Bitcoin Quantum (BTQ)

Bitcoin Quantum
Whitepaper
Author's E-mail address: admin@bitcoinq.xyz

ABSTRACT

Bitcoin Quantum (BTQ) is a cryptocurrency designed to offer a secure, sustainable, and scalable currency platform, addressing the issues of security and mining centralization that Bitcoin currently faces. Utilizing quantum-resistant technologies and a long-term emission plan, BTQ aims to ensure safety and stability for the future of cryptocurrency.

1. INTRODUCTION

Bitcoin Quantum (BTQ) is the culmination of extensive research and development aimed at creating an improved version of Bitcoin, integrating quantum-resistant security solutions to combat the rapidly advancing quantum computing technologies. BTQ focuses not only on security but also emphasizes the sustainability and scalability of the system.

2. PROBLEM STATEMENT

While Bitcoin has become a widely adopted digital currency, it still faces significant limitations regarding security, mining centralization, and scalability. The reliance on traditional cryptographic technologies also increases the risk from quantum computers, which could potentially break current security algorithms.

2.1 Bitcoin Transaction Security

Bitcoin transactions are currently secured using elliptic curve cryptography (ECC), specifically the Elliptic Curve Digital Signature Algorithm (ECDSA). When a Bitcoin transaction is created, it includes a signature generated by the private key associated with the Bitcoin address. This signature ensures the authenticity and integrity of the transaction, allowing the network to verify that the transaction was indeed authorized by the rightful owner of the funds.

The primary security of Bitcoin transactions relies on the computational difficulty of deriving a private key from its corresponding public key. As long as the private key is kept secret, the funds are secure. However, once a transaction is signed and broadcasted, the public key is revealed and stored on the blockchain. This exposure makes the transaction vulnerable to potential future quantum computing attacks, as quantum computers could theoretically reverse-engineer the private key from the public key.

2.2 Quantum Computing Attack Vectors

Quantum computing poses a significant threat to traditional cryptographic systems, including those used in Bitcoin. The most notable quantum algorithm that threatens ECC is Shor's algorithm, which can efficiently solve the discrete logarithm problem underlying ECC. This capability means that a sufficiently powerful quantum computer could potentially derive the private key from a Bitcoin public key, compromising the security of the associated funds.

Currently, the development of quantum computers has not advanced to the point where they can break ECC in practice. However, progress in this field is rapid, and the risk of a breakthrough in quantum computing technology necessitates the exploration of quantum-resistant cryptographic solutions. A silent quantum computing advance followed by targeted attacks on exposed public keys could have devastating consequences for Bitcoin, potentially leading to significant financial losses and a loss of confidence in the system.

2.3 Quantum Computing Attack Vectors

Quantum-resistant cryptographic signatures are designed to remain secure even in the presence of powerful quantum computers. Several cryptographic systems are believed to be resistant to both classical and quantum attacks, including hash-based cryptography, code-based cryptography, lattice-based cryptography, and multivariate quadratic equations cryptography. Among these, hash-based cryptographic signatures are well-studied and represent a promising candidate for post-quantum signatures.

2.4 Hash-Based Digital Signatures

Hash-based digital signatures rely on the security of cryptographic hash functions, which take a message and produce a fixed-length hash digest. The security of these signatures is based on the collision resistance, pre-image resistance, and second pre-image resistance properties of the hash function. Grover's quantum algorithm can reduce the security level of hash functions by performing a pre-image attack in O(2^n/2) operations, necessitating longer hash digest lengths to maintain security.

3. BTQ PROJECT

3.1 Introduction

BTQ employs a Proof of Work (PoW) algorithm augmented with quantum-resistant security measures, such as the Extended Merkle Signature Scheme (XMSS), to create a high-security platform capable of resisting quantum computer attacks. BTQ also designs a currency emission plan that spans 1000 years, aiming to minimize price volatility risks and enhance market stability.

3.2 Extended Merkle Signature Scheme (XMSS)

As the threat of quantum computing looms, the cryptographic community has turned its focus to developing secure and efficient digital signature schemes that can withstand quantum attacks. One such solution is the Extended Merkle Signature Scheme (XMSS). XMSS is a hash-based digital signature scheme designed to offer strong security guarantees against quantum threats, leveraging a structure that ensures forward security and existential unforgeability under chosen message attacks.

3.2.1 Understanding XMSS

XMSS stands out due to its reliance on hash functions rather than traditional number-theoretic problems, which are vulnerable to quantum attacks. Here's a detailed look at its core components and functionalities:

Merkle Tree Structure

- **Leaf Nodes:** Each leaf node in the Merkle tree represents a one-time signature (OTS) key pair. These key pairs are used once to sign individual messages, preventing key reuse and reducing the risk of cryptographic attacks.
- **Root Node:** The root of the Merkle tree acts as the public key. This structure allows for a compact representation of multiple signatures under a single public key.

One-Time Signature (OTS) Approach

- **Security Through Non-Reuse:** By using each key pair only once, XMSS mitigates the risks associated with key reuse, a common vector for cryptographic attacks.
- **Bitmask XOR:** Before concatenating hashes into parent nodes, XMSS utilizes a bitmask XOR of child nodes. This enhances security by ensuring the integrity of the hashing process, contributing to the scheme's resistance to collision attacks.

Forward Security and Unforgeability

- **Forward Security:** XMSS is designed to ensure that even if a private key is compromised, previously generated signatures remain secure. This forward security is crucial for long-term data integrity.
- **Existential Unforgeability:** The scheme is existentially unforgeable under chosen message attacks, meaning that an attacker cannot forge a signature for any message not previously signed,

even if they have access to other signatures and messages.

3.2.2 Advantages of XMSS

- **Quantum Resistance:** Unlike traditional RSA or ECC-based systems, which can be easily broken by quantum computers, XMSS remains secure due to its foundation on hash functions, which are resistant to quantum attacks.
- **Efficient and Scalable:** The Merkle tree structure of XMSS allows for efficient and scalable signature verification, making it suitable for systems requiring high throughput and low latency.
- **Compact Signatures:** Despite its robust security features, XMSS maintains relatively compact signature sizes, ensuring that the overhead for storage and transmission remains manageable.

3.2.3 Implementation and Use Cases

XMSS is particularly well-suited for blockchain and distributed ledger technologies, where security and efficiency are paramount. Its quantum-resistant nature makes it an ideal candidate for securing blockchain transactions and ensuring the integrity of data in a post-quantum world. Here are some potential applications:

- **Blockchain Transactions:** Securing transactions on a blockchain, ensuring that the authenticity and integrity of transactions are maintained against quantum threats.
- **Data Integrity:** Protecting the integrity of data in distributed systems, where secure and verifiable data is critical.
- **IoT Devices:** Ensuring secure communication and data exchange between Internet of Things (IoT) devices, which often operate in environments where quantum security is necessary.

