) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} (u_j^{(i)}.v_j.G + u_j^{(i)}.t_j.m.G \\ &\quad + r_i.m.G - m.u_j^{(i)}.t_j.G \\ &\quad - m.r_i.G + (y_i.X - x_i.Y)) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} \big(u_j^{(i)}.v_j.G \big) \\ &\quad + \sum_{j=1}^{k} (Y.X - X.Y) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} \big(u_j^{(i)}.v_j.G \big) \\ &= \sum_{i=1}^{n} \vec{v}.\overrightarrow{u^{(i)}}^T.G \end{aligned}$$

Thus, $S = \sum_{i=1}^{n} \vec{v}.\overrightarrow{u^{(i)}}^T$.

Note that, if the vectors $\left(\overrightarrow{u^{(0)}}, \overrightarrow{u^{(1)}}, \dots, \overrightarrow{u^{(n)}}\right) \notin \{0,1\}^k$, our proposed protocol still properly executes as long as the range of S is not too large and it can be limited.

E. Proof of Privacy

In this section, we show that the proposed protocol securely protects each honest user's privacy in the semi-honest model under the necessary assumptions.

In this protocol, the miner sends an ElGamal encryption on Elliptic curve $(v_j.G + t_j.M, t_j.G)$ to all users. Thus, our protocol securely preserves the miner's privacy in the semi-honest model. Similarly, because of the security of ElGamal

encryption, an adversary can not distinguish the values $(R_1^{(i)}, R_2^{(i)})$ of each user. Hence, U_i's inputs (i.e. $u_1^{(i)}, \dots, u_k^{(i)}$) is hidden from the adversary.

Next, we prove that the protocol still preserves the privacy of the honest users in the case of some participants colluding as long as the ElGamal encryption scheme is secure. We have the following theorem:

Theorem 2. *The protocol presented in Fig. 2 preserves the privacy of the honest users against the miner and up to n - 2 corrupted users or preserves the privacy of the miner against the collusion of n corrupted users.*

Proof. To obtain complete proof for Theorem 2, we need to show a simulator $\mathcal{M}$ that simulates what the corrupted participants have observed during the protocol execution by a poly-nominal time algorithm. Particularly, we need to give an algorithm that computes the joint view of the corrupted parties in polynomial time using only the corrupted parties' knowledge, the public keys, and some ElGamal encryptions on the Elliptic curve. We considered the two following cases:

The case of the miner colluding with $(n-2)$ corrupted users against two honest users: without loss of generality, it is assumed that U_1 and U_2 do not collude and $I = \{3, 4, \dots, n\}$. We describe the algorithm of simulator $\mathcal{M}$ computes$(R_1^{(1)}, R_2^{(1)})$ and $(R_1^{(2)}, R_2^{(2)})$ as follows:

- $\mathcal{M}$ takes the following encryptions as its input:

$(\mathrm{a}_1, \mathrm{a'}_1) = (\lambda + (\mathrm{x}_2 + \mathrm{y}_1).G, \mathrm{x}_2.G)$; $(\mathrm{a}_2, \mathrm{a'}_2) = ((\mathrm{x}_1 + \mathrm{y}_2).G,\ \mathrm{x}_1.G)$

$(\mathrm{a}_3, \mathrm{a'}_3) = (\theta + (\mathrm{x}_1 + \mathrm{y}_2).G, \mathrm{x}_1.G)$; $(\mathrm{a}_4, \mathrm{a'}_4) = ((\mathrm{x}_2 + \mathrm{y}_1).G,\ \mathrm{x}_2.G)$

Where $\lambda = \sum_{j=1}^{k} u_j^{(1)}.C_1^{(j)} + r_1.M$; $\theta = \sum_{i=1}^{k} u_i^{(2)}.C_2^{(j)} + r_2.M$, and it computes the values:

$$R_1^{(1)'} = a_1 + \sum_{j\in I} x_j\,.\,Y_1 - a_2 - \sum_{j\in I} y_j\,.X_1 + \delta.G \quad (2)$$

$$R_1^{(2)'} = a_3 + \sum_{j\in I} x_j\,.\,Y_2 - a_4 - \sum_{j\in I} y_j\,.X_2 + \delta.G \quad (3)$$

in which, $\delta = S - \sum_{l=3}^{n} \vec{v}.\overrightarrow{u^{(l)}}^{T} - \vec{v}.\vec{\alpha}^{T} - \vec{v}.\vec{\beta}^{T}$ and $\vec{\alpha}, \vec{\beta} \in \{0,1\}^k$.

