Liu and Danso *EPJ Quantum Technology* (2026) 13:38
https://doi.org/10.1140/epjqt/s40507-026-00482-2

EPJ.org

EPJ Quantum Technology

RESEARCH Open Access

An efficient and secure multi-party convex hull protocol using quantum secret commitment

Wen-Jie Liu[1,2*] and Solomon Danquah Danso[1]

*Correspondence: wenjiel@163.com
[1]School of Computer Science, Nanjing University of Information Science and Technology, No. 219 Ningliu Road, Nanjing, 210044, Jiangsu, China
[2]Engineering Research Center of Digital Forensics, Ministry of Education, No. 219 Ningliu Road, Nanjing, 210044, Jiangsu, China

Abstract

Convex hull computation is a fundamental problem in secure multi-party computational geometry (SMCG), classified under secure multi-party computation (SMC) for finding the convex hull of a set of points. Existing quantum solutions largely depend on quantum homomorphic encryption (QHE), which introduces significant computational overhead due to frequent key updates by a trusted third party (TTP). Furthermore, most current protocols lack a mechanism for input commitment, making them vulnerable to post-computation input tampering or denial by the TTP. To overcome these limitations, we propose an efficient convex hull protocol that utilizes quantum secret commitment (QSC) as a more secure alternative to QHE. Our protocol enables a designated party (the committer) to securely commit to input values in a manner that guarantees both binding (no post-hoc alteration) and hiding (input secrecy). We introduce a novel value comparison protocol within an Ideal Quantum K-Party model, ensuring privacy-preserving convex hull computation without reliance on QHE. Rigorous security analysis demonstrates significant improvements in computational efficiency and resilience against quantum adversaries. Our protocol represents a pivotal advancement for quantum-secure multi-party computations and lays the groundwork for scalable, future-proof privacy-preserving geometry in quantum computing contexts.

Keywords: Quantum secret commitment; Quantum value comparison; Convex hull protocol; Secure multi-party computation

1 Introduction

Secure Multi-Party Computation (SMC) enables a group of mutually untrusted parties to jointly compute a function over their private inputs without revealing those inputs to one another. This cryptographic paradigm has proven instrumental in applications requiring collaborative data analysis, decision-making, and verification, where privacy is paramount. With the advent of quantum information science, the classical foundations of SMC have been extended into the quantum realm, giving rise to Quantum Secure Multi-Party Computation (QSMC). By leveraging quantum principles, which include superposition, entanglement, and the no-cloning theorem, QSMC protocols achieve secu-

Springer

rity guarantees unattainable through classical methods, even against quantum-capable adversaries [1]. Within this emerging field, Quantum Secure Multi-Party Computational Geometry (QSMCG) has surfaced as a particularly promising subdomain. QSMCG protocols integrate quantum cryptographic techniques with the intrinsic structural properties of geometric data to facilitate privacy-preserving computation among multiple parties. This fusion not only enhances computational efficiency but also strengthens resistance to quantum-enabled attacks, making it a critical area of exploration in quantum cryptography.

The conceptual roots of multi-party computation trace back to Yao's seminal 1982 work on the so-called "millionaires' problem" [2], which introduced foundational techniques for secure two-party computation. Expanding on these foundations, Atallah [3] formalized Secure Multi-party Computational Geometry (SMCG), extending the scope of SMC to include geometric computations. Atallah's framework addressed key geometric problems such as point inclusion, intersection detection, and convex hull construction, establishing the basis for a rich body of subsequent research. Over the past two decades, a growing corpus of literature has tackled the challenges associated with secure geometric computation. Notable advancements include secure protocols for geometric intersection [4–10], nearest point computation [11–13], and convex hull determination [14–16]. These efforts have laid the foundation for the development of QSMCG, where quantum protocols are now being designed to address similar problems under stronger security assumptions and in the presence of quantum adversaries.

Classical approaches to secure multi-party computational geometry (SMCG) are inherently constrained by their reliance on computational hardness assumptions, which limit scalability and resilience against evolving threats. The emergence of quantum algorithms, most notably Shor's algorithm for integer factorization [17] and Grover's algorithm for unstructured search [18], poses a fundamental threat to the cryptographic primitives underpinning classical SMCG protocols. These developments underscore the urgent need to explore quantum-resilient alternatives, thereby catalyzing the progression toward Quantum Secure Multi-Party Computational Geometry (QSMCG).

Early efforts in this direction began to surface in the mid-2010s. In 2016, Shi et al. [9] introduced a privacy-preserving point-inclusion protocol using phase-encoded quantum private queries. Building on this foundation, Peng et al. [19] proposed a novel quantum phase-coding approach for secure two-party distance computation. However, this scheme was later critiqued by Chen et al. [20], who identified security vulnerabilities, analyzed their origins, and introduced a corrected protocol. Further advancing this line of inquiry, Peng et al. [21] developed a quantum key distribution (QKD)-based protocol tailored for secure distance calculations between two parties.

In 2019, Liu et al. [22] proposed a quantum protocol for two-party geometric intersection, utilizing oracle-based quantum counting algorithms. Despite its technical merit, the approach faces challenges in extending to multi-party scenarios. To address malicious adversaries, Liu et al. [23] subsequently introduced a protocol in 2022 for secure Manhattan distance computation. More recently, Wang and Zhou [24] extended a two-party protocol by Xu et al. [25] into a multi-party quantum convex hull protocol using quantum homomorphic encryption (QHE). While this solution marks an important milestone, it suffers from high computational costs due to frequent key updates managed by a trusted third party (TTP).

A critical vulnerability shared by these protocols is the absence of a robust input commitment mechanism. This deficiency allows adversaries, including a potentially malicious TTP to alter or repudiate inputs post-computation, thereby compromising the protocol's integrity. To address more complex computational challenges, Liu et al. [26] introduced a Quantum Secure Two-Party Scalar Product Protocol (QS2PSP) in 2023, which achieved polynomial-time complexity without compromising security. In 2024, Peng et al. [27] proposed a privacy-preserving protocol for determining point–line relationships, incorporating QKD and one-time pad encryption to ensure input confidentiality.

In this work, we present the first quantum protocol to incorporate a quantum secret commitment (QSC) mechanism into the secure multi-party convex hull computation. Our protocol enforces pre-computation input binding, thereby preventing post-computation input tampering or repudiation. Given private point sets distributed across multiple participants, our approach securely computes the shared convex hull without disclosing individual data, except for the final convex hull points known to all. To achieve this, we introduce a quantum value comparison protocol based on secret commitment, which is then embedded into the convex hull computation. This integration ensures both computational correctness and input security in adversarial quantum environments.

The main contributions of this work are summarized as follows:

1. We propose a quantum-secure multi-party convex hull protocol that leverages quantum secret commitment to ensure input binding, non-repudiation, and privacy preservation throughout the computation.
2. We introduce a commitment-based quantum value comparison primitive and demonstrate its applicability as a core building block for secure geometric computations in the multi-party setting.
3. We provide a formal security and efficiency analysis of the proposed protocol under malicious adversary models, establishing information-theoretic security guarantees and polynomial-time computational and communication complexity without reliance on computational hardness assumptions.

