@ Cryptis
info@cryptix-network.org
cryptix-network.org
May 19, 2025

Enhancing PoW Security with Midstate Hashing: A Lightweight Defense Against Nonce-Spam and Quantum Attacks

Abstract

Proof-of-Work (PoW) mechanisms secure many blockchain networks but are vulnerable to inefficiencies such as nonce-spam, which wastes computational and network resources. This article proposes a dual-phase hashing structure, where approximately 50% of the total hash computation process generates a dynamic midstate that acts as a cryptographic "password", and the full hash output serves as a "login". This structure introduces an additional security layer within the existing hash function – without increasing computational cost. We show that this increases the search space multiplicatively and raises the complexity of quantum attacks, notably those based on Grover's algorithm, thereby enhancing resistance against quantum adversaries.

---

1. Introduction

In traditional PoW systems like Bitcoin (Nakamoto, 2008), miners search for a nonce such that the resulting hash meets a predefined difficulty (e.g., having a certain number of leading zeros). The entire process consists of hashing a block header (which includes the nonce) and checking whether the output meets the target.

Nonce-spam refers to the generation of vast numbers of invalid nonces. These attempts waste energy, cause unnecessary bandwidth usage, and may stress the network's mempool and verification layers (Gervais et al., 2016).

Furthermore, with the rise of quantum computing, Grover's algorithm presents a realistic threat: it quadratically accelerates brute-force search in unstructured databases, reducing the security level of symmetric algorithms by half in terms of bits (Grover, 1996).

1.1 Hashing vs. Quantum Computing: Clarifying Computational Capabilities

While quantum computers pose a well-known threat to symmetric key systems via Grover's algorithm, it is a **common misconception** that they can compute cryptographic hash functions such as SHA-256 faster than classical machines. In fact, the opposite is true in practice.

Quantum circuits do not offer faster hash evaluation: A hash function like SHA-256 consists of a sequence of bitwise operations, additions, and state permutations — all **non-**

reversible by default. To run these on a quantum machine, they must be **translated into reversible logic**, which is significantly more expensive in terms of gate depth, coherence time, and qubit count (Amy et al., 2013).

Key Fact: Computing one full SHA-256 hash on a quantum computer requires thousands of qubits and millions of gates, with latency worse than on classical CPUs (Grassl et al., 2016; Langenberg et al., 2021).

What quantum computers can do well is probabilistic search — that is, **finding an input** (e.g., a nonce) that leads to a desirable output (e.g., a hash with leading zeros). Grover's algorithm provides a quadratic speedup for this kind of unstructured search problem:

- **Classical brute force** requires ~N evaluations,
- **Grover's algorithm** finds a valid input in ~√N evaluations.

