Efficient Quantum Secure Vector Dominance and Its Applications in Computational Geometry

Wenjie Liu, Bingmei Su, and Feiyang Sun

***Abstract*—Secure vector dominance is a key cryptographic primitive in secure computational geometry (SCG), determining the dominance relationship of vectors between two participants without revealing their private information. However, the security of traditional SVD protocols is compromised by the formidable computational power of quantum computing, and their efficiency needs further improvement. To address these challenges, an efficient quantum secure vector dominance (QSVD) protocol is proposed. Specifically, we first introduce a quantum private permutation (QPP) subprotocol to shuffle the elements of each participant's private input vector. To further facilitate secure data comparison, we propose an enhanced quantum millionaire subprotocol with equality determination functionality, building upon Jia's original protocol. Based on the above two subprotocols, we propose a QSVD protocol with polynomial complexity, deriving vector dominance in a single interaction with a semi-honest third party. Performance analyses confirm that QSVD protocol is correct, resilient against malicious attacks, and retains polynomial computational complexity, ensuring both security and efficiency. To demonstrate the scalability of the QSVD protocol, we illustrate its applications in several geometric computation problems, such as point-line inclusion determination, line-line intersect determination, and point-in-polygon determination. Finally, we validate the feasibility of our protocol by conducting comprehensive simulations on IBM's Qiskit platform, demonstrating its practical applicability and effectiveness in real quantum computing environments.**

***Index Terms*—Quantum computation, quantum communication, secure multi-party computation, vector dominance, geometric computation.**

I. Introduction

SECURE Multi-Party Computation (SMC) originated from the "Millionaire Problem" proposed by Andrew Yao [1], with the goal of enabling multiple distrustful parties to collaboratively compute a target function without revealing their private data. SMC has since evolved into several subfields, including Anonymous Voting (AV) [2], [3], Private Set Intersection (PSC) [4], [5], [6], and Secure Computational Geometry (SCG) [7], [8], [9], [10], [11], [12], etc.

As an important research branch, SCG focuses on enabling geometric computations while ensuring the privacy of input data. In scenarios where multiple parties collaborate to solve geometric problems, such as intersection testing [7], [8], proximity queries [9], [10], or shape matching [11], [12], SCG techniques ensure that sensitive geometric data from each party remains confidential throughout the computation. Considering the forms of geometric computation, SCG can be categorized into two types: one based on set operations, referred to as set-based SCG [9], [10], [12], and the other based on geometric features, known as geometric-based SCG [7], [8], [11]. The former divides the geometric shape into a grid, with each grid point being sufficiently small, allowing for approximate geometric calculations based on these grid points. While this method is straightforward, it may not provide high accuracy for boundary points. In contrast, geometric-based SCG employs geometric constraints and mathematical definitions to perform privacy-preserving computations. While this approach enables precise determinations, particularly for boundary points, it comes with higher computational complexity.

With the rapid development of quantum information technology, the limitations of Secure Multi-party Computation (SMC) based on computational complexity assumptions have become evident, as it becomes vulnerable to attacks from quantum computers capable of parallel acceleration [13], [14]. However, by integrating quantum mechanisms with cryptographic techniques, it is possible to achieve unconditional security in SMC, leading to the development of Quantum Secure Multi-party Computation (QSMC) [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. This field encompasses Quantum Private Comparison (QPC) [17], [18], [19], Quantum Anonymous Voting (QAV) [20], [21], Quantum Sealed-Bid Auction (QSA) [22], [23], [24], and Quantum Private Set Computation (QPSC) [25], [26], [27], [28].

Recently, scholars have begun adopting quantum cryptographic techniques to facilitate multi-party computations on geometric data, known as Quantum Secure Computational Geometry (QSCG). In 2017, Shi et al. [30] introduced a method that partitions geometric shapes into grids, enabling approximate geometric computations via private set operations, and

Received 30 October 2024; revised 17 February 2025; accepted 28 March 2025. Date of publication 4 April 2025; date of current version 12 May 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62071240, in part by the Innovation Program for Quantum Science and Technology under Grant 2021ZD0302901, and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20231142. Recommended for acceptance by X. Jia. *(Corresponding author: Wenjie Liu.)*

Wenjie Liu and Feiyang Sun are with the School of Software, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: wenjiel@163.com; sunfeiyang97@163.com).

Bingmei Su is with the School of Computer Science, Nanjing University of Information Science and Technology, Nanjing 210044, China (e-mail: bmsu363@163.com).

Digital Object Identifier 10.1109/TC.2025.3557968

proposed a quantum point-inclusion protocol based on phase-encoded queries. Subsequently, many quantum protocols were developed to address two-party distance computations [31], [32], and geometric intersections [33], [34]. More recently, to address cryptographic problems with broader practical applications, researchers have focused on developing cryptographic primitives that incorporate both quantum acceleration effects and quantum unconditional security. In 2023, Liu et al. proposed a secure two-party quantum scalar product (S2QSP) protocol and introduced a quantum matrix multiplication protocol [35]. In 2024, Dou et al. [36] extended the existing two-party addition and multiplication protocols to multi-party settings and proposed three related application protocols. To the best of our knowledge, most existing methods are based on ensemble SCG. However, these approaches encounter limitations, including difficulty in precisely identifying boundary nodes and high resource consumption. The motivation for our paper is to address these limitations by focusing on geometry-based SCG methods, which has not been explored in prior literature.

As is well-known, Secure Vector Domination (SVD) [37], [38] enables participants to securely determine whether one vector dominates another without disclosing their private information, which serves as a fundamental cryptographic primitive in secure computational geometry. However, traditional SVD protocols are vulnerable to quantum computing's computational power, and suffer from efficiency limitations. To overcome these challenges, we propose an efficient Quantum Secure Vector Domination (QSVD) protocol, which is applied to several geometric computation problems, including point-line inclusion determination, line-line intersection determination, and point-in-polygon determination. The contributions of our work are summarized below.

- A quantum private permutation subprotocol is introduced to shuffle the elements of each participant's private input vector. Additionally, an enhanced quantum millionaire subprotocol with equality determination functionality is proposed, building upon Jia's original protocol.
- Based on the above two subprotocols, we propose a QSVD protocol with polynomial complexity, deriving vector dominance in a single interaction with a semi-honest third party.
- To demonstrate the scalability of the QSVD protocol, several privacy-preserving geometric computation protocols are proposed, including point-line inclusion determination, line-line intersect determination and point-in-polygon determination.
- We analyze the performance of the QSVD protocol, demonstrating its correctness and unconditional security. Additionally, its feasibility is validated through the IBM Qiskit simulator.

The remainder of this paper is structured as follows: Section II outlines the definition of notations, fundamental concepts of quantum computing, and definitions of composite quantum gates used in our protocols. In Section III, we propose a quantum private permutation subprotocol for shuffling vector elements and an enhanced quantum millionaire subprotocol for secure data comparison. Based on them, we further propose an efficient quantum secure vector dominance protocol. In Section IV, after analyzing the correctness, security, and complexity of the proposed QSVD protocol, we verify its feasibility on IBM's Qiskit platform. Section V presents the application of the QSVD protocol to several geometric computation problems. Section VI discusses the strategies considered for enhancing the robustness of the QSVD protocol in noisy environments, focusing on quantum error correction codes and error mitigation techniques, followed by a summary of the paper in Section VII.

TABLE I
DEFINITION OF NOTATIONS

Symbols	Meanings
θ	The polar angle between the state vector and the z-axis
ϕ	The azimuthal angle in the xy-plane relative to the z-axis
α, β	Complex coefficients, $\lvert\alpha\rvert^2 + \lvert\beta\rvert^2 = 1$
$\lvert 0\rangle, \lvert 1\rangle$	Quantum basis state
$\lvert\psi\rangle, \lvert\Psi\rangle$	A quantum state in Hilbert space
ζ	$e^{\imath\frac{2\pi}{D}} = \cos(\frac{2\pi}{D}) + \imath\sin(\frac{2\pi}{D})$
$\oplus$	Bitwise XOR
(h, t)	System composed of particles h and t
$m, N = 2^m$	Output's bit number and modulus
$d, D = 2^d$	Particle's qubit number and dimension
$[N]$	Set $\{0, 1, \cdots, N-1\}$
$[D]$	Set $\{0, 1, \cdots, D-1\}$
$\succ$	Dominant symbol
V_A	Fixed vector
$\pi(\cdot)$	Bob's permutation operation
$r'_1 r'_2 \cdots r'_n$	A string of random bits where $r'_i \in \{0, 1\}$
$R = (r_1, r_2, \cdots, r_n)$	Bob's random vector
$C = c_1, c_2, \cdots, c_{4n}$	Alice's random vector
$H = (H_1, H_2, \cdots, H_n)$	Sequence composed of the first particles
$T = (T_1, T_2, \cdots, T_n)$	Sequence composed of the second particles

II. PRELIMINARY

A. Definition of Notations

Table I lists the definition of notations that will be used in our paper.

B. Quantum States

In quantum computing, quantum states are represented by vectors in a Hilbert space. The most basic unit of quantum information is the qubit, analogous to a classical bit, but with one crucial difference: while a classical bit can be either in state 0 or 1, a qubit exists in a superposition of both states simultaneously. Mathematically, the state of a qubit can be expressed as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1}$$

where α and β are complex coefficients, that satisfy the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. To provide a more intuitive

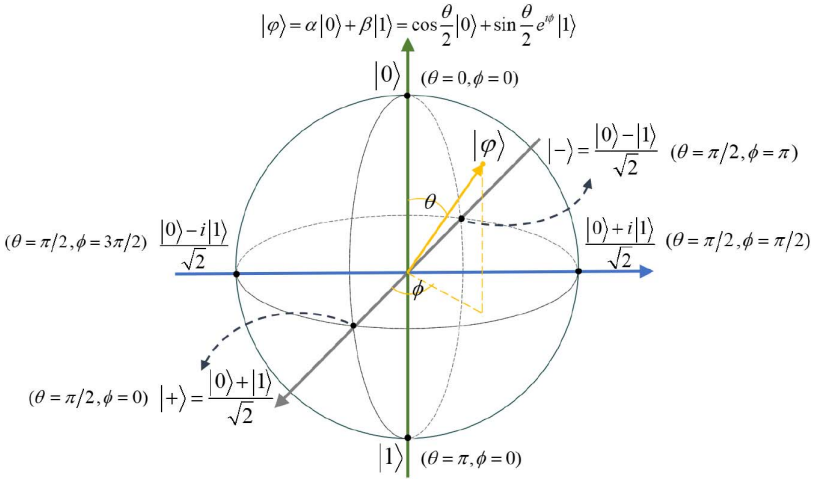


Fig. 1. **Bloch sphere representation of a qubit.** The Bloch sphere provides a geometric representation of a qubit, where any quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ can be mapped to a point on the sphere using two parameters, θ and ϕ. Specifically, $\alpha = \cos\frac{\theta}{2}$ and $\beta = \sin\frac{\theta}{2}e^{i\phi}$, where θ is the angle between the state vector and the z-axis, and ϕ represents the azimuthal angle in the xy-plane relative to the z-axis. As an example, when $\alpha = \frac{1}{\sqrt{2}}$ and $\beta = \frac{1}{\sqrt{2}}$, the quantum state is $|\psi\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, which illustrates the fundamental distinction between a qubit and a classical bit: a qubit can exist in a coherent superposition of both the $|0\rangle$ and $|1\rangle$ states simultaneously.

representation, a Bloch sphere can be used to describe the actual state of a qubit (shown in Fig. 1).