Next, $\mathcal{M}$ simulates $R_2^{(1)'}, R_2^{(2)'}$ using random ElGamal ciphertexts on the Elliptic curve.

The case of the collusion of n corrupted users against the miner: because the miner does not publish the value S, the algorithm $\mathcal{M}$ only needs to simulate $(C_1^{(j)'},\ C_2^{(j)'})$ using random ElGamal ciphertexts on the Elliptic curve.

Thus, according to Definition 1, our protocol is semantically secure and has the same level of security as the original protocol.

F. Performance evaluation

In this section, we implemented our solution and the original protocol [16] in the C# language of Visual Studio 2019 environment, using the System.Numerics namespace to compare the performance of them (i.e., communication overhead and time complexity). Note that all public key operations in our protocol are defined over the safe curve 25519 [19] with 256 bits keys and the protocol [16] uses 256 bits private keys and 3072 bits public keys that have the same security level with the curve 25519. Moreover, our experiments run on the laptop with a 2.60 GHz Intel core i5 processor and 8GB memory.

For the communication overhead comparison, we considered the number of communication messages and these lengths (bits) in all phases of both our solution and the protocol [16].

For the time complexity comparison, we measure the total executing time of each protocol for different numbers of users, from 10.000 to 50.000 with different numbers of vector dimensions from 30 to 50. This time consists of the time for each user to perform the necessary computations and the time required for the miner. We assume that all users perform their tasks at the same time, and the network latency is not included in the total executing time.

1. Communication Overhead

Considering the protocol in [16], before this protocol starts, each user needs to send two public keys $(X_i||Y_i)$ to the miner. After the miner

computes the two public keys, it sends these two public keys and the miner's public key $(M||X||Y)$ to all users. In the first phase of [16], the miner needs to send $2k$ values $(C_1^{(j)}||C_2^{(j)})(j \in [1,k])$ to the user U_i. In the second phase, each user U_i sends two values $(R_1^{(i)}||\ R_2^{(i)})$ to the miner. Since each public key is 3072 bits in length, protocol [16] exchanges $4n$ messages of size $(21504 + 6144\text{k})n$ bits where n is the number of users.

Similar to the original protocol, the proposed protocol also exchanges $4n$ messages, however, each part of the message is a point of the curve consisting of two elements, each element is 256 bits in length, so the solution is recommended to exchange $(3584 + 512\text{k})n$ bits. As a comparison of the communication costs between the proposed protocol and the original protocol in Table 1, it can be seen that the proposed protocol exchanges the same number of messages as the original protocol, but the number of bits transmitted is much lower than that of the original protocol.

TABLE I. THE COMMUNICATION OVERHEAD COMPARISON BETWEEN OUR SOLUTION AND THE ORIGINAL PROTOCOL

Protocols	**The number of messages**	**The number of bits**
The original protocol	$4n$	$(21504 + 6144\text{k})n$
Our protocol	$4n$	$(3584 + 512\text{k})n$

(Note: n is the number of users)

2. Time complexity of the protocol

To measure the running time of the proposed protocol, we run 25 times for different numbers of users from 10,000 to 50,000 and the number of dimensions of private vectors $k = 10, 20, 30, 40, 50$, then we calculate the average running time. Note that the delay time is not included in our evaluation.

