Practical Iterative Quantum Consensus Protocol With Sharding Construction

Chenhao Ying, Yuxuan Du, Weiting Zhang, *Member, IEEE*, Xikun Jiang, Gang Wang, Haiming Jin, *Member, IEEE*, Jie Li, *Fellow, IEEE*, Yuan Luo, *Member, IEEE*, and Dacheng Tao, *Fellow, IEEE*

***Abstract*—With the development of quantum blockchain, the quantum consensus protocols have garnered increasing attention, which play a crucial role in driving the implementation of quantum blockchains. However, existing protocols, derived from the classical consensus algorithms, face practical application challenges due to current quantum technology limitations. The first challenge is the bottleneck in generating large-scale entangled quantum states. The second challenge arises from the generation of malicious quantum states. The final challenge involves privacy concerns. To address these challenges, we propose a practical iterative QUantum consensus protocol with sharding construction, namely, Q-Union. In fact, Q-Union employs an iterative consensus algorithm where participating nodes are divided into multiple smaller shards, with the consensus process occurring within the current shard, and new shards are involved only if consensus is not achieved. Leveraging Greenberger-Horne-Zeilinge states and Aharonov states, Q-Union harnesses the advantages of quantum mechanics to achieve anonymous consensus, protecting the private information of participating nodes. Additionally, by integrating state verification, Q-Union ensures the correctness of the consensus procedure in the presence of malicious nodes generating adversarial quantum states. Finally, it is proven that Q-Union can also defend against Byzantine attacks from adversarial nodes, maintaining the same security level as traditional non-sharded consensus protocols. Specifically, it consistently outputs the correct consensus when the fraction of adversaries among participating nodes is less than $1/2$ with synchronous communication. Both the theoretical analysis and performance illustration demonstrate the superior performance of the proposed Q-Union compared to state-of-the-art protocols.**

***Index Terms*—Quantum blockchain, quantum consensus, Byzantine fault tolerance, anonymous voting.**

I. Introduction

BLOCKCHAIN is a decentralized, transparent, open, immutable, distributed ledger [1] built on a peer to peer (P2P) network and maintained by all participating nodes. Since it was proposed by Nakamoto in 2008 [2], numerous technologies have been integrated to enhance its security, privacy, and efficiency, including cryptography [3], trusted execution environments [4], game theory [5], smart contract [6] and so on. Recently, it has been widely applied to various fields, such as mobile network [7], [8], supply chain management [9], financial services [10], Internet of vehicles [11], [12], electronic voting [13], transportation management [14] and health care [15].

Due to the lack of centralized management, blockchain integrates a consensus protocol to ensure that blocks are successfully appended to the chain [16]. Without a reliable third party, malicious nodes may disrupt the consensus process. Consequently, many Byzantine fault-tolerant (BFT) consensus protocols have been proposed. There are two main approaches to designing consensus protocols. One is proof-based consensus protocols [17] following Nakamoto's protocol [2], typically deployed in permissionless environment. Here, enrollment is open to all nodes, and the right to generate new blocks is randomly assigned through puzzle-solving competitions. However, these protocols rely on the longest chain during forks, limiting scalability options like sharding. The other is voting-based consensus protocols [18], which allow distributed nodes to reach consensus through voting, following classical consensus algorithms in distributed systems.

Although classical blockchains with efficient and secure consensus protocols are widely used in daily life, they typically

Received 1 August 2024; revised 10 December 2024; accepted 11 January 2025. Date of publication 22 May 2025; date of current version 4 September 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2024YFB2705300; in part by the National Natural Science Foundation of China (NSFC) under Grant 62402313, Grant 62372288, and Grant U21A20519; in part by Shanghai Science and Technology Innovation Action Plan under Grant 23511100400; in part by the National Research Foundation, Singapore; in part by the CyberSG Research and Development Program Office ("CRPO"), under the National Cybersecurity Research and Development Program ("NCRP"), RIE2025 NCRP Funding Initiative under Award CRPO-GC1-NTU-002; and in part by the Open Research Fund of the State Key Laboratory of Blockchain and Data Security, Zhejiang University. *(Corresponding authors: Yuan Luo; Xikun Jiang; Weiting Zhang; Gang Wang.)*

Chenhao Ying is with the Department of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China, also with the State Key Laboratory of Blockchain and Data Security, Zhejiang University, Hangzhou 310027, China, and also with the Blockchain Advanced Research Center, Shanghai Jiao Tong University, Wuxi, Jiangsu 214101, China (e-mail: yingchenhao@sjtu.edu.cn).

Yuxuan Du and Dacheng Tao are with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: duyuxuan123@gmail.com; dacheng.tao@gmail.com).

Weiting Zhang is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: wtzhang@bjtu.edu.cn).

Xikun Jiang is with the Department of Computer Science, University of Copenhagen, 1172 Copenhagen, Denmark (e-mail: xikun@di.ku.dk).

Gang Wang is with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: 1910636@stu.neu.edu.cn).

Haiming Jin and Jie Li are with the Department of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: jinhaiming@sjtu.edu.cn; lijiecs@sjtu.edu.cn).

Yuan Luo is with the Department of Computer Science, Shanghai Jiao Tong University, Shanghai 200240, China, and also with Shanghai Key Laboratory of Trusted Data Circulation and Governance in Web3, Shanghai 200240, China (e-mail: yuanluo@sjtu.edu.cn).

Digital Object Identifier 10.1109/JSAC.2025.3568014

rely on traditional cryptographic algorithms that face high security risks from quantum computers. For example, digital signature algorithms in classical blockchains are built on problems like factoring large integers, which is hard for classical computers but can be solved in polynomial time using Shor's quantum algorithm [19]. Therefore, quantum blockchain networks [20] have been proposed to address threats posed by advancements in quantum devices. As quantum blockchains develop, it is necessary to design corresponding quantum-based consensus protocols to support their applications.

However, designing a new quantum-based consensus protocols introduces additional challenges due to the limitations of current quantum technologies.

The first and the biggest challenge stems from the bottleneck in generating large-scale entangled quantum states. To support the applications built on blockchain, the consensus protocols require to ensure their performance in the large-scale scenarios. However, this is difficult when designing quantum-based consensus protocols since generating a large-scale entangled quantum states, such as the Greenberger-Horne-Zeilinge (GHZ) states, that are indispensable for these protocols, remains technologically out of reach today.

The second challenge arises from the generation of malicious quantum states. Due to the lack of centralized management in blockchain, the necessary entangled quantum states need to be generated and distributed by participating nodes. Consequently, apart from the traditional Byzantine attacks, malicious nodes can also generate illegal quantum states to disrupt the consensus procedure. Therefore, it is necessary to verify the quality of shared quantum states, allowing honest states to pass verification while blocking dishonest ones. However, this verification is challenging with current quantum technologies.

The final challenge involves privacy concerns. The voting-based consensus protocol reaches a consistent decision through distributed voting. However, an individual's voting decision is private information, and participating nodes seek to prevent access to this sensitive information, especially by adversaries. Therefore, the design of consensus protocols must consider identity anonymity during the voting procedure to prevent information leakage. Although some anonymous quantum voting schemes [24], [25] have been proposed, they all have vulnerabilities as highlighted in [26]. Ensuring voting anonymity is particularly challenging in quantum-based consensus protocols, especially in the presence of malicious nodes generating illegal quantum states, as adversaries can collude with them to infer the voting information of honest nodes.

In order to solve the first challenge, we divide the participating nodes into small shards, with only one shard involved in each consensus loop. A new shard joins only if the consensus is not reached in the previous loop. However, this method faces difficulty in ensuring accurate decisions, as some shards may contain a higher ratio of adversaries as the shard size decreases, leading to an incorrect consensus decision. To mitigate this risk, we propose a novel iterative quantum consensus algorithm. This algorithm ensures the same level of security as traditional non-sharded consensus approaches. Specifically, the correct decision can always be achieved when the ratio of adversaries in participating nodes is less than $1/2$ under synchronous communication. To address the second challenge, we integrate state verification into the consensus protocol. This ensures that honest states can reliably pass verification with high probability, while malicious states are unable to pass at all. To tackle the final challenge, we leverage the Aharonov state and its lateral rotational invariance property, allowing each participating node to acquire a secret number known exclusively to themselves. Furthermore, with the assistance of secret numbers, we can guarantee voting anonymity by considering the distance between the shared quantum state and the ideal GHZ state.

Therefore, after addressing the above challenges, we propose a practical iterative quantum consensus protocol with sharding construction, namely, Q-Union. The main contributions of this paper are as follows.

- *Novel Protocol:* Unlike existing works, we propose a novel iterative quantum consensus protocol, Q-Union, to provide secure, private, and efficient consensus process. It consists of two main procedures: Consensus Procedure, and Quantum Voting. The Consensus Procedure utilizes an iterative sharded consensus algorithm to improve the efficiency and security of protocol. The Quantum Voting utilizes GHZ and Aharonov states to ensure the correctness and anonymity of the voting procedure after integrating three sub-phases, Verification of GHZ States, Random Secret Number, and Anonymous Voting.
- *Consensus Procedure:* To improve the practicality and security of the quantum consensus protocol, Q-Union divides the participating nodes into multiple small shards, after which, it employs an iterative sharded consensus algorithm to ensure the correctness of consensus with $\lfloor (N-1)/2 \rfloor$ fraction of malicious nodes under synchronous communication, where N is the number of participating nodes. This achieves the same security guarantee as the traditional non-sharded consensus protocols.
- *Voting Procedure:* In order to utilize the advantage of quantum mechanics, Q-Union integrates a quantum voting procedure. Leveraging the GHZ and Aharonv states, it ensures the voting anonymity of participating nodes. In fact, the probability that the malicious nodes successfully guess the voting result of honest nodes is less than $\frac{1}{(1-S_l)M} + \epsilon$, where S_l is a safety threshold in the l-th consensus loop, M is the number of nodes in each shard, and ϵ is the distance between the shared quantum state and the ideal GHZ state.
- *Verification Procedure:* In order to prevent the malicious behaviors from adversarial leaders who prepare and distribute illegal quantum state to participating nodes to disrupt the consensus procedure, Q-Union integrates a verification procedure to assess the quality of shared states. Through verification, honest quantum states reliably pass with high probability, whereas counterfeit states have a negligible chance of passing the check. Specifically, an honest GHZ state passes verification with probability $1 - \frac{\epsilon}{2}$, while a counterfeit state passes with probability $\frac{4 \times 2^{-MT}}{\epsilon^2 (1-S_l)}$. Similarly, an honest Aharonov state

TABLE I

PERFORMANCE COMPARISON AMONG SOME QUANTUM CONSENSUS PROTOCOLS, WHERE N IS THE TOTAL NUMBER OF NODES IN THE NETWORK AND M IS THE NUMBER OF NODES IN EACH SHARD. IN FACT, THE DECENTRALIZATION OF THE COUNTERFACTUAL QUANTUM BFT (CQ-BFT) CONSENSUS PROTOCOL IS PROPOSED IN [21]. QUANTUM DELEGATED POS (QDPOS) IS PROPOSED IN [22]. THE QUANTUM BFT (Q-BFT) CONSENSUS PROTOCOL IS PROPOSED IN [23]

Consensus Protocols		Q-Union (Ours)	CQ-BFT [21]	QDPoS [22]	Q-BFT [23]
Security	Fault Tolerance	$\lfloor \frac{N-1}{2} \rfloor$	$\lfloor \frac{N-1}{2} \rfloor$	$\lfloor \frac{N-1}{2} \rfloor$	$\lfloor \frac{N-1}{2} \rfloor$
	Other Attacks	Double voting Malicious state attacks Eavesdropping attacks Tampering records	List inconsistency Trojan horse attacks	Tampering records	List inconsistency
Scalability	Quantum resources	$\zeta(M+1)$ qubits for $1 \leq \zeta \leq \frac{N}{M}$	$N + 2^N - 1$ qubits	N qubits	N qubits
	Involved voting nodes	ζM	N	ξM for $1 \leq \xi \leq \frac{N}{M}$	N
Decentralization		High	Moderate	High	Low

passes with probability 1 while a counterfeit passes with probability $2^{-MT}S_l^{-1}$, where M is the number of nodes in each shard, which is also referred to as the shard size, T is the number of each node's coins, S_l is a safety threshold in the l-th consensus iteration.

This paper is organized as follows. We present related work, preliminaries, as well as system model and threat model in Section II, III, IV, respectively. Furthermore, the framework of our proposed algorithm is illustrated in Section V, followed by theoretical analysis in Section VI. After showing the performance illustration in Section VII, we conclude our work in Section VIII.

II. RELATED WORK

In this section, some related works are introduced, mainly in terms of quantum consensus protocols.

Blockchain is a distributed ledger maintained and replicated by all participating nodes in the network [27]. It integrates a consensus protocol to ensure that blocks are successfully appended to the blockchain [1]. However, without a reliable third party, malicious nodes can engage in adversarial behavior that disrupts the consensus process [28]. Therefore, many consensus protocols with BFT have been proposed.

Although classical blockchains with efficient and secure consensus protocols have been integrated into our daily lives, they often rely on classical cryptographic algorithms that are vulnerable to the security risks posed by quantum computers [29], [30]. For example, the digital signature algorithms are based on some difficult problems such as solving discrete logarithms or factorizing big integers. However, factorizing large integers, a hard problem, can be solved in polynomial time using Shor's quantum algorithm [19]. Consequently, several quantum blockchain networks have been proposed to mitigate the threats posed by advancements in quantum technology [20]. Alongside the development of quantum blockchains, new quantum-based consensus protocols have also been introduced [21].

A quantum-based delegated PoS consensus protocol [22] has been developed using quantum voting to enhance the decentralization, efficiency, and security of quantum blockchains. Utilizing counterfactual unitary computation with chained quantum Zeno gates, a counterfactual quantum BFT protocol [21] has been proposed, enabling agreement among parties without the transmission of any physical particles through the quantum channel. A Byzantine agreement protocol [23] has been introduced using quantum channels with three participating nodes. Another synchronous Byzantine agreement protocol [31] operates in the presence of an adaptive, fully informed, and computationally unbounded adversary. Additionally, various probabilistic generalizations of classical consensus states have been identified by redefining new consensus situations based on invariance and symmetry properties [32]. With the support of quantum channels, partially synchronous consensus protocols achieve the same security level as traditional synchronous protocols [33].

