mathematics

IMPACT FACTOR 2.4

CITESCORE 3.5

Review

A Quantum-Resistant Blockchain System: A Comparative Analysis

P. Thanalakshmi, A. Rishikhesh, Joel Marion Marceline, Gyanendra Prasad Joshi and Woong Cho

Special Issue

On the Security of the Blockchain Technology

Edited by
Dr. Badis Hammi

MDPI

https://doi.org/10.3390/math11183947

mathematics

MDPI

Review

A Quantum-Resistant Blockchain System: A Comparative Analysis

P. Thanalakshmi [1], A. Rishikhesh [1], Joel Marion Marceline [1], Gyanendra Prasad Joshi [2,*] and Woong Cho [3,*]

[1] Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore 641004, India; ptl.amcs@psgtech.ac.in (P.T.); rishiyashvanth@gmail.com (A.R.); joel.marceline3@gmail.com (J.M.M.)
[2] Department of Computer Science and Engineering, Sejong University, Seoul 05006, Republic of Korea
[3] Department of Electronics, Information and Communication Engineering, Kangwon National University, Samcheok 25913, Gangwon State, Republic of Korea
* Correspondence: joshi@sejong.ac.kr (G.P.J.); wcho@kangwon.ac.kr (W.C.)

Abstract: Blockchain transactions are decentralized, secure, and transparent, and they have altered industries. However, the emergence of quantum computing presents a severe security risk to the traditional encryption algorithms used in blockchain. Post-quantum signatures are required to preserve integrity and reliability. Furthermore, combining the InterPlanetary File System (IPFS) with blockchain provides a long-term strategy for data storage and sharing. This study investigates the integration of post-quantum signatures with the IPFS in a blockchain system, which can considerably enhance blockchain system efficiency. We increase security and efficiency by recording hash values of signatures and public keys within the blockchain and storing their actual content using the IPFS. The study compares NIST-recommended post-quantum signatures with the ECDSA in a Bitcoin exchange scheme to show how effective the system is in countering quantum threats while maintaining optimal performance. This research makes an important addition to the long-term viability and dependability of blockchain technology in the face of the growing landscape of quantum computing breakthroughs.

Keywords: blockchain; blockchain security; cryptography; quantum computing; post-quantum signatures

MSC: 11T71; 94A62

check for updates

Citation: Thanalakshmi, P.; Rishikhesh, A.; Marion Marceline, J.; Joshi, G.P.; Cho, W. A Quantum -Resistant Blockchain System: A Comparative Analysis. *Mathematics* **2023**, *11*, 3947. https://doi.org/10.3390/math11183947

Academic Editor: Badis Hammi

Received: 21 August 2023
Revised: 5 September 2023
Accepted: 14 September 2023
Published: 17 September 2023

1. Introduction

Blockchain, a decentralized digital ledger technology, has gained widespread popularity due to its inherent traits such as non-tampering, non-forgery, traceability, transparent data, and safety. By leveraging public key cryptography, blockchain enables secure and trustless information sharing among peers, effectively resolving the double-spending problem. As blockchain finds its applications in smart factories, measurement systems, logistics, and e-voting, ensuring the integrity and authenticity of transactions becomes paramount.

However, the rise of quantum computing presents a significant threat to the security of blockchain systems. Communication and trust between dispersed blockchain network nodes must depend on digital signature mechanisms, which principally permit verification of information identity, authenticity, and integrity. Quantum computing's ability to solve complex mathematical problems efficiently, such as breaking down numbers into their prime factors and solving discrete logarithmic problems, undermines the security of traditional digital signature schemes, like the RSA, ECDSA, ECDH, and DSA. The robustness of blockchain, which relies on these signature schemes for integrity and authentication, faces uncertainty in the face of quantum computing advancements. Current blockchain systems could become obsolete due to quantum attacks on cryptography algorithms, which could lead to fraudulent transactions and unauthorized data access. The capacity of quantum computers to solve complicated mathematical problems may quickly undermine blockchain's decentralized nature, raising questions about the security and integrity of distributed ledger networks. Consequently, there exists a pressing demand to explore and implement post-quantum signature schemes to ensure the sustained security and resilience of blockchain networks in the quantum era.

Comparing the ECDSA with NIST-recommended post-quantum schemes allows us to weigh the balance between current cryptographic standards and the imperative of future quantum resistance. This evaluation is rooted in established standards with global recognition. Consequently, this paper initiates with a comprehensive exploration of the present landscape of post-quantum cryptography, offering crucial insights for researchers aspiring to construct secure blockchain networks. Subsequently, the paper delves into the practical execution of the most promising post-quantum signature methods in conjunction with the ECDSA within the blockchain context. Given the substantial size of public keys and signatures in post-quantum schemes, they can consume a significant portion of a block's capacity. To address this, we employ the IPFS for storing public keys and signatures, while only storing their hash values on the blockchain.

This study meticulously examines the challenges in implementing these schemes. Additionally, it extensively compares the performance of post-quantum signature schemes to pinpoint the optimal solutions for constructing robust and quantum-resistant blockchains. This research serves as a valuable contribution to the enduring sustainability and dependability of blockchain technology amid the evolving landscape of quantum computing advancements.

The structure of this paper is outlined as follows: Related research on post-quantum signature techniques for blockchain-enabled systems is covered in Section 2 of this article. Section 3 discusses blockchain-enabled systems, blockchain security, and the IPFS network. Section 4 presents the effects of quantum computing on blockchains and an overview of post-quantum cryptosystems. In Section 5, we present post-quantum signature schemes approved by the NIST. Section 6 includes the comparative study of blockchain performance on the IPFS network, evaluating the effectiveness of post-quantum signatures suggested by the NIST versus the standard ECDSA signatures. Finally, Section 7 provides the conclusions drawn from our study.

2. Related Works

Blockchain transactions that are transparent, safe, and decentralized have changed entire sectors. Two basic strategies—quantum-secured blockchain and quantum-resistant blockchain—have been employed by researchers to protect blockchain in recent years from future quantum computer attacks. Digital signatures derived from quantum-resistant algorithms are used in quantum-resistant blockchain, although their practical applications are still limited, and the problem of long public keys still exists. Numerous industries have adopted blockchain technology, and almost 3000 virtual currencies, including well-known ones like Ethereum [1], Ripple, and Tether, use it for cross-border transactions.

The family of post-quantum algorithms with the highest efficacy in countering quantum attacks, which have garnered a lot of attention recently, is lattice cryptography. A few signature techniques based on lattice cryptography have been proposed in references [2–4]. These approaches aim to enhance transaction handling within blockchain systems, yet they prove ineffective for blockchain technologies operating in computationally intricate environments. Gao et.al. unveiled a blockchain-compatible double signing method [3]. The scheme's security is solely predicated on the SIS assumption. Li et.al. created a technique for digital signatures using the bonsai trees technology [4]. Its security is established within the framework of the random oracle model. The author in [5] explores how blockchain technology is being used in smart cities and suggests a quantum-resistant blockchain platform built upon lattice cryptography. The paper [6] analyzes the most significant cryptocurrencies in the context of quantum risks, ranking them by market capitalization (MC). The author suggests a blockchain design for the Internet of Things (IoT) using the NTRU lattice, along with a cryptographic security validation for the system, in order to build a highly effective post-quantum blockchain infrastructure.

There are no recent data breaches or security breaches in the domain of blockchain using quantum computers, but there are some potential vulnerabilities due to the development of quantum computing, which highlights the urgency for introducing quantum-resistant algorithms. Demonstrations of quantum algorithms like Shor's have showcased their potential to compromise widely used encryption techniques, placing data security at risk. Additionally, there is a concern regarding historical data encrypted using vulnerable methods, as they could be exposed in the future, posing potential breaches. Quantum computing's ability to intercept and decrypt secure communications has significant implications for national security. Furthermore, blockchain technology, which relies on public key cryptography for its decentralized trust model, is particularly susceptible to quantum attacks that could expose private keys from public ones. Notably, a recent study reveals that a substantial portion of cryptocurrency holdings, including 25% of Bitcoin and 65% of ether, resides in addresses with publicly known public keys, raising concerns about their vulnerability to theft by powerful quantum computers. To mitigate these risks, it is imperative to expedite the development and adoption of quantum-resistant cryptographic solutions to ensure the continued security of digital systems and assets.

