On the Suitability of Syndrome Decoding for Proof-of-Work under Quantum Adversaries: Design and Analysis

Aleck Nash, Henry Chimal-Dzul, and Kim-Kwang Raymond Choo

University of Texas at San Antonio, San Antonio Texas 78249, USA

Abstract. Proof-of-work (PoW) remains a fundamental mechanism for achieving decentralized consensus, most commonly instantiated using cryptographic hash functions. In such constructions, mining takes the form of an unstructured search problem over a large input space, where miners repeatedly evaluate candidate solutions until a valid one is found. While this design has proven effective in practice, it admits a quadratic quantum speedup via Grover's algorithm, raising concerns about the long-term security of hash-based mining. Motivated by this limitation, we investigate the use of code-based cryptographic problems as an alternative foundation for proof-of-work. In particular, we focus on the syndrome decoding problem and examine its classical and quantum complexity based on current state-of-the-art information-set decoding (ISD) algorithms and their quantum variants, comparing the resulting quantum advantage with that of hash-based and lattice-based constructions.
Building on this analysis, we propose a proof-of-work construction based on the Syndrome Decoding Problem (SDP) with a structured profile constraint, which enables controlled variation of the admissible solution density and provides additional parameters that may be used for difficulty calibration. Under the standard random-instance heuristic, we derive expressions for the expected number of valid solutions and the probability that a generated puzzle instance is satisfiable, providing a basis for studying parameter selection.

1 Introduction

Proof-of-Work (PoW) is among the most widely adopted consensus mechanisms and remains a fundamental building block in decentralized systems, most notably in blockchain protocols such as Bitcoin, Litecoin, Zcash, and Tidecoin. PoW facilitates decentralized security by permitting network participants, referred to as *miners*, to dedicate computational resources toward solving formidable cryptographic puzzles, a process known as *mining*. The first miner to find a valid solution earns the right to append a new block to the ledger, validate the included transactions, and receive a block reward. A defining characteristic of PoW is its computational asymmetry: while the mining process is intentionally resource-intensive, the resulting solution must be efficiently verifiable by all other network nodes, thereby securing the blockchain against Sybil attacks [13].

Modern PoW constructions predominantly rely on cryptographic hash functions. In these systems, miners execute a high volume of hash computations to identify a candidate that meets a specific difficulty target. This process essentially reduces to an unstructured search problem over a vast input space. While effective in practice, these designs are inherently susceptible to quantum acceleration. Specifically, Grover's algorithm provides a quadratic speedup for unstructured searches, effectively halving the security margin of hash-based PoW from 2^d to $2^{d/2}$. As quantum computing capabilities advance, this reduction poses a significant threat to the long-term equilibrium and security of hash-based mining systems.

Motivated by this impending threat, recent research has explored alternative PoW constructions based on quantum-resistant structured problems. Notably, lattice-based proposals have emerged aiming to mitigate the quantum advantage by utilizing problems such as the Shortest Vector Problem and its variants. For these structured lattice problems, the best-known quantum algorithms offer only marginal improvements over their classical counterparts [5]. This suggests that replacing unstructured search with structured hardness assumptions may provide a viable pathway toward quantum-resilient proof-of-work.

Code-based cryptography stands as one of the most established pillars of post-quantum cryptography, leveraging the computational hardness of problems originating from linear error-correcting codes. This paradigm was pioneered by Robert McEliece with the McEliece cryptosystem [18], whose security rests upon the intractability of the Syndrome Decoding Problem (SDP). This problem is known to be NP-complete [6], and its presumed average-case hardness under random instances forms the basis of numerous code-based cryptographic constructions. However, its potential as a quantum-resistant framework for PoW remains largely unexplored within the cryptographic community.

Our goal. In this paper, we leverage the properties of the Syndrome Decoding Problem in Coding Theory to propose a novel code-based PoW consensus mechanism. To the best of our knowledge, this is the first proof-of-work construction based on the Syndrome Decoding Problem where the search for a low-weight syndrome solution is utilized as a verifiable and adjustable metric of computational effort. We also provide an analysis framework for systematically evaluating the proposal and motivating further research into whether structured code-based problems can provide a viable foundation for PoW.

Our contributions. We make the following contributions:

- **Comparative complexity analysis.** In section 4, we review the best known classical and quantum attacks on the Syndrome Decoding Problem. Information-set decoding algorithms and their quantum counterparts are the best known solvers and represent the current state of the art. We quantify the resulting quantum advantage and compare it with that of hash-based proof-of-work and lattice-based constructions.
- **A novel Syndrome Decoding based PoW.** In section 5, we introduce a variant of the Syndrome Decoding Problem which considers a structured

profile constraint that controls the distribution of the support solution to this problem. We then propose a PoW construction based on this variant. We show that the construction is parameterizable and permits explicit control of the admissible solution density, while practical difficulty calibration depends on the best profile-aware decoding algorithms.
- **Heuristic analysis of solution density and success probability.** In section 6, we derive expressions for the expected number of valid solutions to the proposed profile constraint Syndrome Decoding Problem. By using this information, we compute the probability that a given instance admits a solution, under the standard random-instance heuristic. This provides a principled method for selecting parameters and controlling mining difficulty.
- **Implications for decoding-based mining.** In 6.5, we analyze how the imposed structure interacts with modern decoding algorithms, in particular Information Set Decoding techniques, and discuss how it affects success probabilities and overall complexity.

