IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 December 2023, accepted 2 January 2024, date of publication 9 January 2024, date of current version 22 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3351830

RESEARCH ARTICLE

Q-RTOP: Quantum-Secure Random Transaction Ordering Protocol for Mitigating Maximal Extractable Value Attacks in Blockchains With a Priority Gas-Fee Policy

NDAY KABULO SINAI[1] **AND HOH PETER IN**[2]
[1]Department of Computer Science and Engineering, Korea University, Seongbuk-gu, Seoul 02841, South Korea
[2]DAO Solution Inc., Gangnam-gu, Seoul 06247, South Korea

Corresponding author: Hoh Peter In (hoh_in@korea.ac.kr)

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) and funded by the Korean government under the Ministry of Science and ICT (MSIT) (No.2021-0-00177), and Technology Incubator Program for Startup (TIPS) Program (S3306708) funded by the Ministry of Small and Medium Enterprises and Startups (MSS, Korea). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the institutes.

ABSTRACT Public blockchains, such as Ethereum, rely on decentralized networks of peer-to-peer nodes known as validators or miners to verify all transactions and create new valid blocks. These validators can prioritize transactions, primarily based on high gas fees, allowing miners to maximize their block rewards, a concept referred to as maximal extractable value (MEV). However, MEV is vulnerable to front-running, back-running, and sandwich attacks (FBSAs), and is exploited by malicious nodes and bots to manipulate users' valuable transactions. These malicious activities adversely impact the Blockchain's scalability, transparency, and security. Flashbots, as one of the solutions, introduces centralization since all nodes have to forward all blocks to the central node. To address these issues, we have designed a new Blockchain transaction ordering protocol called Quantum Random Transaction Ordering Protocol (Q-RTOP). The proposed protocol operates on top of the existing Blockchain transaction ordering mechanism. However, instead of allowing validators to select transactions based on high gas fees, decentralized nodes running Q-RTOP securely randomize all transactions and then forward them to the validators, which proceed with the block validation without any change. Our protocol primarily focuses on randomizing transactions before being processed by the validators by utilizing a quantum random generator as a secure source of randomness. The final results demonstrated that Q-RTOP effectively secured user transactions and randomized 8192 transactions within 25 milliseconds.

INDEX TERMS Back-running, front-running, maximal extractable value (MEV), quantum computing, quantum-resistance algorithm, quantum random number, sandwich attack.

I. INTRODUCTION

Public blockchains store transactions on an immutable distributed ledger. The data is secured using cryptographic hashing algorithms and digital signatures and is validated by nodes through the consensus protocol [18]. The consensus protocols ensure that all copies of the data distributed across the network are identical [38]. Ethereum and its popular decentralized finance (DeFi) applications rely on validators (nodes) as intermediaries to verify transactions and update the ledger.

Blockchain transaction ordering protocol empowers intermediaries to select transactions to be included in the ledger and determine their order. Therefore, malicious nodes and smart contract-based bots engage in activities that weaken the security of the Blockchain ordering protocol. These

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

malicious activities, mainly known as front-running, back-running, and sandwich attacks (FBSAs), have been targeting users' transactions in the DeFi ecosystem, and millions of dollars have been lost [1].

Front-running is a form of exploitation wherein a malicious actor, with prior knowledge of impending transactions, exploits this information to manipulate the sequence in which transactions are added to the Blockchain [39]. By strategically placing their transactions ahead of others, the attacker gains an unfair advantage, often at the expense of other users participating in the decentralized system. This strategic reordering allows the malevolent actors to capitalize on market movements or other transaction outcomes.

On the other hand, back-running involves leveraging foreknowledge of future transactions to gain an advantageous position in the Blockchain [5]. By preemptively executing transactions based on anticipated future activities, the malicious actor aims to profit from the subsequent market changes or alterations in the Blockchain state caused by the predicted transactions.

Sandwich attacks are a sophisticated manipulation tactic where a malicious actor positions their transactions strategically between two legitimate transactions [40]. By doing so, the attacker influences the market conditions, typically intending to affect the price of an asset. This orchestrated interference allows the malicious actor to exploit price differentials or manipulate market conditions to their financial advantage.

Through these attacks, more than 192.5 million dollars have been spent by back-runners and bots to spam valuable transactions [2], leading to gas price surges and network load. These attacks are made possible because bots can manipulate the Blockchain transaction ordering protocol by adding a bigger tip (a fee added on top of the regular gas fee) to their transactions so that they are executed first, expecting to gain more than what they spent.

Front-running is a well-known strategy traders employ to profit from other user orders through high-frequency trading [3]. Similarly, for DeFi, the order in which transactions are processed plays a crucial role in revenue extraction activities. Typically, miners prioritize transactions based on transaction fees, enabling DeFi traders to engage in FBSAs of pending transactions by offering competitive transaction fees. Only miners possess the exclusive authority to determine the transaction order, granting them a monopoly on extracting value from the Blockchain. This concept is referred to as the maximal extractable value (MEV), which was further generalized as the Blockchain extractable value (BEV) by Zhou et al. [4]. For 32 months, the BEV extracted from Ethereum amounted to a staggering 540.54 million USD, originating from sandwich attacks, liquidations, and arbitrage [5]. The emergence of services, such as front-running as a service (e.g. Flashbots), mitigates the risks associated with extracting the BEV by colluding with miners within a private network, affecting Blockchain decentralization [6]. Similar to bribes [7] which poses a significant threat to the security of the Blockchain consensus, as it incentivizes miners to engage in Blockchain forking. In contrast, the proof-of-reputation consensus scheme generates random numbers that can be compromised by quantum computers [8]. We propose a new transaction ordering protocol to address these problems based on the random selection mechanism and quantum random number generator (QRNG). The proposed solution acts as an intermediary between the memory pool and validators; therefore, validators can order transactions already randomized, and once the block is created, gas fees are normally paid to motivate nodes to process transactions and protect Blockchain networks from being tampered with.

In summary, this paper addresses critical security issues and introduces a novel transaction ordering protocol. Our primary contributions are described as follows:

- **In-Depth Analysis of current Ethereum transaction ordering protocol:**

We provide an in-depth technical analysis of the existing Ethereum transaction ordering protocol, highlighting the drawbacks of priority-based gas-fee transaction ordering and its implications on Blockchain security and stability.

- **Transaction Randomization:**

We propose a new approach that randomizes all pending transactions in the memory pools before being processed by the validators. This new approach mitigates front-running, back-running, and sandwich attacks. It enhances Blockchain consensus stability and combats illegal behaviors in decentralized exchanges.

- **Quantum-Secure Protocol:**

We utilized a quantum random number generator (QRNG) as a secure source of randomness to feed our algorithm. Our proposed protocol protects the Blockchain against current FBSAs and future quantum attacks, ensuring the long-term security of Blockchain networks.