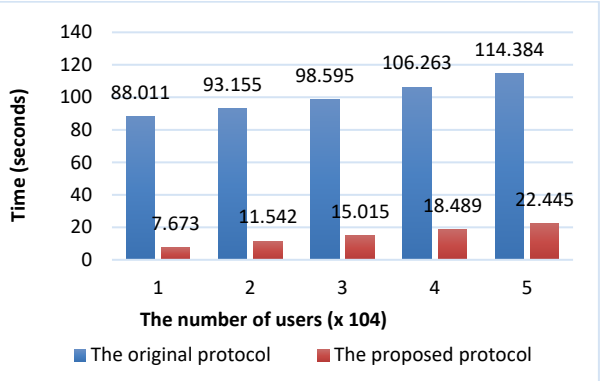

Fig. 3. The computation time of protocols with vector $k = 50$ dimensionality.

Fig. 3 shows the execution times of the two protocols in the cases $n = 10.000, 20.000, 30.000, 40.000, 50.000$ and $k = 50$. The results show that the new protocol is much more efficient than the original, especially as the number of users increases.

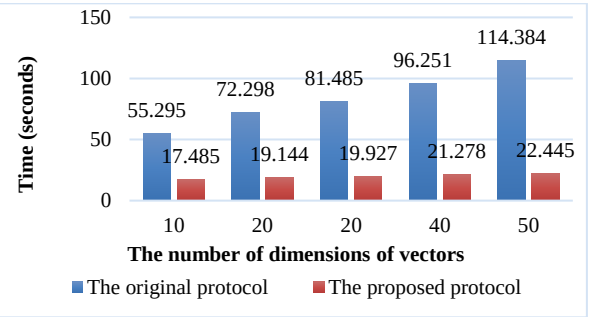

Fig. 4. The calculation time of protocols with the number of users $n = 50000$.

Next, we present the executing time of the protocols in the cases of k = 10,20,30,40,50, and n = 50,000 in Fig. 4. The results show that the proposed protocol is much more efficient than the original protocol, especially when the number of dimensions of the vector increases, for example in the case of n = 50,000 and k = 50, our protocol only runs for about 22 seconds while the original protocol runs for about 114 seconds.

In summary, considering the above experimental results, it can be seen that the proposed protocol is more efficient than the one in [16].

CONCLUSION

In this work, we have proposed an efficient sum of multi-scalar products protocol. The protocol is able to protect the privacy of the parties involved, ensuring the correctness of the calculation results. We also carry on some experiments in order to evaluate the new solution's performance. The experimental and theoretical results indicate that the proposed protocol outweighs the original one. Therefore, it is possible to integrate this protocol into practical applications.

REFERENCES

[1] David Evans, Vladimir Kolesnikov and Mike Rosulek, A Pragmatic Introduction to Secure Multi-Party Computation, NOW Publishers, 2018.

[2] O. Goldreich., The Foundations of Cryptography, volume 2, Cambridge University Press, 2004.

[3] Dong, Changyu and Chen, Liqun, "A Fast Secure Dot Product Protocol with Application to Privacy Preserving Association Rule Mining," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2014.

[4] Y. Zhu,T. Takagi, "Efficient scalar product protocol and its privacypreserving application," in *International J. of Electronic Security and Digital Forensics, Vol. 7, No. 1*, 2015.

[5] Clifton, Chris and Kantarcioglu, Murat and Vaidya, Jaideep and Lin, Xiaodong and Zhu, Michael Y, "Tools for privacy preserving distributed data mining," *ACM Sigkdd Explorations Newsletter*, vol. 4, no. 2, pp. 28-34, 2002.

[6] Goethals, Bart and Laur, Sven and Lipmaa, Helger and Mielikäinen, Taneli, "On private scalar product computation for privacy-preserving data mining," in *International Conference on Information Security and Cryptology*, 2004.

[7] Atallah M J, Du W, "Secure multiparty computational geometry," in *Proc. the 7th International Workshop on Algorithms*, 2011.

[8] Yang B, Sun A D, Zhang W Z, "Secure two-party protocols on planar circles," *Journal of Information & Computational*, vol. 8, no. 1, pp. 29-40, 2011.

