

NERV Wallet Protocol Specification
A Private, Post-Quantum Light Wallet for the Neural State Embedding Blockchain

Version 1.0 – 17 December 2025

Open-source • MIT/Apache 2.0 • Community-governed

<https://github.com/nerv-bit/wallet-spec>

Abstract

NERV wallets provide users with full control over private funds in a completely shielded environment. No public addresses, balances, amounts, or transaction metadata are ever exposed on-chain. Wallets reconstruct balances and history locally using cryptographic detection of **private notes** (shielded value commitments) and permanent **Verifiable Delay Witnesses (VDWs)** for inclusion proofs.

This specification defines the wallet protocol, including:

- Key management (post-quantum hierarchical deterministic).
- Private note detection and balance computation.
- Transaction construction and onion routing.
- Light-client synchronization.
- VDW handling and offline verification.
- Memo fields and selective disclosure.

All operations preserve NERV's privacy guarantees while enabling mobile/light-client usability (<100 KB permanent sync, <80 ms VDW verification on smartphones).

Reference implementations (Rust SDK, TypeScript/WebAssembly, Swift/iOS, Kotlin/Android) are available under MIT/Apache 2.0.

Table of Contents

1. Introduction
- 1.1 Privacy Model
- 1.2 Light-Client Assumptions
- 1.3 Design Principles
2. Cryptographic Primitives
3. Key Management and Hierarchical Deterministic Wallets
4. Private Notes and Value Commitments
5. Transaction Flows
6. Light-Client Synchronization
7. Verifiable Delay Witnesses (VDWs)
8. Memos and Selective Disclosure
9. Backup and Recovery
10. Security Considerations
11. References

1. Introduction

NERV's core protocol (Whitepaper v1.01) replaces traditional state roots with 512-byte neural embeddings, enabling full privacy by default. Wallets must therefore operate without public identifiers:

- Users generate **spending keys** (Dilithium-3) and derive **detection keys** for incoming notes.
- Incoming transfers create **private notes**—blinded commitments detectable only by the recipient's wallet.
- Balances are summed from unspent notes locally.
- History is reconstructed from cached VDWs and optional encrypted memos.

This mirrors shielded pools in Zcash/Orchard but adapted to NERV's homomorphic deltas and post-quantum suite.

1.1 Privacy Model

- **No linkage:** Trial decryption of notes/deltas reveals nothing to non-owners.
- **No metadata leaks:** Onion routing + cover traffic blinds origin/timing/size.
- **Selective disclosure:** VDWs prove specific inclusions without revealing other data.

1.2 Light-Client Assumptions

Wallets are light clients:

- Permanent sync data: <100 KB (cached embedding roots).
- Delta/VDW fetching: HTTP/3 range requests (privacy-preserving via random padding).
- No full shard state download.

1.3 Design Principles

- Privacy-first, no opt-in shielding.
- Post-quantum from genesis.
- Mobile-friendly verification.
- Interoperable reference implementations.
- Seed-based recovery without exposing history.

2. Cryptographic Primitives

All primitives match the core protocol (Section 7):

- **Signatures:** CRYSTALS-Dilithium-3 (primary), SPHINCS+-SHA256-192s-robust (backup).
- **Key Encapsulation:** ML-KEM-768.
- **Hashing:** SHA3-256 + BLAKE3.

- **ZK Proofs:** Halo2 recursive + Nova folding (for tx validity).
- **Randomness:** Hardware RNG or OS sources.

3. Key Management and Hierarchical Deterministic Wallets

Wallets use a post-quantum HD scheme inspired by ZIP-32 (Zcash) but adapted for Dilithium/ML-KEM.

- **Master Seed:** 256-bit entropy → mnemonic (BIP-39 compatible, but PQ-hardened).
- **Derivation Path:** m / purpose' / coin_type' / account' / change / index
 - purpose = 32' (shielded NERV).
 - coin_type = TBD (e.g., 133' for NERV).
- **Spending Key:** Dilithium-3 sk (full authority).
- **Detection Key:** Derived for note trial-decryption.
- **Diversified Receivers:** Each address/index derives unique blinded commitment params.

No public keys are published—receivers share derived "payment codes" offline/encrypted.

4. Private Notes and Value Commitments

Transfers create **private notes**:

- Note = (value, blinding_factor, diversified_receiver_commitment, nullifier_seed)
- On-chain: Blinded commitment in homomorphic delta (undetectable to others).

Wallet detection:

- Scan recent batched deltas/VDWs.
- Trial-decrypt commitments using detection key.
- Success → note owned; add value to local unspent set.
- Nullifier (prevent double-spend) tracked locally.

Balance = Σ unspent note values.

5. Transaction Flows

Sending

1. Select unspent notes covering amount + fee.
2. Generate ZK proof of validity (sufficient value, correct nullifiers).
3. Create outputs: new private notes for recipient(s) + change.
4. Build onion (5-hop TEE routing).
5. Broadcast via light-client relay.

Receiving

- Periodic sync fetches new deltas/VDWs.
- Trial detection adds notes automatically.
- Optional encrypted memo (sender-included, decrypted locally).

6. Light-Client Synchronization

- Initial sync: Fetch genesis embedding root + recent checkpoints (<100 KB).
- Ongoing: Poll for new embedding roots (32-byte hashes).
- Delta fetching: Range requests for recent batches (privacy via chaff).
- Full rescan: Optional for recovery (still light).

7. Verifiable Delay Witnesses (VDWs)

- Auto-cached on send/receive.
- Structure: tx_hash + inclusion_proof + embedding_root + TEE attestation.
- Offline verification: Halo2 recursive check + homomorphic delta application.
- Permanent archival: Arweave/IPFS pins.

8. Memos and Selective Disclosure

- Sender attaches optional encrypted memo (AES-GCM, recipient's derived key).
- Contents: Free text (e.g., "Invoice #123").
- Selective proof: Export single VDW → prove receipt without full history.

9. Backup and Recovery

- Mnemonic seed recovers all derived keys/notes (rescan required).
- Encrypted local backup (device keystore).
- No cloud sync of private data.

10. Security Considerations

- Seed exposure → total loss.
- Trial decryption safe (no false positives).
- Reorgs: Auto-fetch replacement VDWs.
- Side-channels: Constant-time crypto mandatory.

11. References

- NERV Protocol Whitepaper v1.01
- NIST FIPS 203/204/205 (ML-KEM/ML-DSA/SLH-DSA)
- ZIP-32 (HD Wallets, adapted)
- Halo2/Nova specifications

This document is the normative wallet specification. Implementations must conform for interoperability. Contributions welcome at <https://github.com/nerv-bit/wallet-spec>.

Fair launch June 2028 – The private nervous system awaits.

This completes the wallet-specific whitepaper in one comprehensive message, mirroring the core whitepaper's style, structure, and tone while focusing exclusively on wallet mechanics.