A secure Multi-Frequency Computation Protocol in 2-Part Fully Distributed Setting

DOI: https://doi.org/10.54654/isj.v2i22.1051

Vu Thi Van*, Luong The Dung, Luong Ngoc Duong

***Abstract*— Secure frequency computation protocols have been researched and applied to address various privacy-preserving problems in machine learning, data mining, and data analysis. However, these protocols only allow for the calculation of a single frequency value in one execution. In this paper, we propose a protocol for calculating multiple secure frequency values concurrently for the 2PFD data model, which is a relatively new distributed data model that has appeared in many practical problems but has not received much attention. The proposed protocol not only maintains the accuracy of the output results and a high level of security but also exhibits good performance.**

***Tóm tắt*— Giao thức tính tần suất bảo mật đã được nghiên cứu và ứng dụng để giải quyết nhiều bài toán đảm bảo tính riêng tư cho học máy, khai phá dữ liệu, phân tích dữ liệu. Tuy nhiên các giao thức này chỉ cho phép tính một giá trị tần suất trong một lần thực thi. Trong bài báo này, chúng tôi đề xuất giao thức tính đồng thời nhiều giá trị tần suất bảo mật cho mô hình dữ liệu 2PFD, đây là một mô hình dữ liệu phân tán còn khá mới, xuất hiện trong nhiều bài toán thực tế và chưa được nhiều sự quan tâm. Giao thức đề xuất không những duy trì được độ chính xác của kết quả đầu ra, mức an toàn cao mà còn có hiệu suất tốt.**

***Keywords*— Secure frequency computation; Privacy-Preserving; Homomorphic encryption; Secure computation; Elliptic curve cryptography.**

***Từ khóa*— Tính tần suất bảo mật; Đảm bảo tính riêng tư; Mã hoá đồng cấu; Tính toán bảo mật; Hệ mật đường cong Elliptic.**

I. INTRODUCTION

Over the past few decades, Over the last two decades, privacy and data security have become major priorities for both individuals and enterprises. Among these concerns, the problem of secure frequency computation plays a crucial role in privacy-preserving data mining, privacy-preserving data analysis solutions, and other computations. Current privacy-preserving methods include randomization, anonymization (k-anonymity, l-diversity, t-closeness), partitioning-based methods, and encryption [1]. Among these, cryptography-based privacy-preserving methods often give very high levels of data privacy due to their powerful primitives. This approach ensures the privacy of participants' input data and the correctness of the output, even if certain participants are colluded by adversaries [1].

Moreover, the centralized storage of user data in any form raises concerns about data privacy. Therefore, privacy-preserving computational solutions for distributed data models are essential and are receiving significant attention from researchers. The data used in the analysis process can be stored in several distributed models, including horizontal partitioning, vertical partitioning, fully distributed, two-party fully distributed (2PFD), etc. The 2PFD model is widely used in practice, and secure frequency computation is critical for many similar distributed data scenarios [2]. In practice, this issue is quite prevalent, for example, in privacy-preserving Naive Bayes classification for 2PFD [3], [4], or in vertically distributed data models where a central computation hub collaborates with participants to securely compute multiple frequency values necessary for calculating the required probabilities.

The paper was reviewed, accepted and introduced by the XXVII National Conference "Some Selected Issues on Information and Communication Technology" to be published in the Journal on September 5, 2024. It was reviewed on September 20, 2024, revised on September 27, 2024 and accepted on September 29, 2024.

* Corresponding author.

Additionally, there are many other practical privacy-preserving problems, such as decision tree mining [5], association rules [4], clustering, correlation analysis, etc., that ensure data privacy. However, existing SMC-based solutions for frequency estimation that ensure privacy require significant computational and communication costs. Therefore, in order to expand practical applications, there is a need for solutions that maintain accuracy, security, and further improve efficiency.

This paper proposes a secure computation protocol that allows for the efficient simultaneous computation of secure frequency values in the 2PFD model (Secure Multi-Frequency Computation, abbreviated as SMFC). This protocol enables the calculation of the frequency of value pairs in a 2PFD dataset while ensuring the privacy of each component frequency value held by the users. Although this problem can be solved using existing SMC protocols by repeatedly combining a privacy-preserving inner product computation protocol [6]–[8] with a privacy-preserving summation protocol [9]–[11], this approach incurs high computational and communication costs. Furthermore, combining different protocols multiple times complicates the design and may introduce privacy issues during the integration process. Another way to perform this computation is by repeatedly executing the multi-party privacy-preserving inner product protocol in a semi-fully distributed data model [12]. However, this method requires multiple rounds of interaction between the participants and the central computation hub, and it also offers lower security compared to existing 2PFD frequency computation protocols [2], [4], [13], [14].

