Quantum Information Processing (2024) 23:99
https://doi.org/10.1007/s11128-024-04272-6

Performance evaluation of a quantum-resistant Blockchain: a comparative study with Secp256k1 and Schnorr

Nday Kabulo Sinai[1] · **Hoh Peter In**[1,2]

Received: 6 July 2023 / Accepted: 13 January 2024

Abstract

Popular Secp256k1 and Schnorr algorithms offer strong security in current Blockchains. However, they are vulnerable to quantum attacks. To solve this problem, several quantum-resistant algorithms have been proposed. However, the performance evaluations and tangible analyses of these algorithms on current Blockchains have not been studied yet. In this context, a performance analysis of quantum-resistant algorithms on a Blockchain can provide valuable insight into the efficiency of quantum-resistant algorithms in real-world scenarios. To address this need, we prototyped and analyzed a quantum-resistant Blockchain using the Falcon algorithm. Falcon is selected because it provides smaller signature and key size compared to Crystals-Dilithium and Sphincs+. We then measured in real-time the key size, transaction signature size, and transaction verification time. The paper also discusses the potential scalability limitations of the proposed quantum-resistant Blockchain and suggests an approach to select quantum-resistant algorithms based on different Blockchain use cases. Our approach and benchmark results have implications for the future development and adoption of quantum-resistant Blockchains.

Keywords Blockchain · Falcon · Quantum-resistant algorithm · Secp256k1 · Schnorr

Nday Kabulo Sinai and Hoh Peter In contributed equally to this work.

✉ Hoh Peter In
hoh_in@korea.ac.kr

Nday Kabulo Sinai
sinai@korea.ac.kr

[1] Department of Computer Science and Engineering, Korea University, 145 Anam-ro, Seongbuk-gu, Seoul 02841, South Korea

[2] DAO Solution, Inc., Myeongwoo Building, 169 Yeoksam-ro, Gangnam-gu, Seoul, South Korea

Published online: 06 March 2024

1 Introduction

Blockchain technology has been gaining widespread adoption in recent years. It has revolutionized industries ranging from finance and healthcare to supply chain management and digital identity verification. However, the security of Blockchain systems is not guaranteed to threats from emerging technologies like quantum computing. The reason is that quantum computers can solve challenging problems (Finding the private key by using the public key) a million times faster than most supercomputers that exist today. However, current quantum computers cannot crack traditional encryption methods yet, but there is a theoretically proven method called Shor by which most elliptic curve cryptography-based systems would be made vulnerable [1], including Bitcoin and Ethereum. Shor's algorithm can quickly find the prime factors (Secret key in the Blockchain) of any integer (Public key) using a quantum computer. This means the most used Secp256K1 and Schnorr digital signatures are vulnerable [2] since they are based on integer factorization problems to be secure. In a world where quantum computers are quickly becoming a reality [3], it is clear that we need new quantum-resistant algorithms that can protect current Blockchains against quantum attacks.

To tackle this issue, new algorithms that do not rely on the factoring of the prime numbers have been proposed. These algorithms utilize mathematical problems that are believed to be difficult to solve by classical super-computers and even quantum computers. Several quantum-resistant algorithms have been proposed, but only Falcon, Crystals-Dilithium, and Sphincs+ were selected as the safest and sophisticated quantum-resistant digital signatures [4]. However, the applicability of these algorithms raises concerns about their scalability and computational power on a real-world Blockchain. In addition, numerous research papers have been theoretically focusing on studying them and demystifying their advantages and disadvantages [5] in a broader way, and their performance metrics have been shown off-chain [6]. Moreover, there is a lack of real-world performance metrics and tangible analyses of quantum Blockchains that use recent quantum digital signatures.

In this context, a comparative analysis of quantum-resistant Blockchain can provide valuable insight into the robustness and scalability of new quantum-resistant algorithms in a real-world scenario. To address this need, we present our research on the comparative analysis of a quantum-resistant Blockchain based on one of the most recent quantum-resistant algorithms called Falcon. The Falcon algorithm was selected because it provides a high level of security against quantum attacks while requiring fewer computational resources than Sphincs+ and Crystals-Dilithium [7]. We then evaluated the performance of a real-world quantum Blockchain prototype based on Falcon as a digital signature, proof of stake, and Byzantine fault tolerance as consensus protocols. We measure the key size and generation time, signature size and signature generation time, transaction size, and verification time in a decentralized environment. We also discuss the potential scalability limitations of a quantum-resistant ledger (QRL) and propose various ideas for selecting quantum-resistant algorithms based on different scenarios. A step-by-step approach for designing a quantum-resistant Blockchain is proposed.

Our paper is broken down into the following sections: the background section, where we provide an overview of Blockchain and post-quantum cryptography; the second section is about the 5-step approach for protecting Blockchain against quantum attacks, where we itemize all the important steps to consider when adopting quantum-resistant algorithms for future quantum attacks; the implementation section provides the performance metrics of the Falcon algorithm on real-world Blockchain implementation. Additionally, we compare the Falcon performance against Secp256K1 and Schnorr algorithms; and finally, land with the conclusion and future works.

