Song et al. *EPJ Quantum Technology* (2026) 13:12
https://doi.org/10.1140/epjqt/s40507-025-00356-z

EPJ.org

EPJ Quantum Technology

RESEARCH Open Access

Quantum secure multiparty multiplication based on d-level single particles

Xiu-Li Song[1,2*], Jie Yan[1], You-Sheng Zhou[2] and Tao Wu[2]

*Correspondence:
songxl@cqupt.edu.cn
[1]College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China
[2]School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China

Abstract

The quantum secure multiparty multiplication (QSMM) protocol aims to leverage the advantages of quantum computing to ensure data privacy and security in multiparty computation, preventing information from being leaked and protecting privacy against malicious attacks. However, current QSMM protocols suffer from high resource consumption and computation complexity in particle preparation and participant computation stages. In view of this, a QSMM protocol based on single particles is proposed in this paper. The proposed protocol utilizes high-dimensional single particles as information carriers, and the optimized quantum multiplication circuit is employed to embed the participants' secret message and perform the multiplication. All participants collaboratively generate a blind matrix, and each column product of the blind matrix acts as the participant's private key for the blinding of the secret value. Unlike other protocols that use decoy particles for eavesdropping detection, the proposed protocol leverages the properties of mutually unbiased basis particles to ensure its security, while reducing quantum resource consumption and the quantum capabilities required from the participants. Security analysis demonstrates that the proposed protocol can effectively resist attacks from external eavesdroppers and internal participants, and performance analysis shows that the proposed protocol achieves superior execution efficiency compared to other similar protocols.

Keywords: Quantum Secure Multiparty Multiplication; Unbiased Basis Particles; Quantum Circuit

1 Introduction

Classical secure multiparty computation is a significant part of classical cryptography, allowing mutually distrusting parties to collaborate and compute a target function using their private data while safeguarding the privacy of all participants [1]. The security of classical secure multiparty computation protocols [2–4] relies on computationally difficult assumptions, such as large integer factorization or the discrete logarithm problem. However, rapidly advancing quantum computation has the potential to break these mathematical challenges [5–8] by leveraging quantum parallelism and entanglement, enabling them to solve problems that are intractable for classical computers within a shorter time frame. As a result, these protocols are vulnerable to quantum attacks. An improvement strategy is to incorporate quantum technology directly into cryptographic protocol design,

© The Author(s) 2025. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

Springer

leading to Quantum Secure Multiparty Computation (QSMC) [9–15]. Quantum Secure Multiparty Multiplication (QSMM) is a critical component of QSMC [16–18], with clear applications in quantum key agreement [19, 20], quantum private comparison [21, 22], and quantum anonymous voting and elections [23–25], highlighting its broad research significance.

In 2016, Shi et al. [26] proposed a QSMM protocol based on the Quantum Fourier Transform (QFT), Controlled NOT gate (CNOT), Oracle operators, and Inverse Quantum Fourier Transform (IQFT), marking the beginning of QSMM research in the quantum cryptography field. In this protocol, the initiator applies QFT and CNOT on multiple particles to prepare an entangled state, all participants encode their data onto the quantum state by executing Oracle operators, and the initiator finally disentangles the state by performing CNOT, applies IQFT, and measures the quantum state in the computational basis to obtain the product result. In 2019, Lv et al. [27] introduced a protocol using mutually unbiased bases as communication particles, where the product of all secret integers is achieved by repeated exponentiation and a single prime product, with each participant inserting, measuring, and removing decoy particles in order of execution to ensure transmission security.

In 2020, Sutradhar et al. [28] proposed a two-party multiplication protocol based on a (t, n) threshold scheme, in which the two participants distribute share values of their secret integers to auxiliary participants based on the Shamir secret sharing scheme. The initiator prepares an entangled state of t particles, distributing each particle to $t-1$ auxiliary participants. Each auxiliary participant then applies QFT and Pauli operators to the particles, and measures, and broadcasts his result to complete the multiplication. The following year, they used the method of [26], improved the protocol [28], and then designed the protocol in [29]. The initiator prepares the entangled state of two particles, and the particles transmission mode changes from tree to ring. Each auxiliary participant embeds his secret shares into the phase using an Oracle operator. Finally, the initiator performs the IQFT and measures the result to obtain the product. The protocol reduces the number of QFT and SUM gate operations but increases communication cost, while still being limited to two-party computation.

In 2022, Zhang et al. [30] proposed a QSMM protocol based on the Lagrange unitary operator. This protocol employs mutually unbiased bases as message particles, transmitted in a tree transmission mode between the server and the participant. Each participant sets the Lagrange unitary operator angle values to embed secret integers and private keys within the particles. The server detects eavesdropping by verifying hash values and subsequently measures the particle's state to obtain the product result. Additionally, a representative participant is needed to collect random numbers from all participants. In the same year, Li et al. [31] proposed a dynamic multiplication protocol that does not require restarting when new participants join or previous participants leave the computation. The protocol employs a mutually unbiased basis particle to transfer secret values and uses decoy particles to detect eavesdropping in a ring transmission. Each participant sends random numbers and unique identity values to the server. In the result announcement process, the product result can only be recovered if at least t honest participants are verified via hash values. The protocol allows the dynamic addition or removal of participants by changing the next recipient of the particles in the computation process.

In 2024, Lian et al. [32] proposed a semi-quantum multiparty secure computation protocol. This protocol involves two semi-trusted third parties (TPs) and several participants with only semi-quantum capabilities, meaning they can only perform quantum state transmission or reflection, Z-basis measurement, and quantum state preparation. In the protocol, each participant's private data has a length of L, and each TP needs to prepare at least $8L$ single particles for them. Upon receiving the particles, participants either measure and re-prepare the quantum states or directly reflect them back to the TP, while informing the TP of their operations. The TP detects eavesdropping using the reflected states and extracts the shared private key with the participant from the first L quantum states re-prepared after measurement. Each participant then multiplies their private key by their secret integer and transmits the result. Finally, the TP computes the quotient of the product and the private key to obtain the multiplication result of all participants' secret integers.

Analyzing the existing fully-quantum QSMM protocols,, we can find that the protocols in [26, 28, 29], which rely on entangled states, are difficult to prepare and complicated to execute. Protocols in [27, 31], while utilizing single particles, ensure security among participants or between the participant and the server by introducing decoy particles. Each decoy particle is selected from a set of bases, such as the computational, orthogonal, Fourier, or mutually unbiased bases. Since the eavesdropper is unaware of the correct measurement basis of the particles, any randomly chosen measurement basis may produce incorrect results. The probability of detection increases with the number of decoy particles. However, the need for each participant to prepare particles, along with sufficient decoy particles, raises both the quantum capability requirements for participants and the protocol's resource consumption. In the measurement stage of [30], it is necessary to apply Lagrange unitary operators to all particles transmitted by each participant to form multiple measurement bases. These additional operations result in increased computation costs.

To overcome these limitations, a quantum secure multiparty multiplication protocol based on a single particle is proposed. The proposed protocol utilizes single particles as information carriers, avoiding the preparation difficulties associated with entangled states. It employs mutually unbiased bases for eavesdropping detection, enabling participants to avoid preparing any particles, thereby reducing the quantum capabilities required of participants and minimizing quantum resource consumption. The proposed protocol also incorporates an improved quantum modular multiplication circuit, which decreases the number of rotation gates required for implementing modular multiplication, thus reducing computational overhead. Security analysis shows that the protocol can resist external and internal attacks, and performance analysis shows that the protocol has good execution efficiency and low computational complexity. Finally, the protocol proposed in this paper is simulated using the quantum programming toolkit Qiskit.

The contributions of this paper are as follows:

1) A quantum secure multiparty multiplication protocol is proposed, which does not use decoy particles and entangled states has lower computational complexity and resource consumption.

2) The calculation cost of the participant is analyzed from the perspective of the complexity of the circuit in which the participant performs the calculation.

3) Using AQFT instead of QFT, the number of turnstiles used to complete quantum modular multiplication is reduced, the circuit complexity of the QSMM protocol is reduced, and the protocol calculation phase is simulated.

2 Preliminaries

This section mainly introduces the key quantum states and quantum operations involved in the protocol, including mutually unbiased bases, high-dimensional general unitary operators, and the implementation of quantum modular multiplication. These preliminaries provide the necessary theoretical foundation for understanding the protocol description and related content in the subsequent sections.

2.1 Quantum mutually unbiased bases states

When d is a prime number, it is possible to find $d+1$ sets of Mutually Unbiased Bases (MUBs) in a d-dimensional complex vector space [33, 34]. In addition to the computational basis $\{|j\rangle, j = 0, 1, \ldots, d-1\}$, the other d MUBs are represented as

$$\left|e_l^{(j)}\right\rangle = \frac{1}{\sqrt{d}} \sum_{u=0}^{d-1} \omega^{u(l+ju)} |u\rangle , \tag{1}$$

where $j, l = 0, 1, \ldots, d-1$ denote the basis set index and the enumeration of vectors in a given basis, respectively, and $\omega = \mathrm{e}^{2\pi \mathrm{i}/d}$ is a primitive d-th root of unity. For any $j \neq j'$ and $l \neq l'$, the MUBs satisfy the mutually unbiased condition, that is:

$$\langle e_l^{(j)} | e_{l'}^{(j)} \rangle = 0, \quad |\left\langle e_l^{(j)} | e_{l'}^{(j')} \right\rangle|^2 = \frac{1}{d}. \tag{2}$$

There exists a transformation operation as follows:

$$\mathcal{W}(j') = \sum_{r=0}^{d-1} \omega^{j' r^2} |r\rangle \langle r|, \tag{3}$$

applying the above unitary operation to the MUBs yields:

$$\begin{aligned} \mathcal{W}(j')|e_l^{(j)}\rangle &= \frac{1}{\sqrt{d}} \left(\sum_{r=0}^{d-1} \omega^{j' r^2} |r\rangle \langle r| \right) \left(\sum_{u=0}^{d-1} \omega^{u(l+ju)} |u\rangle \right) \\ &= \frac{1}{\sqrt{d}} \sum_{u=0}^{d-1} \omega^{u[l+(j+j')u]} |u\rangle \\ &= |e_l^{(j+j')}\rangle, \end{aligned} \tag{4}$$

where "+" denotes addition modulo d. This expression indicates that $|e_l^{(j)}\rangle$ can be mapped to $|e_l^{(j+j')}\rangle$ via the unitary transformation $\mathcal{W}(j')$, demonstrating the cyclic nature of the mutually unbiased bases.

