Quantum Annealing based Consensus Mechanism

Shubham Joshi, Matthew Jackson, Rasheed Hussain, Zuobin Xiong, and Junggab Son*

***Abstract*—The rise of quantum computing presents both challenges and opportunities for blockchain technology, necessitating the development of consensus mechanisms that are secure, scalable, and energy-efficient. This paper introduces a novel consensus protocol that leverages quantum annealing to address the limitations of classical approaches. Our proposed system selects a single representative node to build and propose the next block by formulating the selection process as a Quadratic Unconstrained Binary Optimization (QUBO) problem, which is then solved by a quantum annealer. The selection is based on a comprehensive, weighted suitability score derived from verifiable node metrics, including uptime, network latency, throughput, and historical on-chain performance. We provide a thorough security analysis demonstrating the mechanism's theoretical robustness against Sybil, block withholding, and collusion attacks. The result is a hybrid classical-quantum framework that presents a promising path toward a secure and efficient consensus model for the post-quantum era.**

***Index Terms*—Quantum Annealing, Blockchain, Quantum Computing, Cryptography, Decentralization, Consensus Mechanism.**

I. Introduction

Blockchain technology represents a paradigm shift in data management, offering a decentralized and immutable ledger distributed across a peer-to-peer network [1]. In this architecture, there is no central authority to validate transactions or maintain the integrity of the record. Instead, network participants, or nodes, must communicate and collaborate to ensure that every member holds an identical and accurate copy of the ledger. This decentralized nature, while foundational to the security and transparency of blockchain, introduces a fundamental challenge: achieving agreement among a diverse and potentially untrustworthy set of participants.

This challenge is addressed by a consensus mechanism, the procedural and algorithmic heart of any blockchain [2]. It is the protocol that enables nodes to agree on the state of the ledger, ensuring that new blocks of transactions are validated and added to the chain in a consistent and tamper-resistant manner. Classical consensus mechanisms like Proof-of-Work (PoW) [3] and Proof-of-Stake (PoS) [4] have been foundational, yet they each present inherent trade-offs involving immense energy consumption, potential for centralization, or limited scalability. As technology advances, the search for more efficient, secure, and equitable consensus models continues [5].

The advent of quantum computing marks the next frontier for this evolution, introducing both new threats and unprecedented opportunities in enhancing the blockchain [6]. The rise of quantum capabilities has shown the development of more efficient and secure blockchains, leading to the emergence of quantum consensus research as discussed in Section II. This field explores how the principles of quantum mechanics can be harnessed to design consensus protocols that are not only secure against quantum attacks but also fundamentally more robust and efficient than their classical counterparts.

The main contribution of this paper can be summarized as follows:

- A new hybrid consensus mechanism has been proposed where a quantum annealer selects a representative node by solving a Quadratic Unconstrained Binary Optimization (QUBO) problem.
- A comprehensive suitability score has been introduced which is based on multiple weighted performance metrics and a unique score perturbation method to serve as a tie-breaker, ensuring fair and unambiguous selection.
- A thorough security analysis has been provided demonstrating that the proposed protocol is theoretically resistant to common attacks such as Sybil, block withholding, and collusion.

The rest of this paper is organized as follows. Section II reviews related work on integrating quantum mechanics with blockchain consensus , covering quantum-enhanced Proof of Authority, stake-based models, and Proof of Quantum Work. Section III covers preliminaries, laying the foundation for our system with introductions to blockchain technology , quantum computing principles , and the QUBO and Ising models. Section IV defines the problem through the system model, threat model, and design goals. Our proposed scheme, including the node scoring metrics, the Probe Protocol, and the QUBO objective function, is presented in Section V. Section VI provides a formal security analysis demonstrating the protocol's theoretical resistance to Sybil, block withholding, and collusion attacks. Section VII discusses the implementation specifics of the simulation environment and presents the results. Finally, Section VIII concludes with a summary of our contributions.

II. Related Work

Several research efforts aim to enhance classical blockchain consensus mechanisms by integrating quantum mechanics, focusing on improving security, decentralization, and efficiency. One area of focus is Proof of Authority (PoA), where randomness is introduced to make leader selection less predictable. For instance, Yuan and Wang (2024) developed VB-PoA, which uses Verifiable Random Functions (VRFs) to randomly select block proposers, while Wang et al. (2024) proposed RQPOA, which employs a Verifiable Delay Function (VDF) for fair leader election. RQPOA further enhances security by using a multi-party Quantum Secret Sharing (QSS) protocol to protect the leader's identity and a quantum threshold signature for fault-tolerant voting [7], [8].

Another popular approach is to create quantum versions of stake and voting-based consensus models. Lin et al. (2024) adapted Proof of Vote (PoV) for consortium blockchains with their Q-PnV mechanism, which integrates quantum voting, quantum digital signatures (QDS), and quantum random number generators (QRNGs) [9]. Similarly, Li et al. (2022) introduced Quantum Delegated Proof of Stake (QDPoS), which also utilizes quantum voting protocols for rapid and secure decentralization [10]. Paul et al. (2025) proposed a Quantum Proof of Stake and Behavior (QPoSB) that combines a Quantum Hash Function (QHF) with an optimized Borda count voting method to fortify the blockchain against quantum attacks [11].

Researchers are also leveraging fundamental quantum properties to devise novel consensus protocols. Two separate works by Wen et al. (2022) explore consensus based on quantum teleportation and quantum zero-knowledge proofs (QZKP), respectively [12], [13]. Both methods utilize the inherent randomness and irreversibility of quantum measurement. Along similar lines, Wang and Yu (2022) presented a protocol based on the stochasticity of quantum measurement, incorporating Quantum Key Distribution (QKD) and a quantum voting mechanism [14]. All these approaches claim to offer unconditional security, resistance to 51% attacks, and significant reductions in energy consumption and latency compared to classical methods.

Efforts in Quantum Byzantine Fault Tolerance (BFT) aim to improve agreement in hostile environments. Gao et al. (2022) enhanced existing Quantum Byzantine Agreement Protocols (QBAP) by adding a trust-value calculation and a dual signature method to identify and prevent malicious nodes from disrupting consensus [15]. Paing et al. (2024) introduced a counterfactual quantum BFT (CQ-BFT) for Metaverse applications, which achieves consensus without the physical passage of information-carrying particles, thereby offering robustness against dephasing noise [16].