Although several quantum consensus protocols have been proposed for integration into quantum blockchains by utilizing entangled quantum states [34], preparing a large-scale quantum state remains technologically out of reach [35]. Table I shows the performance comparison among different protocols.

III. PRELIMINARIES

The preliminaries introduced in this section include quantum computing, quantum communication and quantum blockchain.

A. Quantum Computing

Quantum mechanics can be formulated in terms of linear algebra in the Hilbert space $\mathcal{H}$. Analogous to the fundamental role of bit in classical computing, the basic element in quantum computation is quantum bit (qubit). Under the Dirac notation, a qubit state is a two-dimensional vector and defined as $|a\rangle = a_0|0\rangle + a_1|1\rangle \in \mathbb{C}^2$ where $|0\rangle = [1,0]^T$ and $|1\rangle = [0,1]^T$ specify two unit bases and the coefficients $a_0, a_1 \in \mathbb{C}$ satisfy $|a_0|^2 + |a_1|^2 = 1$. With another qubit $|b\rangle$, the quantum state represented by these two qubits is formulated by their tensor product $|a\rangle \otimes |b\rangle$ that is a four-dimensional vector, which can also be written as $|a\rangle|b\rangle$ or $|a,b\rangle$ for convenience. For clearness, $|a\rangle|b\rangle$ can be further written as $|a\rangle_A|b\rangle_B$, which means that $|a\rangle_A$ is assigned in the quantum register A and similarly $|b\rangle_B$ is in B. Additionally, the inner product of these two qubits is denoted as $\langle a|b\rangle$ where $\langle a|$ is the conjugate transpose of $|a\rangle$. Furthermore, an n-qubit state $|c\rangle \in \mathbb{C}^{2^n}$ is formulated as $|c\rangle = \sum_{i=1}^{2^n} c_i e_i = \sum_{i=1}^{2^n} c_i|i\rangle$ with $\sum_{i=1}^{2^n} |c_i|^2 = 1$ where

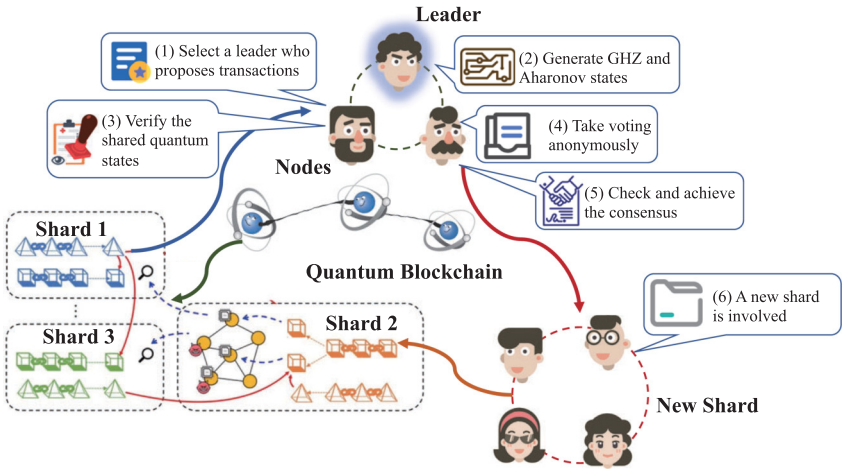


Fig. 1. Workflow of Q-Union, where the numbers represent the event order.

$|i\rangle$ stands for the unit basis vector e_i that is the unit vector with the i-th entry being 1 and other entries being 0. $|a\rangle$ is referred to as in superposition if more than one entries of a are nonzero.

Furthermore, we present the definitions of fidelity and trace distance between two quantum states ρ and σ.

Definition 1: The fidelity between two quantum states ρ and σ, expressed as density matrices, is

$$F(\rho,\delta)=(Tr(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}))^2. \tag{1}$$

Definition 2: The trace distance between two quantum states ρ and σ is

$$\mathcal{D}(\rho,\sigma)=\frac{1}{2}||\rho-\sigma||_1, \tag{2}$$

where $||\rho-\sigma||_1=Tr(\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)})$ with the conjugate transpose $(\cdot)^\dagger$.

B. Quantum Communication

Utilizing the unique properties of qubits, such as superposition and entanglement, quantum communication aims to transfer and exchange some information in quantum networks.

Entanglement is a phenomenon in quantum physics that describes a peculiar correlation between two or more quantum particles. When particles become entangled, the state of one particle cannot be described independently of the others. This means that measuring the state of one entangled particle instantaneously affects the state of the other particles, regardless of the distance between them.

Quantum teleportation allows to transfer information about the unknown quantum state from one location to another without any physical movement of the original qubit. Unlike the traditional communication, the sender in quantum teleportation does not need to know the teleported quantum state and even the location of the receiver. However, for the successful completion of quantum teleportation, classical information about the joint measurement results of sender must be conveyed to receiver. By receiving these measurement outcomes, teleported quantum state can be faithfully reconstructed after performing the necessary operations on receiver's particle. The GHZ state denoted as $|G\rangle=\frac{1}{\sqrt{2}}(|0\rangle^{\otimes N}+|1\rangle^{\otimes N})$ has been widely utilized in quantum teleportation, where it is an entangled state of three or more qubits. In fact, GHZ states have been shown to be also useful in quantum consensus algorithms.

IV. System Model and Threat Model

In this section, we introduce the system overview, threat model, and desirable goals.

A. System Overview

We consider a quantum blockchain system comprising a set $\mathcal{N}=\{n_1,\cdots,n_N\}$ of N participating nodes. Each node n_i holds a quantum computer applying the properties of quantum mechanics to perform computations. These nodes are divided into $K=\lfloor\frac{N}{M}\rfloor$ shards, each of which has M nodes, where M is determined in the following parts. Furthermore, each node has M assistants played by other nodes to record her vote result. The workflow of Q-Union is shown in Fig. 1 and described as follows.

- **Consensus Procedure:** The nodes in shard select a leader who will propose a new transaction (Step (1)). After that, each node n_j implements quantum voting to obtain a voting vector $\boldsymbol{V}_j$, which is sent to the network. Furthermore, each node n_j calculates the voting result $\boldsymbol{R}_j$ that is the voting selection of all nodes calculated by n_j. After that, the protocol arrives a consensus by checking the number of identical votes (Step (5)). If condition is satisfied, the consensus procedure is terminated and the decision is finalized by a classical control chain. Otherwise, a new shard is involved to continue the procedure (Step (6)).
- **Quantum Voting:** The selected leader prepares and distributes some GHZ states $|G\rangle$ and Aharonov states $|A\rangle$ to participating nodes (Step (2)). In order to prevent malicious behaviors from adversarial nodes and leaders to guarantee the correctness and anonymity of quantum voting, the participating nodes verify the received qubits (Step (3)), where the procedure is aborted once the states cannot pass the verification. After that, each node n_j obtains a secret number $\tilde{j}$ by measuring the received Aharonov state. Furthermore, leveraging secret number, the nodes complete the voting procedure anonymously with GHZ state (Step (4)).

B. Threat Model

During the consensus procedure, nodes require to make decision on the transaction information by utilizing iterative quantum consensus algorithm. However, within the quantum blockchain network, nodes are selected from a broad group of participants, which may result in malicious nodes entering this system, who will take some malicious behaviors to disrupt the consensus procedure. In this paper, we consider the following common attacks in quantum consensus protocols:

- **Malicious State Attack [24]:** It refers to adversarial leaders generating malicious quantum sates and distributing to the participating nodes to disrupt the consensus procedure.
- **Eavesdropping Attack [24]:** It refers to adversaries obtaining the voting result of other nodes.
- **Double Voting Attack [24]:** It refers to adversaries voting more than once to disturb the consensus result.
- **Liveness Attack [18]:** It refers to adversaries deviating from the algorithm with invalid operations, hence blocking the progress of the system.
- **Safety Attack [18]:** It refers to multiple adversaries colluding and aiming to coerce the system into accepting an incorrect block.

The first attack comes from the malicious leaders. To achieve consensus in the quantum blockchain, participating nodes must utilize quantum states in the voting process, with these states prepared and distributed by a randomly selected leader within the shard. However, if the leader is an adversary, they may disrupt the consensus procedure by generating a malicious quantum state. Therefore, participating nodes must verify the received states to prevent the sharing of illegal states. The second and the third attacks stem from the malicious nodes. Due to the privacy concerns, it requires to protect the anonymity of participating nodes during the consensus procedure. However, the malicious nodes aim to obtain the voting information of all other nodes or some specific nodes so that they can infer some other private information. Finally, the last two attacks are referred to as the Byzantine attacks, where the votes of adversaries are opposite to their actual observation resulting in the fault consensus outcomes.

C. Design Objectives

The proposed Q-Union aims to achieve the following desirable goals:

- **Voting Correctness:** This property protects against adversarial leaders using illegal quantum states to disrupt the voting procedure. Specifically, it implies that the voting procedure will proceed correctly in the absence of adversarial leaders. Furthermore, if malicious leaders introduce illegal quantum states, the voting procedure will abort with high probability.
- **Anonymity Privacy:** This property ensures that malicious distributed nodes cannot obtain the voting information of honest nodes. In fact, it implies that only the individual node knows how she has voted.
- **Byzantine Robustness:** This property guards against Byzantine nodes casting votes that contradict their actual observations, leading to incorrect decisions. It ensures that the proposed algorithm can always produce the correct decision despite the presence of liveness and safety attacks.

In summary, this paper aims to propose a secure, robust, and efficient quantum consensus protocol for quantum blockchain networks.

V. Framework Illustration

This section presents the design details of Q-Union and shows the outlines of the protocol. In fact, Q-Union consists of five sub-phases: the Consensus Procedure, which achieves consensus by subsequently involving new shards; Quantum Voting, which uses quantum mechanics to implement the voting procedure; Secret Random Number, where each node obtains a secret random number using Aharonov states to ensure voting anonymity; State Verification, which checks the quality of distributed GHZ states used in the voting procedure; and Anonymous Voting, which employs GHZ states to achieve anonymous voting with the help of secret random number.

In brief, Q-Union works as follows. All N nodes are divided into K shards, where $K = \lfloor \frac{N}{M} \rfloor$, for each shard h_j having M nodes. When a new transaction is proposed in a shard, the leader distributes GHZ states and Aharonov states to the nodes within shard. After verifying the quality of these quantum states, each node utilizes the Aharonov state to obtain a random secret number. After that, the nodes apply GHZ states together with the random secret numbers to anonymously report their votes. At the l-th loop, once the number of nodes with the same voting selection in the total involved l shards is more than $S_l \times M \times l$, the voting process will be terminated and the consensus is achieved. Otherwise, it goes to the $(l+1)$-th loop by involving a new shard h_{l+1}. The parameter S_l is the corresponding safety threshold during the l-th loop introduced in Section III. In fact, at the l-th loop,

$$S_l = \beta + z\sqrt{\frac{\beta(1-\beta)}{M \times l}} \tag{3}$$

where $\beta = \frac{\alpha}{N}$ for α being the number adversaries in node set $\mathcal{N}$. Furthermore, z is a variable with cumulative probability $1 - \frac{10^{-b}}{K}$ in standard Gaussian distribution for b being a security parameter.

A. Consensus Procedure

In details, the proposed Q-Union consists of four sub-phases as shown in Algorithm 1.

- *Consensus Preparation:* At the first loop of the consensus procedure, each shard h_l randomly selects a leader who will propose a transaction. In contrast, at the l-th loop, each node n_j in shard h_l receives all vote summaries $\boldsymbol{U}_{l-1}^k$ from the M nodes n_k only in the last involved shard h_{l-1}, where $\forall k : n_k \in h_{l-1}$. The vote summary $\boldsymbol{U}_{l-1}^k$ is a matrix written by node n_k containing M rows and K columns (a binary number for Acceptance (1), Rejection (0)), where the element at the i-th row and j-th column stores the voting status of node $n_i \in h_j$. If there is conflicting information, *e.g.,* the vote status of node n_b

Algorithm 1 Consensus Procedure

Input: Node set $\mathcal{N}$.
Output: Final decision.
LOOP $l \leftarrow 0$;
while *Final decision has not been made* **do**
  `// Consensus Preparation`
  LOOP $l \leftarrow l + 1$;
  if *LOOP* $l = 1$ **then**
    for *Each shard* h_ℓ **do**
      A leader is selected randomly, who proposes a transaction;
  else
    for *Each node* n_j *in shard* h_l **do**
      Receive vote summaries $\boldsymbol{U}_{l-1}^k$ from all nodes n_k in shard h_{l-1};
      Check the summaries and record them to her vote summary;
      if *No more than* S_{l-1} *fraction of nodes in* $\cup_j h_j$ *for* $j : 1 \leq j \leq l-1$ *have the identical voting result* **then**
        Synchronize the latest block state;
  `// Consensus Implementation`
  Go to **Algorithm 2** so that each node n_j can obtain an M-dimensional column voting vector $\boldsymbol{V}_j$;
  for *Each node* n_j *in shard* h_l **do**
    Send the voting vector $\boldsymbol{V}_j$ to all nodes in shard h_l and her assistants;
    Calculate the voting result $\boldsymbol{R}_j$, which is also an M-dimensional column vector, by summing the corresponding elements in all voting vectors $\boldsymbol{V}_j$;
    Record the voting result $\boldsymbol{R}_j$ in her summary;
  `// Consensus Verification`
  Any node n_i checks her vote summary;
  if *No more than* S_l *fraction identical votes in* $\cup_j h_j$ *for* $\forall j : 1 \leq j \leq l$ **then**
    Broadcast a command for going to Consensus Preparation to the all nodes in shard h_{l+1};
  else
    Send the vote summary to the current shard leader;
  `// Consensus Determination`
  for *Leader in shard* h_l **do**
    Request all $M-1$ nodes in shard h_l to submit their vote summaries;
    Check correction of node n_i's vote summary submitted at the end of Consensus Verification;
    Propose a decision block containing all summaries and determined correct vote summaries;
  All N nodes in blockchain make consensus on this decision block by any BFT algorithm;
  if *Decision block is correct* **then**
    Terminate the voting process and broadcast the consensus decision;
  else
    Go to Consensus Preparation by involving shard h_{l+1};

