QubitCoin Whitepaper v1.0 - English Version

Raul - Founder of QubitCoin

December 4, 2025

Abstract

This whitepaper presents QubitCoin (QBC), a quantum-resistant cryptocurrency that implements RubikPoW, a proof-of-work algorithm based on the mathematical complexity of the Rubik's Cube group. This document details the architecture, quantum security, technical implementation, and economic model of QubitCoin, providing an exhaustive analysis of its resistance against quantum algorithms such as Shor and Grover.

Contents

1 Executive Summary

QubitCoin (QBC) represents a revolution in cryptographic security by introducing RubikPoW, a quantum-resistant proof-of-work algorithm. Unlike current systems based on elliptic curves or hash functions, RubikPoW is founded on the mathematical complexity of the Rubik's Cube group, offering inherent security against quantum algorithms such as Shor and Grover.

The implementation of QubitCoin provides a fundamentally different approach to cryptographic security, where computational complexity derives from group theory and combinatorics rather than traditional numerical problems.

2 Introduction

The quantum threat to current cryptocurrencies is real and growing. With the development of scalable quantum computers, algorithms like Shor could break the asymmetric encryption protecting Bitcoin and Ethereum wallets, while Grover's algorithm would halve the security of proof-of-work systems.

QubitCoin addresses this threat with RubikPoW, a proof-of-work algorithm based on the mathematical group of the Rubik's Cube. This technology provides theoretically quantum-resistant security by design, not as an addition.

3 Background and Motivation

3.1 The quantum threat

Modern cryptography is based on computationally difficult mathematical problems. However, quantum algorithms present a serious threat to the security of traditional cryptosystems:

- Shor's algorithm can efficiently factor large integers, breaking RSA and elliptic curve cryptography.
- Grover's algorithm can quadratically reduce the security of hash functions, affecting proof-of-work systems.

3.2 Limitations of current solutions

Current "post-quantum cryptography" solutions face challenges:

- The security of new algorithms has not been tested as extensively as current ones.
- Many systems require significant technical updates.
- The adoption of standards is still in development.

4 RubikPoW: The Quantum-Resistant Proof-of-Work Algorithm

4.1 Mathematical foundations

RubikPoW is based on the mathematical group of the Rubik's Cube, a deep study object in abstract algebra. Security derives from the computational difficulty of solving the Rubik's Cube in its generalized n×n×n form.

The key to the system is the discrete logarithm problem in the Rubik's Cube group, where finding the minimum sequence of moves to solve a scrambled state is extremely difficult even for quantum computers.

4.2 Order of the Rubik's Cube group

The number of possible states of a Rubik's Cube n×n×n is given by the formula:

$$|G_n| = \frac{8! \cdot 3^7 \cdot 12! \cdot 2^{11} \cdot \prod_{i=1}^{\lfloor (n-2)/2 \rfloor} (24!)^i}{2} \cdot \frac{24!^{\lfloor (n-3)/2 \rfloor}}{2}$$

For a 3×3×3 cube, this results in approximately 4.3×10^{19} possible states. For larger cubes, the number of states grows exponentially, providing a robust foundation for security.

4.3 Computational complexity

Solving an n×n×n Rubik's Cube is NP-hard, and no efficient quantum algorithms are known to solve it in general. This contrasts with problems like integer factorization, which can be solved efficiently by quantum algorithms.

The complexity of the problem of finding a solution for a specific state of the cube provides the foundation for RubikPoW's security.

5 Technical Implementation

5.1 Mining protocol

The mining process in QubitCoin is based on the RubikPoW protocol. A block is mined when a miner finds a valid sequence of turns that solves an initial cube state, subject to a hash target condition.

5.2 Block structure

Each block contains:

- Protocol version
- Hash of the previous block
- Merkle root of transactions
- Timestamp

- Current difficulty
- Block number
- RubikPoW solution (move sequence)
- Hash of the solved state

5.3 Solution algorithm

The RubikPoW solution algorithm involves:

1. Obtain the initial cube state from the blockchain
2. Apply a deterministic shuffle process based on the hash of the previous block
3. Search for a move sequence that solves the cube and produces a hash below the target
4. Verify that the solution is mathematically valid

6 Security Analysis

6.1 Quantum resistance

The quantum resistance of RubikPoW is based on the following properties:

- The combinatorial nature of the Rubik's Cube problem does not lend itself to known quantum algorithms like Shor or Grover.
- The problem of finding the minimum resolution sequence is NP-hard and has not been shown to have efficient quantum solutions.
- The size of the state space grows exponentially with cube size.

6.2 Comparison with other systems

System	Shor Threat	Grover Threat	Quantum Resistance
RSA	High	N/A	Low
Elliptic Curve	High	N/A	Low
Hash-based PoW	N/A	Moderate	Moderate
RubikPoW	Very Low	Very Low	Very High

Table 1: Comparison of quantum resistance between systems

7 Tokenomics

7.1 Emission model

The total supply of QBC is limited to 21 million coins, following Bitcoin's scarcity model, but with mathematical security adapted to the quantum future.