For specific industries, tailored quantum consensus mechanisms are being developed that often rely on quantum random number generation and entanglement. For smart forestry, Damaševičius and Maskeliūnas (2024) presented "Quantum Forest" which uses verifiable quantum random numbers (QRNGs) for validator selection [17]. Likewise, Akoramurthy and Surendiran (2024) introduced "QHealth" for smart healthcare, a method that also uses QRNGs to select inspector nodes [18]. Both simulated systems reported enhanced security, throughput, and energy efficiency for their respective real-time data applications.

In a distinct approach that directly targets the mining process, Amin et al. (2025) proposed Proof of Quantum Work (PoqW). Prototyped on D-Wave quantum annealers, this framework aims to replace classical mining entirely. It leverages quantum supremacy by designing a mining process that is only feasible for quantum computers, thus rendering it intractable for classical systems. This strategy seeks to drastically cut the high energy use of traditional PoW while establishing a quantum-safe security layer [19].

III. Preliminaries and Backgrounds

This sections will help you to grasp the foundation of blockchain, quantum computing and cryptographic primitives that will be required to understand the topics discussed later in the paper.

A. Notations

This subsection will present all the notations that have been used throughout this paper. It aims to provide a clear understanding of the various notations used in algorithms defined in later sections. Table I provides the notations and its description.

TABLE I
Notations and their descriptions

Notation	Description
N	Number of nodes in decentralized network
$H(\cdot)$	Cryptographic hash function
pk	Public key in cryptographic systems
sk	Private (secret) key in cryptographic systems
$\|\psi\rangle$	Represents the superposition state of a qubit
α, β	Complex amplitudes in the qubit superposition state
x_i	A binary variable, 1 if node i is chosen, 0 otherwise
S_i	Original suitability score for a node
S_i'	The effective (perturbed) suitability score for node i
P	A large positive number used as a penalty coefficient
σ_i^z	The Pauli-Z operator for qubit i
h_i	Local external fields (biases) acting on spin i
J_{ij}	Coupling strengths between interacting pairs of spins (i, j)
n_s, n_t	Source and target nodes in the Probe Protocol
W	A set of witness nodes
r	A unique nonce in the Probe Protocol
$\mathcal{U}(n_x)$	The total uptime for a node n_x
δ	Maximum delay tolerance for uptime verification
C_p	Proposal Success Count for a node
w_p	A protocol-defined weight constant for the performance score
δ_i	A small perturbation value used to break ties in scores
Q_{ii}, Q_{ij}	Linear and quadratic coefficients in the QUBO model
C	A constant offset in the QUBO model

B. Blockchain

A blockchain is a specific type of distributed database, characterized by its structure and cryptographic security. It functions as an append-only ledger, meaning data is added in sequential "blocks" and once added, it's exceedingly difficult to alter. Each block contains a cryptographic hash of the previous block, effectively creating a chain and ensuring data integrity. This chaining mechanism, coupled with the distribution of the ledger across a network of nodes, makes blockchains highly resistant to tampering and single points of failure.

1) Blockchain Consensus Mechanism: In a blockchain network, a consensus mechanism is a critical process that enables a distributed network of nodes to reach agreement on the validity of transactions and the state of the shared ledger. Instead of relying on a central authority, these mechanisms ensure that all participating nodes maintain a consistent and secure record of transactions. There are a plethora of consensus mechanisms developed based on differing principles of trust, scalability, energy efficiency, and fault tolerance [20].

C. Quantum Computing

Quantum computing is a field that utilizes the properties of quantum mechanics to solve various computing problems. Unlike classical computing, whose information storing relies on bits, quantum computing fundamentally relies on qubits, or quantum bits for information storage.

Mathematically, it is represented as a vector in a two-dimensional complex space, with basis states $|0\rangle$ and $|1\rangle$. Physically, this can be realized through properties like photon polarization or electron spin. But we won't go much in detail about the physical properties. The key advantage of a qubit is its ability to exist in a superposition of states, enabling the simultaneous computation of multiple possibilities.

Superposition is a fundamental property in quantum mechanics, wherein a particle can simultaneously exist in multiple states [21]. A notable example of this phenomenon is a particle being in two different locations at once, signifying that its wave function assumes values in multiple places. While this concept may seem counterintuitive when considering particles, it is entirely natural when applied to waves. In quantum computing, superposition significantly enhances computational power beyond the capabilities of classical computers.

In the context of a qubit, which is analogous to a classical bit with binary states, the $|0\rangle$ and $|1\rangle$ states correspond to distinct orientations. However, unlike a classical bit, a qubit can also exist in a superposition—a linear combination of $|0\rangle$ and $|1\rangle$. The probability of measuring a particular state depends on the qubit's orientation within the superposition. A greater alignment with the $|0\rangle$ state increases the likelihood of obtaining $|0\rangle$ upon measurement, while a stronger alignment with $|1\rangle$ increases the probability of measuring $|1\rangle$. When the qubit is in a perfectly balanced superposition, both states have equal probability of being observed.

Mathematically, the superposition state $|\psi\rangle$ of a qubit is represented by the linear combination:

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

where α and β are complex amplitudes. The probabilities of measuring the states $|0\rangle$ and $|1\rangle$ are given by $|\alpha|^2$ and $|\beta|^2$, respectively, constrained by the normalization condition:

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2}$$

This formalism elucidates the probabilistic nature of quantum measurement and the inherent superposition of quantum states.

D. QUBO and Ising Models

Quantum annealers are natively designed to solve problems that can be expressed in specific mathematical forms, primarily the QUBO problem and its equivalent, the Ising model.

Definition 1 (Quadratic Unconstrained Binary Optimization (QUBO))**.** *A **QUBO** problem involves finding the minimum of a quadratic objective function of binary variables. These variables, denoted as x_i, can only take the value of 0 or 1. The general form of the objective function to be minimized is:*

$$f(x) = \sum_i q_i x_i + \sum_{i<j} Q_{ij} x_i x_j \tag{3}$$

Here, $x \in \{0,1\}^n$ is a vector of n binary variables, q_i represents the linear coefficient (or bias) for each variable x_i, and Q_{ij} is the quadratic coefficient for the interaction between variables x_i and x_j.