2.2 Quantum multiplication implementation

Since the rotation gate approximates the identity matrix when the rotation angle is sufficiently small, we can reduce the use of rotation gates in QFT with angles below a

certain threshold while maintaining accuracy, thereby improving circuit execution efficiency. Based on this observation, our protocol improves upon the Draper adder [35] by replacing the QFT with an Approximate Quantum Fourier Transform (AQFT). Using this approach, we construct a quantum modular multiplication circuit that implements $|x\rangle \rightarrow |ax \bmod d\rangle$, represented as the unitary transformation $\mathcal{MUL}(a)$.

Furthermore, quantum systems inevitably interact with their external environment, causing the coherence of quantum states (i.e., quantum superposition and phase information) to gradually decay - a process known as decoherence, which introduces computational errors. As Barenco et al. [36] demonstrated, in the presence of decoherence, AQFT may actually yield more accurate results than QFT precisely because it reduces the number of quantum gates required.

Let $|b\rangle$ be a quantum state in a d-dimensional complex space. The state $|b\rangle$ can be written in binary form as $b = b_{n-1}b_{n-2}\dots b_0$, more formally expressed as $b = b_{n-1}2^{n-1}b_{n-2}2^{n-2} + \dots + b_0 2^0$. For convenience, the notation $0.b_l b_{l+1}\dots b_m$ denotes the binary fraction $b_l/2 + b_{l+1}/4 + \dots + b_m/2^{m-l+1}$. Let $e(t) = e^{2\pi i t}$, rotation transformation $R_t = diag(1, e^{\frac{\pi i}{2^t}})$. The state $|\phi(b)\rangle$ is used to represent the state $|b\rangle$ after AQFT. Taking a single quantum bit $|b_{n-1}\rangle$ as an example, the state transformation based on AQFT can be expressed as:

$$
\begin{aligned}
|b_{n-1}\rangle &\xrightarrow{\text{Hadamard transform}} \frac{1}{\sqrt{2}}(|0\rangle + e(0.b_{n-1})|1\rangle) \\
&\xrightarrow[\text{conditioned on } b_{n-2}]{R_1 \text{ rotation}} \frac{1}{\sqrt{2}}(|0\rangle + e(0.b_{n-1}b_{n-2})|1\rangle) \\
&\dots \\
&\xrightarrow[\text{on } b_{n-k-1}]{R_k \text{ rotation conditioned}} \frac{1}{\sqrt{2}}(|0\rangle + e(0.b_{n-1}b_{n-2}\dots b_{n-k-1})|1\rangle) \\
&= |\phi_{n-1}(b)\rangle .
\end{aligned}
\tag{5}
$$

According to Ref. [37], if the number of qubits $n \geq 4$, using AQFT with $k \geq \log_2 n + 2$ can achieve the same accuracy as using QFT, and the same efficiency of both increases with the increase of n. By taking the addend $|a\rangle$ as the control qubit and each bit of the quantum state $|\phi(b)\rangle$, which has undergone an AQFT, as the target qubits, rotation gates can be applied iteratively to yield the quantum state $|\phi(a+b)\rangle$.

$$
\begin{aligned}
&|\phi_{n-1}(b)\rangle \\
&\xrightarrow{R_0 \text{ rotation conditioned on } a_{n-1}} \frac{1}{\sqrt{2}}(|0\rangle + e(0.b_{n-1}b_{n-2}\dots b_{n-k-1} + 0.a_{n-1})|1\rangle) \\
&\xrightarrow{R_1 \text{ rotation conditioned on } a_{n-2}} \frac{1}{\sqrt{2}}(|0\rangle + e(0.b_{n-1}b_{n-2}\dots b_{n-k-1} + 0.a_{n-1}a_{n-2})|1\rangle) \\
&\dots \\
&\xrightarrow{R_k \text{ rotation conditioned on } a_{n-k-1}} \\
&\quad \frac{1}{\sqrt{2}}(|0\rangle + e(0.b_{n-1}b_{n-2}\dots b_{n-k-1} + 0.a_{n-1}a_{n-2}\dots a_{n-k-1})|1\rangle) \\
&= |\phi_{n-1}(a+b)\rangle .
\end{aligned}
\tag{6}
$$

Applying the inverse approximate Fourier transform on the quantum state $|\phi(a+b)\rangle$ yields $|a+b \bmod N\rangle$.

The modular multiplication circuit can be constructed by repeatedly applying the modular addition circuit on different bit positions. Without loss of generality, assume two integers a and b and an integer x in the finite field $GF(d)$, where the objective is to compute $b+ax \bmod d$. This can be achieved by transforming as follows:

$$\begin{aligned}
&(b+ax) \bmod d \\
&= (b + a \times \sum_{i=0}^{n-1} x_i 2^i) \bmod d \\
&= (b + a \times (x_0 2^0 + x_1 2^1 + x_2 2^2 + \cdots + x_{n-2} 2^{n-2} + x_{n-1} 2^{n-1})) \bmod d \\
&= (b + x_0 \times a2^0 + x_1 \times a2^1 + x_2 \times a2^2 + \cdots + x_{n-2} \times a2^{n-2} + x_{n-1} \times a2^{n-1}) \bmod d.
\end{aligned} \tag{7}$$

3 The proposed protocol

This section describes the proposed protocol in detail, including the purpose of the protocol, the roles in the protocol, and the specific implementation process.

3.1 Protocol objective

The protocol proposed in this paper involves two types of entities: a semi-honest third party (TP) and n participants $P_1, P_2, \ldots, P_n$. These entities are defined as follows:

TP: The TP is a semi-honest third party responsible for preparing particles, performing eavesdropping detection and measuring the final product result. Semi-honesty implies that the TP may attempt to infer some participants' secret information while executing the protocol but cannot collude with any participants [38].

Participants: There are n participants $P_1, P_2, \ldots, P_n$, each holding a secret integer x_i ($x_i \in \{0, 1, \ldots, d-1\}$). They wish to compute the product of their secret integers without disclosing their individual values, obtaining $M = x_1 x_2 \ldots x_n \bmod d$. Here, d indicates that the secret integers belong to the finite field $GF(d)$.

Assume the protocol is executed over an ideal (noiseless) channel. According to [39], a quantum secure multiparty protocol must satisfy the following requirements:

1. *Correctness:* The computed result must accurately represent the product of all participants' secret values.

2. *Security:* External eavesdroppers cannot obtain the secret values of any participant without being detected.

3. *Privacy:* Each participant should learn no more than what is within their designated scope, meaning that each participant's secret value remains confidential.

The proposed protocol flowchart is shown in Fig. 1. Solid arrows indicate quantum channels under ideal conditions, and the dashed rectangular boxes explain each symbol in the flowchart. We denote the j-th particle in the particle sequence S_i as $S_i^{(j)}$.

3.2 Protocol procedure

Protocol Initialization Stage

The preparation stage is primarily responsible for generating private keys to protect confidential data and producing random numbers for security verification, while simultaneously preparing and distributing the initial quantum particles.

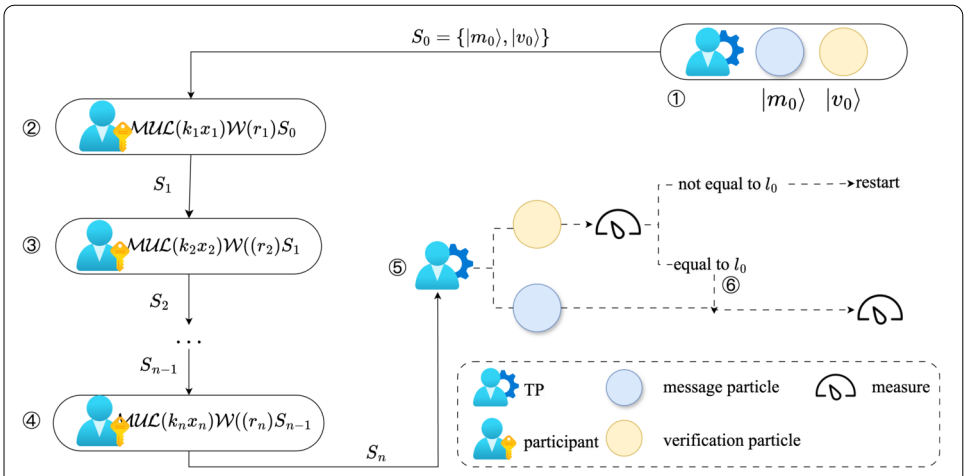

Figure 1 Flow chart of quantum secure multiparty multiplication protocol is proposed

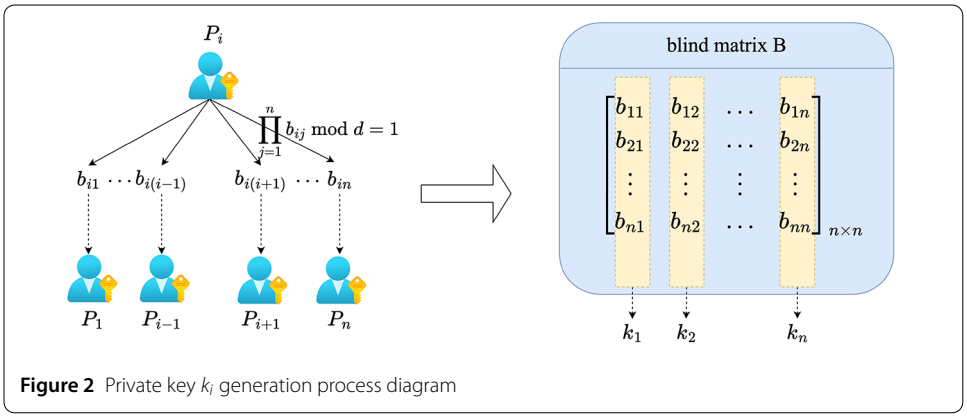

Figure 2 Private key k_i generation process diagram

Step 1: Participants generate a blind matrix.