3. Blockchain Basics and IPFS

3.1. Blockchain-Enabled Systems

A technological infrastructure that uses blockchain technology to offer decentralized, open-source, and safe solutions for diverse applications is referred to as a blockchain-enabled system depicted in Figure 1.

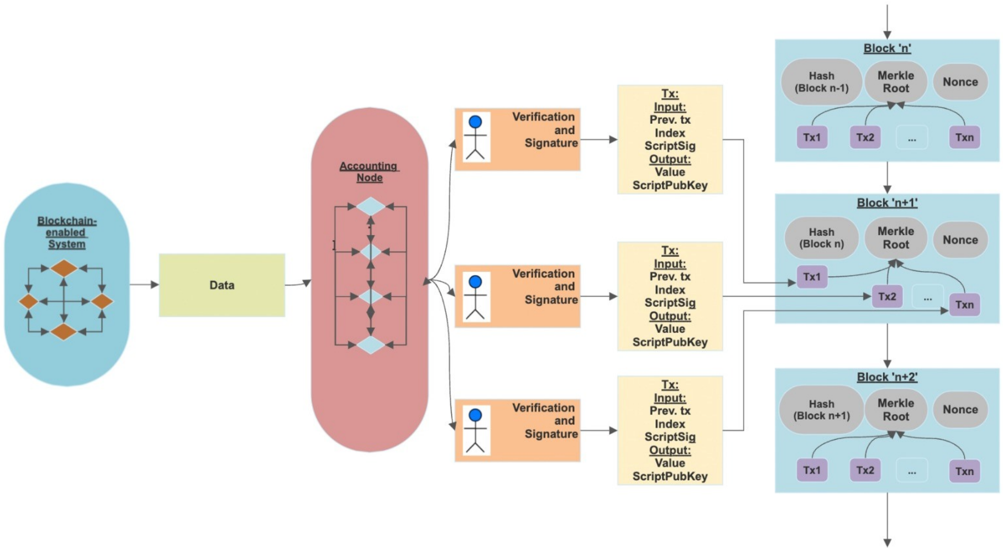
Blockchain-enabled System
Data
Accounting Node
Verification and Signature
Verification and Signature
Verification and Signature
Tx: Input: Prev. tx Index ScriptSig Output: Value ScriptPubKey
Tx: Input: Prev. tx Index ScriptSig Output: Value ScriptPubKey
Tx: Input: Prev. tx Index ScriptSig Output: Value ScriptPubKey
Block 'n'
Hash (Block n-1)
Merkle Root
Nonce
Tx1
Tx2
...
Txn
Block 'n+1'
Hash (Block n)
Merkle Root
Nonce
Tx1
Tx2
...
Txn
Block 'n+2'
Hash (Block n+1)
Merkle Root
Nonce
Tx1
Tx2
...
Txn

Figure 1. Representation of a blockchain-enabled system.

At its foundation, blockchain is a decentralized record keeping system that holds transactional data via an unchangeable and impenetrable method. By incorporating blockchain into many systems, it is possible to improve their effectiveness, reliability, and data integrity, giving them distinct advantages over classic centralized systems. Here are some essential traits and illustrations of systems that utilize blockchain technology:

1. Decentralization: blockchain removes the necessity for a central governing entity or middleman by operating within a decentralized network of interconnected nodes. By ensuring that no single entity controls the entire system, decentralization increases resilience and lowers the possibility of single points of failure.
2. Transparency and immutability: every transaction recorded on the blockchain is openly accessible to the public and is publicly viewable. A transaction becomes immutable once it is added to the blockchain, which means it cannot be changed or removed without network consent.

3. Security and data integrity: transactions are securely recorded and verified using cryptographic methods in blockchain technology. Any attempt to tamper with the data would require changing every transaction in the chain of blocks because each transaction is linked to the one before it.

3.2. Blockchain Security

Blockchain systems need to be secure in order to guard against fraud, tampering, and unauthorized access. Here, we explore key cryptographic techniques enhancing blockchain security.

1. Digital signatures: a crucial cryptographic tool utilized in blockchain technology is the digital signature. A private key and a public key are the two types of cryptographic keys that each member of the blockchain network has. The matching public key is made available to others in order to validate the legitimacy of the signatures, while the private key is kept secret and utilized to create digital signatures. Digital signatures demonstrate that a transaction has been approved by the holder of the private key, demonstrating the validity and authorization of transactions on the blockchain.
2. Hash functions: also known as message digests or hash codes. Hash functions are cryptographic techniques that transform data of any size into fixed-length, singular character strings. Each block in the blockchain contains the data from the preceding block's hash value, resulting in a chain of blocks that are connected by their hashes. The blockchain's immutability is ensured by this chaining method. It is simple to spot tampering because any alteration to the data within a block will result in a different hash value.
3. Merkle trees: Merkle trees are binary trees, also known as known as hash trees, that make it possible to verify huge datasets of data efficiently. Merkle trees are used in blockchain to arrange and compile the transactions contained in a block. Participants can quickly confirm the existence and integrity of particular transactions within a block without having to process all of the individual transactions thanks to the Merkle tree's root hash, which is included in the block header.
4. Public key infrastructure (PKI): public key infrastructure encompasses a series of protocols and processes responsible for handling the generation, dispersion, and invalidation of digital certificates and public keys. In the blockchain context, PKI enables participants in the blockchain to validate each other's public keys and confirm the validity of transactions by enabling them to authenticate each other's public keys.
5. Consensus methods: for transaction authentication and the addition of new blocks in a blockchain, proof-of-work (PoW) and proof-of-stake (PoS) consensus mechanisms employ hash functions. This guarantees that only valid transactions are added.
6. Quantum-resistant cryptography: as quantum computing advances, established cryptographic techniques employed in blockchain, such RSA and ECC, are at risk of being compromised. Adopting quantum-resistant algorithms is essential to protect blockchain security from quantum threats in the future.

Some actual threats to blockchain technology are double spending, Sybil attacks, smart contract vulnerabilities, privacy concerns, key management, distributed denial of service (DDoS) attacks, and transaction malleability.

It is worth noting that, while quantum-resistant algorithms address the threat of quantum computing, they may not directly solve the other threats listed above. The other threats are typically mitigated through various cryptographic and non-cryptographic techniques. However, quantum-resistant cryptography plays a crucial role in ensuring the long-term security of blockchain networks in the face of quantum advancements.

3.3. InterPlanetary File System

The InterPlanetary File System (IPFS) revolutionizes online file storage and sharing by establishing a permanent and decentralized method. The IPFS's primary objective is to offer a distributed file system that guarantees files that may be accessed from various places, removing the possibility of data loss due to server outages or shutdowns. The IPFS enables effective content retrieval and verification over the network by dividing files into smaller portions and giving them distinct cryptographic hashes. This cutting-edge technology has a number of benefits, including improved security, privacy, and user control. Without relying on a centralized authority, users can safely transfer files, lowering the possibility of censorship or unauthorized access to private information. Additionally, the IPFS is perfect for long-term information preservation because of its decentralized design, which permits files to endure forever. With regard to blockchain applications, the IPFS is used to store massive data, such as public keys and signatures, while the blockchain itself just stores the corresponding hash values. By doing this, the storage requirements of the blockchain are drastically decreased, resulting in more scalable and effective blockchain networks.

4. Quantum Computing's Effects on Cryptosystems and the Need for Post-Quantum Cryptosystems

4.1. Quantum Computing

The cutting-edge discipline of computing known as quantum computing uses the concepts of quantum physics to process and modify data. Quantum computers employ quantum bits, also known as qubits, which can exist in several states at once due to the phenomenon known as superposition. This is in contrast to classical computers, which use bits to encode data as either 0 or 1.

Using quantum Fourier transform [7] to solve problems related to integer factorization and discrete logarithms can be exponentially sped up using Shor's approach [8]. The searching problem can be quadratically sped up using Grover's algorithm [9]. It offers a significant speed advantage over the conventional brute force approach, which takes $O(N)$ time in classical attacks. This method can find the original input corresponding to a function's output in approximately $O(\sqrt{N})$ time. Many widely used encryption systems rely on these intricate mathematical challenges. However, quantum computers are expected to solve these problems within a bounded polynomial time.