- **Efficiency and Scalability:**

Our solution is proven to be efficient and scalable. In our real-world evaluation, it took only a fraction of a second to randomize a substantial number of transactions (8192/25ms), making it a practical and efficient alternative to existing methods. By providing a new quantum-secure transaction ordering protocol, our work aims to enhance the security and stability of Blockchain transaction ordering protocols, making them more resilient to MEV-based attacks and emerging threats.

In the subsequent sections, we discuss the closely related works (Section II), delve into the background details (Section III), demystify our proposed solution (Section IV), conduct a comprehensive protocol evaluation (Section V), outline the practical implications of our solution (Section VI), and conclude (Section VII). It's important to note that throughout this document, the terms nodes, validators, and miners are used interchangeably. In addition, Q-RTOP (Quantum Random Transaction Ordering Protocol) and Q-RTOS (Quantum Random Transaction Ordering Standard) are used to refer to our proposed solution.

II. RELATED WORKS

We systematized the closest related works focusing on the following attacks and terms: Front-running, back-running, sandwich attacks, and MEV.

Daian et al. [12] initially introduced the concept of maximal extractable value (MEV). MEV can entice profit-driven miners to fork the Blockchain, potentially causing instability in the consensus system. The authors also identified and discussed the priority gas auctions (PGAs), where bots compete against each other to gain priority execution for MEV extraction. This publication led to the development of Flashbots [6], an initiative to mitigate network congestion resulting from PGAs and create fairer conditions for relaying transactions that generate profits. Qin et al. [5] quantified MEVs, referred to as BEVs, by analyzing arbitrages, sandwich attacks, and liquidations. Furthermore, they argued that introducing Flashbots exacerbates network congestion and MEVs by intensifying competition. Piet et al. [13] used a method for detecting arbitrage, back-running, and front-running transactions on the Blockchain to analyze the ecosystem of MEV extraction. They focused on the effects of Flashbots and private transaction relaying. In addition, Eskandari et al. [14] provided a systematization of knowledge on these MEV-generating transactions.

Another line of research involves the detection of MEV opportunities. Zhou et al. [4] designed a real-time system capable of detecting MEVs using limited arbitrage cycle detection and solver-aided modeling, uncovering advanced attacks in DeFi. In addition, Wang et al. [15] developed a system that identifies arbitrage opportunities but is limited to those using the constant-product invariant pricing model, restricting the analysis to specific platforms, such as Uniswap v2 and Sushi Swap.

In addition to the research on front-running and MEV-related attacks, it's important to acknowledge the state-of-the-art Blockchain techniques that offer tamper-evidence and non-repudiation features with high performance. Notably, Ledger databases like LedgerDB [46] introduced in VLDB2020, and VeDB [47], as presented in SIGMOD2023, have gained prominence for their ability to provide robust security and efficiency in Blockchain applications.

Considering the efforts of these authors, our work aims to further strengthen the security and reliability of Blockchain systems. We propose a novel approach that complements state-of-the-art techniques by introducing a random transaction ordering protocol, leveraging quantum computers to produce highly secure random numbers.

III. PRELIMINARIES

This section outlines the prerequisites covering some crucial Blockchain and quantum computing notions. We provide motivating attack examples.

A. TRANSACTION LIFECYCLE

Blockchain stores user transactions on a decentralized database. Every transaction is signed using the user's private key and is emitted to the network of peer-to-peer nodes called validators [18]. The validators verify and execute all transactions. Upon receiving a new transaction, the transaction is verified and broadcast across the entire peer-to-peer network so that each node adds it to its memory pool (waiting room containing all pending transactions). The transactions in the memory pool are ordered and gathered to form a block. For most Blockchain ordering protocols, these transactions are prioritized based on the gas fee (a mandatory fee attached to each transaction); the higher the gas fee, the faster the transaction gets included in a block. A block is then validated through a consensus algorithm and added to an immutable decentralized ledger containing all valid blocks, as illustrated in Fig. 1.

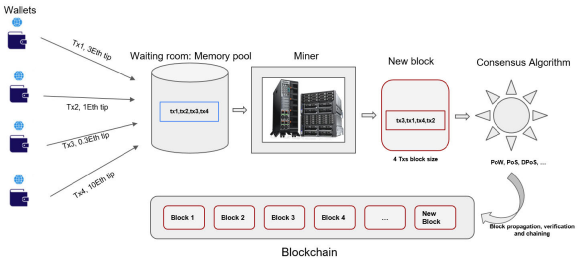

FIGURE 1. Blockchain Transaction lifecycle.

B. BLOCKCHAIN TRANSACTION ORDERING PROTOCOL

The pending transactions are ordered as follows:

1) BASED ON HIGHER GAS FEES

After the Ethereum London fork [9], a miner must build the next block by selecting transactions based on the tip (an extra gas fee the user offers). The transactions with a higher gas fee should be given priority. Assuming the block capacity is three, only fewer than three transactions can be included. The first transaction to enter the new block should be transaction 6 in Fig. 2. Transaction 4 is the next one to enter the block, with the second-highest gas fee. Then, transaction 8 is last. The remaining transactions stall inside the memory pool, causing high traffic for incoming transactions, resulting in a gas price spike that affects Blockchain scalability.

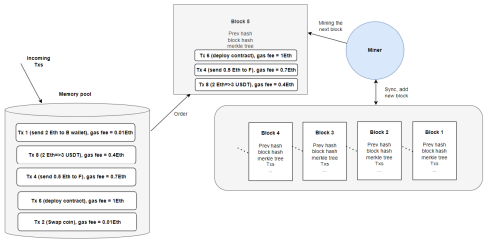

FIGURE 2. Blockchain Transaction ordering protocol: Based on high gas fees.

2) BASED ON MINER'S DECISION

Miners can extract value by ordering the transactions within the block only when they are profitable. In this case, transactions are not ordered based on gas fees but on the profit opportunities they generate. As depicted in Fig. 3,

a miner can insert its transactions before and after the largest transaction; therefore, the miner can earn a profit in addition to the tip. This profit comes at the expense of other market participants, and the miner's transactions delay other legitimate transactions [10], crippling the Blockchain decentralization nature and fairness.

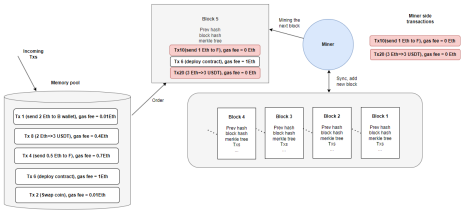

FIGURE 3. Blockchain Transaction ordering protocol: Based on miner's decision.