Our results indicate that the quantum advantage in PoW systems based on structured problems depends critically on the underlying problem and the complexity of the best-known solvers. Syndrome Decoding exhibits a different classical-to-quantum complexity gap than both hash-based and lattice-based PoW constructions, suggesting that code-based approaches provide an alternative point in the design space of PoW mechanisms.

We consider this work as a first step toward a broader exploration of code-based PoW mechanisms. In particular, further research is needed to understand how decoding algorithms can be adapted to exploit structure, to evaluate concrete parameter choices, and to assess practical deployment considerations.

2 Related Work

To mitigate the impact that quantum computing, especially Grover's algorithm, has over hash-based PoW mechanisms, several works have proposed PoW constructions based on structured computational problems. For example, Equihash [8] introduces a memory-hard PoW based on the generalized birthday problem, aimed at improving resistance to specialized hardware. More recently, lattice-based approaches have been proposed, where mining is based on solving instances of the Shortest Vector Problem using lattice reduction algorithms [5]. Ding [10] proposes a PoW based on multivariate cryptography, where the security relies on solving random systems of multivariate quadratic equations over a finite field.

The Syndrome Decoding Problem (SDP) has been extensively studied in code-based cryptography as a canonical NP-hard problem. The best known classical attack algorithms against SDP are based on Information-Set Decoding (ISD), with significant improvements over time through works such as Becker–Joux–May–Meurer [4] and May–Ozerov [17]. Quantum variants of these algorithms have also been developed, most notably by Kachigar and Tillich [15],

providing a detailed understanding of the classical and quantum complexity of decoding. However, this line of work focuses on cryptanalysis and does not consider applications to PoW, which requires specialized analysis due to its structural properties, particularly its tunable difficulty and probabilistic solution space.

3 Preliminaries

In this section we present the necessary definitions and preliminaries for our construction and analysis.

3.1 Proof-of-Work (PoW)

PoW is a computational mechanism that requires a prover to perform a tunable amount of work in order to produce a publicly verifiable witness. The idea originates in the pricing-via-processing framework of Dwork and Naor [11], and independently by [2] who proposed a hash-based PoW. Later, PoW was adopted as the core mining primitive in Bitcoin [20].

Formally, a PoW scheme consists of three algorithms:

$$\Pi_{\mathrm{PoW}} = (\mathsf{Gen}, \mathsf{Solve}, \mathsf{Verify}),$$

which are defined as follows:

- $\mathsf{Gen}(1^\kappa, D, r) \to C_r$: on input the security parameter κ, a difficulty descriptor D, and a public seed r, outputs a challenge instance C_r.
- $\mathsf{Solve}(C) \to w$: on input a challenge C, attempts to produce a witness w.
- $\mathsf{Verify}(C, w) \to \{0, 1\}$: on input a challenge C and a candidate witness w, outputs 1 if w is a valid proof for C, and 0 otherwise.

3.2 Linear Codes

Let $\mathbb{F}_q$ be a finite field with q elements and $n \geq 1$ be an integer. A linear code $\mathcal{C}$ of length n and dimension k is a k-dimensional subspace of $\mathbb{F}_q^n$. Such a code can be described as the kernel of a full rank *parity-check matrix* $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ as $\mathcal{C} = \left\{\mathbf{y} \in \mathbb{F}_q^n : \mathbf{H}\mathbf{y}^T = \mathbf{0}\right\}$. The *(Hamming) weight* of $\mathbf{x} \in \mathbb{F}_q^n$ is denoted as $w_H(\mathbf{x})$ and defined as the number of nonzero entries of $\mathbf{x}$. The *minimum weight* of $\mathcal{C}$ is $w_H(\mathcal{C}) = \min\{w_H(\mathbf{c}) : \mathbf{c} \in \mathcal{C}, \mathbf{c} \neq \mathbf{0}\}$.

3.3 The Syndrome Decoding Problem

One of the most prominent problems in coding theory is the Syndrome Decoding Problem (SDP). To precisely state this problem, recall that given a parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ of a linear code $\mathcal{C}$, the vector $\mathbf{s}_\mathbf{H}(\mathbf{y}) = \mathbf{H}\,\mathbf{y}^T \in \mathbb{F}_q^{n-k}$ is called the *syndrome* of $\mathbf{y} \in \mathbb{F}_q^n$. Notice that the syndrome of $\mathbf{y} \in \mathbb{F}_q^n$ depends on

the choice of a parity-check matrix. Moreover, since $\mathbf{H}$ has rank $n-k$, it follows that every vector in $\mathbb{F}_q^{n-k}$ is the syndrome of some $\mathbf{y} \in \mathbb{F}_q^n$.