The extent to which these quantum benefits can be developed and the duration of the feasibility gap between classical and quantum models are also unknown [10]. The question of whether it is possible to create a large-scale quantum computer is complex and contentious. Many researchers now think that enormous quantum computers are just a very difficult engineering problem, although in the past it was less obvious whether they were a physical reality.

In the next 20 years or so, according to any scientists who still make such predictions [11], powerful quantum computers will be developed that will be able to break all of the current core public key infrastructures quickly. It will take much work to enable a seamless and stable transition from the newest widely used cryptosystems to their counterparts that can withstand quantum computing. Regardless of whether we are able to predict with accuracy when the era of quantum computing will begin, we must continue building more secure communication channels that, for instance, might revolutionize the field of cryptography.

4.2. Post-Quantum Cryptosystems

Due to their reliance on mathematical problems that can be solved effectively by quantum computers, existing cryptographic systems like RSA and ECC may become vulnerable as quantum computing technology develops. A cryptographic system created on alternative mathematical structures and algorithms to survive attacks from quantum computers is known as a post-quantum cryptosystem. With the advent of potent quantum computers, these new cryptographic techniques seek to increase security and guarantee the ongoing preservation of critical data. Post-quantum signatures are known as quantum-resistant signatures or quantum-safe signatures. In the age of quantum computing, post-quantum signature systems are appropriate for secure communications and digital identity verification since they are made to be capable of withstanding assaults from both conventional and quantum computers. Post-quantum signature systems will aid in ensuring the long-term security of digital communications and safeguarding sensitive data from potential quantum attacks once they are widely adopted and put into use. In order to provide a collection of safe post-quantum cryptography algorithms, standardization efforts are still being made. To guarantee these algorithms' efficacy and compatibility, organizations like the National Institute of Standards and Technology (NIST) are driving the technique of soliciting, assessing, and standardizing them.

Post-quantum signature schemes can be classified into five categories: hash-based, lattice-based, code-based, multivariate polynomial-based, and super-singular isogeny-based schemes.

4.2.1. Code-Based Cryptosystem

Code-based cryptography relies on the difficulty of decoding specific structured linear error-correcting codes. Daniel J. Bernstein proposed Classic McEliece, which is a potential post-quantum public key cryptographic system based on error-correcting codes under consideration by the NIST for global standardisation, in 2017 [12–14]. In the McEliece concept, a public key is formed through a combination of the Goppa code and a linear transformation. To encrypt a message, the sender introduces a set level of random noise to the message [15]. Without knowing how to factor in the public key, recovering the message is a computationally difficult job for the attacker. Several code-based cryptographic methods exist, and, among them, certain ones could potentially provide security against quantum attacks: Classic McEliece, BIKE, and HQC.

4.2.2. Hash-Based Cryptosystem

The cryptographic secure hash function used in hash-based signatures is created to exhibit security properties, like being hard to reverse, resistant to finding original inputs, immune to generating similar outputs for different inputs, and robust against collision attacks. Hash-based signature schemes are classified as stateless or stateful based on their implementation approach. They can also be categorized as a one-time signature (OTS), few-time signature (FTS), multi-time signature (MTS), and hierarchical signature (HS). These classifications depend on factors such as how keys are generated, how signatures are generated, and other parameters used in their construction. $SPHINCS^+$ is a hash-based quantum-safe cryptographic algorithm. It is a signature system with no state and an improved version of SPHINCS, designed to reduce signature size [13].

4.2.3. Lattice-Based Cryptosystems

These cryptographic techniques are constructed using lattices, which are sets of points arranged periodically in multi-dimensional spaces. To find the smallest non-zero point within a lattice, a complex problem known as the shortest vector problem (SVP) is utilized. This problem, which is difficult to solve and falls under the NP-hard category, forms the foundation of security in lattice-based systems. Additional challenges related to lattices, such as the closest vector problem (CVP) and the shortest independent vectors problem (SIVP), as mentioned by [16], are currently beyond the capabilities of quantum computers. The algorithm's implementation is relatively efficient, and it provides worst-case hardness-based security proofs that are extremely strong. The quantum-resistant algorithms based on the lattice are CRYSTALS-KYBER, SABER, NTRU, FrodoKEM, NTRU Prime CRYSTALS-Dilithium, and FALCON.

4.2.4. Multivariate-Based Cryptosystems

Multivariate system of equations have been demonstrated to be NP-complete or NP-hard, and multivariate-based techniques rely on this complexity [12]. Despite their resilience to quantum assaults, more research is required to increase their decryption speed, decrease their enormous key size, and lower their ciphertext overhead [17]. Rainbow and GeMSS are the potentially quantum-safe cryptographic techniques based on multivariate quadratic equations.

4.2.5. Super Singular Elliptic Curve Isogeny Cryptosystems

These are a novel approach that first appeared in the year 2000. To create public key cryptosystems, isogeny-based cryptography employs mappings between elliptic curves. Isogeny cryptography relies on the security of solving super singular isogeny problems. These problems involve finding the connection (isogeny mapping) between two super singular elliptic curves that have an equal count of points. In comparison with other post-quantum cryptography possibilities, the protocols based on isogeny need a very small key. SIKE is one of the isogeny cryptography family's putative quantum-safe algorithms [18–20].

5. Description of NIST-Recommended Post-Quantum Signature Schemes and ECDSA

The section describes the CRYSTALS-Dilithium, FALCON, and $SPHINCS^+$ algorithms, which are identified by the NIST for standardisation.

5.1. CRYSTALS-Dilithium

A lattice-based digital signature system called CRYSTALS-Dilithium is renowned for its effectiveness and robust protection against both conventional and quantum adversaries. Algorithms 1–3, respectively, depict the procedures of key creation, signing, and verification, as in [21].

CRYSTALS-Dilithium's salient characteristics include:

- Efficiency: CRYSTALS-Dilithium is optimized for key generation, signing, and verification efficiency, making it appropriate for contexts with limited resources, including Internet of Things devices and embedded systems.
- Strong security: the system offers a strong level of resilience against several assaults, including those launched by quantum computers.

Algorithm 1 Crystal Dilithium Gen

```
Input: Security parameters $\rho, K$
Output: Public key $pk$ and secret key $sk$
$\rho \leftarrow \{0,1\}^{256}$
$K \leftarrow \{0,1\}^{256}$
$(s_1, s_2) \leftarrow S_\eta^l * S_\eta^k$
$A \in R_q^{k*l} = ExpandA(\rho)$                ▷ A is stored in NTT Domain Representation
$t = As_1 + s_2$
$(t_1, t_0) = Power2Round_q(t, d)$
$tr \in \{0,1\}^{348} = CRH(\rho \parallel t_1)$                ▷ CHR is centered rounding
$return\ (pk = (\rho, t_1),\ sk = (\rho, K, tr, s_1, s_2, t_0))$
```

Algorithm 2 Crystal Dilithium Sign

```
**Input:** Message $M$, Secret Key $sk$
**Output:** Signature $\sigma$
$A \in R_q^{\kappa * l} = ExpandA(\rho)$
$\nu \in \{0,1\}^{384} = CRH(tr \parallel M)$
$\kappa = 0, (z,h) = \perp$
**while** $(z,h) = \perp$ **do**
    $y \in S^l_{\gamma 1 - 1} = ExpandMask(K \parallel \nu \parallel \kappa)$
    $w = Ay$
    $w_1 = HighBits_q(w, 2\gamma_2)$
    $c \in B_{60} = H(\nu \parallel w_1)$
    $z = y + cs_1$
    $(r_1, r_0) = Decompose_q(w - cs_2, 2\gamma_2)$
    **if** $||z||_\infty \geq \gamma_1 - \beta or||r_0||_\infty \geq \gamma_2 - \beta orr_1 \neq w_1$ **then**
        $(z,h) = \perp$
    **else**
        $h = MakeHint_q(-ct_0, w - cs_2 + ct_0, 2\gamma_2)$
        **if** $||ct_0||_\infty \geq \gamma_2$ $or\ the\ \#\ of\ 1's\ in\ h\ is\ greater\ than\ w$ **then**
            $(z,h) = \perp$
        **end if**
    **end if**
    $\kappa = \kappa + 1$
**end while**
$return\ \sigma = (z, h, c)$
```

Algorithm 3 Crystal Dilithium Verify

```
**Input:** Message $M$, Signature $\rho$, Public Key $pk$
**Output:** True if the signature is valid, False otherwise
$A \in R_q^{k * l} = ExpandA(\sigma)$                ▷ A is stored in NTT Domain Representation
$\nu \in \{0,1\}^{384} = CRH(CRH(\sigma \parallel t_1) \parallel M)$
$w^{'}_1 = UseHint_q(h, Az - ct_1 \cdot 2^d, 2\gamma_2)$
$return\ [||z||_\infty < \gamma_1 - \beta]\ and\ [c = H(\nu \parallel w^{'}_1)]\ and\ [\#\ of\ 1's\ in\ h\ is \leq w]$
```

5.2. FALCON

This is also a lattice-based digital signature algorithm called FALCON [22].