The Ising model, originating from statistical mechanics, describes the behavior of magnetic spins in a lattice. It is typically formulated with spin variables σ_i that can take values of +1 or -1. The problem Hamiltonian, H_z, which current quantum annealers aim to find the ground state of, is generally expressed as:

$$H_z = \sum_i h_i \sigma_i^z + \sum_{\langle i,j\rangle} J_{ij} \sigma_i^z \sigma_j^z \tag{4}$$

where h_i are local external fields (biases) acting on spin i, and J_{ij} are coupling strengths between interacting pairs of spins $\langle i,j\rangle$. The σ_i^z represent the Pauli-Z operator for qubit i.

These two formulations are mathematically equivalent through a simple linear transformation ($x_i = (\sigma_i^z + 1)/2$). This equivalence is crucial because the physical interactions within quantum annealing hardware are designed to mimic the Ising Hamiltonian, making QUBO and Ising problems the natural input for these devices. Many NP-hard optimization problems, such as Max-Cut, scheduling, and graph coloring, can be mapped onto QUBO or Ising formulations. However, this mapping process itself can be a significant hurdle. The efficiency of the mapping, the potential introduction of large coefficient ranges, the addition of penalty terms to enforce constraints, and the need to embed the problem's logical connectivity onto the hardware's physical qubit connectivity can all impact the feasibility and performance of QA. Problems that map more naturally or with less overhead to the hardware's native structure are inherently better candidates for demonstrating quantum advantage.

IV. Problem Definition

A. System Model

Our system operates as a hybrid system where classical computers manage network communication and data collection, while a quantum annealer performs the core optimization task of selecting a representative node.

The system is composed of N number of nodes that are categorized by their roles. A single "Representative Node" is chosen through the consensus protocol to build and propose the next block, while all other participating "Validator Nodes" are responsible for validating the consensus results and the proposed blocks. Any node is eligible to participate, and the opportunity to become a representative is not predetermined but emerges from a node's real-time and historical performance metrics. All nodes, regardless of their role, maintain a verified copy of the blockchain.

Nodes are interconnected in a peer-to-peer (P2P) network, which facilitates the constant exchange of information necessary for calculating suitability scores, which is further discussed in Section V-B. Nodes can communicate directly with each other and interact with a nearby quantum annealer to submit the objective function for solving and receive back the results as shown in Fig. 2. The network operates under a synchronous assumption. All participating nodes must be synchronized with a trusted time server, providing a common and reliable time reference. This synchronized time is critical for all timestamp-based calculations, particularly within the "Probe Protocol" used for measuring uptime and latency, and for defining the maximum delay tolerance (δ). This removes ambiguity in time-sensitive measurements across the distributed system. Furthermore, the network operates with a globally agreed-upon block proposal timeout. A selected representative node must broadcast a valid block within this time window; failure to do so is considered a proposal failure event. To ensure the integrity of performance metrics, the Probe Protocol utilizes randomized witness selection. For each probe, a set of k witness nodes is chosen at random from the entire pool of active network participants. For a probe measurement to be considered valid and used in scoring, a quorum of at least $k/3$ witnesses must return valid, signed receipts.

The consensus protocol executes by having nodes calculate a "suitability score" for each other based on metrics like uptime, past performance, latency, and throughput. These scores are used to formulate a QUBO problem, which is then solved by a quantum annealer to find the optimal solution—the selection of a single representative node. The protocol is designed to satisfy key consensus properties. Termination is achieved as the quantum annealer is expected to return a solution in each round. Agreement is ensured because the objective function, combined with a deterministic tie-breaking mechanism, guarantees that all honest nodes will converge on the same winner. Finally, Validity is met because the selected node is the one that maximizes the suitability score, a verifiable and objective metric reflecting desirable network characteristics.

B. Threat Model

We have identified the following primary threats to the proposed system:

- An attacker could create a massive number of nodes. Even if each node has a low individual suitability score, the large quantity increases the attacker's chances of having one of their nodes selected as the representative.
- A malicious node, after being selected, could choose not to broadcast the block to stall the network. The system must be able to detect this failure and impose a penalty that makes such an attack irrational.
- A group of nodes could collude to manipulate the suitability scores of non-colluding nodes, for instance, by acting as dishonest witnesses during the latency and uptime measurements.
- The system relies on a Probe Protocol to measure uptime and latency. An attacker could potentially find ways to exploit this protocol, such as by forging or replaying probe-related messages to falsify performance metrics.

In addition to the threat models mentioned above, we assume the Quantum Annealer is a trusted entity. We make this assumption because Quantum Annealers are not yet widespread, and our system model relies on one to solve the QUBO problem for representative selection.

A participant with controlled and stable infrastructure could deploy high-performance nodes with massive computational power, low-latency connections, and perfect uptime. This would ensure their S_i' is always the maximum, giving them de facto control over the network. Such a scenario becomes plausible when economic incentives are introduced. An economic analysis of Bitcoin and Ethereum shows that a core principle holds: a greater pool of available resources, whether higher computational power in Bitcoin mining or a larger stake in Ethereum, incentivizes participation and intensifies competition among those seeking rewards.

C. Design Goals

Our system aims to achieve following design goals:

- **Stable and Fair Node Selection for Proposing Block:** The system is designed to select a single "Representative Node" to create the next block. This selection is not random but is based on finding the optimal node that maximizes a "suitability score". This score is calculated from various measurable and weighted characteristics like uptime, latency, throughput, and past performance. To ensure fairness and prevent ambiguity, the system incorporates a deterministic tie-breaking mechanism by perturbing the scores slightly

V. Proposed Scheme

This section discusses the design of our proposed scheme, detailing the protocols and algorithms used by the network's multiple nodes and the quantum annealer, an essential element of our system.

One of the quantum annealer's primary roles is to select a representative node. A representative node is a participant chosen from a pool of eligible candidates, all seeking to earn rewards by creating a block. The quantum annealer performs this selection by finding the minimum energy state of a given objective function, which is often represented as a QUBO problem.

The objective function is constructed using multiple variables. A crucial variable is the Suitability Score (S_i), which is calculated from several scoring metrics. These metrics combine various measurable characteristics of a node, with each characteristic weighted according to its importance for the role.

In the following subsections, we will first discuss these scoring metrics in detail and then use them to construct the complete objective function.