The rest of the paper is organized as follows: Sect. 2 provides the preliminary knowledge on quantum secret commitment, addictive secret sharing and ideal quantum secret commitment with abort. In Sect. 3, we first proposed a value comparison protocol based on quantum secret sharing in detailed description and extend it into a secure multi-party convex hull protocol. In Sect. 4 we provide the correctness, security and efficiency analysis of our protocol. The conclusion is in Sect. 5

2 Preliminary knowledge

2.1 Quantum secret commitment

A Quantum Secret Commitment (QSC) protocol enables a party, known as the committer, to securely commit to a secret value while ensuring that the committed value cannot be altered subsequently. This protocol guarantees both bindingness and secrecy through the use of quantum states and quantum encryption. The QSC protocol comprises three primary algorithms: the commitment algorithm, the reveal algorithm, and the verification algorithm. These algorithms collectively provide a robust framework for secure and verifiable secret commitment in quantum settings, ensuring privacy and authenticity. The detailed process is as follows:

i. **Commitment Stage:** The sender, Alice, generates a secret key sk and a commitment key ck using a predefined key generation algorithm. The commitment key is used to commit to a secret value s while hiding it from the verifier. The commitment is computed as:

$$C_c(ck, s) \rightarrow \text{commitment value } C. \tag{1}$$

ii. **Quantum Commitment:** Alice uses the quantum commitment algorithm C_q to encode the secret s into a quantum state σ, which is entangled or encoded based on the commitment key ck. The quantum commitment state is:

$$C_q(ck, s) \rightarrow |\psi_s\rangle\,. \tag{2}$$

iii. **Reveal Stage:** When Alice chooses to reveal the secret, she provides the commitment key ck and the corresponding quantum state σ_s to Bob, the verifier. Bob then performs quantum measurements or operations M on the quantum state $|\psi_s\rangle$ and verifies that it matches the committed value.

$$M(|\psi_s\rangle) \rightarrow \text{revealed value } s. \tag{3}$$

iv. **Verification:** Bob verifies the consistency of the revealed quantum state with the original commitment using the verification algorithm V_v, checking if the revealed value matches the committed secret:

$$V_v(ck, C_c(ck, s)) \rightarrow s. \tag{4}$$

2.2 Additive secret sharing

Additive Secret Sharing (ASS) is a cryptographic technique used to securely distribute a secret among multiple participants in such a way that no individual participant knows the entire secret, but a group of participants can reconstruct it. The fundamental concept of ASS is that the secret S is divided into shares, and these shares are combined using an additive operation (often modulo a prime number) to recover the original secret. To achieve this, the secret S is split into n parts, denoted as $S_1, S_2, \ldots, S_n$, with each part being a randomly generated value, except for one of the shares, which is derived by ensuring that the sum of all shares equals the secret. The reconstruction of the secret is achieved by summing the shares together modulo a prime number p, which ensures that the secret can be recovered if enough shares are combined.

For example, in a simple scheme with three participants, the secret S could be split into two random numbers r_1 and r_2, and the third share would be computed as

$$S_3 = -(r_1 + r_2) + S \pmod{p}. \tag{5}$$

When all the participants share their portions, the sum of their shares will provide the secret. This method ensures that the secret remains hidden unless the requisite number of participants collaborate to pool their shares. Since each share contains no information about the secret on its own, the scheme is secure in that it keeps the secret concealed from

any subset of participants smaller than the required threshold. Additive Secret Sharing is known for its simplicity, relying on basic modular arithmetic to achieve secure sharing and efficient reconstruction.

2.3 Ideal quantum secret commitment with abort

In the context of secure quantum communication, we define an *Ideal Quantum Secret Commitment with Abort* protocol as follows: Let $\mathcal{S}$ represent the secret value to be committed by a player (committer) in the protocol. The committer encodes the secret in a quantum state, which is sent to a verifier (or trusted third party, TTP), ensuring the hiding and binding properties of the commitment. The quantum circuit used for the commitment must satisfy the following criteria: $\mathcal{S} \rightarrow |\psi_{\mathcal{S}}\rangle$, where $|\psi_{\mathcal{S}}\rangle$ is a quantum state representing the secret committed by the committer. The protocol follows the steps outlined below:

1. **Commitment Phase:** The committer i prepares a quantum state $|\psi_{\mathcal{S}}\rangle$, corresponding to the secret $\mathcal{S}$, and sends this state to the trusted third party (TTP). This ensures the state is securely transmitted without revealing any information about $\mathcal{S}$ to the verifier (hiding property).
2. **Commitment Verification:** The TTP stores the quantum state $|\psi_{\mathcal{S}}\rangle$ and ensures that it remains unchanged until the unveiling phase, thus maintaining the binding property of the commitment (i.e., the committer cannot alter the committed value).
3. **Abort Condition:** At any point in the protocol, the committer can choose to abort the process by sending an abort signal to the verifier. If the committer chooses to abort, the commitment is invalidated, and the verifier receives a message indicating the commitment did not take place.
4. **Reveal Phase:** When the committer decides to reveal the secret, they send the necessary information (such as measurement results) to the verifier to enable them to verify the commitment. The TTP assists in ensuring the reveal process is correct.
5. **Verification of the Commitment:** The verifier checks the received information against the stored quantum state $|\psi_{\mathcal{S}}\rangle$. If the secret $\mathcal{S}$ corresponds to the previously committed value, the commitment is verified successfully.
6. **Abort by Verifier:** If any party detects a deviation or an attempt to cheat by the committer, the verifier may send an abort message to all participants, invalidating the commitment and ensuring fairness in the protocol.

This protocol ensures that the commitment is both *binding* (the committer cannot change the committed secret after sending it) and *hiding* (the verifier cannot learn anything about the secret until it is revealed). The abort mechanism ensures that the commitment can be invalidated at any stage if a player deviates from the protocol, guaranteeing the security of the overall commitment process.

3 Proposed protocols

3.1 Value comparison protocol based on quantum secret commitment (VCPQSC)

In this section, we design a quantum comparison protocol based on the *Ideal Quantum Secret Commitment with Abort* model. We assume there are k players, each holding a secret input $x_i \in \mathbb{Z}_p$ for some prime p. The players aim to compare their inputs to determine the order relations (less than, greater than, or equal) among all pairs of inputs while ensuring the hiding, binding, and abort properties of the commitment. The protocol enables n players $\{P_1, \ldots, P_n\}$ to securely determine pairwise orderings of their inputs $\{x_1, \ldots, x_n\} \subseteq \mathbb{Z}_p$