Algorithm 2 Quantum Voting

Input: Node set $\mathcal{N}$.
Output: Voting results.
VOTING FLAG $\gamma \leftarrow 0$;
for *Shard* h_l **do**
  while $\gamma = 0$ **do**
    for *Leader* L_l *in shard* h_l **do**
      Prepare M copies of an M-party GHZ state;
      Distribute to each node in shard h_l M qubits of M GHZ states, where different qubits are from different states;
    for *Nodes in shard* h_l **do**
      Select a coin flipper CF who tosses T coins;
    if *Tossing results of* T *coins are different* **then**
      Go to **Algorithm 3** to verify the quality of GHZ states;
      if *The output of* ***Algorithm 3*** *is* 0 **then**
        The algorithm aborted;
    else if *Tossing results of* T *coins are identical* **then**
      Go to **Algorithm 4** so that each node n_j can obtain a secret number $\tilde{j}$;
      Go to **Algorithm 5** so that each node n_j can obtain an M-dimensional column voting vector $\boldsymbol{V}_j$;
      $\gamma \leftarrow 1$;

recorded in $\boldsymbol{U}_{l-1}^k$ is distinct from other summaries, then node n_e will query the vote in question from the assistants of node n_b and arrive at her own conclusion. After that, she records all these votes in her vote summary. If the combination of $\boldsymbol{U}_{l-1}^k$ for $\forall k : n_k \in h_{l-1}$ reveals less than S_{l-1} fraction of all voted nodes, *i.e.,* the fraction of all nodes in shards $\cup_j h_j$ for $\forall j : 1 \leq j \leq l-1$, have the identical vote, each node $n_j \in h_l$ synchronizes the block.

- *Consensus Implementation:* The nodes in shard h_l implement by utilizing Algorithm 2 to obtain their votes, each of which is an M-dimensional column vector $\boldsymbol{V}_j$. The nodes then send the voting vectors $\boldsymbol{V}_j$ to their assistants and all nodes in shard h_l. After that, each node n_j calculates the voting result, which is also an M-dimensional column vector $\boldsymbol{R}_j$, by summing corresponding elements of all voting vectors $\boldsymbol{V}_j$. Finally, the voting results $\boldsymbol{R}_j$ of all nodes are recorded in their summaries $\mathbf{U}_l^j$.
- *Consensus Verification:* Any node $n_i \in h_l$ checks her vote summary. If no more than S_l fraction of nodes cast the identical votes and a pre-defined time bound Δ_{loop} is reached, node n_i broadcasts a command for going to Consensus Preparation to the all nodes in shard h_{l+1}. Otherwise, if more than S_l fraction of nodes in shards $\cup_j h_j$ for $\forall j : 1 \leq j \leq l$ cast the identical votes, node n_i sends the vote summary to the current shard leader.
- *Consensus Determination:* After receiving the summary from node n_i, the leader requests all M node in shard h_l to submit their vote summaries to prevent adversary manipulation. In fact, the leader checks the correction of node n_i's submitted vote summary by comparing with the others. If there are discrepancies among received vote summaries, the leader determines the correct votes from the assistants of nodes whose votes are distinct. Finally, the leader proposes a decision block containing all summaries as well as the correct vote summary determined. After that, all N nodes in network make consensus to this decision block, which can be finished by any BFT algorithm after simply checking the contained vote summaries. If it is correct, the leader terminates the voting process in shard h_l and broadcasts the consensus decision. Otherwise, the voting process goes to Consensus Preparation for the $(l+1)$-th loop.

Remark 1: To select a leader, each participating shard employs a view change mechanism commonly used in traditional blockchain systems. Specifically, the leader is determined using the formula `(ViewNumber + BlockNumber) modular` M,

Algorithm 3 Verification of GHZ State

Input: Node set $\mathcal{N}$.
Output: Verification result of GHZ state.
A node n_i randomly selects M binary input $U_k \in \{0,1\}$, such that $\sum_{k=1}^{M} U_k \equiv 0 \mod 2$;
The node n_i sends the binary inputs to all nodes in shard h_l to verify the GHZ state;
for *Each node* n_j *in shard* h_l **do**
  Perform a Hadamard and a phase shift gate $\sqrt{Z}$;
  if $U_j = 0$ **then**
    Perform a Z operation on her received qubit;
  else
    Perform a Hadamard operation on her received qubit;
  Measure her received qubit on basis $\{|0\rangle, |1\rangle\}$;
  Select a verifier VF randomly and send the corresponding outcomes $V_j \in \{0,1\}$ to the verifier VF, who calculates $\frac{1}{2}\sum_{k=1}^{M} U_k \mod 2$;
if $\sum_{k=1}^{M} V_k \neq \frac{1}{2}\sum_{k=1}^{M} U_k \mod 2$ **then**
  The output is 0;

where `ViewNumber`is the current view number, the `BlockNumber`denotes the current block number, and *M* is the total number of nodes in each shard.

B. Quantum Voting

In order to utilize the advantage of quantum mechanic, we propose a quantum voting protocol. It works as follows which can guarantee the voting correctness and node anonymity.

As shown in Algorithm 2, before starting the quantum voting procedure, the leader L_l prepares *M* copies of an *M*-party GHZ state, where each node in shard h_l will be distributed *M* qubits, each of which is from different states. Then, a coin flipper *CF* is selected who will toss *T* coins, where if the tossing results of *T* coins are different, the nodes go to Algorithm 3 to verify the entanglement of the generated GHZ states. If the states do not pass the verification, the algorithm is aborted. In contrast, if the tossing results of *T* coins are the same, the nodes go to Algorithm 4 so that each node n_j obtains a secret number $\tilde{j}$. With the input $\tilde{j}$, the nodes go to Algorithm 5 and each node n_j obtains an *M*-dimensional column voting vector $\boldsymbol{V}_j$. Finally, the VOTING FLAG is set to $\gamma = 1$.

Remark 2: Similar to the leader election, the CF is selected by the formula `(ViewNumber + TossingNumber) modular` M, where `TossingNumber`is the number of total tossing times until now. In order to ensure its credibility, we utilize a Trusted Execution Environment (TEE) [36], where the coin toss is executed by a secure random function. This approach prevents CF from engaging in any malicious behavior during the tossing process. In fact, in each coin tossing, the CF runs the random function in TEE *T* times, where if the *T* outcomes are the same, function outputs 1, otherwise, it outputs 0. After that, the TEE submits the binary outcome to blockchain directly.

C. Verification of GHZ State

Since the leader is selected randomly, she may be an adversary who distributes an illegal GHZ state to nodes to disrupt the consensus procedure. Therefore, before the consensus, it requires to verify the quality of generated GHZ state by utilizing Verification of GHZ State.

As shown in Algorithm 3, a node n_i generates *M* binary numbers $U_k \in \{0,1\}$, where $\sum_{k=1}^{M} U_k \equiv 0 \mod 2$, which are sent to all nodes in shard h_l to verify GHZ state. For each node n_j, before the state verification, she performs a Hadamard and a phase shift gate, after which the GHZ state is transformed into a mid state *R*

$$|R_0^M\rangle = \frac{1}{\sqrt{2^{M-1}}}\left[\sum_{\Delta r \equiv 0 \mod 4} |r\rangle - \sum_{\Delta r \equiv 2 \mod 4} |r\rangle\right] \quad (4)$$

where *r* is an *M*-binary string with $r = r_0 \ldots r_{M-1}$ for $r_i \in \{0,1\}$ and Δr is the Hamming weight $\Delta r = \sum_{i=0}^{M-1} r_i$ of *r*. Furthermore, if the received number $U_j = 1$, she performs a *Z* operation on her received qubit, otherwise, she performs Hadamard operation. After that, she measures the operated qubit by the basis $\{|0\rangle, |1\rangle\}$ and sends the measured outcome $V_j \in \{0,1\}$ to the selected verifier *VF* who will calculate $\frac{1}{2}\sum_{k=1}^{M} U_k$, where if $\sum_{k=1}^{M} V_k \neq \frac{1}{2}\sum_{k=1}^{M} U_k \mod 2$, the algorithm outputs 0.

Algorithm 4 Secret Random Number

Input: Node set $\mathcal{N}$.
Output: Secret unique index.
INDEXING FLAG $\delta \leftarrow 0$;
while $\delta = 0$ **do**
  for *Leader* L_l *in shard* h_l **do**
    Prepare an M-party Aharonov state;
    Distribute to each node in shard h_l a qubit;
  Coin flipper CF tosses T coins;
  if *Tossing results of* T *coins are different* **then**
    for *Each node* n_j **do**
      Send the received qubit to verifier VF;
    Verifier VF calculates the correlation function $\mathcal{L}(|A\rangle)$ given by Eq.(6) with a set $\mathcal{E}$ of E mutually unbiased bases;
    if $\mathcal{L}(|A\rangle) \leq 1 + \frac{E-1}{M}$ **then**
      The algorithm aborted;
  else if *Tossing results of* T *coins are identical* **then**
    for *Each node* n_j *in shard* h_l **do**
      Obtain a random number $\tilde{j}$ by measuring her received qubit with computational basis;
    $\delta \leftarrow 1$;

Algorithm 5 Anonymous Voting

Input: Node set $\mathcal{N}$.
Output: Voting results.
for *Each node* n_j *in shard* h_l **do**
  for *Each received qubit* **do**
    Node n_j performs Hadamard operation on the received qubit;
    Node n_j measures the received by utilizing computational basis;
  Node n_j obtains an M-dimensional column vector $\boldsymbol{N}_j$;
  Node n_j performs XOR operation between her vote v_j and the element on the $\tilde{j}$-th position in $\boldsymbol{N}_j$ to obtain an M-dimensional voting vector $\boldsymbol{V}_j$;
  Node n_j broadcasts the voting vector $\boldsymbol{V}_j$;

D. Secret Random Number

In order to guarantee the anonymity of participating nodes, each node performs Secret Random Number, Algorithm 4, to

obtain a secret unique random number that will be utilized in voting procedure so that their voting preference cannot be tapped by other adversaries.

As shown in Algorithm 4, the leader L_l prepares an M-party Aharonov state [37], which is distributed to the nodes in shard h_l. In fact, the Aharonov state is denoted as

$$|A\rangle = \frac{1}{\sqrt{M!}} \sum_{A \in \mathcal{A}_M^1} \epsilon_{a_0 \ldots a_{M-1}} |a_0\rangle |a_1\rangle \ldots |a_{M-1}\rangle \quad (5)$$

where $\mathcal{A}_M^1$ is a set of all permutations of all elements in positive integer set $\mathbb{Z} = \{0, 1, 2, \ldots, M-1\}$. $\epsilon_{a_0 \ldots a_{M-1}}$ denotes the generalized Levi-Civita symbol with a permutation in the form $a_0 a_1 \ldots a_{M-1}$. $|A\rangle$ is M-lateral rotationally invariant. This means that if we act on any of them with the tensor product of the M rotation operators referring to all the particles for any arbitrary rotation, the result will be to reproduce the same state (within a possible phase factor). Therefore, whenever the M parties measure the spin of the M separated particles along any direction, each of them finds a different result in the set $\mathbb{Z} = \{0, \ldots, M-1\}$. After receiving the qubit, the coin flipper *CF* tosses T coins to decide whether the participating nodes will verify the quality of received qubit or measure it to obtain the secret number since the leader maybe malicious who aims to distribute the illegal Aharonov state to disrupt the anonymity. If the tossing results of all T coins are different, *i.e.,* some are heads while some are tails, the participating nodes send their qubits to verifier *VF*, who calculates the correlation function [38], [39]

$$\mathcal{L}(|A\rangle) = \sum_{i=1}^{E} \sum_{k=0}^{M-1} \langle k_i| \otimes \langle k_i|\rho|k_i\rangle \otimes |k_i\rangle, \quad (6)$$

with $\rho = |A\rangle\langle A|$ by utilizing a set $\mathcal{E} = \{\mathcal{E}_1, \ldots, \mathcal{E}_E\}$ of E mutually unbiased bases, where $\mathcal{E}_i = \{|0_i\rangle, \ldots, |(M-1)_i\rangle\}$ and $|i_k\rangle \in \mathcal{E}_k$. The mutually unbiased bases mean that for any $i \neq j$, $\mathcal{E}_i$ is mutually unbiased with $\mathcal{E}_j$, *i.e.,* $|\langle l_i|f_j\rangle|^2 = \frac{1}{M}$, for $l, f = 0, \ldots, M-1$. As known from [38] and [39], if $\mathcal{L}(|A\rangle) \leq 1 + \frac{E-1}{M}$, the algorithm is aborted since the state is separable. This procedure is repeated until all tossing results are the same. Then, each node n_j measures the received qubit by utilizing computational basis and regards the measured result $\tilde{j}$ as her secret number. Furthermore, the INDEXING FLAG is set to $\delta = 1$.

E. Anonymous Voting

After verifying the GHZ state and obtaining the random number, each node n_j further receives M qubits, which are from different GHZ states distributed by the leader L_l. Each node n_j then performs Hadamard operation on the received qubits. After that, the operated qubits are measured by utilizing computational basis so that an M-dimensional column vector $\boldsymbol{N}_j$ is obtained since she has M qubits. Then, she performs XOR operation between her vote v_j and the element on the $\tilde{j}$-th position in vetor $\boldsymbol{N}_j$ to obtain an M-dimensional voting vector $\boldsymbol{V}_j$, which is broadcast to other nodes in shard h_l.