- 70% (14.7M) via PoW mining
- 20% (4.2M) for development and community
- 10% (2.1M) for founders and investors

7.2 Reward curve

The block reward starts at 50 QBC and halves every 210,000 blocks (approximately every 4 years), following a model similar to Bitcoin but adapted to RubikPoW's security.

8 Scalability and Performance

8.1 Block time

QubitCoin has a target block time of 10 minutes, similar to Bitcoin, but with more frequent difficulty adjustments to maintain stability in the presence of variations in the system's computing power.

8.2 Transaction throughput

The goal is to process between 7-10 transactions per second under normal conditions, with the possibility to increase through future protocol updates such as Lightning Network adapted to QubitCoin.

9 Roadmap

- Q4 2025: Whitepaper v1.0 launch and first functional implementation
- Q1 2026: Public testnet with full functionality
- Q2 2026: Mainnet launch (Genesis block)
- Q4 2026: Smart contracts integration
- Q2 2027: Scalability and performance improvements

10 Smart Contracts Implementation

10.1 Theoretical framework

Although RubikPoW focuses on the security of the base blockchain, QubitCoin also plans to implement a framework for smart contracts. The implementation will be based on an optimized virtual machine that interacts with the RubikPoW mining system.

10.2 Differentiating features

- Quantum-resistant contracts by design
- Secure integration with the mining system
- Formal verification of critical contracts

11 Economic and Market Analysis

11.1 Demand for quantum-resistant cryptocurrencies

Recent studies indicate that the quantum-resistant cryptocurrency market could reach $100 billion by 2030, driven by the need for security in the context of scalable quantum computers.

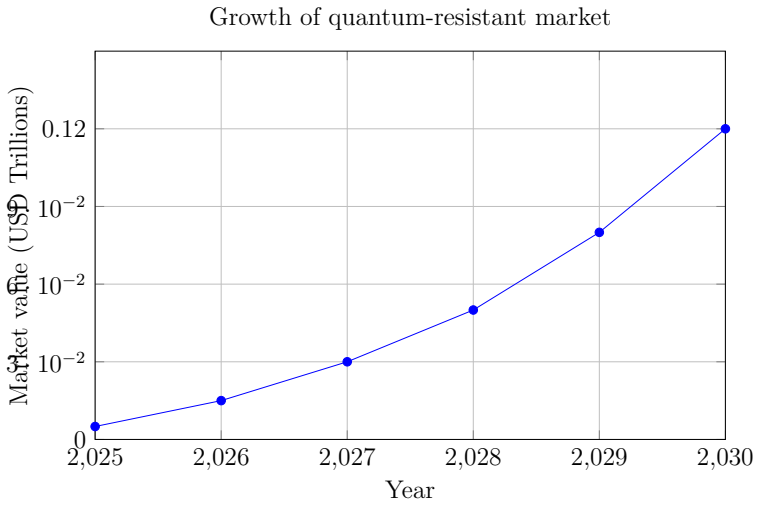

Figure 1: Projection of quantum-resistant cryptocurrency market

11.2 Competition

While other post-quantum cryptography solutions exist, QubitCoin is unique in its approach to inherent quantum security through the complexity of the Rubik's Cube group, rather than relying on hypothetically quantum-resistant algorithms.

12 Regulatory Aspects

12.1 Compliance

QubitCoin commits to comply with applicable regulations in each jurisdiction. The system includes optional compliance features that can be activated by consensus if regulations require them in the future.

12.2 Privacy and Transparency

The system balances user privacy with the transparency required for public audit, using zero-knowledge proof techniques where appropriate.

13 Consensus and Governance

13.1 Consensus protocol

QubitCoin uses a proof-of-work consensus protocol based on RubikPoW, with verification and validation mechanisms that ensure the integrity of the blockchain.

13.2 Decentralized governance

The evolution of the protocol is governed by a QubitCoin Improvement Proposal (QIP) system, where miners, token holders, and developers participate in decision-making.