Each participant $P_i(i = 1, 2, \ldots, n)$ generates the elements $b_{ij}(j = 1, 2, \ldots, n)$ of the i-th row of a blind matrix B, satisfying the condition $\prod_{j=1}^{n} b_{ij} \bmod d = 1$, where b_{ij} denotes the element in the i-th row and j-th column. With the exception of element b_{ii}, TP distributes all other elements $\{b_{ij}|j = 1, 2, \ldots n, j \neq i\}$ separately to the corresponding participant P_j through a secure channel. Once all elements have been distributed, each participant P_j possesses elements $\{b_{ij}|i = 1, 2, \ldots n\}$ and computes their product $\prod_{i=1}^{n} b_{ij} \bmod d$ as its private key k_j. The key generation process is illustrated in Fig. 2.

Step 2: TP generates random numbers.

The TP generates n random numbers $\{r_i|i = 1, 2, \ldots, n; r_i \in GF(d)\}$ that meet the condition $\sum_{i=1}^{n} r_i \bmod d = 0$, and sends r_i to corresponding participant P_i via a secure channel.

Step 3: TP distributes particles.

The TP prepares n auxiliary particles $|0\rangle_i$ in the 0-state for the multiplication circuit and sends each to the corresponding participant P_i over a quantum channel. The TP also prepares an message particle $|m_0\rangle$ and a verification particle $|v_0\rangle$, which together form a

particle sequence S_0 that is sent to participant P_1. Here, m_0, j_0, l_0 are all randomly selected by TP, $m_0 \in GF(d) - \{0\}$, and $|v_0\rangle$ is one of the d mutually unbiased bases states $|e_l^{(j)}\rangle = \frac{1}{\sqrt{d}} \sum_{u=0}^{d-1} e^{\frac{2\pi i}{N} u(l+ju)} |u\rangle$ (for $j, l = 0, 1, \ldots, d-1$) except the computational basis, specifically $|e_{l_0}^{(j_0)}\rangle$.

Privacy Computing Stage

During this stage, each participant performs unitary operations to encode their secret information into the received particle sequences before transmitting them to the next participant.

Step 4: Computation by Participant P_1.

After receiving the particle transmitted from TP, P_1 determines which particle to perform $\mathcal{MUL}$ and $\mathcal{W}$ operations according to its own range of random number r_1.

Specifically, if $r_1 \in \left[0, \frac{d}{2}\right)$, he applies the multiplication circuit $\mathcal{MUL}(k_1 x_1)$ to the first particle of S_0 and his auxiliary particle $|0\rangle_1$, using the product of his private key and secret $k_1 x_1$ as the parameters. He also applies the unitary operator $\mathcal{W}(r_1)$ to the second particle of S_0, yielding $S_1 = \{\mathcal{MUL}(k_1 x_1) S_0^{(1)}, \mathcal{W}(r_1) S_0^{(2)}\}$, as shown in Eq. (8). If $r_1 \in \left[\frac{d}{2}, d\right)$, P_1 applies $\mathcal{W}(r_1)$ to the first particle of S_0 and applies $\mathcal{MUL}(k_1 x_1)$ to the second particle, yielding $S_1 = \{\mathcal{W}(r_1) S_0^{(1)}, \mathcal{MUL}(k_1 x_1) S_0^{(2)}\}$. After the unitary operations, P_1 checks the parity of r_1, if odd, he swaps the two particles; if even, no swap is performed.

$$\begin{aligned} S_1 &= \{\mathcal{MUL}(k_1 x_1) S_0^{(1)}, \mathcal{W}(r_1) S_0^{(2)}\} \\ &= \{\mathcal{MUL}(k_1 x_1) |m_0\rangle, \mathcal{W}(r_1) |v_0\rangle\} \\ &= \{|m_0 k_1 x_1\rangle, \left|e_{l_0}^{(j_0 + r_1)}\right\rangle\}. \end{aligned} \tag{8}$$

Subsequently, P_1 sends the particles sequence S_1 to the next participant P_2.

Step 5: Computation by Participant $P_i (i = 2, 3, \ldots, n-1)$.

Each participant $P_i (i = 2, 3, \ldots, n-1)$, upon receiving the particles sequence S_{i-1} from P_{i-1}, identifies the message and verification particles based on the finite field range of its random number r_i. Specifically, if $r_i \in \left[0, \frac{d}{2}\right)$, P_i applies $\mathcal{MUL}(k_i x_i)$ to the first particle of S_{i-1} and its auxiliary particle $|0\rangle_i$, and $\mathcal{W}(r_i)$ to the second particle. This yields $S_i = \{\mathcal{MUL}(k_i x_i) S_{i-1}^{(1)} |0\rangle, \mathcal{W}(r_i) S_{i-1}^{(2)}\}$. If $r_i \in \left[\frac{d}{2}, d\right)$, P_i reverses the operations. P_i then check the parity of r_i: if odd, he swaps the particles; if even, no swap occurs.

Afterward, P_i sends the updated particles sequence S_i to the next participant P_{i+1}.

Step 6: Computation by Participant P_n.

Participant P_n repeats the same participant actions as above and, upon completion, sends the final message particle $|m_n\rangle$ and verification particle $|v_n\rangle$ as a new sequence S_n back to the TP.

$$\begin{aligned} S_n &= \{\mathcal{MUL}(k_n x_n) S_{n-1}^{(1)}, \mathcal{W}(r_n) S_{n-1}^{(2)}\} \\ &= \{\mathcal{MUL}(x_n k_n) \ldots \mathcal{MUL}(x_2 k_2) \mathcal{MUL}(x_1 k_1) S_0^{(1)}, \mathcal{W}(r_n) \ldots \mathcal{W}(r_2) \mathcal{W}(r_1) S_0^{(1)}\rangle\} \\ &= \{|m_0 \prod_{i=1}^{n} x_i \prod_{i=1}^{n} k_i \bmod d\rangle, |e_l^{(j_0 + r_1 + r_2 + \cdots + r_n)}\rangle\}. \end{aligned} \tag{9}$$

Security Verification Stage

This stage focuses on detecting whether the protocol has been executed securely.

Algorithm 1 The Multiplication Process of the Proposed Protocol

Require: $l_0, m_0, S_0^{(1)}, S_0^{(2)}, k_i, x_i, r_i$

Ensure: product of participants' secret integers X

for $i = 1$ to n **do**
  if $r_i \in \left[0, \frac{d}{2}\right)$ **then**
    $S_i^{(1)} \Leftarrow \mathcal{MUL}(k_i x_i) S_{i-1}^{(1)}$
    $S_i^{(2)} \Leftarrow \mathcal{W}(r_i) S_{i-1}^{(2)}$
  else
    $S_i^{(1)} \Leftarrow \mathcal{W}(r_i) S_{i-1}^{(1)}$
    $S_i^{(2)} \Leftarrow \mathcal{MUL}(k_i x_i) S_{i-1}^{(2)}$
  end if
  if r_i is odd **then**
    $S_i \Leftarrow \mathcal{SWAP}(S_i^{(1)}, S_i^{(2)})$
  end if
end for
$l \Leftarrow$TP measures $S_n^{(2)}$
if $l = l_0$ **then**
  $m_n \Leftarrow$ TP measures $S_n^{(1)}$
  $X \Leftarrow m_n m_0^{-1}$
else
  restart protocol
end if
return X

Step 7: Eavesdropping detection by the TP.

Upon receiving the particles sequence S_n from P_n, the TP measures the verification particle $|v_n\rangle$ in the preparation basis $\{|e_l^{(r_0)}\rangle | l = 0, 1, \ldots, d-1\}$. If the measured value l differs from the initially prepared value l_0, it indicates that there is eavesdropping during the execution of the protocol, and the calculation result is incorrect, and then the protocol should be terminated. If $l = l_0$, the protocol is deemed secure and correctly executed.

Result Output Stage

In this stage, TP computes and distributes the product results.

Step 8: Product computation by the TP.

The TP measures the message particle in the computational basis $\{0, 1, \ldots, d-1\}$ to obtain the result m_n. Then TP calculates the product of m_n and the multiplicative inverse of m_0, $m_n m_0^{-1}$ as the final product of all participants' secret values. The TP publicly announces the result to all participants.

For the proposed protocol, the multiplication process algorithm is shown as Algorithm 1.

4 Proof of correctness

This section proves the correctness of the proposed protocol. Specifically, the proof shows that if the protocol is executed correctly as designed, TP can detect eavesdropping during transmission by verifying particle and accurately obtain product result from message particle, thus achieving the stated goal of the protocol.