Ethereum 2.0 still uses the same algorithm for transaction ordering. Validators order transactions, create blocks, and propose them to a committee that votes on the new proposed block, reducing the MEV strategy because a malicious validator may be slashed and lose some of the stake (32 ETH) [32]. However, some attackers can run back-running or front-running attacks on user funds. A miner can add a new block and is free to assemble this block in any way. The latter profits from manipulating market prices via specific ordering or censoring pending transactions. Because the Blockchain is publicly accessible, these kinds of market manipulations can be observed, even if the underlying identity of the miners or other parties is unknown.

C. QUANTUM COMPUTING

Quantum computing is a branch of computer technology that leverages the laws of quantum mechanics to solve problems too complex for classical computers. A quantum computer is simply a faster computer that can solve complex problems faster than a classical computer. A quantum computer can break most cryptographic algorithms, such as RSA, ECDSA, and the cryptographically secure random number generator, generated by a classical computer [11]. Using quantum computers to build sophisticated solutions is ideal for foreseeing current and future attacks. For instance, a quantum computer can generate random numbers that no one can predict, even the quantum computer itself.

1) QUANTUM BITS AND SUPERPOSITION

Quantum bits (qubits) are the fundamental units in quantum computing. While a classical computer uses bits to express its state (0 or 1), a qubit can simultaneously be in both states, known as superposition. Measurement is needed to define the state of a qubit in superposition. The pure 0 quantum state is denoted as $|0\rangle$, and the pure 1 state as $|1\rangle$. A qubit can be represented as a vector in the vector space spanned by the basis vectors $|0\rangle$ and $|1\rangle$ [45].

2) HADAMARD GATE

Putting a qubit in a superposition state is difficult, as it requires complicated physics to move a specific particle into a superposition state. This is where the Hadamard gate applies. The Hadamard gate is a specific operator that can be applied to a qubit to put it into a superposition state. Initially, the qubit is at the top of the sphere, representing state $|0\rangle$. When applying the Hadamard gate, the qubit moves to the middle of the sphere, creating a superposition state with equal chances of being measured as 0 or 1 [41]. The Hadamard gate matrix is a 2×2 matrix. Each element in the matrix represents a coefficient that determines the transformation applied to the qubit. The Hadamard gate is defined as follows:

$$H = 1/\sqrt{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (1)$$

where the

$$1/\sqrt{2}$$

factor is a normalization factor that ensures the resulting state remains properly normalized. The element in the top left corner, 1, represents the coefficient for the state $|0\rangle$. When the Hadamard gate is applied to $|0\rangle$, it maps it to

$$(1/\sqrt{2}) * (|0> + |1>),$$

which is a superposition state. The element in the top right corner, 1, also represents the coefficient for the state $|1\rangle$. Similarly, when the Hadamard gate is applied to $|1\rangle$, it maps it to

$$(1/\sqrt{2}) * (|0> - |1>),$$

which is another superposition state. The element in the bottom left corner, 1, represents the coefficient for the state $|0\rangle$ when it is measured after applying the Hadamard gate. The element in the bottom right corner, -1, represents the coefficient for the state $|1\rangle$ when it is measured after applying the Hadamard gate. The Hadamard gate allows for exploring multiple possibilities simultaneously. It takes a qubit and spreads it out, opening up new avenues for calculations and algorithms in quantum computing.

IV. PROPOSED SOLUTION

This section presents Q-RTOP, the solution to solve FBSAs by randomizing transactions using QRNG as a source of unpredictable random numbers for current and future quantum attacks.

A. SOLUTION METHODOLOGY

We propose a quantum random transaction ordering protocol (Q-RTOP) that uses a Quantum Random Number Generator (QRNG) as a secure source of randomness to provide a safer way to generate a secure random number each time it is called. It is faster, safer, and can resist future quantum attacks. The proposed protocol is deployed in the memory pool where all transactions are stored, as presented in Fig. 4. Moreover, Q-RTOP is nondeterministic, meaning it can produce different outputs for the same input each time the algorithm is executed. In addition, Q-RTOP dynamically randomizes all transactions as they come, before being

included in the block. However, one of the challenges in a distributed environment is determining a better mechanism to prove to all peers that transactions are truly randomized. In this paper, we assume that nodes pick transactions that Q-RTOP already randomizes, and then include them in the block without any changes.

The first subsection explains how the Q-RTOP algorithm works, and the second subsection describes how the QRNG is implemented using the IBM quantum cloud environment and the *Qiskit* software development kit (SDK).

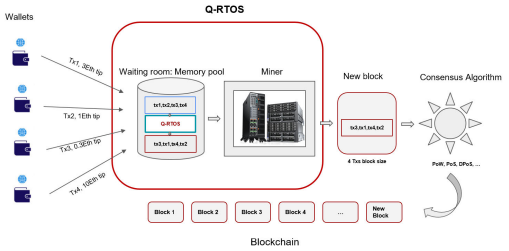

FIGURE 4. High-level architecture illustration of the proposed protocol.

B. Q-RTOP ALGORITHM

In a decentralized manner, the Q-RTOP algorithm works as follows:

- **Step 1**

Depending on the block size, gather transactions and create an array.

- **Step 2**

Index the array containing transactions from 0 to n-1, where n is the number of transactions.

- **Step 3**

Iterate through the list from the last element *(n-1)* to the first element (0).

- **Step 4**

At each iteration, generate a random index *i* (using a quantum random number generator) between 0 and the current index *j (First round j = n-1, second round j = n-2, third round j = n-3, etc.)*

- **Step 5**

Swap the element at index *i* with the element at index *j*.

- **Step 6**

Decrement the value of *j* by 1 and repeat the process until *j* reaches 0.

Fig. 5 reveals all the steps to randomize five transactions in the memory pool. Suppose that the block can contain at most five transactions. In the first round, Q-RTOP picks five transactions indexed from 0 to 4. The last transaction, Transaction 5, is selected, and a random number is generated from a quantum source. The generated number represents the new index of Transaction 5. In this example, the element at Index 2 is swapped with the last index element. The process continues until Index 0.

In the following section, we design and simulate the quantum random generator using the IBM quantum composer.

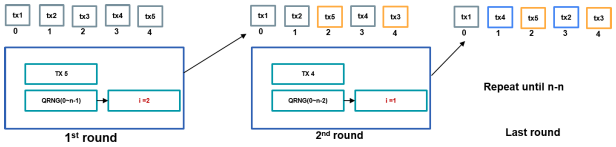

FIGURE 5. Step-by-step randomization process.

In the next subsection, we design and simulate the quantum random generator using the IBM quantum composer.