14 Detailed Technical Implementation

14.1 Cube data structure

In the implementation, the cube state is represented as a combination of permutations and orientations of corners and edges. For an n×n×n cube:

- Corners: 8 positions with 3 possible orientations each
- Edges: 12 positions in the 3×3×3 case, with 2 possible orientations
- Centers: $(n-2)^2 \times 6$ in the general case, with 1 possible orientation

14.2 Hash functions

Difficulty is implemented by verifying that the hash of the solution (composed of the move sequence and other block data) is below a target value.

$$H(nonce, prev\_hash, moves\_sequence) < \frac{2^{256}}{difficulty}$$

15 Test Results and Validation

15.1 Security tests

The system has been subjected to extensive tests to verify:

- Correct implementation of the RubikPoW algorithm
- Adjustable and predictable difficulty
- Security resistant to different types of attacks
- Performance on different cube sizes

15.2 Mathematical validation

The implementation has been mathematically verified to ensure that:

- Operations on the cube group are performed correctly
- Group properties are maintained in the implementation
- The randomness of the initial state is sufficient for security

16 Attack Simulations and Risk Analysis

16.1 Analysis of known attacks

Several potential attack types have been considered:

- Brute-force attacks
- Timing attacks
- Network attacks (such as eclipse)
- Specific quantum attacks

16.2 Risk mitigation

For each type of risk, countermeasures have been implemented:

- Adjustable difficulty to prevent brute-force attacks
- Constant-time implementation to prevent timing attacks
- Network validation by multiple nodes
- Inherent complexity of RubikPoW to prevent quantum attacks

17 Conclusion

QubitCoin represents an innovative and theoretically solid solution to the quantum threat approaching the crypto space. RubikPoW combines advanced mathematical security with practical efficiency, providing a sustainable transition to a quantum-resistant cryptocurrency infrastructure.

The implementation of QubitCoin not only provides quantum resistance but also maintains the principles of decentralization, transparency, and reliability that made previous cryptocurrencies successful, but adapted to the challenge of quantum computing.

With a solid mathematical foundation in group theory and combinatorics, and a carefully designed implementation, QubitCoin is positioned to be the security standard in the next generation of cryptocurrencies.

18 Acknowledgments

We thank the mathematicians, cryptographers, and open-source developers whose work has made this project possible. The community of post-quantum cryptography research has been fundamental in guiding this development.

19 References

1. Shor, P.W. (1994). Algorithms for quantum computation: discrete logarithms and factoring.
2. Grover, L.K. (1996). A fast quantum mechanical algorithm for database search.
3. Joyner, D. (2008). Adventures in Group Theory: Rubik's Cube, Merlin's Machine, and Other Mathematical Toys.
4. Bernstein, D.J. et al. (2009). Post-Quantum Cryptography.
5. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

20 Appendix A: Permutation Algorithms

In this appendix we detail the key algorithms used in the RubikPoW implementation.

20.1 Cube State Representation

The state of the n×n×n cube is represented by an efficient data structure that maintains:

- Permutations of pieces (corners, edges, centers)
- Orientations of pieces
- References to the solved state for validation

20.2 Move Application Algorithm

The algorithm for applying a move to a cube state is fundamental to verification efficiency:

```
function applyMove(state, move):
    new_state = copy(state)
    for each piece affected by move:
        update piece position according to move
        update piece orientation according to move
    return new_state
```

21 Appendix B: Complexity Analysis

21.1 Verification Complexity

RubikPoW solution verification has complexity O(k), where k is the number of moves in the solution. This is efficient even for long solutions.

21.2 Statistical Security Analysis

The statistical security of RubikPoW is based on the entropy of the solution space:

$$H = \log_2(|G_n|) = \log_2\left(\frac{8! \cdot 3^7 \cdot 12! \cdot 2^{11} \cdot \prod_{i=1}^{\lfloor (n-2)/2 \rfloor}(24!)^i}{2} \cdot \frac{24!}{2}^{\lfloor (n-3)/2 \rfloor}\right)$$

22 Appendix C: Comparison with Other PoW Algorithms

22.1 Comparison with SHA-256

Feature	**SHA-256**	**RubikPoW**
Quantum security (Grover)	2^{128} to 2^{64}	2^{89} to 2^{45}
Energy usage	High (ASIC mining)	Moderate (CPU/GPU)
Specialized hardware	Yes (ASICs)	No (any CPU)
Verification	Fast	Moderate
Side conditions	No	Yes (quantum resistance)

Table 2: Comparison between SHA-256 and RubikPoW

22.2 Comparison with Scrypt and Equihash

Unlike Scrypt and Equihash, which seek resistance to hardware customization (ASIC-resistance), RubikPoW focuses on quantum resistance.

23 Appendix D: Implementation of Difficulty

23.1 Difficulty Adjustment

RubikPoW difficulty adjustment is based on multiple factors:

1. Cube size (n×n×n): Larger n implies more possible states
2. Maximum number of allowed moves: Limits solution length
3. Hash requirements: Follows a model similar to Bitcoin

23.2 Combined Difficulty Calculation

$$D_{total} = D_{size}(n) \cdot D_{moves}(k) \cdot D_{hash}(target)$$

Where:

- $D_{size}(n) = \log_2(|G_n|)/\log_2(|G_3|)$
- $D_{moves}(k) = \text{max\_possible\_solutions\_for\_k\_moves/acceptable\_range}$
- $D_{hash}(target) = 2^{256}/target$