Proof of Time: Addendum

DAG Architecture and Tiered Privacy Model

Version 1.1 — December 2025

Abstract

This addendum extends the Proof of Time consensus mechanism with a Directed Acyclic Graph (DAG) structure for horizontal scalability and a tiered privacy model that balances transaction throughput with cryptographic confidentiality. The proposed architecture enables linear TPS scaling with network size while maintaining the core principle that influence derives from time presence rather than capital expenditure.

11. DAG Architecture

The original Proof of Time specification describes a linear blockchain where each block references exactly one predecessor. While this simplifies consensus, it creates a throughput bottleneck: only one block can be produced per 10-minute interval regardless of network size.

We extend the protocol to a Directed Acyclic Graph (DAG) structure where multiple blocks can be produced concurrently, with VDF-weighted ordering determining canonical transaction sequence.

11.1 Block Structure

Each DAG block contains:
• **parents[]** — References to 1-8 parent blocks (vs. single parent in linear chain)
• **vdf_proof** — Verifiable Delay Function proof
• **vdf_weight** — Accumulated VDF difficulty from genesis
• **node_weight** — Producer's time + space + reputation weight
• **transactions[]** — Ordered transaction set
• **timestamp** — Unix timestamp (soft constraint: ±30 seconds from VDF expectation)

11.2 Concurrent Block Production

Unlike linear chains where leader election produces one block per interval, DAG architecture allows any node meeting minimum weight threshold to produce blocks:

```
can_produce(node) = (w_time × f_time + w_space × f_space + w_rep × f_rep) ≥ θ_min

Where θ_min = 0.1 (10% of maximum possible weight)
```

Nodes above threshold produce blocks as transactions arrive, referencing all known unconfirmed block tips as parents. This creates a DAG structure where the network converges on transaction ordering through VDF weight accumulation.

11.3 PHANTOM-PoT Ordering

We adapt the PHANTOM protocol for DAG ordering, replacing proof-of-work with VDF weight. The algorithm identifies a "blue set" of well-connected honest blocks and orders transactions by accumulated weight:

```
1. For each block B, compute anticone(B) = blocks neither ancestor nor
descendant of B
2. Block B is "blue" if |anticone(B) ∩ blue_set| ≤ k (default k = 8)
3. Order blue blocks by cumulative_vdf_weight descending
4. Insert red blocks (non-blue) between blue ancestors and descendants
5. Transaction order follows block order; conflicts resolved by first-seen in
ordered sequence
```

11.4 Scalability Properties

DAG architecture provides linear horizontal scaling:

Active Nodes	Blocks/min	Est. TPS	Storage/year
100	~50	~5,000	~5 GB
1,000	~500	~50,000	~50 GB
10,000	~5,000	~500,000	~500 GB
100,000	~50,000	~5,000,000	~5 TB

Note: TPS estimates assume average transaction size of 250 bytes (public) to 2.5 KB (private). Actual throughput depends on privacy tier distribution.

12. Tiered Privacy Model

The original specification mandates Ring Confidential Transactions (RingCT) for all transactions. While this maximizes privacy, it constrains throughput: each RingCT transaction requires ~2.5 KB and ~40ms verification time.

We introduce a tiered privacy model where users select confidentiality level per transaction, enabling high-throughput public transactions while preserving full privacy as an option.

12.1 Privacy Tiers

Tier	Privacy Level	Size	Verify Time	Fee Multiplier
T0: Public	Addresses + amounts visible	~250 B	~0.5 ms	1×
T1: Stealth	One-time addresses, amounts visible	~400 B	~1 ms	2×
T2: Confidential	Stealth + hidden amounts (Pedersen)	~1.2 KB	~8 ms	5×
T3: Ring	Full RingCT (ring size 11)	~2.5 KB	~40 ms	10×

12.2 Tier Specifications

Tier 0 (Public): Standard UTXO model similar to Bitcoin. Addresses and amounts are visible on-chain. Suitable for merchant payments, exchanges, and applications where transparency is desired or required.

Tier 1 (Stealth): Each transaction generates a one-time destination address using Diffie-Hellman key exchange. Recipient derives private key from shared secret. Amounts remain visible, but address linkage is broken.

```
R = r·G (ephemeral public key, included in tx)
P' = H(r·A)·G + B (one-time address)
Where (A, B) is recipient's public key pair, r is random scalar
```

Tier 2 (Confidential): Combines stealth addresses with Pedersen commitments for amount hiding. Uses Bulletproofs++ for range proofs, reducing proof size to ~512 bytes for 64-bit values.

```
C = a·H + b·G (Pedersen commitment)
Where a = amount, b = blinding factor
Range proof: Bulletproofs++ proves a ∈ [0, 2^64) without revealing a
```

Tier 3 (Ring): Full Ring Confidential Transactions with LSAG signatures. Provides sender anonymity through ring of decoy inputs. Default ring size = 11 (1 real + 10 decoys).

12.3 Cross-Tier Transactions

Transactions may have inputs and outputs at different privacy tiers with constraints:

• Outputs can be any tier ≥ input tier (privacy can increase, not decrease)
• T0 → T3 is valid (public to private)
• T3 → T0 is invalid (would deanonymize ring participants)
• Mixed-tier outputs allowed within same transaction
• Fee component always public (T0) for verification

12.4 Anonymity Set Considerations

Tiered privacy creates potential anonymity set fragmentation. Mitigation strategies:

1. Decoy Selection Pool: Ring signatures (T3) select decoys from T2 and T3 outputs only, maintaining commitment format compatibility.