3.2.4 Conclusion

The Extended Merkle Signature Scheme (XMSS) represents a significant advancement in cryptographic security, particularly in the face of emerging quantum computing threats. By leveraging hash-based techniques and a Merkle tree structure, XMSS provides a robust, scalable, and efficient solution for digital signatures, ensuring the integrity and authenticity of data in various applications. As we move towards a quantum era, adopting schemes like XMSS will be crucial for maintaining the security of our digital infrastructure.

3.3 The Dual Mining Mechanism of BTQ: Ensuring Decentralization and Security

Bitcoin Quantum (BTQ) introduces an innovative dual mining mechanism designed to enhance participation and mitigate the risks of mining power centralization. By incorporating both the RandomX algorithm for CPU mining and Cryptonight cn/0 for GPU mining, BTQ ensures a more inclusive and secure mining process.

3.3.1 RandomX Algorithm for CPU Mining

The RandomX algorithm plays a crucial role in BTQ's mining strategy by prioritizing CPU-friendly mining. Here's a detailed look at its key features:

CPU-Friendly Design

- **Accessibility:** RandomX is optimized for CPUs, allowing individuals with consumer-grade hardware to participate effectively in the mining process. This accessibility is vital for ensuring that mining is not limited to those with specialized and expensive equipment.

Memory-Intensive

- **ASIC Resistance:** The algorithm is designed to be memory-intensive, which makes it resistant to ASIC (Application-Specific Integrated Circuit) dominance. ASICs are typically more efficient than general-purpose hardware for specific tasks, but their high cost and specialization can lead to centralization of mining power. RandomX's memory requirements help to level the playing field by making ASICs less advantageous.

Decentralization

- **Broad Participation:** By enabling more people to mine using readily available CPUs, RandomX promotes decentralization. This broad participation is crucial for maintaining the security and integrity of the BTQ network, as a more distributed mining base reduces the risk of attacks and centralization.

3.3.2 Cryptonight cn/0 for GPU Mining

Complementing RandomX, the Cryptonight cn/0 algorithm is optimized for GPU mining. Here's why this approach is significant:

GPU Optimization

- **Efficiency**: Cryptonight cn/0 is tailored for GPUs, providing an efficient and secure mining process. GPUs, with their parallel processing capabilities, are well-suited for the computational tasks required by this algorithm.

Diverse Device Compatibility

- **Wide Range of Devices**: The algorithm can be performed by a variety of devices, including those already engaged in other GPU mining activities. This flexibility allows miners to use their existing hardware, which can be more economical and accessible than investing in new, specialized equipment.

Security

- **Enhanced Network Security**: By incorporating GPU mining, BTQ ensures that its network benefits from the substantial computational power that GPUs can provide. This enhances the security of the network by making it more resistant to attacks that aim to undermine its integrity.

3.3.3 Benefits of Dual Mining Approach

Inclusive Mining Ecosystem

- **Encourages Participation:** By using both RandomX for CPUs and Cryptonight cn/0 for GPUs, BTQ creates an inclusive mining ecosystem where a diverse range of participants can contribute. This inclusivity is vital for the health and sustainability of the network.

Reduced Centralization Risk

- **Balanced Power Distribution:** The dual mining approach helps to prevent the centralization of mining power. With both CPU and GPU mining, the likelihood of a few entities dominating the mining process is reduced, promoting a fairer and more distributed network.

Robust Security

- **Quantum-Resistant:** Combining two different mining algorithms enhances the security of the BTQ network. Each algorithm has its strengths, and together they provide a robust defense against potential threats, including those posed by the advent of quantum computing.

3.3.4 Conclusion

The dual mining mechanism of BTQ, leveraging RandomX for CPU mining and Cryptonight cn/0 for GPU mining, exemplifies a forward-thinking approach to blockchain security and decentralization. By enabling broad participation and reducing the risk of centralization, BTQ ensures a secure, efficient, and inclusive mining process. This innovative strategy not only strengthens the network's resilience but also aligns with the core principles of blockchain technology: decentralization and democratization of financial power. As BTQ continues to evolve, its dual mining mechanism will undoubtedly play a pivotal role in its success and adoption in the cryptocurrency landscape.

3.4 Decentralized Network Architecture

Bitcoin Quantum (BTQ) stands out in the blockchain landscape with its robust and secure decentralized network architecture. By leveraging a distributed network of nodes, BTQ ensures the integrity and security of its blockchain, mitigating the risks associated with central points of failure and enhancing overall network resilience. Here's an in-depth look at how BTQ's decentralized network architecture works and its benefits.

3.4.1 Distributed Network of Nodes

At the core of BTQ's decentralized architecture is its extensive network of nodes:

Node Diversity

- **Global Distribution:** BTQ nodes are spread across various geographic locations worldwide. This global distribution ensures that the network remains operational and secure, even if some nodes are compromised or offline due to regional issues.
- **Independent Operation:** Each node operates independently, validating transactions and maintaining a copy of the blockchain. This independence prevents any single entity from controlling the network, fostering decentralization.

Redundancy and Availability

- **Continuous Operation:** The decentralized nature of the node network ensures continuous operation of the BTQ blockchain. Even if multiple nodes fail, the network remains functional, as other nodes can seamlessly take over the responsibilities.
- **Fault Tolerance:** Redundancy within the network provides fault tolerance, ensuring that no single point of failure can disrupt the blockchain. This design significantly enhances the

reliability and availability of the BTQ network.

3.4.2 Avoiding Central Points of Failure

Centralized systems are often vulnerable to attacks and failures due to their reliance on a single point of control. BTQ's decentralized network architecture addresses these vulnerabilities effectively:

Enhanced Security

- **Distributed Control:** By distributing control across numerous nodes, BTQ eliminates the risk associated with centralized control points. This makes it considerably more difficult for attackers to compromise the network, as they would need to control a majority of the nodes simultaneously.
- **Attack Resistance:** The decentralized architecture enhances the network's resistance to various types of attacks, including Distributed Denial of Service (DDoS) attacks. Even if some nodes are targeted, the network remains resilient and operational.

Data Integrity

- **Consensus Mechanism:** BTQ utilizes a consensus mechanism to validate transactions across the network. This ensures that all nodes agree on the state of the blockchain, maintaining data integrity and preventing unauthorized changes.
- **Immutable Ledger:** The decentralized nature of the network ensures that once data is recorded on the blockchain, it cannot be altered or deleted. This immutability is crucial for maintaining trust and transparency within the BTQ ecosystem.

