

Proof of Time: A Peer-to-Peer Digital Time System

Alejandro Montana
alejandromontana@tutamail.com

December 25, 2025

Abstract

A purely peer-to-peer version of digital time would allow temporal assets to be distributed directly between participants without centralized emission. Cryptographic proofs provide control of ownership, but the main benefits are lost if trust in capital or computational power is required for consensus. We propose a solution to the problem of plutocratic capture using a peer-to-peer network. The network timestamps blocks through VDF (Verifiable Delay Functions), forming a record that cannot be changed without redoing the proof. The longest chain serves as proof of the sequence of events and accumulated time presence. As long as a majority of weighted influence is controlled by honest nodes, they will generate the longest chain and outpace attackers. The network requires a minimum of 3 active nodes. Nodes can leave and rejoin the network at will, accepting the longest chain as proof of what happened while they were gone.

1 Introduction

Distributed consensus systems rely on resource-dependent mechanisms. Proof of Work scales influence through computational power—a participant with capital purchases ASICs and controls hashrate proportional to investment. Proof of Stake scales influence through capital—a participant with wealth stakes coins and receives rewards proportional to stake. While these systems work well enough for most cases, they suffer from inherent weaknesses of the plutocratic model.

The problem of all resource-dependent consensus mechanisms is plutocratic capture. A participant with capital can purchase computational power, accumulate tokens for staking, or rent storage, gaining proportional influence on consensus. This inevitably concentrates power in the hands of capital owners.

What is needed is a consensus mechanism based on cryptographic proof of irreversible commitments instead of scalable resources. Time cannot be purchased, manufactured, or accelerated. A node operating for 180 days accumulates reputation regardless of the owner’s capital. This time is irreversible and cannot be transferred.

Nodes compete for 21 million minutes of time (40 years of asset) within a window of 131 years of calendar time with reward halving every 210,000 blocks ($50 \rightarrow 25 \rightarrow 12.5$ minutes).

2 Temporal Proofs

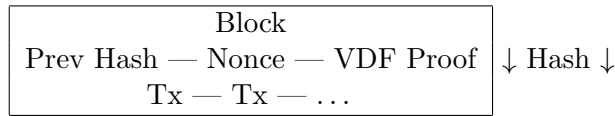
We define a temporal asset as a chain of cryptographic proofs. Each node creates proof-of-time through VDF (Verifiable Delay Function), confirming continuity of the temporal interval between blocks. Recipients can verify proofs to check the chain of time.

VDF guarantees impossibility of pre-computation: each proof depends on the hash of the previous block $\text{VDF}(h_{\text{prev}}, t) \rightarrow \pi$. Computation requires sequential execution of t iterations without possibility of parallelization. Computation time linearly depends on difficulty parameter, verification takes logarithmic time.

Recipients verify timestamp validity through the network’s time chain. Only timestamps sequentially generated after the previous block in chronological order are considered valid. Attempts to generate proof at an earlier timestamp or create conflicting proofs constitute chain rewriting—equivocation, which leads to slashing: immediate reputation reset + 180-day quarantine (26,000 blocks, reward probability = 0).

3 Timestamp Server

The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash. The timestamp proves that the data must have existed at the time to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.



4 Proof-of-Time

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-time system through VDF. Proof-of-time involves sequential computation which, when executed, produces proof that a certain amount of time has passed. Average required time linearly depends on VDF complexity and can be verified in logarithmic time.

Node probability P_i is calculated from three saturating components:

$$P_i = \frac{w_{\text{time}} \cdot f_{\text{time}}(t_i) + w_{\text{space}} \cdot f_{\text{space}}(s_i) + w_{\text{rep}} \cdot f_{\text{rep}}(r_i)}{Z} \quad (1)$$

where Z is normalization constant ($\sum P_i = 1$), and target weights $w_{\text{time}} = 0.60$, $w_{\text{space}} = 0.20$, $w_{\text{rep}} = 0.20$. Functions saturate:

$$f_{\text{time}}(t_i) = \min(t_i/k_{\text{time}}, 1), \quad k_{\text{time}} = 180 \text{ days} = 15,552,000 \text{ seconds} \quad (2)$$

$$f_{\text{space}}(s_i) = \min(s_i/k_{\text{space}}, 1), \quad k_{\text{space}} = 0.80 \times \text{total chain history} \quad (3)$$

$$f_{\text{rep}}(r_i) = \min(r_i/k_{\text{rep}}, 1), \quad k_{\text{rep}} = 2,016 \text{ signed blocks} \quad (4)$$

If a majority of weighted influence is controlled by honest nodes, the honest chain will grow fastest and outpace any competing chains. To compensate for varying interest in running nodes over time, weights are rebalanced every 2,016 blocks toward the 60/20/20 target. If a component becomes dominant, its coefficient is adjusted downward.

