Appendix A – Technical Modules

Author: Ahmad Hemmati | Website: https://www.twincodesworld.com

Status: Open Source – Technical Supplement | Version: v1.0

This appendix specifies the technical modules of LHDNS in detail. It consolidates the architecture with canonical message formats, cryptographic bindings, validation logic, security mechanisms, performance targets, and governance-linked parameters. Defaults are locked here but may be updated by governance proposals.

Global Defaults (Canonical Parameters)

- **Signature**: Ed25519 (32B public key, 64B signature).
- **Key exchange**: X25519 (Curve25519).
- **AEAD encryption**: XChaCha20-Poly1305 (AES-GCM as HW-accelerated fallback).
- **Hash function**: SHA-256 (default; BLAKE3 optional).
- **Nonce length**: 128-bit (CSPRNG).
- **time_window**: 30s (`floor(ts / 30)`).
- **TTL**: 30s (interactive); 60–120s fallback.
- **Gossip fanout (k)**: 6.
- **Bloom filter false-positive rate**: ~1%.
- **Max enc_contact size**: 4096 bytes.
- **Rate-limit**: ≤60 entries/client/minute.
- **PoW initial target**: ~16 leading zero bits, adaptive.
- **Digest interval**: 60s.
- **Grace window**: +5s.
- **Peer ban threshold**: 3 misbehavior strikes.
- **Canonical serialization**: JSON Canonicalization Scheme (JCS) or CBOR canonical form.

Module 1 — Query Submission & Initial Processing

Goal: How clients construct lookups, how local nodes validate, and what an accepted ledger entry looks like.

Roles:

- Client (browser, IoT, app)
- Local node (first-hop validator, gossip)
- Gateway (optional DNS→LHDNS bootstrap)
- DLN network (ephemeral ledger layer)

Ledger entry (canonical JSON):

```
{
  "version": 1,
  "service_id": "svc:sha256:<hex>",
  "hash": "0x<sha256(...)>",
  "eph_pub": "<base64-ed25519-pub>",
  "enc_contact": "<base64-ciphertext>",
  "ts": 1690000000,
  "ttl": 30,
  "sig": "<base64(sig_by_eph_priv)>",
  "metadata": { "client_version": "0.1.0" }
}
```

Hash formula:

```
hash = SHA256(eph_pub || service_id || service_nonce || time_window)
```

enc_contact payload (plaintext before ECIES/X25519+AEAD):

```
{
  "client_eph_pub": "<base64>",
  "client_contact": { "type": "webrtc" | "websocket" | "relay_token",
"value": "<token>" },
  "ts": 1690000000,
  "nonce_client": "<random-128>",
  "client_sig": "<base64(sig_by_client_eph_priv_on(hash||ts))>"
}
```

Validation steps (local node):

1. Schema & version check.
2. Timestamp within ±60s skew.
3. TTL bounds (1–600s).
4. Verify signature with eph_pub.
5. Recompute hash.
6. Size ≤4096 bytes.
7. Rate-limit (≤60/min).
8. Adaptive PoW/micro-fee under load.

Reject codes: 400 schema, 401 sig, 402 fee/PoW, 429 rate-limit, 410 expired.

Module 2 — Ledger & Propagation

Goal: How entries propagate and expire.

Properties:

- Ephemeral (ts+ttl+grace).

- Consensus-light (gossip, no chain).
- Indexed by service_id.
- Audit digests for integrity.

Workflow:

1. Node validates entry.
2. Gossip to k=6 peers.
3. Peers deduplicate (Bloom filter, 1% FPR).
4. Entries indexed until expiry.
5. Nodes publish Merkle root digests every 60s.

Pruning: ts+ttl+grace → drop entry. Bloom filter prevents replay.

Anti-Sybil: node IDs = Ed25519 keys, admission requires PoW/stake, misbehavior → reputation penalty/ban.

Module 3 — Service Delivery / Transport

Goal: Secure session establishment.

Workflow:

- Client resolves service_id → entry.
- Handshake: X25519 ECDH → HKDF → session key.
- Transport: QUIC, WebRTC, WebSocket.
- Relays blind-forward AEAD ciphertext only.
- Multipath: parallel relay forwarding supported.

Session properties: ephemeral, encrypted, anonymous, unlinkable.

Use cases: chat, media streaming, IoT control, RPC, file transfer.

Module 4 — Privacy & Anonymity Enhancements

Threat model: traffic analysis, GPA, malicious relays.

Mechanisms:

- Query padding (fixed 256–512B).
- Batch forwarding + dummy queries.
- Onion routing (3-hop default).
- Multipath transport.
- Cover traffic (≥5%).
- Timing jitter.

- Ephemeral keys with forward secrecy.

Relay incentives: micropayments, staking, reputation.

Module 5 — Trust, Governance & Sybil Resistance

Mechanisms:

- Lightweight PoW per submission.
- Node staking & slashing.
- Rate limits + adaptive PoW.
- Node reputation scores (uptime, correct gossip).
- DAO governance for parameters (TTL, fanout, difficulty).
- Emergency quorum for fast parameter changes.

Module 6 — Integration & Interoperability

Client integration:

- Browser extensions / native APIs (lhdns:// URIs).
- Stub resolver (127.0.0.1:53, DoH endpoint).
- Mobile VPN-style interceptors.

Service integration:

- Gateways: DNS ↔ LHDNS translation.
- Hybrid dual-stack (DNS + LHDNS).
- Protocol adapters for HTTP(S), WebRTC, etc.

Interop: Tor/I2P bridges, DID/PKI bindings, mixnet relays.

Module 7 — Security & Threat Model

Goals: confidentiality, integrity, availability, unlinkability, accountability.

Threats & mitigations:

- Sybil/spam → PoW, staking, reputation.
- Traffic analysis → cover traffic, onion, multipath.
- Malicious relays → redundancy + slashing.
- Descriptor poisoning → signatures, reputation validation.
- DoS → adaptive PoW, rate limits.
- Eclipse/routing → multi-source gossip.
- Downgrade → explicit opt-in fallback only.

Residual risk: GPA correlation (mitigated, not eliminated).

Module 8 — Performance & Scalability

Targets:

- P50 lookup <1s, P90 <2.5s, P99 <5s.
- Throughput: thousands qps/relay cluster.
- Gossip convergence: O(log N).

Optimizations: batching, compression, caching, fast sync.

Large scale: 10k nodes <20s convergence; 100k <40s; 1M requires sharding/clustered gossip.

Module 9 — Incentive & Economic Layer

Native token (LHD): fees, staking, governance.

Mechanisms:

- Rewards: relay forwarding, gossip proof-of-relay.
- Penalties: slashing, exclusion, reputation loss.
- Fees: micro-fees/query, dynamic congestion pricing, off-chain channels for scaling.
- Treasury: DAO-managed funding for audits, grants.

Principles: fairness, sustainability, Sybil resistance, decentralization.

Module 10 — Deployment & Roadmap

Phases:

- Phase 0: prototype (<100 nodes).
- Phase 1: alpha testnet (100–1k).
- Phase 2: beta (1k–10k, DAO test).
- Phase 3: mainnet (>10k, incentives active).
- Phase 4: expansion (100k+, DID/mixnet).
- Phase 5: maturity (browser/OS native).

Testing: unit + integration, adversarial simulations, bug bounties.

Governance evolution: centralized bootstrap → DAO by Phase 3+.