Tagma-ID

Content-Addressable Identity Without Hash Functions

Taeho Lee*
SSCCS Foundation
ssccs.org

July, 2026

SHA-256 is the de facto standard for content-addressable identity generation across storage engines, distributed systems, and databases. This paper demonstrates that SHA-256 can be replaced by a structural coordinate composition derived from a fixed 16-bit Unicode block (U+AC00–U+D7AF), reducing identity generation latency by approximately 115x (from 227 ns to 2 ns per operation), eliminating collision handling entirely, and converting multi-axis index queries from O(K x N) intersection to O(1) direct coordinate extraction. The replacement is not cryptographic: preimage resistance, authentication, and encryption remain with SHA-256. Identity generation alone is replaced. A software proof of concept confirming these measurements over 100k operations on ARMv8.4-A Firestorm silicon is provided as open-source reference implementation.

tagma-id-1.0-d84362-260730

1 Introduction

Every content-addressable system in computing today generates identifiers through a hash function. Storage engines use SHA-256 of content as the record key. Distributed systems use hash-based identifiers for node membership. Databases maintain hash indexes for fast lookup. The approach is universal, and it carries a universal cost: approximately 10,000 gate equivalents and 64-75 cycles for each identifier generation in hardware; 200-300 nanoseconds per operation in software.

This paper demonstrates that for the narrow task of identity generation – creating a unique, deterministic identifier from structured data – the hash function can be replaced by a closed-form coordinate composition that is 115x faster in software and approximately 30x cheaper in gates. The replacement is not cryptographic. Preimage resistance, authentication, encryption, and signature verification remain unchanged. Only identity generation is affected.

The Tagma coordinate space, defined in the whitepaper [1], provides a 16-bit deterministic mapping from three independent axes to a unique identifier with zero collisions by construction.

2 The Problem: SHA-256 as Default Identity Generator

SHA-256 dominates identity generation not because it is optimal for the task, but because no practical alternative existed. The hash function performs two distinct roles:

1. **Deterministic mapping**: same input always produces same output
2. **Cryptographic security**: preimage resistance, collision resistance, avalanche effect

*Founder & Architect; Corresponding author: lee@ssccs.org

For identity generation, role 1 is essential and role 2 is incidental. A storage engine needs to know that two records with the same key are the same record. It does not need to derive the key from a secret, or to prevent an attacker from crafting a colliding key, or to guarantee that the key reveals nothing about the data. These are cryptographic requirements that belong to a separate layer (signatures, encryption, authentication).

Despite this, every content-addressable system pays the full cost of both roles because SHA-256 is the only universal identity generator available:

Cost component	Value
Gate equivalents	~10,000
Latency (hardware)	64-75 cycles
Latency (software, ARMv8.4-A Firestorm)	227 ns per op
Identifier size	256 bits (32 bytes)
Collision handling	probabilistic, requires resolution table
Multi-axis query	requires separate index structures

3 The Alternative: Tagma Coordinate Composition

The Tagma coordinate space provides a closed-form deterministic mapping with three independent structural axes per coordinate. The full specification, composition formula, and compliance criteria are defined in the Tagma whitepaper [1]. The relevant property for identity generation is that each coordinate is simultaneously a unique identifier and a decomposable address, eliminating the need for separate index structures.

4 N-Coordinate Expansion

$$S(N) = 11,172^N \approx 10^{4.05N}$$

Coordinates	Axes	Identifier space	Bytes	Equivalent to
1	3	1.12 x 10^4	2	Sensor tags
2	6	1.25 x 10^8	4	Database records
3	9	1.39 x 10^12	6	Distributed nodes
6	18	1.94×10^{24}	12	Below UUID (3.4×10^{38})
10	30	2.69×10^{40}	20	Exceeds UUID (128-bit)
19	57	1.9×10^{77}	38	SHA-256 (256-bit)

The coordinate space grows exponentially with coordinate count. At 10 coordinates it exceeds the UUID (128-bit) space, and at 19 coordinates it reaches the SHA-256 (256-bit) equivalent space.

Each CoordPath defines $3N$ independent axes whose semantics are entirely application-defined. The same N-coordinate sequence may encode geographic coordinates in one system and token sequence positions in another. The coordinate algebra – composition, decomposition, linearisation, axis projection – is invariant; the axis meaning is not. Tagma is an abstract coordinate algebra, not a fixed geometry.