5 Network

The steps to run the network are as follows:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Leader is selected through VRF based on previous block hash and weighted probability.
4. Selected leader computes VDF proof for their block.

5. When leader finds proof-of-time, they broadcast the block to all nodes.
6. Nodes accept the block only if all transactions are valid, not duplicated, and VDF proof is correct.
7. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

5.1 Transaction Ordering and Comparison with Solana

Within each 10-minute epoch, the leader uses Proof of History (PoH) for transaction ordering. PoH is sequential SHA-256 hashing: $h_n = \text{SHA256}(h_{n-1} \parallel \text{data})$. Each hash proves time passed between h_{n-1} and h_n , creating a verifiable order of events.

Critical difference from Solana: in Solana, PoH is used for consensus and requires high-performance hardware (128GB RAM, 12-core CPU, creating a barrier to entry of $\sim \$5,000+$). In Proof of Time, PoH is used only for ordering within the already-selected leader’s epoch. Consensus is determined by VDF + weighted probabilities, which can be computed by nodes with minimal requirements.

Parameter	Solana	Proof of Time
Consensus mechanism	PoH	VDF + weighted probability
TPS	50,000+	$\sim 10,000$ (estimate)
Finalization	$\sim 400\text{ms}$	10 minutes
Hardware barrier	High ($\$5,000+$)	Low
Validators	$\sim 1,500$	Theoretically unlimited

Table 1: Comparison of Solana and Proof of Time architectures

The system sacrifices sub-second finalization for decentralization. Anyone can run a node on ordinary hardware and accumulate influence through time presence, not through capital or expensive equipment.

Nodes always consider the longest chain to be correct. The network requires a minimum of 3 active nodes to maintain consensus.

6 Incentive

By convention, the first transaction in a block is a special transaction that starts a new temporal reward owned by the block creator. This adds an incentive for nodes to support the network and provides a way to initially distribute temporal units into circulation, since there is no central authority to issue them.

Rewards are emitted on schedule:

- Blocks 1–210,000: $R = 3,000$ seconds (50 minutes)
- Blocks 210,001–420,000: $R = 1,500$ seconds (25 minutes)
- Blocks 420,001–630,000: $R = 750$ seconds (12.5 minutes)

Total emission:

$$\sum_{n=0}^{\infty} 210,000 \times \frac{3,000}{2^n} = 1,260,000,000 \text{ seconds} = 21,000,000 \text{ minutes} \quad (5)$$

The incentive can also be funded with transaction fees. Minimum fee—1 second of time. Once a predetermined number of temporal units have entered circulation, the incentive can transition entirely to transaction fees.

7 Privacy

The traditional banking model achieves a level of privacy by limiting access to information to involved parties and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous.

As an additional firewall, all transactions use ring signatures (LSAG) and stealth addresses. Ring signatures make it cryptographically impossible to determine which output from a set was actually spent. Stealth addresses ensure each transaction creates a unique one-time address, preventing linking of multiple payments to one recipient.

Transaction amounts are hidden through RingCT (Ring Confidential Transactions). Pedersen commitments prove validity without revealing amounts:

$$C = aG + bH \quad (6)$$

where a is amount, b is blinding factor. Range proofs guarantee $a > 0$ without revealing a .

8 Calculations

We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. The success event is the honest chain being extended by one block, increasing its lead by +1, and the failure event is the attacker's chain being extended by one block, reducing the gap by -1.

The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem:

- p = probability an honest node finds the next block
- q = probability the attacker finds the next block
- q_z = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \leq q \\ (q/p)^z & \text{if } p > q \end{cases} \quad (7)$$

Given our assumption $p > q$, the probability drops exponentially as the number of blocks the attacker must catch up with increases.

For a node with maximum saturation of all components:

$$P_{\max} = \frac{0.60 \times 1 + 0.20 \times 1 + 0.20 \times 1}{Z} = \frac{1.0}{Z} \quad (8)$$

With N active nodes at maximum weight: $P_{\max} = 1/N$. Thus $q = 1/N$ for an attacking node. For a 51% attack, control of $\geq 51\%$ of $\sum P_i$ is required.

