Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

QuantumShield: A Quantum-Secured Blockchain Consensus Framework (QS-BFT) for Next-Generation Consumer Applications

Md Farhad Kabir
University of Southern California
Abdullah Rakib Akand
Asian University of Bangladesh
Md Sharif Robbani
Western Illinois University
Ahmed Faizul Haque Dhrubo
North South University
Mohammad Abdul Qayum
North South University
Sufia Zareen
Campbellsville University
Mithila Arman
armanmithila782@gmail.com

Brac University

Research Article

Keywords: quantum secured, encrypted, security

Posted Date: March 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9084418/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

QuantumShield: A Quantum-Secured Blockchain Consensus Framework (QS-BFT) for Next-Generation Consumer Applications

Md Farhad Kabir[1], Abdullah Rakib Akand [2], Md Sharif Robbani [3], Ahmed Faizul Haque Dhrubo[4], Mohammad Abdul Qayum[5], Sufia Zareen [6], Mithila Arman [7*]

1 Marshall School of Business, University of Southern California, Los Angeles, California, USA
2 Department of Computer Science and Engineering, Asian University of Bangladesh, Dhaka, 1341, Bangladesh
3 School of Engineering & Technology, Western Illinois University, Macomb, Illinois, USA
4, 5 Department of Electrical and Computer Engineering, North South University, Dhaka, 1229, Bangladesh
6 Master's in Information Technology and Management, Campbellsville University, USA
7 Department of Computer Science and Engineering, BRAC University, Dhaka, 1212, Bangladesh

* armanmithila782@gmail.com

Abstract

The rapid advancement of quantum computing poses escalating threats to the cryptographic foundations of blockchain consensus mechanisms deployed in consumer applications. This paper proposes QS-BFT (Quantum-Secured Byzantine Fault Tolerance), a novel quantum-resistant blockchain consensus framework designed to protect consumer-oriented blockchain systems against adversaries operating under an *information-theoretically unbounded* threat model—that is, adversaries with unlimited classical and quantum computational resources against whom only information-theoretic security (ITS) provides meaningful protection. This is a strictly stronger threat model than the computational security (CS) model assumed by standard post-quantum cryptographic schemes such as lattice-based signatures. QS-BFT integrates Quantum Key Distribution (QKD) networks to address the fundamental vulnerabilities of classical public-key digital signature schemes. By combining QKD with a multilinear hash function family, the framework introduces the MH-USS (Multilinear Hash Unconditionally Secure Signature) scheme, whose signatures are provably unforgeable, non-repudiable, and transferable under ITS, while remaining compatible with deployment on existing consumer devices. To overcome the throughput limitations of classical Byzantine Fault Tolerance approaches such as PBFT, QS-BFT introduces a dual-mode consensus strategy comprising fast mode and standard mode, and permits nodes to vote on empty blocks, thereby eliminating the need for computationally expensive view-change processes. Formal proofs establish the safety and liveness of the protocol under the ITS threat model. Performance evaluations on a local testbed demonstrate that QS-BFT achieves substantially higher throughput and lower latency than PBFT across all tested network configurations. This work advances quantum-resistant blockchain technology by achieving a principled balance between unconditional cryptographic security and practical consensus efficiency.

Table 1. Nomenclature: acronyms and key terms used in this paper.

Term / Acronym	Definition
BFT	Byzantine Fault Tolerance: the ability of a distributed system to reach consensus despite up to f nodes behaving arbitrarily.
PBFT	Practical Byzantine Fault Tolerance: the classical BFT consensus protocol by Castro and Liskov (1999), with $O(N^2)$ normal-case and $O(N^3)$ view-change message complexity.
QS-BFT	Quantum-Secured Byzantine Fault Tolerance: the dual-mode quantum-resistant BFT consensus protocol proposed in this paper.
QKD	Quantum Key Distribution: a quantum-cryptographic protocol that distributes information-theoretically secure symmetric keys between parties using quantum states.
MH-USS	Multilinear Hash Unconditionally Secure Signature: the QKD-based unconditionally secure signature scheme proposed in this paper, built on the multilinear hash function family.
USS	Unconditionally Secure Signature: a digital signature scheme whose security holds against computationally unbounded adversaries, relying on information-theoretic rather than computational hardness assumptions.
ITS	Information-Theoretic Security: security that holds regardless of the adversary's computational power, as opposed to computational security which relies on hardness assumptions.
CS	Computational Security: security that holds only against adversaries bounded by polynomial time, relying on unproven hardness assumptions (e.g., integer factorization).
Semi-QKD	Semi-Quantum Key Distribution: a QKD variant in which only one communicating party requires full quantum capability; the other party operates classically.
QSYAC	Quantum-Secured Yet Another Consensus: the LC blockchain consensus protocol using Toeplitz-based signatures and QKD, requiring $N = 4f + 1$ nodes.
TPS	Transactions Per Second: the throughput metric used to evaluate consensus performance.
View-change	A sub-protocol in PBFT that replaces a faulty primary node, incurring $O(N^3)$ message complexity.
Fast mode	The QS-BFT optimistic consensus path activated when all $3f + 1$ nodes vote consistently, completing consensus in a single broadcast round.
Standard mode	The QS-BFT fallback consensus path activated when only $2f+1$ votes are received, proceeding through PRE-COMMIT and PRE-RESOLVE phases.
Empty-block vote ($\perp$)	A vote cast by a node that did not receive a valid proposal, used in QS-BFT to replace the PBFT view-change mechanism.

Introduction

In 2008, Satoshi Nakamoto introduced Bitcoin [1], establishing blockchain technology as a decentralized infrastructure for peer-to-peer digital transactions. Since then, blockchain has evolved into a foundational technology integrating cryptographic primitives, consensus mechanisms, distributed data storage, and peer-to-peer communication, enabling tamper-proof distributed systems across a wide range of application domains. The blockchain architecture is typically organized into four layers: the user layer, service layer, core layer, and foundational layer [2], where the foundational and core layers provide the basic operating environment and consensus services. Blockchain systems rely on digital signatures to protect data integrity and authenticity, and on hash pointers to link blocks sequentially into an immutable chain.