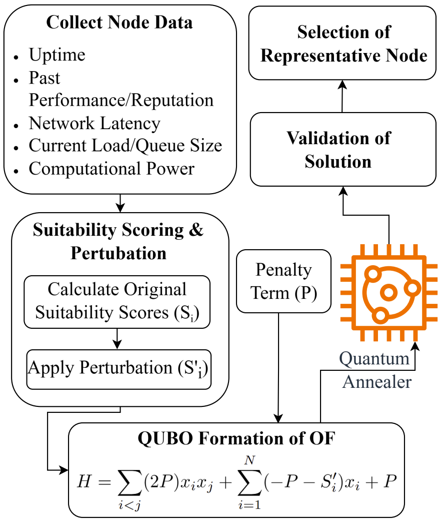
Collect Node Data
Uptime
Past Performance/Reputation
Network Latency
Current Load/Queue Size
Computational Power
Selection of Representative Node
Validation of Solution
Suitability Scoring & Pertubation
Calculate Original Suitability Scores (S_i)
Apply Perturbation (S'_i)
Penalty Term (P)
Quantum Annealer
QUBO Formation of OF
$H = \sum_{i<j}(2P)x_ix_j + \sum_{i=1}^{N}(-P - S'_i)x_i + P$

Fig. 1. Representative Node Selection Process

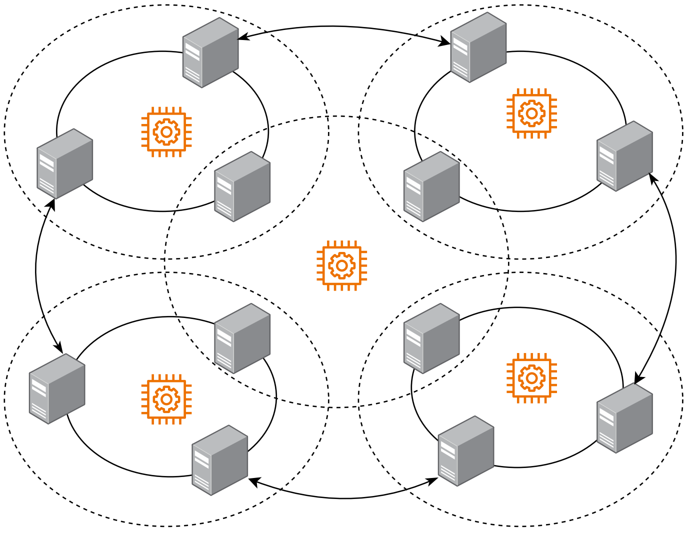

Fig. 2. System Model

A. Scoring Metrics

We propose a unified framework to efficiently calculate scoring metrics for nodes in a decentralized network. The scoring metrics required for calculating the Suitability Score are a node's Uptime, Latency, Transaction Throughput, and Past Performance.

The core of our framework is a multi-purpose **Probe Protocol** that generates a cryptographic proof. This proof simultaneously provides data for Uptime, Latency, and Throughput calculations. In contrast, Past Performance remains a distinct metric derived from historical on-chain data.

Before detailing the scoring metrics, we will first discuss digital signature used for signing and verification to ensure integrity. We will then introduce the Probe Protocol, as it provides the foundation for calculating a node's uptime, latency, and throughput.

Signing and Verification Scheme

To ensure message authenticity and integrity, the protocol uses the **Falcon digital signature algorithm**.

- **Key Generation**: Each node n_x generates a public-private key pair (pk_x, sk_x) using the key generation algorithm. Public keys are distributed and known to all nodes in the network.
- **Signing**: To sign a message M, a node uses its private key sk_x to generate a signature, $\sigma = \mathrm{S}(sk_x, M)$.
- **Verification**: To verify a signature σ on a message M from node n_x, any node can use the sender's public key pk_x to run the verification algorithm, $\mathcal{V}(pk_x, M, \sigma)$. This function returns 'true' if the signature is valid and 'false' otherwise.

Probe Protocol and Proof Generation: A source node, n_s, initiates the process by sending a `ProbeRequest`, signed by its private key to all nodes from the network for the specific ProbeRequest, at time t_{send}. The request contains a unique nonce, r.

The source node assembles these responses into a single, comprehensive **Probe Proof**, $\mathcal{PP}$, which contains:

1) **ProbeRequest (P_{req}):** The initial packet from the source, containing the message $(id_s, id_t, t_{\text{send}}, r)$ and the signature σ_s, generated using sk_s.
2) **TargetReceipt ($P_{receipt}$):** The target's response, which includes the original P_{req}, its receipt time t_{receipt}, and its signature σ_t, generated using sk_t.
3) **WitnessReceipts ($\{P_{w_i}\}$):** A set of signed reports from each witness, including the original P_{req}, their individual receipt times $t_{\text{witness\_receipt\_i}}$, and their unique signatures σ_{w_i}, generated with their respective private keys.

1) Uptime: Uptime verification is derived from the active participation of nodes in the network's Probe Protocol. This approach asserts that a node's ability to respond to a network probe is a direct and sufficient proof of its liveness.

Before calculating uptime or latency, a node must first verify that the number of Witness Receipts $\{P_{w_i}\}$ is greater than or equal to $N/3$. If this quorum is not met, the probe is discarded.

Implicit Liveness Verification. A node's uptime is confirmed whenever it successfully participates in a Probe instance.

- A **target node** (n_t) proves its liveness by issuing a valid and timestamped `TargetReceipt`.
- A **witness node** (n_w) proves its liveness by issuing a valid and timestamped `WitnessReceipt`.

Let $T_{\text{last\_seen}}(n_x)$ be the latest timestamp from any valid `TargetReceipt` or `WitnessReceipt` generated by node n_x and gossiped across the network. A querier can determine this time by querying a quorum of its peers.

Actual Uptime Calculation. The instantaneous uptime state, $S(n_x, t)$, for any node n_x at time t is determined by comparing the current time to its last verified liveness timestamp, allowing for a maximum delay tolerance, Δ. The maximum delay tolerance is based upon factors like average network latency, as it should be large enough to account for normal network delays and prevent active nodes from being mistakenly flagged as down. A smaller Δ allows the network to detect offline nodes more quickly.

$$S(n_x, t) = \begin{cases} 1 & \text{if } (t - T_{\text{last\_seen}}(n_x)) \leq \Delta \\ 0 & \text{if } (t - T_{\text{last\_seen}}(n_x)) > \Delta \end{cases} \tag{5}$$

The total uptime, $\mathcal{U}(n_x)$, over a period is the integral of this state function, accumulating all verified online periods.

$$\mathcal{U}(n_x) = \int_{t_{\text{start}}}^{t_{\text{end}}} S(n_x, t)\, dt \tag{6}$$

This method creates a robust and tamper-evident uptime record for all participating nodes from a single data source.