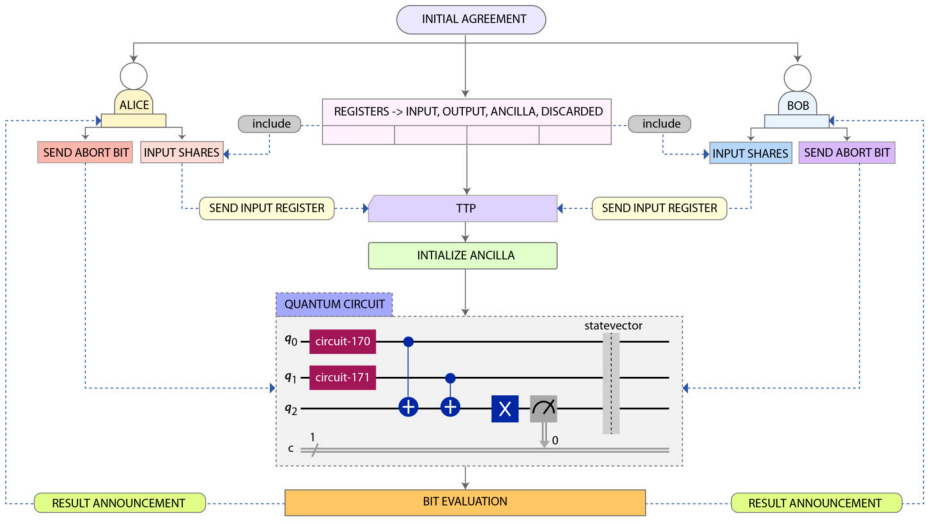

Figure 1 Process flow of VCPQSC Protocol

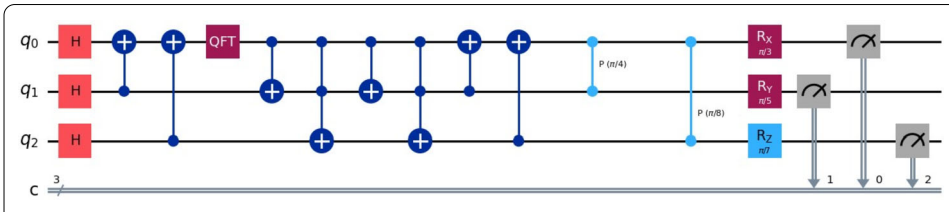

Figure 2 Quantum Circuit of the VCPQSC Protocol

using quantum secret commitment (QSC) and quantum comparison circuits. It consists of seven phases: Commitment → Verification → Circuit Execution → Abort → Reveal → Output → Final Verification and Fig. 1 depicts the graphical protocol design and Fig. 2 depicts the Comparison circuit. The protocol proceeds as follows:

Step 1: Commitment Phase - Let P_i, where $i \in \{1, 2, 3, \ldots, n\}$, denote the n participating players, and let $x_i \in \mathbb{Z}_p$ be the input value of player P_i. Each player P_i performs a secret - sharing operation by splitting x_i into k binary secret shares $s_{i1}, s_{i2}, \ldots, s_{ik}$ such that:

$$x_i = \sum_{j=1}^{k} s_{ij} \pmod{p}, \quad s_{ij} \in \{0, 1\} \tag{6}$$

Player P_i then employs the BB84 scheme. For each share s_{ij}, P_i encodes it into a qubit $|\psi_{s_{ij}}\rangle$ as follows:

$$|\psi_{s_{ij}}\rangle = \begin{cases} |0\rangle_v, & s_{ij} = 0 \text{ (vertical polarization)}, \\ |1\rangle_h, & s_{ij} = 1 \text{ (horizontal polarization)}, \end{cases} \tag{7}$$

Player P_i commits to $|\psi_{s_{ij}}\rangle$ by sending it to a trusted third party (TTP) in a randomly chosen basis $B_{ij} \in \{\mathcal{Z}, \mathcal{X}\}$, where $\mathcal{Z} = \{|0\rangle, |1\rangle\}$ and $\mathcal{X} = \{|+\rangle, |-\rangle\}$, and $|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$. The TTP stores the qubits in input registers R_{ij}^{in} and records the basis B_{ij} for each commitment.

Step 2: Commitment Verification The TTP verifies no tampering by randomly selecting $\lambda\%$ of commitments to measure in their declared basis B_{ij}. For a commitment $|\psi_{s_{ij}}\rangle$ in basis $\mathcal{Z}$, measurement yields s_{ij} with probability 1. In basis $\mathcal{X}$, measurement yields:

$$\langle \pm|\psi_{s_{ij}}\rangle = \frac{1 \pm (-1)^{s_{ij}}}{\sqrt{2}} \implies \text{outcome } s'_{ij} = \begin{cases} 0, & \text{if } |+\rangle \text{ measured,} \\ 1, & \text{if } |-\rangle \text{ measured.} \end{cases} \tag{8}$$

If $s'_{ij} \neq s_{ij}$ in any verified commitment, the TTP aborts the protocol (see Step 4).

Step 3: Quantum Comparison Circuit Execution The TTP initializes an ancillary register $R_{\text{anc}} = |0\rangle^{\otimes m}$, where $m = \mathcal{O}(k)$ is sufficient to reversibly compute modular addition and comparison over $\mathbb{Z}_p$. For each ordered pair of distinct players (P_i, P_j), the TTP constructs a reversible quantum comparison circuit U_{cmp} acting on the tensor product of the committed input registers and the ancillary register:

$$U_{\text{cmp}} : R_i^{\text{in}} \otimes R_j^{\text{in}} \otimes R_{\text{anc}} \longrightarrow R_i^{\text{in}} \otimes R_j^{\text{in}} \otimes R_{\text{anc}}. \tag{9}$$

The circuit coherently computes the comparison predicate

$$f(s_i, s_j) = \begin{cases} 1 & \text{if } \sum_{\ell=1}^{k} s_{i\ell} < \sum_{\ell=1}^{k} s_{j\ell} \pmod p \\ -1 & \text{if } \sum_{\ell=1}^{k} s_{i\ell} > \sum_{\ell=1}^{k} s_{j\ell} \pmod p \\ 0 & \text{if } \sum_{\ell=1}^{k} s_{i\ell} = \sum_{\ell=1}^{k} s_{j\ell} \pmod p, \end{cases} \tag{10}$$

Operationally, U_{cmp} is implemented using reversible modular addition and subtraction circuits composed of CNOT and Toffoli gates, followed by a sign-extraction subroutine that stores the comparison result in a designated ancilla qubit. The circuit realizes the transformation

$$U_{\text{cmp}} |s_i\rangle |s_j\rangle |0\rangle^{\otimes m} \mapsto |s_i\rangle |s_j\rangle |f(s_i, s_j)\rangle |0\rangle^{\otimes (m-1)}, \tag{11}$$

while leaving the original input registers unmeasured and unchanged. This deterministic comparison procedure is executed for all $\binom{n}{2}$ distinct player pairs. Since the comparison predicate is computed reversibly with unit success probability, no amplitude amplification or unstructured quantum search techniques are required.

Step 4: Abort Mechanism The protocol aborts if any player detects inconsistent measurements (Step 2) or the TTP identifies invalid quantum states. Abort signals are classical bits $b_i \in \{0, 1\}$ (0 = proceed, 1 = abort). The protocol continues only if $\sum b_i = 0$.