Current mainstream blockchain systems, including Bitcoin and Ethereum [3], employ classical cryptographic algorithms and consensus mechanisms to maintain stability, security, and availability. However, the security of these systems rests on *computational security* (CS) assumptions— specifically, the intractability of integer factorization and the discrete logarithm problem—that are known to be violated under quantum computing attacks. Blockchain systems employing classical public-key signature schemes are directly vulnerable to Shor's algorithm [4], while proof-of-work consensus mechanisms face amplified 51% attack risks due to the quadratic speedup offered by Grover's algorithm [5]. Post-quantum cryptographic (PQC) schemes such as lattice-based or hash-based signatures offer resistance under the CS model by replacing computationally hard problems with ones believed to resist known quantum algorithms. However, PQC schemes remain *computationally secure*: their security guarantees evaporate if efficient quantum algorithms for the underlying hard problems are eventually discovered. By contrast, *information-theoretic security* (ITS) provides unconditional guarantees that hold against adversaries with unbounded computational resources, as realized in this work through QKD-based key distribution. As quantum computing hardware continues to mature, the threat to deployed blockchain infrastructure becomes increasingly concrete, making the design of ITS-secured blockchain systems an urgent research priority.

This paper addresses the quantum threat by proposing a comprehensive ITS-secured blockchain consensus framework. At the network level, the system implements a quantum-classical two-layer peer-to-peer architecture: the quantum network layer distributes cryptographic keys via QKD, while all other communications proceed over the classical network layer. At the cryptographic layer, the framework introduces the MH-USS signature scheme, which combines QKD-distributed keys with a multilinear hash function family to provide ITS-level, vote-compatible signatures. At the consensus layer, the paper proposes the QS-BFT consensus mechanism, which improves upon PBFT [6] by dynamically selecting between fast and standard consensus modes and replacing the costly view-change process with an empty-block voting mechanism, thereby reducing communication overhead and improving throughput.

Threat model. The adversary is assumed to possess unbounded classical and quantum computational resources (ITS setting). QKD-based key distribution provides information-theoretic security against such adversaries, ensuring that distributed key material cannot be recovered even by a computationally unbounded attacker. The parametric hash function resists quantum attacks by grounding its collision-resistance in the algorithmic unsolvability of Hilbert's Tenth Problem, providing resistance against both Shor's and Grover's algorithms. Byzantine nodes may behave arbitrarily, including sending conflicting, delayed, or forged messages, but are assumed to number at most $f < N/3$ out of N total nodes.

Contributions. The main contributions of this paper are as follows:

1. We propose MH-USS, a novel ITS signature scheme based on QKD and the multilinear hash function family, achieving approximately 8× faster signing and verification than existing Toeplitz-based alternatives, without sacrificing information-theoretic security guarantees.

2. We design QS-BFT, a dual-mode quantum-resistant BFT consensus mechanism that eliminates view-change overhead and achieves $O(N^2)$ message complexity in the common case.

3. We provide formal proofs of the unforgeability, non-repudiation, and transferability of MH-USS under the ITS threat model, as well as formal safety and liveness proofs for QS-BFT.

4. We implement and evaluate QS-BFT against PBFT on a local testbed, demonstrating consistent improvements in both throughput and latency across all tested network sizes.

Related Work

Quantum communication and cryptography have been progressively integrated with blockchain technology to address the security risks posed by quantum computing. An early quantum currency protocol [8] was incorporated into cryptocurrency blockchain platforms, and a classical-quantum hybrid payment blockchain system [9] combined classical smart contracts with quantum lightning channels. Subsequent work [10] introduced a quantum blockchain exploiting temporal entanglement to achieve immutability without classical hash chaining. Quantum Key Distribution (QKD) is the most mature and commercially deployed branch of quantum cryptography, providing information-theoretically secure point-to-point key establishment [9].

The quantum-secured blockchain (QB) [11] was among the first to adopt a two-layer network architecture, using QKD for symmetric key distribution in the quantum network layer while retaining classical communications for all other interactions. However, the QB blockchain incurred high communication complexity due to its state-machine-replication-based consensus protocol [12], limiting scalability as the number of Byzantine nodes increased. The Logicon-tract (LC) framework [13] extended QB with an unconditionally secure Toeplitz group signature system based on QKD. However, Toeplitz-based hashing exhibits lower computation speed and higher storage requirements compared to mainstream universal hash function families [14, 15]. Furthermore, the LC blockchain adopted the Yet Another Consensus (YAC) mechanism [16, 17], which reduced Byzantine fault tolerance to below $N/4$ and increased overall communication complexity.

Recent work has explored further directions relevant to this paper. Ding et al. [26] investigated intelligent optimization techniques for quantum-classical hybrid systems with implications for consensus protocol efficiency. He et al. [27] examined memetic computing approaches applicable to distributed consensus optimization problems. Gong et al. [28] proposed advanced quantum key generation techniques directly applicable to QKD-based blockchain security. Arumugam et al. [29] provided a survey of quantum computing approaches for next-generation AI-based blockchain systems.

Semi-QKD and deployment feasibility. A notable practical barrier to deploying QKD-based blockchain systems is the requirement that every network node be equipped with quantum hardware. Semi-QKD protocols [30] partially address this constraint by requiring quantum capability at only one end of each communication link; the other party operates entirely with classical equipment. This asymmetry makes semi-QKD considerably more accessible for consumer devices with limited or no quantum hardware.

However, semi-QKD protocols currently achieve lower key rates than full QKD, introduce additional authentication overhead, and have not yet been demonstrated at the scale required for multi-node Byzantine consensus. Moreover, their security proofs impose stricter assumptions on the classical party's behavior compared to full QKD [30]. The present work adopts full QKD to maintain the strongest possible ITS guarantees; however, integrating semi-QKD as a fallback mechanism for resource-constrained nodes is an important direction for future work, and would substantially broaden the range of consumer devices on which QS-BFT could be deployed without hardware upgrades.

The present work addresses the identified limitations by combining the two-layer QKD network model with a more efficient universal hash-based signature scheme and a redesigned consensus protocol that eliminates view-change overhead while maintaining the $N = 3f + 1$ Byzantine fault tolerance threshold and full ITS-level security.

Materials and Methods

System Parameters

Table 2 summarizes all key system parameters used throughout this paper.

Table 2. Key system parameters and their descriptions.

Parameter	Description	Value / Range
N	Total network nodes	$3f + 1$; tested 4–10
f	Maximum Byzantine nodes	$\lfloor (N-1)/3 \rfloor$
K	Number of verifiers in MH-USS	$N - 1$
M	Large prime modulus (MH-USS)	Implementation-specific
δ	Signature acceptance threshold	$\frac{1}{2} + d$, where $d = \frac{1}{3}$
T	Maximum message window time	Configurable timeout
T_1	Fast mode timeout	$T_1 < T_2 - T$
T_2	Standard mode timeout	$T_2 > T_1 + T$

Security Model: Information-Theoretic vs. Computational Security

It is important to distinguish clearly between the two security paradigms relevant to this work. *Computational security* (CS) guarantees that no probabilistic polynomial-time (PPT) adversary can break the scheme with non-negligible probability. Post-quantum schemes such as CRYSTALS-Dilithium (lattice-based) and SPHINCS$^+$ (hash-based) operate in the CS model: they resist known quantum algorithms, but their security would fail if polynomial-time quantum algorithms for their underlying hard problems were discovered in the future.