2) Network Latency: Network latency is measured using a unified **Probe Protocol** designed to create a single, verifiable proof that also feeds the Uptime and Throughput metrics. This process involves a source, a target, and a set of witnesses to ensure the integrity of the measurement.

Latency Calculation from Proof Any node can validate the `ProbeProof` by verifying all signatures and timestamp consistency. From a valid proof and the source's final return time, t_{return}, latency is calculated.

- **Total Round Trip Time (RTT):** The total measured time from the source's perspective.

$$\text{RTT} = t_{\text{return}} - t_{\text{send}} \tag{7}$$

$$L_{s \to t} = \begin{cases} \text{RTT} & \text{if } \mathcal{V}(\mathcal{PP}) = \text{true} \\ \text{Invalid} & \text{otherwise} \end{cases} \tag{8}$$

3) Response Throughput: Response Throughput measures a node's capacity to handle requests, calculated directly from the stream of **Probe Proofs**. This synergy means no additional processes are required to measure throughput.

Throughput Calculation Each node in the network listens for and logs all validly gossiped `ProbeProof` structures, $\mathcal{PP}$. To calculate the throughput for a target node, n_t, over a time window $[t_{\text{start}}, t_{\text{end}}]$, a node simply counts the proofs where n_t was the target. First, we define the set of relevant proofs, $\mathcal{P}_{n_t}$, from a local log:

$$\mathcal{P}_{n_t}(t_{\text{start}}, t_{\text{end}}) = \{\mathcal{PP} \mid \mathcal{PP}.id_t = n_t \wedge \mathcal{PP}.t_{\text{t\_receipt}} \in [t_{\text{start}}, t_{\text{end}}]\} \tag{9}$$

The throughput for node n_t is the number of valid proofs it generated as a target, divided by the duration of the time window.

$$\text{Throughput}(n_t) = \frac{|\mathcal{P}_{n_t}(t_{\text{start}}, t_{\text{end}})|}{t_{\text{end}} - t_{\text{start}}} \tag{10}$$

This metric provides a clear, verifiable measure of a node's responsiveness in responses per second, derived efficiently from the data already being generated for latency and uptime verification.

4) Past Performance: Past Performance remains a crucial metric based on a node's historical, on-chain contributions. Its calculation is distinct from the real-time probe metrics, as it relies on the immutable public ledger, requiring no additional network messaging.

On-Chain Event Tallying The primary measure of past performance is the successful proposal of blocks and failed proposals as well. Let the blockchain be an ordered set of blocks, $\mathcal{B} = (B_1, B_2, \dots, B_m)$. We use an accessor function, $proposerB_h$, to identify the creator of any given block B_h. The Proposal Success Count ($C_p(n_x)$) for a node n_x over a block range $[i, j]$ is the total number of blocks it has successfully proposed in that range.

$$C_p(n_x, i, j) = |\{B_h \in \mathcal{B} \mid h \in [i, j] \wedge proposerB_h = n_x\}| \tag{11}$$

Performance Score Calculation The performance score is calculated based on a node's history of both successful and failed proposals. Let $C_p(n_x)$ be the Proposal Success Count and $C_f(n_x)$ be the Proposal Failure Count for a node n_x.The performance score is this count multiplied by a protocol-defined weight constant, w_p.

$$PastPerf(n_x, i, j) = w_p \cdot C_p(n_x, i, j) - w_f \cdot C_f(n_x, i, j) \tag{12}$$

Since the calculation relies solely on public blockchain data, any node can independently verify the score for any other node, ensuring perfect transparency and fairness without impacting network performance.

B. Defining Objective Function

Now, we move on to defining our objective function which we later convert into a QUBO problem that a quantum annealer can solve. Let x_i be a binary variable for each node i. $x_i = 1$ if node i is chosen to create the block, and $x_i = 0$ otherwise. The objective is to select exactly one node that maximizes an effective suitability score that includes a tie-breaking mechanism.

The objective function we're trying to minimize is:

$$H = P\left(\sum_{i=1}^{N} x_i - 1\right)^2 - \sum_{i=1}^{N} S'_i x_i \tag{13}$$

Where:

- N is the total number of nodes.
- x_i is a binary variable: $x_i = 1$ if node i is chosen, and $x_i = 0$ if it's not.
- P is a large positive number (a penalty coefficient).
- S'_i is the **effective suitability score** of node i. This score is derived from an original suitability score S_i and a small perturbation δ_i designed to break ties deterministically. A higher S'_i means node i is more desirable.

The quantum annealer tries to find the combination of x_i values that makes H as small as possible. The objective function for selecting a single, optimal representative node is composed of two primary terms: a constraint penalty term and a score optimization term using these effective scores.

The constant penalty term is the first part of Equation 13 which enforces the selection of exactly one node. Any other configuration (e.g., zero or multiple nodes selected) incurs a significant penalty proportional to P, effectively guiding the minimization process towards valid solutions.

The second component, score optimization term, of the Equation 13 aims to select the node with the highest S'_i. This process begins with calculating an **original suitability score** (S_i) for each node i. This score is a weighted sum of its desirable attributes and subtraction of undesirable ones:

The score (S_i) is calculated as a weighted sum of normalized metrics:

$$\begin{aligned} S_i = (w_{uptime} \cdot \text{norm}(\mathcal{U}_i)) + (w_{PastPerf} \cdot \text{norm}(PastPerf_i)) \\ + (w_{throughput} \cdot \text{norm}(Throughput_i)) \\ - (w_{latency} \cdot \text{norm}(L_{s \to t})) \end{aligned} \tag{14}$$

In this formulation, various factors contribute to a node's score, with their relative importance signified by different weights ($w_{\text{uptime}}, w_{\text{perf}}, \dots$). Desirable attributes such as $\mathcal{U}_i$, $PastPerf_i$, and Throughput$_i$ are all positively weighted ($w_{\text{uptime}}, w_{\text{perf}}, w_{\text{throughput}}$) to increase the score. Conversely, since lower values are preferable for $L_{s \to t}$, it is negatively weighted and subtracted.

The normalization function defined as norm in the Equation 14 is done by using Min-Max scaling method. For the metrics where higher values are better such as $\mathcal{U}_i$, $PastPerf_i$), and $Throughput_i$, the standard Min-Max formula would be applied. The standard Min-Max formula is given as,

$$norm(X) = \frac{\text{value} - \min(\text{X})}{\max(\text{X}) - \min(\text{X})}$$

It transforms each node's raw score into a value between 0 and 1. By doing this, the node with the highest performance in each of these categories receives a normalized score approaching 1, while the lowest performer receives a score closer to 0, creating a consistent scale for comparison.