4.1 Correctness of the product of secret integers

Proposition 1 *In the proposed protocol, when the initial message particle $|m_0\rangle$ is transmitted from the TP to the first participant, if each participant applies the $\mathcal{MUL}$ operation on the particle honestly as specified, the TP can accurately obtain the product of all participants' secrets $\prod_{i=1}^{n} x_i \bmod d$, upon receiving the message particle from the final participant.*

Proof Each row of the blind matrix B generated by the participants satisfies $\prod_{j=1}^{n} b_{ij} \bmod d = 1$, so the product of all participants' private keys, i.e., the product of all elements in this matrix, satisfies:

$$\begin{aligned}
&\prod_{j=1}^{n} k_j \bmod d \\
&= \prod_{i=1}^{n}\prod_{j=1}^{n} b_{ij} \bmod d \\
&= \prod_{j=1}^{n} b_{1j} \times \prod_{j=1}^{n} b_{2j} \times \cdots \times \prod_{j=1}^{n} b_{nj} \\
&= 1^n \bmod d \\
&= 1.
\end{aligned} \tag{10}$$

According to the function of the $\mathcal{MUL}$ circuit, $\mathcal{MUL}(x)|a\rangle|b\rangle = |b+ax\rangle|a\rangle$. When $b = 0$, we have $\mathcal{MUL}(x)|a\rangle|0\rangle = |ax\rangle|a\rangle$. Since the auxiliary particle is in state $|0\rangle$ each time the $\mathcal{MUL}$ circuit is applied, the message particle maintains the modular product of its initial value and the parameter throughout the process,

$$\begin{aligned}
|m_n\rangle &= \mathcal{MUL}(x_n k_n)\ldots\mathcal{MUL}(x_2 k_2)\mathcal{MUL}(x_1 k_1)|m_0\rangle \\
&= |m_0 x_1 k_1 x_2 k_2 \ldots x_n k_n \bmod d\rangle \\
&= |m_0 \prod_{i=1}^{n} x_i \prod_{i=1}^{n} k_i \bmod d\rangle \\
&= |m_0 \prod_{i=1}^{n} x_i \bmod d\rangle.
\end{aligned} \tag{11}$$

The TP then measures the message particle, obtaining $m_n = m_0 \prod_{i=1}^{n} x_i \bmod d$. By the properties of the finite field $GF(d)$, each non-zero element m_0 has a unique multiplicative inverse m_0^{-1}, yielding $\prod_{i=1}^{n} x_i \bmod d = m_n m_0^{-1}$. □

4.2 Correctness of verification particle measurement results

Proposition 2 *In the proposed protocol, the initial verification particle $|e_{l_0}^{(j_0)}\rangle = \frac{1}{\sqrt{d}}\sum_{u=0}^{d-1} e^{\frac{2\pi i}{d}u(l_0+j_0 u)}|u\rangle$ is transmitted from the TP to the first participant. If each participant*

P_i honestly applies the operator $\mathcal{W}(r_i)$ to the particle, the TP can determine the security of the protocol execution based on the measurement result of the verification particle received from the final participant.

Proof When all participants apply the $\mathcal{W}(r_i)$ operator to the verification particle, the state of the particle evolves as follows:

$$\begin{aligned}
&\mathcal{W}(r_n)\dots\mathcal{W}(r_2)\mathcal{W}(r_1)|e_{l_0}^{(j_0)}\rangle \\
&= \frac{1}{\sqrt{d}}\left(\sum_{r=0}^{d-1}\omega^{r_n r^2}|r\rangle\langle r|\right)\dots\left(\sum_{r=0}^{d-1}\omega^{r_1 r^2}|r\rangle\langle r|\right) \\
&\quad\left(\sum_{u=0}^{d-1}\omega^{u(l+ju)}|u\rangle\right) \\
&= \frac{1}{\sqrt{d}}\sum_{u=0}^{d-1}\omega^{u\left[l+(j_0+r_1+r_2+\cdots+r_n)u\right]}|u\rangle \\
&= |e_l^{(j_0+r_1+r_2+\cdots+r_n)}\rangle.
\end{aligned} \tag{12}$$

Since the random numbers prepared by the TP satisfy $\sum_{i=1}^{n} r_i \bmod d = 0$, if there is no eavesdropping or similar disturbance during the protocol, the measurement result of the verification particle received by the TP in the basis $\{e_l^{(j_0)}|l = 0, 1, \dots, d-1\}$ should yield l_0. □

5 Security analysis

This section analyzes the security of the proposed QSMM protocol from two attack dimensions, namely external attacks and internal attacks, to ensure that the attack behavior of external eavesdroppers or dishonest participants trying to steal the private information of honest participants will inevitably be detected, and the attacker cannot obtain any valid information, and the protocol can maintain its security.

5.1 External attacks

The protocol is designed to be robust against several types of external attacks, including intercept-resend, forgery-replay, and entanglement-measurement attacks.

5.1.1 Intercept-resend attack

Assume an external attacker, Eve, tries to intercept the secret of participant P_i (where $i \in \{2, 3, \dots, n-1\}$). She intercepts the particles sequence sent from P_{i-1} to P_i. Since particles sequence are reordered during participant operations, Eve cannot identify the message particle. If she correctly guesses the particle order with a probability of $\frac{1}{2}$, selecting the message particle, she would encounter the measurement value equal to the product of the secret and the private key. However, without knowledge of the private key, Eve cannot infer the participant's secret from this measurement. If Eve chooses the verification particle instead (with $\frac{1}{2}$ probability), the unitary operation $\mathcal{W}$ based on each participant's random number changes the particle's measurement basis. Eve, lacking knowledge of the correct measurement basis, has a $\frac{1}{d}$ probability of choosing the correct basis and a $1-\frac{1}{d}$ probability

of selecting an incorrect basis. When Eve measures with an incorrect basis, she has a $\frac{1}{d}$ probability of obtaining a correct result, resulting in a $1-\frac{2d-1}{d^2}$ probability of detection by TP, with no useful information obtained by Eve.

5.1.2 Forgery-replay attack

Suppose Eve intercepts the particles sequence S_{i-1} sent to participant P_i and replaces the message particle $|m_{i-1}\rangle$ and verification particle $|v_{i-1}\rangle$ with her own particles $\{|1\rangle, |1\rangle\}$ before sending them to P_i. After the P_i performs the unitary operation on all particles, Eve intercepts the particles sequence S_i again. Measuring these particles, she obtains $x_i k_i$ and r_i, but, due to potential reordering, she cannot distinguish between the secret and verification values. Even if she could, without knowledge of the private key k_i, Eve cannot extract secret integer x_i of the P_i.

5.1.3 Entanglement-measurement attack

Eve intercepts the particle sequences S_i sent from P_i to P_{i+1}. If she correctly identifies the message particle with $\frac{1}{2}$ probability, she prepares an auxiliary particle $|e\rangle_A$ to entangle with the message particle $|m_i\rangle_m$, forming the composite system $|\rho\rangle = |m_i\rangle_m |e\rangle_A = |m_0 x_1 k_1 \ldots x_i k_i\rangle |e\rangle_A$. Eve then performs a d-dimensional CNOT operation, with the message particle as the control and the auxiliary particle as the target, resulting in $|\rho\rangle' = |m_0 x_1 k_1 \ldots x_i k_i\rangle_m |m_0 x_1 k_1 \ldots x_i k_i + e\rangle_A$. Measuring the auxiliary particle yields $m_0 x_1 k_1 \ldots x_i k_i + e$. Without knowing the product $m_0 k_1 \ldots k_i$, Eve cannot infer the product of the secret integers $x_1 \ldots x_i$, nor can she deduce any single participant's secret integer. Thus, the attack fails.

5.1.4 Double CNOT attack

The double CNOT attack was first proposed by [40], and widely discussed in quantum secure multi-party computing protocols [41, 42]. In this attack, the eavesdropper Eve stealthily steals the operational information of legitimate participants through two CNOT operations. Suppose Eve aims to intercept the information of participant P_i without being detected. She first intercepts the particle sequence S_{i-1} sent from P_{i-1} to P_i.

If Eve uses the verification particle $|v_{i-1}\rangle_v$ as the control qubit and her prepared auxiliary particle $|0\rangle_A$ as the target qubit, applying a CNOT gate yields the entangled state:

$$\frac{1}{\sqrt{d}} \sum_{u=0}^{d-1} \omega^{u[l+(j+\sum_{i=0}^{i-1} r_i)u]} |u\rangle_v |u\rangle_A . \tag{13}$$

Eve then retains the auxiliary particle and sends the first particle to P_i. After P_i performs the unitary operation, Eve intercepts the particle again and applies a second CNOT gate, evolving the entire system to:

$$\frac{1}{\sqrt{d}} \sum_{u=0}^{d-1} \omega^{u[l+(j+\sum_{i=0}^{i-1} r_i)u]} |u\rangle_v |0\rangle_A . \tag{14}$$

Although this operation disentangles the system and evades security detection, when Eve measures the auxiliary particle, she can only obtain the outcome 0 and fails to extract the participant's secret information.

For the case where Eve uses the message particle $|m_{i-1}\rangle_m$ as the control qubit, the first CNOT operation results in:

$$\text{CNOT}\,|m_{i-1}\rangle_m\,|0\rangle_A = |m_{i-1}\rangle_m\,|m_{i-1}\rangle_A\,. \tag{15}$$

Eve then sends the information particle to P_i and intercepts it again after P_i's operation. The second CNOT operation yields:

$$\text{CNOT}\,|m_{i-1}x_ik_i\rangle_m\,|m_{i-1}\rangle_A = |m_{i-1}x_ik_i\rangle_m\,|m_{i-1}+m_{i-1}x_ik_i\rangle_A\,. \tag{16}$$

By analyzing the states of the auxiliary particle before and after the two CNOT operations, Eve can infer $m_{i-1}+m_{i-1}x_ik_i$, but she cannot deduce the secret integer x_i of participant P_i.

5.1.5 Reflecting attack

Reflecting attacks are a common threat in semi-quantum key agreement protocols [43, 44]. In such protocols, the semi-quantum party, typically Bob, is limited in quantum operational capabilities and can only choose to perform Z-basis measurements followed by re-preparation of the quantum state, or directly reflect the received quantum state back to the sender. An adversary, Eve, may launch a reflection attack by simply forwarding the quantum state without introducing any modifications. This behavior mimics that of a legitimate participant, making it indistinguishable from Bob's allowed operations. Consequently, the sender, Alice, is unable to detect any abnormality and may mistakenly identify Eve as the legitimate Bob. By exploiting the partial information disclosed by Alice during the protocol and leveraging knowledge of the initial quantum states, Eve may be able to infer private data.

In the proposed protocol, it is assumed that Eve intercepts the particle sequence S_{i-1} sent by P_{i-1} to P_i and directly reflects it as S_i to P_{i+1}. However, the reflecting attack causes some participants not to embed random numbers as agreed, and the verification particle measurement value will be different from the initial set value l_0. The attack will be detected, and Eve cannot obtain any information.