Syndrome Decoding Problem (SDP). Let $\mathbb{F}_q$ be a finite field and $k \leq n$ integers. Given $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$ of full rank, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and a positive integer $t \leq n$, find an error vector $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{s_H}(\mathbf{e}) = \mathbf{s}$ and $w_H(\mathbf{e}) \leq t$.

The SDP can be trivially solved using Gaussian elimination if we do not require the condition $w_H(\mathbf{e}) \leq t$. However, the condition $w_H(\mathbf{e}) \leq t$ is non-linear and turns the SDP into a combinatorial search problem, which has been proven to be NP-complete for an arbitrary parity-check matrix $\mathbf{H}$ [6, 3]. Note that given a solution $\mathbf{e}$ to the SDP, verifying it is simple as it only requires to compute $\mathbf{s_H}(\mathbf{e}) = \mathbf{H}\mathbf{e}^t$ and verify that $\mathbf{s_H}(\mathbf{e}) = \mathbf{s}$, which can be efficiently implemented.

4 Comparative Complexity Analysis

The SDP problem has been studied for more than 50 years, and despite the significant amount of research developed around it, the best algorithms to solve the SDP are exponential in the number of errors that must be corrected. All these algorithms are called Information Set Decoding (ISD) algorithms and the first of them was introduced by Prange [22]. They all take advantage of the fact that there might exist a large set of column positions of $\mathbf{H}$ containing an information set (i.e. a subset of coordinate positions in a codeword that completely determines the original message) of the code that is error free.

For an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_2$, the cost of computing an error vector $\mathbf{e}$ of weight t in all these algorithms is $O\left(2^{\alpha(k/n,t/n)n}\right)$, where $\alpha(k/n, t/n)$ is a constant that depends on the rate of the code k/n and the ratio t/n. All the research efforts to improve ISD algorithms have focused on decreasing the value of this constant. In [15], values for the constant $\alpha(k/n, t/n)$ for the most relevant ISD algorithms. Among them, the May–Ozerov algorithm has the best running time with complexity $O(2^{0.0966n})$ [17]. While the May–Ozerov algorithm is an asymptotic improvement, it is widely considered a "galactic algorithm". This means that while it is mathematically faster for large values of n, the constant factors and the memory overhead are so massive that it only becomes more efficient than simpler algorithms (like Stern's or BJMM algorithms) for astronomical code lengths. Although May-Ozerov becomes advantageous only for sufficiently large parameters, it currently provides the best asymptotic exponent and is therefore used for complexity-gap comparison.

The idea of applying quantum algorithms to accelerate ISD decoding was first proposed in [21]. The approach in [21] of incorporating Grover's algorithm into classical ISD algorithm did not provide a significant speedup over classical algorithms. However, in [15] Grover's algorithm is used to find an error-free information set in combination with other quantum algorithms developed in [7] to provide a significant speedup of quantum ISD algorithms. The best running time

Table 1: Quantum speedup for different PoW constructions.

Problem	Best Classical Algorithm	Best Quantum Algorithm	Quantum speedup
Hash-based PoW	2^n	$2^{n/2}$	$2^{n/2}$
SDP-based PoW	$2^{0.0966n}$	$2^{0.05869n}$	$2^{0.03791n}$
SVP-based PoW	$2^{0.292n}$	$2^{0.265n}$	$2^{0.027n}$

of a quantum ISD algorithm developed in [15] is $O(2^{0.05869n})$ and corresponds to MMT + "1+1=0" + Grover + Quantum Walk algorithm.

In lattice-based constructions, the dominant factor in the lattice reduction algorithms corresponds to solving the SVP using heuristic lattice sieving algorithms. The lattice-based PoW proposal in [5] has reported classical and quantum running time complexities equal to $2^{0.292n}$ and $2^{0.265n}$, respectively.

On the other hand, in hash-based PoW mechanisms, the best classical algorithm has running time complexity 2^n, but since the mining process in these PoW mechanisms corresponds to an unstructured search problem, the best known quantum algorithm is Grover's algorithm which has complexity $2^{n/2}$.

Quantum speed up and the ratio of quantum to classical exponents are two natural ways to compare classical and quantum algorithms. The quantum speedup is obtained by comparing the classical and quantum running times, which are of the form $T_C \approx 2^{\alpha n}$ and $T_Q \approx 2^{\beta n}$. Taking the ratio yields $\frac{T_C}{T_Q} \approx \frac{2^{\alpha n}}{2^{\beta n}} = 2^{(\alpha-\beta)n}$. Thus, subtracting the exponents reflects the multiplicative advantage of the quantum algorithm over the classical one. In particular, the difference $\alpha - \beta$ determines the exponential factor by which the quantum solver outperforms the classical solver. The values in Table 2 reflect the quantum speedup through the exponent ratio β/α, where smaller values indicate a larger reduction in complexity. Hash-based proof-of-work achieves the largest reduction ($\beta/\alpha = 0.5$), syndrome decoding exhibits an intermediate reduction (≈ 0.61), and lattice-based constructions show a comparatively small reduction (≈ 0.91). Table 1 summarizes the quantum speedup for different PoW constructions.