A simpler approach to calculating these frequency values is to execute the privacy-preserving frequency computation protocol [2], [4], [13], [14] multiple times in the 2PFD model. These protocols do not require communication between users and ensure strong privacy for each user without compromising accuracy. However, these protocols compute only one frequency value per execution, reducing the efficiency of tasks that require multiple frequency values and necessitating repeated rounds of interaction between participants and the computation centre. This makes the solution less feasible. Therefore, the main contribution of this paper is to develop an efficient protocol that allows for the simultaneous computation of multiple frequency values in the 2PFD model in a single execution. The proposed protocol inherits the advantages of typical protocols, such as ensuring the correctness of the output while protecting the input and output privacy of the participants, without requiring any communication between individual datasets among data providers. Thus, it is appropriate for multi-party computation models. Furthermore, practical results and performance analysis have demonstrated that the new protocol outperforms typical ones in the same computational model. To demonstrate the effectiveness of the proposed solution, the study develops the protocols to compute frequency values for varying numbers of users and shows that the proposed protocol betters the other protocols.

The rest of the paper is organized as follows: Section 2 discusses the key theoretical foundations used in this study. In Section 3, the suggested computation protocol is described, studied, and compared to existing standard protocols in terms of privacy, communication, and computational cost. Section 4 finishes the paper.

II. Preliminaries

A. Simultaneous computation of multiple secure frequency values in the 2PFD model

In the two-party fully distributed setting (2PFD), a dataset (a data table) contains n records, with each record represented by attribute values. The dataset is distributed across two groups of users $U = U_1, U_2, ..., U_n$ and $V = V_1, V_2, ..., V_n$, where each user U_i,V_i holds a set of private binary values $u_{i,j}$,$v_{i,l} \in \{0, 1\}$, with $j \in [0, n_u)$, $l \in [0, n_v)$ and $n_u \geq n_v$. Each pair of users (U_i, V_i) holds one record, where U_i knows the values of a subset of relevant attributes, and V_i knows the values of the remaining attributes. The computation center needs to find the sum of the frequency values: $F = \{f_m\}_{m\in[0,n_u n_v)} = \{\sum_{i=1}^{n} u_0 v_0, \sum_{i=1}^{n} u_0 v_1, ..., \sum_{i=1}^{n} u_0 v_{n_v-1}, \sum_{i=1}^{n} u_1 v_0, ..., \sum_{i=1}^{n} u_{n_u-1} v_{n_v-1}\}$ without revealing the individual values of $u_{i,j}$ and $v_{i,l}$.

B. Definition of Privacy

Since this study follows the semi-honest model [15], in which each user must adhere to the rules of the protocol, but anyone may be corrupted, the definition of security frequency computation in 2PFD as follows:

Definition 1. Assume $(E_i^{(u)}, D_i^{(u)})$ and $(E_i^{(v)}, D_i^{(v)})$ are the corresponding sets of public and secret keys for each user U_i and V_i, respectively. A protocol for the above-defined multi-frequency computation problem protects each user's privacy against the centre computation along with t_1 colluding users U_i and t_2 colluding users V_i in the semi-honest model if, for all I_1, $I_2 \subset [1, n]$ such that $|I_1| = t_1$ and $|I_2| = t_2$, there exists a polynomial-time probabilistic algorithm M such that:
$\{M(F, [u_i, D_i^{(u)}]_{i\in I_1}, [E_j^{(u)}]_{j\in I_1}, [v_k, D_k^{(v)}]_{k\in I_2}, [E_l^{(v)}]_{l\in I_2})\} \stackrel{c}{\equiv}$

$$\{View_{miner, U_{i i\in I_1}, V_{k k\in I_2}}[u_i, D_i^{(u)}, v_i, D_i^{(v)}]_{i=1}^{n}\} \tag{1}$$

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

One key element to remember is that because the parties are semi-honest, they are assured to use the real inputs written on their input tapes. This is significant since it indicates that the output is properly defined and does not depend on the adversary. Specifically, for inputs x and y, the output is defined to be $f(x, y)$, and the simulator can be given that value.

Essentially, the definition states that computation is secure if the joint view of the computing centre and the colluding users (t_1 users U_i and t_2 users V_i) during the protocol execution can be efficiently simulated by a simulator, based on what the computing centre and the colluding users have observed in the protocol using only the result F, the knowledge of the colluding users, and the public keys. Therefore, the computing centre and the colluding users cannot learn anything from F. According to the definition, proving the privacy of a solution requires demonstrating the existence of a simulator that meets the above equation. This formulation implies that the parties learn nothing from the protocol execution beyond what they can derive from public values, their input and prescribed output.