Step 5: Reveal Phase Players send classical keys $K_i = \{B_{ij}, s_{ij}\}$ to the TTP. The TTP measures each committed qubit R_{ij}^{in} in basis B_{ij} to obtain $\hat{s}_{ij}$. For consistency, define:

$$\Delta_{ij} = \begin{cases} 1, & \hat{s}_{ij} \neq s_{ij}, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

If $\sum_{i,j} \Delta_{ij} > 0$, the TTP aborts (Step 4).

Step 6: Output Calculation For each player P_i, the TTP computes the final output based on the results of the quantum comparison circuit. The output outcome_i is determined by:

$$\text{outcome}_i = \bigwedge_{j \neq i} \text{sign}\left(\sum s_{ij} - \sum s_{jj}\right) \tag{13}$$

where $\text{sign}(x) = 1$ if $x < 0 \pmod p$, $\text{sign}(x) = -1$ if $x > 0 \pmod p$, and $\text{sign}(x) = 0$ if $x = 0 \pmod p$. The results are then sent to the output registers R_i^{out}. The output interpretation remains the same:

$$\text{Output of Player } P_i = \begin{cases} 1, & \text{if } x_i \text{ is less than all other } x_j \\ -1, & \text{if } x_i \text{ is greater than all other } x_j \\ 0, & \text{if } x_i \text{ is equal to one or more other } x_j \end{cases} \tag{14}$$

Step 7: Final Verification
The TTP publishes a hash of all outcomes for public verification. Players can cross-check using their shares to ensure outcome_i aligns with x_i.

3.2 Secure multi-party convex hull protocol based on VCPQSC

In the context of secure multi-party computation (MPC), a convex hull protocol enables multiple parties to compute the convex hull of a combined set of points without revealing their individual point sets. This section outlines an optimized approach to the convex hull protocol that reduces communication overhead and enhances efficiency by minimizing the number of rounds of communication and the comparison protocol (VCPQSC) invocations. For simplicity, we consider a two-party scenario for this convex hull protocol. The overall protocol flow is summarized in Algorithm 1, while the secure value comparison sub-protocol employed during the comparison stages is detailed in Algorithm 2. The specific steps are as follows:

Step 1: (Initial Point Selection) Let Alice and Bob each hold private point sets, denoted as $S_A = \{p_1, p_2, \ldots, p_m\}$ and $S_B = \{q_1, q_2, \ldots, q_n\}$, respectively. The first phase of the protocol involves the identification of the extreme points within each point set. Specifically, Alice computes the smallest point $p_{\min} \in S_A$ and Bob computes $q_{\min} \in S_B$, utilizing the Y-axis or alternatively, the X-axis as the criterion for extremity. These extreme points are considered as candidates for the convex hull. This computation can be performed concurrently by Alice and Bob, thus reducing the overall computational time for this phase.

Step 2: (Comparison of Extreme Points) Subsequently, Alice and Bob engage in the comparison of their respective extreme points $p_A^{\min}$ and $q_B^{\min}$ via the Value Comparison Protocol (VCPQSC). The goal of this comparison is to evaluate the relative size of the extreme points.

- **If** $p_A^{\min} > q_B^{\min}$, then $q_B^{\min}$ is selected as an extreme point of the entire set $S_A \cup S_B$ and consequently, as a point on the convex hull. A horizontal unit vector, denoted as $\rightarrow (v_0 = q_B^{\min}, v_1 = q_B^{\min})$, is then generated through $q_B^{\min}$ along the X-axis.
- **If** $p_A^{\min} < q_B^{\min}$, then $p_A^{\min}$ is designated as the extreme point of the entire combined set and a point on the convex hull. A horizontal unit vector, denoted as $\rightarrow (v_0 = p_A^{\min}, v_1 = p_A^{\min})$, is generated through $p_A^{\min}$ along the X-axis.
- **If** $p_A^{\min} = q_B^{\min}$, both $p_A^{\min}$ and $q_B^{\min}$ are identified as extreme points of the combined set and as points on the convex hull. A vector $\rightarrow (v_0 = p_A^{\min}, v_1 = q_B^{\min})$ is created connecting $p_A^{\min}$ and $q_B^{\min}$.

This comparison is executed securely and concurrently by both parties, ensuring that only the outcome of the comparison is revealed, with no additional information being disclosed about the respective private point sets of Alice and Bob.

Step 3: (Computation of Cosine Values and Selection of Convex Hull Points) Upon completion of the extreme point comparison, Alice and Bob proceed to compute the cosine values for each convex hull point. Alice calculates the cosine values $\cos_A(l)$ for $l = 1, 2, \ldots, m$, which represent the cosine of the angle between the vectors $\mathbf{v}_i^P$ and $\mathbf{v}_{i-1}^P$ where $\mathbf{v}_i^P$ represents the i-th convex hull point. Simultaneously, Bob computes the cosine values $\cos_B(t)$ for $t = 1, 2, \ldots, n$ for the convex hull points in his set, defined similarly. Both parties select the maximum cosine values, denoted $\cos_A$ and $\cos_B$, from their respective computations. The maximum cosine values are then compared using the (VCPQSC). Based on the outcome of this comparison:

- If $\cos_A > \cos_B$, Alice's point $p_{\max}$ is selected as the convex hull point and is updated to v_{i+1}.
- If $\cos_A < \cos_B$, Bob's point $q_{\max}$ is selected as the convex hull point and is updated to v_{i+1}.
- If $\cos_A = \cos_B$, the protocol transitions to Step 4.

This step ensures secure and efficient convex hull point selection through a single round of communication.

Step 4: (Refinement Using Distance Comparison) After a convex hull point is selected in Step 3, the protocol proceeds with refinement by calculating the distances from the current convex hull points to reference points $p_{\max}$ or $q_{\max}$. Alice computes the distance dis_A from each convex hull point $\mathbf{v}_i$ to $p_{\max}$, while Bob computes the distance dis_B from each convex hull point $\mathbf{v}_i$ to $q_{\max}$. These distances are compared using the (VCPQSC). Based on the outcome of the distance comparison:

- If $\text{dis}_A > \text{dis}_B$, Alice's point $p_{\max}$ is updated to v_{i+1}.
- If $\text{dis}_A < \text{dis}_B$, Bob's point $q_{\max}$ is updated to v_{i+1}.
- If $\text{dis}_A = \text{dis}_B$, the protocol continues to the next iteration of comparison.

Both Alice and Bob perform these distance calculations concurrently, ensuring the refinement phase is executed efficiently.

Step 5: (Iterative Refinement and Termination) The protocol iterates through Steps 3 and 4 until convergence is achieved. Convergence occurs when the last convex point $\mathbf{v}_i$ coincides with the first convex point $\mathbf{v}_1$, thus completing the convex hull. Upon completion, both Alice and Bob hold identical results, with no information regarding each other's private point sets having been revealed.