[9] Babak Siabi, Mehdi Berenjkoub, Willy Susilo, "Optimally Efficient Secure Scalar Product With Applications in Cloud Computing," *IEEE Access*, vol. 7, no. 1, pp. 42798 - 42815, 29 3 2019.

[10] Zhang J, Li L, Tang Y, Luo S, Yang Y, Xin Y, "Secure two-party computation of solid triangle area and tetrahedral volume based on cloud platform," *PLoS One, Public Library of Science*, vol. 14, no. 1, pp. 1-22, 2019.

[11] Y. Rahulamathavan, S. Dogan, X. Shi, R. Lu, M. Rajarajan and A. Kondoz, "Scalar Product Lattice Computation for Efficient Privacy-preserving Systems," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1417-1427, 2021.

[12] Mehnaz, Shagufta and Bellala, Gowtham and Bertino, Elisa, "A secure sum protocol and its application to privacy-preserving multi-party analytics," in *Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies*, 2017.

[13] Duy-Hien, Vu and The-Dung, Luong and Tu-Bao, Ho, "An Efficient Approach for Secure Multi-party Computation without Authenticated Channel," *Information Science*, 2019.

[14] The-Dung, Luong and Tu-Bao, Ho, "Privacy preserving frequency mining in 2-part fully distributed setting," *IEICE TRANSACTIONS on Information and Systems*, vol. 93, no. 10, pp. 2702-2708, 2010.

[15] Thi Van Vu, The Dung Luong, Van Quan Hoang, "An Elliptic Curve-based Protocol for Privacy Preserving Frequency Computation in 2-Part Fully Distributed Setting," in *12th International Conference on Knowledge and Systems Engineering (KSE)*, Can Tho, 2020.

[16] Luong The Dung, Vu Duy Hien, Vu Thi Van, "A new Approach for secure multi-party computation protocol," *Journal of Military Science and Technology*, vol. Information security special, 2019.

[17] Steven D. Galbraith, Ping Wang and Fangguo Zhang, "Computing Elliptic Curve Discrete Logarithms with Improved," Cryptology ePrint Archive, 2015.

[18] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of computation*, vol. 48, no. 177, pp. 203-209, 1987.

[19] D. J. Bernstein, "Curve25519: new diffie-hellman speed records," in *International Workshop on Public Key Cryptography*, Springer, 2006.

ABOUT THE AUTHORS

Vu Thi Van

Workplace: Academy of Cryptography Techniques

Email: vanvu10101986@gmail.com

Education: She received her Bachelor of Engineering degree from the Academy of Cryptography Techniques in 2009, Master's degree in Information security from Academy of Cryptography Techniques, in 2016. She is currently a PhD candidate of Information security, Academy of Cryptography Techniques.

Recent research direction: secure multi-party computation, data mining and cyber security, etc.

Luong The Dzung

Workplace: Academy of Cryptography Techniques

Email: thedungluong1@gmail.com

Education: He received Bachelor of Engineering in Information Technology from Le Quy Don Technical University in 2001 and PhD degree in 2012 from Institute of Military Science and Technology.

Recent research direction: privacy preserving data mining and computer security.

Hoang Van Quan

Workplace: Staff General, Ministry of Defense

Email: hoangvanquan@gmail.com

Education: He received Engineer degree in Cryptographic Technique from The Academy of Cryptographic Techniques in 1994; Master's degree in Information and Electronic Engineering from Military Institute of Science and Technology in 2005; PhD degree from Electronic Engineering at Military Institute of Science and Technology in 2016.

Recent research direction: Cryptography.

Tran Thi Luong

Workplace: Academy of Cryptography Techniques

Email: luongtranhong@gmail.com

Education: She received Bachelor degree in Mathematics and Informatics of Ha Noi university of Science in 2006; Master degree in cryptographic technique at Academy of Cryptographic Techniques in 2012; PhD degree in cryptographic technique at Academy of Cryptographic Techniques in 2019.

Recent research direction: Cryptography and database security.