Information-theoretic security (ITS), by contrast, holds against adversaries with *unbounded* computational resources—no assumption about algorithmic hardness is made. ITS is achievable in practice through QKD, which uses the laws of quantum mechanics (specifically the no-cloning theorem and the disturbance caused by measurement) to distribute keys that are provably inaccessible to any eavesdropper, regardless of their computational power.

Table 3 summarizes this distinction.

QS-BFT targets the ITS model, providing the strongest achievable security guarantee. This comes at the cost of requiring a QKD network infrastructure. For deployments where QKD is unavailable, lattice-based CS schemes represent the next-best alternative.

Table 3. Comparison of security paradigms relevant to quantum-resistant blockchain design.

Property	**Computational Security (CS)**	**Information-Theoretic Security (ITS)**
Adversary model	Bounded by polynomial time	Computationally unbounded
Hardness basis	Unproven hardness assumptions	Laws of quantum mechanics (QKD)
Quantum resistance	Resists known quantum algorithms	Unconditional; no algorithmic assumption
Future-proofness	Fails if hard problem is solved	Guaranteed regardless of future algorithms
Examples	Lattice (Dilithium), hash-based (SPHINCS$^+$)	QKD + USS (MH-USS proposed here)
Key infrastructure	Public-key infrastructure (PKI)	QKD network required
Deployment cost	Low (software only)	Higher (quantum hardware required)

Multilinear Hash Family

Universal hash function families were originally introduced for constructing symmetric encryption signature schemes [13]. The Multilinear Hash Family [18] is a well-studied universal hash function family recognized for its fast computation speed and minimal storage requirements, making it well-suited for high-throughput consensus applications.

Definition 1. *Let $\mathcal{H} = \{h : X \to Y\}$. Given an input string x, a random key r, and a large prime M, represent x as an n-dimensional vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ with $x_i \in \{0, 1, \ldots, M-1\}$, and represent r as an $(n+1)$-dimensional vector $\mathbf{r} = [r_1, r_2, \ldots, r_{n+1}]$ with $r_i \in \{0, 1, \ldots, M-1\}$. The multilinear hash function family is defined as:*

$$h(x) = r_1 + \sum_{i=1}^{n} r_{i+1} x_i \bmod M. \tag{1}$$

When the length of input x is less than n, the remaining positions are padded with 0.

The multilinear hash function family is a strong universal hash function family [13], satisfying:

1. For every $x \in X$ and $y \in Y$:

$$|\{h \in \mathcal{H} : h(x) = y\}| = \frac{|\mathcal{H}|}{|Y|}. \tag{2}$$

2. For every $x_1 \neq x_2 \in X$, $y_1 \neq y_2 \in Y$, and $\tau \in \mathbb{R}^+$:

$$|\{h \in \mathcal{H} : h(x_1) = y_1,\ h(x_2) = y_2\}| \leq \frac{\tau|\mathcal{H}|}{|Y|}. \tag{3}$$

Parametric Hash Function

Classical hash functions such as SHA-256 and Scrypt are widely used to construct blockchain hash pointers and Merkle trees. While these functions resist certain quantum

attacks because no quantum algorithm has solved general NP-hard problems [19], simply doubling the output length addresses only Grover's quadratic speedup and does not provide protection against structural quantum attacks.

This framework adopts the parametric hash function introduced in [16] to construct quantum-resistant hash pointers. The function is specifically designed for blockchain deployment, is collision-resistant, and exhibits a strong avalanche effect: a single-bit change in the input induces an approximately 50% change in the output. Its collision-resistance is grounded in Hilbert's Tenth Problem—the algorithmically unsolvable problem of finding integer roots of integer polynomial systems with more unknowns than equations when the system includes polynomials of degree three or higher [16]—providing theoretical resistance to both classical and quantum algorithmic attacks.

Definition 2. *The parametric hash function is constructed as follows:*

1. *Let $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_n]$ be an n-dimensional coefficient vector and p a prime. Coefficients are defined as:*

$$a_i = \sigma_1 a_1 + \sigma_2 a_2 + \cdots + \sigma_n a_n, \tag{4}$$

   forming a coefficient matrix $\mathbf{A} = [a_1, a_2, \ldots, a_m]$ with $m > n$.

2. *The parametric hash function is:*

$$f(x) = \left[a_1 f_1(x) + a_2 f_2(x) + \cdots + a_m f_m(x)\right] \bmod p, \tag{5}$$

   where each component f_i is, for example:

$$f_i(x) = \left[a_1 x_1 x_2 x_3 + a_2 x_2 x_3 x_4 + \cdots + a_m x_m x_1 x_2\right] \bmod p. \tag{6}$$

MH-USS Signature Scheme

Unconditionally secure quantum signature schemes exploit quantum true randomness, the quantum no-cloning theorem, and one-time pad encryption to achieve ITS-level security [12]. We propose the MH-USS (Multilinear Hash Unconditionally Secure Signature) scheme, which operates entirely within the ITS model: quantum keys are distributed via QKD; messages and signatures are transmitted over the classical network. The scheme is built on a simplified USS framework [20] augmented with the multilinear hash family, producing signatures that are unforgeable, non-repudiable, transferable, and unconditionally secure [21] under the ITS threat model. Compared to the Toeplitz hash function used in the LC group signature scheme [12], the multilinear hash function achieves approximately 8× faster signing and verification, making MH-USS highly suitable for the vote-intensive operations of Byzantine fault tolerance consensus.

The MH-USS scheme consists of three phases: key distribution, signature generation, and signature verification.

Key Distribution

Assume one signer and K verifiers, all connected via a QKD-enabled quantum network.

1. The signer generates K^2 keys $\mathbf{r} = [r_1, r_2, \ldots, r_{K^2}]$ for use in constructing the multilinear hash function $h(x)$.

2. The signer securely distributes to each verifier P_i the sub-key vector $\mathbf{r}_i = [r_{(i-1)K+1}, \ldots, r_{iK}]$ via QKD.

3. Each verifier P_i forwards each element $r_j \in \mathbf{r}_i$ to the corresponding verifier P_j via QKD; verifier P_j records the received key as $r_{i \to j}$.