C. QRNG: DESIGN AND IMPLEMENTATION

Q-RTOP iterates the array of transactions and generates a random index *i* between 0 and the current index *j* using a QRNG. We designed and implemented a simple but effective random number generator in this section using the Hadamard gate. The Hadamard gate does not generate a random number directly on a quantum computer. However, when combined with measurements, it plays a crucial role in generating randomness. In quantum computing, the Hadamard gate creates superposition, where a qubit can be in a state of both 0 and 1 simultaneously [41]. When applying the Hadamard gate to a qubit in state $|0\rangle$, it transforms it into a superposition state represented by

$$(1/\sqrt{2}) * (|0> + |1>).$$

Similarly, applying the Hadamard gate to a qubit in the state $|1\rangle$ results in

$$(1/\sqrt{2}) * (|0> - |1>).$$

We can extract random numbers from a quantum system by employing superposition and performing measurements. For example, if applying the Hadamard gate to multiple qubits and measuring the qubits, the resulting measurement outcomes are random, following a probability distribution. Each measurement outcome corresponds to a possible random number.

1) IBM QUANTUM COMPOSER

We used the IBM Quantum Composer to design and simulate the QRNG. The *IBM* Quantum Composer provides the *Qiskit SDK* [43] that can be used to simulate quantum algorithms and run them on a real quantum computer.

A) Available Operations The IBM Quantum Composer offers different operations that can be used on qubits. In this paper, we use only the Hadamard and measurement operations circled in red in Fig. 6.

B) Quantum Random Number Generation

We manipulated the states of a certain number of qubits to generate a quantum random number and stored the results in classical registers for easier interpretation by classical computers. The number of qubits depends on the complexity of the algorithm. More qubits can potentially increase the computational power and enable the representation of larger and more complex quantum states. It may allow for exploring a more extensive solution space or the execution

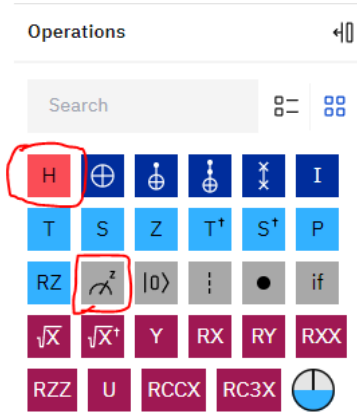

FIGURE 6. Hadamard and measurement operations.

of more sophisticated quantum algorithms. However, it also introduces challenges, such as increased susceptibility to noise and errors.

Moreover, IBM offers up to 133 real qubits to be used on their quantum computer and 5000 qubits for simulation. The simulation runs faster compared to using real qubits [44]. Thus, this paper keeps five qubits for the real-world scenario and thirteen for the simulation. After measuring the states of all qubits, the results must be stored on classical registers. The number of registers depends on the range. For instance, to generate a random number between 0 and 31 (inclusive), 5 classical registers are needed (0 $2^5 - 1$) because there are 32 possible values (0 to 31), represented with five bits. Each qubit can contribute one bit of information, and by combining the measurement outcomes of five qubits and five classical registers, we can obtain a five-bit binary number that represents a random value between 0 and 31.

To generate the random number, we can apply the Hadamard gate to each of the five qubits to create superposition, measure the qubits, and store the results in a five-bit classical register, as presented in Fig. 7, where q represents the qubit, H denotes the Hadamard gate, and a scale symbol represents the measurement. The resulting measurement outcomes can be combined to obtain a random number from 0 to 31.

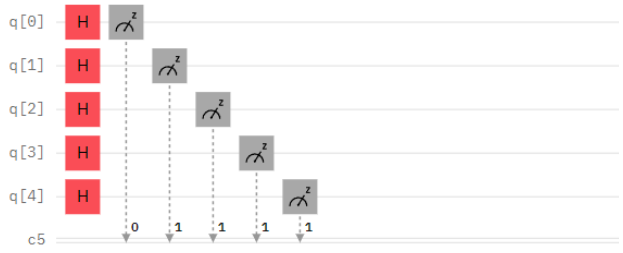

FIGURE 7. Quantum circuit: Measuring 5 qubits in superposition states.

V. QRNG: EVALUATION RESULTS

This section presents the evaluation results of the proposed solution.

A. ENVIRONMENT SETUP

Fig. 8 represents the graphical Blockchain environment setup, where miners run on classical computers. The Q-RTOP uses the IBM Qiskit SDK. The Qiskit SDK allows the construction of quantum circuits using quantum operations and simulating and executing algorithms on the IBM quantum computer. The constructed circuits are dispatched to the IBM quantum computer as jobs, with the results subsequently retrieved from the quantum computer. The IBM quantum computer is often solicited, and a single job may take 12 hours to be executed. Therefore, we used the IBM quantum simulator that provides up to 5000 qubits. Different nodes are deployed to generate quantum random numbers to avoid a single point of failure.

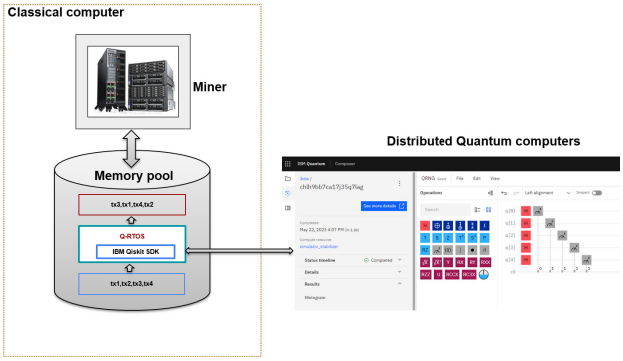

FIGURE 8. Graphical representation of our Q-RTOP environment.

B. Q-RTOP FRAMEWORK DEMYSTIFICATION

This section elucidates the integration of various components, explores their synergy, and outlines the benefits of the proposed framework.

1) INTEGRATION

In the proposed framework, the integration of technologies is pivotal for the successful implementation of the Quantum Random Transaction Ordering Protocol (Q-RTOP). The integration involves the collaborative functioning of two key components: Q-RTOP itself and the Quantum Random Number Generator (QRNG) created using the IBM Quantum Cloud environment and the Qiskit software development kit (SDK). QRNG creates highly secure random numbers that are used by Q-RTOP to randomize the transactions.

- Memory Pool Deployment:

Q-RTOP is deployed within the memory pool, the repository where all transactions are temporarily stored before being added to a block.

- Dynamic Randomization:

Q-RTOP dynamically randomizes transactions as they enter the memory pool, ensuring that each transaction undergoes a secure randomization process before inclusion in a block.

- Non-Deterministic Operation:

Q-RTOP is designed to be non-deterministic, meaning it can produce different outputs for the same input in each

execution. This property enhances the security of the protocol by introducing an additional layer of unpredictability.

