Proof-of-Work Hash Mining for Finite Palette Image Encoding

ImageMaker Foundation

June 2025

Abstract

This paper presents a novel cryptographic protocol for encoding fixed-size digital images using a deterministic proof-of-work (PoW) mechanism. The image consists of a 16×16 grid, with each pixel drawn from a 16-color palette. Each image row must be independently mined by discovering a cryptographic secret such that a SHA3-512 hash satisfies specific positional constraints. The algorithm is analogous to Bitcoin mining: whereas Bitcoin miners secure blocks through computation, the **ImageMaker** protocol assigns value to images by binding them to verifiable work. The mined image becomes a scarce digital asset with intrinsic computational provenance.

1 Introduction

In blockchain systems like Bitcoin, blocks are mined via computational work, ensuring security and scarcity. We extend this concept to digital images: the **ImageMaker** protocol transforms images into cryptographically mined artifacts. Each image, fixed at 16×16 pixels and using a 16-color palette, is mined row-by-row using a chained proof-of-work scheme.

This introduces a new class of digital assets: images whose existence requires demonstrable computation. The result is an object whose authenticity, integrity, and provenance are cryptographically verifiable—analogous to how Bitcoin miners secure blocks.

2 System Design

2.1 Image Format

- **Dimensions:** 16 rows of 16 pixels each.
- **Color Palette:** 16 fixed colors indexed by hexadecimal digits (0–f).
- **Encoding:** Each row is represented as a 16-character hex string.
- **Wildcard:** The index `0` (black) acts as a wildcard during mining and imposes no hash constraint.

Let $I = \{R_0, R_1, \ldots, R_{15}\}$ be the image, where each row $R_y \in \{0, 1, \ldots, f\}^{16}$.

2.2 Hash Input Construction

For row R_y, the input string to be hashed is:

$$\text{Input}_y = \text{ImgName} + \texttt{""} + (S_0 + \texttt{"\_"} + \cdots + S_{y-1}) + \text{HexDigit}(y) + S_y$$

where:

- `ImgName` is the image identifier.
- S_i are secrets mined for previous rows (the first row has no secrets, so this is omitted).

- `HexDigit`(y) is the hexadecimal digit for row index y.
- S_y is the candidate secret being tested.

This string is ASCII-encoded and hashed using SHA3-512:

$$H_y = \text{HexString}(\text{SHA3-512}(\text{ASCII}(\text{Input}_y)))$$

2.3 Hash Validation Rule

Let $H_y^0, H_y^1, \ldots, H_y^{15}$ be the first 16 hex digits of the hash. The hash is valid for row R_y if:

$$\forall i \in [0, 15] : \quad R_y[i] = 0 \quad \text{or} \quad R_y[i] = H_y^i$$

This enforces that the hash prefix matches the row encoding, except in wildcard positions.

3 Mining Algorithm

3.1 Pseudocode

Listing 1: Row mining procedure

```
FUNCTION MineRow(R_y, ImgName, Y, Secrets):
    SecretLength = 1

    FOR EACH AlphanumericString S_y OF LENGTH SecretLength:
        IF S_y IN Secrets:
            CONTINUE

        Input = ImgName + " " + Join(Secrets, "_") + ToHexDigit(Y) + S_y
        Hash = ToHexString(SHA3_512(ASCIIEncode(Input)))

        Valid = True
        FOR i FROM 0 TO 15:
            IF R_y[i] != 0 AND Hash[i] != R_y[i]:
                Valid = False
                BREAK

        IF Valid:
            APPEND S_y TO Secrets
            RETURN S_y

        SecretLength = SecretLength + 1
```

3.2 Sequential Mining

The mining proceeds row-by-row:

1. Start with an empty list of secrets.
2. For row $y = 0$ to 15, mine a unique secret S_y satisfying the hash constraint.
3. Each rows hash input includes all prior secrets, creating a dependency chain.
4. The process is strictly sequential and non-parallelizable.

4 Security Properties

- **Cryptographic Strength:** Uses SHA3-512 for 512-bit resistance to preimage and collision attacks.
- **Sequential Dependence:** Later rows cannot be mined without all prior secrets.
- **Immutability:** Altering a single pixel invalidates the hash and necessitates re-mining.
- **Uniqueness:** All secrets in an image must be distinct, preventing reuse.

5 Analogy with Bitcoin

Just as Bitcoin secures its ledger via block mining, ImageMaker secures image data via row mining:

Bitcoin	**ImageMaker**
Block	Image Row
Nonce	Secret String
Block Hash	Row Hash (prefix)
Proof-of-Work Difficulty	Pixel Pattern Constraints
Mining Reward	Image Value
Chain of Blocks	Chain of Secrets

In both systems, work imparts value. For Bitcoin, blocks derive trust from computation. For ImageMaker, images derive uniqueness and scarcity from computation. This makes each mined image a verifiable artifact of digital labor.

6 Benefits of the Protocol

6.1 Intrinsic Value through Work

The images existence is tied to verifiable computation. This enforces a natural scarcity and value, akin to mined cryptocurrencies.

6.2 Difficulty Scaling

Images with more non-zero pixels impose stricter hash constraints, requiring more time to mine. This links visual complexity directly to economic cost.

6.3 Authenticity and Provenance

The sequence of secrets forms a cryptographic fingerprint for the image. Any modification requires full re-mining, making the image tamper-evident and self-verifiable.

7 ImageMaker NFT Platform

The **ImageMaker NFT Platform** implements this protocol as a decentralized application. Users can:

- Design 16×16 pixel artworks from the 16-color palette.
- Trigger the mining protocol to derive valid secrets.
- Mint the mined image as an NFT, embedding the hash proof.
- Trade or publish the image as a verifiable computational artifact.

The platform ensures that only fully mined images can be minted, tying NFT ownership to a demonstrable act of computational creation.

8 Conclusion

We have introduced a mining protocol for finite-palette digital images inspired by proof-of-work in cryptocurrencies. By binding image structure to hash constraints, and enforcing sequential mining, we enable digital artifacts whose value is backed by computation. This protocol, as implemented by the ImageMaker NFT Platform, represents a new paradigm: mined images as provable, scarce digital assets.

References

- NIST FIPS 202 SHA-3 Standard: *Permutation-Based Hash and Extendable-Output Functions*
- Ethereum Foundation *Proof-of-Work Concepts*
- Keccak Team `https://keccak.team/`