Algorithm 1 Secure Multi-party Convex Hull Protocol based on VCPQSC (Part I)

Require: Point sets $P_A, P_B \subset \mathbb{R}^2$

Ensure: Partial convex hull initialization

function VCPQSC(a, b)
    // Secure value comparison via quantum secret commitment
    if $a > b$ **then return** 1
    else if $a < b$ **then return** -1
    else return 0
    end if
end function
function CompareExtremePoints(P_A, P_B)
    // Each party locally identifies its minimum x-coordinate point
    $p_{A_{\min}} \leftarrow \arg\min_{p_i \in P_A} p_i[x]$
    $p_{B_{\min}} \leftarrow \arg\min_{p_i \in P_B} p_i[x]$
    $c \leftarrow$ VCPQSC($p_{A_{\min}}[x], p_{B_{\min}}[x]$)
    if $c > 0$ **then**
        $v_0, v_1, e \leftarrow p_{B_{\min}}$
    else if $c < 0$ **then**
        $v_0, v_1, e \leftarrow p_{A_{\min}}$
    else
        $v_0 \leftarrow p_{A_{\min}}$, $v_1 \leftarrow p_{B_{\min}}$
        $e \leftarrow (p_{A_{\min}}, p_{B_{\min}})$
    end if return v_0, v_1, e
end function
function ComputeCosine(v_1, v_2)
    // Computes angular ordering for hull traversal
    return $\frac{v_1 \cdot v_2}{\|v_1\| \cdot \|v_2\|}$
end function

4 Analysis

4.1 Correctness analysis

The secure multi-party convex hull protocol based on VCPQSC effectively constructs the convex hull by securely comparing extreme points, cosine values, and distances between Alice's and Bob's datasets. The initial comparison of extreme points is done by securely evaluating the smallest x-coordinate, ensuring correct selection of the starting convex hull point. Subsequently, cosine values, representing the directional angles of the points, are computed and compared, determining the next point on the convex hull. If cosine values are equal, the protocol proceeds to refine the selection using Euclidean distance comparisons, ensuring that the most relevant points are included in the convex hull. For the correctness of the Convex Hull protocol based on VCPQSC protocol: supposed that Alice and Bob have the datasets $\mathbf{S_A}$ and $\mathbf{S_B}$ respectively, where $\mathbf{S_A} = \{(9,5),(10,7),(1,4)\}$ and $\mathbf{S_B} = \{(2,8),(5,0),(3,6)\}$ as depicted in Fig. 3. Both participants first compute the minimum coordinates of their Y-axis in their sets, indicated as $\mathbf{a_3}$ and $\mathbf{b_2}$. VCPQSC is applied on $\mathbf{a_3}$ and $\mathbf{b_2}$ to find the first convex hull point, which is $\mathbf{b_2}$. With respect to $\mathbf{b_2}$, the maximum cosine values of the participants are computed. The equation for the vector determining

Algorithm 2 Secure Multi-party Convex Hull Protocol based on VCPQSC (Part II)

// Initialize hull with globally extreme point
$H_A \leftarrow [\,], H_B \leftarrow [\,]$
$v_0, v_1, e \leftarrow \textsc{CompareExtremePoints}(P_A, P_B)$
H_A.append(v_0), H_B.append(v_1)
function CompareCosines($\cos_A$, $\cos_B$)
    // Securely selects next hull candidate by angle
    $c \leftarrow$ VCPQSC($\cos_A, \cos_B$)
    if $c > 0$ **then return** 'A', $\cos_A$
    else if $c < 0$ **then return** 'B', $\cos_B$
    else
      return 'Tie', null
    end if
end function
$\cos_A \leftarrow \textsc{ComputeCosine}(v_0, e)$
$\cos_B \leftarrow \textsc{ComputeCosine}(v_1, e)$
$w_c, b_c \leftarrow \textsc{CompareCosines}(\cos_A, \cos_B)$
if w_c = 'A' **then**
    H_A.append(v_0)
else if w_c = 'B' **then**
    H_B.append(v_1)
end if
return H_A, H_B

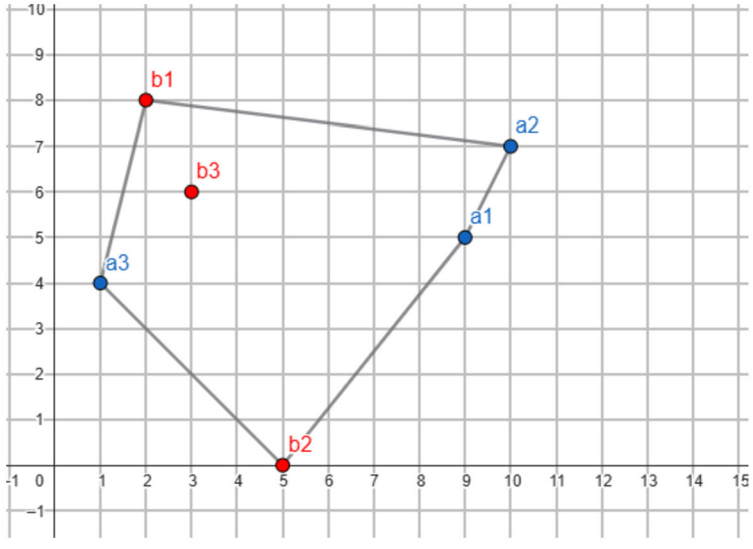

Figure 3 Joint hull of Alice's set $\mathbf{S_A}$ (blue) and Bob's set $\mathbf{S_B}$ (red). The protocol initiates by identifying $\mathbf{b_2}$ as the global minimum Y-coordinate vertex through VCPQSC comparison of local minima ($\mathbf{a_3}$ and $\mathbf{b_2}$)

the maximum cosine value for $\mathbf{S_A}$ is given by:

$$\mathrm{Cos}_{S_A} = \mathrm{MAX}\left\{\mathrm{Cos}\left\langle\overrightarrow{a_1}, \overrightarrow{a_1 b_1}\right\rangle, \mathrm{Cos}\left\langle\overrightarrow{a_1}, \overrightarrow{a_1 b_2}\right\rangle, \mathrm{Cos}\left\langle\overrightarrow{a_1}, \overrightarrow{a_1 b_3}\right\rangle\right\} \tag{15}$$

Table 1 Security Claims and Collusion Analysis Summary

Attack Type	Protection Mechanism	Detection Probability
Intercept-Resend	Random basis + verification	$1 - (3/4)^{\lambda N}$
Replay/Forgery	Reveal + Δ_{ij} check	Always detectable
Malicious Players	Consistency enforced (QSC)	Always detectable
Collusion Attack	Independent QSC + Basis Hiding	$1 - (3/4)^{\lambda N}$

which is used to find the similarities between two vectors $\overrightarrow{a}$ and $\overrightarrow{b}$ and is derived from:

$$\cos\theta = \frac{\overrightarrow{a} \cdot \overrightarrow{b}}{||\overrightarrow{a}|| \cdot ||\overrightarrow{b}||} \tag{16}$$

and for $\mathbf{S_B}$, it is given by:

$$\mathrm{Cos}_{S_B} = \mathrm{MAX}\left\{\mathrm{Cos}\left\langle\overrightarrow{a_1}, \overrightarrow{a_1 a_2}\right\rangle, \mathrm{Cos}\left\langle\overrightarrow{a_1}, \overrightarrow{a_1 a_3}\right\rangle\right\} \tag{17}$$

Thus, the maximum cosine values for $\mathbf{S_A}$ and $\mathbf{S_B}$ are $\frac{19}{\sqrt{530}}$ and $\frac{-14}{\sqrt{340}}$ respectively. The VCPQSC is then used to compare $\mathrm{Cos}_{\mathbf{SA}}$ and $\mathrm{Cos}_{\mathbf{SB}}$ selecting $\mathbf{a_1}$ as the new convex hull point. To find the next convex hull points, the protocol sequentially follows this process $\mathbf{b_2} \longrightarrow \mathbf{a_1} \longrightarrow \mathbf{a_2} \longrightarrow \mathbf{b_1} \longrightarrow \mathbf{a_3} \longrightarrow \mathbf{b_2}$. The entire solution process is robust, and the assurance of quantum secret commitment further strengthens its security.