3.4.3 Increasing Overall Network Robustness

BTQ's decentralized network architecture not only enhances security but also contributes to the robustness and scalability of the network:

Scalability

- **Efficient Resource Utilization:** Decentralization allows for efficient utilization of computational resources across the network. As the number of nodes increases, the network can handle more transactions and maintain high performance.
- **Adaptive Growth:** BTQ’s network can adapt to increasing demand by adding more nodes, ensuring scalability without compromising security or performance.

Resilience

- **Dynamic Adaptation:** The network can dynamically adapt to changes and disruptions, ensuring continuous operation and reliability. Nodes can join or leave the network without affecting its overall functionality.
- **Community Participation:** Decentralization encourages participation from a diverse community of users and miners, fostering a collaborative environment that supports the network's growth and sustainability.

3.4.4 Conclusion

Bitcoin Quantum’s (BTQ) decentralized network architecture is a testament to its commitment to security, integrity, and robustness. By leveraging a distributed network of nodes, BTQ eliminates central points of failure, enhances resistance to attacks, and ensures continuous operation and scalability. This architecture not only fortifies the network against potential threats but also promotes a more inclusive and resilient blockchain ecosystem. As the blockchain landscape evolves, BTQ’s decentralized approach will continue to play a pivotal role in its success and adoption, providing users with a secure and reliable platform for their transactions and data.

3.5 Multi-Layer Encryption in BTQ: Ensuring Comprehensive Security

In the evolving landscape of cybersecurity, Bitcoin Quantum (BTQ) stands out by employing a sophisticated multi-layer encryption strategy. This approach protects user data and transactions from potential threats, ensuring that the integrity and confidentiality of information are maintained even if one encryption layer is compromised. Here's an in-depth look at how BTQ's multi-layer encryption works and its benefits.

3.5.1 The Concept of Multi-Layer Encryption

Multi-layer encryption involves using multiple encryption methods and layers to secure data. This strategy ensures that even if one layer is breached, additional layers continue to protect the information, providing a robust defense against unauthorized access and attacks.

Layered Security Approach

- **Redundancy:** Each layer of encryption acts as a backup to the others. If an attacker manages to breach one layer, they are faced with additional layers of encryption that protect the data.
- **Enhanced Protection:** The use of multiple encryption layers enhances the overall security of the system, making it significantly harder for attackers to access sensitive information.

Sequential Encryption

- **Complexity:** By encrypting data sequentially with different algorithms, BTQ increases the complexity of breaking through the encryption. Each layer adds an additional level of security, making the task of decrypting the data without authorization increasingly difficult.

3.5.2 Implementation in BTQ

BTQ’s multi-layer encryption is designed to protect user data and transactions comprehensively. Here’s how it is implemented:

Data Encryption

- **First Layer:** The initial layer encrypts the user data and transactions using a strong symmetric encryption algorithm. This ensures that data is secure during transmission and storage.
- **Subsequent Layers:** Additional layers of encryption are applied using different algorithms. Each layer is independent, meaning that compromising one does not affect the integrity of the others.

Transaction Security

- **End-to-End Encryption:** BTQ ensures that all transactions are end-to-end encrypted. This means that data is encrypted on the sender's side and only decrypted on the receiver's side, ensuring that it remains secure during the entire transmission process.
- **Encryption at Rest and in Transit:** Data is encrypted both when it is stored (at rest) and when it is transmitted over the network (in transit). This dual approach ensures comprehensive protection at all times.

User Data Protection

- **Layered Access Control:** BTQ employs layered access control mechanisms to ensure that only authorized users can access sensitive data. This involves using encryption keys that are securely managed and distributed.
- **Regular Key Rotation:** To further enhance security, BTQ implements regular key rotation policies. This means that encryption keys are periodically changed, reducing the risk of key compromise.

3.5.3 Benefits of Multi-Layer Encryption

Enhanced Security:

- **Increased Resistance to Attacks:** Multiple layers of encryption increase the resistance to various types of attacks, including brute force attacks and cryptographic breaches. Each layer adds complexity, making it significantly harder for attackers to decrypt the data.

- **Protection Against Quantum Threats:** As quantum computing evolves, traditional encryption methods may become vulnerable. BTQ's multi-layer approach incorporates quantum-resistant algorithms, ensuring long-term security.

Data Integrity

- **Tamper-Proof:** Multi-layer encryption ensures that any attempt to tamper with the data is immediately detectable. Each layer verifies the integrity of the data, ensuring that it has not been altered.
- **Secure Transactions:** By securing transactions with multiple encryption layers, BTQ guarantees the authenticity and integrity of every transaction. This ensures that transactions cannot be altered or forged.

User Privacy

- **Confidentiality:** Multi-layer encryption ensures that user data remains confidential and protected from unauthorized access. This is particularly important for maintaining user trust and complying with data protection regulations.
- **Anonymity:** By encrypting user data at multiple levels, BTQ enhances user anonymity, ensuring that personal information is kept private.

3.5.4 Conclusion

Bitcoin Quantum's (BTQ) multi-layer encryption strategy represents a significant advancement in the field of blockchain security. By employing multiple layers of encryption, BTQ ensures that user data and transactions are protected from potential threats, providing a robust and resilient security framework. This layered approach not only enhances the security and integrity of the blockchain but also fosters trust and confidence among users. As cybersecurity threats continue to evolve, BTQ’s commitment to multi-layer encryption will remain a cornerstone of its security strategy, ensuring comprehensive protection for its users.

3.6 BTQ Emission Plan: Ensuring Long-Term Economic Stability

Bitcoin Quantum (BTQ) introduces a carefully crafted emission plan designed to promote long-term economic stability and sustainable growth. By mirroring certain aspects of Bitcoin's successful model and introducing unique elements, BTQ aims to maintain its value proposition while ensuring a stable market environment. Here's an in-depth look at BTQ's emission plan and its key features.

3.6.1 Total Supply

One of the foundational elements of BTQ's emission plan is its fixed total supply:

21 Million Units

- **Scarcity:** Similar to Bitcoin, BTQ has a total supply limit of 21 million units. This fixed supply ensures that BTQ remains a scarce asset, which supports its value proposition and provides a hedge against inflation.
- **Value Preservation:** The finite supply model is designed to preserve the value of BTQ over time, as the limited availability of coins enhances their worth as demand increases.