The ratio of quantum to classical exponents captures the relative reduction in the exponential complexity when moving from classical to quantum algorithms. Smaller ratios correspond to larger quantum speedups. Table 2 summarizes the ratios of quantum to classical exponents for different PoW constructions.

The comparison exhibits qualitatively different quantum advantages, depending on the structure of the underlying computational problem. These comparisons are based on asymptotic complexity estimates under standard heuristic models, and therefore the results should be interpreted as comparisons of exponent scaling behavior rather than direct runtime equivalences. This indicates that the quantum speedup for SDP is smaller than the full quadratic speedup obtained via Grover's algorithm in hash-based PoW. In particular, the exponent reduces from 0.0966 to 0.05869, rather than to $0.0483 = 0.0966/2$ as would be

Table 2: Ratio of quantum to classical exponents for different PoW constructions.

Problem	Classical exponent α	Quantum exponent β	Ratio β/α
Hash-based PoW	1	0.5	0.50
SDP-based PoW	0.0966	0.05869	≈ 0.6075
SVP-based PoW	0.292	0.265	≈ 0.9075

expected under a full quadratic speedup. Therefore, while the structured nature of the SDP leads to a reduction in the achievable quantum speedup compared to unstructured search, the resulting advantage remains significant. In this sense, the SDP exhibits a classical-to-quantum complexity gap that is intermediate between hash-based constructions and those based on the SVP.

5 Syndrome Decoding-based PoW

In this section, we present a proposal for a PoW puzzle based on the SDP. This proposal is inspired by the work of Esser and Santini [12]. Let $n, k, t, \omega \in \mathbb{N}$ be public parameters satisfying $1 \leq t \leq n$, $0 < k < n$, and $0 \leq \omega \leq n$. Let $(n_1, \ldots, n_t)$ be positive integers such that $\sum_{i=1}^{t} n_i = n$ and let the target profile $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_t)$ satisfy

$$0 \leq \omega_i \leq n_i \quad \text{for all } i \in [t], \qquad \sum_{i=1}^{t} \omega_i = \omega.$$

On input a public seed $r \in \{0,1\}^{\kappa}$, the instance-generation algorithm outputs $C_r = (H_r, s_r, \mathcal{B}_r, \boldsymbol{\omega})$, where

- $H_r \in \mathbb{F}_2^{(n-k)\times n}$ is a uniformly random full rank parity-check matrix,
- $s_r \in \mathbb{F}_2^{n-k}$ is a uniformly random syndrome,
- $\mathcal{B}_r = \{B_1(r), \ldots, B_t(r)\}$ is a partition of $[n] := \{1, \ldots, n\}$ with $|B_i(r)| = n_i$.

Define the admissible set

$$\mathcal{A}_r := \{x \in \{0,1\}^n : \|x\|_H = \omega \text{ and } |\operatorname{supp}(x) \cap B_i(r)| = \omega_i \text{ for all } i \in [t]\}.$$

The goal is to find $x \in \mathcal{A}_r$ such that $H_r x^T = s_r^T$. The verification predicate is

$$\mathsf{Verify}(C_r, x) = \begin{cases} 1 & \text{if } x \in \mathcal{A}_r \text{ and } H_r x^T = s_r^T, \\ 0 & \text{otherwise.} \end{cases}$$

Instance generation, solving procedure, and verification are described in algorithms 1, 2, and 3 respectively.

Algorithm 1: $\mathsf{Gen}(1^\kappa, D, r)$ — Challenge Generation

Input: Security parameter 1^κ, difficulty descriptor D, and public seed $r \in \{0,1\}^\kappa$
Output: Puzzle instance $C_r = (H_r, s_r, \mathcal{B}_r, \boldsymbol{\omega})$

$\sigma \leftarrow \mathsf{PRG}(r)$;
repeat
  Parse the next $(n-k)n$ bits of σ as $M \in \mathbb{F}_2^{(n-k)\times n}$;
until $\mathrm{rank}(M) = n-k$;
$H_r \leftarrow M$;
Parse the next $n-k$ bits of σ as $s_r \in \mathbb{F}_2^{n-k}$;
Derive a permutation π_r of $[n]$ from σ;
$a_0 \leftarrow 0$;
for $i \leftarrow 1$ **to** t **do**
  $a_i \leftarrow a_{i-1} + n_i$;
  $B_i(r) \leftarrow \{\pi_r(a_{i-1}+1), \ldots, \pi_r(a_i)\}$;
$\mathcal{B}_r \leftarrow \{B_1(r), \ldots, B_t(r)\}$;
return $C_r = (H_r, s_r, \mathcal{B}_r, \boldsymbol{\omega})$;

Algorithm 2: $\mathsf{Solve}(C_r)$ — Solving Procedure

Input: $C_r = (H_r, s_r, \mathcal{B}_r, \boldsymbol{\omega})$
Output: $x \in \{0,1\}^n$ or $\perp$

Find $x \in \{0,1\}^n$ such that
$H_r x^T = s_r^T$, $\|x\|_H = \omega$, $|\,\mathrm{supp}(x) \cap B_i(r)| = \omega_i \quad \forall i \in [t]$

return x if found, otherwise $\perp$

Full-row-rank matrix. The matrix H_r is generated by deterministic rejection sampling from a pseudorandom bitstream until a full-row-rank matrix in $\mathbb{F}_2^{(n-k)\times n}$ is obtained. This guarantees $\mathrm{rank}(H_r) = n-k$ and matches the standard random-instance model used in syndrome decoding literature [17], [15].