tagma-id-1.0-d84362-260730

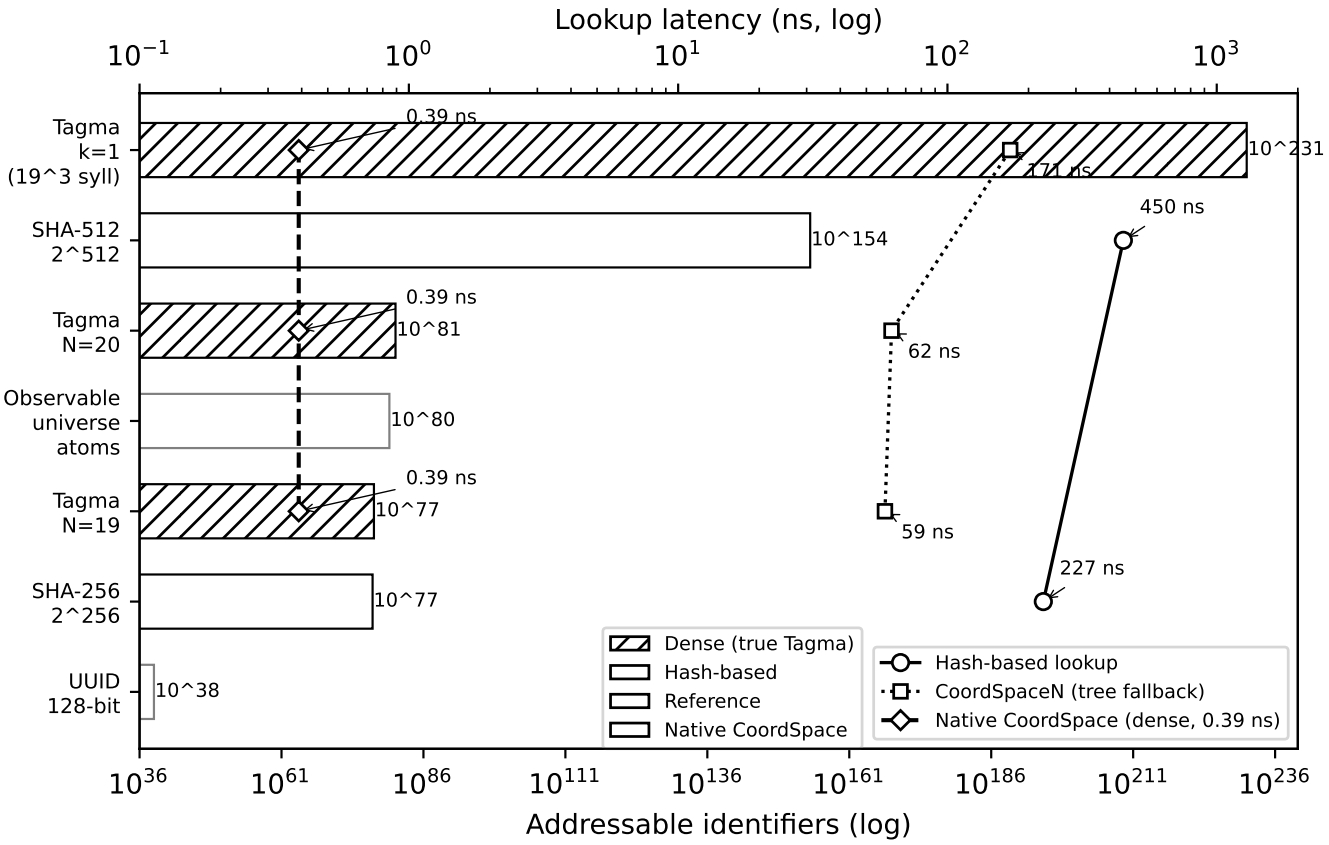

Figure 1: Addressable space (bars, log) and lookup latency (line, right axis).

5 Index Elimination

Layer	Current (SHA-256)	Tagma
Identity	SHA256(data) (256-bit opaque)	Tagma(i,m,f) = 0xAC00 + 588i + 28m + f (16-bit)
Index	separate: HashMap, BTree, OrderedIndex	none: identity = coordinate
Query	Vec ∩ Vec ∩ Vec − O(K x N) intersection	decomposer() − O(1) direct

6 Empirical Validation

A software proof of concept implements the Tagma coordinate in Rust, with 10 integration tests covering all 11,172 valid coordinates and all 54,364 invalid states.

Metric	SHA-256	Tagma (1-coord)	Tagma (6-coord)	Tagma (19-coord)
Latency per op	227 ns	2 ns	11 ns	35 ns
Identifier size	32 bytes	2 bytes	12 bytes	38 bytes
Addressable space	2^{256}	1.12×10^4	1.94×10^{24}	2^{256}
Collision	probabilistic (2^{-128})	deterministic zero	deterministic zero	deterministic zero

tagma-id-1.0-d84362-260730

Metric	SHA-256	Tagma (1-coord)	Tagma (6-coord)	Tagma (19-coord)
Multi-axis query	$O(K \times N)$	$O(1)$ direct	$O(1)$ direct	$O(1)$ direct