3.6.2 Long-Term Emission Plan

BTQ's emission plan is structured to span an impressive duration of approximately 1000 years, fostering a stable and sustainable economic environment:

Gradual Block Reward Reduction

- **5% Reduction Every 525,600 Blocks:** The emission plan includes a block reward reduction of 5% every 525,600 blocks (roughly every four years). This gradual reduction minimizes the economic impact of new coin issuance, preventing sudden shifts in supply that could lead to volatility.
- **Sustainable Growth:** By spreading the coin issuance over a millennium, BTQ ensures a steady and predictable supply of new coins, supporting long-term growth and stability in the cryptocurrency market.

Minimized Economic Impact

- **Reduced Volatility:** The slow and steady reduction in block rewards helps to mitigate the potential for drastic market fluctuations. This approach supports a more stable market environment, encouraging long-term investment and participation.
- **Balanced Incentives:** The emission plan balances the need to incentivize miners with the goal of maintaining a stable and sustainable economy. As block rewards decrease gradually, miners can adjust their strategies without facing abrupt changes.

3.6.3 Predictable Block Rewards

Transparency and predictability are crucial components of BTQ's emission plan:

Framework for Planning

- **Predictable Rewards:** The emission plan provides a clear and predictable framework for block rewards, allowing miners and investors to plan for the long term. This predictability fosters confidence in BTQ's future, as stakeholders can make informed decisions based on a known reward schedule.
- **Economic Stability:** A predictable emission schedule contributes to economic stability by

reducing uncertainty and speculation. Investors and miners can rely on the consistent reduction in block rewards, supporting a stable investment environment.

Transparency

- **Open Emission Plan:** BTQ's emission plan is transparent and publicly available, ensuring that all participants have access to the same information. This openness promotes trust and accountability within the BTQ ecosystem.
- **Community Confidence:** By providing a clear roadmap for coin issuance, BTQ enhances community confidence and encourages long-term commitment from both miners and investors.

3.6.4 Conclusion

Bitcoin Quantum's (BTQ) emission plan is a testament to its commitment to long-term economic stability and sustainable growth. By incorporating a fixed total supply of 21 million units, a gradual block reward reduction over 1000 years, and a predictable reward framework, BTQ ensures that its currency remains scarce, valuable, and stable. This well-designed emission plan not only supports BTQ's value proposition but also fosters confidence among miners and investors, laying the foundation for a resilient and thriving cryptocurrency ecosystem. As BTQ continues to evolve, its emission plan will play a pivotal role in maintaining its economic stability and market appeal.

3.7 Lamport-Diffie One-Time Signature

3.7.1 Introduction

In 1979, Lamport described a hash-based one-time signature for a message of length m bits (usually the output of a collision-resistant hash function). This method aims to ensure high security in signing and authenticating messages.

3.7.2 Key Pair Generation

The key pair generation process involves creating m pairs of random secret keys, each pair having the form: $\mathrm{sk}_j^m \in \{0,1\}^n$ where $j \in \{0,1\}$. Specifically, the private key will be: $\mathrm{sk} = ((\mathrm{sk}_0^1, \mathrm{sk}_1^1), \ldots, (\mathrm{sk}_0^m, \mathrm{sk}_1^m))$. A one-way hash function $f : \{0,1\}^n \to \{0,1\}^n$ will be used to generate the pairs of public keys: $\mathrm{pk}_j^m = f(\mathrm{sk}_j^m)$. Thus, the public key will be: $\mathrm{pk} = ((\mathrm{pk}_0^1, \mathrm{pk}_1^1), \ldots, (\mathrm{pk}_0^m, \mathrm{pk}_1^m))$

3.7.3 Signing the Message

To sign a message, we perform a bitwise inspection of the message's hash to select sk_j. Specifically, if $bit = 0$, we select sk_0; if $bit = 1$, we select sk_1. The signature is created as follows:

$$s = (\mathrm{sk}_j^1, \ldots, \mathrm{sk}_j^m)$$

This process reveals half of the private key.

3.7.4 Verifying the Signature

To verify the signature, we inspect the bits of the message's hash (with $j \in \{0,1\}$) and ensure that:

$$\mathrm{pk}_j^m = f(\mathrm{sk}_j^m)$$

3.7.5 Security After Grover's Algorithm

To ensure 128-bit security after using Grover's algorithm, we assume the message length is a fixed hash output from SHA256, with $m = 256$ and $n = 256$. This results in the public and private keys having a length of: $\mathrm{pk} = \mathrm{sk} = 16\mathrm{kb}$

The signature for each use will have a length of: $8\mathrm{kb}$

3.7.6 Disadvantages

Lamport signatures should be used only once and can be generated very quickly. However, they have some disadvantages, such as large key sizes, large signature sizes, and consequently large transaction sizes, making them impractical for a public blockchain ledger.

3.7.7 Conclusion

The Lamport-Diffie one-time signature method offers a high level of security but faces challenges regarding data size, making it difficult to apply in practice, especially in public blockchain systems. However, for applications requiring absolute security and not constrained by data size, this method remains valuable.

3.8 Winternitz One-Time Signature (WOTS)

3.8.1 Introduction

The Winternitz One-Time Signature (WOTS) is an enhancement of the Lamport-Diffie signature, designed to provide a more efficient way to generate cryptographic signatures. By leveraging an iterating hash function and a Winternitz parameter, WOTS achieves smaller key sizes and signatures at the cost of increased computational effort.

3.8.2 Key Pair Generation

For a message digest M of length m bits, with secret and public keys of length n bits, and a one-way function $f : \{0,1\}^n \rightarrow \{0,1\}^n$, the WOTS scheme uses a Winternitz parameter $w \in \mathbb{N}$ where $w > 1$. The key pair generation process involves creating chains of hashes from random secret keys. The secret keys are:

$$\mathrm{sk} = (\mathrm{sk}_1, \ldots, \mathrm{sk}_{m/w})$$

Each secret key sk_x is hashed $w - 1$ times to generate the corresponding public key pk_x:

$$\mathrm{pk}_x = f^{2^{w-1}}(\mathrm{sk}_x)$$

Thus, the public key is:

$$\mathrm{pk} = (\mathrm{pk}_1, \ldots, \mathrm{pk}_{m/w})$$

3.8.3 Signing the Message

Unlike the bitwise inspection of the message digest in Lamport signatures, WOTS parses the message digest Mw bits at a time to extract a number $i \in \mathbb{N}$ where $i < 2^w - 1$. The signature for each segment is generated by hashing the corresponding secret key i times:

$s_x = f^i(\mathrm{sk}_x)$

The complete signature is:

$s = (s_1, \ldots, s_{m/w})$

3.8.4 Verification

To verify a signature, the verifier generates:

$$\mathrm{pk}_x = f^{2^{w-1}-i}(s_x)$$

from the message digest M and the signature s, then confirms that the calculated public keys match the originally provided public keys.