Profile constraint. The vector $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_t)$ specifies how the support of a valid solution must be distributed across the partition $\mathcal{B}_r = \{B_1(r), \ldots, B_t(r)\}$. Thus, a valid solution must simultaneously satisfy the syndrome equation, the global Hamming weight constraint, and the blockwise support constraints. The matrix H_r and syndrome s_r are modeled as random instances, following the standard random-code assumption used in syndrome decoding analyses [23, 17, 4, 16]. Accordingly, the decoding complexities discussed in this paper are average-case estimates over randomly generated instances.

Difficulty Parameter. The mining difficulty is determined by the public parameter set $D = (n, k, t, \omega, (n_1, \ldots, n_t), \boldsymbol{\omega})$, where $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_t)$ and $\sum_{i=1}^t n_i = n$,

Algorithm 3: Verify(C_r, x) — Verification

Input: $C_r = (H_r, s_r, \mathcal{B}_r, \boldsymbol{\omega})$, x
Output: 1 if valid, 0 otherwise

if $x \notin \{0,1\}^n$ **then**
  return 0

1. Check Hamming weight
if $\|x\|_H \neq \omega$ **then**
  return 0

2. Check profile constraints
for $i \leftarrow 1$ **to** t **do**
  if $|\operatorname{supp}(x) \cap B_i(r)| \neq \omega_i$ **then**
    return 0;

3. Check syndrome equation
if $H_r x^T \neq s_r^T$ **then**
  return 0

return 1

$\sum_{i=1}^{t} \omega_i = \omega$. For a fixed security parameter κ, different choices of D induce syndrome decoding instances with different expected solution densities and decoding complexities. Consequently, difficulty adjustment is achieved by modifying the code parameters and support-profile parameters rather than by changing the verification procedure itself. As we will discuss in section 6, the expected number of valid solutions satisfies $\mathbb{E}[N_{\text{sol}}] \approx \left(\prod_{i=1}^{t} \binom{n_i}{\omega_i}\right) 2^{-(n-k)}$. Therefore, the network can regulate mining difficulty by adjusting the parameter set D. Increasing $n-k$ or decreasing the admissible support space reduces the expected solution density and therefore lowers the probability that a randomly generated instance admits a valid solution. Conversely, relaxing these parameters increases the success probability. The security parameter κ determines the length of the public seed and the security of the pseudorandom generation mechanism, whereas the difficulty descriptor

$$D = (n, k, t, \omega, (n_1, \ldots, n_t), \boldsymbol{\omega})$$

specifies the coding-theoretic and combinatorial parameters that govern the expected cost of solving the puzzle. The descriptor D identifies the parameters that determine the puzzle family, but the exact work factor must be estimated using the best known decoding algorithms for the corresponding parameter regime. A practical deployment could standardize a finite sequence of audited parameter sets and adjust difficulty by moving between them according to observed block times. Developing such a calibrated parameter schedule is outside the scope of this paper.

The source of r. In a blockchain deployment, the public seed r is derived deterministically from the current blockchain context, for example from the previous

block hash, the candidate block contents, and a miner-selected nonce. Thus, all participants can reconstruct the same puzzle instance from public information.

Connection to RSD. The proposed profile-constrained problem is related to structured syndrome-decoding variants, including Regular Syndrome Decoding (RSD) [12]. Standard RSD corresponds to a special profile in which the coordinates are divided into blocks and a prescribed number, often exactly one, of nonzero positions occurs in each block. Our formulation is different as it allows general block sizes and arbitrary target weights $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_t)$.

The public profile may in principle be incorporated into a specialized decoding algorithm. Therefore, the factor

$$\rho = \frac{\prod_{i=1}^{t} \binom{n_i}{\omega_i}}{\binom{n}{\omega}}$$

exactly measures the reduction in admissible support density, but does not by itself determine the complexity of the best profile-aware decoder.

For a decoder that generates ordinary weight-ω candidates without explicitly enforcing the profile, only a fraction ρ of the candidates satisfy the block constraints. Within this profile-oblivious attack model,

$$P_{\mathrm{succ}}^{\mathrm{profile}} \approx \rho P_{\mathrm{succ}}^{\mathrm{plain}}, \qquad T_{\mathrm{profile}} \approx \frac{1}{\rho} T_{\mathrm{plain}}.$$

This relation should be interpreted as a filtering heuristic rather than as an estimate of the optimal profile-aware decoding complexity.

6 Success Probability and Solution Density

In this section, we analyze the expected number of valid solutions and derive a heuristic expression for the probability that a generated instance is satisfiable. This analysis is used to tune the puzzle difficulty and to compare the proposed construction with the standard syndrome decoding problem.