Remark 3: Taking Hadamard operation on the corresponding M qubits in the same GHZ state can obtain

$$2^{-M/2} \sum_{\sum_{i=0}^{M-1} \Delta r \equiv 0 \mod 2} |r\rangle. \quad (7)$$

When the state is measured in the computational basis in Algorithm 5, the summation of the outcomes of each particle modulus 2 is equal to zero. Since each node n_j has M qubits from M GHZ states, summing the corresponding measured outcomes of nodes' qubits from the same GHZ state modulus 2 is 0. Furthermore, the utilized Aharonov state to obtain secret number in Algorithm 4 leads to the difference among the numbers of different nodes, where each number belongs to $\mathbb{Z} = \{0, \ldots, M-1\}$. It means that the elements of different nodes in the M-dimensional column vector $\boldsymbol{N}_j$ that are changed by applying XOR operations have distinct positions. Therefore, taking summation among the corresponding elements in voting vector $\boldsymbol{V}_j$ of all nodes n_j obtains the votes of different nodes. For example, summing the $\tilde{j}$-th elements of all voting vectors $\boldsymbol{V}_j$ is the vote decision of node n_j, which is the $\tilde{j}$-th element of voting result vector $\boldsymbol{R}_j$ of each node n_j.

F. Toy Example

To illustrate Q-Union more clearly, we will provide a toy example. As shown in the following sections, when the nodes are divided into $K = 5$ shards, each shard must have at least $M = 5$ nodes to ensure that there is at least one honest node in each shard with probability more than 0.85.

At the beginning of the consensus protocol, nodes in the shard select a leader L_1 who proposes a transaction. After that, the leader prepares 5 copies of a 5-party GHZ state $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes 5} + |1\rangle^{\otimes 5})$ which are distributed to the nodes in shard h_1, where each node receives 5 qubits that are from different 5 states. After receiving the qubits from GHZ states, each node n_j for $1 \leq j \leq 5$ checks the entanglement of the received qubits to prevent the malicious leader from distributing illegal GHZ states. In details, the nodes select a coin flipper *CF* who tosses 10 coins, where if the results of 10 tossing results are not the same, for example, there are 3 heads and 7 tails, the nodes verify the state quality. Each node n_j performs Hadamard operation and phase shift operation on received qubits, after which, she randomly selects 5 binary inputs $U_k \in \{0, 1\}$ for $1 \leq k \leq 5$ to verify one of 5 GHZ states, where $\sum_{k=1}^{5} U_k \equiv 0 \mod 2$, and distributes to each node one input. If the node for example n_1 receives $U_1 = 0$, she performs Z operation

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (8)$$

on the qubit, otherwise, she performs Hadamard operation

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (9)$$

After that, node n_j measures the operated qubit by utilizing $\{|0\rangle, |1\rangle\}$ and sends the outcomes V_j to the selected verifier. Finally, if $\sum_{k=1}^{5} V_k \not\equiv \frac{1}{2} \sum_{k=1}^{5} U_k \mod 2$, the algorithm is aborted.

When the tossing results of 10 coins are the same, and all verifications are successful, the nodes perform voting procedure to show their voting preference. Nevertheless, before the voting, each node requires to obtain a random secret number to guarantee the identity anonymity. In fact, the leader prepares a 5-party Aharonov state and distributes to each node one qubit. Similar to the GHZ states, nodes require to verify the entanglement of Aharonov states to prevent the malicious behaviors from adversarial leader. The coin flipper tosses 10 coins to decide whether the nodes perform verification or obtain the random number. If the coins do not give the same tossing result, the nodes select a node and send their qubits to the verifier. After that, verifier calculates $\mathcal{L}(|A\rangle)$. With simple calculation, it can be seen that when the state is separable, we have $\mathcal{L}(|A\rangle) \leq 2$. If all 10 tossing results are the same and the all previous verification are successful, nodes measure their received qubits with computational basis to obtain the random number, for example, the random number of node n_1 is 2. Since the Aharonov state $|A\rangle$ is 5-lateral rotationally invariant, the measured outcomes of the qubits from the identical state are distinct with each other, which means that the random numbers of 5 nodes belong to $\mathbb{Z} = \{0, 1, \ldots, 4\}$ and are distinct with each other. For convenience, let's assume that the pairs are $n_1 \to 2$, $n_2 \to 1$, $n_3 \to 3$, $n_4 \to 0$, $n_5 \to 4$, which can be denoted as a pair matrix

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 3 & 0 & 4 \end{pmatrix}. \tag{10}$$

After obtaining the random number that is only known by the node herself, each node measures the 5 qubits from 5 GHZ states distributed by the leader. After the measurement, each node n_j obtains a 5-dimensional column vector denoted as for example $\boldsymbol{N}_1 = (0, 0, 1, 1, 1)'$, where $(\cdot)'$ is the transposition. Since before the measurement, each node performs Hadamard operation to the received qubit. Therefore, the summation of outcomes of measured 5 qubits modulus 2 is 0. Accordingly, writing the measured outcomes of each node as a matrix, it can be obtained

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}. \tag{11}$$

Finally, taking XOR operation between each node n_j's vote $v_j \in \{0, 1\}$ and the element at $\tilde{j}$-th position of her voting vector $\boldsymbol{N}_j$, she obtains her voting vector $\boldsymbol{V}_j$, for $1 \leq j \leq 10$. For example, assuming the vote of n_1 is 1 (Acceptance) and taking XOR between the vote 1 and the 3-rd element 1, her vote vector $\boldsymbol{V}_1 = (0, 0, \mathbf{0}, 1, 1)'$. Similarly, we can obtain

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 0 & \mathbf{1} & 1 \\ 0 & \mathbf{1} & 1 & 1 & 0 \\ \mathbf{0} & 0 & 1 & 1 & 1 \\ 1 & 1 & \mathbf{0} & 1 & 1 \\ 1 & 1 & 0 & 0 & \mathbf{0} \end{pmatrix}, \tag{12}$$

where the votes of these nodes are assumed to be $n_1 \to 1$, $n_2 \to 1$, $n_3 \to 0$, $n_4 \to 1$, $n_5 \to 0$. Finally, summing the elements in each row, each node n_j obtains a voting result vector $\boldsymbol{R}_j = (1, 1, 1, 0, 0)'$, where the first element 1 is the vote of node n_4, the second element 1 is the vote of node n_2 and so on. Each node n_j records the result vector $\boldsymbol{R}_j$ to her voting summary $\boldsymbol{U}_1^j$ and sends vector $\boldsymbol{V}_j$ to her assistants.

From Eq.(3), we know that the safety threshold is $S_1 = 0.723$ by setting $l = 1$, $g = 1$, and $M = 5$. However, the actual accepted voting ratio is $\frac{3}{5} = 0.6$, which means the consensus protocol continues by involving a new shard h_2. Each node n_i in shard h_2 requests the voting summaries from the nodes in shard h_1 and checks the consistency of the received votes, to prevent malicious nodes manipulating their vote summaries. After checking the voting summaries, each node n_i in shard h_2 records the corresponding voting information to her voting summary $\boldsymbol{U}_2^j$. After that, the nodes implement the consensus procedure similar to those nodes in shard h_1. In the second iteration, the safety threshold is $S_2 = 0.658$ by setting $l = 2$, $g = 1$, and $M = 5$. Once there are more than 4 nodes voting 1 for acceptance, the consensus procedure terminates since the actual accepted voting ratio is larger than $\frac{3+4}{5+5} = 0.7$. Finally, some nodes submit the voting results to the current leader L_2, who will check the voting summaries of nodes in shard h_2 and put their summaries in a new traditional control block for finalization that involves all 25 nodes in the networks.

Remark 4: As shown in the toy example, only one shard is involved in each consensus loop. Consequently, the quantum voting is conducted among the participating nodes within the current smaller shard, which has far fewer nodes compared to the entire network. For instance, in the example, each shard contains $M = 5$ nodes, while the entire network consists of $N = 25$ nodes. Therefore, unlike other quantum consensus protocols [21], [22], which require the preparation of a large-scale entangled quantum state, the leader in Q-Union only needs to generate and distribute a small-scale entangled state during each consensus loop, such as 5-party GHZ states and Aharonov states in the toy example.

VI. Theoretical Analysis

In this section, it will be shown that Q-Union achieves the desirable properties introduced in Section III. In brief, it guarantees the voting correctness (Theorem 1), anonymity privacy (Theorem 2), Byzantine robustness (Theorem 3).

During the quantum consensus procedure, the leader in each shard plays an important role, who requires to propose a transaction, prepares GHZ state and Aharonov state for quantum voting. However, since all leaders are selected randomly from distributed nodes, they may take some malicious behaviors to destroy the consensus procedure. In order to guarantee the implementation of consensus procedure, the proposed Q-Union applies a verification procedure to check the quality of distributed states. At the beginning of the theoretical analysis, we will show that the honest quantum state generated by honest leaders will pass the verification with high probability when all participating nodes are honest.

Q-Union utilizes GHZ states to implement the quantum voting, which is verified by applying Algorithm 3 before voting procedure. In Algorithm 3, each node applies Hadamard operation and phase shift operation on local qubits, which in

fact transforms GHZ state $|G\rangle$ into state $|R_0^M\rangle$ given in Eq.(4). After that, each node further applies Hadamard operation and Z operation on local qubits, followed by measurements to realize quality verification. Therefore, we will consider the distance between the generated GHZ states and $|R_0^M\rangle$ state instead of the ideal GHZ states for convenience.

Lemma 1: If all participating nodes are honest, and $\mathcal{D}(|S\rangle, |R_0^M\rangle) \leq \epsilon$ with $\epsilon \leq 2^{-T}$, the honest GHZ state $|S\rangle$ will pass the verification shown in Algorithm 3 with probability $P \leq 1 - \frac{\epsilon}{2}$, where T is the number of coins tossed by each node, $\mathcal{D}(\cdot,\cdot)$ denotes the distance between two quantum states, and $|R_0^M\rangle$ is given in Eq.(4).

Proof: Please refer to the detailed proof in Appendix A.□

Apart from the GHZ states, the leader in Q-Union also prepares and distributes the Aharonov states so that each node can obtain a random unique secret number that is utilized in voting procedure to guarantee voting anonymity. Similarly, Q-Union also applies another verification procedure to verify the quality of distributed Aharonov states. As shown in Algorithm 4, the operations taken by selected node in fact calculate Eq.(6) by utilizing a set $\mathcal{E}$ of E mutually unbiased bases.

Lemma 2: Utilizing a set $\mathcal{E}$ of E mutually unbiased bases, if all participating nodes are honest and the honest Aharonov sate $|A\rangle$ satisfies

$$\mathcal{L}(|A\rangle) = \sum_{i=1}^{E} \sum_{k=0}^{M-1} \langle k_i| \otimes \langle k_i|\rho|k_i\rangle \otimes |k_i\rangle > 1 + \frac{E-1}{M}, \tag{13}$$

where $\rho = |A\rangle\langle A|$ is the density matrix, it is entangled and will pass the verification shown in Algorithm 4 with probability 1.

Proof: Please refer to the detailed proof in Appendix B.□

Furthermore, when $M = p^w$ for a prime p and a positive integer w, it can construct a complete set of mutually unbiased bases where $E - 1 = M$ and the condition becomes $\mathcal{L}(|A\rangle) \leq 2$. In contrast, whether a complete set of mutually unbiased bases exists or not in non-prime-power dimensions is still unknown.

Since leaders and verifiers are selected from the participating nodes, they may be adversaries aiming to generate malicious quantum states to disrupt the consensus procedure. Therefore, the sate verification requires to deny the dishonest states. In the following lemmas, we will show that malicious quantum states cannot pass verification, even when some malicious participating nodes collude with adversarial leaders and verifiers. For convenience, we assume that there are S_l fraction of malicious nodes in the union of involved shards, where these nodes will generate malicious quantum states when they play as a leader, while they will pass the malicious states when they play as a verifier in verification procedure.

Lemma 3: If the malicious leaders generate a fake Aharonov state $|B\rangle$ in presence of malicious verifiers that satisfies

$$\mathcal{L}(|B\rangle) = \sum_{i=1}^{E} \sum_{k=0}^{M-1} \langle k_i| \otimes \langle k_i|\rho|k_i\rangle \otimes |k_i\rangle \leq 1 + \frac{E-1}{M}, \tag{14}$$

with utilizing a set $\mathcal{E}$ of E mutually unbiased bases, where $\rho = |B\rangle\langle B|$ is the density matrix, the fake state passes the verification in the l-th consensus loop with probability $P \leq 2^{-T} S_l^{-1}$, where S_l is the safety threshold in the l-th consensus loop.

Proof: Please refer to the detailed proof in Appendix C.□

Similarly, with the help of verification procedure, the fake GHZ state also cannot pass the check. In the following lemma, we will show that even if the participating nodes have unlimited power to take malicious behaviors aiming to increase the probability of successfully passing malicious states, the fake state still cannot pass the verification shown in Algorithm 3.

Lemma 4: When malicious participating nodes can take local operation U to minimize the distance between ideal GHZ state and fake GHZ state $|B\rangle$ generated by malicious leaders as $\epsilon = \min_U \mathcal{D}((I \otimes U)|B\rangle, |R_0^M\rangle)$ with $\epsilon \geq 2^{-T}$, the fake state passes the verification shown in Algorithm 3 with the probability $P \leq \frac{4\times 2^{-T}}{\epsilon^2(1-S_l)}$, where I is the identical matrix.

Proof: Please refer to the detailed proof in Appendix D.□

Combining lemmas 1, 2, 4, and 3, it obtains that the quantum voting procedure can be implemented correctly.

Theorem 1: Algorithm 2 of Q-Union is able to defend the malicious state attack, double voting attack so that the correctness of quantum voting procedure can be guaranteed.

Proof: Please refer to the detailed proof in Appendix E.□

In fact, since the voting decision is the private information, it requires to guarantee the node anonymity so that no any other malicious nodes can tap their voting information. In order to achieve this target, Q-Union applies Algorithm 4 to allow the participating nodes to obtain a random secret number by utilizing the Aharonov states. Furthermore, with the assistance of random secret number that is known by its owner, the nodes can vote anonymously, which means the malicious nodes cannot link their identity and vote.

