Ethereal

A Capped-Supply ERC-20 with Adaptive On-Chain Mining

Ethereal Core Developers

July 30, 2025

Abstract

Ethereal (`ETH`) is an ERC-20 token that combines a fixed monetary ceiling of 72,000,000 ETH with a lightweight, self-contained mining mechanism executed entirely in a smart contract. Holders may compete for block rewards by submitting arbitrary data; the highest hash value of the data within each Ethereum block height earns the reward. Issuance begins at 5 ETH per block and halves every 2^{21} or 2,097,152 blocks. The mechanism yields predictable long-term supply, requires no external oracles, and is fully auditable on-chain.

1 Introduction

Bitcoin demonstrated that a transparent halving schedule can bootstrap a digital currency without central authority. Ethereal adapts this principle to the ERC-20 standard while leveraging Ethereum's existing consensus. Instead of proof-of-work, Ethereal employs a *proof-of-hash ordering mechanism* that minimises external energy cost yet preserves fair, permissionless distribution.

2 Monetary Parameters

Maximum supply, $S_{\max}$	72,000,000 ETH
Genesis block reward, R_0	5 ETH per block
Halving interval, H	$2^{21} = 2{,}097{,}152$ blocks
Decimals	10^{18} (standard ERC-20)

Reward schedule. After n complete halvings the per-block reward is

$$R_n = \frac{R_0}{2^n}, \qquad n \in \mathbb{N}_{\geq 0}. \tag{1}$$

Let B denote the block height counted **inside the contract**:

$$B = \texttt{block.number} - \texttt{BLOCK\_OFFSET}.$$

Defining the current era $n = \left\lfloor \frac{B}{H} \right\rfloor$, the cumulative emission up to B is

$$S(B) \;=\; \sum_{i=0}^{n-1} R_i H \;+\; R_n \left(B - nH\right). \tag{2}$$

When $R_n \to 0$ the series converges to $S_{\max}$.

3 Mining Mechanism

3.1 Submission

During block B any address with non-zero Ethereal balance may call `mine(bytes data)` with an arbitrary payload. The contract evaluates $h = \text{KECCAK256}(\texttt{data})$ and enforces:

- *Uniqueness.* h must not appear in the global bitmap `hashes`$[h]$. Duplicate hashes revert with `HashUsed`.
- *Balance gate.* The caller must hold > 0 ETH to discourage Sybil spam (`InsufficientBalance`).

3.2 Scoring Rule

Let $h^\star_B$ be the best hash recorded for height B. The ordering rule is *maximum as winner*:

$$h > h^\star_B \implies \text{caller becomes } \texttt{CURRENT\_MINER}.$$

If a new Ethereum block begins (B+1), the previous `CURRENT_MINER` is finalised and rewarded R_n ETH, where n is the era per Sec. 2. Multiple elapsed blocks are aggregated into one payout to bound gas.

3.3 Halving Enforcement

Whenever B crosses a multiple of H, the contract iterates through any skipped eras, issuing rewards retroactively (Figure 1). Gas is bounded because the loop executes at most once per halving boundary.

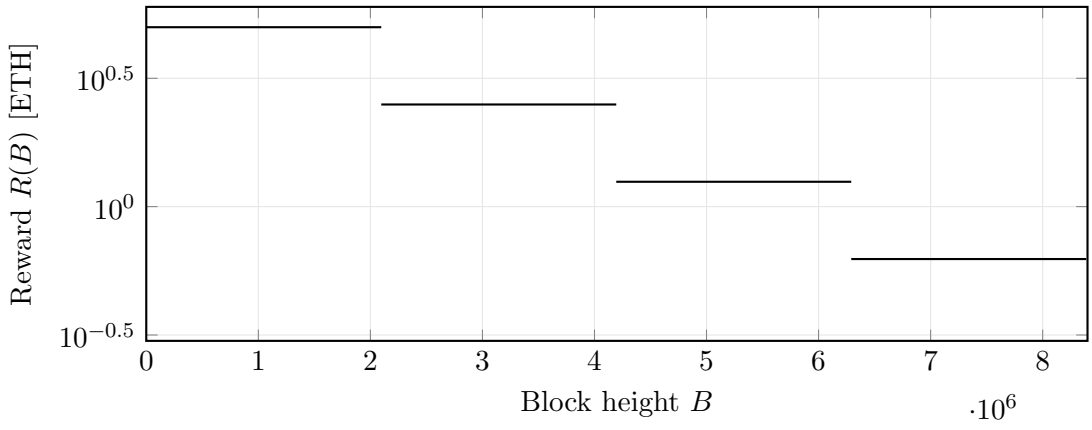

Figure 1: Block reward over time. Each flat segment shows a halving era; the reward halves every 2^{21} blocks.

4 Security Considerations

Replay protection. Hash uniqueness prevents miners from re-submitting the same winning payload in a later block.

Grinding. Because KECCAK256 behaves as a random oracle, the expected search complexity for improving $h_B^\star$ is 2^{255} on average, rendering grinding economically infeasible within the 12 s block time.

Denial-of-Service. Requiring an existing balance before mining raises the cost of spamming the contract: attackers must purchase and risk ETH each attempt.

5 Conclusion

Ethereal delivers a transparent issuance curve (Eq. 2), a self-contained on-chain mechanism that substitutes traditional energy-intensive computation with a lightweight hash-ordering process, and full ERC-20 compatibility. The design targets long-term monetary predictability while remaining efficient enough to operate entirely on the Ethereum base layer.

Acknowledgements

We thank the broader open-source community and the maintainers of OpenZeppelin Contracts for their audited ERC-20 implementation.

References

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

[2] G. Bertoni *et al.*, *Keccak sponge function family main document*, Submission to NIST (Round 3), 2011.