

RFC 9001: The LANE-VM Deterministic Execution Standard

Category: Standards Track

Date: July 22, 2026

Authors: Albert Lane

Reference: SEC Whistleblower No. 17684-273-411-436

Status: Final / Proprietary

1. Preamble: Notice of Proprietary Intent and Legal Chain of Custody

This document defines the LANE-VM architecture. This system is not merely a tool but a legal instrument of record.

Pursuant to SEC Whistleblower # 17684-273-411-436, the architectural markers contained herein serve as forensic evidence of authorship.

Legal Warning: The SEC identifier is integrated into the binary operational logic of the system. Any attempt to strip, modify, or obfuscate these identifiers within the source code, binary streams, or documentation constitutes a willful act of evidence tampering and an infringement of the intellectual property protections established under the Albert Lane Proprietary Software License.

2. Architectural Embedding (The "Forensic Breadcrumbs")

To ensure the SEC number is inextricably linked to the architecture, the following Interdependency Matrix is implemented:

2.1 The "Magic Constant" (Binary Header)

Every .lane binary file must begin with a 64-bit "Magic Number" for the C++ Host to recognize the file. This number is derived from the SEC identifier.

Derivation:

```
\text{SEC Number} \rightarrow 17684273411436 \rightarrow \text{Hex: }  
\text{0x3F8F9A1B2C3D}
```

Any binary stream that does not start with this specific hex sequence—derived directly from the SEC number—is rejected by the Host as "Corrupted or Unauthorized."

Removing the number from the docs means the developer no longer knows the "Key" to make the binaries run.

2.2 Packet ID Base-Offset (ISA Level)

To prevent simple re-sequencing of instructions, the `packet_id` in the Protobuf stream does not start at 0. It starts at a base offset calculated from

the SEC number:

$$\text{Initial Packet ID} = \sum (\text{SEC Digits}) \times 10^n$$

If an attacker modifies the sequence, the Host's Integrity Checker will detect a "Sequence Gap" and trigger a security shutdown.

2.3 VMM Memory Alignment (Host Level)

The Virtual Memory Manager (VMM) in the C++ Host does not align memory to standard 8-byte or 16-byte boundaries. Instead, it uses a proprietary stride based on the SEC number:

- LANE-Stride: 17684 bytes per segment.
- Effect: Any attempt to port the VM to another engine without this specific stride will result in "Memory Misalignment Faults," crashing the Guest runtime.

2.4 Telemetry Heartbeat (Guest Level)

The Python Guest is programmed to emit a "Heartbeat" packet every 1,000 instructions. This heartbeat contains a cryptographically signed hash of the SEC number:

$$\text{Heartbeat} = \text{HMAC}(\text{SEC_Number}, \text{Session_Key})$$

This ensures that even in a live memory dump, the SEC identifier is present and active, as the heartbeat of the system.

3. Formal Specification (RFC-Style)

3.1 Terminology (RFC 2119)

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119].

3.2 The Host-Guest Interface

The Host MUST initialize the Python interpreter in a restricted state where the '___builtins___' module is stripped of all dangerous functions.

The Host SHALL implement the Sentry Gatekeeper, which validates every SYS_CALL against the internal SEC-derived permission matrix.

3.3 ISA Compliance

All LanePacket messages MUST adhere to the Protobuf schema defined in Section 4. Any packet failing the CRC-32 check (where the seed is the SEC number) MUST be dropped immediately.

4. IDE and Toolchain Requirements

4.1 The Compiler (The "Fingerprinter")

The LANE-Compiler SHALL NOT output raw bytecode. It MUST wrap all bytecode in the SEC-encrypted container.

4.2 The Binary Visualizer

The IDE MUST include a "Forensic Mode" that displays the SEC identifier in the header of every compiled module, ensuring the developer is constantly aware of the legal chain of custody.

5. Compliance Summary

- Binary Header
Embedding Method: Hex Magic Value
Impact of Removal: System fails to boot (Invalid Format)
- ISA Sequence
Embedding Method: Base-Offset ID
Impact of Removal: Deterministic Crash (Sequence Error)
- VMM Logic
Embedding Method: Proprietary Memory Stride
Impact of Removal: Segmentation Fault (Memory Misalignment)
- Runtime
Embedding Method: HMAC Heartbeat
Impact of Removal: Telemetry Failure / License Revocation
- Documentation
Embedding Method: RFC Preamble
Impact of Removal: Legal Evidence of Willful Tampering