# LIQUID KMS

## Unified Key Management and MPC Custody for Institutional Digital Securities

---

Zach Kelling

Satschel, Inc.

Version 1.0 — April 2026

CGGMP21 + FROST · 2-of-3 Threshold · HSM Integration
Passkey/WebAuthn Binding · Envelope Encryption · KMS-Backed Secrets

# Contents

## Abstract

We present Liquid Threshold, a unified key management and MPC custody architecture for the Liquidity Network. The system provides three distinct security layers: (1) **MPC threshold signing** using CGGMP21 [4] for ECDSA and FROST [5] for EdDSA, implementing 2-of-3 custody where no single party can move assets; (2) **secrets management** via an Infisical-fork KMS [3] with per-environment rotation, envelope encryption, and Kubernetes operator integration; (3) **validator key management** for Quasar consensus staking keys with HSM backing and automated rotation. The architecture ensures that Liquidity.io operates as a non-custodial platform despite its FINRA-registered ATS/BD status — the platform can facilitate trades but cannot unilaterally access customer assets.


## Introduction

Regulated securities platforms face a fundamental tension: regulators require operational control (AML, sanctions enforcement, trade surveillance), while customers demand non-custodial guarantees (the platform cannot steal funds). Liquid Threshold resolves this tension through threshold cryptography — the platform participates in signing but cannot sign alone.

The system builds on three upstream Lux Network research projects:

- **Liquid Threshold MPC** [1]: Decentralized threshold custody with 2-of-3 CGGMP21
- **Lux KMS**: Validator staking key management with MPC backing
- **Hanzo KMS** [3]: Secrets management with envelope encryption and rotation

## MPC Threshold Signing

### Protocol Selection

| Protocol | Curve | Use Case | Security |
|---|---|---|---|
| CGGMP21 | secp256k1 (ECDSA) | EVM transactions, bridge signing | Classical |
| FROST | ed25519 (Schnorr) | Consensus signatures, Warp messages | Classical |
| Ringtail | Module-LWE (lattice) | Post-quantum consensus backup | Post-quantum |

CGGMP21 provides identifiable abort — if a signer misbehaves, the protocol identifies the cheater without restarting from scratch [4].

### 2-of-3 Custody Model

```
Shard 1: User Device
  - Stored in IndexedDB (web) or Secure Enclave (mobile)
  - Biometric-gated via Passkey/WebAuthn (FIDO2)
  - Never leaves the device

Shard 2: ATS Auto-Signer
  - Rate-limited (max 100 signatures/minute)
  - Policy-enforced (compliance checks before co-signing)
  - Runs in TEE (Intel SGX or AMD SEV-SNP)
```

```
Shard 3: HSM Escrow
  - Held by independent third party (law firm + auditor)
  - PKCS#11 interface to hardware security module
  - Used only for recovery (user + escrow, bypassing ATS)
```

### Non-Custodial Guarantee

The critical property: **the ATS (Shard 2) cannot move funds alone**. It needs either the user (Shard 1) or the escrow (Shard 3) to co-sign. This means:

- Normal operation: User (Shard 1) + ATS (Shard 2) sign trades
- Recovery: User (Shard 1) + Escrow (Shard 3) recover without ATS
- Emergency: Escrow (Shard 3) + User (Shard 1) bypass a compromised ATS
- Impossible: ATS (Shard 2) + Escrow (Shard 3) cannot act without user consent

## Passkey/WebAuthn Binding

User shards are bound to FIDO2 Passkeys rather than passwords or biometric fuzzy extractors:

| Method | Security | User Experience |
|---|---|---|
| Password | Phishable, reusable | Poor (forgotten, reused) |
| Biometric fuzzy extractor | Novel, unproven | Medium (calibration needed) |
| Passkey/WebAuthn (FIDO2) | Phishing-resistant, hardware-backed | Excellent (fingerprint/face) |

The Passkey authenticates the user to their device's Secure Enclave, which then releases the MPC shard for signing. The shard itself never leaves the enclave.