5.2 Internal attacks

Internal attacks may involve a single dishonest participant or collusion among multiple dishonest participants. Additionally, a semi-honest TP might also attempt to compromise the protocol and retrieve some participants' secrets.

5.2.1 Attack from a single dishonest participant

Suppose P_i wants to steal the secret of P_{i+1}. Similar to the forgery-replay attack, P_i replaces the message particle with $|1\rangle$ and intercepts the sequence again after P_{i+1} has performed the unitary operation. Since P_i does not know r_{i+1}, they cannot determine which is the message particle. Even if they identify $x_{i+1}k_{i+1} \bmod d$ with some probability, without knowing k_{i+1}, P_i cannot infer x_{i+1}.

5.2.2 Collusion attack by multiple participants

Assume P_{i-1} and P_{i+1} collude to steal P_i's secret. P_{i-1} obtains the message particle $|m_0x_1k_1\ldots x_{i-1}k_{i-1}\rangle$, and P_{i+1} obtains the message particle $|m_0x_1k_1\ldots x_{i-1}k_{i-1}x_{i+1}k_{i+1}\rangle$. Together, they can calculate x_ik_i, but without knowing k_i, they cannot deduce x_i. Even if more

participants (up to $n-2$ in number) collude, since the private keys are generated jointly by all n participants, they cannot ascertain any specific participant's private key.

5.2.3 Attack from a semi-honest TP

As a semi-honest third party, TP may attempt to obtain participants' secrets without colluding with them. Suppose TP intercepts the sequence $\{|m_i\rangle, |v_i\rangle\}$ by an intercept-resend or forgery-replay attack. Since TP distributes and verifies the random numbers, any eavesdropping on verification particles is undetectable for TP. For the message particle $|m_i\rangle$, TP's eavesdropping is similar to the case of collusion among multiple participants. TP can only obtain the product $x_i k_i$, but, as TP is not involved in the generation of k_i, it cannot infer x_i, and the attack fails.

6 Performance analysis and comparison

In this section, we analyze the performance of the proposed quantum secure multiparty multiplication protocol and compare it with other existing multiplication protocols. To facilitate comparison across protocols, we normalize certain characteristics: the initial state of the prepared particles is assumed to be in a d-dimensional Hilbert space, the number of participants in the computation is n, and the product result is in the finite field $GF(d)$. Additionally, some protocols use decoy particles to ensure security, so the number of decoy particles used in each transmission is standardized to t. The performance of each secure multiparty multiplication protocol is assessed and compared as shown in Tables 1 and 2. In these tables, resource cost represents the number of particles required to execute the protocol, and the computational cost reflects the number of unitary operations needed. Communication cost represents the total particle transmission consumption in the protocol execution process, measured in units of a single particle being transmitted once over the quantum channel. Circuit complexity, also referred to as circuit scale, represents the number of basic gates used during a single execution of the secret embedding operation by an individual participant in all protocols.

In the protocol [26], the process of calculating the product of all participants' secret values is divided into a summation of exponents and a final multiplication. During the summation process, the initiator prepares a two-particle entangled state using one QFT and one CNOT operation, keeping the first particle and passing the second message particle around in a ring among participants. Each participant prepares a particle according to their secret value and applies an Oracle operator on their secret particle and the message particle to perform an entanglement swap, embedding their secret into the phase. The initiator then measures the first particle to detect eavesdropping and performs an IQFT and a measurement on the second particle to obtain the summation result. Participants $P_2, \ldots, P_n$ apply the Oracle operation on the auxiliary particle and message particle to form a two-particle entangled state. In the multiplication stage, the initiator's operations are similar to the summation process. They prepare a two-particle entangled state and apply transmissions and unitary transformations on both particles. The protocol concludes with eavesdropping detection and obtaining the result of the odd product. The protocol requires participants to also be capable of preparing particles, with a total particle preparation count of $n+3$. The communication cost is $3n$, the computational cost is $5n+3$, and the number of measurements is 4.

In the protocol proposed by [27], a secret integer is decomposed into the sum of powers of multiple prime factors (denoted as p) and an integer coprime to the dimension.

Specifically, let $x_i = s_i q_1^{k_i^1} q_2^{k_i^2} \cdots q_p^{k_i^p}$. The product of all secret numbers can then be achieved through p summations and one final multiplication. During the summation, the server selects a quantum state from two mutually unbiased bases as an message particle to embed the secret number, using decoy particles to prevent eavesdropping. The resource consumption per summation is $1 + (n + 1)t$. In the multiplication process, since s_i is coprime with the dimension, a multiplicative inverse s_i^{-1} exists. By using s_i as a parameter for the unitary operator acting on the message particle, s_i or s_i^{-1} can be embedded as a multiplier into the quantum state. The total resource consumption is $(p + 1)[1 + (n + 1)t]$, the communication cost is $(p + 1)(nt + n + t)$, the computational cost is $n(p + 1)$, and the number of measurements is $(p + 1)[1 + (n + 1)t]$.

In the quantum-secure multiparty multiplication protocol proposed by [28], the computation is initially designed as a two-party protocol, utilizing Shamir's secret sharing, which requires n auxiliary participants. To compare this protocol effectively with other multiparty multiplication protocols, it is repeated $n - 1$ times to extend from a two-party to an n-party computation. During a single product process, the initiator prepares an entangled particle state through one QFT and $n-1$ SUM gates, distributing the $n-1$ particles in a tree transmission mode to the corresponding participants, except for the first particle. Each participant then performs a QFT and a Pauli operator on their particle, measures it, and broadcasts the measurement result. After executing the protocol $n - 1$ times, the total resource consumption is $n(n - 1)$, the number of unitary operations is $3n(n - 1)$, the number of measurements is $n(n - 1)$, and the communication cost is $(n - 1)^2$.

The protocol proposed in [29] is an improved version of the protocol in [28]. To enable two-party computation, one of the n auxiliary participants is chosen as the initiator (denoted as P_1). This initiator prepares a particle and performs a QFT on it. Subsequently, they prepare an auxiliary particle and perform a CNOT operation on the two particles to form an entangled state. P_1 then sends the second particle in the entangled state to the next participant. Each participant prepares a particle based on their secret value, performs an Oracle operation on the two particles they hold, and passes them to the next participant. This process repeats $n - 1$ times until the message particle returns to the initiator P_1. P_1 performs a CNOT operation on the entangled particle, measures the auxiliary particle to detect any eavesdropping, and performs an IQFT and a measurement on the first particle to obtain the computation result. Since this is a two-party protocol, achieving n-party computation requires repeating the protocol $n - 1$ times. Therefore, the total resource consumption is $n^2 - 1$, the number of unitary operations is $3n^2 - 2n - 1$, the number of measurements is $2(n - 1)$, and the communication cost is $n(n - 1)$.

In the protocol proposed by [30], the Server prepares n particles in mutually unbiased bases and distributes them to each participant through a tree-structured transmission. Each participant performs a Lagrange unitary operator on their message particle to embed their secret value, then sends the particle back to the Server. The Server applies a Lagrange unitary operator based on the initial state to determine the measurement basis for each particle. After measuring all particles, the Server retrieves the secret values and performs the product calculation. Thus, the protocol requires the preparation of n particles, $3n$ unitary operations, n measurements, and a communication cost of $2n$. Additionally, the protocol introduces an extra entity, the "participant representative" to collect participants' random numbers during the result calculation stage.

In the computation stage of the protocol proposed by [31], TP prepares t decoy particles and one particle in a mutually unbiased basis. The TP then sends the sequence of particles to the first participant, who measures the decoy particles and compares them against a threshold to detect eavesdropping. The participant performs a phase unitary operation on the message particle to embed their secret value, prepares additional decoy particles, and randomly inserts the message particle among them before sending the sequence to the next participant. This process repeats for n participants until the particles return to the TP. After performing eavesdropping detection, the TP measures the particles to obtain the logarithmic sum; by exponentiating this sum, the product result can be retrieved. Thus, the protocol requires the preparation of $nt + t + 1$ particles, n unitary operations, $nt + t + 1$ measurements, and communication cost of $(n + 1)(t + 1)$.

In the proposed protocol, the TP prepares n particles and sends them separately to n participants, each owning one. Additionally, two particles are prepared and circulated among the participants in a ring structure to embed message. This setup results in a particle consumption of $n + 2$ and communication cost of $2n + 1$. Each participant performs 2 unitary operations on the particles, leading to the computational cost of $2n$. After receiving the particle sequence from the last participant, TP measures the verification particle to detect eavesdropping and measures the message particle to obtain the product value. The total number of measurements is 2.

The Oracle operator used in [26, 28] and [29] can be decomposed into $2d$ phase operations and one QFT and one $\mathrm{QFT}^{\dagger}$. Thus, when the Oracle operator is applied to an m-particle entangled state, the entangling and disentangling scale order is $O(md^2)$, the Oracle operator circuit scale order is $O(d^3)$, and the overall circuit is $O(md^2 + d^3)$. In Ref. [30], the Lagrange unitary operator is used to embed the secret numbers into mutually unbiased bases particles. For a 2-dimensional particle, the Lagrange unitary operator can be implemented using $M(\theta) = HP(\theta)H$. For higher-dimensional superposition states in mutually unbiased bases, the Lagrange unitary operator requires generalized Hadamard transformations and phase transformations, involving quantum Fourier transforms, controlled rotation gates, and inverse quantum Fourier transforms, with a circuit scale of $O(d^3)$.

In the proposed protocol, the constant addition circuit includes many continuous classically controlled rotations on a single qubit, apart from the AQFT and $\mathrm{AQFT}^{\dagger}$. These rotations can be combined into a single rotation operation via matrix multiplication, resulting in a circuit scale of 1 and d for this part. Considering the AQFT circuit and ignoring the $O(d)$ phase gates for addition operations, the asymptotic order of the constant addition circuit scale is $O(d \log d)$. Since a $\mathcal{MUL}$ gate consists of n controlled constant modular addition circuits and $3d$ CNOT gates (where one SWAP gate is composed of 3 CNOT gates), the circuit scale of a single modular multiplication operation is $O(d^2 \log d)$. The $\mathcal{W}$ operation applied to the verification particle is similar to the high-dimensional universal unitary operation applied to high-dimensional single particles, with a circuit scale of $O(d^2)$. In a single participant's computation, the overall circuit scale is $O(d^2 \log d)$.