9 Emission Schedule

Total blocks: 6,930,000 blocks. Blocks generated every 10 minutes (600 seconds). Calendar duration: $6,930,000 \times 10 \text{ minutes} = 69,300,000 \text{ minutes} = 131.7 \text{ years}$.

Emission per era e is calculated:

$$E(e) = 210,000 \times \frac{3,000}{2^{e-1}} \text{ seconds} \quad (9)$$

- Era 1: $E(1) = 210,000 \times 3,000/1 = 630,000,000$ seconds
- Era 2: $E(2) = 210,000 \times 3,000/2 = 315,000,000$ seconds
- Era 3: $E(3) = 210,000 \times 3,000/4 = 157,500,000$ seconds

$q = 0.10$		$q = 0.30$	
z	P	z	P
0	1.0000000	0	1.0000000
5	0.0009137	5	0.1773523
10	0.0000012	10	0.0416605
20	≈ 0	20	0.0024804

Table 2: Probability of attacker catching up from z blocks behind

Total emission:

$$\sum E(e) = 630,000,000 \times (1 + 1/2 + 1/4 + \dots) = 1,260,000,000 \text{ seconds} = 21,000,000 \text{ minutes} = 350,000 \text{ hours} = 14,166.67 \text{ days} \quad (10)$$

Temporal compression ratio: 40 years of asset / 131.7 years of calendar time ≈ 0.304 . In the initial epoch, the network produces time-asset at a 5:1 ratio (50 minutes reward / 10 minutes block interval), which decreases with each halving.

10 Sybil Resistance

Probability normalization prevents Sybil attacks from increasing influence. If an attacker creates N Sybil nodes, each starts with zero reputation and zero uptime. Probability of each Sybil node: $P_{\text{sybil}} \approx 0$ until time accumulates.

Median rate of node connections M is tracked in time windows $W = 2,016$ blocks. If rate of new node connections $N_{\text{new}}/W > 2M$, the network places new nodes on 180-day probation: $P_{\text{new}} = P_{\text{calculated}} \times 0.1$ for 180 days.

Sybil attack cost to achieve 51% influence:

- Time: N nodes \times 180 days each = $N \times 15,552,000$ seconds accumulated uptime
- Storage: N nodes \times 80% chain history each
- Reputation: N nodes \times 2,016 signed blocks each
- Risk: Equivocation detection \rightarrow slashing all N nodes \rightarrow loss of all time investments

11 Network Architecture

The network requires a minimum of 3 active nodes for consensus functioning. Full nodes ($\geq 80\%$ history) participate in leader selection and governance. Light nodes (last 2,016 blocks) have reduced probability $P_{\text{light}} = P_{\text{calculated}} \times 0.5$.

Distribution follows Bitcoin model: DNS seeds for bootstrap, addr messages for peer discovery, gossip protocol for transactions and blocks. Synchronization: headers-first with IBD (Initial Block Download). Transactions are ordered through PoH slots within 10-minute epochs.

VRF selects leader:

$$H(\text{prev_block_hash} \parallel \text{node_pubkey}) \bmod Z < P_i \times Z \quad (11)$$

Node with smallest valid hash becomes leader. Leader computes VDF proof and broadcasts block. If leader is offline for more than 12 minutes, next VRF candidate takes leadership.

12 Conclusion

We have proposed a system for temporal transactions without reliance on trust in capital. We started with the usual framework of cryptographic signatures, which provides strong control of ownership, but is incomplete without a way to prevent plutocratic capture. To solve this, we proposed a peer-to-peer network using proof-of-time to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of weighted influence. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the longest proof-of-time chain as proof of what happened while they were gone. They vote with their accumulated time, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

References

- [1] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008.
- [2] D. Boneh, J. Bonneau, B. Bünz, B. Fisch, *Verifiable Delay Functions*, Advances in Cryptology – CRYPTO 2018, pp. 757–788, 2018.
- [3] B. Wesolowski, *Efficient Verifiable Delay Functions*, Advances in Cryptology – EUROCRYPT 2019, pp. 379–407, 2019.
- [4] A. Yakovenko, *Solana: A new architecture for a high performance blockchain*, 2018.
- [5] S. Noether, *Ring Signature Confidential Transactions for Monero*, Ledger, vol. 1, pp. 1–18, 2016.
- [6] N. van Saberhagen, *CryptoNote v2.0*, 2013.
- [7] M. Bellare, P. Rogaway, *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols*, Proceedings of the 1st ACM Conference on Computer and Communications Security, pp. 62–73, 1993.
- [8] W. Feller, *An Introduction to Probability Theory and Its Applications*, Vol. 1, Wiley, 1957.