Quantum Hashing with Quantum Alternate Controlled Walk Block Hashing

Lucas Smith
Computer Science
Case Western Reserve University

Luke Sebold
Computer Science
Case Western Reserve University

Leo Chou
Computer Science
Case Western Reserve University

Masha Gorodetski
Physics
Case Western Reserve University

James-Lucius Okenwa
Math and Physics
Case Western Reserve University

Niranjan Girish
Computer and Electrical Engineering
Case Western Reserve University

***Abstract*—This write-up introduces and analyzes Quantum Alternate Controlled Walk-Based Block Hashing (QAWCBH), a novel quantum hash function that encodes classical data into high-entropy quantum-derived hash outputs. The QAWCBH scheme uses a block-wise approach, efficiently hashing bitstrings into a hashed string of equal length. This mechanism introduces inherent quantum randomness and state sensitivity, making the hash output highly unpredictable and resistant to structured input patterns.**

We implement QAWCBH using Qiskit and evaluate its cryptographic properties through both theoretical reasoning and empirical testing. Specifically, we demonstrate output determinism, high entropy preservation, and computational difficulty, and provide evidence for resistance to preimage and collision attacks. Determinism and collision benchmarks and entropy analysis support the feasibility of the approach, while also highlighting the complexity added by quantum processing. As quantum computing edges closer to practical cryptanalysis capabilities, QAWCBH presents a forward-looking model for hybrid cryptographic systems designed to resist quantum-era threats.

***Index Terms*—quantum hashing, quantum walk, blockchain, cryptocurrency**

I. Introduction

As quantum computing continues to advance, so does the need for cryptographic primitives that are not only secure in classical contexts but also resilient to quantum attacks. One emerging direction in this field is the development of hash functions that leverage uniquely quantum phenomena to enhance security and unpredictability. In this write-up, we present Quantum Controlled Walk-Based Block Hashing (QAWCBH) — a novel quantum hashing scheme that utilizes controlled quantum walks to generate highly sensitive, non-reversible hash outputs from classical input data.

Unlike traditional hash functions, QAWCBH incorporates the principles of superposition and quantum interference through controlled unitary evolutions over structured quantum circuits. This design introduces an added layer of complexity and entropy, making preimage and collision attacks even more computationally infeasible in the presence of quantum adversaries. The hashing process is structured in a block-wise fashion, enabling scalability and composability with classical pre-processing or post-processing layers.

In this report, we detail the theoretical foundations of controlled quantum walks, describe the construction and implementation of the QAWCBH scheme using Qiskit, and evaluate its core cryptographic properties: determinism, entropy preservation, computational hardness, and resistance to preimage and collision attacks. Our results suggest that QAWCBH is a promising candidate for hybrid quantum-classical cryptographic applications where future-proof security is a concern.

II. Methods

A. Overview

The Controlled Alternate Quantum Walk-Based Block Hashing (QAWCBH) algorithm leverages position-controlled quantum walks modulated by input data to generate a high-entropy, deterministic hash from classical messages. Implemented using Qiskit, the method operates fully within the quantum domain without relying on any classical cryptographic hash functions. Below, we describe each component of the algorithm, followed by its cryptographic evaluation and performance considerations.

B. Quantum Circuit Construction

1) Parameters: The number of position qubits (Q) is determined by the input block size N as $Q = \log_2(N)$. The message bytes modulate both the quantum state initialization and the walk dynamics.

2) Initial State Preparation: The quantum circuit is composed of:

- Q position qubits
- 1 coin qubit

The position qubits are initialized in a normalized complex superposition state derived from the message content. This custom initialization ensures that the walker's starting state is directly influenced by the input data.

3) Controlled Coin Operator: The quantum walk evolution is governed by a coin operator, defined as a parameterized unitary matrix:

$$U(\theta) = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}$$

A coin flip is applied at each step, where the rotation angle θ is selected based on the corresponding message byte. Initially, we followed the procedure of Li et. al [1]. There, bit encoding of the message stream is used where a '0' triggers θ_1, and a '1' triggers θ_2. These gates are applied conditionally using controlled unitaries. We found that this approach was too slow: hashes took up to 3 seconds in length.