2 Preliminaries

This section provides an overview of Blockchain and quantum cryptography. We examine Blockchain security, transaction lifecycle, consensus algorithms, quantum computing, and lattice-based cryptography.

2.1 Blockchain overview

2.1.1 Definition

Blockchain is a set of existing technologies put together to create a secure, trusted, and immutable database called a digital ledger. Blockchain can also be defined as a secure, trusted decentralized database and network all in one [8]. A Blockchain contains blocks that are securely interconnected, and each block contains information such as transactions, Merkle tree, block hash, block number, and difficulty, to mention a few. The data stored on the Blockchain are immutable and secured by computers called miners/validators.

2.1.2 Transaction lifecycle

As mentioned in the previous section, each block contains transactions. Each transaction passes through different stages before being confirmed and added to a new block as shown in Fig. 1.

A transaction lifecycle starts when a user signs a transaction using their private key (The Blockchain security section explains how the key pair is generated), the transaction is then sent to the closest node (light node) to be verified, and once verified, the transaction is broadcast across the Blockchain P2P network, and each node (computer) adds that transaction to its local memory pool (A waiting room for unconfirmed transactions). The next step is collecting all transactions inside the memory pool, grouping them into a block, and then starting the mining/validating process on that block following a predefined consensus protocol. When a block is validated by a specific node, all nodes must verify it and then add it to their local Blockchain. Each Blockchain can validate new blocks based on a specified consensus algorithm (Figs. 2, 3, 4).

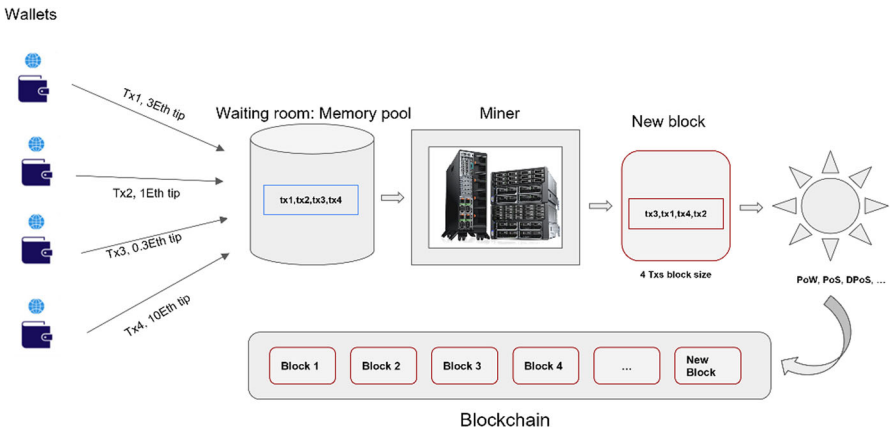

Fig. 1 Transaction lifecycle

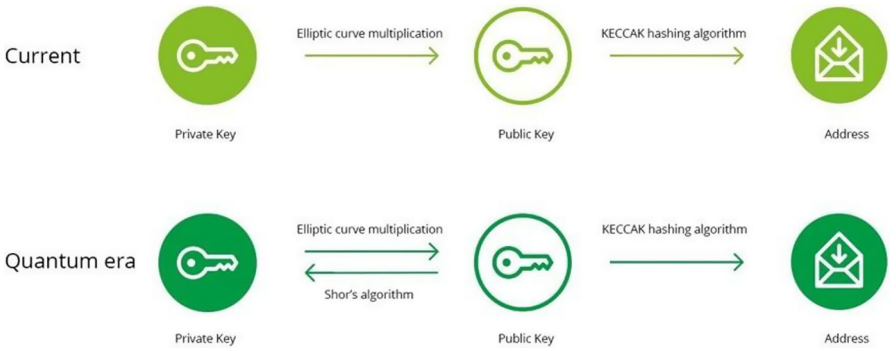

Fig. 2 Ethereum-based key-pair and wallet address generation steps

2.1.3 Consensus algorithms

Behind every Blockchain implementation there is a consensus layer, the consensus layer is the most crucial layer since it deals with validating the blocks, ensuring there is not any altered block or transaction. The consensus algorithms are mostly categorized into two groups, the consensus-based voting mechanism (Byzantine Fault Tolerance) and the Sybil control consensus-based (Proof of Work and Proof of Stake). Table 1 lists some of the consensus algorithms and describes the differences between them.