Signature Generation

1. The signer computes the signature vector using Eq. (1):

$$\mathrm{Sign}(m) = [h_1(m), h_2(m), \ldots, h_{K^2}(m)] = [t_1, t_2, \ldots, t_{K^2}]. \tag{7}$$

2. The signer transmits the message-signature pair $(m, \mathrm{Sign}(m))$ to each verifier over the classical network channel.

Signature Verification

1. Upon receiving $(m, \mathrm{Sign}(m))$, verifier P_j computes $h_{i\to j}(m)$ using the cross-distributed key $r_{i\to j}$. If $t_{i\to j} = h_{i\to j}(m)$, the indicator $T_{i\to j}^{(m)}$ is set to 1; otherwise it is set to 0.

2. The signature is accepted by P_j if and only if:

$$\sum_{i=1}^{K} T_{i\to j}^{(m)} > K\delta, \tag{8}$$

where $\delta = \frac{1}{2} + d$ and $d = \frac{1}{3}$ is the maximum tolerated Byzantine node ratio. Otherwise the signature is rejected.

Security Analysis and Formal Proofs

The adversary in our model is computationally unbounded, operating in the ITS setting. QKD guarantees that the keys $\mathbf{r}$ distributed between the signer and verifiers are information-theoretically secure: no quantum or classical algorithm, regardless of computational power, can recover undistributed key material.

Theorem 1 (Unforgeability of MH-USS)**.** *Under the MH-USS scheme, no computationally unbounded adversary can produce a valid forgery* $\mathrm{Sign}'(m')$ *for any message* m' *without access to the QKD-distributed keys, except with probability at most* $\frac{1}{|Y|}$.

Proof. By Definition 1 (Condition 1), for any fixed message m' and any target hash value $y \in Y$, the fraction of keys $h \in \mathcal{H}$ satisfying $h(m') = y$ is exactly $\frac{|\mathcal{H}|}{|Y|}$. Since the keys are distributed via QKD and are information-theoretically hidden from the adversary, the adversary has no information to distinguish any particular key; its best strategy is therefore a uniformly random guess over Y, which succeeds with probability $\frac{1}{|Y|}$. The MH-USS scheme is thus unforgeable even against an unbounded adversary. □

Theorem 2 (Non-repudiation of MH-USS)**.** *A signer cannot successfully repudiate a valid signature* $\mathrm{Sign}(m)$ *that has been accepted by more than* $K\delta$ *verifiers.*

Proof. By the key distribution protocol, each verifier P_j holds the cross-distributed keys $r_{i\to j}$ for all i. For the signer to successfully repudiate, it would need to have distributed inconsistent key material to different verifiers such that fewer than $K\delta$ verifiers hold keys consistent with the submitted signature. However, the signer generates all K^2 keys deterministically and distributes them via QKD before any message is signed; any post-hoc key inconsistency would require forging keys already delivered over the quantum channel, which is information-theoretically impossible. Therefore, a valid signature that passes the acceptance threshold cannot be repudiated. □

Theorem 3 (Transferability of MH-USS)**.** *A signature* Sign(m) *accepted by verifier* P_j *will be accepted by any other verifier* $P_{j'}$ *that holds the same cross-distributed key material, provided fewer than* f *verifiers are Byzantine.*

Proof. The cross-distribution step ensures that every verifier P_j holds $r_{i\to j}$ for all $i \in \{1, \ldots, K\}$. Since all verifiers apply the same acceptance threshold $K\delta$ to the same key material and signature tuple $(m, \text{Sign}(m))$, acceptance at P_j implies acceptance at all verifiers $P_{j'}$ with high probability. This follows from the strong universal hash property (Definition 1, Condition 2) and the quorum structure $N = 3f + 1$, which ensures that among any $K\delta$ accepting verifiers, at least one is honest. □

QS-BFT Consensus Mechanism Design

PBFT [6] is the foundational strong-consistency BFT consensus protocol. However, PBFT incurs $O(N^3)$ message complexity during view-change operations, severely limiting scalability in large or adversarial networks. Existing optimizations including Zyzzyva [22], SBFT [23], HotStuff [24], and Tendermint [25] improve upon PBFT's communication efficiency but do not address resistance to quantum computing attacks. QS-BFT introduces three key improvements:

- **Quantum-resistant cryptography:** QS-BFT employs the MH-USS signature scheme and the parametric hash function, providing ITS-level resistance against quantum computing attacks, including those based on Shor's and Grover's algorithms.
- **Dual-mode consensus:** QS-BFT partitions the consensus process into fast mode and standard mode. Using speculative execution [12], the primary node collects and aggregates votes. When all $3f + 1$ nodes vote for the same block, fast mode resolves consensus in a single broadcast round, substantially reducing communication latency. When the primary node receives fewer than $3f + 1$ valid votes, the protocol falls back to standard mode, proceeding through commit and resolution phases analogous to PBFT but without the view-change mechanism.
- **View-change elimination:** Rather than initiating a view-change upon receiving invalid or no messages, nodes vote for an empty block ($\perp$). This preserves protocol liveness while avoiding the $O(N^3)$ complexity of PBFT view-change, maintaining round continuity and substantially lowering worst-case communication overhead.

System Model

Network Model

As illustrated in Fig 1, the QS-BFT consensus network is organized into a quantum network layer and a classical network layer, with all nodes communicating via peer-to-peer connections. The quantum network layer is used exclusively for QKD-based key distribution required by the MH-USS signature scheme; all other consensus communications are handled by the classical network layer. The network contains $N = 3f + 1$ nodes, of which at most f are Byzantine [17]. Each node is assigned a unique index $i \in \{0, 1, \ldots, N - 1\}$. The network model is synchronous, supporting state machine replication.