C. Elliptic curve cryptography

Given an elliptic curve $E(F_p)$ over the field F_p with the point O called the point at infinity and p being a large prime number, the discrete logarithm problem on the elliptic curve is considered difficult. Additionally, G is a base point on the elliptic curve E with order d (i.e., $d.G = O$). The secret key is a random number $q \in [1, d)$, and the corresponding public key is $Q = qG$. To encrypt a plaintext message m, the sender uses the recipient's public key Q to compute the ciphertext C of the plaintext m as follows: the sender chooses a random number $k \in [1, d)$ and computes the ciphertext $C = (C_1 = P_m + kQ, C_2 = kG)$, where P_m is a point on the curve E with $x_{P_m} = m$. To decrypt the ciphertext C using the secret key q, the recipient can compute $m = x_M$, where $M = C_1 + (-qC_2)$.

III. Secure Multi-Frequency Computation Protocol in 2PFD

A. The Proposed Protocol

The detailed protocol for secure multi-frequency computing in the 2PFD model using the elliptic curve encryption scheme PPMFC() is presented in detail in Algorithm 1, with n_k and n_{k1} computed according to Equations 2 and 3, respectively.

Each user uses two pairs of common private and public keys to compute each private value $f_m = \sum_{i=1}^{n} u_{i,j} v_{i,l}$. Therefore, if the parties use n_k pairs of common private and public keys, they can encrypt up to $C_{n_k}^2$ private values. Hence, only n_k pairs of private and public keys need to be prepared, where n_k is determined as follows:

$$n_k = \left\lceil \frac{1}{2} + \sqrt{2n_u n_v + \frac{1}{4}} \right\rceil \tag{2}$$

As seen in Algorithm 1, each pair of common private and public keys is formed from two pairs of private and public keys from each U_i and V_i. Therefore, if each party uses n_{k1} pairs of private and public keys, they can generate up to $F_{n_{k1}}^2$ pairs of common private and public keys. Thus, each participating party only needs to prepare n_{k1} pairs

Input: n: the number of participant parties
$u_{i,j}$: the private values of i^{th} participant party in the first data domain, $i \in [1,n], j \in [1,n_u]$
$v_{i,j}$: the private values of i^{th} participant party in the second data domain, $i \in [1,n], j \in [1,n_v]$
Output: $f_j = \sum_{i=1}^{n} u_{i,\lfloor j/n_v \rfloor} v_{i,j\%n_v}$ where $j \in [0, n_u n_v)$

Begin
1. Initialization phase: - Each user U_i **does the following:**
$x_i = Random(1, d-1); X_i = x_i G;$
for $j \leftarrow 0$ **to** $n_{k1} - 1$ **do**
  $ksu_{i,j} = Random(1, d-1);$
  $KPU_{i,j} = ksu_{i,j} G;$
end
Sends to The computing center (CC): $\{KPU_{i,j}, X_i\}_{j \in [0, n_{k1})}$;
- Each user V_i **does the following:**
$y_i = Random(1, d-1); Y_i = y_i G;$
for $j \leftarrow 0$ **to** $n_{k1} - 1$ **do**
  $ksv_{i,j} = Random(1, d-1);$
  $KPV_{i,j} = ksv_{i,j} G;$
end
Sends to CC: $\{KPV_{i,j}\}_{j \in [0, n_{k1})}$;
- CC does the following:
$j = 1;$
for $t \leftarrow 0$ **to** $n_{k1} - 1$ **do**
  for $k \leftarrow 0$ **to** $n_{k1} - 1$ **do**
    $KP_j = \sum_{i=1}^{n} KPU_{i,t} + KPV_{i,k}; j++;$
    if $(j == n_k - 1)$ break;
  end
  if $(j == n_k - 1)$ break;
end
Sends to all users: $\{KP_j\}_{j \in [0, n_k)}$; **2. Phase 1: Each user** U_i **does the following: for** $j \leftarrow 0$ **to** $n_u - 1$ **do**
  $c_{i,j}^{(1)} = Random(1, d-1);$
  $C_1^{(i,j)} = u_{i,j} G + c_{i,j}^{(1)} X_i; C_2^{(i,j)} = c_{i,j}^{(1)} G;$
end
Sends to CC: $\{C_1^{(i,j)}, C_2^{(i,j)}\}_{j \in [0, n_u)}$;
3. Phase 2: Each user V_i **does the following:**
Gets $\{C_1^{(i,j)}, C_2^{(i,j)}\}_{j \in [0, n_u)}$ from CC;
$j = 0;$
for $t \leftarrow 0$ **to** $n_{k1} - 2$ **do**
  for $k \leftarrow t+1$ **to** $n_{k1} - 1$ **do**
    $c_{i,j}^{(2)} = Random(1, d-1);$
    $Q_1^{i,j} = v_{i,j\%n_v} C_1^{(i,\lfloor j/n_v \rfloor)} + ksv_{i,k\%n_{k1}} KP_t - c_{i,j}^{(2)} y_i C_2^{(i,\lfloor j/n_v \rfloor)} - ksv_{i,t\%n_{k1}} KP_k;$
    $Q_2^{(i,j)} = c_{i,j}^{(2)} Y_i - v_{i,j\%n_v} X_i; j++;$
    if $(j == n_u n_v - 1)$ break;
  end
  if $(j == n_u n_v - 1)$ break;
end
Sends to CC: $\{Q_1^{i,j}, Q_2^{(i,j)}\}_{j \in [0, n_u n_v)}$;
4. Phase 3: Each user U_i **does the following:**
Gets $\{Q_1^{i,j}, Q_2^{(i,j)}\}_{j \in [0, n_u n_v)}$ from CC;
$j = 0;$
for $t \leftarrow 0$ **to** $n_{k1} - 2$ **do**
  for $k \leftarrow t+1$ **to** $n_{k1} - 1$ **do**
    $A_{i,j} = Q_1^{i,j} + c_{i,\lfloor j/n_v \rfloor}^{(1)} Q_2^{(i,j)} - ksu_{i,\lfloor t/n_{k1} \rfloor} KP_k + ksu_{i,\lfloor k/n_{k1} \rfloor} KP_t;$
    $j++;$
    if $(j == n_u n_v - 1)$ break;
  end
  if $(j == n_u n_v - 1)$ break;
end
Sends to CC: $\{A_{i,j}\}_{j \in [0, n_u n_v)}$;
5. Phase 4: CC does the following:
for $t \leftarrow 0$ **to** $n_u n_v - 1$ **do**
  $A_j = \sum_{i=1}^{n} A_{l,j};$
end
$F = Dlog_E C(E, G, n, A);$
//Using the brute-force algorithm once
end