4.2 Security analysis

The security of the VCPQSC protocol is analyzed in the context of the ideal quantum k-party computation with abort model. We show that, assuming the players adhere to the protocol, no information about the input of any player is leaked to the others, and the protocol is secure even in the presence of malicious players who may attempt to deviate from the protocol. The abort mechanism ensures that any discrepancies detected by the players lead to the cancellation of the protocol, preventing the propagation of incorrect results. We consider these active attacks on our protocol and a summary is provided in Table 1:

1. Intercept-Resend Attack

Definition 1 (Intercept-Resend Attack) An *intercept-resend attack* is a strategy wherein an adversary $\mathcal{A}$ intercepts quantum states transmitted over a quantum channel, performs a measurement in a randomly selected basis, and subsequently resends newly prepared qubits to the intended recipient based on the observed outcomes.

Assumption 1 The sender chooses each qubit's encoding basis uniformly at random from a set of mutually unbiased bases (e.g., $\{Z, X\}$). The receiver performs measurement in a similarly random basis. A subset of transmitted qubits, denoted as *check bits*, are used solely for eavesdropping detection.

Lemma 1 (Detectability of Intercept-Resend Attack) *Let λN be the number of check qubits used to verify the integrity of the transmission. Then the probability that an intercept-resend attack by $\mathcal{A}$ remains undetected is upper bounded by $\left(\frac{3}{4}\right)^{\lambda N}$.*

Proof Assume each qubit is encoded in either the Z or X basis with equal probability. An adversary $\mathcal{A}$ who intercepts and resends a qubit without knowledge of the original basis has a $\frac{1}{2}$ chance of selecting the correct basis. If $\mathcal{A}$ selects the correct basis, the qubit is undisturbed; otherwise, the measurement collapses the state, and the re-prepared qubit deviates from the original with probability $\frac{1}{2}$. Hence, the probability that a single check qubit passes undetected is:

$$P_{\text{pass}} = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}. \tag{18}$$

Assuming independence across the λN check qubits, the total probability that all pass undetected is:

$$\Pr[\text{undetected}] = \left(\frac{3}{4}\right)^{\lambda N}. \tag{19}$$

This expression decays exponentially in λN, establishing the claim. □

Corollary 1 *By choosing* $\lambda = \frac{c \log n}{N}$ *for a constant* $c > 0$, *the probability of an undetected intercept-resend attack can be made negligible in* n, *i.e.*, $\Pr[\textit{undetected}] \in \mathcal{O}(n^{-c \log(4/3)})$.

2. Man-in-the-Middle Attack

Definition 2 (Man-in-the-Middle (MITM) Attack) A *man-in-the-middle attack* is one where an adversary $\mathcal{A}$ intercepts the quantum communication between two honest parties, performs a measurement on each transmitted qubit in a basis of her choosing, and sends replacement qubits to the intended recipient based on the measurement outcomes.

Assumption 2 Each qubit is prepared by the sender in a basis $B_{ij} \in \{\mathcal{Z}, \mathcal{X}\}$ chosen uniformly at random. The TTP performs measurement either with unknown basis (Step 2) or with basis revealed after transmission (Step 5). The adversary $\mathcal{A}$ lacks knowledge of the basis prior to basis reconciliation.

Threat model The adversary intercepts a qubit $|\psi_{s_{ij}}\rangle$, measures it in a basis $B_E \in \{\mathcal{Z}, \mathcal{X}\}$, and sends a newly prepared qubit $|\psi'_{ij}\rangle$ based on the outcome to the TTP. In Step 2, the TTP chooses measurement basis randomly. In Step 5, the original basis B_{ij} is revealed and the TTP verifies against the expected value.

Lemma 2 (Bound on MITM Attack Success Probability) *Let* k *be the number of qubits subjected to both Step* 2 *and Step* 5 *verification. The probability that an MITM adversary* $\mathcal{A}$ *successfully passes both steps on all* k *qubits is bounded above by*:

$$\Pr[\textit{success}] \leq \left(\frac{9}{16}\right)^{k}. \tag{20}$$

Proof In Step 2, the adversary intercepts a qubit and guesses the encoding basis. With probability $\frac{1}{2}$, the guess is correct, and the correct qubit can be reproduced. With probability $\frac{1}{2}$, the basis is incorrect, and measurement disturbs the state, with only a $\frac{1}{2}$ chance

of producing the correct replacement outcome. Therefore, the total probability of passing Step 2 for one qubit is:

$$P_2 = \frac{1}{2}(1) + \frac{1}{2}\left(\frac{1}{2}\right) = \frac{3}{4}. \tag{21}$$

In Step 5, since the adversary no longer retains the original qubit, and the basis is now revealed, she must guess both the basis and the correct bit value. This yields a maximum success probability of: $P_5 = \frac{3}{4}$. Hence, the joint success probability for one qubit is:

$$P_{\text{MITM}} = P_2 \cdot P_5 = \left(\frac{3}{4}\right)^2 = \frac{9}{16}. \tag{22}$$

Extending to k independently checked qubits, we have:

$$\Pr[\text{undetected MITM}] \leq \left(\frac{9}{16}\right)^k, \tag{23}$$

which is exponentially small in k. □

Corollary 2 *The protocol guarantees information-theoretic security against MITM attacks provided that the number of verification qubits k satisfies* $k = \omega(\log n)$, *rendering the adversary's success probability negligible in the security parameter n.*

3. Input Forgery Attack

Definition 3 (Input Forgery Attack) An *input forgery attack* is an attempt by an adversary to craft a forged quantum state $|\phi_{ij}\rangle \neq |\psi_{s_{ij}}\rangle$ that passes the protocol's verification mechanisms as if it were valid, despite not originating from a legitimate party.

Assumption 3 The adversary lacks knowledge of the encoding basis $B_{ij} \in \{\mathcal{Z}, \mathcal{X}\}$, which is uniformly random and unknown during Step 2. In Step 5, the basis is revealed and measurement is performed against the expected outcome. Verification relies on BB84-style randomness and measurement collapse.