Example with SHA-256

Using SHA-256 as the one-way cryptographic hash function, with $m = 256$ and $n = 256$, and choosing $w = 8$:

- The size of the public key, private key, and signature is:

$$\frac{(m/w) \times n}{8} \text{ bytes} = 1 \text{ KB}$$

- To generate the public key, \(f \) needs to be iterated:

$$i = \frac{m}{w} \times 2^{w-1} = 8160$$

times per WOTS key pair generation.

For $w = 16$, the size of keys and signatures is halved, but the number of hash iterations becomes impractical: $i = 1048560$

3.8.5 Trade-offs

WOTS provides a trade-off between the size of the keys and signatures and the computational effort required to generate them. Increasing the Winternitz parameter \(w \) reduces the size of the keys and signatures but increases the computational workload due to the higher number of hash iterations needed.

3.8.6 Conclusion

The Winternitz One-Time Signature offers an efficient solution for cryptographic signatures with smaller key sizes and signatures compared to Lamport signatures. However, the trade-off is the increased computational effort required for key and signature generation. This method is particularly useful in scenarios where storage efficiency is crucial, and computational resources are readily available.

3.9 Winternitz One-Time Signature Plus (W-OTS+)

3.9.1 Introduction

Buchmann introduced a variant of the original Winternitz OTS by changing the iterating one-way function to instead be applied to a random number, x, repeatedly but this time parameterized by a key, k, which is generated from the previous iteration of $f_k(x)$. This method is strongly unforgeable under adaptive chosen message attacks when using a pseudo-random function (PRF) and a security proof can be computed for given parameters.

Huelsing introduced a further variant, W-OTS+, enabling the creation of smaller signatures for equivalent bit security through the addition of a bitmask XOR in the iterative chaining function. Another difference between W-OTS (2011 variant)/W-OTS+ and W-OTS is that the message is parsed $\log_2(w)$ bits at a time rather than w, decreasing hash function iterations but increasing key and signature sizes.

3.9.2 Description of W-OTS+

Key Pair Generation

With a security parameter $n \in \mathbb{N}$, corresponding to the length of message m, keys, and signature in bits, being determined by the cryptographic hash function chosen and the Winternitz parameter, $w \in \mathbb{N}$ where $w > 1$ (usually $4, 16$), the number of n bit string elements in a W-OTS+ key or signature, l, is computed as:

$$l = l_1 + l_2$$

where,

$$l_1 = \left\lceil \frac{m}{\log_2(w)} \right\rceil$$

$$l_2 = \left\lceil \frac{\log_2(l_1(w-1))}{\log_2(w)} \right\rceil + 1$$

A keyed hash function is used, $f_k : \{0,1\}^n \rightarrow \{0,1\}^n \mid k \in \{0,1\}^n$. In pseudocode:

$$f_k(M) = \text{Hash}(\text{Pad}(K)||\text{Pad}(M))$$

Where,

$$\text{Pad}(x) = (x||1||00\ldots0) \text{ for } |x| < b$$

The chaining function, $c_k^i(x, r)$, on input of $x \in \{0,1\}^n$, iteration counter i, key $k \in K$, and randomization elements, $r = (r_1, \ldots, r_j) \in \{0,1\}^{n*w}$, with $j \geq i$, is defined as follows:

$$c^i(x,r) = \begin{cases} x & \text{if } i = 0 \\ f_k(c_k^{i-1}(x,r) \oplus r_i) & \text{if } i > 0 \end{cases}$$

This process creates chains of hashes of length $w - 1$, ending with public keys.

Signature Key

To create the secret key, sk, $l + w - 1$ n bit strings are chosen uniformly at random (with PRF), of which the first l make up the secret key, $sk = (\mathrm{sk}_1, \ldots, \mathrm{sk}_l)$ and the remaining $w - 1$ n bit strings become $r = (r_1, \ldots, r_{w-1})$. A function key, k, is chosen uniformly at random.

Verification Key

The public key is:

$$pk = (\mathrm{pk}_0, \mathrm{pk}_1, \ldots, \mathrm{pk}_l) = ((r, k), c_k^{w-1}(\mathrm{sk}_1, r), c_k^{w-1}(\mathrm{sk}_2, r), \ldots, c_k^{w-1}(\mathrm{sk}_l, r))$$

Note that pk_0 contains r and k.

Signing

To perform a signature: message M, of length m, is parsed such that $M = (M_1, \ldots, M_{l_1})$, $M_i \in \{0, w - 1\}$ (creating a base-w representation of M).

Next, the checksum C, of length l_2, is calculated and appended:

$$C = \sum_{i=1}^{l_1} (w - 1 - M_i)$$

such that: $M + C = (b_0, \ldots, b_l)$. The signature is:

$$s = (s_1, \ldots, s_l) = (c_k^{b_1}(\mathrm{sk}_1, r), \ldots, c_k^{b_l}(\mathrm{sk}_l, r))$$

Verification

To verify a signature $b = (b_1, \ldots, b_l)$ is reconstructed from$M$.

If $pk = (c_k^{w-1-b_1}(s_1), \ldots, c_k^{w-1-b_l}(s_l))$, then the signature is valid.

W-OTS+ provides a security level of at least $n - w - 1 - 2\log(lw)$ bits. A typical signature where $w = 16$ using SHA-256 ($n = m = 256$) is ln bits or 2.1 KB.

3.10 Merkle Tree Signature Schemes for BTQ

The Merkle tree signature scheme offers a robust solution for enhancing the cryptographic security of digital signatures used in BTQ transactions. While one-time signatures (OTS) provide strong security for single-use scenarios, they face a significant limitation: each key pair can only be used safely once. This restriction can lead to inefficiencies and security risks in blockchain applications, where numerous transactions may need to be signed from a single address.

3.10.1 Extending the Signature Scheme with Merkle Trees

To address the limitations of OTS, the signature scheme can be extended to support multiple valid signatures for each ledger address. This is achieved by organizing the OTS key pairs into a binary hash tree, known as a Merkle tree. This structure allows for the efficient generation and verification of multiple signatures from a single root hash.

3.10.2 Binary Hash Tree Overview

A Merkle tree is an inverted tree structure where parent nodes are derived by hashing the concatenation of child sibling nodes. The tree's root hash provides a cryptographic proof of the integrity and existence of any node or leaf within the tree. In the context of BTQ, each leaf node represents a hash of a pre-generated OTS public key, and the tree's root hash acts as the public key for the entire set of OTS key pairs.