Multi-qubit systems refer to quantum systems composed of multiple qubits, with their state expressed as:

$$|\Psi\rangle = \sum_{i_1,i_2,\ldots,i_n} c_{i_1 i_2 \ldots i_n} |i_1\rangle |i_2\rangle \ldots |i_n\rangle , \tag{2}$$

where $i_1, i_2, \ldots, i_n \in [0, 1]$ represent the possible states of each qubit, and $c_{i_1 i_2 \ldots i_n}$ are complex coefficients that satisfy the normalization condition:

$$\sum_{i_1,i_2,\ldots,i_n} |c_{i_1 i_2 \ldots i_n}|^2 = 1. \tag{3}$$

Taking two-qubit state as an example, in comparison to the four possible states of two classical bits (00, 01, 10, 11), two qubits correspondingly have four basis states: $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. Thus, a two-qubit state can be expressed as a linear combination of these basis states:

$$|\Psi\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle , \tag{4}$$

where $\sum_{i_1 i_2 \in \{0,1\}^2} |c_{i_1 i_2}|^2 = 1$.

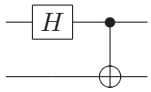
C. Quantum Gates

Similar to how classical computers use logic gates to manipulate bits, quantum computers employ quantum gates to manipulate qubits. These quantum gates are represented by unitary matrices, and their application to a quantum state transforms it into a new state.

(1) **Hadamard Gate (H):** This gate puts a qubit into an equal superposition of $|0\rangle$ and $|1\rangle$, and is represented by the matrix:

$$H = \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{5}$$

And its corresponding circuit is:

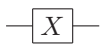


When applied to $|0\rangle$, it produces the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

(2) **Pauli-X Gate (X):** This gate is equivalent to the classical NOT gate, flipping $|0\rangle$ to $|1\rangle$ and vice versa:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{6}$$

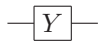
Circuit:



(3) **Pauli-Y Gate (Y):** The Pauli-Y gate flips the state and introduces a phase shift.

$$Y = \begin{pmatrix} 0 & -\imath \\ \imath & 0 \end{pmatrix} \tag{7}$$

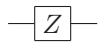
Circuit:



(4) **Pauli-Z Gate (Z):** The Pauli-Z gate introduces a phase shift of π when the qubit is in state $|1\rangle$.

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{8}$$

Circuit:

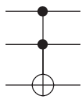


(5) **Controlled-NOT Gate (CNOT):** This is a two-qubit gate where the second qubit (target) is flipped if the first qubit (control) is $|1\rangle$. The CNOT gate is crucial for creating entanglement between qubits. Circuit:

(6) **Toffoli Gate (T):** The Tofolli gate is a phase gate that introduces a phase shift of $\frac{\pi}{4}$. It is represented by:

$$T = \begin{pmatrix} 1 & 0 \\ 0 & e^{\imath\pi/4} \end{pmatrix} \tag{9}$$

Circuit:



(7) **SWAP Gate:** The SWAP gate exchanges the states of two qubits. Its matrix representation is:

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

Circuit:

D. Definition of Composite Quantum Gates

To provide a cohesive description of the composite quantum gates used in the following sections, their symbolic representations and corresponding functions are presented using two d-qubit particles, t and h. All addition and multiplication operations are performed modulo $D = 2^d$, where $[D] = 0, 1, \cdots, D-1$. The gates are shown below.

(1) Modular multiplication gate $MUL(y)$, where $y \in [D]$ is an odd number:

$$MUL(y)_t : |x\rangle_t \to |xy\rangle_t . \tag{11}$$

(2) Rotation gate $ROT(y)$ where $y \in [-D, D]$ is a real number:

$$ROT(y)_t : |x\rangle_t \to \zeta^{yx} |x\rangle_t . \tag{12}$$

(3) Bi-particle modular summation gate $BSUM$:

$$BSUM_{(t,h)} : |x\rangle_t |y\rangle_h \to |x\rangle_t |y + x\rangle_h . \tag{13}$$

(4) Bitwise XOR gate XOR:

$$XOR_{(t,h)} : |x\rangle_t |y\rangle_h \to |x\rangle_t |y \oplus x\rangle_h . \tag{14}$$

(5) Quantum fourier transform QFT and its inverse:

$$QFT_t : |x\rangle_t \to \frac{1}{\sqrt{D}} \sum_{j\in[D]} \zeta^{jx} |j\rangle_t , \tag{15}$$

$$QFT_t^{\dagger} : \frac{1}{\sqrt{D}} \sum_{j\in[D]} \zeta^{jx} |j\rangle_t \to |x \bmod D\rangle_t , \tag{16}$$

where $\zeta = e^{\frac{\imath 2\pi}{D}}$. Even if x is not an integer, Eq (16) still holds approximately, as a special example of the phase estimation algorithm [39].

III. Proposed Protocol

In this section, We propose a quantum private permutation subprotocol and an enhanced quantum millionaire subprotocol. Based on this work, we propose a quantum secure vector dominance protocol to realize the determination of the dominant relationship between the vectors held by two participants without revealing their respective private information.

A. Quantum Private Permutation Subprotocol

To ensure the privacy security of the vector dominance determination, we need to solve the private permutation problem first. Based on the following, we present the quantum private permutation subprotocol.

Protocol 1: Quantum private permutation (QPP).

Input: $N = 2^m$. Alice has a vector $A = (a_1, a_2, \cdots, a_n) \in [N]$. Bob has a permutation $\pi(\cdot)$ and a random vector $R = (r_1, r_2, \cdots, r_n) \in [N]$.

Output: Alice gets $\pi(A + R)$.

Step 1 Set $d = m + 2$, $D = 2^d$. Alice encodes each of its elements a_i into a quantum state:

$$|\alpha\rangle = \frac{1}{\sqrt{D}} \sum_{j_i=0}^{D-1} \zeta^{a_i j_i} |j_i\rangle_{h_i} |j_i\rangle_{t_i}, \tag{17}$$

where t_i and h_i are particles. Then Alice picks out the first particle from each state to form an ordered sequence $H = \{H_1, H_2, \cdots, H_n\}$, the second particle to form sequence $T = \{T_1, T_2, \cdots, T_n\}$.

Step 2 Alice prepares a particle g and initializes it to $|0\rangle_g$, then performs $BSUM$ to all particles in the H sequence with particle g:

$$\begin{aligned} &|j_1\rangle_{h_1} |0\rangle_g \overset{BSUM_{(h_1,g)}}{\longrightarrow} |j_1\rangle_{h_1} |j_1\rangle_g , \\ &|j_2\rangle_{h_2} |j_1\rangle_g \overset{BSUM_{(h_2,g)}}{\longrightarrow} |j_2\rangle_{h_2} |j_1 + j_2\rangle_g , \\ &\qquad\vdots \\ &|j_n\rangle_{h_n} |j_1 + \cdots + j_{(n-1)}\rangle_g \\ &\overset{BSUM_{(h_n,g)}}{\longrightarrow} |j_n\rangle_{h_n} |j_1 + j_2 + \cdots + j_n\rangle_g , \end{aligned} \tag{18}$$

then sends H to Bob.

Step 3 Bob shuffles particles H_i to $H_{\pi(i)}$ and sends them back to Alice, marked $\pi(H)$. Alice performs $MUL(D-1)$ to particle h_i to calculates $|-j_i\rangle_{h_i}$:

$$|j_i\rangle_{h_i} \overset{MUL(D-1)_{h_i}}{\longrightarrow} |Dj_i - j_i \bmod D\rangle_{h_i} = |-j_i\rangle_{h_i}, \tag{19}$$

then performs $BSUM$ to particle g in turn:

$$\begin{aligned} &|-j_n\rangle_{h_n} |j_1 + j_2 + \cdots + j_n\rangle_g \\ &\overset{BSUM_{(h_n,g)}}{\longrightarrow} |-j_n\rangle_{h_n} |j_1 + j_2 + \cdots + j_{(n-1)}\rangle_g , \\ &\qquad\vdots \\ &|-j_1\rangle_{h_1} |j_1\rangle_g \overset{BSUM_{(h_1,g)}}{\longrightarrow} |-j_1\rangle_{h_1} |0\rangle_g , \end{aligned} \tag{20}$$

then measures the particle g. If the measurement result is $|0\rangle$, then continue; otherwise, the protocol is terminated.

Step 4 Alice performs Bi-particle modular summation gate $BSUM$ to all particles in the T sequence with particle g, same as step 2, then sends T to Bob.

Step 5 Bob prepares a random vector $R=(r_1,r_2,\cdots,r_n)$. He shuffles the particles T and vector R to $\pi(T)$ and $\pi(R)$. Then performs rotation gate $ROT(r_i)$ to each particle, then sends them back to Alice.

Step 6 Alice performs the same operations in step 3 with particles $\pi(T)$, then measures the particle g. If the measurement result is $|0\rangle$, then continue; otherwise, the protocol is terminated.

Step 7 Alice gets the state:

$$|\alpha'\rangle=\pi(\frac{1}{\sqrt{D}}\sum_{j_i=0}^{D-1}\zeta^{(a_i+r_i)j_i}|j_i\rangle_{h_i}|j_i\rangle_{t_i}), \tag{21}$$

then Alice applies XOR:

$$|j_i\rangle_{h_i}|j_i\rangle_{t_i}\overset{XOR_{(h_i,t_i)}}{\longrightarrow}|j_i\rangle_{h_i}|0\rangle_{t_i}, \tag{22}$$

then performs $QFT^{\dagger}$ on h_i:

$$\pi(\frac{1}{\sqrt{D}}\sum_{j_i=0}^{D-1}\zeta^{(a_i+r_i)j_i}|j_i\rangle_{h_i})\overset{QFT_{h_i}^{\dagger}}{\longrightarrow}\pi(|a_i+r_i\rangle_{h_i}), \tag{23}$$

and measures h_i to obtain the result $\pi(A+R)$.