2.1.4 Blockchain security: Ethereum use-case

One of the key features of Blockchain is security. Blockchain security relies on cryptographic algorithms such as elliptic curve cryptography for digital signature, and other cryptographic algorithms and hash functions are intensively used during key pair generation, transaction signature, transaction verification, consensus verification,

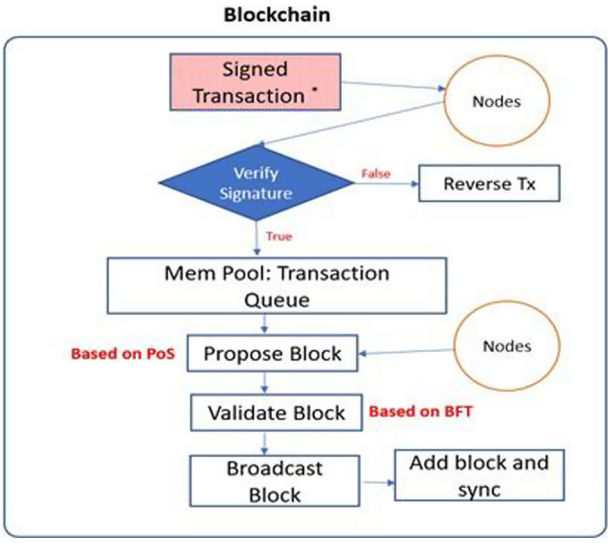

Fig. 3 Transaction verification process

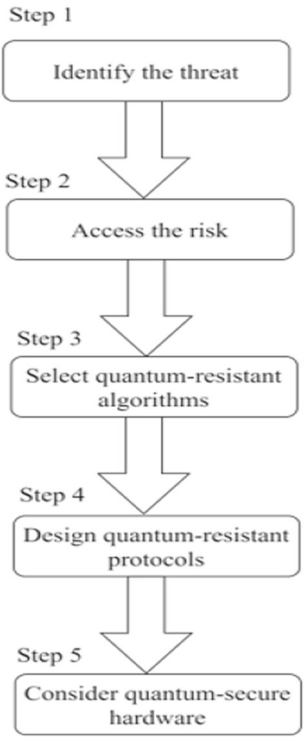

Fig. 4 A five-Step approach for designing a quantum-resistant Blockchain

Table 1 Consensus algorithms

Facts	PoW	PoS	BFT
Blockchain type	Permissionless	Permissionless and Permissioned	Permissioned
Native coin	Yes	Yes	No
Fork rule	The longest chain	Gasper: the heaviest chain	No fork
Example Usage	Bitcoin, Ethereum 1.0	Ethereum2.0, Tendermint	Hyperledger fabric, Tendermint, Ethereum
Pros	Secure	Fast and secure	Fast and secure for private Blockchains
cons	Energy consumption, slow, 51 attack	The richer gets richer, younger and less battle-tested, complex to implement	The more nodes are added to the network, the slower the response*[a]
Quantum-resistant	No	Yes	Yes
Facts	PoW	PoS	BFT
Blockchain type	Permissionless	Permissionless and Permissioned	Permissioned
Native coin	Yes	Yes	No
Fork rule	The longest chain	Gasper: the heaviest chain	No fork
Example Usage	Bitcoin, Ethereum 1.0	Ethereum2.0, Tendermint	Hyperledger fabric, Tendermint, Ethereum

[a]Ethereum2.0 uses a simplified and faster version of BFT which is a 2 step-commit stage

Merkle tree generation, block hashing, to mention a few. In this section, we will focus on Ethereum security as our case study.

- **Key pair generation process**

The key pair represents a mathematically related set of keys called the private key and public key. Ethereum Blockchain provides key pairs differently from Bitcoin. For Ethereum, the first step is to pick a nonzero number between a predefined valid range slightly less than 2**256, that number is defined as the order of the elliptic curve cryptography. The obtained random number is called a private key. The second step is to apply an Elliptic Curve called Secp256K1 (Bitcoin uses a Schnorr curve instead) on the private key to get the public key. In the last step, the public key is hashed using keccak256 (secure hash algorithm) to obtain the wallet address, and users see only the last 40 characters (40bytes) prefixed with "Zero X" (0x) [9]. Since the private key is super long, some wallet implementations show only the mnemonic or seed phrase which varies between 12 and 25 words (human-readable version of the private key), or the 64 hexadecimal digits obtained after conversion.

- **Transaction signature**

Ethereum wallet funds are stored on the Blockchain, only the account owner can unlock the coins and make a transfer when needed, the funds are reflected in the wallets as well. To unlock the coins, the user should send a transaction together with the proof (Signature) to the Blockchain. Upon transaction reception, the local node will check whether that transaction is really coming from the owner by using the owner's public key and the signature itself.

Any transaction coming from externally owned accounts (EOAs) has the following values:

- **Current nonce**

A nonce is a number that can be used only once, it's an initial number given to the miners/validators to solve the proof of work; however, this value is always equal to zero for the Ethereum beacon chain since there is no longer a mining process with proof of stake (PoS).

- **To**

The "To" represents the recipient's address.

- **Value**

The "value" field is the total amount to be sent to the recipient.

- **Gas limit**

The "Gas Limit" is the maximum amount of gas that a user is willing to pay to process their transactions.

- **Gas Price**

The "Gas Price" is the transaction processing fee (Tip); this tip is paid to the validators/miners after creating a new block.