Theorem 2: Performing anonymous voting shown in Algorithm 5 at the l-th iteration with the GHZ states $|G\rangle$ satisfying $\mathcal{D}(|G\rangle, |R_0^M\rangle) \leq \epsilon$ and the Aharonov states $|A\rangle$ satisfying $\mathcal{L}(|A\rangle) > 1 + \frac{E-1}{M}$, for the optimal strategy that $S_l \times M$ malicious nodes can apply to guess the identity of voting nodes correctly, the successful probability is

$$P \leq \frac{1}{(1-S_l)\times M} + \epsilon. \tag{15}$$

Proof: Please refer to the detailed proof in Appendix F.□

Similar to the traditional consensus protocol, there also exist some malicious nodes in blockchain when implementing quantum consensus, who aim to vote oppositely to their observation to destroy the consensus procedure. Therefore, the proposed Q-Union requires to defend the Byzantine attacks from the adversaries.

Preparing and sharing the N-party GHZ state and Aharonov state for large-scale quantum blockchain network is today still technologically out of reach. In contrast, it is feasible when the number of voters is small. Therefore, in order to implement quantum consensus in large-scale blockchain network, it requires to divide the node set into some small shards. However, the smaller shard size leads to higher security risk since the fraction of adversaries in one shard usually increases. Therefore, it requires to determine the shard size

M, so that the nodes cannot make a false decision during the consensus procedure.

Lemma 5: Given a node set $\mathcal{N}$ with size N and α adversaries such that $\beta = \frac{\alpha}{N} < \frac{1}{2}$, for any $\beta < S_l < 1$, to guarantee that the probability of having at least S_l fraction of adversaries in each of K randomly selected shards is greater than or equal to $1 - \frac{10^{-b}}{K}$, the size M of any shard h_j must satisfy $M \geq \frac{\beta(1-\beta)}{\left(\frac{S_l-\beta}{z}\right)^2}$, where z is a Gaussian variable with a cumulative distribution probability of $1 - \frac{10^{-b}}{K}$ for a security parameter b.

Proof: Please refer to the detailed proof in Appendix H.□

It is known that the number M of the nodes involved in the first loop of consensus procedure is the smallest during the implementation of consensus protocol. Therefore, to avoid a false decision during the first loop, there must be at least one honest node in the first-loop shards.

Corollary 1: When dividing nodes into K shards, in order to guarantee at least one honest node in each shard with high probability, the shard size must satisfy $M \geq \frac{4+z^2+z\sqrt{z^2+8}}{2}$ with a standard Gaussian variable z which has the cumulative distribution probability $1 - \frac{10^{-b}}{K}$ for a security parameter b.

Proof: Please refer to the detailed proof in Appendix G.□

As shown in Algorithm 1, the assistants of each node are involved to record the vote information, where the information is utilized for vote correction when there are discrepancies among the received vote summaries. Therefore, it requires to at least one honest member in M assistants.

Theorem 3: The proposed Q-Union achieves the same security guarantee as that of a traditional non-sharded consensus algorithm. In fact, when there are $\beta \leq \lfloor N-1)/2 \rfloor$ adversaries in node set, Q-Union can always output correct consensus.

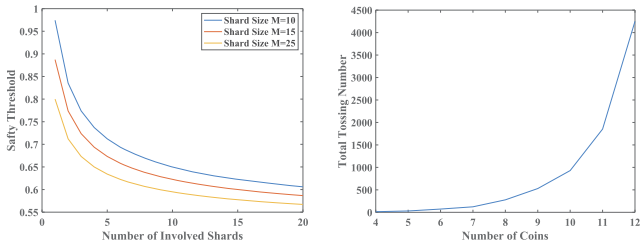
Proof: Please refer to the detailed proof in Appendix I.□

VII. Performance Illustration

We illustrate the performance of the proposed Q-Union in this section.

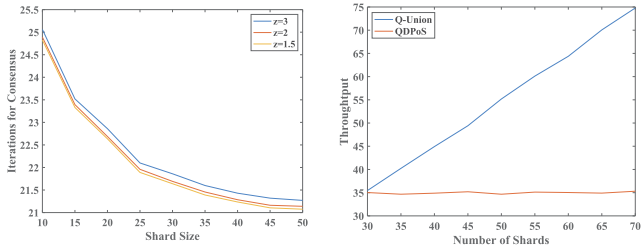
A. Scalability

As a distributed network, scalability is one of the most important performance metric for blockchain, where the blockchain is scalable if it can process a large volume of transactions without any delay or difficulties. Fig. 2(a) shows the safety threshold with the increasing number of involved shards. As shown in Algorithm 1, during the l-th iteration of Q-Union, the protocol will terminate and output the consistent decision only when the fraction of identical votes exceeds the safety threshold S_l. From the proof of Lemma 5, it follows that as the number of involved nodes increases, the fraction of malicious nodes decreases. This results in a lower safety threshold and facilitates easier consensus achievement. Furthermore, Fig. 2(b) shows the number of tossing times with the increasing number of coins. Due to the presence of malicious leaders and participating nodes, it is necessary to verify the shared quantum states prepared by leaders, as adversarial leaders may distribute illegal states to disrupt the consensus procedure. To determine the corresponding operation, whether voting or verifying, the distributed nodes require to toss coins.



(a) (b)

Fig. 2. (a) Safety threshold versus number of shard sizes with shard size (b) Number of tossing times for one voting versus number of coins.



(a) (b)

Fig. 3. (a) Number of iterations to reach the consensus versus shard size (b) Throughput versus number of shards.

They will proceed to vote only if all tossed coins show the same result. Although increasing the number of coins requires more tosses for voting, we can slightly modify the tossing rule so that only a subset of participating nodes needs to toss.

Q-Union is an iterative consensus protocol, where the number of iterations impacts its complementing efficiency. Fig. 3(a) shows the corresponding number with the various shard size. In fact, increasing the shard size leads to fewer iteration rounds since, according to Eq.(3), the threshold decreases as the number of participating nodes increases, making consensus easier to achieve. Furthermore, Fig. 3(a) also shows the required iteration rounds by fixing the number of nodes in network as $N = 1200$ under different parameter z which is a Gaussian variable. As mentioned in Lemma 5, the parameter z indicates the probability that there exists S_l fraction of malicious nodes. For example, by calculating the corresponding cumulative distribution function of Gaussian distribution, $z = 3$ means that the probability of at most S_l fraction of malicious nodes is 0.999, while $z = 2$ means the probability is 0.977 and $z = 1.5$ means it is 0.933. Fig. 3(b) shows the throughput of Q-Union compared with the quantum delegated proof of stake (QDPoS) proposed in [17]. In fact, it is the number of exchanges that can be processed in the whole blockchain networks per second. Since each shard in Q-Union can propose a new transaction and start the consensus procedure independently in parallel, with the increasing number of shards, more transactions can be proposed to take consensus. In contrast, although in QDPoS, only some of nodes are selected to take votes while the other participating nodes do not take any behaviors, the selected nodes can not propose and determine more transactions at the same time, resulting in the invariant throughput with the increasing number of shards. It should be pointed out that the results shown in Fig 3 are just the analysis rather than the practical implementation due to the limitation of current

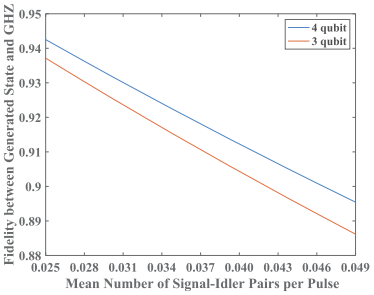
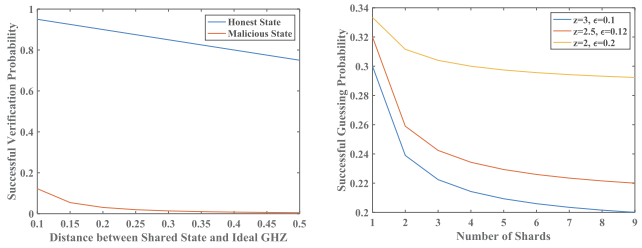


Fig. 4. Fidelity between generated state and GHZ state versus mean number of signal-idler pairs per pulse.



(a) (b)

Fig. 5. (a) Successful verification probability versus distance between shared states and ideal GHZ states (b) Successful guessing probability versus number of shards.

quantum technology. Instead, we practically implement the corresponding classical protocols in the following parts to demonstrate the actual performance of Q-Union.

B. Correctness

Compared to the classical consensus protocol, quantum-based consensus protocol is more complicated since it involves preparation and distribution of quantum states. Fig. 4 shows the fidelity of generated state by utilizing the method proposed in [40]. In the state generation experiment, the GHZ state source consists of two micro-structured photonic crystal fibers (PCFs). Each PCF produces a photon pair through spontaneous four-wave mixing, with the signal wavelength at $623nm$ and the idler at $871nm$. Furthermore, three- and four-photon GHZ states are generated through a parity check followed by post-selection. Accordingly, the fidelity between the generated four-qubit state and the ideal GHZ state is $F = \frac{2\lambda^4}{2\lambda^4+5\lambda^6}$ where $\lambda = \frac{\tilde{n}}{1+\tilde{n}}$ for $\tilde{n}$ being the mean number of signal-idler pairs per pulse. Furthermore, for the three-qubit state, the fidelity is $F = \frac{\lambda^4}{\lambda^4+\frac{11}{4}\lambda^6}$. In contrast, compared to the GHZ states, the Aharonov state is more complicated and difficult to be prepared. In particular, when there are only two parties, *i.e.,* $M = 2$, the sate degrades to the Bell state, which can be denoted as $\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$, whose generation can be found in many works [41]. In fact, for any two quantum states ρ and σ, their trace distance $\mathcal{D}$ and their fidelity F satisfy $1 - \sqrt{F(\rho,\sigma)} \leq \mathcal{D}(\rho,\sigma) \leq \sqrt{1-F(\rho,\sigma)}$ [42], [43], [44]. Therefore, with the increase in fidelity, the distance of quantum states decreases, which results in the higher successful verification probability.

Following the fidelity analysis of generated quantum state, Fig. 5(a) shows the successful probability that the generated states pass the verification. For the honest scenario, where all participating nodes are honest and the leaders honestly generate the shared quantum state, the successful probability decreases with the increasing distance between the generated state and the ideal GHZ state. Furthermore, once the distance is larger than 0.5, the state passes the verification with probability larger than 0.8. Even though the distance is 1, the successful probability is also larger than 0.5. In contrast, for the dishonest scenario, the successful probability of illegal quantum state passing the verification is less than 0.15. Furthermore, one of objective of Q-Union is to guarantee the identity anonymity of participating, which means that the malicious nodes can not infer the voting selection of honest nodes even though they can collaborate with each other. Therefore, Fig. 5(b) shows the probability that the malicious nodes successfully guess the voting selection of each honest node. In fact, as shown in Eq.(15), with the increasing number of shards, the probability decreases since the fraction of malicious nodes also decreases.

VIII. Conclusion

In this paper, we proposed Q-Union, a practical iterative quantum consensus protocol with sharding construction to support quantum blockchain. To address the challenge of generating large-scale entangled quantum states, Q-Union leverages a sharded iterative consensus algorithm, where participating nodes are divided into smaller shards, with new shards involved only if consensus isn't reached in the previous iteration. Additionally, Q-Union employs a quantum voting procedure using GHZ and Aharonov states, enabling anonymous consensus. To prevent disruptions from malicious leaders and nodes, Q-Union integrates state verification. It is proven to defend against Byzantine attacks while maintaining the same security level as traditional non-sharded protocols. Both theoretical analysis and performance evaluations demonstrate the superior performance of Q-Union.

Appendix A
Proof of Lemma 1

The proof is similar to that in [34]. After applying a Hadamard and a phase shift operations to each local qubit, the GHZ state can be expressed by the locally equivalent state

$$|R_0^M\rangle = \frac{1}{\sqrt{2^{M-1}}}\left[\sum_{\Delta r\equiv 0 \mod 4}|r\rangle - \sum_{\Delta r\equiv 2 \mod 4}|r\rangle\right] \tag{16}$$

where r is an M-binary string with $r = r_0 \dots r_{M-1}$ for $r_i \in \{0,1\}$ and Δr is the Hamming weight $\Delta r = \sum_{i=0}^{M-1} r_i$ of r. Denoting

$$|R_1^M\rangle = \frac{1}{\sqrt{2^{M-1}}}\left[\sum_{\Delta r\equiv 1 \mod 4}|r\rangle - \sum_{\Delta r\equiv 3 \mod 4}|r\rangle\right], \tag{17}$$

it can be verified that

$$|R_0^M\rangle = \frac{1}{\sqrt{2}}[|R_0^k\rangle|R_0^{M-k}\rangle - |R_1^k\rangle|R_1^{M-k}\rangle]. \tag{18}$$

From condition $\sum_{k=1}^{M} U_k \equiv 0 \mod 2$, we have

- $\frac{1}{2}\sum_{k=1}^{M} U_k \equiv 0 \mod 2$: This means that the sum of the inputs is a multiple of 4. Using Eq.(18), it can be proven

that the state $|R_0^M\rangle$ goes to $\pm|R_0^M\rangle$ when we apply to it an operator consisting of a $0 \mod 4$ number of single-qubit Hadamard, and Z operations on the remaining qubits. Therefore, it always has $\sum_{k=1}^{M} V_k \equiv 0 \mod 2$.

- $\frac{1}{2}\sum_{k=1}^{10} U_k \equiv 1 \mod 2$: This means that the sum of the inputs is even but not a multiple of 4. Using Eq.(18), it can be proven that the state $|R_0^M\rangle$ goes to $\pm|R_1^M\rangle$ when we apply to it an operator consisting of a $2 \mod 4$ number of single-qubit Hadamards, and Z operations on the remaining qubits. Therefore, it always has $\sum_{k=1}^{M} V_k \equiv 1 \mod 2$.