- Quantum Random Number Generator (QRNG):

The QRNG, developed using the IBM quantum cloud environment and Qiskit SDK, serves as a secure source of randomness for Q-RTOP. It generates true random numbers, contributing to the unpredictability crucial for securing transactions against potential quantum attacks.

The choice of Q-RTOP and the IBM quantum cloud environment with Qiskit SDK is guided by several key considerations:

2) SECURITY ENHANCEMENT

Q-RTOP aims to enhance the security of transaction ordering by introducing a quantum-randomized approach. This approach is particularly relevant in the context of potential future quantum attacks. The use of a QRNG ensures a robust and secure source of randomness, essential for cryptographic operations.

- Speed and Safety:

Q-RTOP is designed to be faster and safer compared to traditional methods. The QRNG contributes to the efficiency of the protocol by providing a swift and secure means of generating random numbers.

- Quantum-Resistant Design:

The rationale behind employing quantum technologies is to future-proof the protocol against potential quantum threats. By leveraging quantum randomness, Q-RTOP takes a proactive stance in ensuring resilience against advancements in quantum computing.

3) SYNERGY

The synergy between Q-RTOP and the QRNG is instrumental in achieving the desired outcomes of enhanced security and quantum resistance:

- Quantum-Resistant Architecture:

The use of the IBM quantum cloud environment and Qiskit SDK for QRNG creation aligns with the project's goal of quantum resistance. This synergy ensures that the randomness injected into the protocol remains secure against potential quantum attacks, enhancing the overall resilience of the framework.

C. BENCHMARK RESULTS

This section reveals the evaluation results of Q-RTOP, which are divided into two parts. Part one reveals the results of the QRNG using different ranges, and the second part presents the results of Q-RTOP from the miner's perspective.

1) QRNG: SIMULATION RESULTS

We crafted a quantum circuit comprising n qubits and captured the outcomes in classical registers consisting of n bits. Here, n denotes a user-defined range within which the generated random number should reside. As an illustration, when aiming for a random number between 0 and 31, it necessitates the use of 5 qubits to accommodate 32 entries ($2^5 = 32$).

As mentioned in Section IV, in Steps 3 and 4, Q-RTOP iterates through the list of transactions from the last element to the first, then generates a random index i at each iteration using a QRNG between 0 and the current index j (first round $j = n\text{-}1$, second round $j = n\text{-}2$, third round $j = n\text{-}3$, etc.). The quantum results must be stored in classical registers at each iteration. The classical registers store binary results for classical computers to interpret them. We converted binary results into decimals to obtain the random number. Table 1 lists all the numbers.

TABLE 1. Number of qubits, transactions, and classical registers needed for the simulation.

# Qubits	Number of Transactions to randomize	Number of classical registers
5	32	5
6	64	6
7	128	7
8	256	8
9	512	9
10	1024	10
11	2048	11
12	4096	12
13	8192	13

```
[74]: from qiskit import Aer, execute
      from qiskit.visualization import plot_histogram
      from ibm_quantum_widgets import CircuitComposer
      from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
      from numpy import pi

      qubits_number = 5
      classical_reg_number= 5

      qreg_q = QuantumRegister(qubits_number, 'q')
      creg_c = ClassicalRegister(classical_reg_number, 'c')
      circuit = QuantumCircuit(qreg_q, creg_c)

      for i in range (qubits_number):
          circuit.h(qreg_q[i])

      for j in range (classical_reg_number):
          circuit.measure(qreg_q[j], creg_c[j])


      editor = CircuitComposer(circuit=circuit)
      editor
```

FIGURE 9. Quantum circuit having 5 qubits and 5 classical registers.

- Quantum circuits construction

Fig. 9 presents an example of a quantum circuit constructed using five qubits and five classical registers. To simulate our proposed solution, thirteen qubits were used. However, the

number of qubits can be increased depending on the number of transactions to randomize.

- Performance Results

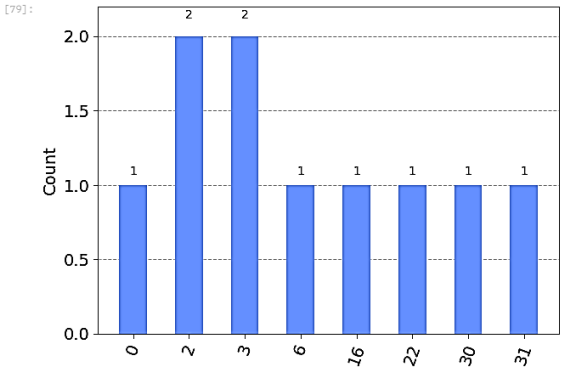

FIGURE 10. Decimal results for ten runs using five qubits and five classical registers: Y-axis represents the number of runs (count) and X-axis the output results in decimal.

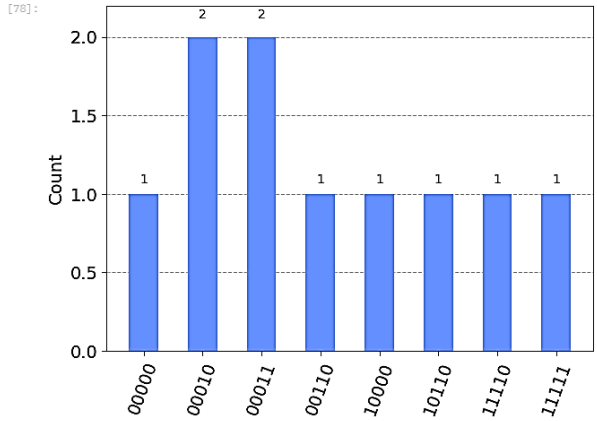

FIGURE 11. Binary results for ten runs using five qubits and five classical registers: Y-axis represents the number of runs (count) and X-axis the output results in binary.

In Figs. 10 and 11, the algorithm runs ten times, and each time a different number is produced, meaning the algorithm is stable, sufficient, and produces random numbers as expected. These numbers are fed into the Q-RTOP algorithm to randomize transactions at each iteration. Table 2 indicates that the QRNG generates quantum-secure random numbers extremely fast.

TABLE 2. Random number generation time in milliseconds.

Qubits	Classical registers	Shots	Time in milliseconds
5	5	1	0.7
6	6	1	0.93
7	7	1	0.97
9	9	1	0.99
10	10	1	1.3
11	11	1	1.46
12	12	1	1.56

2) Q-RTOP: REAL-WORLD PERFORMANCE RESULTS

To assess the real-world performance of Q-RTOP, we conducted experiments by deploying the system on ten decentralized nodes, each equipped with the specifications as outlined in Table 3. These nodes were chosen to mimic a diverse and representative network environment, ensuring robustness and decentralization. The results of these experiments, which provide valuable insights into Q-RTOP's behavior in practical scenarios, are depicted in Figure 12. This figure illustrates key performance metrics and how they evolve during the performance of Q-RTOP.