- **Signature**

Transaction signature is the final output that is produced after signing the transaction with a private key. The final result is r,s,v, where "r" and "s" are the outputs from the ECDSA signature and "v" is the recovery id, and the tuple r,s,v can be used to obtain the "from" public key of the sender wallet.

- **Transaction verification**

Once a transaction is signed and sent to the Blockchain network, the local node processes it by checking the signature through the ECDSA mechanism. In addition, the gas fee is checked to make sure the transaction will be processed successfully before forwarding it to other peers.

If the gas fee is not enough or the signature does not belong to the owner, the transaction will be reverted, if not the transaction will be broadcast to all available nodes, and every single node must include it inside its own memory pool (waiting room). At this point, miners prioritize transactions with higher fees to be included in the block for maximizing their profits (via PGA and MEV mechanisms: Priority Gas Auction and Miner Extractable Value).

2.2 Quantum computing: post-quantum cryptography

In this section, we will look at quantum computing and the lattice-based problem called the Shortest Vector Problem (SVP), and why it is quantum-resistant.

2.2.1 General

Quantum computing is a computer technology branch 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 difficult problems that a classical computer cannot solve. When a quantum computer of sufficient size is built on the way to achieving its full potential, some of the cryptographic algorithms like the Elliptic Curve Digital Signature Algorithm (ECDSA) become vulnerable by the means of Shor's algorithm. To foresee the future disaster, many organizations have been proposing solutions against quantum attacks, and lattice-based cryptography is one of the leading candidates for a secure post-quantum defense mechanism.

2.2.2 Lattice-based cryptography

Lattice-based cryptography is a cryptographic algorithm based on mathematical problems, and it is one of the leading candidates for secure post-quantum cryptography methods. It is believed to be secure against quantum attacks because quantum computers are not well suited to solve these types of problems. Some of the well-known problems are Learning with Errors (LWE), Ring Learning With Errors (RLWE), and the Shortest Vector Problem (SVP), to mention a few. The SVP is a problem in mathematics that asks for the shortest vector in a high-dimensional lattice. It is believed to be hard for quantum computers to solve because of the way quantum computers perform calculations. Quantum computers are only able to exploit the superposition principle when the states of the quantum bits are perfectly aligned.

However, when the states of the quantum bits are not perfectly aligned, the quantum computer will not be able to exploit the superposition principle and will have to perform the calculation in a more traditional way. This means that if a quantum computer tries to solve the SVP, it will likely make a lot of errors and will not be able to find the correct answer. This is why lattice-based cryptography is quantum resistant. To keep this section short and focus more on the danger of quantum computing, the lattice-based cryptography paper [10] can be explored for more details on the mathematical demonstration of lattice problems and quantum algorithms.

3 A 5-step approach for protecting Blockchain against quantum attacks

In this section, we present a new approach to designing a quantum-resistant Blockchain. We clearly explain all the steps to be considered while switching to quantum safety. We start from identifying the quantum threat to selecting a quantum-resistant algorithm, and much more.

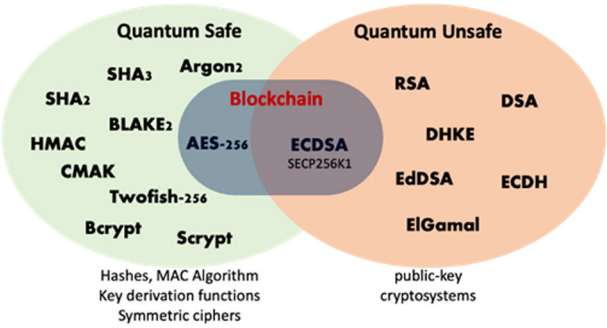

Fig. 5 Quantum-resistant and quantum-unsafe algorithms

3.1 Step 1. Identify the threat

Quantum computers pose a threat to traditional cryptographic methods because they can break many of the algorithms that are currently used to secure communication and data. This is because quantum computers can perform certain types of calculations much faster than classical computers, allowing them to quickly solve the mathematical problems that are at the heart of many cryptographic algorithms.

For example, quantum computers are capable of quickly solving the "discrete logarithm problem," which is a key component of the cryptographic algorithms used to secure communication on the internet, such as the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols. They are also capable of breaking many of the symmetric key algorithms that are commonly used to encrypt data, such as the Advanced Encryption Standard (AES). Figure 5 shows some of the quantum-resistant and non-resistant algorithms.

$$N = p * q$$

3.2 Step 2. Assess the risk

All cryptographic algorithms based on the integer factorization problem are quantum-broken. The problem involves finding the prime factors of a large composite number. It is believed to be computationally infeasible for classical computers to solve this problem for numbers with more than a few hundred digits, but quantum computers can solve it much faster. The integer factorization problem is expressed mathematically as follows: given a composite integer N, find its prime factors p and q.

$$N = p * q \tag{1}$$

The security of many cryptographic algorithms, such as the RSA and ECDSA algorithms, relies on the assumption that it is difficult to solve this problem. However, quantum computers can use algorithms such as Shor's algorithm to solve the integer

factorization problem much faster than classical computers, which means that they can break the encryption provided by these algorithms. In addition to the integer factorization problem, quantum computers are also able to solve the discrete logarithm problem which is widely used in Blockchain. This problem involves finding the value of x in the following equation:

$$g^x = h(modp) \tag{2}$$

where g and h are known values, and p is a prime number. The security of many cryptographic algorithms, such as the Diffie-Hellman key exchange algorithm, relies on the assumption that it is difficult to solve this problem. However, quantum computers can use algorithms such as the quantum discrete logarithm algorithm to solve the discrete logarithm problem much faster than classical computers, which means that they are able to break the encryption provided by these algorithms.

