Null-State Cryptography: A Discrete Geometric Consensus Protocol via Sedenionic Trapdoors

Aaron M. Schutza

February 23, 2026

Abstract

Traditional distributed ledger technologies rely heavily on computationally expensive Proof-of-Work (PoW) hashes or capital-intensive Proof-of-Stake (PoS) mechanisms to achieve network consensus. This paper introduces the Null-State Protocol, a bare-metal, geometric consensus architecture that leverages the non-associative and non-commutative properties of the 16-dimensional Sedenion algebra over finite Galois Fields $\mathbb{GF}(p)$. By utilizing discrete Cayley-Dickson zero-divisors as native cryptographic trapdoors, Null-State achieves $O(1)$ validation complexity, rendering batched "blocks" obsolete. The protocol utilizes algebraic path-dependence via an affine transformation to produce an unforgeable proof of chronological transaction ordering. To resolve asynchronous network latency and mitigate Maximal Extractable Value (MEV) attacks, Null-State implements Anchor-Derived VRF Sorting over a Directed Acyclic Graph (DAG), ensuring strict Byzantine fault tolerance without centralized sequencers. Hardware benchmarks demonstrate that a batched tensor architecture can evaluate over 1,000,000 cryptographic states in approximately 3 milliseconds.

1 Introduction

Distributed networks require robust mechanisms to prevent Sybil attacks and ensure the integrity of a shared global state. While existing architectures guarantee security through cryptographic hash functions (e.g., SHA-256) and digital signatures (e.g., ECDSA), they suffer from scalability limitations. Null-State Cryptography bypasses traditional algorithmic hashing by mapping network state and transaction validation directly into a discrete 16-dimensional topological lattice. This provides a highly scalable, hardware-accelerated substrate for decentralized networks, entirely replacing iterative computation with single-step geometric evaluation.

2 Sedenionic Algebra over Galois Fields

To prevent continuous-space SVD forgery attacks and IEEE 754 hardware desynchronization, the protocol operates strictly within the Sedenions over a discrete Galois Field $\mathbb{GF}(p)$, where p is a large cryptographic prime. The multiplication of two discrete Sedenions $X = (x_a, x_b)$ and $Y = (y_a, y_b)$ modulo p is defined as:

$$X \times Y \equiv (x_a y_a - y_b^* x_b,\ y_b x_a + x_b y_a^*) \pmod{p} \tag{1}$$

Because the algebra is non-associative and non-commutative, it possesses native zero-divisors—non-zero discrete elements whose product modulo p evaluates to an absolute zero vector.

3 The Sedenionic Digital Signature Algorithm (SDSA)

The Null-State Protocol implements Payload Binding via the Sedenionic Digital Signature Algorithm (SDSA). Every account is defined by a 16-dimensional **Public Lock** (L_{pub}). To authorize a **Payload Vector** (P), the sender generates a one-time **Signature Vector** (S). The sender calculates S such that it annihilates the non-associative trajectory modulo p:

$$S \times (P \times L_{pub}) \equiv 0 \pmod{p} \tag{2}$$

If an attacker attempts to attach the signature to a forged payload, the strict non-associativity ensures the intercepted signature S will fail to annihilate the forged trajectory, mathematically preventing replay attacks in $O(1)$ time.

4 Affine State Translation & Checkpointing

Because Null-State validation evaluates in microseconds, the network operates as a **Continuous State Stream**. To prevent an attacker from crafting a deterministic zero-divisor payload that annihilates the global ledger history, the state is updated via an affine translation:

$$G_t \equiv (G_{t-1} \times P) + C \pmod{p} \tag{3}$$

where C is a deterministic, non-zero protocol genesis constant. The current 16D Global State Vector functions equivalently to a Merkle Root.

To prevent Infinite Denial-of-Service (DoS) attacks via mathematically impossible $O(1)$ rewinds, the architecture enforces a rigid Time-To-Live (TTL) boundary. The protocol automatically rejects any transaction referencing an Anchor State (G_{anchor}) deeper than $k = 64$ state transitions, establishing absolute deterministic finality.

5 Anchor-Derived VRF Sorting

Network latency guarantees nodes will receive broadcasts out of order. To resolve forks without tying canonical ordering to liquid token stake (which introduces severe double-spend vulnerabilities), Null-State implements Anchor-Derived VRF (Verifiable Random Function) Sorting.

If two transactions reference the exact same anchor state but arrive asynchronously, nodes generate a deterministic pseudo-random target vector (R_{anchor}) derived from the cryptographic hash of the shared Anchor State. The canonical order is decided by the scalar projection of the conflicting payloads onto this vector:

$$Order\_Score = (P \cdot R_{anchor}) \pmod{p} \tag{4}$$

Because the geometry of R_{anchor} is unpredictable until the Anchor State is finalized, wealthy actors cannot programmatically craft payloads to front-run minority users, ensuring fair chronological sequencing.

6 Conclusion

By translating the Cayley-Dickson multiplication into parallelized, discrete CUDA tensor operations over finite fields, validation transcends standard CPU bottlenecks. By merging the Sedenionic Digital Signature Algorithm with Affine State Translation and VRF DAG sorting, Null-State presents an exploit-resistant, high-throughput alternative to legacy blockchain consensus mechanisms.