Consider the polynomial $\phi = x^n + 1$, where n is a power of two denoted by $n = 2^\kappa$, and let q be a positive integer. In the NTRU cryptography context, a set of NTRU secrets comprises four polynomials: f, g, F, and G, all of which are elements of the polynomial ring $\mathbb{Z}[x]/(\phi)$. These polynomials are chosen to satisfy the NTRU equation: $fG - gF =$ mod ϕ. Algorithms 4–6, respectively, depict the procedures of key creation, signing, and verification, as in [22].

The key creation, signing, and verification steps are represented in Algorithms 4–6, respectively. FALCON's salient features include:

- High performance: FALCON is ideal for a variety of applications since it is built to strike a balance between security and performance.
- Small signature sizes: FALCON has smaller signature sizes than certain other lattice-based schemes, which is helpful in situations when bandwidth is limited.
- Compact key generating: the speedy setup for users is made possible by FALCON's effective key-generating procedure.

Algorithm 4 FALCON Key Generation

```
**Input:** Security parameter $n$
**Output:** Public key $pk$ and secret key $sk$
$A \in Z_q^{m \times n}$
$T \in Z_q^{m \times m}$ invertible modulo $q$
$s \in Z_q^n$
$b = As \bmod q$
$pk = (A, b)$
$sk = (T, s)$
$return\ (pk, sk)$
```

Algorithm 5 FALCON Sign

```
**Input:** Message $msg$, Secret key $sk$
**Output:** Signature $\sigma$
$r \in Z_q^m$
$R = Ar \bmod q$
$e = \text{SHA3}(\text{concat}(R, msg)), e \in Z_q^n$
$y = s + e \bmod q$
$c = \text{SHA3}(\text{concat}(R, y, msg)), c \in Z_q^m$
$z = r + Tc \bmod q$
$\sigma = (R, z)$
$return\ \sigma$
```

Algorithm 6 FALCON Verify

```
**Input:** Message $msg$, Signature $\sigma$, Public key $pk$
**Output:** True if the signature is valid, False otherwise
$\sigma = (R, z)$
$pk = (A, b)$
$c = \text{SHA3}(\text{concat}(R, Az - R)), c \in Z_q^m$
$W = Az - R + c \bmod q$
$v = b - W \bmod q$
$e' = \text{SHA3}(\text{concat}(R, z, msg)), e' \in Z_q^n$
**if** $v = e'$ **then**
    **return** False
**end if**
**return** True
```

5.3. $SPHINCS^+$

Modern hash-based signature technology, such as $SPHINCS^+$ using $WOTS^+$ and FORS (hash-based signatures), provides strong security against both classical and quantum adversaries [13]. It addresses some of the flaws in the original SPHINCS system and is an enhanced version of it.

For clarity and a more in-depth look into the algorithm and its parameters, and on $WOTS^+$ and FORS, refer to [13]. Algorithms 7–9, respectively, depict the procedures of key creation, signing, and verification, as mentioned in [23].

Algorithm 7 $SPHINCS^+$ Key Generation

```
Input: Security parameter $n$
Output: $SPHINCS^+$ key pair $(SK, PK)$
$SK$.seed $\leftarrow$ sec_rand$(n)$
$SK$.prf $\leftarrow$ sec_rand$(n)$
$PK$.seed $\leftarrow$ sec_rand$(n)$
$PK$.root $\leftarrow$ ht_PKgen$(SK$.seed$, PK$.seed$)$
return $((SK$.seed$, SK$.prf$, PK$.seed$, PK$.root$), (PK$.seed$, PK$.root$))$
```

$SPHINCS^+$'s salient attributes include:

- Quantum resistance: $SPHINCS^+$ is built to withstand Shor's algorithm and other quantum attacks, making it secure with quantum computers.
- Fast signing and verification: $SPHINCS^+$ strives for fast signing and verification while retaining excellent security.
- Hash-based security: $SPHINCS^+$ is built on hash functions, which have received much attention and are commonly regarded as being quantum resistant.

Algorithm 8 $SPHINCS^+$ Signing Algorithm

```
Input: Message $M$, Private key $SK$
Output: $SPHINCS^+$ signature $SIG$
$ADRS \leftarrow$ toByte$(0, 32)$
$opt \leftarrow$ toByte$(0, 32)$
if $RANDOMIZE$ then
    $opt \leftarrow$ rand$(n)$
end if
$R \leftarrow$ PRF_msg$(SK$.prf$, opt, M)$
$SIG \leftarrow SIG \,||\, R$
$digest \leftarrow H_{\text{msg}}(R, PK$.seed$, PK$.root$, M)$
$tmp\_md \leftarrow$ first floor$((ka + 7)/8)$ bytes of $digest$
$tmp\_idx\_tree \leftarrow$ next floor$((h - h/d + 7)/8)$ bytes of $digest$
$tmp\_idx\_leaf \leftarrow$ next floor$((h/d + 7)/8)$ bytes of $digest$
$md \leftarrow$ first $ka$ bits of $tmp\_md$
$idx\_tree \leftarrow$ first $(h - h/d)$ bits of $tmp\_idx\_tree$
$idx\_leaf \leftarrow$ first $(h/d)$ bits of $tmp\_idx\_leaf$
$ADRS$.setLayerAddress$(0)$
$ADRS$.setTreeAddress$(idx\_tree)$
$ADRS$.setType(FORS_TREE)
$ADRS$.setKeyPairAddress$(idx\_leaf)$
$SIG\_FORS \leftarrow$ fors_sign$(md, SK$.seed$, PK$.seed$, ADRS)$
$SIG \leftarrow SIG \,||\, SIG\_FORS$
$PK\_FORS \leftarrow$ fors_pkFromSig$(SIG\_FORS, M, PK$.seed$, ADRS)$
$ADRS$.setType(TREE)
$SIG\_HT \leftarrow$ ht_sign$(PK\_FORS, SK$.seed$, PK$.seed$, idx\_tree, idx\_leaf)$
$SIG \leftarrow SIG \,||\, SIG\_HT$
return $SIG$
```

Algorithm 9 $SPHINCS^+$ Verification Algorithm

```
Input: Message $M$, Signature $\sigma$, Public key $pk$
Output: True if the signature is valid, False otherwise
$ADRS \leftarrow \text{toByte}(0, 32)$
$R \leftarrow$ SIG.getR()
$SIG\_FORS \leftarrow$ SIG.getSIG_FORS()
$SIG\_HT \leftarrow$ SIG.getSIG_HT()
$digest \leftarrow H_{\text{msg}}(R, PK.\text{seed}, PK.\text{root}, M)$
$tmp\_md \leftarrow$ first floor$((ka + 7)/8)$ bytes of $digest$
$tmp\_idx\_tree \leftarrow$ next floor$((h - h/d + 7)/8)$ bytes of $digest$
$tmp\_idx\_leaf \leftarrow$ next floor$((h/d + 7)/8)$ bytes of $digest$
$md \leftarrow$ first $ka$ bits of $tmp\_md$
$idx\_tree \leftarrow$ first $(h - h/d)$ bits of $tmp\_idx\_tree$
$idx\_leaf \leftarrow$ first $(h/d)$ bits of $tmp\_idx\_leaf$
$ADRS.\text{setLayerAddress}(0)$
$ADRS.\text{setTreeAddress}(idx\_tree)$
$ADRS.\text{setType}(\text{FORS\_TREE})$
$ADRS.\text{setKeyPairAddress}(idx\_leaf)$
$PK\_FORS \leftarrow \text{fors\_pkFromSig}(SIG\_FORS, md, PK.\text{seed}, ADRS)$
$ADRS.\text{setType}(\text{TREE})$
return $\text{ht\_verify}(PK\_FORS, SIG\_HT, PK.\text{seed}, idx\_tree, idx\_leaf, PK.\text{root})$
```

5.4. ECDSA

The equation $y^2 = x^3 + 7$ defines the Secp256k1 curve as a prime-order elliptic curve over a finite field for use in cryptographic techniques. A popular cryptographic algorithm is the elliptic curve digital signature algorithm (ECDSA), which uses the secp256k1 curve for generating digital signatures.