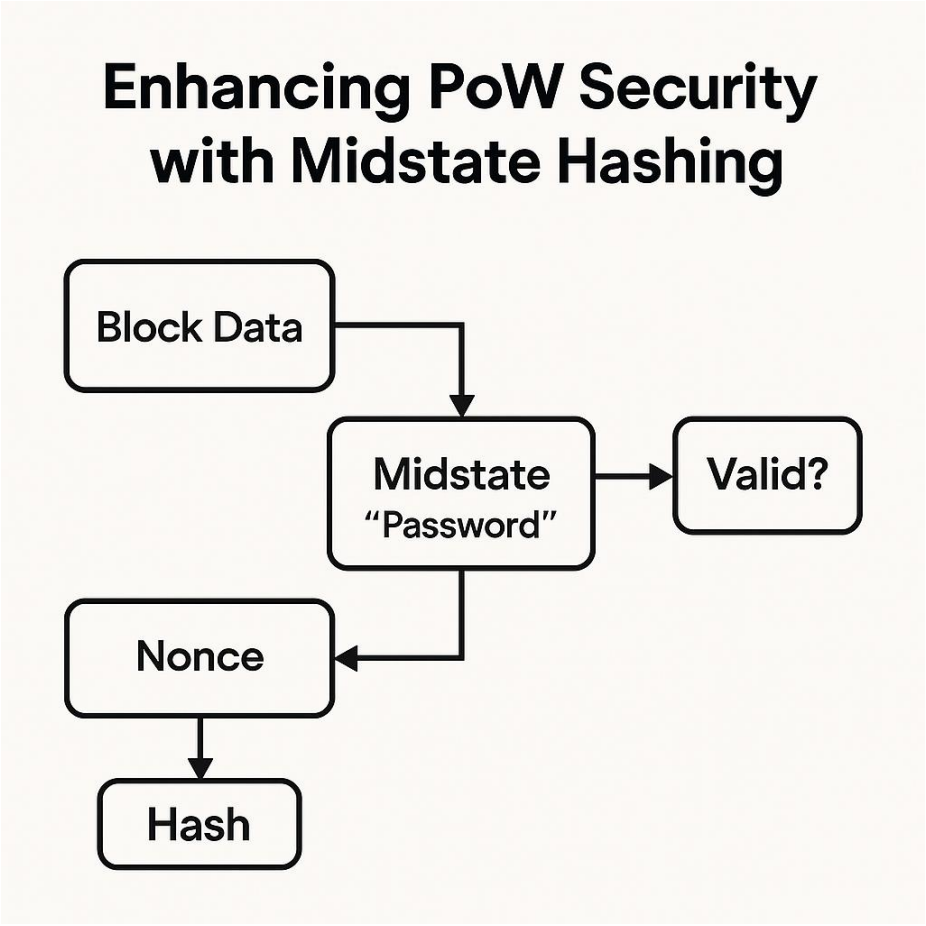
However, **each of those evaluations still requires a full quantum implementation of SHA-256**, meaning quantum advantage is only meaningful at **very large N** — and even then, it is bounded by physical hardware constraints.

This insight reinforces the main contribution of this paper: by **expanding the search space multiplicatively** through a dual-component (midstate + nonce) model, the effective **quantum workload grows exponentially**, outpacing the square-root speedup of Grover's algorithm.

In other words, **quantum machines may guess more cleverly — but they still have to guess**, and expanding the dimensionality of what they must guess (midstate *and* nonce) makes this significantly harder.

---

2. Concept: Midstate as “Password”, Full Hash as “Login”



This approach divides the PoW hash computation into two logical stages:

- **Stage 1** (≈ 50% of hash rounds): Produces a midstate `w` from the block data (excluding the nonce). This acts as a fixed dynamic “password”.
- **Stage 2** (remaining ≈ 50%): Uses `w` and the nonce `x` to produce the final hash `H(w, x)`.

Important clarification:

The midstate is **not** a truncation of the hash output. It is the internal state of the hash function after ≈50% of its rounds. For instance, in SHA-256 (64 rounds), the midstate is taken after 32 rounds and includes full internal state variables.

Analogy:

- The **midstate `w`** = password (fixed per block input)
- The **nonce `x`** = guessed input
- The **final hash** = login result
  Only when `w` and `x` align correctly is the output below the target threshold.

---

3. Quantitative Analysis

3.1 Entropy and Search Space

In traditional PoW:

- Nonce search space is typically 2^{32} (or 2^{64} in extended variants)
- Success probability per trial is $\approx 1 / 2^n$ (where `n` is difficulty level)

In the midstate-extended model:

- The midstate `w` has entropy ≈ 128 bits (from half-round SHA-256)
- The search space becomes `S = { (w, x) | w ∈ W, x ∈ N }` with size $\approx 2^{128} \times 2^{32} = 2^{160}$

This increases the size of the brute-force space from 2^{32} to 2^{160}.

3.2 Impact on Grover's Algorithm

Grover's search complexity is √N for a space of size N.

Model	Search Space	Grover Complexity
Classical PoW	2^n	O(2^(n/2))
Midstate Model	$2^{128} \times 2^{32} = 2^{160}$	$O(2^{80})$

Hence, compared to a classical target of $O(2^{16})$ (≈ 65k operations), this design raises the quantum attack cost to $O(2^{80})$, which is well beyond the foreseeable capabilities of quantum hardware. This means that quantum resistance is present.

