

Deep Research: Fork ZeroTier 1.14.1 dengan Enkripsi Quantum-Resistant

Disusun oleh Luci (OpenClaw)

29 September 2026

Ringkasan Eksekutif

ZeroTier 1.14.1 memakai kriptografi klasik: **Curve25519/Ed25519** (key agreement + signature), **Salsa20** (stream cipher), **Poly1305** (MAC). Semua ini **rusak oleh komputer kuantum** (Shor untuk ECC; Grover untuk symmetric — Salsa20 256-bit masih aman dari Grover, tapi kunci yang bocor dari ECDH membuatnya ikut rusak).

Catatan penting: **ZeroTier, Inc. sudah merilis “ZeroTier Quantum”** (Maret 2026) — produk komersial terpisah berbasis **Rust**, dengan skema **ML-KEM-1024 (FIPS 203) + ML-DSA-87 (FIPS 204) + P-384 hybrid + AES-256-GCM**, sesuai **NSA CNSA 2.0**. Produk itu **bukan fork dari kode 1.14.1** dan tidak terbuka di repo yang sama. Fork 1.14.1 dengan PQC tetap pekerjaan valid — pakai prinsip desain yang sama (hybrid PQC) di atas kode C++ yang ada.

Rekomendasi inti: Hybrid PQC — jalankan ML-KEM (Kyber) **paralel** dengan X25519, dan ML-DSA (Dilithium) **paralel** dengan Ed25519, gabungkan output. Keamanan tetap terjaga jika salah satu keluarga algoritma ternyata lemah.

1 Arsitektur Kriptografi ZeroTier 1.14.1 (Baseline)

Sumber: docs.zerotier.com/protocol, repo github.com/zerotier/ZeroTierOne

Komponen	File	Fungsi
Identity (X25519/Ed25519 keypair)	<code>node/Identity.hpp/.cpp</code>	40-bit ZT address diturunkan dari public key; PoW anti-collision
ECDH key agreement	<code>node/C25519.*</code>	Ephemeral key exchange antar peer
Signature (Ed25519)	<code>impl Ed25519</code>	Signing credential, certificate of membership
Packet encryption	<code>node/Salsa20.*</code> (NaCl-style box)	Salsa20 encrypt-then-MAC Poly1305
MAC	<code>node/Poly1305.*</code>	Autentikasi pesan
Credential signing	<code>CertificateOfMembershipController</code>	Controller sign membership dengan secret key

Alur Koneksi Peer (yang dimodifikasi)

1. Node A kirim **HELLO** berisi identity public key (X25519 + Ed25519) — 40-bit address diturunkan dari hash public key.

2. Handshake **ECDH** menghasilkan shared secret → derive key Salsa20+Poly1305.
3. Semua payload VL1 di-encrypt end-to-end dengan kunci tersebut.
4. Credential (certificate of membership, dll) ditandatangani Ed25519.

Batasan Penting

- **Tidak ada forward secrecy** (by design — GitHub issue #204).
- Salsa20/Poly1305 komposisi = NaCl reference (`crypto_box`).
- Trusted paths bisa skip enkripsi (jangan disentuh).

2 Standar Post-Quantum (NIST + NSA CNSA 2.0)

3.1 FIPS Final (Agustus 2024)

Standar	Nama lama	Fungsi	Parameter
FIPS 203 — ML-KEM	CRYSTALS-Kyber	Key Encapsulation (ganti ECDH)	ML-KEM-512/768/1024
FIPS 204 — ML-DSA	CRYSTALS-Dilithium	Signature (ganti Ed25519)	ML-DSA-44/65/87
FIPS 205 — SLH-DSA	SPHINCS+	Signature hash-based (backup)	SLH-DSA-128s/192s/256s

3.2 CNSA 2.0 (NSA) — target ZeroTier Quantum

- **Key exchange:** ML-KEM-1024; **Signature:** ML-DSA-87
- **Symmetric:** AES-256-GCM, HMAC-SHA-512; **Classical hybrid:** P-384 (opsional, disarankan)
- Timeline: transisi wajib sampai **2033**; data “harvest now, decrypt later” **mulai sekarang**.

3.3 Ukuran Parameter (implikasi overhead protokol)

Algoritma	Public key (B)	CT/Signature (B)
ML-KEM-768	1184	1088
ML-KEM-1024	1568	1568
ML-DSA-65	1952	3293
ML-DSA-87	2592	4627
X25519 (baseline)	32	32
Ed25519 (baseline)	32	64

Implikasi: HELLO/identity packet harus diperbesar ~2–4 KB. Perlu cek MTU/fragmentasi packet VL1. Hybrid mode = gabung klasik+PQC (mis. KDF atas kedua shared secret).

3 Strategi Integrasi (fork 1.14.1, C++)

4.1 Library PQC

Pilihan utama: liboqs — <https://github.com/open-quantum-safe/liboqs>

- Implementasi C, portable, sudah di-review luas; API `oqs_kem_encaps/decaps` (ML-KEM) dan `oqs_sig_sign/verify` (ML-DSA)
- License Apache-2.0 (kompatibel fork). Alternatif: reference NIST (C, exact-FIPS tapi lebih mentah).
- Build: git submodule / static lib di `make-linux.sh` / CMake.