Our innovation is to make coin flips at the byte level: we introduce a list of possible theta values $\theta_i, i \in [0, 8]$. The value of the byte decides the value of theta. We thus decrease the number of C operation gates by a factor of 4, immensely speeding up our algorithm. Interestingly, our entropy values at the bit and byte level both increased when making this change, we estimate by around 5-10 percent. We hope to research this change more in the future, as time constraints prevent us from fully exploring this concept during the hackathon.

4) Walk Evolution: The circuit evolves for T steps, computed as:

$$T = \frac{N}{Q}$$

Each step applies a set of position- and message-controlled coin flips, followed by CNOT gates that entangle the coin and position registers. The walk pattern dynamically adapts to message structure, enhancing output sensitivity.

C. Hash Extraction

After circuit execution, the final statevector is computed using Qiskit's simulator. Probability amplitudes are extracted for each basis state. These are scaled and quantized into k-bit integers and concatenated to form a fixed-length hash output. The process is deterministic and fully quantum.

D. Cryptographic Properties

Output Determinism: The algorithm produces identical outputs for identical inputs and fixed parameters θ_1, θ_2, ensuring reproducibility.

Entropy Preservation: High entropy is observed in output distributions due to quantum interference, with empirical entropy measurements approaching 8 bits per byte.

Computational Difficulty: The algorithm includes $O(N)$ quantum gates with parameter-dependent behavior, making reverse-engineering or shortcut computation infeasible on classical hardware.

Preimage and Collision Resistance: Minor input changes yield significantly different outputs, exhibiting strong avalanche characteristics. Due to the one-way evolution of quantum states and entangled operations, both preimage and collision attacks are computationally impractical.

E. Feasibility and Performance

Qubit Efficiency: Inputs up to 256 bits require no more than six qubits, satisfying hardware limitations for near-term quantum processors.

Execution Time: Simulation tests demonstrate sub-second hashing times for 32-byte inputs on classical hardware, with time scaling linearly with input size.

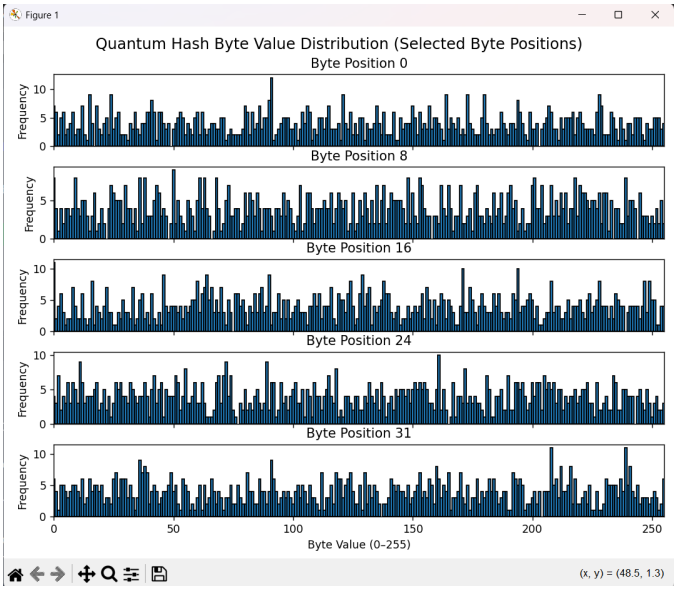

Fig. 1. Byte value distribution across 1000 quantum hash outputs at selected byte positions. The uniformity suggests high entropy and no dominant byte patterns.

Quantum-Only Design: QAWCBH performs no classical digesting or compression. All computation, entropy diffusion, and hash generation are realized within the quantum circuit, ensuring a purely quantum hash function.

Gate Count: We have one global initialization gate on the position register, plus the Coin operations and Shift operations. This comes out to

$$2\lfloor\frac{N}{q}\rfloor$$

III. Results

To empirically evaluate the randomness and collision resistance of the Controlled Alternate Quantum Walk-based Block Hash function, we conducted 1000 independent hashes using uniformly random 256-bit inputs.