Therefore, the quantum state $|R_0^M\rangle$ can pass the verification with probability 1.

In the following, we will show that when $\mathcal{D}(|G\rangle, |R_0^M\rangle) \leq \epsilon$, the GHZ state $|G\rangle$ passes the verification with probability $P \leq 1 - \frac{\epsilon^2}{2}$. For $\frac{1}{2}\sum_{k=1}^{M} U_k \equiv 0 \mod 2$, and randomly selected inputs, it has

- $U_1 = 0$ and $V_1 = 0$, the verification passes if

$$\sum_{k=2}^{M} V_k \equiv \begin{cases} 0 \mod 2 & \text{if } \sum_{k=2}^{M} U_k \equiv 0 \mod 4 \\ 1 \mod 2 & \text{if } \sum_{k=2}^{M} U_k \equiv 2 \mod 4 \end{cases} \tag{19}$$

- $U_1 = 0$ and $V_1 = 1$, the verification passes if

$$\sum_{k=2}^{M} V_k \equiv \begin{cases} 1 \mod 2 & \text{if } \sum_{k=2}^{M} U_k \equiv 0 \mod 4 \\ 0 \mod 2 & \text{if } \sum_{k=2}^{M} U_k \equiv 2 \mod 4 \end{cases} \tag{20}$$

- $U_1 = 1$ and $V_1 = 1$, the verification passes if

$$\sum_{k=2}^{M} V_k \equiv \begin{cases} 0 \mod 2 & \text{if } \sum_{k=2}^{M} U_k \equiv 1 \mod 4 \\ 1 \mod 2 & \text{if } \sum_{k=2}^{M} U_k \equiv 3 \mod 4 \end{cases} \tag{21}$$

- $U_1 = 1$ and $V_1 = 0$, the verification passes if

$$\sum_{k=2}^{M} V_k \equiv \begin{cases} 1 \mod 2 & \text{if } \sum_{k=2}^{M} U_k \equiv 1 \mod 4 \\ 0 \mod 2 & \text{if } \sum_{k=2}^{M} U_k \equiv 3 \mod 4 \end{cases} \tag{22}$$

Let's consider the above verification on the $M-1$ nodes. When $\sum_{k=2}^{M} U_k \equiv 0 \mod 2$, the $M-1$ nodes perform a POVM $\{P_{M-1}, I - P_{M-1}\}$, where the first outcome corresponds to case 1 and the second to case 2. When $\sum_{k=2}^{M} U_k \equiv 1 \mod 2$, the verification is equivalent to performing a POVM $\{Q_{M-1}, I - Q_{M-1}\}$, where the first outcome corresponds to case 3 and the second to case 4.

Let J_M be the subspace of quantum pure states of M qubits that are orthogonal to both $|R_0^M\rangle$ and $|R_1^M\rangle$. Furthermore, let I_{J_M} be the projection on this subspace. We will prove by induction that

$$P_M = |R_1^M\rangle\langle R_1^M| + \frac{1}{2} I_{J_M}, \; Q_{M-1} = |R_+^M\rangle\langle R_-^M| + \frac{1}{2} I_{J_M}, \tag{23}$$

where $|R_+^M\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes M} + |1\rangle^{\otimes M})$. For $M = 1$, J_M is the empty subspace, $P_1 = |0\rangle\langle 0|$ and $Q_1 = |+\rangle\langle +|$, so the statement holds. Let's assume that the statement also holds for the number of nodes less than $M-1$ with $M \geq 2$ and the remaining is to prove it is true for M nodes. Since the selection of U_1 is uniformly random, it has

$$\begin{aligned} P_M = &\frac{1}{2}(|0\rangle\langle 0| \otimes P_{M-1} + |1\rangle\langle 1| \otimes (I - P_{M-1})) \\ &\frac{1}{2}(|+\rangle\langle +| \otimes (I - Q_{M-1}) + |-\rangle\langle -| \otimes Q_{M-1}). \end{aligned} \tag{24}$$

Further denoting $|R_-^M\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes M} - |1\rangle^{\otimes M})$, it can be seen $I - P_{M-1} = |R_1^{M-1}\rangle\langle R_1^{M-1}| + \frac{1}{2} I_{J_{M-1}}$ and $I - Q_{M-1} = |R_-^{M-1}\rangle\langle R_-^{M-1}| + \frac{1}{2} I_{J_{M-1}}$. It means

$$\begin{aligned} &P_M \\ &= \frac{1}{2}(|0\rangle\langle 0| \otimes |R_0^{M-1}\rangle\langle R_0^{M-1}| + |1\rangle\langle 1| \otimes |R_1^{M-1}\rangle\langle R_1^{M-1}| \\ &\quad |+\rangle\langle +| \otimes |R_-^{M-1}\rangle\langle R_-^{M-1}| + |-\rangle\langle -| \otimes |R_+^{M-1}\rangle\langle R_+^{M-1}| \\ &\quad + I_1 \otimes I_{J_{M-1}}), \end{aligned} \tag{25}$$

where I_1 is the identity operator on one qubit. Furthermore, applying $|R_-^{M-1}\rangle\langle R_-^{M-1}| + |R_+^{M-1}\rangle\langle R_+^{M-1}| = |R_0^{M-1}\rangle\langle R_0^{M-1}| + |R_1^{M-1}\rangle\langle R_1^{M-1}|$ and $|R_-^{M-1}\rangle\langle R_-^{M-1}| + |R_+^{M-1}\rangle\langle R_+^{M-1}| = -|R_0^{M-1}\rangle\langle R_1^{M-1}| - |R_1^{M-1}\rangle\langle R_0^{M-1}|$, it has

$$\begin{aligned} P_M = &|0\rangle\langle 0| \left(\frac{3}{4}|R_0^{M-1}\rangle\langle R_0^{M-1}| + \frac{1}{4}|R_1^{M-1}\rangle\langle R_1^{M-1}|\right) \\ &- |0\rangle\langle 1| \left(\frac{1}{4}|R_0^{M-1}\rangle\langle R_1^{M-1}| + \frac{1}{4}|R_1^{M-1}\rangle\langle R_0^{M-1}|\right) \\ &- |1\rangle\langle 0| \left(\frac{1}{4}|R_0^{M-1}\rangle\langle R_1^{M-1}| + \frac{1}{4}|R_1^{M-1}\rangle\langle R_0^{M-1}|\right) \\ &+ |1\rangle\langle 1| \left(\frac{1}{4}|R_0^{M-1}\rangle\langle R_0^{M-1}| + \frac{3}{4}|R_1^{M-1}\rangle\langle R_1^{M-1}|\right) \\ &+ I_1 \otimes I_{J_{M-1}}. \end{aligned} \tag{26}$$

Furthermore, it has

$$\begin{aligned} |R_0^M\rangle &= \frac{1}{\sqrt{2}}|0\rangle|R_0^{M-1}\rangle - \frac{1}{\sqrt{2}}|1\rangle|R_1^{M-1}\rangle) \\ |R_1^M\rangle &= \frac{1}{\sqrt{2}}|0\rangle|R_1^{M-1}\rangle + \frac{1}{\sqrt{2}}|1\rangle|R_0^{M-1}\rangle). \end{aligned} \tag{27}$$

After defining

$$\begin{aligned} |W_0^M\rangle &= \frac{1}{\sqrt{2}}|0\rangle|R_0^{M-1}\rangle + \frac{1}{\sqrt{2}}|1\rangle|R_1^{M-1}\rangle) \\ |W_1^M\rangle &= \frac{1}{\sqrt{2}}|0\rangle|R_1^{M-1}\rangle - \frac{1}{\sqrt{2}}|1\rangle|R_0^{M-1}\rangle), \end{aligned} \tag{28}$$

it can be obtained

$$\begin{aligned} P_M = &|R_0^M\rangle\langle R_0^M| + \frac{1}{2}(|W_0^M\rangle\langle W_0^M| \\ &+ |W_1^M\rangle\langle W_1^M| + I_1 \otimes I_{J_{M-1}}). \end{aligned} \tag{29}$$

Combining

$$I_{J_M} = I_1 \otimes I_{J_{M-1}} + |W_0^M\rangle\langle W_0^M| + |W_1^M\rangle\langle W_1^M|, \tag{30}$$

which concludes the proof for P_M. To complete the induction, it can be shown $Q_M = |R_+^M\rangle\langle R_+^M| + \frac{1}{2} I_{J_{M-1}}$. Assuming that M nodes share a state $|F\rangle$ with $\mathcal{D}(|F\rangle, |R_0^M\rangle) = \epsilon$. We can present $|F\rangle$ as follows:

$$|F\rangle = \sqrt{1 - \epsilon^2}|R_0^M\rangle + \sum_{j=1}^{2^M - 1} \epsilon_j |R_j^M\rangle \tag{31}$$

where $j \geq 2$, $|R_j^M\rangle \in J_M$ and $\sum_{j=1}^{2^M-1} \epsilon_j^2 = \epsilon^2$. The state $|F\rangle$ passes the verification with probability $\mathrm{Tr}(P_M|F\rangle)$. Therefore, it has

$$p = 1 - \epsilon^2 + \frac{\sum_{j=2}^{2^M-1} \epsilon_j^2}{2} \leq 1 - \epsilon^2 + \frac{\epsilon^2}{2} = 1 - \frac{\epsilon^2}{2}. \quad (32)$$

Furthermore, as shown in Algorithm 3, before the state verification, coin flipper *CF* tosses T coins to decide the selection of verification and voting. In fact, the node will vote when all T tossing results are the same, which means that the probability of voting is 2^{-T}. Therefore, the probability that the nodes verify the state in the t-th iteration is $(1-2^{-T})^{t-1}$. In each verification, the probability that the state passes the verification is $p \leq 1 - \frac{\epsilon^2}{2}$ when all leaders are honest. As shown in Algorithm 3, the nodes will vote when all checks are correct and the T tossing results are the same. Since all participating nodes are honest, the verifier *VF* pass the honest state with probability 1. Therefore, the probability of the state passing the verification at the t-th iteration is

$$Pr \leq 2^{-T}(1-2^{-T})^{t-1}p^{t-1}. \quad (33)$$

Considering all values of t, we can derive the probability that a malicious state passes the verification is

$$\begin{aligned} P &= \int_0^\infty 2^{-T}(p(1-2^{-T}))^t dt \\ &= \frac{2^{-T}}{-\ln\left(1-\frac{\epsilon^2}{2}\right)(1-2^{-T})} \leq 1 - \frac{\epsilon}{2} \end{aligned}$$

where the last inequality holds since for $\epsilon \leq 2^{-T}$. Therefore, the conclusion holds.

Appendix B
Proof of Lemma 2

Similar to that in [39], in order to complete the proof, let's consider an arbitrary pure product state $|w\rangle \otimes |u\rangle \in \mathbb{C}^M \otimes \mathbb{C}^M$. It has

$$\mathcal{L} = \sum_{i=1}^{E} \sum_{k=0}^{M-1} |\langle k_i|w\rangle|^2 |\langle k_i|u\rangle|^2. \quad (34)$$

Due to the inequality of arithmetic and geometric means for positive numbers, it has

$$\mathcal{L} = \frac{1}{2} \sum_{i=1}^{E} \sum_{k=0}^{M-1} |\langle k_i|w\rangle|^4 + |\langle k_i|u\rangle|^4. \quad (35)$$

Exploiting any pure state $|w\rangle \in \mathbb{C}^M$ and E mutually unbiased bases, it holds that

$$\sum_{i=1}^{E} \sum_{k=0}^{M-1} |\langle k_i|w\rangle|^4 + |\langle k_i|u\rangle|^4 \leq 1 + \frac{E-1}{M}. \quad (36)$$

Finally, since $\mathcal{L}$ is linear in the density matrix ρ, it follows that the inequality holds for all separable states as pure states represent extreme points. Therefore, the conclusion holds.

Appendix C
Proof of Lemma 3

As shown in Algorithm 4, coin flipper *CF* tosses T coins to decide the selection of verification or measurement. In fact, the node will measure the received qubit to obtain secret number when all T tossing results are the same, which means that the probability of voting is 2^{-T}. Therefore, the probability that the nodes verify the state in the t-th iteration is $(1-2^{-T})^{t-1}$. In each verification, the probability that the state passes the verification is less than $S_l \times 1$ since there are at most S_l fraction of malicious nodes, where the probability that malicious nodes pass the illegal state is assumed to be 1, while that of honest nodes is 0. As shown in Algorithm 4, the nodes will obtain the secret number when all checks are correct and the T tossing results are the same. Therefore, the probability of malicious state passing the verification at the t-th iteration is

$$P \leq 2^{-MT}(1-2^{-MT})^{t-1}S_l^{t-1}. \quad (37)$$

Considering all values of t, we can derive the probability that a malicious state passes the verification is

$$Pr = \int_0^\infty 2^{-MT}(1-2^{-MT})^t S_l^t \ dt \leq \frac{2^{-MT}}{S_l}, \quad (38)$$

which is the conclusion.