Nodes operate in one of two roles: primary or secondary. A consensus round begins when the client sends a signed request to the primary node and concludes when the client receives a valid response. The primary node is responsible for aggregating client requests into a proposed block and broadcasting signed consensus messages. Secondary

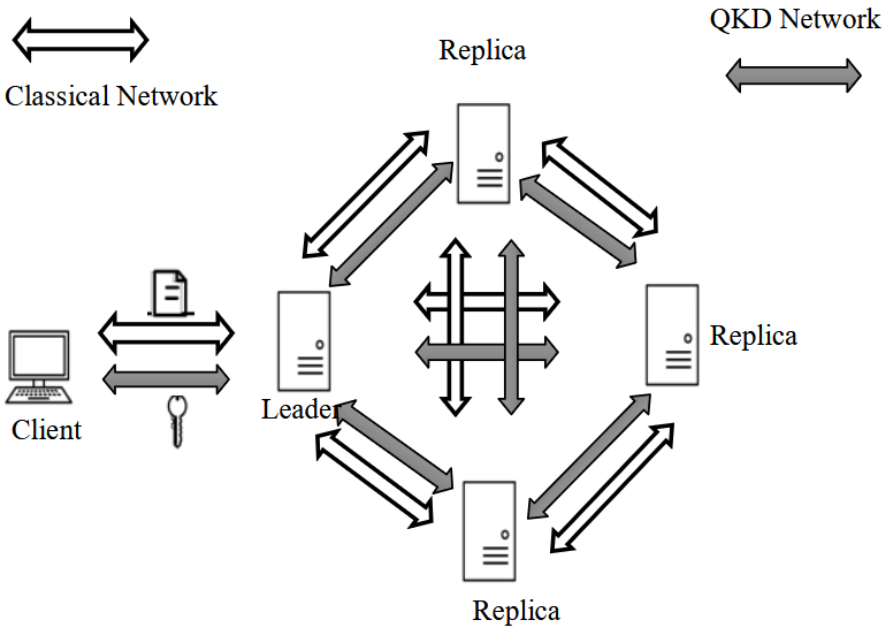


Fig 1. Two-layer peer-to-peer network architecture of QS-BFT. The quantum network layer handles MH-USS key distribution between nodes via QKD, while the classical network layer carries all consensus messages including PRE-PREPARE, VOTE, and FEEDBACK. The primary node index for round r is determined by $L = r \bmod N$.

nodes verify incoming messages, cast votes, execute state transitions, and maintain local logs. The primary node for round r is determined by $L = r \bmod N$.

Message Format

All nodes maintain timers with a maximum message window duration T. Primary node messages take the form:

$$\langle \text{phase},\ r,\ \text{Cert}(\sigma) \rangle_{\sigma_L}, \tag{9}$$

where *phase* identifies the current protocol stage, r is the round number, $\text{Cert}(\sigma)$ is a certificate aggregating all valid votes, and σ_L is the MH-USS signature of primary node L.

Secondary node messages take the form:

$$\langle \text{phase},\ r,\ x \rangle_{\sigma_i}, \tag{10}$$

where $x = h(p)$ denotes a vote for proposed block p (with $h(p)$ computed by the parametric hash function), and $x = \perp$ denotes a vote for an empty block.

Upon receiving a message M, a node validates:

1. Whether M was received within the time limit T;
2. Validity of the MH-USS signature σ;
3. Whether the round number r matches the node's current round;
4. Validity of the aggregated certificate Cert;
5. Whether the hash value $h(p)$ is correctly computed by the parametric hash function.

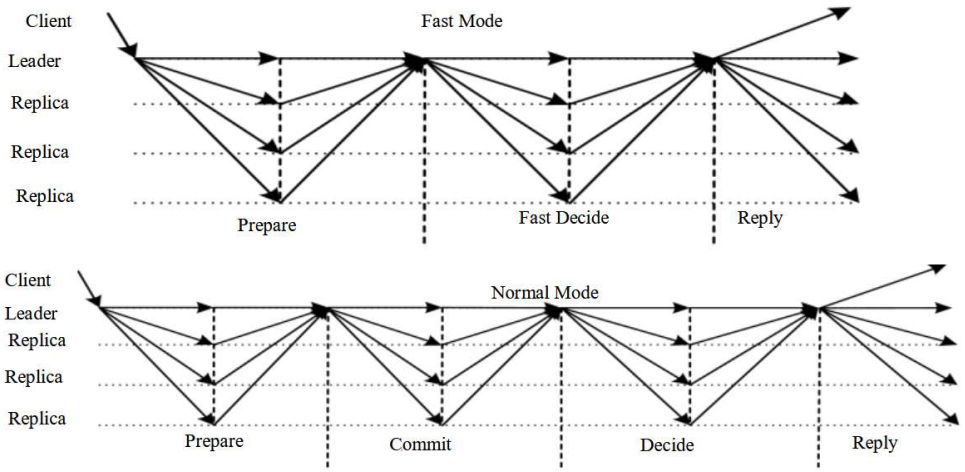


Fig 2. QS-BFT consensus message flow in fast mode and standard mode. In fast mode (top), the primary node broadcasts a FAST-RESOLVE certificate upon collecting $\geq 3f + 1$ valid votes, committing the proposed block in a single broadcast round. In standard mode (bottom), PRE-COMMIT and PRE-RESOLVE broadcast rounds each require $\geq 2f + 1$ valid votes before advancing. Dashed arrows indicate empty-block ($\perp$) vote paths, replacing the PBFT view-change mechanism.

Fig 2 illustrates the message flow in both fast mode and standard mode.

QS-BFT Consensus Protocol

Preparation Phase

The client sends an MH-USS-signed request to the current primary node, which aggregates valid pending requests into a proposed block p and broadcasts a signed PRE-PREPARE message containing the parametric hash $h(p)$. Each secondary node independently verifies the PRE-PREPARE message: if valid, it casts a vote for $h(p)$; otherwise, it casts an empty-block vote $\perp$.

Mode Selection

The primary node starts timer T immediately after broadcasting the PRE-PREPARE message, with timeouts satisfying $T_1 < T_2 - T$. If the primary node collects $\geq 3f + 1$ valid votes for $h(p)$ before the timer expires, it enters fast mode and broadcasts a FAST-RESOLVE certificate, completing consensus in a single round. If only $\geq 2f + 1$ valid votes are collected, or the timer expires, it enters standard mode and proceeds through PRE-COMMIT and PRE-RESOLVE phases.

Feedback Phase

Once the primary node has verified at least $f + 1$ resolution votes, it aggregates them into a feedback certificate and broadcasts a FEEDBACK message. All secondary nodes and the client advance to round $r + 1$ upon receipt of a valid FEEDBACK message.

The full QS-BFT consensus flow is illustrated in Fig 3.