We model the parity-check matrix H_r and syndrome s_r as pseudorandom and approximately uniformly distributed instances of the syndrome decoding problem. This modeling assumption is standard in code-based cryptography, where security relies on the hardness of decoding random linear codes [1], [4], [17], [15].

Our success-probability analysis follows the standard random-instance heuristic used in decoding complexity studies, in which candidate supports are assumed to behave approximately independently, and the number of solutions is estimated via expectation arguments [4], [15], [19].

This approach is analogous to that used in proof-of-work constructions, where the success probability per attempt determines the effective difficulty of the puzzle [8], [13].

6.1 Admissible Support Set

Based on our proposed construction, we know that the coordinate set $[n]$ is partitioned into blocks

$$\mathcal{B}_r = \{B_1(r), \ldots, B_t(r)\}, \qquad |B_i(r)| = n_i,$$

with $\sum_{i=1}^{t} n_i = n$. The target weight profile is

$$\boldsymbol{\omega} = (\omega_1, \ldots, \omega_t), \qquad \sum_{i=1}^{t} \omega_i = \omega, \qquad 0 \le \omega_i \le n_i.$$

A vector $x \in \{0,1\}^n$ is admissible if

$$|\mathrm{supp}(x) \cap B_i(r)| = \omega_i \quad \text{for all } i \in [t].$$

Equivalently, the admissible set is

$$\mathcal{A}_r = \{x \in \{0,1\}^n : |\mathrm{supp}(x) \cap B_i(r)| = \omega_i \ \forall i\}.$$

The number of admissible supports is therefore

$$N_{\mathrm{supp}} = |\mathcal{A}_r| = \prod_{i=1}^{t} \binom{n_i}{\omega_i}.$$

For comparison, the number of weight-ω vectors without profile constraints is

$$N_{\mathrm{supp}}^{\mathrm{plain}} = \binom{n}{\omega}.$$

Since

$$\prod_{i=1}^{t} \binom{n_i}{\omega_i} \le \binom{n}{\omega},$$

the profile constraint never increases the size of the admissible support space relative to the unconstrained weight-ω space. This quantity depends only on the block sizes $(n_1, \ldots, n_t)$ and the profile $\boldsymbol{\omega}$, and is therefore independent of the particular choice of r.

6.2 Probability That a Fixed Vector Satisfies the Syndrome

Let $H_r \in \{0,1\}^{(n-k)\times n}$ be a parity-check matrix generated pseudorandomly from the seed r, and let $s_r \in \{0,1\}^{n-k}$ be the target syndrome. Under the standard random-code heuristic, we model H_r as a uniformly random full-row-rank binary matrix and s_r as a uniformly random syndrome. For any fixed vector $x \in \{0,1\}^n$, the value $H_r x^T$ is then distributed approximately uniformly over $\{0,1\}^{n-k}$. Therefore,$\Pr[H_r x^T = s_r^T] \approx 2^{-(n-k)}$. This modeling assumption is standard in code-based cryptography, where security relies on the hardness of decoding random linear codes [9], [14].

6.3 Expected Number of Valid Solutions

In a proof-of-work setting, different public seeds r define fresh puzzle instances. Under the random-instance model, these generated instances are treated as independently distributed for the purpose of analyzing solution existence. This modeling assumption concerns the distribution of puzzle instances; it does not establish independence of optimized solver running times or rule out cross-instance amortization.

Let $\mathcal{A}_r$ denote the set of admissible vectors, i.e.

$$\mathcal{A}_r = \{x \in \{0,1\}^n : \|x\|_H = \omega,\ |\operatorname{supp}(x) \cap B_i(r)| = \omega_i\ \forall i\}\,.$$

For each $x \in \mathcal{A}_r$, we define the indicator random variable

$$I_x = \begin{cases} 1, & \text{if } H_r x^T = s_r^T, \\ 0, & \text{otherwise.} \end{cases}$$

Let $N_{\text{sol}} = \sum_{x \in \mathcal{A}_r} I_x$ denote the number of valid solutions.

Taking expectations and using linearity of expectation, we obtain

$$\mathbb{E}[N_{\text{sol}}] = \mathbb{E}\left[\sum_{x \in \mathcal{A}_r} I_x\right] = \sum_{x \in \mathcal{A}_r} \mathbb{E}[I_x] = \sum_{x \in \mathcal{A}_r} \Pr[H_r x^T = s_r^T].$$

Under the random-code heuristic, for any fixed x,

$$\Pr[H_r x^T = s_r^T] \approx 2^{-(n-k)}.$$

Furthermore, the size of $\mathcal{A}_r$ depends only on the block sizes $(n_1, \ldots, n_t)$ and the profile $\boldsymbol{\omega}$, and is given by

$$|\mathcal{A}_r| = \prod_{i=1}^{t} \binom{n_i}{\omega_i}.$$

Thus,

$$\mathbb{E}[N_{\text{sol}}] \approx \left(\prod_{i=1}^{t} \binom{n_i}{\omega_i}\right) 2^{-(n-k)}.$$

For the standard (unconstrained) syndrome decoding problem,

$$\mathbb{E}[N_{\text{sol}}^{\text{plain}}] \approx \binom{n}{\omega} 2^{-(n-k)}.$$

Hence the ratio

$$\rho = \frac{\prod_{i=1}^{t} \binom{n_i}{\omega_i}}{\binom{n}{\omega}}$$

quantifies the size of the constrained admissible set relative to the unconstrained case.