B. Enhanced Quantum Millionaire Subprotocol

In our protocol, a private comparison function is essential. Building upon Jia's quantum protocol for the millionaire problem [17], we propose an enhanced quantum millionaire subprotocol that includes functionality to determine equality between the participants' comparison data. The core steps are as follows:

Protocol 2: Enhanced quantum millionaire protocol.

Input: $N=2^m$. Alice inputs an n-dim vector $W=(w_1,w_2,\cdots,w_n)\in[N]$; Bob inputs a private vector $S=(s_1,s_2,\cdots,s_n)\in[N]$.

Output: Alice outputs $R_1,R_2,\cdots,R_n$, where each R_i indicates whether $w_i>s_i$.

Step 1 Set $d=m+2$, $D=2^d$, Alice shares a string of random bits $r_1'\cdots r_n'$ with Bob privately where $r_i'\in\{0,1\}$ for $i=1,\cdots,n$;

Step 2 Trent creates a sequence of n three-particle entangled states as

$$|\psi_0\rangle_i=\frac{1}{\sqrt{D}}\sum_{j\in[D]}|j\rangle_{T_i}|j\rangle_{W_i}|j\rangle_{S_i}, \tag{24}$$

where the particles T_i,W_i,S_i correspond to Trent, Alice and Bob. Trent picks out the second and third particles from each state to form an ordered sequence $SW=\{|j\rangle_{W_1},\cdots,|j\rangle_{W_n}\}$ and $SS=\{|j\rangle_{S_1},\cdots,|j\rangle_{S_n}\}$. Then Trent sends them to Alice and Bob respectively;

Step 3 Alice inputs w_i by applying $ROT((-1)^{r_i'}w_i)$ on W_i:

$$|j\rangle_{W_i}\overset{ROT((-1)^{r_i'}w_i)_{W_i}}{\longrightarrow}\zeta^{j(-1)^{r_i'}w_i}|j\rangle_{W_i}, \tag{25}$$

on particle W_i, $\zeta=e^{\frac{2\pi i}{D}}$. Similarly, Bob applies $ROT((-1)^{(r_i'+1)}s_i)$ on S_i:

$$|j\rangle_{S_i}\overset{ROT((-1)^{(r_i'+1)}s_i)_{S_i}}{\longrightarrow}\zeta^{j(-1)^{(r_i'+1)}s_i}|j\rangle_{S_i}. \tag{26}$$

then they send SW, SS back to Trent.

Step 4 Trent gets the state

$$|\psi_{w_i-s_i}\rangle=\frac{1}{\sqrt{D}}\sum_{j\in[D]}\zeta^{j(-1)^{r_i'}(w_i-s_i)}|j\rangle_{T_i}|j\rangle_{W_i}|j\rangle_{S_i}, \tag{27}$$

then Trent applies XOR gate to the state $|\psi_{w_i-s_i}\rangle$:

$$\begin{aligned}&\frac{1}{\sqrt{D}}\sum_{j\in[D]}\zeta^{j(-1)^{r_i'}(w_i-s_i)}|j\rangle_{T_i}|j\rangle_{W_i}|j\rangle_{S_i}\\&\overset{XOR_{(T_i,W_i)},XOR_{(T_i,S_i)}}{\longrightarrow}\\&\frac{1}{\sqrt{D}}\sum_{j\in[D]}\zeta^{j(-1)^{r_i'}(w_i-s_i)}|j\rangle_{T_i}|0\rangle_{W_i}|0\rangle_{S_i},\end{aligned} \tag{28}$$

then performs $QFT^{\dagger}$ on T_i:

$$\begin{aligned}&\frac{1}{\sqrt{D}}\sum_{j\in[D]}\zeta^{j(-1)^{r_i'}(w_i-s_i)}|j\rangle_{T_i}|0\rangle_{W_i}|0\rangle_{S_i}\\&\overset{QFT_{T_i}^{\dagger}}{\longrightarrow}\left|(-1)^{r_i'}(w_i-s_i)\right\rangle_{T_i},\end{aligned} \tag{29}$$

and measures the particle T_i to obtain the result

$$D_i=(-1)^{r_i'}(w_i-s_i)\bmod D. \tag{30}$$

Step 5 Trent records $R_i=0$ if $0<D_i<\frac{m}{2}$, records$R_i=2$ if $D_i=0$, otherwise $R_i=1$, and declares the string $R_1R_2\cdots R_n$ in order.

This protocol differs from the original in the following aspect: In step 4, the extraction of phase information using a measurement operator is replaced by first applying XOR gate for disentanglement, followed by the application of $QFT^{\dagger}$ to extract the phase information, after which the particles are measured. In step 5, the results are categorized into three parts: greater than, less than, and equal. At this point, Alice and Bob can compare w_i and s_i by computing $R_i\oplus r_i'$. If the result is 0, then $w_i>s_i$; if the result is 1, then $w_i<s_i$; otherwise, $w_i=s_i$.

C. Quantum Secure Vector Dominance Protocol

The secure vector dominance problem can be described as follows: For simplicity, we consider that there are three participants, Alice, Bob, and Trent. Alice has an n-dim vector $A=(a_1,a_2,\cdots,a_n)$ and Bob has a vector $B=(b_1,b_2,\cdots,b_n)$. They want to determine if A dominates B (denoted as $A\succ B$), that is, whether $a_i>b_i$ for all $i=1,\cdots,n$. To facilitate this, a non-colluding third party Trent is introduced.

Based on the subprotocols proposed above, we provide the specific process of our protocol as follows.

Protocol 3: Quantum secure vector dominance (QSVD).
Input: $N = 2^m$. Alice inputs a vector $A = (a_1, a_2, \cdots, a_n) \in [N]$ and Bob inputs $B = (b_1, b_2, \cdots, b_n) \in [N]$.
Output: Alice and Bob both get the result that if $A \succ B$.

Step 1 Using a disguise technique:

$$A' = (2a_1, \cdots, 2a_n, (2a_1+1), \cdots, (2a_n+1), \\ -2a_1, \cdots, -2a_n, -(2a_1+1), \cdots, -(2a_n+1)), \tag{31}$$

$$B' = ((2b_1+1), \cdots, (2b_n+1), 2b_1, \cdots, 2b_n, \\ -(2b_1+1), \cdots, -(2b_n+1), -2b_1, \cdots, -2b_n). \tag{32}$$

The purpose of the inputs disguise is to get the same number of $a'_i > b'_i$ situations as that of $a'_i < b'_i$, then $2a_i > 2b_i + 1, (2a_i + 1) > 2b_i, -2a_i < -(2b_i + 1)$, and $-(2a_i + 1) < -2b_i$, which generates two $>$'s, and two $<$'s. At the same time, it avoids the situation that the data of Alice and Bob is the same. If $a = b$, the inputs disguise can get $2a \neq 2b + 1$. After that, Alice gets the disguised input $A' = (a'_1, \cdots, a'_{4n})$, and Bob gets the disguised input $B' = (b'_1, \cdots, b'_{4n})$. Let $V_A = (\overbrace{1, 1, \cdots, 1}^{2n}, \overbrace{0, 0, \cdots, 0}^{2n})$.

Step 2 Alice and Bob execute **Protocol 2**, where Alice inputs $A' = (a'_1, \cdots, a'_{4n})$ and Bob inputs a permutation $\pi(\cdot)$ and a random vector $R = (r_1, r_2, \cdots, r_{4n}) \in [N]$. After the protocol process, Alice outputs

$$A'' = \pi(A' + R),\ a''_i = \pi(a'_i + r_i). \tag{33}$$

Step 3 Bob performs the permutation operation $\pi(\cdot)$ on the vector $B' = (b'_1, \cdots, b'_{4n})$ and V_A, then adds the random vector $R = (r_1, \cdots, r_{4n})$ to B' :

$$b'_i \xrightarrow{\pi(\cdot), R} b'_{\pi(i)} + r_{\pi(i)},\ V_A \xrightarrow{\pi(\cdot)} \pi(V_A), \tag{34}$$

then Bob gets

$$B'' = \pi(B' + R),\ b''_i = \pi(b'_i + r_i), V'_A = \pi(V_A), \tag{35}$$

respectively.

Step 4 To prevent Bob from pretending to perform a permutation operation, Alice adds a random vector $C = (c_1, c_2, \cdots, c_{4n}) \in [N]$ on A'' and sends C to Bob. Bob then adds it on B''.

Step 5 Alice and Bob execute **Protocol 1**, Alice inputs $A'' + C$ and Bob inputs $B'' + C$. After the protocol process, Alice outputs the result $U = (u_1, u_2, \cdots, u_{4n})$, where for $i = 1, \cdots, 4n$, if $a''_i > b''_i$, $u_i = 1$, otherwise $u_i = 0$.

Step 6 Alice and Bob execute **Protocol 1** again to calculate whether U and V'_A are equal. Alice inputs U and Bob inputs V'_A. After the protocol process, Trent outputs the result R_i. If all R_i are equal to 2, that is, $U = V'_A$, Trent declares the result: $A \succ B$. Otherwise, Trent declares the result $A \not\succ B$.

IV. Performance Analysis

A. Correctness

Here we analyze the correctness of the proposed protocols.

In **Protocol 3**, Alice and Bob calculate the result of if $a_i > b_i$ for $i = 1, \cdots, n$, mean that $U = V'_A$. In step 1, after using a disguise technique, Alice gets $A' = (a'_1, \cdots, a'_{4n})$, Bob gets $B' = (b'_1, \cdots, b'_{4n})$. In step 2, Alice and Bob execute **Protocol 1**. In step 5 of **Protocol 1**, Bob performs the same permutation on H and T sent by Alice, the state should be:

$$|\alpha''\rangle = \frac{1}{\sqrt{D}} \sum_{j_i=0}^{D-1} \zeta^{a_i j_i} |j_i\rangle_{h_{\pi(i)}} |j_i\rangle_{t_{\pi(i)}}. \tag{36}$$

After Bob performs $ROT(ri)_{T_i}$ to each particle in the sequence T, Alice gets the state shown in Eq (21). After apply the $QFT^\dagger$, Alice gets $\pi(A' + R) \bmod D$. Throughout the entire protocol process, conducting a meticulous overflow analysis is crucial because potential overflow situations can impact the accuracy of calculations during various operations and transformations. Given that $R \in [N]$ and $A' \in [2N + 1]$, it follows that $\pi(A' + R) \in [3N + 1]$. Since $D = 2^d = 2^{m+2} = 4 \cdot 2^m = 4N$, it implies that $\pi(A' + R) \bmod D = \pi(A' + R)$. The final result is Alice gets $A'' = \pi(A' + R)$ and Bob gets R respectively, While Bob knows the permutation $\pi(\cdot)$. After that, Bob performs the $\pi(\cdot)$ on his vector B' and V_A to obtain $B'' = \pi(B' + R)$ and $V'_A = \pi(V_A + R)$. As a precautionary measure against Bob, Alice adds C and sends it to Bob. In step 5, Alice and Bob execute **Protocol 2**, Alice inputs

$$A'' + C = (a'_{\pi(1)} + c_1, \cdots, a'_{\pi(4n)} + c_{4n}), \tag{37}$$

and Bob inputs

$$B'' + C = (b'_{\pi(1)} + c_1, \cdots, b'_{\pi(4n)} + c_{4n}). \tag{38}$$

They protect their data by pre-sharing a string of random bits $r'_1 r'_2 \cdots r'_{(4n)}$. After executing **Protocol 2**, Alice obtains $U = (u_1, \cdots, u_{4n})$, where:

(1) If $A'' + C > B'' + C$, then $u_i = 1$ (i.e., $a'_{\pi(i)} > b'_{\pi(i)}$);
(2) If $A'' + C < B'' + C$, then $u_i = 0$ (i.e., $a'_{\pi(i)} < b'_{\pi(i)}$).