Algorithm 1: A secure protocol for computing multi-frequency in one round of computation

of private and public keys, where n_{k1} is calculated using the following formula

$$n_{k1} = \lceil \sqrt{n_k} \rceil \tag{3}$$

B. Proof of correctness

To demonstrate the correctness of the proposed protocol, the following proposition needs to be proven:

Theorem 1. The PPMFC protocol correctly computes the frequency values

$f_j = \sum_{i=1}^{n} u_{i,\lfloor j/n_v \rfloor} v_{i,j\%n_v}$.

Proof:

We have $f_j = Dlog_E C(E, G, n, A_j)$, so:

$f_j G = A_j = \sum_{i=1}^{n} Q_1^{i,j} + c_{i,\lfloor j/n_v \rfloor}^{(1)} Q_2^{(i,j)} - ksu_{i,\lfloor t/n_{k1} \rfloor} KP_k + ksu_{i,\lfloor k/n_{k1} \rfloor} KP_t$

$= \sum_{i=1}^{n} u_{i,\lfloor j/n_v \rfloor} v_{i,j\%n_v} G$

Since $f_j G = \sum_{i=1}^{n} u_{i,\lfloor j/n_v \rfloor} v_{i,j\%n_v} G$ therefore $f_j = \sum_{i=1}^{n} u_{i,\lfloor j/n_v \rfloor} v_{i,j\%n_v}$.

Note that for values of f_j that are not too large, we can use a brute-force algorithm to find f_j. This protocol remains valid even when the private values held by the users are positive integers rather than just $\{0, 1\}$, as long as the value of f_j is not too large and can be bounded (for example, 10^6).

C. Security analysis

We prove the privacy of the proposal protocol by the following theorem:

Theorem 2. The PPMFC protocol preserves each user's privacy in the semi-honest model. In the case of collusion among participants, the protocol protects the privacy of honest users against the computation centre and up to $2n - 2$ colluding users. In the case where there are only two honest users, this remains true as long as the two honest users do not possess attribute values from the same record.

Proof:

It is easy to see that the messages generated during the execution of the protocol form the set $P = \{KPU_{i,j}, KPV_{i,j}, KP_k, C_1^{(i,l)}, C_2^{(i,l)}, Q_1^{(i,t)}, C_2^{(i,t)}, A_{(i,t)}\}$ where $i \in [1, n]$, $j \in [0, n_{k1})$, $k \in [0, n_k)$, $l \in [0, n_u)$, $t \in [0, n_u n_v)$. Since $\{x_i, ksu_{i,j}, ksv_{i,j}, c_{i,l}^{(1)}, c_{i,t}^{(2)}\}$ are uniformly and randomly chosen from Z_d^*, so:

$Pr(t \leftarrow P, D(t) = 1) = \frac{1}{|Z_d^*|}$

Therefore, according to Definition 1, the PPMFC protocol ensures the privacy of each honest user in the semi-honest model.

Next, we present a simulator M (also referred to as a polynomial-time algorithm) that simulates the joint view of CC and the colluding users observed during the execution of the protocol, using only the results F, the colluding users' information, and the public keys. The algorithm outputs simulated values for the encryptions created by an EC simulator. Without loss of generality, assume U_1 and V_2 do not collude, the U_l collude ($l \in I_1 = [2, n]$), and the V_l collude ($l \in I_2 = 1, 3, 4, ..., n$). Therefore, the algorithm M proceeds as follows for $j \in [0, n_u n_v)$:

• M simulates $C_1^{(1,j)}$, $C_2^{(1,j)}$ using random EC ciphertexts.