3.3 Step 3. Select quantum-resistant algorithms

Post-quantum algorithms have been proposed to solve current quantum-unsafe algorithms. They have been submitted to NIST for approval, and around the fifth of July 2022, NIST came up with four post-quantum algorithms that are proven to be secure against both quantum computers and traditional computers. They grouped them into two categories:

- **General encryption:** CRYSTALS-Kyber
- **Digital signature:** Crystals-Dilithium, Falcon, and Sphincs+

The selected algorithms will be incorporated into the post-quantum cryptographic standard which is being developed by NIST and is anticipated to be completed in 2024. Here are some details about these algorithms:

- **The CRYSTALS-Kyber**

The CRYSTALS-Kyber algorithm has been chosen by NIST for general encryption, which is utilized when accessing secure websites. Kyber is based on lattice cryptography and benefits include the speed of operation and very minimal encryption keys that two parties can simply exchange. The algorithm offers a variety of hybrid settings and overall excellent performance on hardware and software.

- **Crystals-Dilithium**

Crystals-Dilithium is suggested by NIST as the main algorithm. The digital signature algorithm is based on the Fiat-Shamir paradigm. Dilithium is a signature method that has a strong theoretical security foundation and offers a simple implementation with great efficiency. It is a great option for a wide range of cryptographic applications.

- **The Falcon**

The Falcon signature has the smallest bandwidth and is fast. The algorithm was chosen for its solid security and due to its low bandwidth, which may be necessary for applications that require smaller signatures.

Table 2 Comparison of quantum-resistant algorithms

Facts	Falcon	Crystals-Dilithium	Sphincs+
Purpose	Digital signatures	Digital signatures	Digital signatures
Based on	Lattice-based	Lattice-based	Hash functions
Advantages	Scalable, Simple, Fast	Efficient	Efficient
Disadvantages	Smaller signature Still at its early stage	Difficult to implement, Slow, Bigger signature	Larger keys and signatures, Slower

- **Sphincs+**

Sphincs+is slightly larger and slower than the other algorithms but is useful as a backup for one main reason it does not use lattices-based cryptography. The algorithm uses a stateless hash-based signature scheme and offers a different arithmetic methodology than the other three selected algorithms [4]. Overall, each algorithm has its strong and weak points, and depending on the one's system purpose, one of them can be selected as the best-suited algorithm. Table 2 shows the differences between all three algorithms.

3.4 Step 4. Design quantum-resistant protocols

When designing a quantum-resistant Blockchain, designing quantum-resistant protocols is an essential step. This includes designing quantum-resistant routing, node discovery, and peer-to-peer communication protocols, as well as a quantum-resistant consensus mechanism.

- **Quantum-resistant routing**

Quantum-resistant routing protocols should be implemented to ensure that data are transmitted securely and efficiently across the network, even in the face of potential quantum attacks. This may include implementing quantum-resistant routing algorithms, such as quantum-resistant link-state or distance-vector routing protocols.

- **Quantum-resistant node discovery**

Quantum-resistant node discovery protocols should be implemented to ensure that nodes can securely and efficiently discover and connect to other nodes in the network, even in the face of potential quantum attacks. This may include implementing quantum-resistant node discovery algorithms, such as quantum-resistant flooding or gossip-based protocols.

- **Quantum-resistant peer-to-peer communication**

Quantum-resistant peer-to-peer communication protocols should be implemented to ensure that nodes can securely and efficiently communicate with each other, even

in the face of potential quantum attacks. This may include implementing quantum-resistant communication algorithms, such as quantum-resistant key exchange protocols or quantum-resistant message authentication codes.

- **Quantum-resistant consensus**

Quantum-resistant consensus algorithms should be implemented to ensure that the Blockchain can achieve secure and decentralized consensus among nodes, even in the face of potential quantum attacks. This may include implementing quantum-resistant proof-of-work or proof-of-stake consensus algorithms or implementing a quantum-resistant variant of existing consensus algorithms. All these protocols should be designed to protect against any potential quantum-based attack and to withstand such attacks. Regular testing and evaluations should be made to ensure the robustness of the protocols.

3.5 Step 5. Consider quantum-resistant Hardware

Blockchain is broken down into five layers as shown in Fig. 6: Application, Consensus, network, Data, and hardware layers. These layers must implement a quantum-resistant mechanism to make the Blockchain fully quantum-resistant. So, the hardware/infrastructure must be secure against quantum attacks.