Threat model The adversary prepares a qubit $|\phi_{ij}\rangle$ without knowledge of the true basis B_{ij}, aiming to deceive both the random basis measurement in Step 2 and the deterministic check in Step 5.

Lemma 3 (Bound on Forgery Success Probability) *Let k be the number of verification qubits. Then the probability that an adversary successfully forges k valid-looking qubits is bounded above by*:

$$\Pr[\textit{forgery}] \leq \left(\frac{9}{16}\right)^k. \tag{24}$$

Proof In Step 2, without knowledge of the measurement basis, the adversary's qubit passes the verification with probability at most $\frac{3}{4}$ (as shown in the MITM analysis). In Step 5, even with the basis revealed, the adversary cannot reverse engineer the correct bit, as no

entanglement or history exists. Hence, again the success probability is $\leq \frac{3}{4}$. Combining both:

$$\Pr[\text{single-qubit forgery}] \leq \left(\frac{3}{4}\right)^2 = \frac{9}{16}. \tag{25}$$

For k independent verification qubits:

$$\Pr[\text{total forgery}] \leq \left(\frac{9}{16}\right)^k. \tag{26}$$

□

Corollary 3 *The protocol is unforgeable in the information-theoretic sense. Given that the number of checked qubits k grows superlogarithmically in the security parameter, the adversary's success probability becomes negligible.*

4. Collusion Attack Analysis

We consider a malicious adversarial model in which an arbitrary subset of dishonest players may collude by sharing all classical information and coordinating their quantum strategies. The security arguments established in the preceding attack analyses extend directly to this setting. At the quantum commitment layer, collusion does not provide any advantage in predicting or influencing the randomly chosen encoding bases B_{ij} prior to the verification phase. Since each committed qubit is independently prepared, transmitted, and verified, and the basis information remains information-theoretically hidden until the reveal phase, coordinated strategies among colluding parties cannot asymptotically increase the probability of successful manipulation or forgery.

From the geometric perspective, collusion among k dishonest parties does not enable reconstruction of an honest party's private coordinate set $\mathbf{P}_H$. Each invocation of the VCPQSC protocol reveals only a ternary comparison outcome $\sigma \in \{-1, 0, 1\}$, which constrains the honest point to a half-plane defined by the comparison predicate. Importantly, combining multiple such outcomes across colluding parties does not refine this uncertainty beyond what is obtainable by a single adversary. Consequently, collusion does not amplify information leakage, and the privacy and security guarantees derived for individual adversaries remain unchanged under arbitrary colluding coalitions.

4.3 Efficiency analysis

We analyze the efficiency of the proposed protocol in terms of time and space complexity, focusing on the dominant cryptographic operations required to securely compute the convex hull over private inputs contributed by n parties. In our setting, the primary computational cost arises from repeated invocations of the secure value comparison subprotocol (VCPQSC), which is used to compare geometric quantities such as coordinates, cosine values, and distances. The total time complexity can be decomposed as:

$$T_{\text{total}} = (\text{number of secure comparisons}) \times (\text{cost per comparison}). \tag{27}$$

Each invocation of VCPQSC operates on secret-shared inputs of bit-length k. The comparison is implemented using reversible modular addition and subtraction circuits composed primarily of CNOT and Toffoli gates. The resulting circuit depth is $\mathcal{O}(k^2)$, which

Table 2 Efficiency Comparison Across Protocols

Protocol	Approach	TP	Input Binding	Adversary	Time	Space
Ref [28]	Classical	×	×	None	$\mathcal{O}(n \log n)$	$\mathcal{O}(n)$
Ref [29]	Classical	×	×	None	$\mathcal{O}((m+n)d)$	$\mathcal{O}(n)$
Ref [30]	Classical	×	×	None	$\mathcal{O}(K^{3/2}N^2(\log K))$	$\mathcal{O}(n)$
Ref [24]	QHE	✓	×	Passive	$\mathcal{O}(m^2 \cdot n)$	$\mathcal{O}(n)$
Proposed	QSC	✓	✓	Malicious	$\mathcal{O}(n^{3/2} \cdot (\log n)^2)$	$\mathcal{O}(nk)$

dominates the cost of a single comparison. During the convex hull construction, each hull vertex is selected by securely comparing all remaining candidate points using cosine or distance-based criteria. For planar point sets, the expected number of convex hull vertices is $\mathcal{O}(n^{1/2})$ under standard random input assumptions, which is commonly adopted in secure geometric analysis. For each hull vertex selection, up to $\mathcal{O}(n)$ secure comparisons are required, yielding a total of $\mathcal{O}(n^{3/2})$ VCPQSC invocations.

Combining these two factors, the overall time complexity of the proposed protocol is $T_{\text{total}} = \mathcal{O}(n^{3/2} \cdot k^2)$. Under the standard assumption that the input bit-length satisfies $k = \Theta(\log n)$, this yields a polynomial-time bound of $T_{\text{total}} = \mathcal{O}\left(n^{3/2}(\log n)^2\right)$

The proposed protocol requires $\mathcal{O}(nk)$ qubits to store committed secret shares and ancillary registers during comparison, along with $\mathcal{O}(nk)$ classical memory for basis information and verification data. No intermediate ciphertext expansion is required, in contrast to quantum homomorphic encryption-based approach by Wang and Zhou [24].

The efficiency and security characteristics of the proposed QSC-based protocol are compared against classical and quantum state-of-the-art methods in Table 2. Classical geometric algorithms, such as those presented in [28] and [29], achieve optimal or near-optimal time complexities of $\mathcal{O}(n \log n)$ and $\mathcal{O}((m+n)d)$, respectively. While these methods are highly efficient for local computations, they provide no mechanisms for data privacy or secure multi-party interaction. The approximation approach in [30] reduces the computational burden for large-scale sets (N) but introduces a trade-off in geometric precision. Our protocol, while possessing a higher theoretical complexity of $\mathcal{O}(n^{3/2}(\log n)^2)$, is designed for scenarios where the datasets $\mathbf{S_A}$ and $\mathbf{S_B}$ must remain private, a feature these classical methods cannot support. Compared to the Quantum Homomorphic Encryption (QHE) approach in [24], the proposed QSC protocol offers a significant scalability advantage. The complexity of the QHE method, $\mathcal{O}(m^2 \cdot n)$, scales quadratically with the coordinate bit-length m. This makes it computationally expensive for high-precision data. In contrast, our protocol's complexity is independent of the coordinate range, scaling only with the number of points n. Furthermore, the shift from QHE to Quantum Secret Commitment (QSC) marks a transition from a **passive** to a **malicious** adversary model. By utilizing the "binding" property of QSC, our protocol ensures that no participant can dishonestly alter their input coordinates once the protocol has commenced. This provides a level of integrity and verifiability that is absent in the semi-honest QHE framework.