Algorithm 1 QS-BFT Consensus Protocol

Require: Network of $N = 3f + 1$ nodes; client request req
Ensure: All honest nodes commit the same block or an empty block in round r

// — **Preparation Phase** —
Client sends $\langle \text{REQUEST, req} \rangle_{\sigma_c}$ to primary node L
L aggregates valid requests into block p; broadcasts $\langle \text{PRE-PREPARE, } r, \ h(p) \rangle_{\sigma_L}$
for each secondary node i **do**
    if PRE-PREPARE message is valid **then**
        Send $\langle \text{VOTE, } r, \ h(p) \rangle_{\sigma_i}$ to L
    else
        Send $\langle \text{VOTE, } r, \ \perp \rangle_{\sigma_i}$ to L
    end if
end for
// — **Mode Selection** —
L starts timer T
if L collects $\geq 3f + 1$ valid votes for $h(p)$ before T expires **then**
    Enter Fast Mode
    L broadcasts $\langle \text{FAST-RESOLVE, } r, \ \text{Cert}(\sigma) \rangle_{\sigma_L}$
    for each secondary node i **do**
        if FAST-RESOLVE is valid for block p **then**
            Append p to local blockchain; send $\langle \text{VOTE, } r, \ h(p) \rangle_{\sigma_i}$ to L
            Lock block p for round r
        else
            Send $\langle \text{VOTE, } r, \ \perp \rangle_{\sigma_i}$ to L
        end if
    end for
else if L collects $\geq 2f + 1$ but $< 3f + 1$ valid votes, or T expires **then**
    Enter Standard Mode
    // **Commit Phase**
    L broadcasts $\langle \text{PRE-COMMIT, } r, \ \text{Cert}(\sigma) \rangle_{\sigma_L}$
    for each secondary node i **do**
        if PRE-COMMIT is valid for block p **then**
            Send $\langle \text{VOTE, } r, \ h(p) \rangle_{\sigma_i}$ to L; lock p
        else
            Send $\langle \text{VOTE, } r, \ \perp \rangle_{\sigma_i}$ to L
        end if
    end for
    // **Resolution Phase**
    L broadcasts $\langle \text{PRE-RESOLVE, } r, \ \text{Cert}(\sigma) \rangle_{\sigma_L}$
    for each secondary node i **do**
        if PRE-RESOLVE is valid for block p **then**
            Append p to local blockchain; send $\langle \text{VOTE, } r, \ h(p) \rangle_{\sigma_i}$ to L
        else
            Send $\langle \text{VOTE, } r, \ \perp \rangle_{\sigma_i}$ to L
        end if
    end for
end if
// — **Feedback Phase** —
L waits for $\geq f + 1$ resolution votes; broadcasts $\langle \text{FEEDBACK, } r, \ \text{Cert}(\sigma) \rangle_{\sigma_L}$
All nodes advance to round $r + 1$

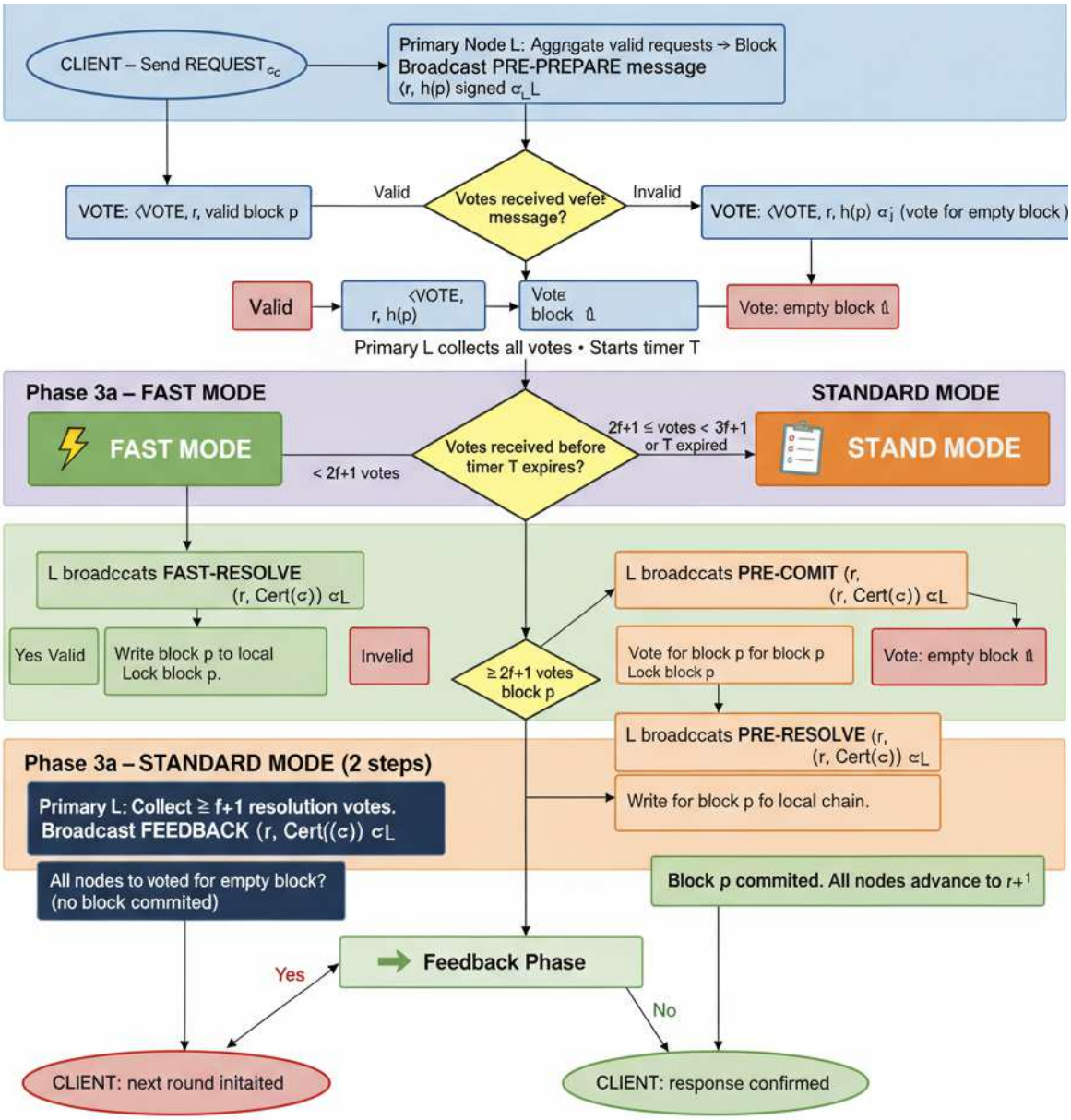


Fig 3. QS-BFT consensus flowchart illustrating the four protocol phases: Preparation, Mode Selection ($\geq 3f+1$ triggers fast mode; $\geq 2f+1$ triggers standard mode), Fast or Standard Mode execution, and Feedback ($\geq f+1$ votes required). Empty-block ($\perp$) fallback paths eliminate the PBFT view-change process.