• M takes the following encryptions as its input:

$(a_{1,j}, a_{2,j}) = (\alpha_j - ksv_{2,t\%n_{k1}}(ksu_{1,\lfloor k/n_{k1} \rfloor} + ksv_{2,k\%n_{k1}})G + ksv_{2,k\%n_{k1}}(ksu_{1,\lfloor t/n_{k1} \rfloor} + ksv_{2,t\%n_{k1}})G, ksv_{2,k\%n_{k1}} G)$

Where $\alpha_j = v_{2,j\%n_v}.C_1^{(2,\lfloor j/n_v \rfloor)} - y_2.c_{2,j}^{(2)}.C_2^{(2,\lfloor j/n_v \rfloor)}$ and it computes the following values:

$(Q_1^{'(2,j)} = a_{1,j} - (\sum_{i \in I_1} ksu_{i,\lfloor k/n_{k1} \rfloor} + \sum_{i \in I_2} ksv_{i,k\%n_{k1}}) KPU_{2,t\%n_{k1}} + (\sum_{i \in I_1} ksu_{i,\lfloor t/n_{k1} \rfloor} + \sum_{i \in I_2} ksv_{i,t\%n_{k1}}) KPV_{2,k\%n_{k1}} - (f_j - \sum_{l=3}^{n} u_{l,\lfloor j/n_v \rfloor} v_{l,j\%n_v} - \epsilon_j v_{1,j\%n_v} - \theta_j u_{2,\lfloor j/n_v \rfloor})G$

Where: ϵ_j, $\theta_j \in \{0, 1\}$. Next, M simulates $Q_2^{(2,j)}$ using a random ElGamal ciphertext.

• M simulates $A_{1,j}^{'}$:

$(A_{1,j}^{'} = b_{1,j} + (\sum_{i \in I_1} ksu_{i,t_1} + \sum_{l \in I_2} ksu_{l,t_2}) KPU_{1,k_1} - (\sum_{i \in I_1} ksu_{i,k_1} + \sum_{l \in I_2} ksu_{l,k_2}) KPU_{1,t_1})$

Where:

$(b_{1,j}, b_{2,j}) = (Q_1^{(1,j)} + c_{1,\lfloor j/n_u \rfloor}^{(1)} Q_2^{(1,j)} - ksu_{1,\lfloor t/n_{k1} \rfloor}(ksu_{1,\lfloor k/n_{k1} \rfloor} + ksu_{2,k\%n_{k1}})G + ksu_{1,\lfloor k/n_{k1} \rfloor}(ksu_{1,\lfloor t/n_{k1} \rfloor} + ksu_{2,t\%n_{k1}})G, ksu_{1,\lfloor k/n_{k1} \rfloor} G)$

Since $\{ksu_{i,\lfloor t/n_{k1} \rfloor}, ksu_{i,\lfloor k/n_{k1} \rfloor}, ksu_{l,t\%n_{k1}}, ksu_{l,k\%n_{k1}}\}_{i \in I_1, l \in I_2} \in_R Z_d^*$, it follows that:

$Pr(t \leftarrow \{Q_1^{'(2,j)}, A_{1,j}^{'}\}_{j \in [0, n_u n_v)}, D(t) = 1) = \frac{1}{|Z_d^*|}$

From the above arguments, we can see that the simulator M satisfies Definition 1, thus Theorem 2 is proved.

D. Performance Evaluation

In this section, the protocols [2], [4], [14] with the same level of security and the same number of interactions between the participants and CC are compared with the proposed protocol in terms of communication cost, computational cost, and execution time. The protocols [2], [4] will use the Elgamal encryption

TABLE 1. THE COMMUNICATION OVERHEAD COMPARISON BETWEEN THE PROTOCOLS (BIT)

Protocols	Communication overhead
The protocol [2]	$64n.n_u n_v \|p\|$
The protocol [4]	$76,8n.n_u n_v \|p\|$
The protocol [14]	$19n.n_u n_v \|p\|$
The new protocol	$2n_{k1} + 4n_u + 5n_u n_v + 2n_k + 2$ <2n($\sqrt{0,5n_u n_v + 2}$ $+2n_u + 3n_u n_v + 3)\|p\|$

system with a secret key size of 160 bits and a public key size of 1024 bits [16], while the remaining protocols use the Elliptic Curve Cryptography system with the BrainpoolP160r1 curve, having a secret key size of 160 bits and a public key size of 320 bits [17], Both encryption systems offer the same level of security. The size of each message in the Elliptic Curve Cryptography system is $|p|$.