Table 1 and Table 2 respectively show the comparison of the basic properties and performance of the above protocols and the proposed protocols. It can be observed that the protocols in [26, 28], and [29] all use entangled states as message carriers. The protocol in [30], although using single particles in mutually unbiased bases, requires the Server to perform the Lagrange unitary operator n times in the measurement stage to prepare n sets of measurement bases. Additionally, the protocols in [26–28], and [29] require partic-

Table 1 Characteristics comparison of QSMM protocols

Protocol	Particle Type	Particle Preparation	Execution Count	Quantum Circuit Simulation	Circuit Complexity Analysis
Shi et al. [26]	Entangled State	Server, P	2	No	No
Lv et al. [27]	Single Particle	Server, P	$p+1$	No	No
Sutradhar et al. [28]	Entangled State	Server	$n-1$	No	No
Sutradhar et al. [29]	Entangled State	Server, P	$n-1$	No	No
Zhang et al. [30]	Single Particle	Server	1	Yes	No
Li et al. [31]	Single Particle	Server, P	1	No	No
Proposed Protocol	Single Particle	Server	1	Yes	Yes

Table 2 Performance comparison of QSMM protocols

Protocol	Computation Cost	Resource Cost	Communication Cost	Measurement Count	Circuit Complexity
Shi et al. [26]	$5n+3$	$n+3$	$3n$	4	$O(md^2+d^3)$
Lv et al. [27]	$n(p+1)$	$(p+1)[1+(n+1)t]$	$(p+1)(nt+n+t)$	$(p+1)(1+tn+t)$	$O(d^3)$
Sutradhar et al. [28]	$3n(n-1)$	$n(n-1)$	$(n-1)^2$	$n(n-1)$	$O(md^3+d^4)$
Sutradhar et al. [29]	$3n^2-2n-1$	n^2-1	$n(n-1)$	$2(n-1)$	$O(md^3+d^4)$
Zhang et al. [30]	$3n$	n	$2n$	n	$O(d^3)$
Li et al. [31]	n	$t(n+1)+1$	$(t+1)(n+1)$	$t(n+1)+1$	$O(d^3)$
Proposed Protocol	$2n$	$n+2$	$2n+1$	2	$O(d^2 \log d)$

ipants to prepare particles, which increases the quantum capability requirements for the participants. The protocols in [26, 28], and [29] apply the Oracle operator to entangled states, and the protocols in [27] and [31] apply the Oracle operator to single particles. The protocol in [30] applies the Lagrange unitary operator to superposition states. The circuit complexity of all the above protocols is higher than that of the protocol proposed in this paper.

The number of participants n is proportional to the dimension d, and the number of protocol executions $p = log_2 d$. The probability of eavesdropping detection is $1-(\frac{1}{d})^t$, where t can be taken as 3 when d is sufficiently large. Based on the selection of the above parameters, we analyze the resource cost, the number of measurements, and the circuit complexity of each protocol, respectively, as shown in Fig. 3, Fig. 4 and Fig. 5.

7 Simulation experiments

In this section, we simulate the proposed protocol using specific examples on a classical computer. The simulation follows the principles of quantum mechanics and does not involve physical-layer implementations. The computational environment consists of an Apple M1 Pro processor with 16 GB of RAM, and the programming language used is Python. The simulation focuses on the privacy computing stage, security verification stage, and result output stage. The development environment includes VS Code 1.94.1, Python 3.9.19, and Qiskit 0.45.0.

Assuming that all quantum states exist in a 7-dimensional Hilbert space, i.e. d=7, and the proposed protocol involves 5 participants in the multiplication computation task. In the initialization stage, the server randomly generates numbers $r_1 = 1, r_2 = 4, r_3 = 5, r_4 = 2, r_5 = 2$, and the participants' secret numbers are $x_1 = 1, x_2 = 3, x_3 = 2, x_4 = 5, x_5 = 6$. The simulations for each stage are described as follows.

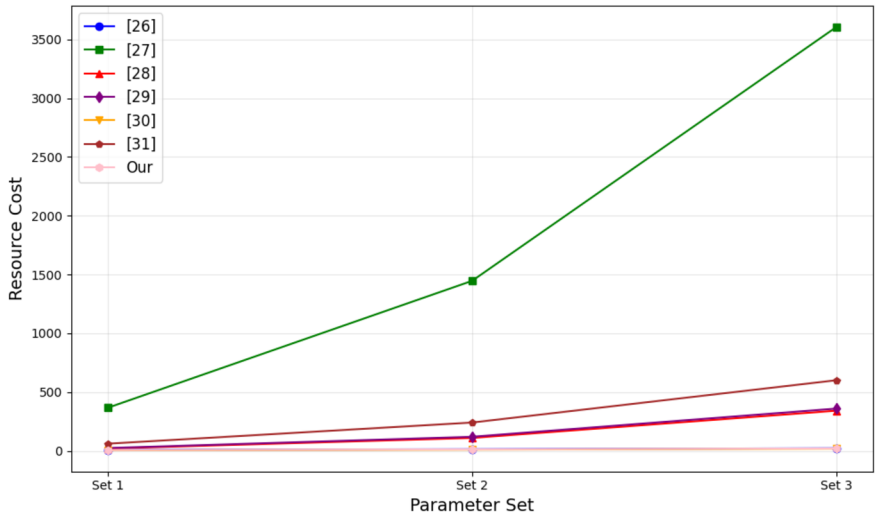

Figure 3 Comparison of resource consumption of all QSMM protocols under three sets of parameters

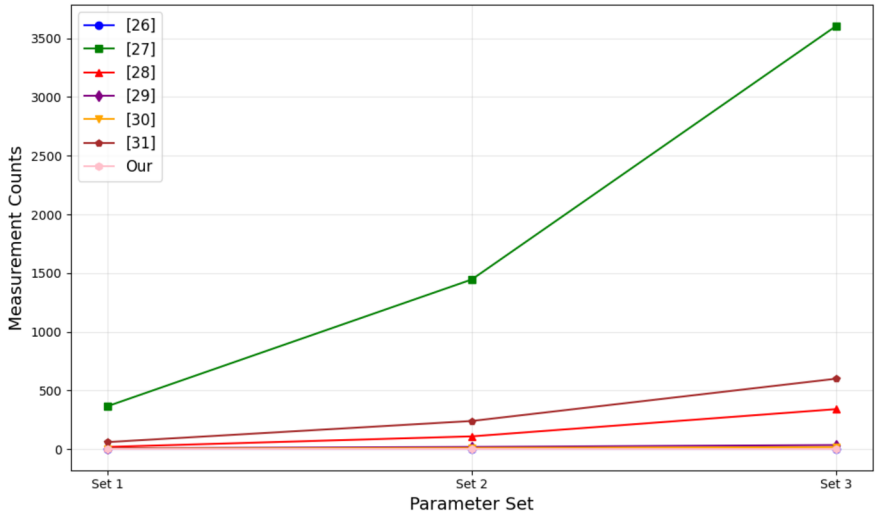

Figure 4 Comparison of measurement counts of all QSMM protocols under three sets of parameters

7.1 Initialization stage

All participants jointly generate a blind matrix $B = \begin{bmatrix} 2 & 4 & 1 & 1 & 1 \\ 2 & 2 & 2 & 4 & 2 \\ 2 & 1 & 5 & 5 & 1 \\ 3 & 2 & 1 & 2 & 3 \\ 1 & 3 & 1 & 1 & 5 \end{bmatrix}$, and participant P_1 is assigned the first column $[2, 2, 2, 3, 1]$. By calculating the product modulo d, P_1 obtain his private key $k_1 = 3$. Similarly, the other participants obtain $k_2 = 6, k_3 = 3, k_4 = 5, k_5 = 2$. The TP selects the message particle $m_0 = |4\rangle$ and the verification particle $v_0 = |e_2^{(5)}\rangle$. TP then prepares the particle sequence $S_0 = \{|4\rangle, |e_2^{(5)}\rangle\}$ and sends it to P_1.

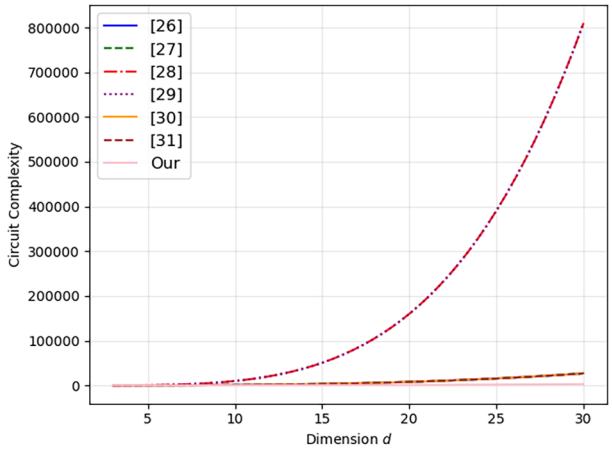

Figure 5 Comparison of participant computing circuit complexity of all QSMM protocols under three sets of parameters