3.10.3 Example of a Merkle Tree Signature Scheme

In a simple form, a Merkle tree with a height of $h = 2$ can support four signatures. Each leaf node (layer 0) is a hash of an OTS public key, and parent nodes are hashes of concatenated child nodes. The root hash is derived from the topmost parent nodes and serves as the ledger address's public key.

The full signature S for a message M includes:

1. The OTS signature s.

2. The OTS key pair index n.

3. The path of hashes from the leaf to the root ($H2, H6, root$).

The validity of S is verified by recalculating the Merkle root from the signature path and comparing it with the stored public key.

3.10.4 Stateless Cryptographic Signatures

While the Merkle Signature Scheme (MSS) provides a stateful method for managing multiple signatures, the need to track used keys can complicate its implementation. A newer approach, such as the SPHINCS (Stateless Practical Hash-based Incredibly Nice Cryptographic Signature) scheme, offers stateless signatures with enhanced security, utilizing 128-bit security as reported in recent studies.

3.10.5 Hypertrees for Extended Capacity

A limitation of basic MSS is the finite number of signatures it can support, constrained by the number of pre-generated OTS key pairs. To overcome this, hypertrees—trees of Merkle trees—can be used. This structure extends the capacity by chaining multiple Merkle trees, enabling the generation of a significantly higher number of signatures.

For example, a hypertree composed of four chained Merkle trees with height $h = 5$ can support up to 2^{20} signatures, vastly increasing the signature capacity without proportional increases in computational overhead.

3.10.6 Conclusion

Implementing Merkle tree signature schemes in BTQ enhances the scalability and security of cryptographic signatures. By leveraging binary hash trees and hypertrees, BTQ can efficiently manage a large number of signatures while maintaining robust security. Additionally, exploring stateless cryptographic schemes like SPHINCS can further streamline and secure transaction processes in the BTQ network.

3.11 Proposed Signature Scheme for BTQ

3.11.1 Security Requirements

In the design of BTQ, ensuring the cryptographic security of the signature scheme against both classical and quantum computing attacks is paramount. XMSS (eXtended Merkle Signature Scheme) using SHA-256, where $w = 16$, offers 196-bit security. This level of security is predicted to remain robust against brute force computational attacks until the year 2164.

3.11.2 BTQ Signatures

BTQ proposes an extensible stateful asymmetrical hypertree signature scheme composed of chained XMSS trees. This approach has two primary benefits: it utilizes a validated signature scheme and enables the generation of ledger addresses that can sign transactions without the lengthy pre-computation delays associated with large XMSS constructions. W-OTS+ (Winternitz One-Time Signature Plus) is chosen as the hash-based one-time signature for the scheme due to its security and performance advantages.

3.11.3 Hypertree Construction

Key and Signature Sizes

As the number of trees within a hypertree increases, key and signature sizes grow linearly, while signature capacity rises exponentially. Here are the sizes for various XMSS tree-derived public keys and signatures, based on the 2011 description, with $w = 16$, $m = 256$, tree height h, and SHA-256 as the cryptographic hash algorithm:

- $h = 2$, 2^2 signatures: public key 0.59KB, signature 2.12KB (0.4s)

- $h = 5$, 2^5 signatures: public key 0.78KB, signature 2.21KB (0.6s)

- $h = 12$,2^{12} signatures: public key 1.23KB, signature 2.43KB (32s)

- $h = 20$, 2^{20} signatures: public key 1.7KB, signature 2.69KB (466s)

The trade-off for creating an XMSS hypertree (4 trees, $j = 3$, $h = 5$) with an eventual signature capacity of 2^{20} in less than 3 seconds, compared to 466 seconds, is an acceptable increase in signature size to 8.84KB from 2.69KB.

Asymmetry

Creating an asymmetrical tree allows early signatures to occur with a single XMSS tree construction, which can be extended as needed for later signatures, impacting overall signature capacity. This approach is likely inconsequential for a blockchain ledger application, as the wallet can offer users a choice between signature capacity and signature/key sizes. A maximum tree depth of $j = 2$ should be sufficient for all circumstances.

3.11.4 Conclusion

The proposed signature scheme for BTQ addresses the dual needs of security and efficiency by leveraging an extensible stateful asymmetrical hypertree signature scheme composed of chained XMSS trees. This approach ensures robust protection against quantum and classical attacks while maintaining practical performance for blockchain applications.

3.12 BTQ Hypertree Specification

For the standard hypertree construction in BTQ, the following default parameters are to be adopted:

- $j = 0$ ($j \in \{0 \leq x \leq 2\}$)

- $h = 12$ ($h \in \{1 \leq x \leq 14\}$)

- Upper bound of signatures possible: 2^{36}

- Minimum signature size: 2.21KB

- Maximum signature size: 7.65KB

This implies a single XMSS tree with $h = 12$ can generate 4096 signatures. This tree can be extended with additional trees up to $h = 14$ as required, though most users are unlikely to need additional trees.

3.12.1 Example BTQ Signature

Consider the most complex hypertree construction with $j = 2$ and $h = 14$. A signature for a transaction message m, where n is the OTS keypair position for each XMSS tree, would require the following components:

Signature tree, $j = 2$:

- OTS signature of m
- n
- Merkle authentication proof
- Merkle root of the signature tree

Certification tree, $j = 1$:

- OTS signature of the merkle root from the signature tree ($j = 2$)
- n
- Merkle authentication proof
- Merkle root

Original XMSS tree, $j = 0$:

- OTS signature of the merkle root ($j = 1$)

- n

- Merkle authentication proof

- Merkle root

Verification involves generating the OTS public key from m and the signature, then confirming that the supplied merkle authentication proof generates the signature tree merkle root. This becomes the message for the next OTS signature. From this, the next OTS public key is generated, and the supplied merkle authentication proof is used to recreate the certification tree merkle root. This process continues until the merkle root of the highest tree, the original XMSS tree ($j = 0$), is correctly generated.

3.12.2 OTS Public Keys and Merkle Roots

Notice that OTS public keys are not required for verifying the XMSS tree signature. The merkle root for each tree can be deduced and therefore omitted during hypertree signature verification if the sending ledger address is known. This address is a computed derivative of the merkle root for the highest XMSS certification tree ($j = 0$) within the BTQ signature.

3.12.3 Stateful Signature Scheme

As the signature scheme is stateful, the wallet implementation must retain and update \(n\) for each XMSS tree generated in the hypertree for a given address. This ensures the proper tracking and management of signature keys for secure transactions.