1) Communication Overhead

First, we review the protocol in [2], to compute each frequency value, it requires exchanging $20n$ messages, corresponding to $20n\frac{16|p|}{5} = 64n|p|$ bits. To compute $n_u n_v$ frequency values, it is necessary to exchange $20n.n_u n_v$ messages, corresponding to $64n.n_u n_v|p|$ bits. By applying a similar analysis to the remaining protocols, we obtain the communication costs of the protocols for computing $n_u n_v$ frequency values as shown in Table 1.

We have:

$n_k < \lceil\sqrt{2n_u n_v}\rceil + 1 < 0,5n_u n_v + 2$

From Table I, it can be seen that the proposed protocol exchanges a lower number of bits compared to the other typical protocols.

2) Computational cost

Because the computational cost of the protocols mainly involves the execution time of operations such as modular multiplication, modular exponentiation, modular multiplicative inverse, modular point addition, and point multiplication on the Elliptic Curve with large integers. Therefore, the computational cost of the protocols is evaluated through these operations. First, the paper tabulates the number of calculations for each protocol. Then, for convenience in evaluating the computational cost,

TABLE 2. DEFINITIONS AND CONVERSIONS OF VARIOUS OPERATIONAL UNITS

Notations	Definition	Conversion
T_m	Time complexity for executing the modular multiplication	
T_e	Time complexity for executing the modular exponentiation	$1T_e \approx 240T_m$ [18]
T_i	Time complexity for executing the modular inversion operation	$1T_i \approx 11,6T_m$ [19]
T_{ap}	Time complexity for executing the addition of two points in an elliptic curve	$1T_{ap} \approx 0,12T_m$ [18]
T_{mp}	Time complexity for executing the multiplication of a number and an elliptic curve point	$1T_{mp} \approx 29T_m$ [18]
T_{DL}	Time complexity for executing the discrete logarithm	

the paper denotes the execution time of these operations and converts the time complexity of different calculations into a unified unit, which is the time complexity of performing modular multiplication [18] as shown in Table 2:

The details of the theoretical computational cost of the protocols are specifically presented in the Table 3.

From the table above, it can be seen that the computational cost for each user U_i is converted to the equivalent execution of $2.164n_u n_v$ modular multiplications in the protocol [2], $2.408n_u n_v$ modular multiplications in the protocol [4], $261,6n_u n_v$ modular multiplications in the protocol [14] and less than $(29 + 87,12n_u + 87,36n_u n_v + 29\lceil\sqrt{0,5n_u n_v + 2}\rceil)$ modular multiplications in the new protocol. Therefore, the computational cost for each user U_i in the proposed protocol is lower than in the other protocols. A similar analysis shows that the computational cost for each user V_i and CC in the protocols [2], [4], [14] is higher the proposed protocol. This is due to the significantly reduced number of keys used in the proposed protocol and the optimization of the computations between the parties, resulting in lower computational costs for each user and CC compared to the other

TABLE III. THE COMPUTATIONAL COMPLEXITY COMPARISONS AMONG THE PROTOCOLS

Protocols	U_i	V_i	CC
The protocol [2]	$n_u n_v(9T_e + 4T_m)$ $\approx$ $2.164 n_u n_v T_m$	$n_u n_v(9T_e + 3T_m + T_i)$ $\approx$ $2.174, 6 n_u n_v T_m$	$n_u n_v(T_i + T_{DL} + 6nT_m)$ $\approx T_m(6n + 11, 6) n_u n_v + n_u n_v T_{DL}$
The protocol [4]	$n_u n_v(10T_e + 8T_m)$ $\approx$ $2.408 n_u n_v T_m$	$n_u n_v(9T_e + 3T_m + T_i)$ $\approx$ $1.934, 6 n_u n_v T_m$	$n_u n_v(T_i + T_{DL} + 4nT_m)$ $\approx T_m(4n + 11, 6) n_u n_v + n_u n_v T_{DL}$
The protocol [14]	$n_u n_v(9T_{mp} + 5T_{ap}) \approx$ $261, 6 n_u n_v T_m$	$n_u n_v(9T_{mp} + 3T_{ap}) \approx$ $261, 36 n_u n_v T_m$	$n_u n_v(5nT_{ap} + T_{DL}) \approx$ $n_u n_v(T_{DL} + 0, 6nT_m)$
The new protocol	$(3n_u + 3n_u n_v + n_{k1} + 1)T_{mp} + (3n_u n_v + n_u)T_{ap}$ $< T_m(29 + 87, 12 n_u + 87, 36 n_u n_v + 29\lceil\sqrt{0, 5 n_u n_v + 2}\rceil)$	$(6n_u n_v + n_{k1} + 1)T_{mp} + 4n_u n_v T_{ap}$ $< T_m(29 + 174, 48 n_u n_v + 29\lceil\sqrt{0, 5 n_u n_v + 2}\rceil) n_u n_v) T_m$	$n(2n_k + n_u n_v)T_{ap} + T_{DL}$ $< T_{DL} + 0, 24n(2 + n_u n_v) T_m$

protocols. Furthermore, instead of requiring each user U_i to compute and send $2n_u n_v$ messages, the new protocol only requires the computation and sending of $2n_u$ messages. This makes the proposed protocol more efficient.