6.4 Heuristic Success Probability

Let $\mathcal{A}_r$ denote the set of admissible vectors, and for each $x \in \mathcal{A}_r$ define the indicator random variable

$$I_x = \begin{cases} 1, & \text{if } H_r x^T = s_r^T, \\ 0, & \text{otherwise.} \end{cases}$$

Then

$$N_{\mathrm{sol}} = \sum_{x \in \mathcal{A}_r} I_x, \qquad \mathbb{E}[N_{\mathrm{sol}}] = \sum_{x \in \mathcal{A}_r} \Pr[H_r x^T = s_r^T].$$

Under the random-code heuristic, for any fixed admissible vector,

$$\Pr[H_r x^T = s_r^T] \approx 2^{-(n-k)},$$

so that

$$\mathbb{E}[N_{\mathrm{sol}}] \approx |\mathcal{A}_r|\, 2^{-(n-k)}.$$

Assuming approximate independence between the events $\{H_r x^T = s_r^T\}_{x \in \mathcal{A}_r}$, the random variable N_{sol} may be modeled heuristically by a Poisson distribution with parameter $\lambda = \mathbb{E}[N_{\mathrm{sol}}]$.

For a Poisson random variable with parameter κ,

$$\Pr[N_{\mathrm{sol}} = 0] = e^{-\lambda}, \qquad \Pr[N_{\mathrm{sol}} \geq 1] = 1 - e^{-\lambda}.$$

Let p denote the probability that a puzzle instance admits at least one valid solution. Then $p = \Pr[N_{\mathrm{sol}} \geq 1] \approx 1 - \exp(-\lambda)$.

Substituting the estimate for λ yields

$$p \approx 1 - \exp\left(- \left(\prod_{i=1}^{t} \binom{n_i}{\omega_i} \right) 2^{-(n-k)} \right).$$

This expression enables tuning of the puzzle parameters so that the probability of solvability is sufficiently high. In particular, it is desirable to operate in a setting where λ is moderately large, so that p is close to one and miners are unlikely to encounter unsatisfiable instances. Here, the dominant computational cost arises from solving a single constrained decoding instance, rather than from repeated trials.

6.5 Implications for Puzzle Difficulty

Modern decoding algorithms are based on structured search techniques such as Stern's algorithm, MMT, BJMM, and May–Ozerov style information-set decoding (ISD) algorithms [23, 17, 4, 16], together with list merging and nearest-neighbor filtering techniques. These algorithms construct partial candidates and combine them to obtain a full solution. The profile constraint restricts the admissible support patterns, and therefore reduces the number of valid solutions

relative to the unconstrained case. If the constraint cannot be effectively incorporated during the early stages of the decoding procedure, then the cost of list construction remains essentially unchanged, while the probability that a candidate satisfies the profile constraint is reduced. Consequently, the profile constraint primarily affects the success probability of candidate solutions in the final stages of the algorithm.

Compared with the standard syndrome decoding problem, the number of valid solutions is reduced by a factor $\rho = \frac{\prod_{i=1}^{t} \binom{n_i}{\omega_i}}{\binom{n}{\omega}}$. Thus, for algorithms that generate candidate solutions without explicitly enforcing the profile constraint, only a fraction ρ of weight-ω candidates satisfy the required profile. Consequently, the probability that a candidate produced by an ISD iteration corresponds to a valid solution is reduced by the same factor.

For information-set-decoding-style algorithms, the expected running time may be expressed as

$$T \approx \frac{C_{\text{iter}}}{P_{\text{succ}}},$$

where C_{iter} is the cost of one iteration and P_{succ} is the probability that the iteration produces a valid solution.

For a profile-oblivious decoder that generates ordinary weight-ω candidates without incorporating the block constraints, only a fraction ρ of such candidates satisfy the prescribed profile. Within this restricted attack model,

$$P_{\text{succ}}^{\text{profile}} \approx \rho P_{\text{succ}}^{\text{plain}}, \qquad T_{\text{profile}} \approx \frac{1}{\rho} T_{\text{plain}}.$$

This is a filtering heuristic for profile-oblivious decoding and should not be interpreted as an estimate of the optimal profile-aware decoding complexity. Specialized algorithms may exploit the public block structure directly.

7 Concluding Remarks

In this work, we studied syndrome decoding as a foundation for proof-of-work and analyzed its classical and quantum complexity using state-of-the-art decoding algorithms. Our results quantify the corresponding classical-to-quantum gap and place code-based constructions alongside hash-based and lattice-based approaches within a unified framework. Furthermore, we introduced a profile-constrained variant of syndrome-decoding proof-of-work that enables structured control over the solution space and supports principled parameter tuning via solution density and success probability analysis. Taken together, these results identify code-based proof-of-work as a structured alternative to hash-based mining with a distinct quantum advantage profile, and provide a basis for further investigation of structured hardness assumptions in proof-of-work design.