Specifically, to measure the time taken to randomize the transactions in the memory pool, we developed custom scripts tailored for this purpose. These scripts captured the transaction randomization process and allowed us to obtain precise timing information, which was then used to generate the data presented in Figure 12. Our choice to deploy the system on multiple nodes with varying specifications aimed to simulate real-world conditions and to demonstrate Q-RTOP's adaptability across different nodes having different specifications. This approach enhances the validity and applicability of our results to a wider range of practical use cases.

TABLE 3. Blockchain nodes' specifications.

Quantity	CPU core	Memory (TBi)	Bandwidth
10 VMs	64	3	19 Gbps

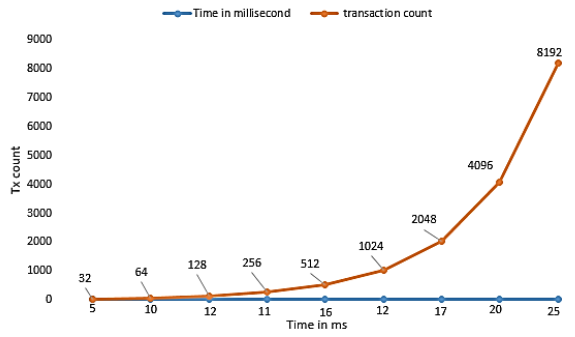

FIGURE 12. Q-RTOP: performance results.

D. RESULTS INTERPRETATION AND IMPLICATION

Fig. 12 reveals that, with 32 transactions in the memory pool, the algorithm took 5 ms to randomize them, 10 ms for 64 transactions, and 25 ms for 8192 transactions. These numbers prove that the solution is efficient and scalable for the current Ethereum, which can theoretically process 3000 transactions/sec at most.

Randomizing Blockchain transactions guarantees enhanced security against FBSAs, transaction copying, and block

forging. By introducing randomness into the transaction ordering process, the proposed protocol removes the predictability and exploitable patterns that malicious actors, including miners, can leverage. In addition, random numbers are generated from a quantum environment, making the solution secure and unpredictable.

Front-running attacks occur when an attacker observes a pending transaction and attempts to submit a transaction with higher gas fees to prioritize and execute it before the original transaction. Randomizing the transaction order makes it difficult for attackers to anticipate the position of a specific transaction, reducing the effectiveness of front-running strategies. Back-running attacks involve miners or validators manipulating the transaction order to their advantage. By randomizing the transaction order, the protocol prevents miners from selectively including or excluding transactions based on their interests, mitigating back-running attacks. However, miners will still get rewarded since the proposed solution does not remove the incentive mechanism.

Sandwich attacks exploit the predictable transaction ordering to execute trades profitably between different decentralized exchanges. Randomization disrupts the predictable patterns attackers use, making executing these profitable arbitrage strategies challenging. Transaction copying refers to duplicating transactions to increase the chances of successful execution or to spam the network. Randomizing the transaction order introduces uncertainty, making it harder for attackers to duplicate and exploit transactions effectively.

Block forging attacks involve malicious entities creating their blocks or manipulating the order of transactions within a block to gain an unfair advantage. Randomizing the transaction order makes it challenging for attackers to manipulate the block structure and ensures a fairer and unbiased selection of transactions within each block. By incorporating transaction randomization, the protocol significantly enhances the security of the Blockchain by reducing the effectiveness of various attack vectors that rely on predictable transaction ordering.

E. COMPARISON OF Q-RTOP WITH FLASHBOTS

We compared Q-RTOP with Flashbots on different aspects as shown in Table 4.

TABLE 4. Comparison of Q-RTOP with Flashbots.

Features	Q-RTOP	FLASHBOTS
Robust against FBSAs	fully secured against FBSAs	Partially
Secure transaction randomization	Yes	No
Decentralization	Yes	No
Fork encouragement	Nonexistent	Yes
Quantum-secure	Yes	No

- FBSAs:

The Q-RTOP protocol demonstrates a high level of security and robustness against various types of attacks, including front-running, back-running, and sandwich attacks. It is designed to provide full protection against these manipulative trading strategies. In contrast, FLASHBOTS doesn't fully address the issues related to front-running, back-running, and sandwich attacks. Instead, it establishes a system for equitable profit distribution generated by the MEV-Share Node, thereby introducing a degree of centralization into the process [42]

- Secure transaction randomization:

Q-RTOP implements secure transaction randomization, ensuring that transactions are conducted in a way that prevents predictability and manipulation. This adds an extra layer of security to transactions carried out using the Q-RTOP protocol. In contrast, FLASHBOTS lacks secure transaction randomization. This implies that transactions conducted through FLASHBOTS may be more susceptible to certain types of attacks that rely on predictable patterns, as randomization is not fully implemented in the protocol.

- Decentralization:

Flashbots introduces a level of centralization by requiring nodes to forward blocks to a central node. This may raise concerns about the overall decentralization of the network, as the central node becomes a critical point of coordination. The centralization in Flashbots could potentially undermine the decentralized nature of Blockchain networks, impacting principles of censorship resistance and peer-to-peer interaction.

In comparison, Q-RTOP is designed with a focus on maintaining decentralization. The protocol leverages decentralized nodes to securely randomize transactions before forwarding them to validators. Q-RTOP's emphasis on decentralization aligns with the foundational principles of Blockchain technology, fostering a more resilient and censorship-resistant network.

- Fork Encouragement:

While Flashbots may not directly address fork scenarios, the introduction of centralization might incentivize forks, particularly if disputes arise regarding the role and decisions of the central node. Fork encouragement could lead to a fragmented Blockchain network, potentially compromising consensus and transaction history. In contrast, Q-RTOP inherently discourages forks.

- Quantum-Security:

While Flashbots remains vulnerable to quantum attacks, especially with the increasing power of quantum computers capable of breaking existing cryptographic schemes, Q-RTOP explicitly emphasizes quantum security. By leveraging a quantum random generator for transaction randomization, Q-RTOP aims to enhance the protocol's resistance to potential quantum attacks on traditional cryptographic methods.

VI. DISCUSSION

A. PRACTICAL IMPLICATIONS

The practical implications of Q-RTOP are substantial, introducing a pioneering approach to fortify the security and reliability of transaction processing in Blockchain scenarios. However, it is imperative to acknowledge potential constraints associated with the current state of quantum technologies.

While Q-RTOP leverages the power of quantum randomness through a Quantum Random Number Generator (QRNG), the increased complexity and specialized nature of quantum hardware may pose challenges in its widespread implementation. Moreover, the practicality of Q-RTOP is contingent upon the accessibility and availability of quantum computing resources. The limited accessibility of quantum computers may present barriers to the widespread adoption of Q-RTOP, necessitating careful consideration of the quantum computing landscape.