3) Running time

In this section, the typical protocols [2], [4], [14] with the same security level and the same number of interactions between the participating parties and the computation centre, along with the proposed protocol, are implemented in the same environment to compare execution time. The protocols to be compared will be executed using $C\#$ language in the Visual Studio 2019 environment, with the support of the System.Numerics library to compare the performance of these protocols. The experiments are run on an Intel Core i5-4210M CPU 2.60GHz laptop with 8GB RAM. To minimize differences in the experimental environment, each protocol will be executed independently with no other applications running on the computer. The execution time of the protocols will be measured

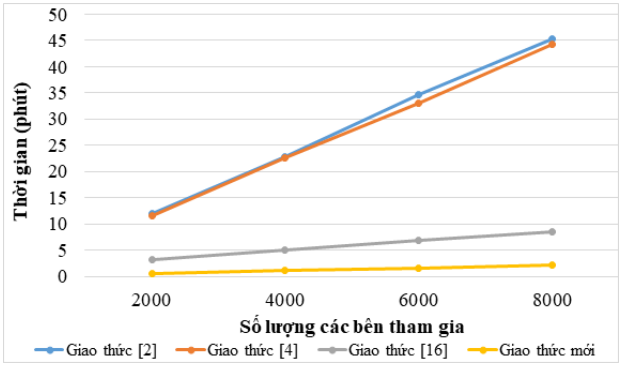

Figure 1. Comparison of execution time between the protocols

using the Stopwatch class in the System.Diagnostics library.

To compare the execution time of the protocols, each protocol is executed 10 times for each parameter set, and the average value is taken. We assume that all users perform their tasks simultaneously, and network latency is not included in the total execution time. Therefore, the computational cost on the user side can be reduced to the computational cost for a single user, while the computational and communication costs on the consulting server side are calculated for all users participating in the system, as they depend on the collaboration of all users with the server. The experimental results are presented in Figure 1.

From the above results, it can be seen that the proposed protocol has better performance compared to the typical protocols previously published within the same computational model.

IV. CONCLUSION

Preserves the accuracy of the output equivalent to the original operation before applying security measures but also offers better computational and communication efficiency compared to previously published protocols in the same computational model. The theoretical proof further demonstrates that the protocol provides a level of security equivalent to that of typical high-security protocols. The proposed protocol ensures the privacy of honest participants in a semi-honest model, and in cases where a large number of semi-honest users collude $(2n - 2)$ the privacy of the honest participants is still maintained. The experimental results once again demonstrate the effectiveness

of the proposed protocol. Therefore, this protocol can be practically applied.

References

[1] M. I. Pramanik, R. Y. Lau, M. S. Hossain, M. M. Rahoman, S. K. Debnath, M. G. Rashed, and M. Z. Uddin, "Privacy preserving big data analytics: A critical analysis of state-of-the-art," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 11, no. 1, p. e1387, 2021.

[2] T. D. LUONG and T. B. HO, "Privacy preserving frequency mining in 2-part fully distributed setting," *IEICE Transactions on Information and Systems*, vol. E93.D, no. 10, pp. 2702–2708, 2010.

[3] D. L. . T., "Luận án tiến sĩ: Nghiên cứu xây dựng một số giải pháp đảm bảo an toàn thông tin trong quá trình khai phá dữ liệu," *Viện Khoa học và Công nghệ quân sự*, 2011.

[4] T. D. Luong and D. H. Tran, "Erratum: Privacy preserving frequency-based learning algorithms in two-part partitioned record model," in *Knowledge and Systems Engineering: Proceedings of the Fifth International Conference KSE 2013, Volume 2*. Springer, 2014, pp. 445–445.

[5] V. T. Van, L. T. Dung, H. V. Quan, T. T. Luong, and H. D. Tho, "Privacy-preserving decision tree solution in the 2-part fully distributed setting," *Journal of Science and Technology on Information security*, vol. 1, no. 15, pp. 92–101, 2022.

[6] C. Dong and L. Chen, "A fast secure dot product protocol with application to privacy preserving association rule mining," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2014, pp. 606–617.

[7] B. Siabi, M. Berenjkoub, and W. Susilo, "Optimally efficient secure scalar product with applications in cloud computing," *IEEE Access*, vol. 7, pp. 42 798–42 815, 2019.

[8] Q. Tang and H. Wang, "Privacy-preserving hybrid recommender system," in *Proceedings of the Fifth ACM International Workshop on Security in Cloud Computing*, 2017, pp. 59–66.

[9] D.-H. Vu, T.-D. Luong, and T.-B. Ho, "An efficient approach for secure multi-party computation without authenticated channel," *Information Sciences*, 2019.