3.13 Pseudorandom Function (PRF)

In the context of BTQ, the PRF is used to generate pseudorandom values from a seed. This is achieved using HMAC DRBG (Deterministic Random Bit Generator based on HMAC), ensuring that the generated values are cryptographically secure and suitable for use in key generation processes within the XMSS (eXtended Merkle Signature Scheme) hypertree construction.

3.14 Deterministic Wallet

A deterministic wallet in BTQ uses a single SEED to generate a large XMSS tree, providing a prolonged period of usability for most users. The process involves using a secure source of entropy to generate the SEED, which is then passed through a secure PRF function. This function generates a set of pseudorandom keys that form the XMSS tree. One drawback of using the same XMSS tree is the user being confined to a single address, though public key exposure is not a concern with a Merkle Signature Scheme (MSS).

In traditional blockchain systems like Bitcoin or Ethereum, an address is derived from the associated public key, and a single private or public key can only create one address. However, in BTQ, an XMSS address is derived from the public key (PK), which contains the Merkle root and public SEED. If the SEED remains constant but the number of OTS keypairs used to compute the tree varies, the Merkle root will change accordingly. Thus, for every single addition or subtraction of an OTS keypair, the derived address will change.

This feature allows wallet and node software to generate numerous variations of the XMSS tree, extending or contracting it as required using the same initial SEED. This enables the creation of as many unique addresses as needed. Recording this information in a safe, stateful, and compact manner is computationally trivial.

3.14.1 Example of Usage

To illustrate, let's assume a user has a SEED from which they generate their initial XMSS tree with $h = 12$, allowing for 4096 signatures. Over time, as the user needs more addresses or wants to manage multiple identities, the wallet can adjust the number of OTS keypairs, effectively changing the

Merkle root and generating new addresses while maintaining the same SEED.

Key Points:

- **Seed Generation:** Secure entropy is used to generate a SEED.

- **PRF Function:** The SEED is passed through a secure PRF (HMAC DRBG) to generate pseudorandom keys.

- **XMSS Tree Construction:** These keys form the XMSS tree, enabling a large number of signatures.

- **Address Variation:** Adjusting the number of OTS keypairs changes the Merkle root, creating new addresses.

- **State Management:** The wallet retains and updates the state, including the number of OTS keypairs used, ensuring seamless address generation and management.

By leveraging these mechanisms, BTQ ensures robust, secure, and flexible address generation, catering to the diverse needs of its users while maintaining strong cryptographic protections.

3.15 Cryptocurrency Design Parameters

This section outlines the proposed design parameters for the BTQ ledger. The primary focus is to ensure that the public blockchain is highly secure against both classical and quantum computing attack vectors. These parameters are in a draft state and are subject to change.

3.15.1 Fees

Due to the larger transaction sizes in BTQ compared to other ledgers, a transaction fee must be paid with each transaction. The philosophy here is to avoid artificial fee markets, as seen in Bitcoin, which can undermine the openness of a public blockchain. Each transaction, if it pays a minimum fee, should be as valid as any other. The minimum fee will float and be set by market dynamics, with nodes/miners competitively setting the lower bound of fees. An absolute minimum value will be enforced at the protocol level. Miners will have the discretion to order transactions from the mempool for inclusion in a block.

3.15.2 Blocks

Block-Times

BTQ aims to have a block-time of 60 seconds. This decision is influenced by the need to balance transaction speed and network security. Unlike Bitcoin, which has a block-time of roughly 10 minutes, or Ethereum with 15-second blocks, BTQ's 60-second block-time aims to reduce the incidence of long delays between blocks while maintaining security and minimizing orphaned blocks.

Block-Rewards

Each new block will include a "coinbase" transaction containing a mining address. This address will receive a reward that is the sum of the coinbase reward and the total transaction fees within the block. The block reward is recalculated by the mining node for each block and follows a predefined coin emission schedule.

Blocksize

BTQ will implement an adaptive blocksize solution based on a model proposed by Bitpay. This model increases the blocksize as a multiple (x) of the median size (y) of the last z blocks, thereby preventing manipulation by miners. The calculation for maximum blocksize (b) is:

$$b = x \cdot y$$

3.15.3 Currency Unit and Denominations

BTQ will use a monetary token called the quantum (plural: quanta) as the base currency unit. Each quantum is divisible down to a smallest unit called the Shor. The denominations are as follows:

- 1 Quantum (plural: Quanta)

- 10^{-9} Quantum: Shor

Transaction fees and calculations will be conducted in Shor units.

3.15.4 Accounts

BTQ addresses are designed to be extensible and support a wide range of formats. The first three bytes of any address (descriptor) encode information describing the hash function, signature scheme, address format, and additional parameters.

A typical BTQ address looks like this:

`Q01070050d31c7f123995f097bc98209e9231d663dc26e06085df55dc2f6afe3c2cd62e8271a6bd`

Address Structure

BTQ addresses are structured as follows:

Address Structure

Name	Bytes	Count	Description
DESC	0 .. 2	3	Address Descriptor
DATA	3 .. N	??	N will depend on the address format

At the moment, only one address format is supported: sha256.2X. When using sha256.2X, a BTQ address is composed of 39 bytes. This format is used internally by any API or module in the project. For representational purposes (e.g., user interface, debugging, logs), the address may be represented as a hex string prefixed with "Q" (79 hexadecimal characters). This format is suitable for user-related purposes but will be rejected by the API.

Address Fields

Name	Bits	Count	Description
DESC	0 .. 2	3	Hash Function
HASH	3 .. 35	32	SHA-256(DESC + PK)
VERH	36 .. 40	4	SHA-256(DESC + HASH) (only last 4 bytes)

In Pythonic pseudocode, this structure can be represented as follows:

```
address = "Q" + DESC[:3] + HASH[:32] + VERH[:4]
```

By incorporating these design parameters, BTQ aims to ensure a secure, efficient, and user-friendly blockchain that can withstand future advancements in computational power, particularly quantum computing.

3.15.5 Descriptor

The descriptor for BTQ addresses encodes critical information about the address format, hash function, and signature scheme. The structure and parameters are as follows:

Descriptor Structure

Name	Bits	Count	Description

HF	0..3	4	Hash Function
SIG	4..7	4	Signature Scheme
P1	8..11	4	Parameters 1 (e.g., height)
P2	12..15	4	Address Format
P3	16..23	8	Parameters 2

In the case of using XMSS, the parameters are defined as follows:

XMSS Parameters

Name	Bits	Count	Description
HF	0..3	4	SHA-256, SHAKE128, SHAKE256
SIG	4..7	4	XMSS
P1	8..11	4	XMSS Height / 2
AF / P2	12..15	4	Address Format
P3	16..23	8	Not used

SIG - Signature Type

Value	Description

0	XMSS
1..15	Reserved - Future expansion

HF - Hash Function

Value	Description
0	SHA-256
1	SHAKE-128
2	SHAKE-256
3..15	Reserved - Future expansion

AF - Address Format

Value	Description
0	SHA256.2X
1..15	Reserved - Future expansion

This descriptor structure provides flexibility and future-proofing for BTQ addresses, allowing for the incorporation of new hash functions, signature schemes, and address formats as they are developed. By defining these parameters, BTQ ensures that its addresses are robust, secure, and adaptable to future advancements in cryptographic technology.