Let $\pi(U') = U$, meaning $u'_i = 1$ if $a'_i > b'_i$ and $u'_i = 0$ if $a'_i < b'_i$. The result $U = V'_A$ implies $\pi(U') = \pi(V_A)$, where V_A satisfies:

(1) For $i = 1, 2, \cdots, 2n$, $a'_i > b'_i$;
(2) For $i = 2n + 1, \cdots, 4n$, $a'_i < b'_i$.

Due to the disguise technique, this implies that $a_i > b_i$ for all $i = 1, 2, \cdots, 4n$.

B. Security

Here, we analyze the security of the proposed Quantum Secure Vector Dominance (QSVD) protocol, considering various aspects related to information leakage and resistance to malicious behaviors.

(1) **Input Forgery Attack**
In Protocol 3, Alice could input a fraudulent vector to deduce Bob's permutation operation $\pi(\cdot)$. However, Bob adds a random vector R_B to each particle before

sending it to Alice, introducing sufficient randomness to prevent Alice from deducing the permutation. Even if Alice returns the original sequence, the perturbation introduced by R_B makes it infeasible for her to recover Bob's private data. Bob might attempt to falsify the permutation process by keeping Alice's particles in their original order and adding his own random vector. This would prevent Alice from detecting the absence of a permutation operation, allowing a third party, such as Trent, to obtain Alice's unpermuted data. To counter such attacks, Protocol 3 introduces a random vector C in Step 4, requiring Bob to add it. This ensures the uniqueness of each computation, making it impossible for Bob to modify the permutation and accurately determine the final result.

(2) **Measurement Attack**
An attacker might attempt to measure the exchanged quantum particles between Alice and Bob, as represented by the state:

$$\frac{1}{\sqrt{D}}\sum_{j\in[D]}\zeta^{jx}\left|j\right\rangle_t\left|j\right\rangle_h, \tag{39}$$

in an effort to gain private information. However, the security of the phase information in the Fourier entangled state has been proved in Ref. [35], which shows that the specific structure of the quantum systems h and t ensures the resistance of the phase factor x to manipulation. This structure prevents an attacker from extracting phase information from the global density operator, thus preserving the confidentiality of the phase domain. Consequently, if an attacker attempts to measure the quantum particles exchanged between Alice and Bob, the entanglement properties ensure that any measurement on a part of the system would collapse the state. As a result, the measurement would not reveal any private information, making such an attack ineffective.

Furthermore, in Protocol 2, suppose Alice attempts to determine Bob's secret s_i after the phase shifting operations. The particle state $|\psi_0\rangle_i$ becomes $|\psi\rangle_i$, and the reduced density matrix of Alice's subsystem is:

$$\rho^{W_i}=\mathrm{Tr}_{T_iS_i}(\rho^{T_iW_iS_i})=\frac{1}{D}\sum_{n\in\mathbb{Z}_D}\left|n\right\rangle_{W_i}\left\langle n\right|_{W_i}, \tag{40}$$

which remains unchanged under any operation. Therefore, Alice cannot extract information from the particle W_i or learn s_i from the transmitted particle S_i. Only by performing collective measurements on T_i, W_i, and S_i can the secret be obtained. Even if Alice holds both W_i and S_i, no information is exposed, as the reduced density matrix is:

$$\begin{aligned}\rho^{W_iS_i}&=\mathrm{Tr}_{T_iS_i}(\rho^{T_iW_iS_i})\\&=\frac{1}{D}\sum_{n\in\mathbb{Z}_D}\left|n\right\rangle_{W_i}\left|n\right\rangle_{S_i}\left\langle n\right|_{W_i}\left\langle n\right|_{S_i},\end{aligned} \tag{41}$$

which is independent of the phase. Since the particle T_i is not transmitted, Alice cannot perform collective measurements on all three particles.

(3) **Entanglement-Measurement Attack**
In this attack, Bob might prepare an auxiliary particle e, entangle it with the received particle h_i, and attempt to measure it to gain information. However, since Alice does not perform individual measurements and the inverse quantum Fourier transform results are undisclosed, Bob cannot fully decode Alice's private information without access to the entire quantum state.

(4) **Intercept-Resend Attack**
An adversary may intercept and measure the quantum particles, subsequently forging new particles and resending them. In Protocol 1, Alice employs entangled particles, which prevents Bob from decoding her private information. Additionally, Alice verifies the sum of the particles to ensure data integrity, thereby safeguarding against tampering during transmission. The inherent quantum entanglement further ensures that the data remains secure and cannot be altered by an attacker. In Protocol 2, Trent retains the entangled particles T_i, which are neither transmitted nor disclosed, further protecting the integrity of the communication.

(5) **External Attacks**
In the case of an external attack, such as a man-in-the-middle attack where Eve intercepts the quantum particles, Eve cannot extract useful information due to the quantum nature of the particles. Quantum measurements cause state collapse, and the legitimate parties can detect any information leakage. Moreover, quantum encryption and entanglement protection prevent Eve from acquiring private information. Replay attacks are thwarted by Alice's verification steps and Bob's use of random vectors, ensuring the integrity of the information.

Therefore, Protocol 1, Protocol 2, and Protocol 3 ensure security by preventing unauthorized access to the participants' private vector information. As a result, none of the parties involved can extract or infer any private information from the protocol execution.

C. Complexity

Here, we analyze the communication and computational complexities of the proposed protocols. As shown in Fig. 2(e), a QFT (or $QFI^\dagger$) gate can be decomposed into $O(d^2)$ H and $CP(i):\left|a\right\rangle\left|b\right\rangle\rightarrow e^{\frac{\imath 2\pi 2^i}{D}ab}\left|a\right\rangle\left|b\right\rangle$ gates, the ROT gate, represented as $ROT_h=\prod_{i,k\in[d]}\prod(i+k)_{h_i}^{b_i}$, has complexity $O(d^2)$, the XOR gate decomposes into d $CNOT$ gates, resulting in a complexity of $O(d)$ [39]. The MUL gate has complexity $O(d^2)$, as shown by Shor's algorithm [13]. In general, the complexity of the gates in Section II-D is all below $O(d^2)$.

The Quantum Secure Vector Dominance Protocol (QSVD) calculates the dominance relationship between two vectors $A=(a_1,a_2,\ldots,a_n)$ and $B=(b_1,b_2,\ldots,b_n)$ by invoking Protocol 1 and Protocol 2, along with operations such as input vector hiding, encryption, exchange, and computation. First, the protocol disguises the input vectors, with the complexity of the disguise operation primarily determined by data processing and disguise, which is considered a constant-level overhead.

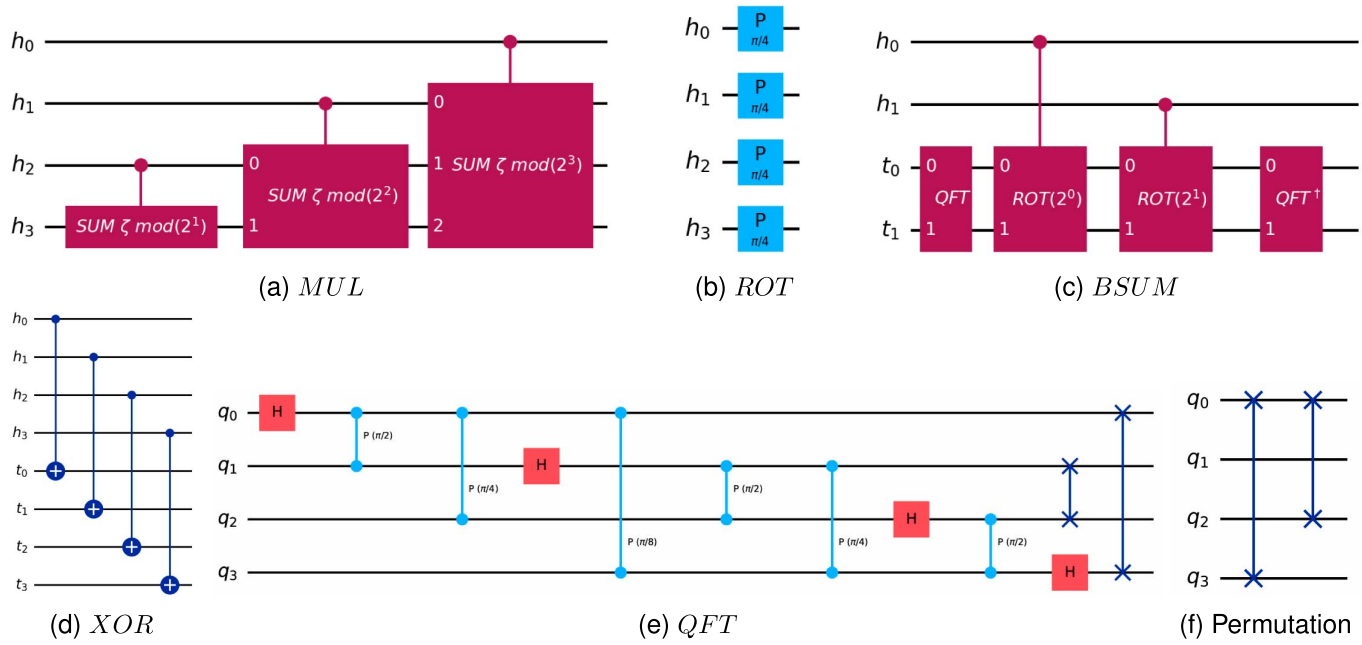


Fig. 2. The circuits of quantum gates. (a)–(f) are circuits of MUL, ROT, $BSUM$, XOR, QFT, and $\pi(\cdot)$ respectively.