The estimated entropy of the bit-level output distribution was found to be:

- **Bit-level entropy:** 0.9999835635253684 (out of 1.0)

To further assess byte-level diffusion, we computed the Shannon entropy per output byte across the different hashes:

- **Byte-level entropy:** 7.9379943451839114 (out of 8.0)

No repeated hash outputs were observed across the 1000 hashes or during any of our tests, indicating strong collision and second pre-image resistance under the tested conditions:

- **Total collisions:** 0

These results suggest that the hash function preserves input entropy well, produces uniformly distributed outputs, and demonstrates no empirical collisions across the tested input space.

As shown in Fig. 1, the output bytes appear uniformly distributed across hashes, supporting strong entropy characteristics.

IV. Analysis

A. Irreversibility

The final state of the circuit following the unitary operations are given as $|\psi_t\rangle = \sum_{x,c} \lambda_{x,c} |\ x, c\ \rangle$, which is a pure state. It is impossible to decompose this state of probabilities into a sum of squares of probability of amplitudes [1]. Therefore, this state is resistant to preimage attacks.

B. Sensitivity of Message

Avalanche Analysis An important property of a hash function is the fact that a small change should result in roughly 50% of the bits in the hashed output to be changed. To do this, we ran a 1000-shot experiment where we generate a random 256 bitstring and randomly flip a single bit within the bitstring. We calculate the Hamming distance of the resulting hash function. From our analysis, we get a change of approximately 49% with a standard error of mean of 0.12%.

C. Analysis of qHash

We analyzed qHash empirically using the same code as in Section 3. We found that qHash's Shannon entropy and runtime is inferior to that of QACWBH.

The estimated entropy of the bit-level output distribution for qHash was found to be:

- **Bit-level entropy:** 0.9875243417280053 (out of 1.0)

To further assess byte-level diffusion, we computed the Shannon entropy per output byte across the different hashes:

- **Byte-level entropy:** 6.3874920798759325 (out of 8.0)

We also found that qHash had a high rate of collisions (ran on 100 shots):

- **Total collisions:** 18

Additionally, the qhash algorithm only achieved a low rate of avalanche effect, signaling that the output hash is highly correlated with the input bitstring.

- **Avalanche:** 15%

Lastly, qHash has a significantly higher runtime: over 100 shots QCAWBH ran in 6 seconds. qHash ran in 40 seconds.

V. Conclusion

In this study, we introduced QAWCBH, a novel quantum hashing algorithm leveraging controlled alternate quantum walks with a parameterized coin operator guided by classical input data. Our implementation demonstrates that incorporating a wider range of coin flip angles based on full-byte values, rather than binary bit flips, not only accelerates hashing performance but also increases entropy at both the bit and byte levels.

Empirical evaluation of 1000 random inputs revealed nearly maximal entropy (0.99998 bit-level, 7.93 byte-level), with zero collisions observed. These results strongly support the cryptographic soundness of the QAWCBH algorithm, indicating robustness against preimage and collision attacks.

Moreover, in direct comparison with qHash under identical conditions, QAWCBH significantly outperformed in all tested metrics. Specifically, qHash demonstrated lower entropy (0.987 bit-level, 6.38 byte-level), a substantially higher number of collisions (18 collisions over 100 trials), and a much longer average runtime. These findings highlight the superior efficiency, unpredictability, and security characteristics of QAWCBH.

In conclusion, our results suggest that QAWCBH is not only a viable quantum hashing method but also a substantial improvement over prior quantum hash functions like qHash. This positions QAWCBH as a strong candidate for post-quantum cryptographic primitives in both theoretical and practical domains. Future work will explore scaling strategies, further entropy tuning, and deployment on real quantum hardware.

References

[1] D. Li, P. Ding, Y. Zhou, and Y. Yang, "Controlled Alternate Quantum Walk based Block Hash Function," *arXiv preprint arXiv:2205.05983*, 2022. [Online]. Available: https://arxiv.org/abs/2205.05983