Conversely, for Network Latency ($L_{s \to t}$), where lower values are preferable, an inverted version of the Min-Max formula is required to properly align it with the other scores. This inverted formula,

$$norm(X) = \frac{\max(\text{X}) - \text{value}}{\max(\text{X}) - \min(\text{X})}$$

ensures that the node with the lowest latency receives a normalized score closer to 1. Once every metric is normalized to this common $[0, 1]$ range, the weighted sum in Equation 14 can be calculated to produce a final, comprehensive suitability score (S_i) for each node.

The Problem of Ambiguity with Tied Scores: When exactly one node j is chosen, such that $x_j = 1$ and $x_k = 0$ for $k \neq j$, the constraint penalty term becomes zero. The objective function then simplifies to:

$$H_j = -S_j$$

Consequently, if two or more nodes possess an identical highest original score (e.g., $S_A = S_B = S_{\text{max}}$), selecting any of these tied nodes would yield the same objective function value ($H_A = -S_A = -S_{\text{max}}$, $H_B = -S_B = -S_{\text{max}}$). For instance, if $S_A = S_B = 50$, both choices result in $H = -50$. The annealer would see these configurations as equally optimal ground states and might select any one of them. This ambiguity can be problematic if a specific or consistent tie-breaking rule is required for deterministic selection.

Our Solution-Slight Score Perturbation: To resolve this, the original scores S_i are slightly altered using a perturbation value δ_i to create new, effective scores S'_i. The primary calculation for this is:

$$S'_i = S_i + \delta_i \tag{15}$$

The perturbation δ_i for each node i must be small enough not to alter the ranking if scores were already different, yet unique or ordered enough to definitively break ties.

A key aspect is the choice of δ_i, which often incorporates a non-deterministic tie-breaking criterion. A common fair method involves using a random function that generates a random value for δ_i. For each consensus round, a shared random value is extracted from the data of the previously validated block. To prevent the prior block's proposer from manipulating the next selection outcome, this random value must be generated using a Verifiable Random Function.

The perturbation for each node is then calculated by combining this shared value with the node's unique pk_i. We can then use following formula:

$$\delta = Hash(VRF_{output} + pk_i) \cdot \epsilon \tag{16}$$

where H is a hash function and ϵ be a very small positive number (e.g., 10^{-5}). This method leverages the blockchain's immutable history to provide a common random input, while the unique public key ensures that each node receives a distinct perturbation value.

Effective Scores in the QUBO: These perturbed scores S'_i are then used in the optimization term of the QUBO, as initially defined for H. Since the overall objective function H is being minimized, this term encourages the selection of a node k (where $x_k = 1$) that maximizes its effective score S'_k, thereby making $-\sum S'_i x_i$ as small (i.e., most negative) as possible. This ensures a unique selection even in the presence of ties in the original S_i values.

By combining the constraint penalty term with the score optimization term using perturbed scores S'_i, the objective function effectively directs the quantum annealer to find a solution where exactly one node is chosen, and that node is the one with the highest effective suitability score, with ties in original scores being broken deterministically.

C. Derivation of QUBO Coefficients from the Objective Function

The objective function for selecting a representative node, incorporating a penalty for constraint violation and a term for maximizing suitability scores, is given by equation 13.

This function is to be mapped to the standard QUBO model:

$$H_{\text{QUBO}} = \sum_{i=1}^{N} Q_{ii} x_i + \sum_{i<j} Q_{ij} x_i x_j + C \tag{17}$$

where Q_{ii} are the linear coefficients (biases), Q_{ij} are the quadratic coefficients (couplings for $i < j$), and C is a constant offset.

The derivation proceeds as follows:

Step 1: Expansion of the Penalty Term The penalty component of Equation (13) is expanded:

$$P\left(\sum_{i=1}^{N} x_i - 1\right)^2 = P\left[\left(\sum_{i=1}^{N} x_i\right)^2 - 2\left(\sum_{i=1}^{N} x_i\right) + 1\right]$$

Step 2: Expansion of the Squared Summation Term The term $\left(\sum_{i=1}^{N} x_i\right)^2$ is expanded. Given that x_i are binary variables, $x_i^2 = x_i$.

$$\left(\sum_{i=1}^{N} x_i\right)^2 = \sum_{i=1}^{N} x_i^2 + \sum_{i\neq j} x_i x_j = \sum_{i=1}^{N} x_i + \sum_{i\neq j} x_i x_j$$

The term $\sum_{i\neq j} x_i x_j$ accounts for all pairs (i, j) where $i \neq j$. This can be rewritten to align with the standard QUBO summation convention $\sum_{i<j}$ as $2\sum_{i<j} x_i x_j$. Thus,

$$\left(\sum_{i=1}^{N} x_i\right)^2 = \sum_{i=1}^{N} x_i + 2\sum_{i<j} x_i x_j$$

Step 3: Substitution into the Penalty Term Expansion Substituting the expanded squared summation back into the penalty term, we get:

$$2P\sum_{i<j} x_i x_j - P\sum_{i=1}^{N} x_i + P$$

Step 4: Combination with the Full Objective Function Combining the expanded penalty term with the score maximization term from Equation (13):

$$H = \sum_{i<j} (2P) x_i x_j + \sum_{i=1}^{N} (-P - S'_i) x_i + P \tag{18}$$

By comparing Equation (18) with the standard QUBO form in Equation (17), the coefficients can be identified. The linear coefficient, representing the weights of individual variables, is given by $Q_{ii} = -(P + S'_i)$. The quadratic coefficients, Q_{ij} for $i < j$, which define the strength of pairwise interactions, is given by $Q_{ij} = 2P$. Finally, a constant offset $C = P$ is same as previously defined penalty coefficient. The constant offset C does not influence the determination of the optimal variable assignments $\{x_i\}$ that minimize H and is often omitted when specifying the QUBO problem to a solver.

VI. Security Analysis

In this section, we analyse and show that our proposed system model is secure against the threat model defined in sub section IV-B.