Next, Protocol 3 calls Protocol 1 to perform the exchange of input vectors. Protocol 1 involves quantum encoding, $BSUM$, MUL, rotation gates, and other operations. The computational complexity of this step is $O(n)$, while the communication complexity, due to the exchange of quantum bits, is $O(mn)$, where m is the number of quantum bits and n is the vector dimension. Then, Protocol 3 invokes Protocol 2 for vector comparison. Protocol 2 compares the elements of the vectors using quantum rotation gates (ROT), XOR, and $QFT^{\dagger}$. The complexity of this step is $O(n)$, and both computational and communication complexities are linearly related to the input dimension n. In summary, the communication complexity of Protocol 3 is determined by the communication steps in Protocol 1 and Protocol 2, yielding $O(mn)$. The computational complexity is mainly derived from the quantum operations in the protocol, and is $O(n)$.

Thus, the overall computational complexity of the QSVD protocol is $O(mn)$ for communication and $O(n)$ for computation. Consequently, the QSVD protocol exhibits polynomial-time complexity, rendering it appropriate for practical quantum computing applications.

D. Example

Assume that $N = 16$, $n = 3$, Alice has a vector $A = (4, 5, 3)$, Bob has a vector $B = (3, 3, 1)$. The expected result is that $A \succ B$.

(1) In step 1, after performing a disguise technique, we have $A' = (8, 10, 6, 9, 11, 7, -8, -10, -6, -9, -11, -7)$, $B' = (7, 7, 3, 6, 6, 2, -7, -7, -3, -6, -6, -2)$, $V_A = (1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0)$.

(2) In step 2, we set $\pi(\cdot) = (6, 11, 3, 1, 9, 5, 8, 2, 10, 7, 4, 12)$ and $R = (2, 3, 1, 4, 5, 7, 3, 6, 2, 6, 9, 6)$, then Alice and Bob call **Protocol 1** to calculate $A'' = \pi(A') + \pi(R) = (9, -10, 6, -11, 7, 8, -9, -8, 11, -6, 10, -7) + (4, 6, 1, 9, 7, 2, 6, 3, 5, 2, 3, 6) = (13, -4, 7, -2, 14, 10, -3, -5, 16, -4, 13, -1)$.

(3) In step 3, Bob performs $\pi(\cdot)$ on the vector B' and V_A, then $B'' = (10, -1, 4, 3, 9, 9, 0, -4, 11, -1, 10, 4)$, $V'_A = (1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0)$.

(4) In step 4, since Bob has $\pi(\cdot)$, to prevent him from colluding with Trent to obtain the vector sent by Alice, Alice generates a random vector $C = (c_1, \cdots, c_{4n}) = (3, 2, 1, 4, 2, 7, 5, 9, 7, 6, 8, 4)$ and sends it to Bob, allowing both parties to add a layer of security, and finally obtaining $A'' + C = (16, -2, 8, 2, 16, 17, 2, 4, 23, 2, 21, 3)$ and $B'' + C = (13, 1, 5, 7, 11, 16, 5, 5, 18, 5, 18, 8)$.

(5) In step 5-6, Alice and Bob execute **Protocol 2** to output the result $U = (1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0)$, and it is clear that $U = V_A$. Prove that the vector A dominates the vector B, i.e. $A \succ B$.

E. Simulated Experiments

To verify the correctness and feasibility of the protocol, we conducted circuit simulation experiments using the IBM Qiskit simulator (Qiskit-1.2.0; Python-3.9; OS-Windows). This simulation allowed us to model the quantum circuits and test the protocol's performance in a controlled environment. Set $m = 2$ (i.e., $d = 4$). Fig. 2 depicts all quantum gate circuits described in Section II-D. The circuit design of $BSUM$ adopts a Fourier adder, as detailed in [40]. Although its complexity is $O(d^2)$, it does not need to consume quantum auxiliary bits.

The quantum state preparation begins with the application of QFT to each qubit in the h register, creating the superposition

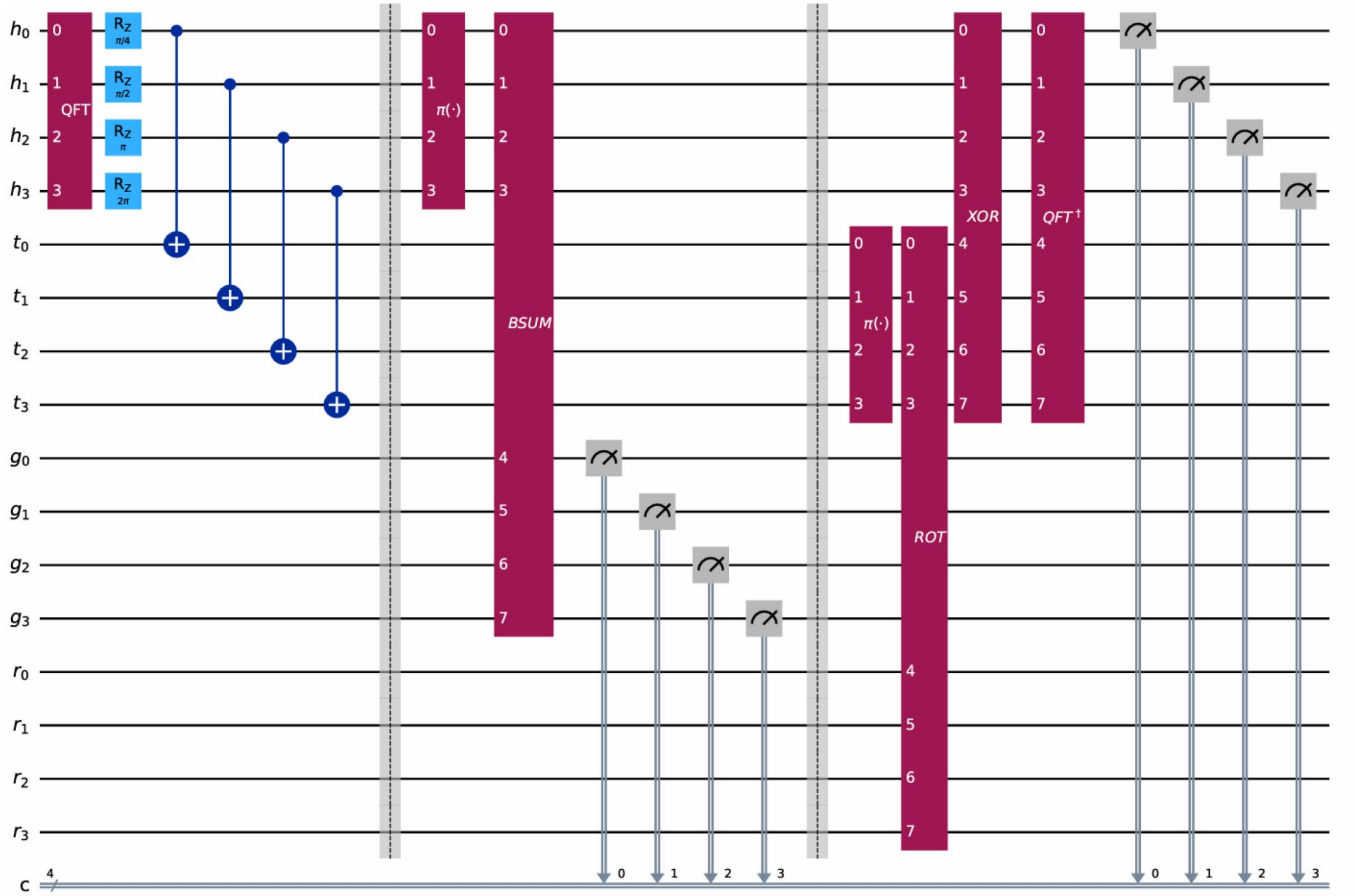


Fig. 3. The circuit of Quantum private permutation subprotocol, where the first part is the quantum state preparation process, the second part is the replacement of h_i particles after the particles are added to the initialization of g particles for subsequent honest testing, the third part is the replacement of t_i particles, and then the random vector R is added in the way of phase shifting, and finally $QFT^{\dagger}$ and measurement operation are carried out.

state $\frac{1}{\sqrt{D}}$. Phase modulation is then applied using the R_z gate with phase factors $\zeta^{a_i j_i}$. Finally, XOR gates are used to entangle each h_i qubit with its corresponding t_i.

Assuming that the input $a_i = 3$, the quantum state preparation process is shown in the first part of Fig. 3. Then we design a permutation circuit diagram, assuming that $\pi(\cdot)$ is (4,2,1,3), the permutation circuit can be designed as shown in Fig. 2(f). Thus the quantum private permutation subprotocol is shown in Fig. 3. Finally, by combining all the proposed protocols, the overall circuit diagram is derived as shown in Fig. 4.

Table II presents the input and output parameters, along with the selection of intermediate parameters r_i, $\pi(i)$ and c_i. The quantum program for $i = 1, 2, 3, 4$ was executed 1024 times, with the results presented in Fig. 5. The outcomes indicate that our protocol operated flawlessly with a 100% success rate, validating its accuracy and reliability.

V. Applications

In this section, we present some applications of **Protocol 3** in several PGI problems, i.e., point-line inclusion determination, line-line intersect determination, and point-in-polygon determination.

A. Point-Line Inclusion Determination Protocol

Definition 1: (Point-Line inclusion determination).
Input: $N = 2^m$. Alice inputs a point $P = (m, n)$, Bob inputs two point information from the line, namely $W = (x_0, y_0)$ and $Q = (x_1, y_1)$.
Output: Alice and Bob both get the result of whether the point is on the line.

When a point is not on a line, the cross product of vectors $\overrightarrow{WP}$ and $\overrightarrow{WQ}$ can be computed. If the cross product is non-zero, the point P is not on the line. For example, the vectors $\overrightarrow{WP} = (x_0 - m, y_0 - n)$ and $\overrightarrow{WQ} = (x_1 - x_0, y_1 - y_0)$ give the following cross product:

$$\begin{aligned}\overrightarrow{WP} \times \overrightarrow{WQ} &= (x_0 - m)(y_1 - y_0) - (y_0 - n)(y_1 - y_0) \\ &= m(y_1 - y_0) + n(x_0 - x_1) + (x_1 y_0 - x_0 y_1), \end{aligned} \tag{42}$$

which can be expressed as a scalar product of two vectors. In protocol design, these results can be used to construct the vector components and determine the intersection of points and lines through vector domination outcomes.

Protocol 4: Quamtum secure point-line inclusion determination (QSPLLD).