Algorithms 10–12, respectively, depict the procedures of key generation, signing, and verification, as in [24].

Algorithm 10 ECDSA Key Generation

```
Input: Elliptic curve parameters $a, b, p, G, n$
Output: Public key $Q$ and private key $d$
$K \in \{1, 2, ...n - 1\}$
$Q = kG$
$d \in \{1, 2, ...n - 1\}$
```

Algorithm 11 ECDSA Sign

```
Input: Message $msg$, Private key $d$, Elliptic curve parameters $a, b, p, G, n$
Output: Signature $(r, s)$
$K \in \{1, 2, ...n - 1\}$
$P = kG$
$r \equiv x_P \pmod{n}$, where $x_P$ is the x-coordinate of $P$
if $r = 0$ then
    go to Step 1
end if
$e = \text{HashToNumber}(\text{SHA3}(msg)) \in \mathbb{Z}_n$
$s \equiv (e + dr) \cdot k^{-1} \pmod{n}$
if $s = 0$ then
    go to Step 1
end if
```

Algorithm 12 ECDSA Verify

```
**Input:** Message $msg$, Signature $(r, s)$, Public key $Q$, Elliptic curve parameters $a, b, p, G, n$
**Output:** True if the signature is valid, False otherwise
$e = \text{SHA3}(msg) \in \mathbb{Z}_n$
$w \equiv s^{-1} \pmod{n}$
$u_1 \equiv ew \pmod{n}$ and $u_2 \equiv rw \pmod{n}$
$P = u_1 G + u_2 Q$
**if** $P = \infty$ or $r \equiv x_P \pmod{n}$ **then**
    **return** False
**end if**
**return** True
```

The ECDSA's salient characteristics include:

- Based on the discrete logarithm problem for elliptic curves, security.
- Efficient key sizes, suitable for constrained environments.
- Widely used in various cryptographic applications due to its practicality and effectiveness.
- Potential vulnerability to quantum attacks in the future.

The performance metrics of the post-quantum signature schemes Dilithium3, $SPHINCS^+$ + SHAKE256s, Falcon1024, and $SPHINCS^+$ + SHA256s are depicted in Figure 2.

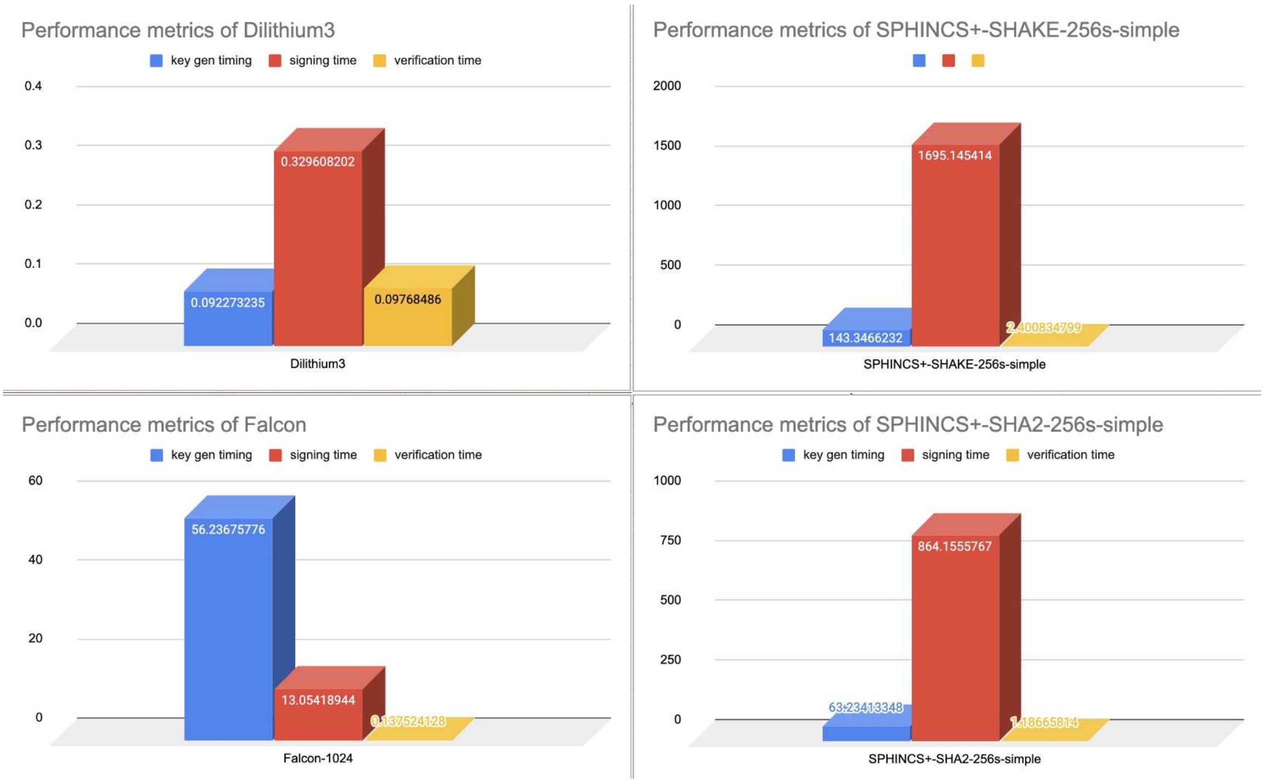

Figure 2. Performance of the respective algorithms.

6. Experiment, Analysis, and Results

To provide complete security, the system makes use of quantum-resistant digital signatures. We will examine the system's performance and efficiency in this section. Ten alternative blockchain systems are examined in the same simulation scenario:

- Falcon1024, Dilithium3, $SPHINCS^+$ + SHA256s, and $SPHINCS^+$ + SHAAKE256s are used in blockchains with the IPFS network, since the ECDSA, which is not quantum resistant, is excluded from this comparison.

- Falcon1024, Dilithium3, $SPHINCS^+$ + SHA256s, $SPHINCS^+$ + SHAKE256s, and the ECDSA are used in blockchains without an IPFS network.

Efficiency:

- For blockchains that do not employ the IPFS, the performances of key sizes, signature sizes, signature time, key generation time, and signature verification time are compared, and specific variations of the algorithms are chosen accordingly. The parameters of the algorithms are chosen to obtain the same level of security 256 across every algorithm.
- The same parameters are used to compare blockchains using the IPFS. Every aspect of the algorithms is evaluated 1000 times to establish the average duration of key generation, signing, and verification.

Tables 1–6 present a comparative analysis of five signature schemes: ECDSA, $SPHINCS^+$ + SHA256s, $SPHINCS^+$ + SHAKE256s, Dilithium, and Falcon, which are represented in Figures 3–8, respectively. The parameters for each algorithm are selected appropriately to attain the same security level of 256 bits, which results in a range of key lengths. The comparisons are based on various factors, including key generation time, signing time, verification time, and sizes of the secret key, public key, and signature. The results indicate that the ECDSA stands out as the most efficient algorithm, although its vulnerability to quantum attacks is a significant drawback.

Given the ECDSA's susceptibility to quantum threats, the focus shifts to the remaining four quantum-resistant algorithms: $SPHINCS^+$ + SHA256s, $SPHINCS^+$ + SHAKE256s, Dilithium, and Falcon. When evaluating the same criteria as above, it becomes evident that Dilithium3 emerges as the most efficient alternative.

However, a closer examination of Table 3, which compares the public key sizes of the quantum-resistant algorithms and is depicted in Figure 5, reveals that Dilithium3 has the largest public key size. This aspect becomes a notable drawback in blockchain technology, as each transaction in the network incorporates the public key, leading to an increase in the transaction data size.