4 Implementation and results

To implement a quantum-resistant Blockchain prototype, we used our proposed approach. However, due to time limitations and a strong team, we skipped the steps

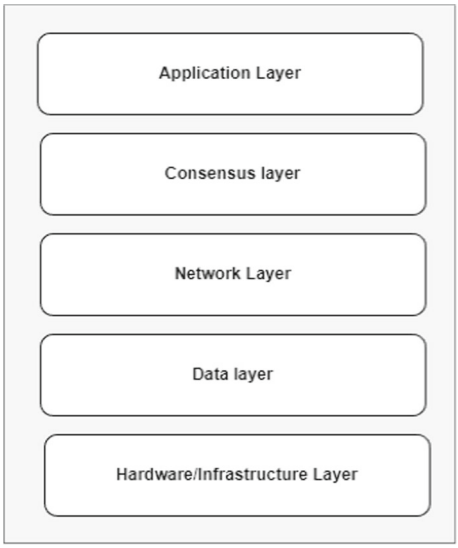

Fig. 6 Blockchain layers

of designing quantum-resistant protocols and quantum-resistant hardware selection. Our prototype was built using Falcon digital signature for key generation, transaction signature, and transaction verification. We then evaluated its performance against popular Blockchains based on the Secp256K1 and Schnorr algorithms.

In addition, the prototype uses Proof of Stake (PoS) as a Sybil attack mechanism and a Byzantine Fault Tolerance (BFT) consensus algorithm for block validation. PoS and BFT are well-known for being quantum-resistant. The Falcon algorithm was implemented in the user wallet and Blockchain nodes (or validators).

4.1 Quantum-resistant Blockchain

4.1.1 Conception

To achieve quantum security, we implemented the quantum-resistant into the user's wallet and Blockchain nodes as shown in Fig. 7. The process starts when a user joins the network. Upon joining a quantum wallet is generated using the Falcon algorithm, and the private and public keys are stored inside that wallet. The next step is emitting transactions to the Blockchain.

To simplify the development, the transaction is composed of the signature and a "hello" message only. The signed transaction is emitted to the closest Blockchain node which will verify the signature validity. Once the transaction is valid, the validator broadcasts it to other modes. Based on the Proof of Stake mechanism, a block proposal is selected, and the proposed block is broadcast across the network. All nodes verify it using the Byzantine Fault Tolerance; once confirmed, the newest block containing that transaction is added to the node's local ledger, and the process continues on and on.

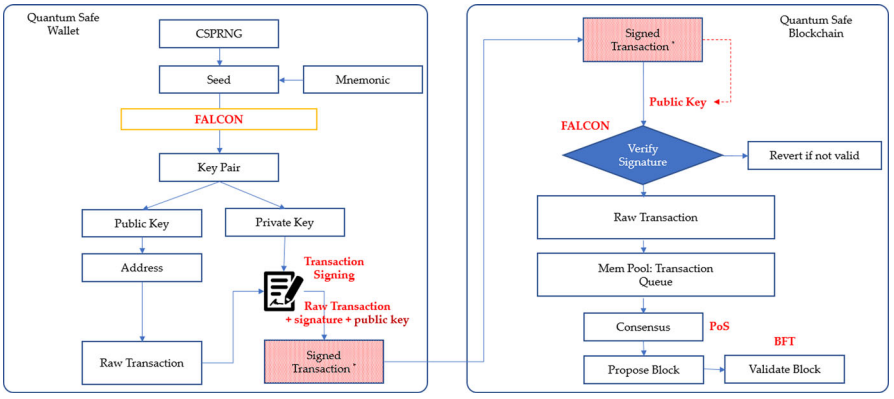

Fig. 7 Quantum-resistant Blockchain and wallet flowchart

Table 3 Testbed: nodes specifications

Name	Operating system	Memory	CPU mark and number of cores	Disk space	Bandwidth	Location
Node 1	Ubuntu 22.10 × 64	1 GB	1 Intel CPU	25 GB NVMe SSDs	1000 GB	Cape Town
Node 2	Ubuntu 22.10 × 64	2 GB	1 Intel CPU	25 GB NVMe SSDs	2 TB	California
Node 3	Ubuntu 22.10 × 64	4 GB	2 Intel CPU	80 GB NVMe SSDs	4 TB	Seoul
Node 4	Ubuntu 22.10 × 64	8 GB	4 Intel CPU	160 GB NVMe SSDs	5 TB	Paris

4.1.2 Performance analysis

We deployed four nodes in different zones across the globe, all validators have different specifications as shown in Table 3. The Blockchain nodes are responsible for verifying the transactions emitted by the client (wallet) running on a personal computer. To avoid the wallet using many resources during key generation, the client had to send a request for key generation to the closest Blockchain node.

We measure the time taken for a node to generate a quantum-resistant key pair, key size, time taken for a wallet to sign a transaction, the signature size, and the verification time. We also measured the transaction per second (TPS) by sending multiple transactions simultaneously.