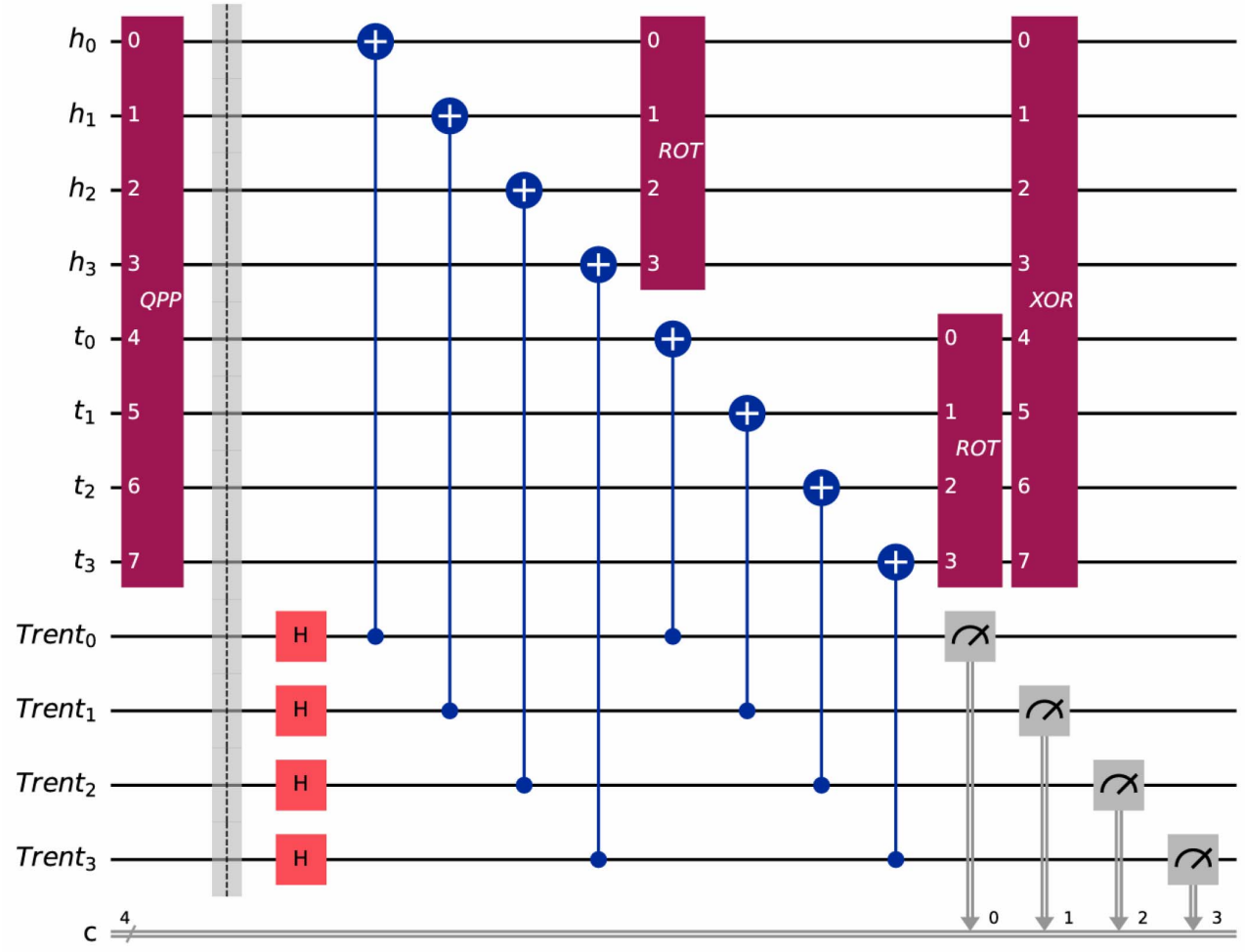


Fig. 4. The total circuit of quantum secure vector dominance.

TABLE II
PARAMETER OF THE EXPERIMENT

External Parameter		Input				Intermediate Parameter					Output	
m	N	i	a'_i	b'_i	V_A	d	D	r_i	$\pi(i)$	c_i	V'_A	U
2	4	1	4	3	1	4	16	2	3	3	0	0
		2	5	2	1			1	2	1	1	1
		3	-4	-3	0			3	1	2	1	1
		4	-5	-2	0			1	4	3	0	0

Step 1 Alice computes her vector $A_4 = (m, n, 1)$, Bob computes his vector $B_4 = (y_1 - y_0, x_0 - x_1, x_1y_0 - x_0y_1)$.

Step 2 Alice and Bob execute a quantum secure two-party scalar product (QS2PSP) protocol [35] to get results of $A_4 \cdot B_4$.

Step 3 After all the steps of QS2PSP are completed, Alice can obtain the result

$$A_4 \cdot B_4 = m(y_1 - y_0) + n(x_0 - x_1) + (x_1y_0 - x_0y_1) + r. \tag{43}$$

Bob can obtain the random vector r generated by himself.

Step 4 Then they execute **Protocol 3** to completed vector a and r.

Step 5 After all the steps of the protocol are completed, Alice gets the result if the point is on the line.

Analysis:

(1) **Correctness:** In **Protocol 4**, Alice and Bob calculate the result of if $a \succ r$. That is

$$m(y_1 - y_0) + n(x_0 - x_1) + (x_1y_0 - x_0y_1) > 0 \rightarrow \overrightarrow{WP} \times \overrightarrow{WQ} > 0. \tag{44}$$

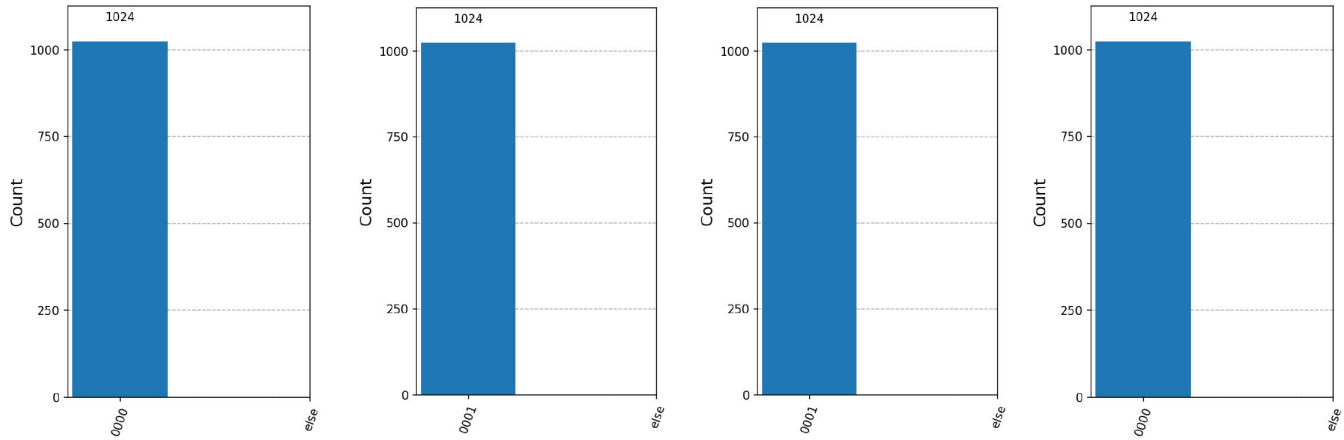


Fig. 5. The results U for $i = 1, 2, 3, 4$ of the experiment.

(2) **Security:** Since QS2PSP and **Protocol 3** are secure, then any party cannot learn more information other than their vector.
(3) **Complexity:** Alice and Bob call QS2PSP once, with computational and communication complexity $O(m^2)$ and $O(m)$ respectively. They also call **Protocol 3** once, with computational and communication complexity $O(n)$ and $O(mn)$ respectively. Therefore the communication complexity and computational complexity of **Protocol 4** are $O(mn)$ and $O(m^2)$ respectively.

B. Line-Line Intersect Determination Protocol

Definition 2: (Line-Line intersect determination (LLID)).
Input: $N = 2^m$. Alice inputs a line $y_A = k_A x + b_A$, Bob inputs a line $y_B = k_B x + b_B$.
Output: Alice and Bob both get the result of whether the two lines intersect.

To determine if two lines intersect, compare their slopes k_A and k_B. If they intersect at a negative abscissa, they do not intersect at positive abscissas. Trent can select an abscissa z and compare the ordinates at this point. If one line's slope and ordinate are greater, the lines do not intersect in the calculation area.

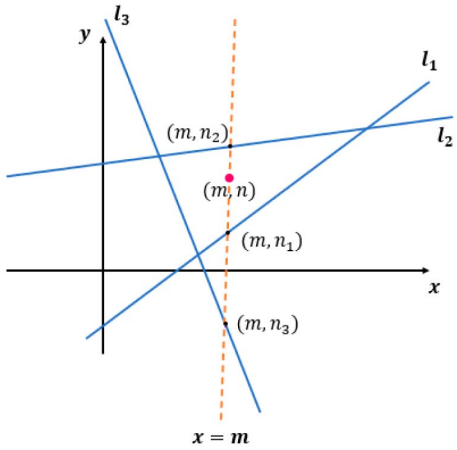
Protocol 5: Quamtum secure line-line intersect determination (QSLLID).
Step 1 Trent announces a number z. Alice computes $k_A z + b_A$ and prepares her vector $A = (k_A z + b_A, k_A)$, Bob computes $k_B z + b_B$ and prepares her vector $B = (k_B z + b_B, k_B)$;
Step 2 Then they execute Quantum secure vector dominance protocol to completed vector A and B;
Step 3 After all the steps of **Protocol 3** are completed, Trent announces the result of Whether the lines intersect.
Analysis:
(1) **Correctness:** In **Protocol 5**, Alice and Bob calculate the result of if $(k_A z + b_A, k_A) \succ (k_B z + b_B, k_B)$. That is, the transverse coordinates of the intersection of line y_A and $x = z$ are greater than the transverse coordinates of the intersection of line y_B, and the slope of line y_A is greater than that of line y_B. It is expressed as two lines that do not intersect after crossing the line $x = z$.
(2) **Security:** The security of **Protocol 5** is based on **Protocol 3**, which was analyzed in Section IV-B
(3) **Complexity:** Alice and Bob call **Protocol 3** once, with computational and communication complexity $O(n)$ and $O(mn)$ respectively. Therefore the communication complexity and computational complexity of **Protocol 5** are $O(mn)$ and $O(n)$ respectively.