B. POTENTIAL CHALLENGES

The deployment of Q-RTOP is accompanied by potential challenges inherent to the current state of quantum technologies. Quantum computers, while promising, are not universally accessible at present. This poses a significant hurdle to the practical implementation and market adoption of Q-RTOP.

Furthermore, the susceptibility of quantum systems to errors and the evolving nature of quantum technologies add layers of complexity. Quantum computers are not yet mature, and their error-prone nature necessitates careful consideration in the development and deployment of quantum-resistant solutions like Q-RTOP. These challenges underscore the need for ongoing advancements and a meticulous approach to ensure the reliability of Q-RTOP in real-world scenarios.

C. REGULATORY COMPLIANCE

Given the specific focus of Q-RTOP on the protection of user transactions and the integrity of the Blockchain, regulatory compliance may not be directly applicable to the proposed solution. Q-RTOP primarily operates within the Blockchain network, safeguarding transactional data through quantum-resistant methods.

Unlike solutions that handle sensitive personal or financial information, Q-RTOP is designed to fortify the cryptographic foundations of the Blockchain without necessitating adherence to traditional regulatory frameworks related to data privacy or financial transactions. As such, the considerations for regulatory compliance in areas like Know Your Customer (KYC) or Anti-Money Laundering (AML) may not be directly relevant to the implementation of Q-RTOP.

D. MARKET ADOPTION

The viability of market adoption for Q-RTOP is intricately tied to the accessibility and commercial availability of quantum computers. As of now, quantum computers are not widespread, limiting the practical implementation of Q-RTOP in real-world Blockchain ecosystems. Acknowledging the current limitations, strategies for market adoption are considered in anticipation of the eventual availability and accessibility of quantum computing resources.

VII. CONCLUSION

We proposed a novel Blockchain transaction ordering protocol to mitigate front-running, back-running, and sandwich attacks, called Quantum Random Transaction Ordering Protocol (Q-RTOP). Q-RTOP emphasizes the randomization of transactions before validators process them. Utilizing a quantum random generator as a secure source of randomness, Q-RTOP has proven to be highly effective in securing user transactions, randomizing 8192 transactions within 25 milliseconds.

Randomizing Blockchain transactions guarantees enhanced security against FBSAs, transaction copying, and block forging. By introducing randomness into the transaction ordering process, the proposed protocol removes the predictability and exploitable patterns that malicious actors, including miners, can leverage.

However, our solution does not consider a scenario in which validators rearrange the randomized transactions upon reception. Additionally, a challenge persists in providing proof to all peers, demonstrating that the proposed block has truly randomized transactions. It is important to note that our primary objective was to propose a faster and more secure method for randomizing transactions. In future work, we will address these issues, refining our protocol to accommodate validator actions and enhancing the mechanisms for proving the inclusion of randomized transactions to all network peers.

ACKNOWLEDGMENT

Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the institutes.

REFERENCES

[1] @phabc. *Backrunning Bots Gas Consumption*. Accessed Apr. 11, 2022. [Online]. Available: https://dune.com/phabc/backrunning-bots-gas-consumption

[2] S. Katte. *7 DeFi Protocol Hacks in Feb See 21 Million USD in Funds Stolen: DefiLlama*. Accessed Apr. 15, 2023. [Online]. Available: https://cointelegraph.com/news/7-defi-protocol-hacks-in-feb-sees-21-million-in-funds-pilfered-defillama

[3] M. Lewis, *Flash Boys: A Wall Street Revolt*. New York, NY, USA: WW Norton and Company, 2014.

[4] L. Zhou, K. Qin, A. Cully, B. Livshits, and A. Gervais, "On the just-in-time discovery of profit-generating transactions in DeFi protocols," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 919–936, doi: 10.1109/SP40001.2021.00113.

[5] L. Zhou, K. Qin, C. F. Torres, D. V. Le, and A. Gervais, "High-frequency trading on decentralized on-chain exchanges," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 428–445.

[6] A. Obadia. *Flashbots: Frontrunning the MEV Crisis*. Accessed Jun. 23, 2023. [Online]. Available: https://medium.com/flashbots/frontrunning-the-mev-crisis-40629a613752

[7] J. Bonneau, "Why buy when you can rent?" in *Proc. Int. Conf. Financ. Cryptogr. Data Secur.* Cham, Switzerland: Springer, 2016, pp. 3–4.

[8] S. Shyamsukha, P. Bhattacharya, F. Patel, S. Tanwar, R. Gupta, and E. Pricop, "PoRF: Proof-of-Reputation-based consensus scheme for fair transaction ordering," in *Proc. 13th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Pitesti, Romania, Jul. 2021, pp. 1–6, doi: 10.1109/ECAI52376.2021.9515090.