To gauge the impact of quantum-resistant algorithms on the network, we constructed a UTXO model. This model facilitated a comparison of block mining time, transaction size, and overall network efficiency. Throughout all operations, including mining and transaction creation, the difficulty level of zeros was fixed at 3. In assessing key generation, signing, and verification times, we conducted these operations 1000 times to obtain averaged timing results.

Table 1. In terms of key generation time, algorithms are compared.

System	Falcon-1024	Dilithium3	$SPHINCS^+$ -sha2-256s	$SPHINCS^+$ -shake-256s	ECDSA
Key Generation Time (ms)	56.23675776	0.092273235	63.23413348	143.3466232	0.8477787971

Table 2. Algorithms are compared with respect to secret key size (bytes).

System	Falcon-1024	Dilithium3	$SPHINCS^+$ -sha2-256s	$SPHINCS^+$ -shake-256s	ECDSA
Secret key size (bytes)	1281	4000	128	128	32

Table 3. Algorithms are compared with respect to public key size (bytes).

System	Falcon-1024	Dilithium3	$SPHINCS^+$ -sha2-256s	$SPHINCS^+$ -shake-256s	ECDSA
Public key size (bytes)	897	1952	64	64	32
Public key size + IPFS (bytes)	32	32	32	32	32

Table 4. Algorithms are compared with respect to signature size (bytes).

System	Falcon-1024	Dilithium3	$SPHINCS^+$ -sha2-256s	$SPHINCS^+$ -shake-256s	ECDSA
Signature size (bytes)	666	3293	29,792	29,792	64
Signature size + IPFS (bytes)	32	32	32	32	32

Table 5. Algorithms are compared with respect to signature time (ms).

System	Falcon-1024	Dilithium3	$SPHINCS^+$ -sha2-256s	$SPHINCS^+$ -shake-256s	ECDSA
Signing time (ms)	13.05418944	0.329608202	864.1555767	1695.145414	1.154619694

Table 6. Algorithms are compared with respect to verification time (ms).

System	Falcon-1024	Dilithium3	$SPHINCS^+$ -sha2-256s	$SPHINCS^+$ -shake-256s	ECDSA
Verification time (ms)	0.13752412	0.09768486	1.18665814	2.40083479	3.692177057
Verification time + IPFS (ms)	1.285585641	0.273122786	2.277437449	3.61640429	4.23109493

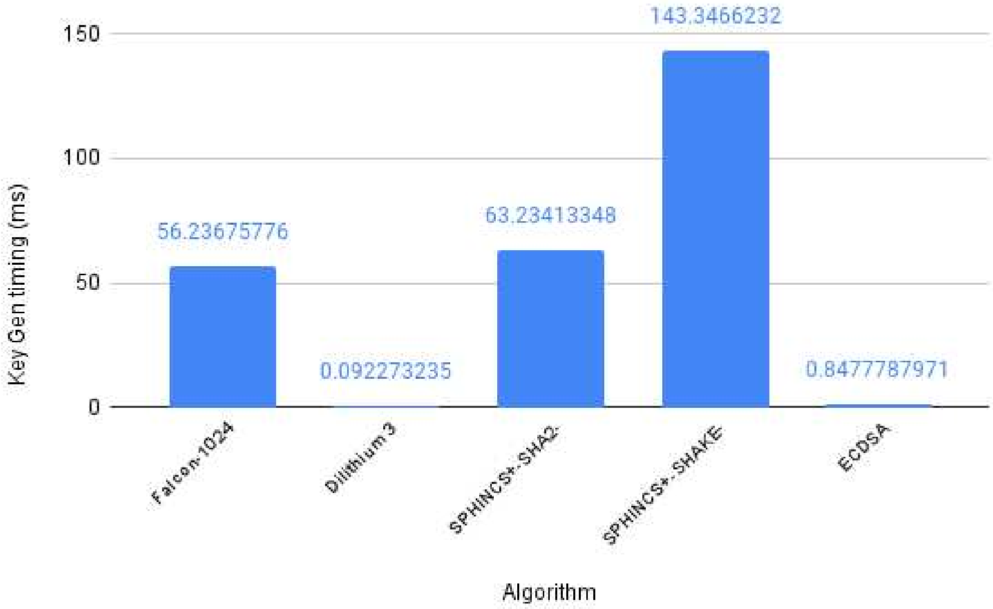

Figure 3. Comparison of key generation time.

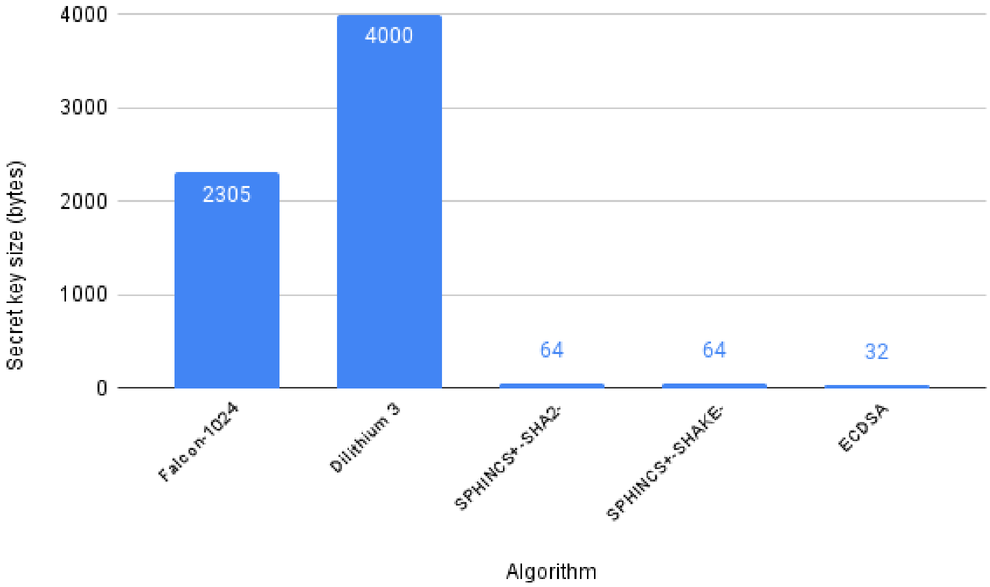

Figure 4. Comparison of secret key size.

On Using IPFS

A comparison is made with the same algorithms but with the addition of IPFS (InterPlanetary File System) storage for reducing signature/public key sizes. The proposed method adds extra time for verification for all algorithms, but drastically reduces the size of the public key and the size of the signature. In Tables 1–6, the results show that the time for key generation, time for signing, and time for verification are increased marginally for all schemes. The signature/public key sizes are significantly reduced, with only 32 bytes required for each of those values. Dilithium remains the fastest signature scheme after applying the IPFS-based approach. Another interesting inference is that the signature size of $SPHINCS^+$ is reduced by more than 99%. In Table 7, the results divulge a considerable reduction in block mining times through the utilization of the IPFS which is shown in Figure 9. This reduction can be attributed to the diminished public key size facilitated by the IPFS, consequently leading to a reduction in the transaction size as well.

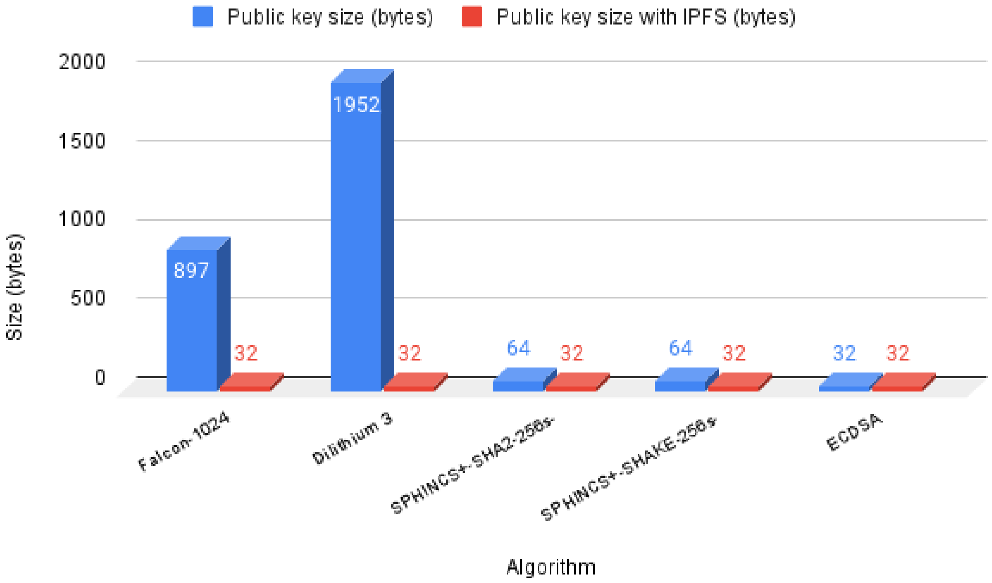

Figure 5. Comparison of public key size.