C. Point-in-Polygon Determination Protocol

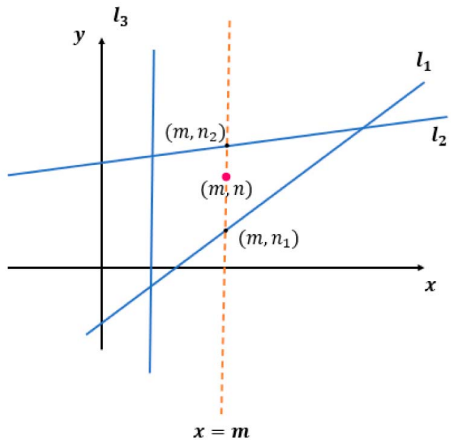
First, we provide a precise definition of the problem.
Definition 3: (Point-in-Polygon determination (PIPD)).
Input: $N = 2^m$. Alice inputs a point $A = (m, n)$, Bob inputs line information for its polygon with p sides: $l_1 : y = k_1 x + b_1$, $l_2 : y = k_2 x + b_2$, $l_3 : y = k_x + b_3, \cdots, l_p : y = k_p x + b_p$.
Output: Alice and Bob both get the result of whether the point is inside the polygon. In addition, each party cannot learn the exact position of the other party's point.

To study the problem of the positional relationship between points and polygons, we should first study the positional relationship between points and triangles(illustrated in Fig. 6), and then extend it to the case of polygons. Taking the problem of determining the point inclusion of a triangle as an example, we use the following analysis method:

The triangle is enclosed by the image of these three function expressions of degree one. Specifically, a triangle is enclosed by an image of the inequality group $l_1 : y = k_1 x + b_1$, $l_2 : y = k_2 x + b_2$, $l_3 : y = k_3 x + b_3$ in a planar rectangular coordinate system. It can be seen from Fig. 6(a) that the straight $x = m$ intersects the line l_1, l_2, l_3, denoted $(m, n_1), (m, n_2), (m, n_3)$ in turn, and according to Fig. 6(a), it is clear that the point is



(a) Intersection of a point with three triangle edges.



(b) Intersection of a point with two triangle edges.

Fig. 6. Two cases where the point is inside the triangle.

inside the triangle only under the following conditions.

$$\begin{cases} n_2 - n > 0 \\ n - n_1 > 0 \\ n - n_3 > 0 \end{cases} \tag{45}$$

If one of the sides of the triangle is perpendicular to the X-axis according to the Fig. 6(b). Assume the triangle is enclosed by an image of the inequality group $l_1 : y = k_1 x + b_1$, $l_2 : y = k_2 x + b_2$, $l_3 : x_{y_3} = n_3$ in a planar rectangular coordinate system. The point is inside the triangle only under the following conditions.

$$\begin{cases} n_2 - n > 0 \\ n - n_1 > 0 \\ m - n_3 > 0 \end{cases} \tag{46}$$

Protocol 6: Quamtum secure point-in-triangle determination (QSPITD).

Step 1 Alice computes her vector $A = (m, 1, n)$, Bob computes his vector $B_1 = (-k_1, -b_1, 1), B_2 = (k_2, b_2, -1), B_3 = (-k_3, -b_3, 1)$;

Step 2 Alice and Bob execute a quantum scalar product protocol [35] to get results of $A \cdot B_1, A \cdot B_2, A \cdot B_3$;

Step 3 After all the steps of quantum scalar product protocol are completed, Alice can obtain three results

$$\begin{cases} x_1 = (m, 1, n) \cdot (-k_1, -b_1, 1) = -k_1 m + n - b_1 + v_1 \\ x_2 = (m, 1, n) \cdot (k_2, b_2, -1) = k_2 m - n + b_2 + v_2 \\ x_3 = (m, 1, n) \cdot (-k_3, -b_3, 1) = -k_3 m + n - b_3 + v_3, \end{cases} \tag{47}$$

Bob can obtain the random vector $V = (v_1, v_2, v_3)$ generated by himself;

Step 4 Then they execute Quantum secure vector dominance **Protocol 3** to completed vector $X = (x_1, x_2, x_3)$ and $V = (v_1, v_2, v_3)$;

Step 5 After all the steps of Protocol are completed, Alice gets the result if $X = (x_1, x_2, x_3) \succ V = (v_1, v_2, v_3)$.

Analysis:

(1) **Correctness:** In **Protocol 6**, Alice and Bob calculate the result of if $X = (x_1, x_2, x_3) \succ V = (v_1, v_2, v_3)$. That satisfies the condition

$$\begin{cases} x_1 = -k_1 m + n - b_1 + v_1 > v_1 \\ x_2 = k_2 m - n + b_2 + v_2 > v_2 \\ x_3 = -k_3 m + n - b_3 + v_3 > v_3, \end{cases} \tag{48}$$

which means

$$\begin{cases} -k_1 m + n - b_1 = n - n_1 > 0 \\ k_2 m - n + b_2 = n_2 - n > 0 \\ -k_3 m + n - b_3 = n - n_3 > 0. \end{cases} \tag{49}$$

Therefore, the protocol correctly gives the determination result of the inclusion of points in the triangle.

(2) **Security:** For each $1 \le i \le 3$, Alice and Bob execute one-time QS2PSP protocol. Then Alice gets $Output_i = A \cdot B_i + v_i$, Bob only gets v_i. Since QS2PSP is secure enough with high probability, the result Alice can only obtain is $Output_i$. Since $Output_i$ is an input to the **Protocol 3**, that is, it acts as a whole, it does not provide any more information than Alice deserves. Similarly, Bob is unable to receive any improper benefits. Therefore, **Protocol 6** is at least as secure as **Protocol 3**.

(3) **Complexity:** The two participants call QS2PSP, with computational and communication complexity $O(m^2)$ and $O(m)$ respectively. They also call **Protocol 3** once, with computational and communication complexity $O(n)$ and $O(mn)$ respectively. Therefore the communication complexity and computational complexity of **Protocol 4** are $O(mn)$ and $O(m^2)$ respectively.

VI. Quantum Error Correction and Error Mitigation

In practical quantum communication, protocols are inevitably affected by noise, such as bit-flip, phase-flip, and decoherence. To mitigate these challenges, we propose an integrated approach with two key components:

Quantum Error Correction (QEC) employs techniques such as surface codes, CSS codes, and multi-hypercube codes to detect and correct errors at the physical qubit level. Recent research, including the Google team's Willow chip, a 105-qubit superconducting processor using surface codes, has shown significant improvements in logical error suppression and real-time decoding [41]. Error Mitigation (EM) incorporates methods like Zero-Noise Extrapolation (ZNE), Probabilistic Error Cancellation (PEC), and Randomized Compiling (RC) to further reduce residual noise. RC, in particular, transforms complex noise channels into stochastic Pauli noise, providing a more manageable error model and aiding post-processing [42].

A. Surface Code Encoding and Error Correction

In the proposed QSVD protocol, the initial quantum states are prepared as:

$$|\alpha\rangle = \frac{1}{\sqrt{D}} \sum_{j_i=0}^{D-1} \zeta^{a_i j_i} |j_i\rangle_{h_i} |j_i\rangle_{t_i}, \tag{50}$$

where $\zeta^{a_i j_i}$ is a phase factor based on Alice's input. To enhance robustness during transmission, these states are encoded using a distance-3 surface code, mapping each logical qubit to 9 physical qubits. Data qubits store the computational information, while syndrome qubits detect bit-flip and phase-flip errors.

We periodically measure the stabilizer operators S_X and S_Z to detect errors:

$$S_X = (-1)^{m_X}, \quad S_Z = (-1)^{m_Z}, \tag{51}$$

where m_X and m_Z represent bit-flip and phase-flip error syndromes, respectively. Upon detecting an error, a corresponding X or Z operation is applied. To minimize logical error rates, we use the Minimum-Weight Perfect Matching (MWPM) algorithm to correct the most likely error patterns.

B. Randomized Compiling and Encoded Quantum State Transmission

In addition to QEC, we apply Randomized Compiling (RC) to reduce residual noise. RC inserts Pauli gates T_k and their corrections T_{k-1}^c around each quantum operation C_k in the decomposed QSVD unitary:

$$U = G_K C_K \cdots G_1 C_1, \tag{52}$$

such that

$$C'_k = T_k \, C_k \, T_{k-1}^c. \tag{53}$$

This randomization transforms the noise channels into a stochastic Pauli model, simplifying error estimation and enabling post-processing mitigation. Combined with other EM strategies, RC further reduces error rates beyond what QEC alone can achieve.

After applying surface code encoding and RC, the final quantum state is:

$$|\bar{\alpha}\rangle = \frac{1}{\sqrt{D}} \sum_{j_i=0}^{D-1} \zeta^{a_i j_i} |\bar{j_i}\rangle_{h_i} |\bar{j_i}\rangle_{t_i}, \tag{54}$$

where each logical state $|\bar{j_i}\rangle$ is encoded across multiple physical qubits using the surface code. This encoding preserves quantum information necessary for QSVD, even under bit-flip and phase-flip errors, ensuring more reliable computations in noisy environments.

By integrating surface codes for QEC and RC for EM, our approach enhances fault tolerance in practical quantum systems. These methods not only lower error rates but also improve the feasibility of secure vector dominance computations in noisy settings.

VII. Conclusion

In this paper, we propose a novel Quantum Secure Vector Dominance (QSVD) protocol, enabling two participants to determine vector dominance without revealing their private data. The protocol is built upon a quantum private permutation subprotocol and an enhanced quantum millionaire subprotocol, ensuring both efficiency and security. We analyze its correctness, security, and complexity, validate its feasibility on IBM's Qiskit platform, and demonstrate its scalability through applications in geometric computations. To address noise-related challenges in quantum communication, we propose an integrated approach that combines Quantum Error Correction (QEC) with Error Mitigation (EM), enhancing the protocol's reliability in noisy environments.

While the proposed QSVD protocol represents a significant advancement in secure quantum computation, several challenges and limitations remain that require further exploration.

First, scaling QSVD to multi-party and distributed quantum systems remains a challenge. While the protocol currently supports two-party computations, real-world applications require extending it to multiple parties. Ensuring security, privacy, and scalability in such settings necessitates further exploration of distributed quantum computing models and multi-party secure computation protocols.

Second, advancing quantum error correction is vital for improving fault tolerance. While surface codes show promise, exploring alternatives like concatenated codes, color codes, or topological codes could offer better trade-offs in efficiency, fault tolerance, and error rates. Identifying the most suitable QEC technique for QSVD is crucial for enhancing performance in noisy environments.

Third, effective noise characterization and modeling are essential for optimizing protocol robustness. While QEC and EM techniques improve noise resilience, further research is needed to model the diverse noise types affecting quantum systems. Developing accurate noise models will facilitate more efficient

noise mitigation and better integration with error correction methods.