2. Minimum Ring Size: Enforced minimum ring size of 11 regardless of available same-tier outputs. Network pads with synthetic decoys if necessary during bootstrap.

3. Tier Distribution Targets: Protocol recommends wallet defaults that maintain healthy distribution: T0 (40%), T1 (30%), T2 (20%), T3 (10%).

13. Storage Architecture

Horizontal scalability requires efficient storage that can handle high write throughput while maintaining fast read access for transaction verification.

13.1 Layered Storage Model

Hot Layer (RAM + NVMe SSD):
• Recent DAG tips (last 100 blocks per active branch)
• UTXO set for all tiers
• Node state table (time_presence, reputation, space_score)
• Mempool with fee-priority indexing

Warm Layer (SSD):
• Blocks from last 30 days
• Transaction index (txid → block location)
• VDF proof cache for recent verification

Cold Layer (HDD / Archival):
• Full block history
• Historical VDF proofs
• Prunable after checkpoint finalization

13.2 Database Selection

Recommended database engines by component:

Component	Engine	Rationale
UTXO Set	MDBX	Memory-mapped, ACID, excellent read performance
Block Storage	RocksDB	LSM-tree optimized for write-heavy workloads
DAG Index	Custom B+ tree	Optimized for parent/child traversal
Mempool	In-memory skiplist	O(log n) insert with fee ordering
VDF Cache	LRU + RocksDB	Hot proofs in RAM, overflow to disk

13.3 Serialization Format

All on-disk and network serialization uses FlatBuffers for zero-copy parsing:

• **Zero-copy access:** Read fields directly from buffer without deserialization
• **Schema evolution:** Forward/backward compatible field additions
• **Performance:** 10-100× faster than Protocol Buffers for read-heavy workloads
• **Size efficiency:** Comparable to Protobuf, smaller than JSON/XML

13.4 Pruning and Checkpoints

To bound storage growth, nodes may prune historical data after checkpoint finalization:

Checkpoint Creation: Every 10,000 blocks (~70 days), network produces checkpoint containing UTXO set commitment, node state merkle root, and cumulative VDF weight.

Pruning Rules:
• Full nodes: Must retain ≥80% of post-checkpoint history (as per original spec)
• Archival nodes: Retain complete history, serve historical sync requests
• Light nodes: Retain only checkpoint + subsequent blocks

Space Score Calculation: f_space(s) now computed as min(s / 0.80 × post_checkpoint_history, 1)

14. Updated Security Analysis

14.1 DAG-Specific Attack Vectors

Balancing Attack: Attacker attempts to maintain two competing DAG branches to enable double-spending. Mitigation: PHANTOM ordering ensures convergence; transactions in both branches receive same ordering relative to conflict resolution.

Spam Attack: Attacker floods network with low-value blocks to increase orphan rate. Mitigation: Minimum weight threshold ($\theta_{min} = 0.1$) prevents new/low-weight nodes from block production. 180-day saturation period limits attacker throughput.

Parasitic Mining: Attacker references only own blocks as parents, attempting to create isolated subgraph. Mitigation: Blocks with insufficient parent diversity (< 3 unique parent producers) receive 50% weight penalty.

14.2 Privacy Tier Attack Vectors

Tier Downgrade Attack: Attacker attempts to force T3 outputs into lower tiers. Mitigation: Protocol enforces output_tier ≥ max(input_tiers). Violations rejected at consensus layer.

Intersection Attack: Statistical analysis of tier transitions to deanonymize users. Mitigation: Wallet software implements tier stickiness—outputs default to same tier as inputs unless user explicitly requests upgrade.

Timing Analysis: Correlating transaction timing with real-world events. Mitigation: Wallets implement random delay (0-60 seconds) before broadcast. Not protocol-enforced but strongly recommended.

14.3 Throughput vs. Security Trade-offs

DAG architecture introduces confirmation latency variance. Security recommendations:

Transaction Value	Recommended Confirmations	Expected Time
< 100■	1 confirmation	~1-2 minutes
100-10,000■	3 confirmations	~5-10 minutes
10,000-1,000,000■	6 confirmations	~15-30 minutes
> 1,000,000■	12+ confirmations	~60+ minutes

15. Conclusion

This addendum extends Proof of Time with two complementary enhancements: DAG-based block structure for horizontal scalability, and tiered privacy for throughput optimization.

The core principle remains unchanged: consensus influence derives from time presence, not capital expenditure. A student node operating for 180 days achieves the same weight as a corporate datacenter. This property is preserved—and amplified—by DAG architecture, which allows more participants to produce blocks without concentrating power in high-throughput nodes.

Tiered privacy acknowledges that not all transactions require maximum confidentiality. By offering choice, the network achieves higher throughput while maintaining strong privacy guarantees for users who need them. The fee multiplier structure ensures that privacy-heavy transactions compensate for their verification cost.

Together, these extensions enable Proof of Time to scale from the original ~10 TPS to potentially millions of TPS as the network grows—while remaining accessible to anyone with consumer hardware and six months of time.

References

[9] Y. Sompolinsky, A. Zohar, "PHANTOM: A Scalable BlockDAG Protocol," IACR ePrint 2018.

[10] B. Bünz et al., "Bulletproofs: Short Proofs for Confidential Transactions," IEEE S&P; 2018.

[11] P. Todd, "Stealth Addresses," Bitcoin Wiki, 2014.

[12] Google, "FlatBuffers: Memory Efficient Serialization Library," 2014.

[13] H. Chu, "LMDB: Lightning Memory-Mapped Database," OpenLDAP, 2011.