## Secrets Management (KMS)

The platform KMS (Infisical fork) manages all non-MPC secrets:

| Secret Type | Management |
|---|---|
| Database credentials | Auto-rotated every 24h |
| API keys (external providers) | Per-environment, versioned |
| JWT signing keys | Rotated weekly, JWKS published |
| Webhook HMAC keys | Per-tenant, per-service |
| Validator staking keys | HSM-backed, MPC-generated |
| Encryption keys (data at rest) | Envelope encryption, per-user DEK |

### Envelope Encryption

Data at rest uses two-layer envelope encryption:

1. **Master Key (MEK)**: Stored in HSM, never exported
2. **Data Encryption Key (DEK)**: Per-user, encrypted by MEK
3. **Data**: Encrypted by DEK (AES-256-GCM)

To read user data: KMS decrypts DEK using MEK, then DEK decrypts the data. The MEK never leaves the HSM.

## Validator Staking Keys

Quasar consensus validators require BLS signing keys for block production and Ringtail keys for post-quantum certificates. These keys are:

- Generated via distributed key generation (DKG) across 3 datacenters
- Stored in HSMs at each datacenter (SFO, NYC, MCI)
- Rotated via proactive secret sharing (no downtime)
- Backed up encrypted to KMS with geographic redundancy

## Kubernetes Integration

The KMS Operator syncs secrets from KMS to Kubernetes Secrets:

```
apiVersion: secrets.liquid.network/v1alpha1
kind: KmsSecret
metadata:
  name: ats-credentials
  namespace: liquidity
spec:
  kmsEndpoint: https://\texttt{<internal>}/api
  projectSlug: liquid-infra
  environment: prod
  secretPath: /ats
  resyncInterval: 60
  target:
    name: ats-credentials
    namespace: liquidity
```

## Security Audit Results

| Component | Finding | Severity | Status |
|---|---|---|---|
| MPC key generation | DKG completes in 3 rounds | Info | N/A |
| Shard storage | IndexedDB uses origin-isolated storage | Low | Accepted |
| ATS rate limiter | 100 sig/min enforced by TEE | Info | N/A |
| HSM integration | PKCS#11 tested with Thales Luna | Info | Verified |
| Passkey binding | WebAuthn attestation verified | Info | N/A |
| Envelope encryption | AES-256-GCM with 96-bit nonce | Info | N/A |
| Key rotation | Zero-downtime proactive resharing | Info | Verified |

## Regulatory Compliance

| **Regulation** | **KMS Compliance** |
|---|---|
| SEC Rule 15c3-3 | Non-custodial (2-of-3, ATS cannot move funds alone) |
| FINRA 4370 | Key backup in 3 geographic locations |
| SEC 17a-4 | All key operations logged immutably on-chain |
| GDPR Art. 32 | Encryption at rest (envelope), in transit (mTLS) |
| SOC 2 Type II | Audit trail for all secret access |

## Conclusion

Liquid Threshold provides a unified key management architecture that satisfies both the non-custodial guarantees demanded by institutional clients and the operational control requirements of SEC/FINRA regulation. By combining CGGMP21 threshold ECDSA, FROST threshold EdDSA, Passkey/WebAuthn user authentication, envelope encryption, and HSM-backed validator keys, the system ensures that no single party — including the platform operator — can unilaterally access customer assets.

## References

[1] Z. Kelling, "Liquid Threshold: Decentralized MPC Custody with Quantum-Safe Threshold Signatures," Lux Industries, 2023.

[2] Z. Kelling, "Universal Threshold Signatures: Multi-Chain Cryptographic Infrastructure," Lux Industries, 2021.

[3] Z. Kelling, "Hanzo KMS: Secrets Management with Per-Environment Rotation," Hanzo AI, 2021.

[4] R. Canetti et al., "UC Non-Interactive, Proactive, Threshold ECDSA with Identifiable Aborts," ACM CCS, 2020.

[5] C. Komlo, I. Goldberg, "FROST: Flexible Round-Optimized Schnorr Threshold Signatures," SAC, 2020.

[6] Z. Kelling, "Ringtail: Lattice-Based Post-Quantum Threshold Signatures," Lux Industries, 2024.

[7] Z. Kelling, "Post-Quantum Cryptographic Suite for EVM," Lux Industries, 2024.

[8] FIDO Alliance, "Web Authentication: An API for accessing Public Key Credentials," W3C, 2021.