In summary, the QSVD protocol lays the groundwork for secure vector dominance in quantum computing. Overcoming challenges in multi-party scalability, error correction, and noise modeling is crucial for its practical deployment. Future research will focus on overcoming these challenges, optimizing the protocol for real-world applications, and exploring new avenues for extending its functionality to more complex secure computation problems.

REFERENCES

[1] A. C. Yao, "Protocols for secure computations," in *Proc. 23rd Annu. Symp. Found. Comput. Sci. (SFCS)*. Chicago, IL, USA: IEEE, 1982, pp. 160–164.

[2] H. Y. Shu, R. S. Yu, W. Y. Jiang, and W. X. Yang, "Efficient implementation of *k*-nearest neighbor classifier using vote count circuit," *IEEE Trans. Circuits Syst. II, Exp. Briefs.*, vol. 61, no. 6, pp. 448–452, Jun. 2014.

[3] Y. Q. Gu and Z. W. Shen, "An accountable anonymous voting scheme based on one-time ring signature," in *Proc. Int. Conf. Front. Electron. Inf. Comput. Technol. FEICT*. Piscataway, NJ, USA: IEEE Press, 2023, pp. 51–57.

[4] L. F. Wei et al., "Efficient multi-party private set intersection protocols for large participants and small sets," *Comput. Stand. Interfaces*, vol. 87, 2024, Art. no. 103764.

[5] S. Y. Lv et al., "New approach for efficient malicious multiparty private set intersection," *Inf. Sci.*, vol. 678, 2024, Art. no. 120995.

[6] G. Xu, D. L. Kong, and K. Zhang, "A model value transfer incentive mechanism for federated learning with smart contracts in AIOT," *IEEE Internet Things J.*, vol. 12, no. 3, pp. 2530–2544, 2025.

[7] Y. L. Luo, L. S. Huang, and W. W. Jing, "Privacy-preserving cross product protocol and its application," *Chin. J. Comput.*, vol. 30, no. 2, pp. 248–254, 2007.

[8] L. Chen and B. G. Lin, "Privacy-preserving point-inclusion two-party computation protocol," in *Proc. Int. Conf. Comput. Inf. Sci.*, 2013, pp. 257–260.

[9] M. Ciampi, R. Ostrovsky, H. Waldner, and V. Zikas, "Round-optimal and communication-efficient multiparty computation," in *Proc. Adv. Cryptol. (EUROCRYPT)*, Lecture Notes in Computer Science, vol. 13275. New York: Springer, 2022, pp. 65–95.

[10] A. Goel, M. H. Andersen, A. Hegde, and A. Jain, "Secure multiparty computation with free branching," in *Proc. Adv. Cryptol. (EUROCRYPT)*, Lecture Notes in Computer Science, vol. 13275. New Year: Springer, 2022, pp. 397–426.

[11] M. J. Atallah and W. L. Du, "Secure multi-party computational geometry," in *Proc. WADS2001: Workshop Algorithms Data Structures*. New York: Springer, 2001, pp. 165–179.

[12] C. Hazay and T. Toft, "Computationally secure pattern matching in the presence of malicious adversaries," *J. Cryptol.*, vol. 27, pp. 358–395, 2014.

[13] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.

[14] F. Arute et al., "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[15] H. Kwon and J. Bae, "A hybrid quantum-classical approach to mitigating measurement errors in quantum algorithms," *IEEE Trans. Comput.*, vol. 70, no. 9, pp. 1401–1411, 2021.

[16] R. H. Shi and Y. F. Li, "Quantum secret permutating protocol," *IEEE Trans. Comput.*, vol. 72, no. 5, pp. 1223–1235, May 2023.

[17] H. Y. Jia, Q. Y. Wen, T. T. Song, and F. Gao, "Quantum protocol for millionaire problem," *Opt. Commun.*, vol. 284, no. 1, pp. 545–549, 2011.

[18] W. J. Liu, C. Liu, H. B. Wang, J. F. Liu, F. Wang, and X. M. Yuan, "Secure quantum private comparison of equality based on asymmetric W state," *Int. J. Theor. Phys.*, vol. 53, no. 6, pp. 1804–1813, 2014.

[19] J. Y. Lian, X. Li, and T. Y. Ye, "Multi-party semiquantum private comparison of size relationship with d-dimensional bell states," *EPJ Quantum Technol.*, vol. 10, no. 1, 2023, Art. no. 10.

[20] J. A. Vaccaro, J. Spring, and A. Chefles, "Quantum protocols for anonymous voting and surveying," *Phys. Rev. A.*, vol. 75, no. 1, 2007, Art. no. 012333.

[21] X. T. Xu, R. H. Shi, and W. Y. Ke, "Decentralized quantum anonymous veto voting scheme based on measurement-device-independence," *Phys. Scr.*, vol. 98, no. 9, 2023, Art. no. 095116.

[22] R. H. Shi, "Quantum sealed-bid auction without a trusted third party," *IEEE Trans. Circuits Syst. I, Regul. Pap.*, vol. 68, no. 10, pp. 4221–4231, 2021.

[23] W.-J. Liu, H.-B. Wang, and G.-L. Yuan, "Multiparty quantum sealed-bid auction using single photons as message carrier," *Quantum Inf. Process.*, vol. 15, no. 2, pp. 869–879, 2016.

[24] R. H. Shi, "Anonymous quantum sealed-bid auction," *IEEE Trans. Circuits Syst. II, Exp. Briefs.*, vol. 69, no. 2, pp. 414–418, Feb. 2022.

[25] C. Y. Wei, X. Q. Cai, B. Liu, T. Y. Wang, and F. Gao, "A generic construction of quantum-oblivious-key-transfer-based private query with ideal database security and zero failure," *IEEE Trans. Comput.*, vol. 67, no. 1, pp. 2–8, Jan. 2018.

[26] W. J. Liu, Q. Yang, and Z. X. Li, "Quantum multi-party private set union protocol based on least common multiple and shor's algorithm," *Int. J. Quantum Inf.*, vol. 21, no. 7, 2023, Art. no. 2340006.

[27] Z. X. Li, W. J. Liu, and B. M. Su, "Efficient quantum secure multi-party greatest common divisor protocol and its applications in private set operations," *EPJ Quantum Technol.*, vol. 11, no. 57, 2024.

[28] R. H. Shi and Y. F. Li, "Quantum private set intersection cardinality protocol with application to privacy-preserving condition query," *IEEE Trans. Circuits Syst. I, Regul. Pap.*, vol. 69, no. 6, pp. 2399–2411, Jun. 2022.

[29] G. Xu, S. Xu, and Y. Cao, "Aaq-peks: An attribute-based anti-quantum public key encryption scheme with keyword search for e-healthcare scenarios," *Peer-to-Peer Netw. Appl.*, vol. 18, no. 2, 2025, Art. no. 64.

[30] R. H. Shi, Y. Mu, H. Zhong, J. Cui, and S. Zhang, "Privacy-preserving point-inclusion protocol for an arbitrary area based on phase-encoded quantum private query," *Quantum Inf. Process.*, vol. 16, pp. 1–9, 2017.

[31] Z. W. Peng, R. H. Shi, H. Zhong, J. Cui, and S. Zhang, "A novel quantum scheme for secure two-party distance computation," *Quantum Inf. Process.*, vol. 16, pp. 1–12, 2017.

[32] X. Liu, X. M. Liu, R. L. Zhang, D. Luo, G. Xu, and X. B. Chen, "Securely computing the Manhattan distance under the malicious model and its applications," *Appl. Sci.*, vol. 12, no. 22, 2022, Art. no. 11705.

[33] W. J. Liu, Y. Xu, J. C. N. Yang, W. B. Yu, and L. H. Chi, "Privacy-preserving quantum two-party geometric intersection," *Comput. Mater. Contin.*, vol. 60, no. 3, pp. 1237–1250, 2019.

[34] Z. X. Li, Q. Yang, B. Feng, and W. J. Liu, "Quantum privacy-preserving two-party circle intersection protocol based on phase-encoded query," *Int. J. Theor. Phys.*, vol. 62, no. 7, 2023, Art. no. 138.

[35] W. J. Liu and Z. X. Li, "Secure and efficient two-party quantum scalar product protocol with application to privacy-preserving matrix multiplication," *IEEE Trans. Circuits Syst. I, Regul. Pap.*, vol. 70, no. 11, pp. 4456–4469, Nov. 2023.

[36] Z. Dou, Y. F. Wang, Z. Q. Liu, J. G. Bi, X. B. Chen, and L. X. Li, "Quantum secure multi-party computational geometry based on multi-party summation and multiplication," *Quantum Sci. Technol.*, vol. 9, no. 2, 2024, Art. no. 025023.

[37] J. Yuan, Q. S. Ye, H. X. Wang, and J. Pieprzyk, "Secure computation of the vector dominance problem," in *Proc. 4th Int. Conf. Inf. Secur. Pract. Exp. (ISPEC)*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 319–333.

[38] S. D. Li, X. J. Zuo, X. L. Yang, and L. M. Gong, "Secure vector dominance protocol and its applications," *Acta Electron. Sin.*, vol. 45, no. 5, pp. 1117–1123, 2017.

[39] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. New York: Cambridge Univ. Press, 2010.

[40] T. G. Draper, "Addition on a quantum computer," 2000. *arXiv:0008033*. [Online]. Available: https://arxiv.org/abs/quant-ph/0008033

[41] R. Acharya, D. A. Abanin, and L. Aghababaie-Beni, "Quantum error correction below the surface code threshold," *Nature*, vol. 638, p. 920–926, Feb. 2025.

[42] Y. W. Gu, Y. H. Ma, N. Forcellini, and D. E. Liu, "Noise-resilient phase estimation with randomized compiling," *Phys. Rev. Lett.*, vol. 130, Jun. 2023, Art. no. 250601.

Wenjie Liu received the B.S. degree from Huazhong Agricultural University, in 2001, the M.S. degree from Wuhan University, in 2004, and the Ph.D. degree from the Southeast University, in 2011. Currently, he is a Professor with the School of Software, Nanjing University of Information Science and Technology. His main research interests include quantum secure multiparty computation, quantum machine learning, knowledge graphs, and graph neural networks.

Bingmei Su received the B.S. degree from Nanjing University of Information Science and Technology, in 2022. She is currently working toward the M.S. degree with the School of Computer Science, Nanjing University of Information Science and Technology. Her main research interests include quantum secure multiparty computation and blind quantum computation.

Feiyang Sun received the B.S. degree from Nanhang Jingcheng College, in 2023. He is currently working toward the M.S. degree with the School of Software, Nanjing University of Information Science and Technology. His main research interests include federated learning and quantum secure multiparty computation.