A. Sybil Attack

Theorem 1 (Sybil Attack Mitigation)**.** *The selection mechanism, which chooses the single node with the highest suitability score, renders a Sybil attack ineffective, as the quantity of low-score nodes is irrelevant to the outcome.*

Proof. The quantum annealer minimizes the objective function provided by equation 13. With the constraint that exactly one node is chosen ($\sum x_i = 1$), the penalty term is zero, and the function simplifies to minimizing $H = -\sum S'_i x_i$. This is equivalent to selecting the node j that maximizes the suitability score S'_j.

Let an attacker create k Sybil nodes with a set of low scores $\{S'_{s_1}, S'_{s_2}, ..., S'_{s_k}\}$. Let the set of legitimate nodes have scores $\{S'_{l_1}, S'_{l_2}, ..., S'_{l_M}\}$. An attacker wins if and only if:

$$\max(\{S'_{s_1}, ..., S'_{s_k}\}) > \max(\{S'_{l_1}, ..., S'_{l_M}\})$$

Increasing the number of Sybil nodes, k, does not increase the value of the attacker's maximum score. The selection depends on the **single highest score** in the network, not the quantity of nodes. Thus, the attack is ineffective. □

B. Block Withholding Mitigation

Theorem 2 (Block Withholding Mitigation)**.** *A timeout detection mechanism combined with an on-chain penalty system makes a block withholding attack an irrational strategy for any node wishing to maintain influence in future consensus rounds.*

Proof. Let a malicious node, n_a, be selected as the representative. Node n_a attempts to attack the network by not broadcasting its block.

1) **Detection**: The network operates on a predefined block proposal timeout, $t_{timeout}$. When n_a fails to produce a block within this time, all nodes detect the failure.
2) **Punishment**: This failure triggers a penalty that is recorded on-chain and directly impacts the attacker's $PastPerf_i$ score. We introduce a "Proposal Failure Count" (C_f) into the scoring metric. The performance score is now defined as:

   $$PastPerf(n_x) = (w_p \cdot C_p(n_x)) - (w_f \cdot C_f(n_x))$$

   Where C_p is the success count, C_f is the failure count, and the penalty weight w_f is significantly larger than the reward weight w_p (i.e., $w_f \gg w_p$).

Upon detection, the failure count for node n_a is incremented: $C_f(n_a) \leftarrow C_f(n_a) + 1$.

This single failure event causes a substantial negative impact on $Score(n_a)$, which in turn drastically lowers its overall suitability score, S_a. Since the consensus mechanism selects the node with the highest suitability score S'_j, the new, much lower score of n_a makes its probability of being selected in future rounds very low.

Because the attack results in a verifiable and significant loss of future influence and rewards, it is an irrational strategy. □

C. Collusion Attack Mitigation

Theorem 3 (Collusion Mitigation)**.** *Random witness selection combined with a quorum requirement makes collusion statistically ineffective, as attackers cannot reliably gain majority control over a given probe's witness set.*

Proof. Let the network have N nodes, with an attacker controlling A colluding nodes, where the attacker's fraction of the network is significantly less than half (i.e., $A/N \ll 1/2$). For each probe, a set of k witnesses is chosen at random from the N nodes.

For a probe's result to be considered valid, at least $k/3$ of the selected witnesses must respond and submit their signed receipts.

To successfully manipulate the median timestamp of a probe, an attacker must control a majority of the responding witnesses. In the best-case scenario for the attacker (where only the minimum required quorum of $k/3$ witnesses respond), they must control at least $\lceil (k/3)/2 \rceil$ of the witnesses in that responding set.

The probability of an attacker having a sufficient number of their A nodes chosen in the random sample of k witnesses is described by a hypergeometric distribution. The probability of achieving the majority control condition is exceptionally low, especially as the witness set size k and the total node count N increase.

Because an attacker cannot reliably ensure their nodes will be in a position of power for any given probe, the attack is not a viable or effective strategy for manipulating suitability scores. □

Theorem 4 (Probe Protocol Integrity)**.** *The Probe Protocol is secure against forging and replay attacks. Forging is prevented by the requirement of digital signatures on all messages, and replay attacks are prevented by the inclusion of a unique, single-use nonce and a timestamp in each ProbeRequest.*

Proof. The proof is divided into two parts, addressing each threat individually.

Security Against Forging

A valid PP consists of a P_{req}, $P_{receipt}$, and P_{w_i}. The protocol specifies that each of these components must be digitally signed.

Let $S(sk, P_{req})$ be a message P_{req} signed with a secret key sk. For an attacker to forge a P_{req} from a legitimate node n_s, they must be able to compute $S(sk, P'_{req})$ for some malicious request P'_{req}. Under the standard cryptographic assumption that the digital signature scheme is unforgeable, it's computationally infeasible to produce a valid signature without possessing the secret key sk_s.

The same principle applies to forging receipts from target or witness nodes. Since an attacker does not have the secret keys of honest nodes, they cannot forge any part of the ProbeProof, and any node can reject a forged proof by verifying the signatures.

Security Against Replay Attacks

A replay attack involves an attacker capturing a valid PP, and resubmitting it at a later time. The protocol mitigates this using two components within the P_{req}: a unique nonce (r) and a timestamp (t_{send}).

Let's assume an attacker captures a valid PP generated with nonce r at time t_{send}. The attacker attempts to replay it at a later time, $t_{replay} > t_{send}$. When an honest node receives this replayed proof, it performs two checks:

Nonce Check: Each node maintains a set of recently processed nonces, R_{seen}. The node checks if $r \in R_{seen}$. Since the original proof was already processed, the nonce r will be in this set, causing the replayed proof to be rejected.

Timestamp Check: The node checks if the proof is stale. It rejects the proof if the current time minus the send time exceeds a reasonable network delay threshold, Δ_t. The condition is: if $(t_{current} - t_{send}) > \Delta_t$, the proof is invalid.

A replayed proof will fail at least one of these checks, making the attack ineffective. The combination of mandatory digital signatures and unique, time-stamped nonces ensures the integrity and authenticity of the Probe Protocol.

□

VII. Implementation and Results

A. Simulation Specifics

The network simulation of the blockchain nodes and communication was deployed using NS-3.44. The program was deployed and interacted with Python 3.10 on WSL, using Ubuntu as the distribution. Cppyy 3.5.0 was the library to implement the Python bindings to interact with NS-3.44. This entire deployment was on a computer with Intel i5-1135G7 and 8Gb of RAM, running Windows 11.