[10] D.-H. Vu, T.-S. Vu, and T.-D. Luong, "An efficient and practical approach for privacy-preserving naive bayes classification," *Journal of information Security and Applications*, vol. 68, p. 103215, 2022.

[11] S. Mehnaz, G. Bellala, and E. Bertino, "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*, ser. SACMAT '17 Abstracts. New York, NY, USA: Association for Computing Machinery, 2017, p. 219–230. [Online]. Available: https://doi.org/10.1145/3078861.3078869

[12] D.-H. Vu, "Privacy-preserving naive bayes classification in semi-fully distributed data model," *Computers & Security*, vol. 115, p. 102630, 2022.

[13] T. D. Luong and D. H. Tran, "A new method of privacy preserving computation over 2-part fully distributed data," in *The 9th International Conference on Computing and InformationTechnology (IC2IT2013)*, P. Meesad, H. Unger, and S. Boonkrong, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 115–123.

[14] T. Van Vu, T. D. Luong *et al.*, "An elliptic curve-based protocol for privacy preserving frequency computation in 2-part fully distributed setting," in *2020 12th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2020, pp. 91–96.

[15] O. Goldreich, *Foundations of Cryptography, Volume 2*. Cambridge university press Cambridge, 2004.

[16] M. Lepinski and S. Kent, "Rfc 5114: Additional diffie-hellman groups for use with ietf standards," 2008.

[17] M. Lochter and J. Merkle, "Elliptic curve cryptography (ecc) brainpool standard curves and curve generation," Tech. Rep., 2010.

[18] Y. F. Chung, K. H. Huang, F. Lai, and T. S. Chen, "Id-based digital signature scheme on the elliptic curve cryptosystem," *Computer Standards & Interfaces*, vol. 29, no. 6, pp. 601–604, 2007.

[19] S. H. Islam and G. Biswas, "A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks," *Annals of telecommunications-annales des telecommunications*, vol. 67, pp. 547–558, 2012.

ABOUT THE AUTHORS

Vu Thi Van
Workplace: Academy of Cryptography Techniques, Vietnam Government Information Security Commission
Email: vanvu10101986@gmail.com
Education: Engineer of Information Security from the Academy of Cryptography, Hanoi, Viet Nam, in 2009, Master's degree in Information Security from Academy of Cryptography Techniques, in 2016. She is currently a Ph.D. candidate in Infomation security, at the Academy of Cryptography, Vietnam
Recent research interests: Secure multi-party computation; data mining; cyber security.

Tên tác giả: **Vũ Thị Vân**
Cơ quan công tác: Học viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ Việt Nam
Email: vanvu10101986@gmail.com
Quá trình đào tạo: Nhận bằng Kỹ sư và Thạc sỹ An toàn thông tin tại Học viện Kỹ thuật mật mã lần lượt vào năm 2009 và năm 2016. Hiện đang làm Nghiên cứu sinh An toàn thông tin tại Học viện Kỹ thuật mật mã
Hướng nghiên cứu hiện nay: Bảo mật tính toán đa bên; khai phá dữ liệu, an ninh mạng.

Luong The Dung
Workplace: Academy of Cryptography Techniques, Vietnam Government Information Security Commission
Email: thedungluong1@gmail.com
Education: Received Bachelor of Information Technology from Le Quy Don Technical University, Ha Noi, Viet Nam in 2001 and a Ph.D. degree in 2012 from Institute of Military Science and Technology, Ha Noi, Viet Nam
Recent research interests: Privacy-preserving data mining and computer security.

Tên tác giả: **Lương Thế Dũng**
Cơ quan công tác: Học viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ Việt Nam
Email: thedungluong1@gmail.com
Quá trình đào tạo: Nhận bằng Cử nhân Công nghệ thông tin tại Trường Đại học Kỹ thuật Lê Quý Đôn vào năm 2001; Nhận bằng Tiến sỹ tại Viện Khoa học và Công nghệ quân sự vào năm 2012.
Hướng nghiên cứu hiện nay: Khai phá dữ liệu bảo vệ quyền riêng tư; bảo mật máy tính.

Luong Ngoc Duong
Workplace: Academy of Cryptography Techniques, Vietnam Government Information Security Commission
Email: luongngocduongkma@gmail.com
Education: He is currently a final-year student at the Academy of Cryptography Techniques, Hanoi, Viet Nam, majoring in Information Security.
Recent research interests: Secure multi-party computation; data mining; malware.

Tên tác giả: **Lương Ngọc Dương**
Cơ quan công tác: Học viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ Việt Nam
Email: luongngocduongkma@gmail.com
Quá trình đào tạo: Hiện đang là sinh viên năm cuối chuyên ngành An toàn thông tin tại Học viện Kỹ thuật mật mã.
Hướng nghiên cứu hiện nay: Bảo mật tính toán đa bên; khai phá dữ liệu; mã độc.