---

4. Benefits

- **Zero additional computational cost**: The midstate is part of the existing hash pipeline and requires no extra function calls or cycles.
- **Reduced Nonce-Spam**: Early rejection is possible. Since `w` is fixed for given block data, nonce trials that result in invalid combinations with `w` can be filtered before full hashing.
- **Quantum Hardening**: Increases the effective security level against Grover's algorithm from 2^n to 2^{n+m}, where `m` is the midstate entropy.
- **Backward-Compatible**: Can be integrated with existing hash functions like SHA-256 or SHA3 without modifying their core logic.

---

5. Scientific References

- **Nakamoto, S.** (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://bitcoin.org/bitcoin.pdf

- **Grover, L. K.** (1996). *A Fast Quantum Mechanical Algorithm for Database Search*. Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC '96), pp. 212–219.
- **Gervais, A., Karame, G. O., Capkun, V., & Capkun, S.** (2016). *On the Security and Performance of Proof of Work Blockchains*. Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 3–16.
- **Bernstein, D. J., & Lange, T.** (2017). *Post-Quantum Cryptography*. *Nature*, 549(7671), 188–194. DOI: 10.1038/nature23461
- **Bausch, J., Cubitt, T. S., & Fitzsimons, J. F.** (2020). *Quantum Attacks on Proof of Work Algorithms: An Analysis*. *Journal of Quantum Computing*, preprint available via arXiv:2003.04219
- **Sompolinsky, Y., & Zohar, A.** (2015). *Secure High-Rate Transaction Processing in Bitcoin*. *Financial Cryptography and Data Security*, pp. 507–527. Springer, Berlin, Heidelberg.
- **Amy, M., Maslov, D., Mosca, M., & Roetteler, M.** (2013). *A Meet-in-the-Middle Algorithm for Fast Synthesis of Depth-Optimal Quantum Circuits*. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(6), 818–830.
- **Grassl, M., Langenberg, B., Roetteler, M., & Steinwandt, R.** (2016). *Applying Grover's Algorithm to AES: Quantum Resource Estimates*. *Post-Quantum Cryptography*, PQCrypto 2016, Lecture Notes in Computer Science, vol 9606. Springer.
- **Langenberg, B., Grassl, M., & Steinwandt, R.** (2021). *Reversible SHA-2 Circuits with Optimized Depth*. In *Post-Quantum Cryptography*, PQCrypto 2021, Lecture Notes in Computer Science, vol 12841. Springer.

---

6. Result

This approach leverages the internal structure of cryptographic hash functions to introduce a so-called *midstate* as an internal, password-like verification element. Without modifying the hash function or increasing computational overhead, it effectively doubles the dimensionality of the search space. This introduces a multiplicative barrier against brute-force attacks—both classical and quantum, particularly those using Grover's algorithm.

By transforming the Proof-of-Work process from a simple nonce search into a compound validation of (midstate, nonce), this method not only mitigates nonce-spam efficiently, but also provides a practical and backward-compatible pathway toward quantum-resistant security in existing blockchain systems.

This method demonstrates that intelligent use of existing hash processes is sufficient to elevate the security of PoW systems—without hardware upgrades, without new algorithms, but with lasting resilience against both current and emerging threats.

Additional note:

The discussion about quantum threats is often overdramatized in specialized crypto projects. As this article shows, potential vulnerabilities in proof-of-work systems can be mitigated with simple, efficient adjustments within existing structures – without any fundamental changes or

speculative technologies. Quantum security doesn't have to be complicated or expensive if the underlying cryptology is used intelligently. It should also be noted that this is an example using Sha3, a very simple hash. With a custom hash and/or higher bit values, the security could be increased to almost infinite levels—making it impossible even for thousands of quantum computers to succeed. This example also aims to demonstrate that any cryptocurrency can be protected from quantum computers during the mining process with less than 10 lines of code.