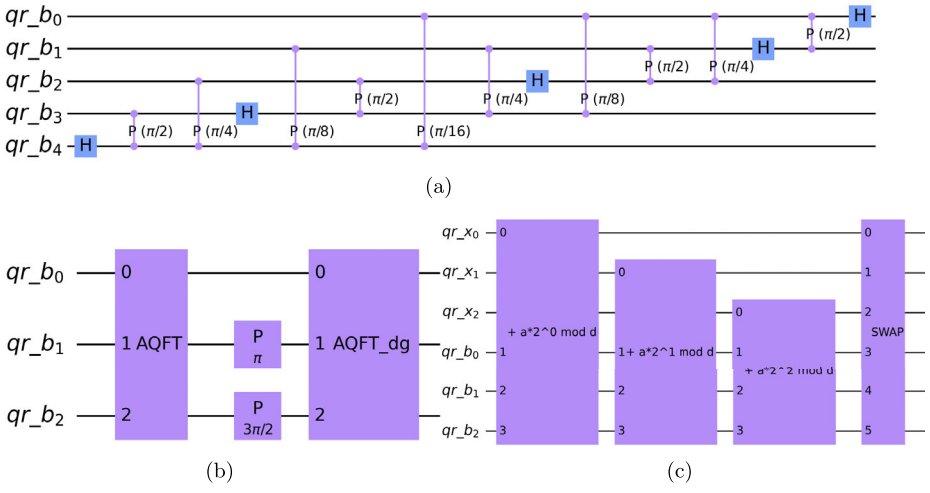

Figure 6 The circuits of quantum gates, (a), (b), (c) are circuits of $\mathcal{AQFT}$, addition derived from $\mathcal{AQFT}$, in-place modular multiplication, respectively

7.2 Privacy computing stage

The $\mathcal{MUL}$ used in the protocol is implemented by a repeat-use modular addition circuit, as shown in Fig. 6c. This protocol improves the QFT used by the Draper adder [35] by utilizing an AQFT, which reduces the use of rotation gates below a certain angle threshold and enhances circuit execution efficiency. Since the secret number in the protocol is a classically known value, the addend can be simplified as a phase gate parameter, leading to the modified adder circuit shown in Fig. 6b. Here, the implementation circuit for applying AQFT to any quantum state $|b\rangle$ is illustrated in Fig. 6a. The matrix expression of the phase gate $\mathcal{P}_k$ is given by $\mathcal{P}_k = \text{diag}(1, e^{\frac{\pi i}{2^k}})$, and $k \in [0, \lceil \log_2 n \rceil + 2]$.

As shown in Table 3, each participant performs unitary operations on the received particle sequences using their private key and random number. For participant P_1, $r_1 = 1 \in [0, d/2)$, he applies the operation $\mathcal{MUL}(3)$ on the first particle with $x_1 k_1 \mod d = 3$ as the parameter, and applies the operation $\mathcal{W}(1)$ on the second particle, resulting in $|5\rangle|e_2^{(6)}\rangle$.

Table 3 Simulation of privacy computing stage

Participant P_i	Privacy Input x_i	Private Key k_i	Random Number r_i	$\mathcal{MUL}$ Action Particle Number	Resulting Particle Sequence S_i
P_1	1	3	1	1	$\|e_2^{(6)}\rangle\|5\rangle$
P_2	3	6	4	2	$\|e_2^{(3)}\rangle\|6\rangle$
P_3	2	3	5	2	$\|1\rangle\|e_2^{(1)}\rangle$
P_4	5	5	2	1	$\|4\rangle\|e_2^{(3)}\rangle$
P_5	6	2	2	1	$\|6\rangle\|e_2^{(5)}\rangle$

Table 4 The measurement basis of verification particle

Participant P_i	After-operation Verification Particle v_i
TP	$0.378\|0\rangle + 0.378\|1\rangle + (-0.3405 + 0.164j)\|2\rangle + (-0.0841 + 0.3685j)\|3\rangle$ $+(-0.3405 - 0.164j)\|4\rangle + (-0.0841 + 0.3685j)\|5\rangle + (-0.3405 + 0.164j)\|6\rangle$
P_1	$0.378\|0\rangle + (0.2357 + 0.2955j)\|1\rangle + 0.378\|2\rangle + (-0.3405 - 0.164j)\|3\rangle$ $+(0.2357 - 0.2955j)\|4\rangle + (0.2357 - 0.2955j)\|5\rangle + (-0.3405 - 0.164j)\|6\rangle$
P_2	$0.378\|0\rangle + (-0.0841 - 0.3685j)\|1\rangle + (-0.0841 + 0.3685j)\|2\rangle + (-0.0841 - 0.3685j)\|3\rangle$ $+0.378\|4\rangle + (0.2357 + 0.2955j)\|5\rangle + (0.2357 + 0.2955j)\|6\rangle$
P_3	$0.378\|0\rangle + (-0.3405 + 0.164j)\|1\rangle + (0.2357 + 0.2955j)\|2\rangle + (0.2357 + 0.2955j)\|3\rangle$ $+(-0.3405 + 0.164j)\|4\rangle + 0.378\|5\rangle + (0.2357 - 0.2955j)\|6\rangle$
P_4	$0.378\|0\rangle + (-0.0841 - 0.3685j)\|1\rangle + (-0.0841 + 0.3685j)\|2\rangle + (-0.0841 - 0.3685j)\|3\rangle$ $+0.378\|4\rangle + (0.2357 + 0.2955j)\|5\rangle + (0.2357 + 0.2955j)\|6\rangle$
P_5	$0.378\|0\rangle + 0.378\|1\rangle + (-0.3405 + 0.164j)\|2\rangle + (-0.0841 + 0.3685j)\|3\rangle$ $+(-0.3405 - 0.164j)\|4\rangle + (-0.0841 + 0.3685j)\|5\rangle + (-0.3405 + 0.164j)\|6\rangle$

Since r_1 is odd, the message particle and the verification particle are swapped, forming the particle sequence $S_1 = \{|v_1\rangle, |m_1\rangle\}$, which is then sent to participant P_2.

Participants P_2 to P_4 perform similar operations, with each participant applying specific transformations based on their respective r_i values. For participant P_2, with $r_2 = 4 \in (d/2, d)$, the operation results in $|e_2^{(3)}\rangle|6\rangle$. Since r_2 is even, no swap occurs, and the sequence $S_2 = \{|v_2\rangle, |m_2\rangle\}$ is sent to P_3. Participant P_3 applies the operations resulting in $|e_2^{(1)}\rangle|1\rangle$. Because r_3 is odd, the particles are swapped, forming the sequence $S_3 = \{|m_3\rangle, |v_3\rangle\}$. For participant P_4, the operations yield $|4\rangle|e_2^{(3)}\rangle$. Since r_4 is even, no swap occurs, and the sequence $S_4 = \{|m_4\rangle, |v_4\rangle\}$ is sent to P_5.

Finally, for participant P_5, with $r_5 = 2 \in [0, d/2)$, the operations result in $|6\rangle|e_2^{(5)}\rangle$. As r_5 is also even, no swap occurs, and the sequence $S_5 = \{|m_5\rangle, |v_5\rangle\}$ is sent back to TP. The state transformation of the verification particle during this phase is shown in Table 4.

7.3 Security verification stage

TP measures the verification particle $|v_5\rangle$ using the measurement basis $\{|e_l^{(5)}\rangle | l = 0, 1, \ldots, 6\}$ prepared from the message particle. The measurement basis is shown in Table 5. The measured value $l = 2$ matches the initially set value l_0, indicating that the verification is successful.

7.4 Result output stage

The circuit was executed 1024 times, with TP measuring the message particle on the computational basis. The results are shown in Fig. 7, $m_5 = 6$. In the finite field $GF(7)$, the element $m_0 = 4$ has a unique multiplicative inverse, $4^{-1} = 2$. Therefore, the multiplication result is $m_n m_0^{-1} \equiv 6 \times 2 \equiv 5 \bmod 7$.

Table 5 Measurement bases of verification particle

Value of l	Measurement Bases
$l = 0$	$\|0.378\|0\rangle + (-0.0841 - 0.3685j)\|1\rangle + (0.2357 - 0.2955j)\|2\rangle + (-0.3405 + 0.164j)\|3\rangle$ $+(-0.3405 + 0.164j)\|4\rangle + (0.2357 - 0.2955j)\|5\rangle + (-0.0841 - 0.3685j)\|6\rangle$
$l = 1$	$0.378\|0\rangle + (0.2357 - 0.2955j)\|1\rangle + (0.2357 + 0.2955j)\|2\rangle + (0.2357 - 0.2955j)\|3\rangle$ $+0.378\|4\rangle + (-0.3405 - 0.164j)\|5\rangle + (-0.3405 - 0.164j)\|6\rangle$
$l = 2$	$0.378\|0\rangle + 0.378\|1\rangle + (-0.3405 + 0.164j)\|2\rangle + (-0.0841 + 0.3685j)\|3\rangle$ $+(-0.3405 - 0.164j)\|4\rangle + (-0.0841 + 0.3685j)\|5\rangle + (-0.3405 + 0.164j)\|6\rangle$
$l = 3$	$0.378\|0\rangle + (0.2357 + 0.2955j)\|1\rangle + (-0.0841 - 0.3685j)\|2\rangle + (-0.0841 - 0.3685j)\|3\rangle$ $+(0.2357 + 0.2955j)\|4\rangle + 0.378\|5\rangle + (-0.0841 + 0.3685j)\|6\rangle$
$l = 4$	$0.378\|0\rangle + (-0.0841 + 0.3685j)\|1\rangle + 0.378\|2\rangle + (0.2357 + 0.2955j)\|3\rangle$ $+(-0.0841 - 0.3685j)\|4\rangle + (-0.0841 - 0.3685j)\|5\rangle + (0.2357 + 0.2955j)\|6\rangle$
$l = 5$	$0.378\|0\rangle + (-0.3405 + 0.164j)\|1\rangle + (-0.0841 + 0.3685j)\|2\rangle + (-0.3405 - 0.164j)\|3\rangle$ $+(-0.0841 + 0.3685j)\|4\rangle + (-0.3405 + 0.164j)\|5\rangle + 0.378\|6\rangle$
$l = 6$	$0.378\|0\rangle + (-0.3405 - 0.164j)\|1\rangle + (-0.3405 - 0.164j)\|2\rangle + 0.378\|3\rangle$ $+(0.2357 - 0.2955j)\|4\rangle + (0.2357 + 0.2955j)\|5\rangle + (0.2357 - 0.2955j)\|6\rangle$

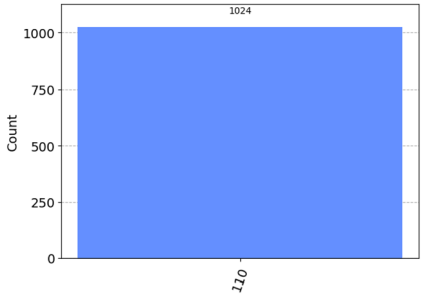

Figure 7 Measurement result of message particle

8 Conclusion

This paper presents a novel quantum secure multiparty multiplication protocol based on single-particle systems, and its effectiveness against both external and internal attacks is thoroughly validated. Performance comparisons and analyses are conducted, demonstrating the protocol's excellent overall performance, and the correctness of the protocol is further confirmed through simulation results. Compared to quantum secure multiparty multiplication protocols that rely on entangled states, the complexity of particle preparation is reduced in the proposed protocol. In contrast to protocols that use decoy particles for security, the requirements for participants' quantum capabilities are lowered, and the consumption of quantum resources is minimized. Moreover, the improved multiplication circuit, used as the foundation for modular multiplication operations, reduces the complexity of the circuits required for the multiplication task.