3.16 Coin Emission Schedule

BTQ adopts a coin emission schedule inspired by Bitcoin, characterized by a scarcity model with a fixed upper limit on the number of tokens that can be issued. This approach establishes a finite supply of 21 million quanta, aiming to prevent inflation and promote value stability over time.

3.16.1 Key Features of the Coin Emission Schedule

- **Fixed Supply:** The total coin supply is capped at 21×10^6 quanta, providing a predictable monetary base.

- **Initial Supply Reduction:** The initial supply, calculated from the genesis block, starts at the total minus any coins issued in the genesis block. This initial amount decreases exponentially, setting a foundation for scarcity and value retention.

- **Exponential Decay:** The coin emission follows an exponential decay model to gradually reduce the block reward. This method smooths out the reductions in coin issuance, unlike Bitcoin's abrupt halving events, which can lead to market volatility.

- **Mathematical Model:**

  - **Remaining Supply at Block t : $Z_t = Z_0 e^{-\lambda t}$, where:**

    - Z_0 is the initial coin supply.

    - λ is the decay constant.

    - t is the total number of blocks.

  - Decay Constant λ: Calculated using $\lambda = \frac{\ln Z_0}{t}$. This constant dictates the rate at which the coin supply decays over time.

  - **Block Reward Calculation:** The reward for each block, b, is determined by the difference in remaining supply between consecutive blocks: $b = Z_{t-1} - Z_t$.

Implementation Considerations

- **Long-Term Stability:** The emission schedule extends over approximately 200 years, ensuring that coin rewards and supply decrease predictably. This long timeline aims to stabilize the token's value and utility as a transaction medium and store of value.

- **Block Generation:** Blocks are generated every 60 seconds, facilitating consistent and timely transaction confirmations compared to Bitcoin’s 10-minute interval.

Implications for Miners and Users:

- **Miners:** Miners are incentivized to continue supporting the network long-term due to the predictable reward schedule, which balances the decreasing block reward with transaction fees.

- **Users:** Users benefit from a stable and predictable supply model that aims to mitigate abrupt price changes due to supply shocks, typical in systems with sudden halving events.

The emission model used in BTQ not only ensures a controlled supply of tokens but also incorporates features that may lead to a more stable economic environment for the cryptocurrency. This approach addresses some of the criticisms of Bitcoin’s model while aiming to maintain the advantages of a deflationary currency.

4. CONCLUSION

The BTQ (Bitcoin Quantum) whitepaper outlines a comprehensive framework for a next-generation cryptocurrency that addresses critical challenges in the current digital currency landscape. By leveraging advanced cryptographic techniques and a well-considered economic model, BTQ aims to offer a secure, scalable, and sustainable blockchain solution.

Key Highlights

1. Quantum Resistance: BTQ integrates quantum-resistant cryptographic algorithms, particularly the XMSS and W-OTS+ signature schemes, to safeguard against potential future quantum computing threats. This forward-thinking approach ensures the longevity and robustness of BTQ's security model.

2. Hybrid Mining: Combining CPU-friendly RandomX and GPU-optimized Cryptonight cn/0 algorithms, BTQ aims to decentralize mining, making it more accessible and less prone to centralization. This dual-mining approach balances security with inclusivity.

3. Adaptive Blocksize and Efficient Transaction Handling: BTQ's adaptive blocksize mechanism, inspired by Bitpay's proposal, ensures that the network can handle varying transaction loads efficiently without sacrificing performance or security. The 60-second block time strikes a balance between quick confirmations and network stability.

4. Predictable Coin Emission: The emission schedule features a smoothly exponential decay in block rewards, preventing the abrupt supply shocks seen in Bitcoin’s halving events. This design aims to provide long-term stability and predictability for both miners and users, supporting a sustainable economic model.

5. Extensible Address Structure: The versatile address format, designed for future expansions, allows BTQ to adapt to new cryptographic standards and address formats as they emerge. This adaptability ensures that BTQ can remain at the forefront of blockchain technology advancements.

6. Deterministic Wallets: By using a single SEED to generate large XMSS trees, BTQ offers a secure and efficient way to manage multiple addresses. This feature provides flexibility for users while maintaining a high level of security through stateful signature schemes.

Vision for the Future

BTQ aims to be more than just another cryptocurrency; it aspires to set new standards in blockchain security, efficiency, and inclusivity. By addressing both present and future challenges, BTQ seeks to foster a resilient ecosystem that can adapt to technological advancements and evolving user needs.

In summary, BTQ represents a significant leap forward in cryptocurrency design. Its innovative use of quantum-resistant cryptography, dual mining algorithms, and a forward-thinking economic model positions BTQ as a strong contender in the digital currency space. As the blockchain landscape continues to evolve, BTQ is well-equipped to offer a secure, sustainable, and inclusive platform for the next generation of digital finance.

5. REFERENCES

[1] D. Bernstein. Sphincs: practical stateless hash-based signatures. 2015.

[2] J Buchmann. On the security of the winternitz one-time signature scheme.

[3] J. Buchmann. Xmss – a practical forward secure signature scheme based on minimal security assumptions. 2011.

[4] V Buterin, 2013. *Ethereum whitepaper*.

[5] A. Hulsing. W-ots+ - shorter signatures for hash-based signature schemes. 2013

[6] A. Hulsing. Xmss: Extended hash-based signatures. 2015.

[7] A Karina. An efficient software implementation of the hash-based signature scheme mss and its variants. 2015.

[8] A. Lenstra. Selecting cryptographic key sizes. 2001.

[9] R. Merkle. A certified digital signature. CRYPTO, 435, 1989.

[10] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[11] S Pair. A simple, adaptive blocksize limit. 2016.

[12] Yonatan Sompolinsky. Accelerating bitcoin’s transaction processing fast money grows on trees, not

chains. 2014.

[13] A. Toshi. The birthday paradox. 2013

[14] SPHINCS: Practical Stateless Signatures. Cryptographic Study, 2015.

[15] “Merkle Tree Signature Schemes.” Research Paper.