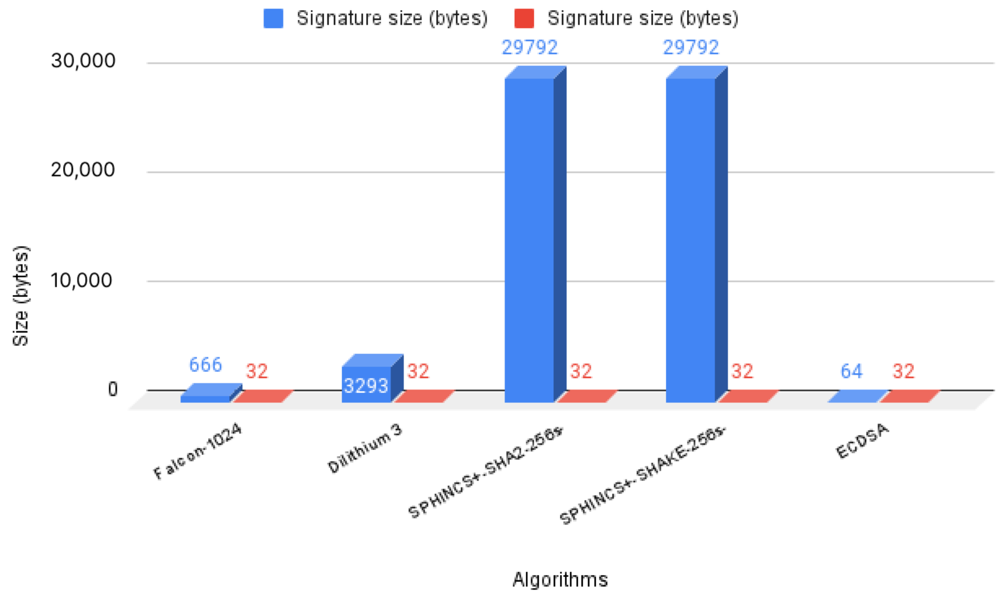

Figure 6. Comparison of signature size.

Table 7. Comparison of mining time (ms).

System	Falcon-1024	Dilithium3	$SPHINCS^{+}$ -sha2-256s	$SPHINCS^{+}$ -shake-256s
Mining time (ms)	409.2272282	653.0937672	2219.734669	3405.236483
Mining time + IPFS (ms)	176.66924	329.6942472	2198.122358	2405.548263

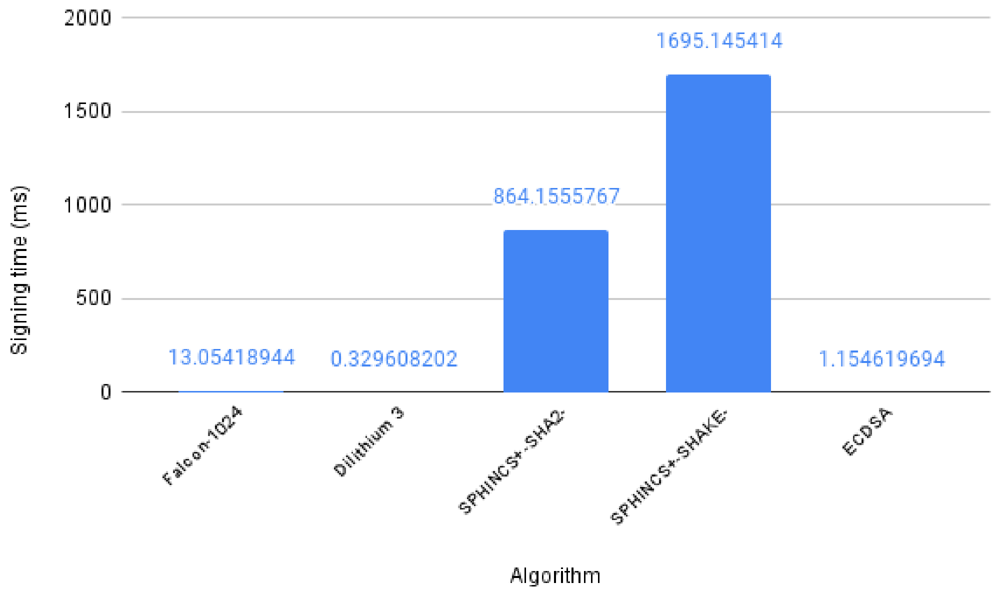

Figure 7. Comparison of signature time.

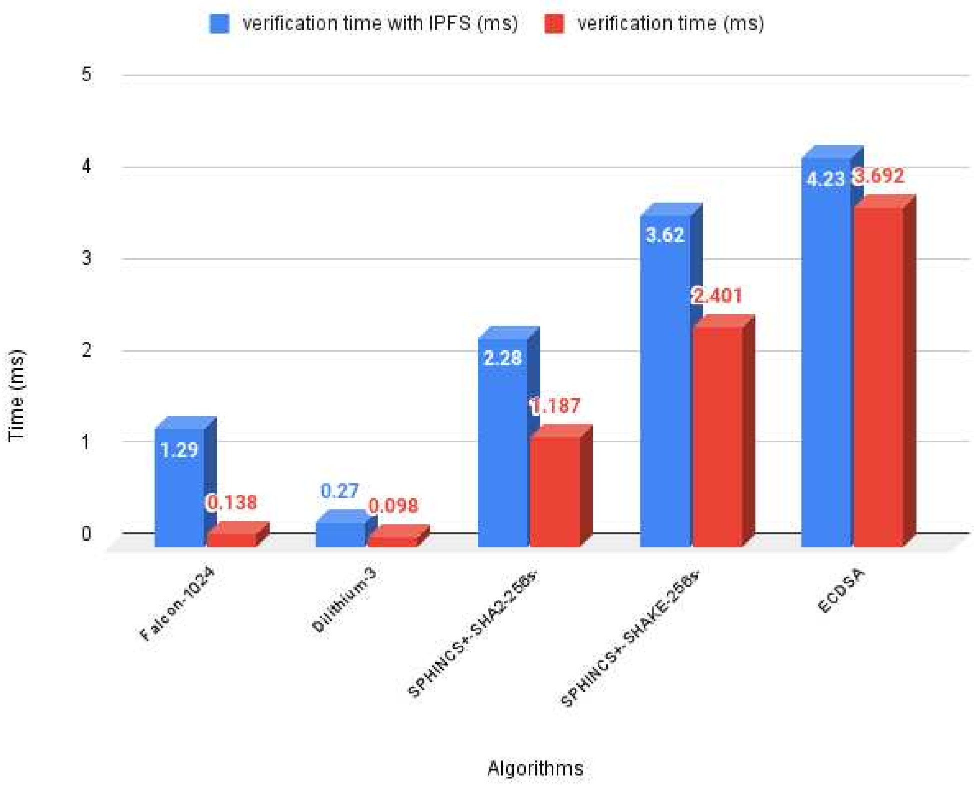

Figure 8. Comparison of verification time.