Appendix D
Proof of Lemma 4

As shown in [35], let's assume that the M nodes share a dishonest state with

$$|W\rangle = |R_0^K\rangle|W_0\rangle + |R_1^K\rangle|W_1\rangle + |Y\rangle, \quad (39)$$

where the sates of honest nodes $|Y\rangle$ is orthogonal to both $|R_0^K\rangle$ and $|R_1^K\rangle$ and the dishonest state is not supposed to be normalized. The malicious nodes require to guess the honest output V_H before reporting their measured outcomes V_D so that they can pass the verification. In fact, the Helstrom measurement can provide the optimal guess of V_H from checking the malicious states for honest outcome $V_H = 0$ and $V_H = 1$. For convenience, denoting $\rho_{M-K} = Tr_K|Y\rangle\langle Y|$ as the reduced density operation when the honest nodes are traced out of the state $|Y\rangle$. Since $P_K = |R_0^K\rangle\langle R_0^K| + \frac{1}{2}I_{J_M}$ and $I - P_K$, with the input $U_j = 0$ of honest node n_j, it has

$$\begin{aligned} &Pr[\text{Guess}V_j|U_j = 0] \\ &= \frac{1}{2} + \frac{1}{2}||Tr_K[P_K \otimes I_{M-K}|W\rangle] - (I-P_K)\otimes I_{M-K}|W\rangle|| \\ &= \frac{1}{2} + \frac{1}{2}\sqrt{(|||W_0\rangle||^2 + |||W_1\rangle||^2)^2 - 4\langle W_0|W_1\rangle^2} \end{aligned} \quad (40)$$

where the trace norm is computed as the sum of the absolute values of the eigenvalues of the matrix. Accordingly, utilizing $Q_{M-1} = |R_+^M\rangle\langle R_-^M| + \frac{1}{2}I_{J_M}$ and $I - Q_{M-1}$ to measure perform measurement with $U_j = 1$, it has

$$\begin{aligned} &Pr[\text{Guess}V_j|U_j = 1] \\ &= \frac{1}{2} + \frac{1}{2}\sqrt{(|||W_+\rangle||^2 + |||W_-\rangle||^2)^2 - 4\langle W_+|W_-\rangle^2}. \end{aligned} \quad (41)$$

After defining $u = |||W_0\rangle||^2$ and $q = |||W_1\rangle||^2$ with the angle θ between two states, we can obtain $\langle W_+|W_-\rangle^2 =$

$\frac{1}{4}(|||W_0\rangle||^2 - |||W_1\rangle||^2)^2 = (u-v)^2/4$ and $|||W_+\rangle||^2 + |||W_-\rangle||^2 = |||W_0\rangle||^2 + |||W_1\rangle||^2 = u + v \leq 1$. Due to the randomness of U_j, it has

$$\begin{aligned} P &= \frac{1}{2}(Pr[\text{Guess}V_j|U_j=0] + Pr[\text{Guess}V_j|U_j=1]) \\ &= \frac{1}{2} + \frac{1}{4}(\sqrt{(u+v)^2 - 4uv\cos^2\theta} + 2\sqrt{uv}) \\ &\leq 1 - \frac{1}{4}\left(1 - \frac{(u-v)^2 + 4uv\sin^2\theta}{2}\right). \end{aligned} \tag{42}$$

Let's consider that the malicious nodes can take a local operation on their state to maximize their passing probability. Therefore, the distance of the malicious state from the honest one is $\epsilon = \min_U \mathcal{D}(I \otimes U)|W\rangle, |R_0^M\rangle = \min_U \sqrt{1 - F^2((I \otimes U)|W\rangle, |R_0^M\rangle)}$, where $F(\cdot,\cdot)$ is the fidelity between two states. Denoting the reduced density matrices of the honest nodes n_j as ρ_j and the perfect state as ρ_* respectively, there exists an operation C on the malicious sate so that

$$F(I \otimes C|W\rangle, |R_0^M\rangle) = F(\rho_*, \rho_j). \tag{43}$$

Therefore, it has

$$\epsilon^2 \leq 1 - F^2(I \otimes C|W\rangle, |R_0^M\rangle) = 1 - F^2(\rho_*, \rho_j). \tag{44}$$

It has

$$\begin{aligned} \rho_* &= \frac{1}{2}(|R_0^M\rangle\langle R_0^K| + |R_1^M\rangle\langle R_1^K|) \\ \rho_j &= u|R_0^K\rangle\langle R_0^K| + v|R_0^K\rangle\langle R_0^K| \\ &\quad + \sqrt{uv}\cos\theta(|R_0^K\rangle\langle R_1^K| + |R_1^K\rangle\langle R_0^K|) + f(|Y\rangle), \end{aligned} \tag{45}$$

where $f(|Y\rangle)$ is the function of $|Y\rangle$. It means

$$F(\rho_*, \rho_j) = tr^2(\sqrt{\rho_*^{1/2}\rho_j\rho_*^{1/2}}) = \frac{1}{2}(u + v + 2\sin\theta\sqrt{uv}), \tag{46}$$

which gives $\epsilon^2 \leq 1 - \frac{u+v}{2} - \sqrt{uv}\sin\theta$. Therefore, it has

$$P = 1 - \frac{1}{4}\left(1 - \frac{(u-v)^2 + 4uv\sin^2\theta}{2}\right) \leq 1 - \frac{\epsilon^2}{4} \tag{47}$$

The optimal cheating strategy of malicious nodes that maximizes the probability of passing verification is to prepare a state $|W\rangle$ with $\mathcal{D}(U|W\rangle, |R_0^M\rangle) = \epsilon$, where U is an operation malicious. In fact, if the nodes utilize a state with smaller distance, it does not contribute to the probability of passing the verification, and even worse, it leads to large resource consumption. On the other hand, if the nodes utilize a state with distance larger than ϵ, it increases the probability of failed verification. Therefore, the optimal strategy for malicious nodes is to set the distance $\epsilon = \mathcal{D}(U|W\rangle, |R_0^M\rangle)$, under which, the malicious state passes the verification with probability $P \leq 1 - \frac{\epsilon^2}{4}$.

Furthermore, as shown in Algorithm 3, before the state verification, coin flipper *CF* tosses T coins to decide the selection of verification and voting. In fact, the node will vote when all T tossing results are the same, which means that the probability of voting is 2^{-T}. Therefore, the probability that the nodes verify the state in the t-th iteration is $(1-2^{-T})^{t-1}$. Since there are S_l fraction of malicious nodes in shard, the verifier *VF* is malicious with probability S_l and is honest with probability S_l. Furthermore, the malicious verifier will pass the dishonest state with probability 1, while the honest verifier will pass the state with probability $1 - \frac{\epsilon^2}{4}$. Therefore, as shown in Algorithm 3, the nodes will vote when all checks are correct and the T tossing results are the same. Denoting $p = S_l \times 1 + (1 - S_l) \times \left(1 - \frac{\epsilon^2}{4}\right)$, the probability of malicious state passing the verification at the t-th iteration is

$$P \leq 2^{-T}(1-2^{-T})^{t-1}p^{t-1} \tag{48}$$

Considering all values of t, we can derive the probability that a malicious state passes the verification is

$$Pr = \int_0^\infty 2^{-T}(1-2^{-T})^t p^t \, dt \leq \frac{4 \times 2^{-T}}{\epsilon^2(1-S_l)}, \tag{49}$$

where the last inequality holds since $p = S_l \times 1 + (1 - S_l) \times \left(1 - \frac{\epsilon^2}{4}\right) \geq 2^{-T}$, which is the conclusion.

Appendix E
Proof of Theorem 1

Lemmas 1 and 2 guarantee that the honest state can always pass the verification. Furthermore, Lemmas 4 and 3 show that the illegal state can not pass the verification. Therefore, combining these lemmas guarantees that Q-Union can defend the disturbing attack. Additionally, Q-Union can also defend the double voting attack. In fact, double voting is taken care of easily if the number of nodes in each shard is known in advance, which in fact is necessary in Q-Union to prepare the shared quantum state and guarantee that there exist at least one honest node in each shard. Therefore, the conclusion holds.

Appendix F
Proof of Theorem 2

The malicious nodes can take attacks during the distribution of entangled Aharonov state, where they entangle the states with an auxiliary system prepared in advance and return the operated particles to honest nodes. To avoid the detection in state verification in Algorithm 4, it requires all measured outcomes are distinct. Since there are at most MS_l malicious nodes in the l-th consensus iteration, the malicious nodes require to determine the secret number of the other $M(1-S_l)$ honest nodes, which in fact form a set $\mathcal{A}_M^{(1-S_l)M}$ being the set of all $(1-S_l)M$ permutations of $\mathbb{Z}_M$ and $|\mathcal{A}_M^{(1-S_l)M}| = \frac{M!}{(1-S_l)M!}$. In fact, $\mathcal{A}_M^{(1-S_l)M}$ can be divided into $\binom{M}{(1-S_l)M}$ subsets, each of which corresponds to the set of all $(1-S_l)M!$ permutations of a $(1-S_l)M$-combination of $\mathbb{Z}_M$. Furthermore, any two elements in two subsets of $\mathcal{A}_M^{(1-S_l)M}$ with non-equal value should be orthogonal to each other, otherwise, the malicious nodes cannot deterministically know the subset of the honest nodes' measured outcomes and distinguish the correct secret numbers of honest nodes. However, due to the nature of Aharonov state, the malicious nodes can at most know the information about which subset the honest nodes' index numbers are in. However, this general entangle-measure attack is trivial in the sense that without any attack the malicious can cooperate to obtain this information.

Therefore, the malicious nodes cannot obtain the identity of honest nodes by eavesdropping distributed Aharonov state.

Furthermore, it also cannot distinguish the node identity from voting procedure. In order to tap the identity information of honest nodes, the only way that the adversaries can take is to replace the GHZ state $|G\rangle$ by a malicious state $|B\rangle$. From the Lemma 4, the fake state $|B\rangle$ requires to satisfy $\mathcal{D}(|B\rangle, |G\rangle) \leq \epsilon$ so that the state can pass the verification. For any POVM element P, we can write the trace distance between two states ρ and σ, as $Tr[P(\rho - \sigma)] \leq \mathcal{D}(\rho, \sigma)$. Therefore, for the fake state $|R\rangle$ and the honest state $|G\rangle_i$ for $1 \leq i \leq M$, it has

$$Tr(P(|G\rangle_i\langle G|_i)) - Tr(P(|B\rangle\langle B|)) \leq \epsilon. \tag{50}$$

Simply assuming that the probability of each node eavesdropped by malicious nodes is the same, which means

$$\begin{aligned} P[\text{Guess identity}] &= \sum_{i=1}^{(1-S_l)M} \frac{1}{(1-S_l)M} Tr(P(|G\rangle_i\langle G|_i)) \\ &\leq \frac{1}{(1-S_l)M} + \epsilon \end{aligned} \tag{51}$$

which is the conclusion.

Appendix G
Proof of Lemma 5

Since there are α adversaries among N nodes, the probability of a randomly selected node in shard with size M being an adversary is $\beta = \frac{\alpha}{N}$. Therefore, for any ratio $\beta < S_l < 1$, to guarantee there are at most $f = S_l \times M$ adversaries in any shard with a probability larger than or equal to $1 - 10^{-b}$, it requires to find the minimum size M of shard. It can be seen the probability is $P(k \leq f) = \sum_{k=0}^{f} \binom{M}{k} \beta^k (1-\beta)^{M-k} \geq 1 - \frac{10^{-b}}{K}$, where K is the number of shards. Since the number M of nodes is large in the blockchain system, we can apply Gaussian distribution to approximate the above binomial distribution, where the mean is $\mu = \beta \times M$ and the variance is $\beta(1-\beta) \times M$. It means the probability can be rewritten as $N\left(\frac{f-\mu}{\sigma}\right) = N\left(\frac{f-\beta\times M}{\sqrt{\beta(1-\beta)\times M}}\right) \geq \left(1 - \frac{10^{-b}}{K}\right)$. Therefore, after defining a Gaussian variable z with the cumulative distribution probability $N(z) = \left(1 - \frac{10^{-b}}{K}\right)$, it has $M \geq \frac{\beta(1-\beta)}{\left(\frac{S_l-\beta}{z}\right)^2}$, which is the conclusion.

Appendix H
Proof of Corollary 1

Since only one shard is involved at the first loop of consensus procedure, it requires us to guarantee that the consensus algorithm can not be terminated at the beginning of consensus protocol. Therefore, any shard requires to have at least one honest node. According to Lemma 5, the shard size M satisfies $M = \frac{\beta(1-\beta)}{\left(\frac{S_1-\beta}{z}\right)^2}$, where $\beta = 1/2$ and $S_1 = 1 - 1/M$. It means

$$M\left(\frac{1}{2} - \frac{1}{M}\right)^2 = \frac{1}{4}z^2. \tag{52}$$

Therefore, when $z = 3$, the probability that at least one honest node in each shard is larger than 99.9%, under which, it can obtain that $M \geq 10$, which is the conclusion.

Appendix I
Proof of Theorem 3

In order to obtain the conclusion, we require the following lemmas.

Lemma I1: The adversary cannot compel the activation of a new loop if voting process has terminated in previous loop.

Proof: When a new loop requires to be activated, some nodes in the starting shard h_s will send their vote summaries to the nodes in current shard h_l to request the nodes in current shard to activate a new voting loop. After receiving the vote summaries, the nodes in the current shard h_l will further receive all vote summaries from the starting shard h_s. It is shown in Corollary 1 that there is at least one honest node in the starting shard due to the design of shard size. Therefore, the nodes in current shard h_l can check the vote summaries and arrive to her conclusion on activation of new voting loop, which is the conclusion.□

Lemma I2: An adversarial shard leader who can be identified through additional measures.

Proof: The adversarial shard leader can isolate an honest node within the shard by withholding a block, hindering its verification and voting. This scenario typically occurs between the leader and the first nodes involved in the first loop. Since subsequent nodes can synchronize with at least one honest node from the starting shard, the following approach can be applied to counter an adversarial shard leader. When the leader fails to respond to a node n_j. Node n_j requests the block from the other nodes in starting shard h_s to act as middlemen in retrieving this block from the leader. Upon reception, these nodes relay it to node n_j. If the leader remains unresponsive, they communicate this silence to node n_j using a signed message. If the number of responded nodes is insufficient, node n_j continues engaging nodes from shard $h_s \cap h_{s+1}$ until the likelihood of all nodes confirming the leader's silence being adversarial falls below 10^{-b}. Subsequently, node n_j sends these signed messages to her assistants. This shard leader will be penalized when these signed messages are sent to the blockchain.