[9] V. Buterin, E. Conne, R. Dudley, M. Slipper, I. Norden, and A. Bakhta. *Ethereum Improvement Proposals*. Accessed Apr. 25, 2023. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1559
[10] Dune. (2022). *Backrunning-Bots-Gas-Consumption*. Accessed: Nov. 5, 2022. [Online]. Available: https://dune.com/phabc/backrunning-bots-gas-consumption
[11] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 15–17, Aug. 1995.
[12] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2020, pp. 910–927.
[13] J. Piet, J. Fairoze, and N. Weaver, "Extracting godl [sic] from the salt mines: Ethereum miners extracting value," 2022, *arXiv:2203.15930*.
[14] S. Eskandari, S. Moosavi, and J. Clark, "SoK: Transparent dishonesty: Front-running attacks on blockchain," in *Proc. Int. Conf. Financ. Cryptogr. Data Secur.*, 2020, pp. 170–189.
[15] Y. Wang, Y. Chen, H. Wu, L. Zhou, S. Deng, and R. Wattenhofer, "Cyclic arbitrage in decentralized exchanges," in *Proc. Companion Web Conf.*, Apr. 2022, pp. 12–19.
[16] E. Zeydan, J. Baranda, and J. Mangues-Bafalluy, "Post-quantum blockchain-based secure service orchestration in multi-cloud networks," *IEEE Access*, vol. 10, pp. 129520–129530, 2022, doi: 10.1109/ACCESS.2022.3228823.
[17] L. Gan and B. Yokubov, "A performance comparison of post-quantum algorithms in blockchain," *J. Brit. Blockchain Assoc.*, vol. 6, no. 1, pp. 1–10, May 2023.
[18] A. M. Antonopoulos and G. Wood, *Mastering Ethereum*. Sebastopol, CA, USA: O'Reilly Media, 2018.
[19] W. Wang, Y. Yu, and L. Du, "Quantum blockchain based on asymmetric quantum encryption and a stake vote consensus algorithm," *Sci. Rep.*, vol. 12, no. 1, p. 8606, May 2022.
[20] N. Sinai. *Is it Hard to Build a Blockchain From Scratch?* Accessed: Oct. 25, 2022. [Online]. Available: https://medium.com/swlh/is-it-hard-to-build-a-Blockchain-from-scratch-2662e9b873b7
[21] D. Micciancio and O. Regev, "Lattice-based cryptography," in *Post-Quantum Cryptography*, D. J. Berstein, J. Buchmann, and E. Dahmen, Eds. Berlin, Germany: Springer, 2008, doi: doi.org/10.1007/978-3-540-88702-7_5.
[22] R. Auer, J. Frost, and J. M. V. Pastor, "Miners as intermediaries: Extractable value and market manipulation in crypto and DeFi," *Bank Int. Settlements Bullet.*, vol. 58, pp. 3–6, Jun. 2022. [Online]. Available: https://econpapers.repec.org/RePEc:bis:bisblt:58
[23] I. Pedone and A. Lioy, "Quantum key distribution in Kubernetes clusters," *Future Internet*, vol. 14, no. 6, p. 160, 2022, doi: 10.3390/fi14060160.
[24] P. J. Clarke, R. J. Collins, V. Dunjko, E. Andersson, J. Jeffers, and G. S. Buller, "Experimental demonstration of quantum digital signatures using phase-encoded coherent states of light," *Nature Commun.*, vol. 3, no. 1, p. 1174, Nov. 2012, doi: 10.1038/ncomms2172.
[25] National Institute of Standards and Technology. *NIST Announces First Four Quantum-Resistant Cryptographic Algorithms*. NIST. Accessed Dec. 11, 2022. [Online]. Available: https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms
[26] T. M. Fernández-Caramès and P. Fraga-Lamas, "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks," *IEEE Access*, vol. 8, pp. 21091–21116, 2020, doi: 10.1109/ACCESS.2020.2968985.
[27] K. Buchholz. (Dec. 2021). *Quantum Leap for Quantum Computing*. Statista. Accessed: Nov. 27, 2022. [Online]. Available: https://www.statista.com/chart/26317/quantum-computing-market-value/
[28] C. Joshi, C. Bhole, and N. Vaswani, "A scrutiny review of CPS 4.0-based blockchain with quantum resistance," in *Advancements in Quantum Blockchain With Real-Time Applications*, M. Shrivas, K. Hiran, A. Bhansali, and R. Doshi, Eds. Hershey, PA, USA: IGI Global, 2022, pp. 131–157, doi: 10.4018/978-1-6684-5072-7.ch007.
[29] J. Hoffstein, "NTRU: A ring-based public key cryptosystem," in *Proc. Int. Algorithmic Number Theory Symp.*, 2011, pp. 267–288, doi: 10.1007/BFb0054868.
[30] *Dune*. Accessed: May 25, 2023. [Online]. Available: https://dune.com/phabc/backrunning-bots-gas-consumption
[31] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in *Proc. 3rd Symp. Operating Syst. Design Implement.*, New Orleans, LA, USA, 1999, pp. 2–3.
[32] V. Buterin, D. Hernandez, T. Kamphefner, K. Pham, Z. Qiao, D. Ryan, J. Sin, Y. Wang, and Y. X. Zhang, "Combining GHOST and casper," 2020, *arXiv:2003.03052*.
[33] V. Buterin and V. Griffith, "Casper the friendly finality gadget," 2017, *arXiv:1710.09437*.
[34] V. Buterin. (2014). *Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform*. Accessed: Apr. 25, 2023. [Online]. Available: https://bitcoinmagazine.com/business/ethereum-next-generation-cryptocurrency-decentralized-application-platform-1390528211
[35] Wikipedia. *Shor's Algorithm*. Wikipedia. Accessed: Feb. 15, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Shor algorithm
[36] National Institute of Standards and Technology. (Jul. 2022). *NIST Announces First Four Quantum-Resistant Cryptographic Algorithms*. Accessed Oct. 28, 2022. [Online]. Available: https://www.nist.gov/news events/news/2022/07/nist-announces-first-four-quantum-resistantcryptographic-algorithms
[37] *Hyperledger Whitepaper*, IBM, Armonk, NY, USA, 2018.
[38] H.-Y. Paik, X. Xu, H. M. N. D. Bandara, S. U. Lee, and S. K. Lo, "Analysis of data management in blockchain-based systems: From architecture to governance," *IEEE Access*, vol. 7, pp. 186091–186107, 2019.
[39] C. F. Torres and R. Camino, "Frontrunner Jones and the raiders of the dark forest: An empirical study of frontrunning on the Ethereum blockchain," in *Proc. 30th USENIX Secur. Symp.*, 2021, pp. 1343–1359.
[40] L. Heimbach and R. Wattenhofer, "Eliminating sandwich attacks with the help of game theory," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, May 2022, pp. 153–167.
[41] D. C. Marinescu, *Classical and Quantum Information*. Cambridge, MA, USA: Academic Press, 2012, pp. 63–80.
[42] *Flashbots*. Accessed Nov. 28, 2022. [Online]. Available: https://www.flashbots.net
[43] IBM. *IBM Quantum Documentation*. Accessed Nov. 30, 2022. [Online]. Available: https://docs.quantum.ibm.com/
[44] IBM. *IBM Quantum Compute resources*. Accessed Nov. 30, 2022. [Online]. Available: https://quantum.ibm.com/services/resources?tab=systems
[45] Wikipedia. *Quantum Superposition*. Accessed Nov. 30, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Quantum_superposition
[46] X. Yang, Y. Zhang, S. Wang, B. Yu, F. Li, Y. Li, and W. Yan, "LedgerDB: A centralized ledger database for universal audit and verification," *Proc. VLDB Endowment*, vol. 13, no. 12, pp. 3138–3151, 2020.
[47] X. Yang, R. Zhang, C. Yue, Y. Liu, B. C. Ooi, Q. Gao, Y. Zhang, and H. Yang, "VeDB: A software and hardware enabled trusted relational database," *Proc. ACM Manag. Data*, vol. 1, no. 2, pp. 1–27, 2023.

NDAY KABULO SINAI received the bachelor's degree in software engineering from Université Protestante de Lubumbashi and the master's degree in computer engineering from Kookmin University. He is currently pursuing the Ph.D. degree in computer science and engineering with Korea University. His research interests include blockchain, security, and quantum computing.

HOH PETER IN is currently a Professor with Korea University. He is also the Founder and the Chair of the Korea Society of Blockchain and a Korea Blockchain Association Board Member. Additionally, he is also a Co-Founder of DAO Solution, a web3-based company. His research interests include software engineering, embedded systems, software security, and blockchain.