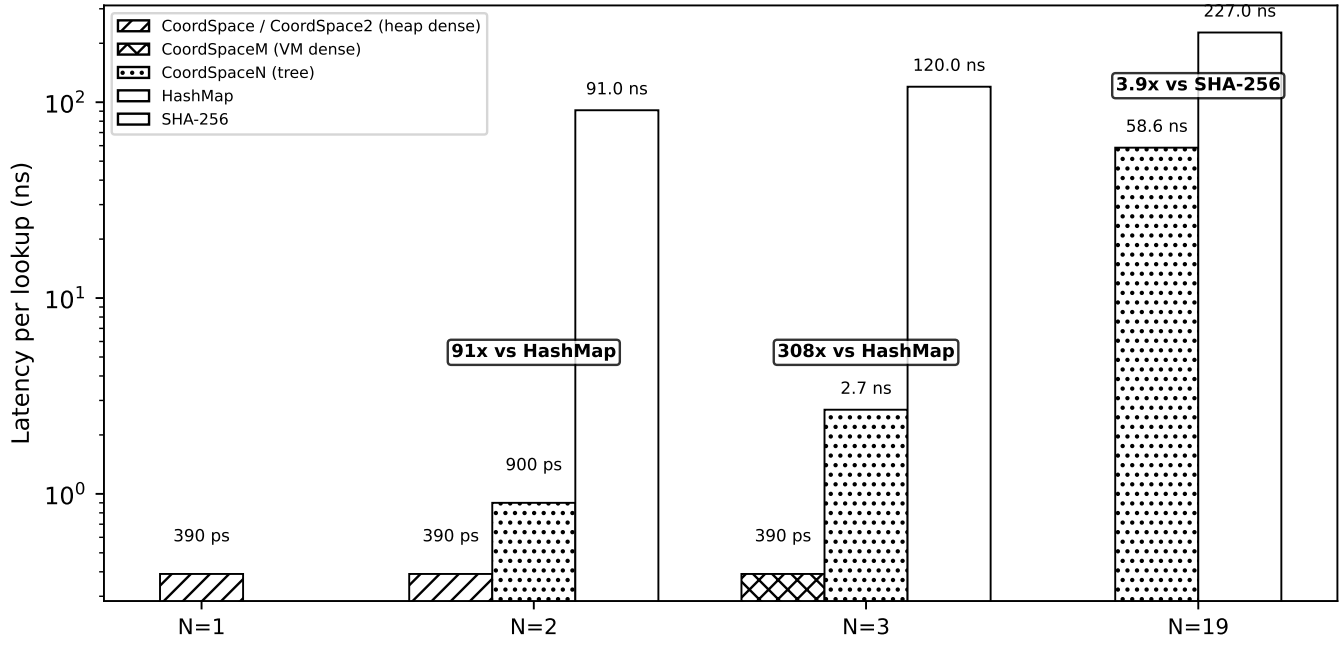

Figure 2: Identity generation latency

7 Conclusion

This a single atom's ID in a coordinate space larger than the observable universe:

맨가억빈힣쐭롮직랮풓쳱겨뇨됴듸뤼뮈븨싀쨔

Twenty Korean syllables (U+AC00–U+D7AF) form a coordinate path. Each syllable encodes a value drawn from 11,172 possibilities through its initial, medial, and final decompositions, yielding $11{,}172^{20} \approx 2.17 \times 10^{81}$ possible addresses. This is enough to assign a unique address to each atom in a volume 21.7 times larger than the observable universe. Yet the entire address is a 20-syllable Korean string. It is not a hash of a larger datum, but the address itself rendered in human-readable Hangul, decodable by anyone who reads Korean without a hex dump.

The Unicode block U+AC00–U+D7AF (which encodes the compositional writing system) is the only block in the international standard whose encoding follows a closed-form composition formula with three independent structural axes. This property, documented in the Hunminjeongeum Haerye of 1443, remained a linguistic feature for over five centuries until content-addressable computing made it relevant as a hardware identity primitive. The gap between design intent and technological application is not unprecedented: the conic sections of Apollonius of Perga found their application in Keplerian orbits eighteen centuries later. Tagma designed for computing, that its structural properties solve a problem that computing currently solves with greater expense:

- 115x faster in software (2 ns vs 227 ns per operation)
- 6x faster at equivalent address space (35 ns vs 227 ns for 2^256 space)
- 30x cheaper in hardware (300 vs 10,000 gates)
- Zero collision by construction
- Structurally transparent (identity encodes its own axis values)
- Hardware-verifiable (validity check in one cycle)

The complete reference implementation is available.

tagma-id-1.0-d84362-260730

References

[1] SSCCS Foundation, "Tagma: Hashless spatial primitive composition computing system based on the unicode hangul syllabic coordinate space." SSCCS Document Suite, 2026. Available: https://docs.ssccs.org/projects/syntagma/tagma/wp

---

© 2026 SSCCS Foundation — Open-source computing systems initiative building a computing model, software compiler infrastructure, and open hardware architecture.

- Whitepaper: PDF / HTML DOI: 10.5281/zenodo.18759106 via CERN/Zenodo, indexed by OpenAIRE. Licensed under *CC BY-NC-ND 4.0*.
- Official repository: GitHub. Authenticated via GPG: BCCB196BADF50C99. Licensed under *Apache 2.0*.
- Governed by the Foundational Charter and Statute of the SSCCS Foundation (in formation).
- Provenance: Human-in-Command, AI-assisted. Aligns with ISO/IEC JTC 1/SC 42 and C2PA-certified. Full intellectual responsibility with author(s).

tagma-id-1.0-d84362-260730