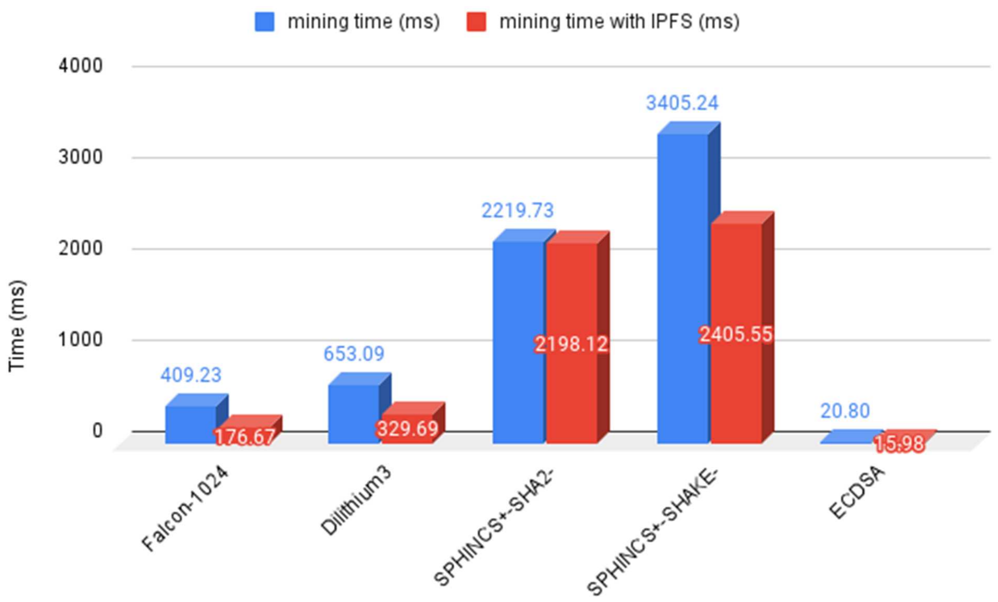

Figure 9. Comparison of mining time.

7. Conclusions

The blockchain system requires digital signatures for authenticity and integrity. Despite the fact that the ECDSA is still in use today in the blockchain system, it is not advised because the security will be undermined after quantum technology. We added the NIST-recommended post-quantum signatures Dilithium, FALCON, and $SPHINCS^{+}$ to the blockchain and analysed their performance compared with the widely used ECDSA. The Falcon and Dilithium-based systems are recommended for applications that prioritize strong performance in key generation, signing, and verification times, especially when utilizing the suggested IPFS for managing large keys. If the IPFS is not preferred, then Falcon is a suitable choice. Block capacity as well as the issue of quantum attack are both resolved in this manner. Overall, the proposed IPFS-based approach successfully reduces the signature/public key sizes for all signature schemes evaluated, which can greatly improve the efficiency of blockchain systems. The paper primarily delves into the UTXO model, primarily utilized in Bitcoin systems. Nonetheless, it is worth emphasizing that these quantum-resistant algorithms could be extended to various other blockchain models, including those employed by Ethereum, Polygon, Cosmos, and similar platforms, but mention applicability to Ethereum, Polygon, and similar platforms. While Bitcoin systems primarily facilitate transactions, Ethereum and related systems allow users to include data other than transactions in the blockchain. This key distinction introduces variability in block sizes, thus rendering the problem more intricate. Consequently, it imposes additional constraints on the calculation of mining times. These aforementioned areas represent promising avenues for future research in the realm of quantum-resistant blockchain systems.

Open Problem: More research into establishing better post-quantum signature schemes that provide reduced key and signature sizes and faster key generation, signatures, and verification times would be one of the main open problems in this domain. Another area of attention would be strengthening the IPFS component of the system.

Author Contributions: Conceptualization, P.T. and A.R.; data curation, J.M.M. and G.P.J.; formal analysis, J.M.M.; funding acquisition, G.P.J. and W.C.; investigation, J.M.M.; methodology, P.T. and W.C.; project administration, G.P.J. and W.C.; resources, A.R. and W.C.; software, A.R.; supervision, P.T. and G.P.J.; validation, P.T., J.M.M. and W.C.; visualizations, A.R. and J.M.M.; writing—original draft, P.T. and A.R.; writing—review and editing, G.P.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ulsan City & Electronics and Telecommunications Research Institute (ETRI) grant funded by the Ulsan City [23AS1600, the development of intelligentization technology for the main industry for manufacturing innovation and Human-mobile-space autonomous collaboration intelligence technology development in industrial sites].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
2. Yin, W.; Wen, Q.; Li, W.; Zhang, H.; Jin, Z. An anti-quantum transaction authentication approach in blockchain. *IEEE Access* **2018**, *6*, 5393–5401. [CrossRef]
3. Gao, Y.L.; Chen, X.B.; Chen, Y.L.; Sun, Y.; Niu, X.X.; Yang, Y.X. A secure cryptocurrency scheme based on post-quantum blockchain. *IEEE Access* **2018**, *6*, 27205–27213 [CrossRef]
4. Li, C.Y.; Chen, X.B.; Chen, Y.L.; Hou, Y.Y.; Li, J. A new lattice-based signature scheme in post-quantum blockchain network. *IEEE Access* **2018**, *7*, 2026–2033. [CrossRef]
5. Zhang, P.; Wang, L.; Wang, W.; Fu, K.; Wang, J. A blockchain system based on quantum-resistant digital signature. *Secur. Commun. Netw.* **2021**, *2021*, 6671648. [CrossRef]
6. Ciulei, A.T.; Creţu, M.C.; Simion, E. Preparation for post-quantum era: A survey about blockchain schemes from a post-quantum perspective. *Cryptol. ePrint Archive* **2022**. Available online: https://eprint.iacr.org/2022/026 (accessed on 16 March 2023).
7. Nielsen, M.A.; Chuang, I.L. Quantum computation and quantum information. *Phys. Today* **2001**, *54*, 60.
8. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
9. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.

10. Chen, L.; Chen, L.; Jordan, S.; Liu, Y.K.; Moody, D.; Peralta, R.; Perlner, R.; Smith-Tone, D. *Report on Post-Quantum Cryptography*; US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2016; Volume 12.
11. Mosca, M. Cybersecurity in an era with quantum computers: Will we be ready? *IEEE Secur. Priv.* **2018**, *16*, 38–41. [CrossRef]
12. Bernstein, D.J. Introduction to post-quantum cryptography. In *Post-Quantum Cryptography*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 1–14
13. Bernstein, D.J.; Hülsing, A.; Kölbl, S.; Niederhagen, R.; Rijneveld, J.; Schwabe, P. The $SPHINCS^+$ signature framework. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 2129–2146.
14. Singh, H. Code based cryptography: Classic mceliece. *arXiv* **2019**, arXiv:1907.12754.
15. McEliece, R.J. A public-key cryptosystem based on algebraic. *Coding Thv.* **1978**, *4244*, 114–116.
16. Ajtai, M. Generating hard instances of lattice problems. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 99–108.
17. Petzoldt, A.; Bulygin, S.; Buchmann, J. Selecting parameters for the rainbow signature scheme. In Proceedings of the Post-Quantum Cryptography: Proceedings of the Third International Workshop, PQCrypto 2010, Darmstadt, Germany, 25–28 May 2010; Proceedings 3; Springer: Berlin/Heidelberg, Germany, 2010; pp. 218–240.
18. Costello, C. Supersingular isogeny key exchange for beginners. In Proceedings of the Selected Areas in Cryptography, SAC 2019: 26th International Conference, Waterloo, ON, Canada, 12–16 August 2019; Revised Selected Papers 26; Springer: Berlin/Heidelberg, Germany, 2020; pp. 21–50.
19. De Feo, L. Mathematics of isogeny based cryptography. *arXiv* **2017**, arXiv:1711.04062.
20. Taraskin, O.; Soukharev, V.; Jao, D.; LeGrow, J.T. Towards isogeny-based password-authenticated key establishment. *J. Math. Cryptol.* **2020**, *15*, 18–30. [CrossRef]
21. Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2018**, *2018*, 238–268. [CrossRef]
22. Fouque, P.A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Prest, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Submiss. *NIST's Post-Quantum Cryptogr. Stand. Process* **2018**, *36*, 1–75.
23. Augot, D.; Batina, L.; Bernstein, D.; Bos, J.; Buchmann, J.; Castryck, W.; Dunkelman, O.; Guneysu, T.; Gueron, S.; Hulsing, A.; et al. *Post-Quantum Cryptography for Long-Term Security*; Rep. ICT-645622; PQCRYPTO: Eindhoven, The Netherlands, 2015
24. Johnson, D.; Menezes, A.; Vanstone, S. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.