4.2 Patch per Titik Integrasi

A. Handshake Key Agreement (wajib)

```
Lama:  shared = X25519(my_eph_priv, their_eph_pub)
Baru:  shared = KDF( X25519(...) || ML-KEM-768/1024_encaps(...) )
```

Tambah field di packet HELLO/KEY: `kem_public_key` (1184–1568 B) + `kem_ciphertext`. **Hybrid:** derive final key dari **kedua** shared secret via BLAKE2b/KDF — aman jika salah satu pecah.

```
// node/PQHybrid.hpp
#include <oqs/oqs.h>
struct HybridShared {
    uint8_t classical[32];    // X25519
    uint8_t pq[KEM_LEN];     // ML-KEM shared secret
};
void hybrid_kem(const uint8_t* peer_kem_pk,
                const uint8_t* my_eph_sk,
                const uint8_t* peer_eph_pk,
                HybridShared& out, uint8_t* kem_ct) {
    crypto_scalarmult(out.classical, my_eph_sk, peer_eph_pk);
    OQS_KEM* kem = OQS_KEM_new("ML-KEM-768");
    kem->encaps(out.pq, kem_ct, peer_kem_pk);
    OQS_KEM_free(kem);
}
// final key = KDF(classical || pq) -> key schedule Salsa20/ChaCha20
```

B. Identity & Signature (wajib)

- Identity saat ini: X25519 + Ed25519, address 40-bit dari hash public key.
- **Tambah ML-DSA-87 keypair** ke identity (field baru di serialisasi Identity).
- Tantangan: ukuran identity membengkak (2592 B pk + 4627 B sig per sign). Serialisasi `identity.public` wire dan file harus diperbesar — pakai **versioned format** (byte versi baru = identity v2, fallback v1 untuk backward compat).
- Signing credential controller: verify **kedua** signature (Ed25519 DAN ML-DSA) selama transisi (double-sign).

C. Packet Encryption (opsional, rendah risiko)

- Salsa20 256-bit: Grover butuh $\sim 2^{128}$ — **masih aman** selama key tidak bocor. Setelah handshake aman, enkripsi data tidak wajib diganti.
- Jika mau ikut CNSA 2.0 penuh: ganti ke **AES-256-GCM** (OpenSSL/liboqs tersedia), buang Poly1305. Lakukan terakhir.

4.3 Versioning & Kompatibilitas

- Capability bit di handshake: PQC_SUPPORTED.
- Peer lawas (v1.14.1 vanilla) → fallback mode klasik.
- Peer baru → mode hybrid / pure-PQC, dikonfigurasi via `local.conf`:

```
{ "settings": { "pqcMode": "hybrid",
  // "off" | "hybrid" | "pqconly"
  "kem": "ML-KEM-768",
  "sig": "ML-DSA-65" } }
```

Preset: `hybrid` disarankan; `pqconly` = CNSA 2.0 strict (ML-KEM-1024 / ML-DSA-87).

4 Roadmap Implementasi

Fase	Item	Estimasi
1	Fork repo, tag <code>v1.14.1-pqc</code> , integrate liboqs, build pipeline	1–2 hari
2	Patch handshake: hybrid X25519+ML-KEM, wire format v2	3–5 hari
3	Patch identity/signature: ML-DSA-87 double-sign, versioned serialize	3–5 hari
4	Credential (COM) controller double-sign + verify	2–3 hari
5	Config flag + negotiation + backward-compat test	2 hari
6	Test matrix: hybrid↔hybrid, hybrid↔vanilla, soak throughput	3–5 hari
7	(Opsional) AES-256-GCM ganti Salsa20 untuk CNSA penuh	3–5 hari

Total realistis: 3–4 minggu untuk 1 developer yang familiar dengan codebase.

Risiko

1. **Ukuran packet** — identity/HELLO +~4 KB; cek fragmentasi MTU 1280/1500 dan frame L2 VL2.
2. **CPU overhead** — ML-KEM keygen ~2–3 ms, ML-DSA sign ~5–10 ms (sekali per handshake, bukan per-packet). Dampak kecil; verifikasi dengan benchmark.
3. **Identity derivation** — address 40-bit dari public key; format baru harus tetap menjaga PoW.
4. **Audit krypto** — jangan deploy produksi tanpa review. liboqs = implementasi standard, tapi *pemakaian* (KDF hybrid, wire format) harus di-audit.
5. **Lisensi** — ZeroTier One: BSL 1.1 (Open Source Edition). Fork untuk pemakaian internal/non-komersial OK; cek klausa sebelum distribusi komersial.

5 Referensi

1. ZeroTier Protocol Docs — <https://docs.zerotier.com/protocol/>
2. ZeroTier Quantum (produk resmi) — <https://www.zerotier.com/quantum/>
3. Research Notes on 2.x Cryptography — <https://www.zerotier.com/news/research-notes-on-2-x-cry>
4. ZeroTierOne source — <https://github.com/zerotier/ZeroTierOne>
5. NIST FIPS 203/204/205 (ML-KEM, ML-DSA, SLH-DSA), Agustus 2024
6. NSA CNSA 2.0 FAQ — https://media.defense.gov/2022/Sep/07/2003071834/-1/-1/0/CSA_CNSA_2.0_ALGORITHMS_.PDF
7. liboqs (Open Quantum Safe) — <https://github.com/open-quantum-safe/liboqs>
8. ZeroTier blog: “Defusing the Q-Bomb” — ML-KEM key establishment, ML-DSA signature
9. ZeroTier blog: “NIST FIPS 203 Explained for Defense and Government Network Architects”
10. IETF hybrid KEM design — draft-ietf-tls-hybrid-design; NIST IR 8547 (migration guidance)