5 Conclusion

This work addresses critical limitations in existing quantum solutions for secure multi-party convex hull computation by introducing a novel protocol that leverages quantum secret commitment (QSC) over quantum homomorphic encryption (QHE). Our proposed method successfully mitigates the computational overhead associated with frequent key updates in QHE-based schemes and, crucially, incorporates a robust input commitment

mechanism. This mechanism, facilitated by QSC, ensures both the binding and hiding properties of input values, thereby preventing post-computation tampering or denial. Furthermore, the development of a novel value comparison protocol within an Ideal Quantum K-Party model enables privacy-preserving convex hull computation without the need for QHE. The rigorous security analysis presented demonstrates that our protocol offers substantial improvements in computational efficiency and enhanced resilience against quantum adversaries. Ultimately, this research marks a significant step forward for quantum-secure multi-party computations, providing a scalable and future-proof foundation for privacy-preserving geometric operations in the evolving landscape of quantum computing.

Author contributions
W.L provided critical guidance throughout the development of the research design, contributed to the refinement of the methodology, and reviewed and revised the manuscript for intellectual content. S.D.D conceptualized the research idea, conducted the literature review, developed the methodology, performed the simulations, and drafted the manuscript.

Funding information
This work was supported by the Quantum Science and Technology–National Science and Technology Major Project (2021ZD0302901), the Basic Research Program of Jiangsu (BK20231142), the Postgraduate Research and Practice Innovation Program of Jiangsu Province (KYCX25_1663), and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

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

Declarations

Ethics approval and consent to participate
This research did not involve human participants or animals, so ethical approval was not required.

Consent for publication
The authors give their consent for the publication of this manuscript and all its contents, including figures, tables, and any other data contained herein.

Competing interests
The authors declare no competing interests.

Received: 5 August 2025 Accepted: 11 February 2026 Published online: 23 February 2026

References

1. Goyal R. Quantum cryptography: secure communication beyond classical limits. J Quantum Sci Technol. 2024;1(1):1–5.
2. Yao AC. Protocols for secure computations. In: 23rd annual symposium on foundations of computer science (sfcs 1982). IEEE; 1982. p. 160–4.
3. Atallah MJ, Du W. Secure multi-party computational geometry. In: Dehne F, Sack J-R, Tamassia R, editors. Algorithms and data structures. Lect. Notes Comput. Sci. vol. 2125. Berlin: Springer; 2001. p. 165–77.
4. Troncoso-Pastoriza JR, Katzenbeisser S, Celik M, Lemma A. A secure multidimensional point inclusion protocol. In: Proceedings of the 9th workshop on multimedia & security. 2007. p. 109–20.
5. Luo Y-L, Huang L-S, Zhong H. Secure two-party point-circle inclusion problem. J Comput Sci Technol. 2007;22(1):88–91.
6. Pawlik A, Kozik J, Krawczyk T, Lasoń M, Micek P, Trotter WT, Walczak B. Triangle-free geometric intersection graphs with large chromatic number. Discrete Comput Geom. 2013;50(3):714–26.
7. Ye Y, et al. Efficient secure protocols to determine whether a point is inside a convex hull. In: Proceedings of the 2009 international symposium on information engineering and electronic commerce. 2009. p. 100–5.
8. Erlebach T, Jansen K, Seidel E. Polynomial-time approximation schemes for geometric graphs. In: Proceedings of the twelfth annual ACM-Siam symposium on discrete algorithms, SODA '01, USA. SIAM; 2001. p. 671–9.
9. Shi RH, Mu Y, Zhong H, Cui J, Zhang S. Privacy-preserving point-inclusion protocol for an arbitrary area based on phase-encoded quantum private query. Quantum Inf Process. 2017;16:1–9.
10. Li Z-X, Yang Q, Feng B, Liu W-J. Quantum privacy-preserving two-party circle intersection protocol based on phase-encoded query. Int J Theor Phys. 2023;62(7):138.
11. Tao Y, Yi K, Sheng C, Kalnis P. Efficient and accurate nearest neighbor and closest pair search in high-dimensional space. ACM Trans Database Syst. 2010.
12. Li C, Ni R. Derivatives of generalized distance functions and existence of generalized nearest points. J Approx Theory. 2002;115:44–55.

13. Corral A, Manolopoulos Y, Theodoridis Y, Vassilakopoulos M. Algorithms for processing k-closest-pair queries in spatial databases. Data Knowl Eng. 2004;49(1):67–104.
14. Qi W, Yonglong L, Liusheng H. Privacy-preserving protocols for finding the convex hulls. In: ARES 2008-3rd int. conf. availab. secur. reliab. proc. 2008.
15. Assarf B, Gawrilow E, Herr K, Joswig M, Lorenz B, Paffenholz A, Rehn T. Computing convex hulls and counting integer points with polymake. Math Program Comput. 2017;9:1–38.
16. Löffler M, Van Kreveld M. Largest and smallest convex hulls for imprecise points. Algorithmica. 2010;56:235–69.
17. Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th annual symposium on foundations of computer science. 1994. p. 124–34.
18. Grover LK. Quantum mechanics helps in searching for a needle in a haystack. Phys Rev Lett. 1997;79:325–8.
19. Peng Z, Shi R, Zhong H, Cui J, Zhang S. A novel quantum scheme for secure two-party distance computation. Quantum Inf Process. 2017;16:1–12.
20. Chen B, Yang W, Huang L. Cryptanalysis and improvement of the novel quantum scheme for secure two-party distance computation. Quantum Inf Process. 2019;18:1–14.
21. Peng Z, Shi R, Wang P, Zhang S. A novel quantum solution to secure two-party distance computation. Quantum Inf Process. 2018;17:1–12.
22. Liu W, Xu Y, Yang JCN, Yu W, Chi L. Privacy-preserving quantum two-party geometric intersection. Comput Mater Continua. 2019;60:1237–50.
23. Liu X, Liu XM, Zhang RL, Luo D, Xu G, Chen XB. Securely computing the Manhattan distance under the malicious model and its applications. Appl Sci. 2022;12(22):11705.
24. Wang C, Zhou RG. Secure multi-party convex hull protocol based on quantum homomorphic encryption. Quantum Inf Process. 2023;22:24.
25. Xu G, Yun F, Chen XB, Xu S, Wang J, Shang T, Chang Y, Dong M. Secure multi - party quantum summation based on quantum homomorphic encryption. Intell Autom Soft Comput. 2022;34:531–41.
26. Liu WJ, Li ZX. Secure and efficient two - party quantum scalar product protocol with application to privacy - preserving matrix multiplication. IEEE Trans Circuits Syst I, Regul Pap. 2023;70(11):4456–69.
27. Peng ZW, Shi RH, Ding R, Zhang FF. Novel quantum solutions to privacy - preserving point - line relation determination. Phys Scr. 2024;99,045113.
28. Pérez-Lantero P, Seara C, Urrutia J. Rectilinear convex hull of points in 3d and applications. J Glob Optim. 2024;90(2):551–71.
29. Zhu Y, Huang L, Yang W, Li D, Li L, Luo Y, Dong F. Privacy-preserving approximate convex hulls protocol. In: 2009 first international workshop on education technology and computer science. vol. 2. 2009. p. 208–14.
30. Sartipizadeh H, Vincent TL. Computing the approximate convex hull in high dimensions. 2016.

Publisher's note