Safety and Liveness Proofs

Theorem 4 (Safety of QS-BFT)**.** *In any consensus round r, no two distinct blocks $p \neq q$ can be simultaneously committed to the local blockchains of honest nodes.*

Proof. For a block to be committed in round r, at least $2f + 1$ nodes must cast commitment votes for it. Suppose, for contradiction, that both p and q are committed in round r. Then there exist sets S_p and S_q, each of size $\geq 2f + 1$, whose members voted for p and q respectively. Since $|S_p| + |S_q| \geq 4f + 2 > 3f + 1 = N$, the sets must share at least $f + 1$ nodes. Since there are at most f Byzantine nodes, at least one shared node must be honest. However, the QS-BFT locking mechanism ensures an honest node locks at most one block per round, yielding a contradiction. Furthermore, Byzantine nodes cannot forge votes since producing a valid MH-USS signature without QKD-distributed keys succeeds with probability at most $\frac{1}{|Y|}$ by Theorem 1. □

Theorem 5 (Liveness of QS-BFT)**.** *In a synchronous network with at least $2f + 1$ honest nodes, every round of QS-BFT terminates in finite time.*

Proof. In a synchronous network, all messages from honest nodes are delivered within bounded time T. Since there are $\geq 2f + 1$ honest nodes, the primary node is guaranteed to receive $\geq 2f + 1$ valid votes within T, satisfying the standard mode threshold. If the primary node is Byzantine and fails to broadcast valid messages, all secondary nodes cast empty-block votes; the empty-block certificate satisfies the threshold with $2f + 1$ honest votes, and the round terminates on the empty block. Since every phase has a bounded timeout and the empty-block mechanism prevents indefinite blocking, every round terminates in finite time. □

Results

Message Complexity Analysis

For QS-BFT with N nodes: fast mode incurs $O(N^2)$ total messages; standard mode incurs $O(2N^2)$. PBFT incurs $O(2N^2)$ in normal operation but degrades to $O(N^3)$ during view-change. For $N = 10$, this corresponds to a $5\times$ reduction in worst-case message count, consistent with the empirical throughput ratios in Fig 4.

Table 4. Message complexity of the core consensus process for QS-BFT, QSYAC, and PBFT.

Protocol	Fault Model	No-fault Complexity	Faulty-node Complexity
QS-BFT	$N = 3f + 1$	$O(N^2)$	$O(2N^2)$
QSYAC	$N = 4f + 1$	$O(2N^2)$	$O(2N^2)$
PBFT	$N = 3f + 1$	$O(2N^2)$	$O(N^3)$

Experimental Setup

QS-BFT and PBFT were implemented in Go (version 1.21) on a local testbed with an Intel Core i7-11700 CPU, 16 GB RAM, and Ubuntu 22.04 LTS. Each node was emulated as an independent goroutine communicating over local loopback TCP connections. QKD key distribution was simulated using cryptographically random 256-bit keys distributed prior to each run. Block size was fixed at 1 KB; each transaction was 256 bytes. Both protocols used identical MH-USS signature authentication to isolate consensus protocol differences from cryptographic overhead.

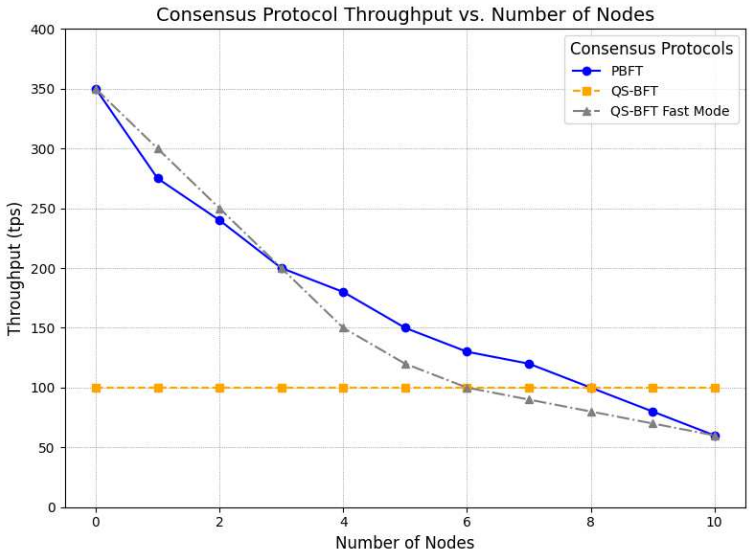


Fig 4. Throughput comparison of QS-BFT and PBFT ($N = 4$–10 nodes; mean ± 1 SD over 10 independent runs). QS-BFT consistently achieves higher TPS, consistent with its $O(N^2)$ fast-mode versus PBFT's $O(2N^2)$ normal-operation message complexity.

Network size ranged from 4 to 10 nodes; 1000 transactions were processed per configuration; each experiment was repeated 10 times independently.

Throughput Results

QS-BFT consistently achieves higher throughput than PBFT across all tested network sizes, with the advantage being most pronounced at smaller N where fast mode is more frequently activated. QS-BFT's $O(N^2)$ fast-mode complexity versus PBFT's $O(2N^2)$ normal-operation complexity yields an expected throughput ratio of approximately 2, consistent with the empirical measurements.

Latency Results

QS-BFT achieves lower latency than PBFT for all tested configurations due to two factors: (1) fast mode eliminates one full commit round-trip, and (2) centralized vote aggregation at the primary node reduces per-secondary message processing overhead.

Conclusion

Consumer-facing blockchain applications are increasingly exposed to security risks posed by advancing quantum computing capabilities. This paper introduced QS-BFT, a quantum-resistant blockchain consensus framework that integrates the MH-USS signature scheme with a parametric hash function to provide ITS-level protection for both hash pointers and digital signatures against computationally unbounded adversaries. Unlike post-quantum cryptographic schemes operating in the computational security model, QS-BFT's security guarantees hold unconditionally, regardless of future algorithmic advances. The QS-BFT protocol achieves dynamic dual-mode consensus selection and Byzantine fault tolerance while eliminating view-change overhead, reducing worst-case message complexity from $O(N^3)$ to $O(2N^2)$. Protocol correctness is rigorously established through formal safety and liveness proofs.