To prevent DDOS attacks, the shard leader requests a signed receipt from middleman nodes. When the number of collected receipts suggests that the leader indeed sent the block, and at least one receipt is from an honest node, the leader sends these receipts to the witness network and redirects future middlemen to the witness network for these receipts. Notably, except for only one node, all the other nodes in shard are supposed to receive the block from leader. Thus, we expect node n_j could obtain the block without requiring too many middlemen.□

Combining these two lemmas, we can obtain the conclusion. Lemma I1 first ensures new loop activation when a decision cannot be reached, resilient against adversarial influence. It further prevents new loop activation once a decision is reached and known to an honest node in the latest M nodes of the current union shard, minimizing the number of involved loops. Lemma I2 shows measures to identify adversarial shard leaders.

Furthermore, the adversaries cannot send more than S_l fraction of malicious votes in the total involved $l \times M$ nodes

except for a negligible probability. This ensures decisions are supported by honest nodes. Therefore, these lemmas guarantee the safety and liveness of outputting decisions in the voting process.

Finally, the voting decisions are written to the decision block in Consensus Determination and are voted on by all nodes in $\mathcal{N}$. Each blockchain epoch concludes when the decision block is accepted. Therefore, the security upper bound is, the maximum number $\beta \leq \lfloor (N-1)/2 \rfloor$ of adversarial nodes involved to make consensus on decision block, which resembles that of a traditional non-sharded blockchain.

REFERENCES

[1] J. Xu, C. Wang, and X. Jia, "A survey of blockchain consensus protocols," *ACM Comput. Surveys*, vol. 55, no. 13s, pp. 1–35, Dec. 2023.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," Tech. Rep., 2008, vol. 4, no. 2, p.15.

[3] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 839–858.

[4] X. Luo et al., "EtherCloak: Enabling multi-level and customized privacy on account-model blockchains," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 1, pp. 771–786, Jan. 2025, doi: 10.1109/TDSC.2024.3418617.

[5] S. Feng, W. Wang, Z. Xiong, D. Niyato, P. Wang, and S. S. Wang, "On cyber risk management of blockchain networks: A game theoretic approach," *IEEE Trans. Services Comput.*, vol. 14, no. 5, pp. 1492–1504, Sep. 2021.

[6] W. Zou et al., "Smart contract development: Challenges and opportunities," *IEEE Trans. Software Eng.*, vol. 47, no. 10, pp. 2084–2106, Oct. 2021.

[7] K. Xue, X. Luo, H. Tian, J. Hong, D. S. L. Wei, and J. Li, "A blockchain based user subscription data management and access control scheme in mobile communication networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 3108–3120, Mar. 2022.

[8] J. Xu, K. Xue, H. Tian, J. Hong, D. S. L. Wei, and P. Hong, "An identity management and authentication scheme based on redactable blockchain for mobile networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6688–6698, Jun. 2020.

[9] M. Du, Q. Chen, J. Xiao, H. Yang, and X. Ma, "Supply chain finance innovation using blockchain," *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1045–1058, Nov. 2020.

[10] B. Wang, X. Yuan, L. Duan, H. Ma, C. Su, and W. Wang, "DeFiScanner: Spotting DeFi attacks exploiting logic vulnerabilities on blockchain," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 2, pp. 1577–1588, Feb. 2024.

[11] S. Kumar et al., "Beet: Blockchain enabled energy trading for e-mobility oriented electric vehicles," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 3018–3034, Apr. 2024.

[12] K. Xue, X. Luo, Y. Ma, J. Li, J. Liu, and D. S. Wei, "A distributed authentication scheme based on smart contract for roaming service in mobile vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5284–5297, May 2022.

[13] M. Li et al., "S3 Voting: A blockchain sharding based E-voting approach with security and scalability," *IEEE Trans. Dependable Secure Comput.*, vol. 22, no. 2, pp. 1596–1611, Mar. 2025, doi: 10.1109/TDSC.2024.3446392.

[14] Z. Ning et al., "Blockchain-enabled intelligent transportation systems: A distributed CrowdSensing framework," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4201–4217, Dec. 2022.

[15] S. Warnat-Herresthal et al., "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021.

[16] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1432–1465, 2nd Quart., 2020.

[17] C. Li et al., "A decentralized blockchain with high throughput and fast confirmation," in *Proc. USENIX Annu. Tech. Conf.*, 2020, pp. 515–528.

[18] Y. Xu, J. Zheng, B. Düdder, T. Slaats, and Y. Zhou, "A two-layer blockchain sharding protocol leveraging safety and liveness for enhanced performance," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2024, pp. 1–12.

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

[20] S. Banerjee, A. Mukherjee, and P. K. Panigrahi, "Quantum blockchain using weighted hypergraph states," *Phys. Rev. Res.*, vol. 2, no. 1, Mar. 2020, Art. no. 013322.

[21] S. N. Paing et al., "Counterfactual quantum Byzantine consensus for human-centric metaverse," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 4, pp. 905–918, Apr. 2024.

[22] Q. Li, J. Wu, J. Quan, J. Shi, and S. Zhang, "Efficient quantum blockchain with a consensus mechanism QDPoS," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 3264–3276, 2022.

[23] M. Fitzi, N. Gisin, and U. Maurer, "Quantum solution to the Byzantine agreement problem," *Phys. Rev. Lett.*, vol. 87, no. 21, Nov. 2001, Art. no. 217901.

[24] Q. Wang, C. Yu, F. Gao, H. Qi, and Q. Wen, "Self-tallying quantum anonymous voting," *Phys. Rev. A, Gen. Phys.*, vol. 94, no. 2, Aug. 2016, Art. no. 022333.

[25] M. Bonanome, V. Bužek, M. Hillery, and M. Ziman, "Toward protocols for quantum-ensured privacy and secure voting," *Phys. Rev. A, Gen. Phys.*, vol. 84, no. 2, Aug. 2011, Art. no. 022331.

[26] M. Arapinis, N. Lamprou, E. Kashefi, and A. Pappa, "Definitions and security of quantum electronic voting," *ACM Trans. Quantum Comput.*, vol. 2, no. 1, pp. 1–33, Mar. 2021.

[27] C. Cachin and M. Vukolic, "Blockchain consensus protocols in the wild (Keynote talk)," in *Proc. 31st Int. Symp. Distrib. Comput. (DISC)*, Jan. 2017, p. 16.

[28] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd USENIX Symp. Operating Syst. Design Implement.*, 1999, pp. 173–186.

[29] A. K. Fedorov, E. O. Kiktenko, and A. I. Lvovsky, "Quantum computers put blockchain security at risk," Tech. Rep., 2018, vol. 563, no. 7732, pp. 465–467.

[30] K. Ikeda, "Security and privacy of blockchain and quantum computation," *Adv. Comput.*, vol. 111, pp. 199–228, Jan. 2018.

[31] M. Ben-Or and A. Hassidim, "Fast quantum Byzantine agreement," in *Proc. 37th Annu. ACM Symp. Theory Comput.*, May 2005, pp. 481–485.

[32] L. Mazzarella, A. Sarlette, and F. Ticozzi, "Consensus for quantum networks: Symmetry from Gossip interactions," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 158–172, Jan. 2015.

[33] M. Fitzi, N. Gisin, U. Maurer, and O. von Rotz, "Unconditional Byzantine agreement and multi-party computation secure against dishonest minorities from scratch," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Amsterdam, The Netherlands. Cham, Switzerland: Springer, Apr. 2002, pp. 482–501.

[34] A. Pappa, A. Chailloux, S. Wehner, E. Diamanti, and I. Kerenidis, "Multipartite entanglement verification resistant against dishonest parties," *Phys. Rev. Lett.*, vol. 108, no. 26, Jun. 2012, Art. no. 260502.

[35] A. Unnikrishnan, I. J. MacFarlane, R. Yi, E. Diamanti, D. Markham, and I. Kerenidis, "Anonymity for practical quantum networks," *Phys. Rev. Lett.*, vol. 122, no. 24, Jun. 2019, Art. no. 240501.

[36] C. Liu et al., "Extending on-chain trust to off-chain–trustworthy blockchain data collection using trusted execution environment (TEE)," *IEEE Trans. Comput.*, vol. 71, no. 12, pp. 3268–3280, Dec. 2022.

[37] A. Cabello, "N-ParticleN-level singlet states: Some properties and applications," *Phys. Rev. Lett.*, vol. 89, no. 10, Aug. 2002, Art. no. 100402.

[38] B. Chen, T. Ma, and S.-M. Fei, "Entanglement detection using mutually unbiased measurements," *Phys. Rev. A, Gen. Phys.*, vol. 89, no. 6, Jun. 2014, Art. no. 064302.

[39] C. Spengler, M. Huber, S. Brierley, T. Adaktylos, and B. C. Hiesmayr, "Entanglement detection via mutually unbiased bases," *Phys. Rev. A, Gen. Phys.*, vol. 86, no. 2, Aug. 2012, Art. no. 022311.

[40] J. Fulconis, O. Alibart, W. J. Wadsworth, and J. G. Rarity, "Quantum interference with photon pairs using two micro-structured fibres," *New J. Phys.*, vol. 9, no. 8, p. 276, Aug. 2007.

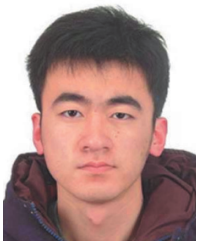
[41] S. Arahira, T. Kishimoto, and H. Murai, "1.5-μm band polarization entangled photon-pair source with variable bell states," *Opt. Exp.*, vol. 20, pp. 9862–9875, Apr. 2012.

[42] C. A. Fuchs and J. van de Graaf, "Cryptographic distinguishability measures for quantum-mechanical states," *IEEE Trans. Inf. Theory*, vol. 45, no. 4, pp. 1216–1227, May 1999.

[43] R. Batra and R. Jain, "Commitments are equivalent to statistically-verifiable one-way state generators," in *Proc. IEEE 65th Annu. Symp. Found. Comput. Sci. (FOCS)*, Oct. 2024, pp. 1178–1192.

[44] Q. Wang, "Optimal trace distance and fidelity estimations for pure quantum states," *IEEE Trans. Inf. Theory*, vol. 70, no. 12, pp. 8791–8805, Dec. 2024.

Chenhao Ying received the B.E. degree from the Department of Communication Engineering, Xidian University, China, in 2016, and the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, in 2022. He is currently a Research Assistant Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His current research interests include mobile crowd sensing, blockchain, and quantum computing.



Yuxuan Du received the Ph.D. degree from the School of Computer Science, The University of Sydney. He is currently an Assistant Professor with the College of Computing and Data Science, Nanyang Technological University, Singapore. Before that, he was a Senior Researcher with the JD Explore Academy, China. His research interests include fundamental algorithms for quantum machine learning, quantum learning theory, and AI for quantum science. He has published his research outcomes in many top-tier journals and conferences in physics and computer science, including *Physical Review Letters*, *Physical Review X Quantum*, IEEE TRANSACTIONS ON INFORMATION THEORY, International Conference on Learning Representations, and the Conference on Neural Information Processing Systems.

Weiting Zhang (Member, IEEE) received the Ph.D. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2021. From November 2019 to November 2020, he was a Visiting Ph.D. Student with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Since December 2021, he has been an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. His research interests include the industrial Internet of Things, federated learning, edge intelligence, and machine learning for network optimization.

Xikun Jiang received the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China, in 2023. She currently holds a post-doctoral position with the Department of Computer Science (DIKU), University of Copenhagen (UCPH), Denmark. Her research interests include mobile crowdsensing, online markets, and machine learning.

Gang Wang received the B.S. and M.S. degrees in 2015 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Northeastern University, Shenyang. His research interests include blockchain, federated learning, and natural language processing.

Haiming Jin (Member, IEEE) received the B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2012, and the Ph.D. degree from the University of Illinois at Urbana-Champaign (UIUC), Urbana, IL, USA, in 2017. He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. Before this, he was a Post-Doctoral Research Associate with the Coordinated Science Laboratory, UIUC. His research interests include crowd and social sensing systems, reinforcement learning, and mobile pervasive and ubiquitous computing.

Jie Li (Fellow, IEEE) received the B.E. degree in computer science from Zhejiang University, Hangzhou, China, the M.E. degree in electronic engineering and communication systems from China Academy of Posts and Telecommunications, Beijing, China, and the Ph.D. degree from The University of Electro-Communications, Tokyo, Japan. He was a Full Professor with the Department of Computer Science, University of Tsukuba, Japan. He was a Visiting Professor with Yale University, USA, Inria, France. He is currently a Chair Professor with the Department of Computer Science and Engineering and the Director of the Blockchain Research Centre, Shanghai Jiao Tong University (SJTU), Shanghai, China. His research interests include big data and AI, blockchain, network systems, and security. He is the Co-Chair of the IEEE Technical Community on Big Data, the Founding Chair of the IEEE ComSoc Technical Committee on Big Data, and the Co-Chair of the IEEE Big Data Community.

Yuan Luo (Member, IEEE) received the B.S. degree in applied mathematics and the M.S. and Ph.D. degrees in probability statistics from Nankai University, Tianjin, China, in 1993, 1996, and 1999, respectively. From July 1999 to April 2001, he was a Post-Doctoral Researcher with the Institute of Systems Science, Chinese Academy of Sciences, Beijing, China. From May 2001 to April 2003, he was a Post-Doctoral Researcher with the Institute for Experimental Mathematics, University of Duisburg-Essen, Essen, Germany. Since June 2003, he has been with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. Since 2006, he has been a Full Professor. His current research interests include coding theory, information theory, and big data analysis.

Dacheng Tao (Fellow, IEEE) is currently a Distinguished University Professor with the College of Computing and Data Science, Nanyang Technological University. He mainly applies statistics and mathematics to artificial intelligence, and his research is detailed in one monograph and over 300 publications in prestigious journals and proceedings at leading conferences, with best paper awards, best student paper awards, and test-of-time awards. His publications have been cited over 140K times and he has an H-index more than 180 in Google Scholar. He is a fellow of Australian Academy of Science, AAAS, and ACM. He received the 2015 and 2020 Australian Eureka Prize, the 2018 IEEE ICDM Research Contributions Award, the 2020 Research Super Star by The Australian, the 2019 Diploma of The Polish Neural Network Society, and the 2021 IEEE Computer Society McCluskey Technical Achievement Award.