- **Nodes specifications**

  See Table 3.

- **Benchmark results**

  See Figs. 8, 9, 10, 11.

4.1.3 Results interpretation

The performance of our quantum Blockchain was evaluated by measuring the key generation time, key size, signature size, and the time taken to generate the key pair, sign and verify the transactions using different key strengths. The key generation time is given in milliseconds, the key pair size and signature size are expressed in bytes, the signing and verification times in microseconds, and each key size is represented by its own color. The results show that for a 256-key-size, it took 6.79 ms to generate, 234 micro-seconds to sign a transaction, 26.9 microseconds for transaction verification,

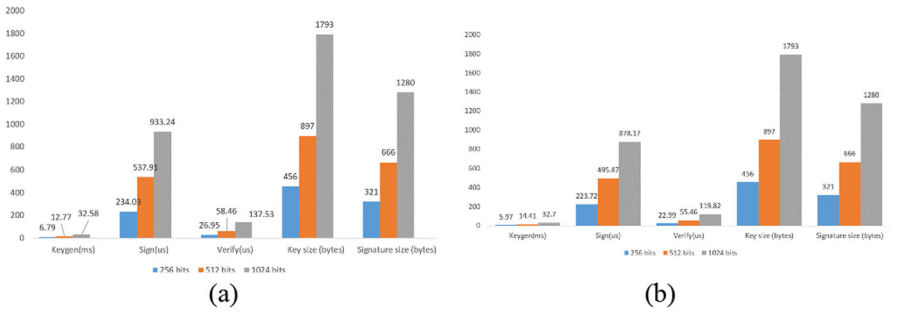

Fig. 8 a Performance metrics using 256,512 and 1024 bits key strength on the first node. **b** Performance metrics using 256,512 and 1024 bits key strength on the second node

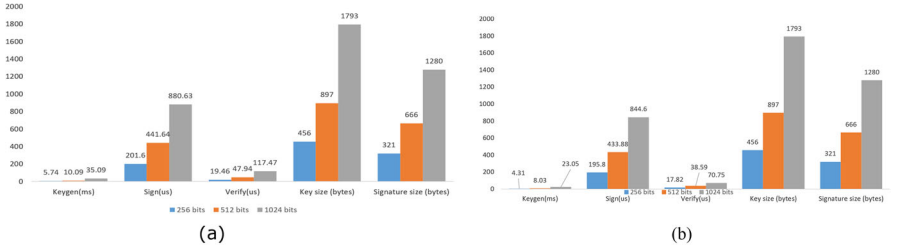

Fig. 9 a Performance metrics using 256,512 and 1024 bits key strength on the third node. **b** Performance metrics using 256,512 and 1024 bits key strength on the fourth node

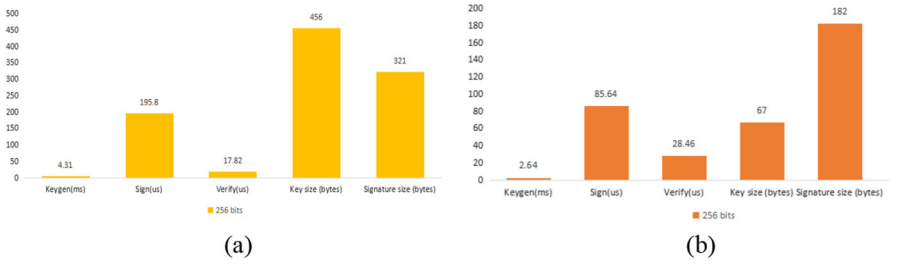

Fig. 10 a Falcon performance metrics using 256 bits key strength on the fourth node. **b** Secp256k1 Performance metrics using 256 bits key strength on the fourth node

and the key and signature sizes are 456 and 321 bytes, respectively. Also, we see that the key generation time increases as the key strength increases from 256 to 512 bits to 1024 bits. This is expected as more complex and longer keys require more computation to generate. However, the signature and key sizes remain relatively constant across all key strengths, indicating that the algorithm is efficient in terms of transaction verification. The security of the Blockchain based on the Falcon algorithm is directly proportional to the key strength used. In general, the longer and more complex the key, the more secure the algorithm is. Therefore, using a key strength of 1024 bits