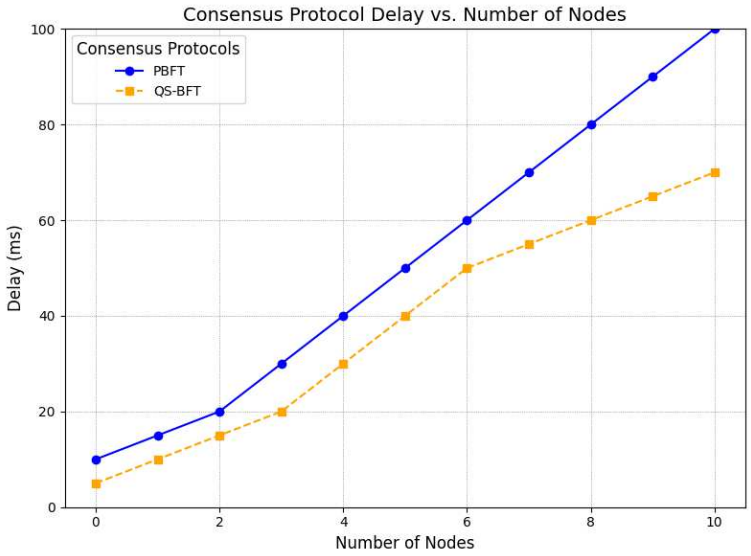


Fig 5. Latency comparison of QS-BFT and PBFT ($N = 4$–10 nodes; mean ± 1 SD over 10 independent runs). QS-BFT achieves lower end-to-end latency than PBFT across all tested network sizes.

Experimental evaluations demonstrate consistently higher throughput and lower latency than PBFT across all tested configurations. Future work will investigate quantum threshold signatures to further reduce per-round communication overhead, integration of semi-QKD protocols to lower the quantum hardware barrier for consumer devices, and node credit-scoring mechanisms to raise the cost of Byzantine misbehavior in permissioned deployments.

Author Contributions

Conceptualization: Md Farhad Kabir, Abdullah Rakib Akand, Sufia Zareen

Formal analysis: Md Farhad Kabir, Abdullah Rakib Akand

Methodology: Md Farhad Kabir, Abdullah Rakib Akand

Investigation: Md Farhad Kabir, Abdullah Rakib Akand, Md Sharif Robbani, Ahmed Faizul Haque Dhrubo, Mohammad Abdul Qayum

Validation: Md Sharif Robbani, Mithila Arman, Sufia Zareen

Resources: Md Farhad Kabir, Abdullah Rakib Akand

Data curation: Md Farhad Kabir, Abdullah Rakib Akand

Visualization: Md Farhad Kabir, Abdullah Rakib Akand

Writing – original draft: Md Farhad Kabir, Abdullah Rakib Akand, Mithila Arman

Writing – review & editing: Md Farhad Kabir, Mithila Arman, Md Sharif Robbani, Sufia Zareen

Supervision: Md Sharif Robbani, Mithila Arman

Competing Interests

No authors have competing interests. This does not alter our adherence to PLOS ONE policies on sharing data and materials.

Data Availability

All simulation code (Go implementation of QS-BFT and PBFT) and the raw throughput and latency datasets underlying the findings of this study are publicly available at:

GitHub:

`https://github.com/ahmeddhruboanondo-sys/quantumshield-plosone`

References

1. Nakamoto S 2008 Bitcoin: A peer-to-peer electronic cash system. *Bitcoin.org* Available at: https://bitcoin.org/bitcoin.pdf
2. Ferrari D, Carretta S and Amoretti M 2023 *IEEE Trans. Quantum Eng.* **4** 1–13 Art no. 2500213
3. Rezai M and Salehi J A 2023 *IEEE Trans. Quantum Eng.* **4** 1–22 Art no. 2100122
4. Anshu A, Bab Hadiashar S, Jain R, Nayak A and Touchette D 2023 *IEEE Trans. Inf. Theory* **69** 5788–5804
5. Scott J R and Balram K C 2022 *IEEE Trans. Quantum Eng.* **3** 1–20 Art no. 6000220
6. Ohkura Y, Satoh T and Van Meter R 2022 *IEEE Trans. Quantum Eng.* **3** 1–10 Art no. 2500210
7. Asfaw A *et al* 2022 *IEEE Trans. Educ.* **65** 220–242
8. Xu R, Zhou R G and Li Y 2023 *IEEE Trans. Wireless Commun.* **22** 5170–5184
9. Angara P P, Stege U, MacLean A, Müller H A and Markham T 2022 *IEEE Trans. Quantum Eng.* **3** 1–15 Art no. 3100115
10. Chung J *et al* 2022 *IEEE Trans. Quantum Eng.* **3** 1–20 Art no. 4100920
11. Kubo K, Miyamoto K, Mitarai K and Fujii K 2023 *IEEE Trans. Quantum Eng.* **4** 1–17 Art no. 3100717
12. Paramasivam S, Jenitha J, Sanjana S and Haghparast M 2023 *IEEE Access* **11** 66767–66776
13. Li Y, Zhou R G, Xu R, Luo J and Jiang S X 2022 *IEEE Trans. Emerg. Top. Comput.* **10** 211–222
14. Chawla D and Mehra P S 2023 *Internet Things* **24** 100950

15. Sharma J and Mehra P S 2023 *Internet Things* **23** 100883

16. Odoom J, Huang X, Zhou Z, Danso S, Zheng J and Xiang Y 2022 *J. Syst. Archit.* **134** 102786

17. Azahari N S B, Harun N Z B and Zukarnain Z B A 2023 *ICT Express* **9** 534–547

18. Aissaoui R, Deneuville J, Guerber C and Pirovano A 2023 *Veh. Commun.* **44** 100661

19. Varma I M and Kumar N 2023 *Veh. Commun.* **44** 100663

20. Alipour P B and Gulliver T A 2023 *MethodsX* **11** 102366

21. Zhang J, Rong Y, Cao J, Rong C, Bian J and Wu W 2022 *IEEE Trans. Dependable Secure Comput.* **19** 3387–3400

22. Liu Y *et al* 2023 *IEEE Trans. Inf. Forensics Security* **18** 2276–2291

23. Alshehri S, Bamasaq O, Alghazzawi D and Jamjoom A 2023 *IEEE Internet Things J.* **10** 4239–4256

24. Berger C, Reiser H P, Sousa J and Bessani A 2022 *IEEE Trans. Dependable Secure Comput.* **19** 1605–1620

25. Huang C *et al* 2021 *IEEE Internet Things J.* **8** 4291–4304

26. Ding Y F *et al* 2025 *Adv. Eng. Informatics* **68** 103622

27. He X C *et al* 2025 *Memetic Comput.* **17**(2) 23

28. Gong L H *et al* 2025 *Adv. Quantum Technol.* **8**(10) e2500224

29. Arumugam S K, Kumari S and Tiwari S 2025 Quantum computing for next-generation artificial intelligence-based blockchain. In: *Quantum Computing.* Auerbach Publications, pp. 251–267

30. Zhou S *et al* 2024 *Laser Phys. Lett.* **21**(6) 065207