Hash Layer: Permissionless PoW Infrastructure for Sui

Version 1.0

Anton Masanavets

October 2025

#Abstract

Hash Layer is a decentralized Proof-of-Work protocol implemented entirely in smart contracts on the Sui blockchain. Unlike traditional PoW systems that rely on separate consensus layers, Hash leverages Sui’s high-throughput architecture to record a transparent, verifiable, and immutable history of computational effort.

Each mined block is recorded as a network event, forming a cryptographically linked chain of work. Through Sui’s RPC interface, the entire chain can be validated from any point back to genesis. Hash introduces dynamic difficulty adjustment to maintain a consistent block tempo, and a linear reward decay mechanism that leads to a fixed total supply of 10 billion Hash tokens. Hash is a programmable mining layer—an engine for decentralized applications built on provable computation.

## 1. Introduction: The Cypherpunk Legacy Reborn

Online commerce has long depended on trusted intermediaries to process payments. While functional, this model inherits the vulnerabilities of trust.

In 2008, Satoshi Nakamoto proposed a peer-to-peer system that timestamps transactions by hashing them into an ongoing chain of proof-of-work. This chain, Bitcoin, was powered by SHA-256—a cryptographic primitive published by NIST in 2001. Its collision resistance and simplicity made it the ideal foundation for digital gold. The cypherpunk movement had long envisioned such systems. Their ethos—that privacy and sovereignty should be guaranteed by cryptography, not institutions—was finally realized.

Hash Layer returns to these principles, reimagined for a modern smart contract platform. It embeds the timeless security of SHA-256 into Sui’s object-oriented architecture, creating a digital commodity mined by computation and composable within decentralized applications.

“Privacy is necessary for an open society in the electronic age. Privacy is not secrecy.” — Eric Hughes, A Cypherpunk’s Manifesto, 1993

Hash is a monument to that privacy—and to the work required to secure it.

## 2. Architecture: A Blockchain Within a Blockchain

### 2.1. The Block as a Verifiable Event

In Hash Layer, each mined block is recorded as a network event. Its structure is defined in Move as:

```
public struct Block has drop, store, copy {
    height: u64,
    previous_hash: vector<u8>, // 32 bytes
    nonce: u64,
    data: Option<vector<u8>>   // Optional payload
}
```

Each block includes a reference to the previous hash, forming a cryptographic chain. Events are emitted directly on-chain, and can be retrieved and verified via RPC to reconstruct the full mining history.

### 2.2. Mining and Verification

Mining involves finding a nonce such that the SHA-256 hash of the block's contents meets the current difficulty target:

```
    let previous_hash = get_hash(chain);
    let mut pow_input = vector::empty<u8>();
    vector::append(&mut pow_input, previous_hash); //hash
    vector::append(&mut pow_input, bcs::to_bytes(&nonce)); //nonce
    vector::append(&mut pow_input, bcs::to_bytes(&data)); //extra nonce
    let hash = hash::sha2_256(pow_input);
    assert!(count_leading_zero_bits(hash) >= chain.difficulty, EINVALID_POW);
```

Once a valid nonce is found, the miner submits a transaction to the mine_block function. The contract verifies the proof-of-work and emits a Block event, permanently recording the computation.

### 3. Emission and Monetary Policy

Hash follows a predictable and transparent emission model inspired by Bitcoin:

- Initial Block Reward: 100 Hash
- Total Supply Cap: 10,000,000,000 Hash
- Decay Mechanism: Linear reward decay over time

Unlike Bitcoin’s abrupt halving schedule, Hash uses a smooth, continuous reduction in block rewards.

### First-Year Projection

At a target rate of 1 block per second, approximately 31,536,000 blocks will be mined in the first year. With an average reward of ~90 Hash, the estimated issuance is: 31,536,000 × 90 = ~2.83 billion Hash. This represents ~28% of the total supply, ensuring broad early distribution.

## 4. Dynamic Difficulty Adjustment

To maintain a consistent tempo, Hash adjusts difficulty every 1,000 blocks:

new_difficulty = old_difficulty * TARGET_TIME / actual_time

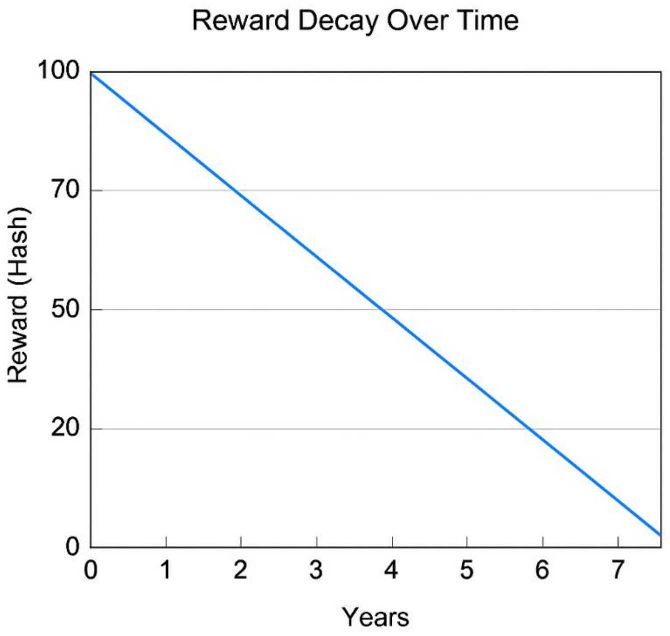
The target time for 1,000 blocks is 1,000 seconds. This algorithm ensures long-term stability by adapting to changes in aggregate hash rate.

## 5. Linear Reward Decay

Hash implements a linear decay over a 10-year mining horizon:

Reward = START_REWARD - Block Number * START_REWARD / TOTAL_BLOCKS

At 1 block per second, ~31.5 million blocks are mined per year. This creates a predictable emission curve and a smooth transition from early distribution to long-term scarcity.



## 6. The Programmable Mining Layer

Hash is more than a token—it’s a programmable layer of computation. Because each block is verifiable on-chain, smart contracts can:

- Verify Proof-of-Work: Any app can confirm that a user performed valid computation
- Create Derivative Assets: Build financial instruments based on hash rate or future rewards
- Implement Proof-of-Delay: Use block timestamps as decentralized time sources
- Power Games and NFTs: Mining a block can grant rights to mint unique assets or collectibles

Hash transforms raw computation into a composable primitive for Web3.

## 7. Ultra-Low Cost Validation

Hash Layer is engineered for minimal gas consumption without compromising verifiability. Instead of storing mined blocks as shared objects, each block is emitted as a lightweight event using event::emit<Block>. This architectural choice is foundational to the protocol’s efficiency.

1. No shared object mutations. Avoids expensive borrow_object_mut operations, which typically increase gas cost in Sui.
2. No persistent storage overhead. Events are ephemeral and do not occupy long-term storage, yet remain fully accessible via RPC.

3. Compact serialization. Each Block structure is ~49 bytes, plus minimal event metadata.
4. Linear chain structure. Each block includes a previous_hash, enabling full-chain validation from any block back to genesis.

This makes Hash Layer one of the most cost-efficient PoW systems ever deployed on-chain—fully verifiable, yet scalable to millions of blocks with negligible cost.

## References

- [1] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System

- [2] NIST. (2001). FIPS PUB 180-2: Secure Hash Standard

- [3] Sui Foundation. (2023). Sui Documentation: Objects and Assets

- [4] Hughes, E. (1993). A Cypherpunk’s Manifesto