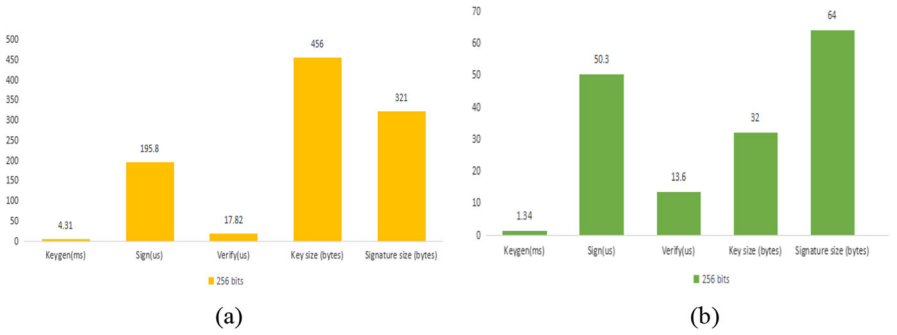

Fig. 11 **a** Falcon performance metrics using 256 bits key strength on the fourth node. **b** Schnorr performance metrics using 256 bits key strength on the fourth node

provides the highest level of security, but it also comes with a longer key generation time. On the other hand, using a key strength of 256 or 64 bits provides a lower level of security, but it also comes with a shorter key generation time. Therefore, it is essential to find a balance between security and performance depending on the Blockchain purpose, for example, a private Blockchain may use a weaker key size compared to a public one that can use a stronger key size. The results show that Blockchain based on the Falcon algorithm is slightly slower compared to Secp256k1 and Schnorr used by almost 90 percent of current Blockchains; however, our Blockchain is fairly efficient and quantum-resistant. The Falcon algorithm is designed to be highly secure and efficient, making it well-suited for Blockchain digital signature.

5 Conclusion

In this paper, we built and analyzed a quantum-resistant Blockchain using one of the recent quantum-resistant algorithms called Falcon along with Proof of Stake and Byzantine Fault Tolerance algorithms by following our newly proposed approach. We then tested its performance against popular Blockchains based on Secp256k1 and Schnorr digital signatures which are quantum vulnerable. The results show that quantum-resistant Blockchain is theoretically and mathematically secure against quantum attacks, yet slower compared to quantum-broken Blockchains. This is due to the complexity of mathematical formulas used under the hood. However, the signature and key sizes remain relatively constant across all key strengths, indicating that the algorithm is efficient. In addition, the security of a digital signature algorithm is directly proportional to the key strength used. In general, the longer and more complex the key, the more secure the algorithm is.

Therefore, using a key strength of 1024 bits provides the highest level of security, but it also comes with a longer key generation time. On the other hand, using a key strength of 256 or 64 bits provides a lower level of security, but it also comes with a shorter key generation time. Therefore, it is essential to find a balance between security and

performance depending on the Blockchain purpose; for instance, a private Blockchain can use a weaker key size compared to a public one that can use stronger keys.

Our test environment was conducted on four nodes located in different countries around the world for key pair generation, transaction verification, signature, and block validation. The better the server on which the node is running, the quicker the generation and verification are done, this proves that the obtained results may vary depending on the server specifications, and many other factors that are out of the scope of this paper.

5.1 Suggestions and future work

To solve the speed problem, one can adjust the key size to a desirable strength depending on the Blockchain application use case, however, this solution may have negative effects. We suggest that one can implement a multi-party computation mechanism that is faster and quantum-resistant, or even better consider an IOTA-like design, however, these alternatives have their own weaknesses, so it is essential to find a balance between security, decentralization, and scalability. In addition, changing the Blockchain application and/or consensus layer is not the ideal solution against quantum attacks, since Blockchain nodes are interconnected via a peer-to-peer network that utilizes TCP/IP and other protocols to exchange information, quantum-resistant algorithms should therefore be implemented at the network level, data, and hardware layers to make a robust and secure Blockchain against future quantum attacks. Thus, our future work will include a step-by-step design standard, where all Blockchain layers will be studied and securely improved for future quantum attacks. In addition, the block size, energy consumption, bandwidth, and latency will be measured.

Acknowledgements This work was supported by the Korea University grant, Institute of Information and Communications Technology Planning and Evaluation (IITP) under the Ministry of Science and ICT (MSIT) (No.2021-0-00177), and Technology Incubator Program for Startup (TIPS) Program (S3306708) under 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.

Data availability The data will be available on request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Shor, P.W.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, pp. 15–17, 30 August (1995)
2. Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Shor
3. Katharina, "Statista," Statista, 2 Dec 2021. [Online]. Available: https://www.statista.com/chart/26317/quantum-computing-market-value/. [Accessed 27 Nov 2022]
4. NIST,"NIST Announces First Four Quantum-Resistant Cryptographic Algorithms," NIST, [Online]. Available: https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic algorithms. [Accessed 12 11 2022]
5. Joshi, C.: A Scrutiny Review of CPS 4.0-Based Blockchain with Quantum Resistance. Advancements in Quantum blockchain with real-time applications (2022)
6. T. M.: Towards Post-Quantum Blockchain: A Review on IEEE Access (2020)
7. NIST website, 7 July 2022. [Online]. Available: https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms. [Accessed 28 Oct 2022]
8. Sinai, Medium website, [Online]. Available: https://medium.com/swlh/is-it-hard-to-build-a-blockchain-from-scratch-2662e9b873b7. [Accessed 25 Oct 2022]
9. A. M. A. a. D. Wood.: Mastering Ethereum, O'Reilly (2018)
10. D. M. a. O. Regev. Lattice-based Cryptography (2008)