B. Performance Analysis

VIII. Conclusion

In this paper, we propose a novel blockchain consensus mechanism that utilizes quantum annealing to overcome the efficiency and security limitations of classical protocols. The system selects a single representative node to propose a new block by formulating and solving a Quadratic Unconstrained Binary Optimization (QUBO) problem. This selection is based on a comprehensive suitability score for each node, which is calculated from weighted performance metrics such as uptime, network latency, throughput, and on-chain historical performance. To ensure deterministic and fair selection, the protocol incorporates a slight, verifiable perturbation to the scores to break any ties. We also provide a security analysis demonstrating that this quantum-assisted approach is theoretically robust against common threats, including Sybil, block withholding, and collusion attacks, thereby presenting a scalable, energy-efficient, and secure consensus alternative.

References

[1] Q. Wei, B. Li, W. Chang, Z. Jia, Z. Shen, and Z. Shao, "A survey of blockchain data management systems," *ACM Trans. Embed. Comput. Syst.*, vol. 21, no. 3, May 2022. [Online]. Available: https://doi.org/10.1145/3502741

[2] C. Zhang, C. Wu, and X. Wang, “Overview of blockchain consensus mechanism,” in *Proceedings of the 2020 2nd International Conference on Big Data Engineering*, ser. BDE ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 7–12. [Online]. Available: https://doi.org/10.1145/3404512.3404522

[3] S. Nakamoto, “Bitcoin whitepaper,” *URL: https://bitcoin. org/bitcoin. pdf-(: 17.07. 2019)*, vol. 9, p. 15, 2008.

[4] B. Sriman, S. Ganesh Kumar, and P. Shamili, “Blockchain technology: Consensus protocol proof of work and proof of stake,” in *Intelligent Computing and Applications*, S. S. Dash, S. Das, and B. K. Panigrahi, Eds. Singapore: Springer Singapore, 2021, pp. 395–406.

[5] S. Naz and S. U.-J. Lee, “Why the new consensus mechanism is needed in blockchain technology?” in *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*, 2020, pp. 92–99.

[6] J. Gomes, S. Khan, and D. Svetinovic, “Fortifying the blockchain: A systematic review and classification of post-quantum consensus solutions for enhanced security and resilience,” *IEEE Access*, vol. 11, pp. 74 088–74 100, 2023.

[7] Y. Yuan and Y. Wang, “Poa consensus mechanism based on verifiable random functions and bls,” in *Proceedings of the International Conference on Algorithms, Software Engineering, and Network Security*, ser. ASENS ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 450–454. [Online]. Available: https://doi.org/10.1145/3677182.3677262

[8] Z. Wang, J. Li, A. Liu *et al.*, “RQPoA: A random quantum PoA Consensus Mechanism in Blockchain Based on Quantum Methods,” *Research Square*, February 2024, pREPRINT (Version 1). [Online]. Available: https://doi.org/10.21203/rs.3.rs-3942478/v1

[9] J. Lin, H. Li, H. Xing, R. Huang, W. Huang, S. Deng, Y. Zhang, W. Zeng, P. Lu, X. Wang, T. Sun, and X. Tang, “Q-PnV: A Quantum Consensus Mechanism for Security Consortium Blockchains,” *arXiv preprint arXiv:2412.06325v1*, December 2024, submitted on 9 Dec 2024 (v1). [Online]. Available: https://arxiv.org/abs/2412.06325v1

[10] Q. Li, J. Wu, J. Quan, J. Shi, and S. Zhang, “Efficient quantum blockchain with a consensus mechanism qdpos,” *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 3264–3276, 2022.

[11] S. Paul, M. M. Mazumdar, and S. Chakraborty, “Quantum resistance blockchain algorithm using qhf and novel consensus mechanism,” in *2025 IEEE International Students’ Conference on Electrical, Electronics and Computer Science (SCEECS)*, 2025, pp. 1–6.

[12] X. Wen, Y. Chen, W. Zhang, Z. L. Jiang, and J. Fang, “Blockchain consensus mechanism based on quantum teleportation,” *Mathematics*, vol. 10, no. 14, 2022. [Online]. Available: https://www.mdpi.com/2227-7390/10/14/2385

[13] X.-J. Wen, Y.-Z. Chen, X.-C. Fan, W. Zhang, Z.-Z. Yi, and J.-B. Fang, “Blockchain consensus mechanism based on quantum zero-knowledge proof,” *Optics & Laser Technology*, vol. 147, p. 107693, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030399221007817

[14] H. Wang and J. Yu, “A Blockchain Consensus Protocol Based on Quantum Attack Algorithm,” *Computational Intelligence and Neuroscience*, vol. 2022, p. 1431967, Aug 2022.

[15] X. Gao, J. Xu, and J. Fan, “A novel quantum byzantine consensus protocol based on malicious node prevention mechanism,” in *2022 International Conference on Blockchain Technology and Information Security (ICBCTIS)*, 2022, pp. 202–205.

[16] S. N. Paing, J. W. Setiawan, M. A. Ullah, F. Zaman, T. Q. Duong, O. A. Dobre, and H. Shin, “Counterfactual quantum byzantine consensus for human-centric metaverse,” *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 4, pp. 905–918, 2024.

[17] R. Damaševičius and R. Maskeliūnas, “Leveraging entangled quantum states to develop consensus mechanisms in blockchain networks for smart forestry applications,” in *2024 6th International Conference on Computing and Informatics (ICCI)*, 2024, pp. 356–360.

[18] A. B and B. Surendiran, “Qhealth: A blockchain based smart healthcare consensus method,” in *2024 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT)*, 2024, pp. 1–7.

[19] M. H. Amin, J. Raymond, D. Kinn, F. Hamze, K. Hamer, J. Pasvolsky, W. Bernoudy, A. D. King, and S. Kortas, “Blockchain with proof of quantum work,” *arXiv preprint arXiv:2503.14462v2*, May 2025, version 2, last revised 16 May 2025. Submitted on 18 Mar 2025 (v1). [Online]. Available: https://arxiv.org/abs/2503.14462v2

[20] B. Lashkari and P. Musilek, “A comprehensive review of blockchain consensus mechanisms,” *IEEE Access*, vol. 9, pp. 43 620–43 652, 2021.

[21] Y. Lu, A. Sigov, L. Ratkin, L. A. Ivanov, and M. Zuo, “Quantum computing and industrial information integration: A review,” *Journal of Industrial Information Integration*, vol. 35, p. 100511, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2452414X23000845