In this paper, we assume that quantum communication channels are ideally noise-free, which is difficult to achieve in real scenarios. Furthermore, the proposed protocol is not resistant to collusive attacks by $n-1$ dishonest participants, and when the product results of all n participants are accessible, the dishonest participants can cooperate to infer the private data of the remaining honest participant. In future work, we will investigate QSMM protocols that can operate effectively in real quantum channels with noise, and develop strategies to defend against $n-1$-party collusive attacks to ensure the robustness and security of the protocols in practical applications.

Abbreviations
QSMC, Quantum secure multiparty computation; QSMM, Quantum secure multiparty multiplication; QFT, Quantum Fourier transform; CNOT, Controlled not; IQFT, Inverse quantum Fourier transform; AQFT, Approximate quantum Fourier transform; TP, Third party.

Author contributions
XLS and JY co-wrote the paper. YSZ and TW reviewed and suggested revisions.

Funding information
This work was supported by the National Natural Science Foundation of China (Grant No. 62376047), the General Project of Chongqing Natural Science Foundation (Grant No.CSTB2023NSCQ-MSX1093), the Key Project of Science and Technology Research Plan of Chongqing Education Commission (Grant No.KJZD-K202300603), the Henan Key Laboratory of Network Cryptography Technology (Grant No.LNCT2022-A15).

Data Availability
No datasets were generated or analysed during the current study.

Declarations

Competing interests
The authors declare no competing interests.

Received: 2 January 2025 Accepted: 28 April 2025 Published online: 31 December 2025

References
1. Goldreich O, Micali S, Wigderson A. How to play any mental game, or a completeness theorem for protocols with honest majority. In: Providing sound foundations for cryptography: on the work of Shafi Goldwasser and Silvio Micali. New York: Association for Computing Machinery; 2019. p. 307–28.
2. Damgård I, Nielsen JB. Scalable and unconditionally secure multiparty computation. In: Menezes A, editor. Advances in cryptology - CRYPTO 2007. Berlin: Springer; 2007. p. 572–90.
3. Du W, Atallah MJ. Secure multi-party computation problems and their applications: a review and open problems. In: Proceedings of the 2001 workshop on new security paradigms. New York: Association for Computing Machinery; 2001. p. 13–22.
4. Du W, Zhan Z. A practical approach to solve secure multi-party computation problems. In: Proceedings of the 2002 workshop on new security paradigms. New York: Association for Computing Machinery; 2002. p. 127–35.
5. Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. 1994. p. 124–34.
6. Grover LK. Quantum mechanics helps in searching for a needle in a haystack. Phys Rev Lett. 1997;79(2):325–8.
7. Grover LK. Quantum computers can search arbitrarily large databases by a single query. Phys Rev Lett. 1997;79(23):4709–12.
8. Nielsen MA, Chuang IL. Quantum computation and quantum information: 10th anniversary edition. Cambridge: Cambridge University Press; 2010.
9. Crépeau C, Gottesman D, Smith A. Secure multi-party quantum computation. In: Proceedings of the thiry-fourth annual ACM symposium on theory of computing. New York: Association for Computing Machinery; 2002. p. 643–52.
10. Cleve R, Gottesman D, Lo HK. How to share a quantum secret. Phys Rev Lett. 1999;83(3):648–51.
11. Guo GP, Guo GC. Quantum secret sharing without entanglement. Phys Lett A. 2003;310(4):247–51.
12. Zj Z, Zx M. Multiparty quantum secret sharing of classical messages based on entanglement swapping. Phys Rev A. 2005;72(2):022303.
13. Ty W, Wen Q, Gao F, Lin S, Fc Z. Cryptanalysis and improvement of multiparty quantum secret sharing schemes. Phys Lett A. 2008;373(1):65–8.
14. Morimae T, Fujii K. Blind quantum computation protocol in which Alice only makes measurements. Phys Rev A. 2013;87(5):050301.
15. Shi R, Zhang M. Privacy-preserving quantum sealed-bid auction based on Grover's search algorithm. Sci Rep. 2019;9(1):7626.
16. Weinstein YS, Hellberg CS. Effects of symmetries on quantum fidelity decay. Phys Rev E. 2005;71(3):035203.
17. Weinstein YS, Hellberg CS. Energetic suppression of decoherence in exchange-only quantum computation. Phys Rev A. 2005;72(2):022319.
18. Weinstein YS, Chai D, Xie N. Improving ancilla states for quantum computation. Quantum Inf Process. 2016;15(4):1445–53.
19. Shukla C, Alam N, Pathak A. Protocols of quantum key agreement solely using bell states and bell measurement. Quantum Inf Process. 2014;13(11):2391–405.
20. Liu WJ, Chen ZY, Ji S, Wang HB, Zhang J. Multi-party semi-quantum key agreement with delegating quantum computation. Int J Theor Phys. 2017;56(10):3164–74.
21. Cheng ST, Wang CY. Quantum switching and quantum merge sorting. IEEE Trans Circuits Syst I, Regul Pap. 2006;53(2):316–25.
22. Wu W, Zhou G, Zhao Y, Zhang H. New quantum private comparison protocol without a third party. Int J Theor Phys. 2020;59(6):1866–75.
23. Shu H, Yu R, Jiang W, Yang W. Efficient implementation of K-nearest neighbor classifier using vote count circuit. IEEE Trans Circuits Syst II, Express Briefs. 2014;61(6):448–52.
24. Wang Q, Liu J, Li Y, Yu C, Pan S. Quantum bell states-based anonymous voting with anonymity trace. Quantum Inf Process. 2021;20(4):142.

25. Khabiboulline ET, Sandhu JS, Gambetta MU, Lukin MD, Borregaard J. Efficient Quantum Voting with Information-Theoretic Security. arXiv preprint. (2021)
26. Shi R, Mu Y, Zhong H, Cui J, Zhang S. Secure multiparty quantum computation for summation and multiplication. Sci Rep. 2016;6(1):19655.
27. Lv SX, Jiao XF, Zhou P. Multiparty quantum computation for summation and multiplication with mutually unbiased bases. Int J Theor Phys. 2019;58(9):2872–82.
28. Sutradhar K, Om H. Hybrid quantum protocols for secure multiparty summation and multiplication. Sci Rep. 2020;10(1):9097.
29. Sutradhar K, Om H. Secret sharing based multiparty quantum computation for multiplication. Int J Theor Phys. 2021;60(9):3417–25.
30. Zhang L, Song X, Li C, Liu Y. Quantum secure multiparty multiplication based on Lagrange unitary operator. Sci Sin-Phys Mech Astron. 2022;52(6):260311.
31. Li F, Hu H, Zhu S. A (k, n)-threshold dynamic quantum secure multiparty multiplication protocol. Quantum Inf Process. 2022;21(12):394.
32. Lian JY, Ye TY. Hybrid protocols for multi-party semiquantum private comparison, multiplication and summation without a pre-shared key based on d-dimensional single-particle states. EPJ Quantum Technol. 2024;11(1):17.
33. Ivonovic ID. Geometrical description of quantal state determination. J Phys A, Math Gen. 1981;14(12):3241.
34. Wootters WK, Fields BD. Optimal state-determination by mutually unbiased measurements. Ann Phys. 1989;191(2):363–81.
35. Draper TG. Addition on a Quantum Computer. arXiv preprint. arXiv:quant-ph/0008033 (2000)
36. Barenco A, Ekert A, Suominen KA, Törmä P. Approximate quantum Fourier transform and decoherence. Phys Rev A. 1996;54(1):139–46.
37. Cheung D. Improved Bounds for the Approximate QFT. arXiv preprint. arXiv:quant-ph/0403071 (2004)
38. Yang YG, Xia J, Jia X, Zhang H. Comment on quantum private comparison protocols with a semi-honest third party. Quantum Inf Process. 2013;12(2):877–85.
39. Chen XB, Xu G, Yang YX, Wen QY. An efficient protocol for the secure multi-party quantum summation. Int J Theor Phys. 2010;49(11):2793–804.
40. Boyer M, Kenigsberg D, Mor T. Quantum key distribution with classical Bob. Phys Rev Lett. 2007;99(14):140501.
41. Gu J, Lin P, Hwang T. Double C-NOT attack and counterattack on 'three-step semi-quantum secure direct communication protocol'. Quantum Inf Process. 2018;17(7):182
42. Lin PH, Hwang T, Tsai CW. Double CNOT attack on "Quantum key distribution with limited classical Bob". Int J Quantum Inf. 2019;17(02):1975001
43. Wang HW, Tsai CW, Lin J, Huang YY, Yang CW. Efficient and secure measure-resend authenticated semi-quantum key distribution protocol against reflecting attack. Mathematics. 2022;10(8):1241
44. Yang CW, Huang YY, Tsai CW, Lin J. Reflecting attack and improvement of a semi-quantum private comparison protocol with three-particle GHZ-like states. Mod Phys Lett A. 2024;39(37):2450175.

Publisher's note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com