Limitations. To the best of our knowledge, this work provides a first analysis of syndrome-decoding-based proof-of-work under standard heuristic models. The

results are based on random-instance assumptions and approximate independence heuristics commonly used in decoding complexity analyses. While these assumptions are widely used in the literature, they do not constitute formal guarantees. Additionally, our proposed construction does not eliminate the quantum advantage: existing quantum ISD algorithms still yield a non-negligible speedup. Our contribution is to quantify this gap and to show that it differs from that of unstructured search. Moreover, the effect of the profile constraint depends on how decoding algorithms interact with structured solution spaces. Our analysis captures the baseline setting in which this structure is not fully exploited. However, the extent to which it can be leveraged algorithmically remains open. Finally, our comparisons are asymptotic and based on exponent-level models, and should be interpreted as comparative indicators rather than exact runtime predictions.

Future work. Several directions remain open. In particular, it is important to understand how modern decoding algorithms can be adapted to exploit structured constraints, to evaluate concrete parameter choices in practice, and to investigate alternative constructions that further shape the solution space. Additionally, this work focuses on the construction and decoding complexity of the underlying computational puzzle. A complete deployment in Nakamoto-style consensus would additionally require analysis of precomputation, possible cross-instance amortization, parallel scaling, and the distribution of block-generation times. These protocol-level questions are left for future work. These directions may lead to a more refined understanding of quantum advantage in structured proof-of-work systems.

Bibliography

[1] Carlos Aguilar-Melchor, Olivier Blazy, Jean-Christophe Deneuville, Philippe Gaborit, and Gilles Zémor. Efficient encryption from random quasi-cyclic codes. *IEEE Transactions on Information Theory*, 64(5):3927–3943, 2018.

[2] Adam Back. Hashcash – a denial of service counter-measure. Technical report, 2002.

[3] Sergey Barg. Some new np-complete coding problems. *Problems of Information Transmission*, 30(3):209–214, 1994. Originally published in Russian as "Problemy Peredachi Informatsii", 30(3):23–28, 1994.

[4] Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20: How 1+ 1= 0 improves information set decoding. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 520–536. Springer, 2012.

[5] Rouzbeh Behnia, Eamonn W Postlethwaite, Muslum Ozgur Ozmen, and Attila Altay Yavuz. Lattice-based proof-of-work for post-quantum blockchains. In *International Workshop on Data Privacy Management*, pages 310–318. Springer, 2021.

[6] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978. https://doi.org/10.1109/TIT.1978.1055873.

[7] Daniel J. Bernstein, Stacey Jeffery, Tanja Lange, and Alexander Meurer. Quantum algorithms for the subset-sum problem. In Philippe Gaborit, editor, *Post-Quantum Cryptography (PQCrypto 2013)*, volume 7932 of *Lecture Notes in Computer Science*, pages 16–33, Heidelberg, 2013. Springer. https://doi.org/10.1007/978-3-642-38616-9_2.

[8] Alex Biryukov and Dmitry Khovratovich. Equihash: Asymmetric proof-of-work based on the generalized birthday problem. *Ledger*, 2:1–30, 2017.

[9] Miguel Cueto Noval, Simon-Philipp Merz, Patrick Stählin, and Akin Ünal. On the soundness of algebraic attacks against code-based assumptions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 385–415. Springer, 2025.

[10] Jintai Ding. A new proof of work for blockchain based on random multivariate quadratic equations. In *International Conference on Applied Cryptography and Network Security*, pages 97–107. Springer, 2019.

[11] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *CRYPTO*, 1992.

[12] Andre Esser and Paolo Santini. Not just regular decoding: asymptotics and improvements of regular syndrome decoding attacks. In *Annual International Cryptology Conference*, pages 183–217. Springer, 2024.

[13] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. *Journal of the ACM*, 71(4):1–49, 2024.

[14] Anna-Lena Horlemann-Trautmann and Violetta Weger. Information set decoding in the lee metric with applications to cryptography. *arXiv preprint arXiv:1903.07692*, 2019.

[15] Ghazal Kachigar and Jean-Pierre Tillich. Quantum information set decoding algorithms. In *International Workshop on Post-Quantum Cryptography*, pages 69–89. Springer, 2017.

[16] Alexander May and Ilya Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 203–228. Springer, 2015.

[17] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 107–124. Springer, 2011.

[18] Robert McEliece. A public key cryptosystem based on algebraic coding theory. *DSN Progress Report*, 42-44:114–116, 1978.

[19] Alexander Meurer. A coding-theoretic approach to cryptanalysis. 2013.

[20] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[21] Raphael Overbeck and Nicolas Sendrier. Code-based cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 95–145. Springer, Heidelberg, 2009. https://doi.org/10.1007/978-3-540-88702-7_4.

[22] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, September 1962. https://doi.org/10.1109/TIT.1962.1057777.

[23] Jacques Stern. A method for finding codewords of small weight. *Coding Theory and Applications*, 388:106–113, 1989.