# Quantum-Resistant Counter-Exploitation: Confidential Compute and Coercion-Resistant Communications

Zach Kelling — Hanzo Team

Version 1.0 — February 7, 2026 February 2026

**Abstract**

We present a suite of counter-exploitation technologies providing quantum-resistant secure messaging, coercion-resistant governance, confidential computation on encrypted data, and post-quantum anonymous communication. The SessionVM delivers end-to-end encrypted messaging using ML-KEM-768 (FIPS 203) and ML-DSA-65 (FIPS 204) with fresh key encapsulation per message, achieving 526,000 messages per second with forward secrecy. ThresholdVM provides fully homomorphic encryption (FHE) with GPU-accelerated BFV/BGV/CKKS/TFHE schemes and threshold decryption ensuring no single party observes plaintext. LatticeLSAG ring signatures enable post-quantum anonymous group signing with linkability for double-action detection. Zero-knowledge proof systems provide privacy pools, range proofs, and confidential transactions. A multi-agent adversarial framework with 83 specialized agents enables automated red/blue team operations. Together, these capabilities prevent adversary exploitation of cryptographic, network, and AI systems while maintaining operational security in contested environments.

**Executive Summary.** This is a toolkit for denying an adversary any use of what they intercept, capture, or coerce. If they record your messages, the encryption is quantum-resistant, so harvested traffic stays unreadable even against a future quantum computer; if they seize a key, fresh keys per message mean past communications remain sealed. If they pressure a person, the voting and governance schemes are built so no one can prove how they voted, removing the leverage behind coercion. Sensitive data can be computed on while still encrypted—no operator and no single machine ever sees it in the clear—and operators can authenticate and file reports anonymously, yet still be held accountable for duplicate or contradictory actions. For defense and intelligence users operating in contested or denied environments, the practical result is that compromise of a device, a key, or a person does not cascade into compromise of the mission.

## Contents

# 1 Introduction

Counter-exploitation requires not merely defending systems against attack but actively preventing adversary use of intercepted communications, compromised keys, and captured infrastructure. The convergence of quantum computing, advanced persistent threats, and AI-powered adversary tools demands countermeasures that operate at multiple layers simultaneously.

This paper addresses five counter-exploitation domains:

1. **Communication security**: Quantum-resistant messaging immune to harvest-now-decrypt-later.
2. **Governance resilience**: Coercion-resistant decision-making under adversary pressure.
3. **Computation privacy**: Processing encrypted data without exposing plaintext.
4. **Identity protection**: Anonymous authentication and reporting.
5. **Active defense**: AI-powered adversarial testing and threat detection.

# 2 Post-Quantum Secure Messaging: SessionVM

## 2.1 Architecture

SessionVM is a post-quantum secure messaging virtual machine deployed on the Pars Network S-Chain. It provides end-to-end encrypted communication using exclusively NIST-standardized algorithms.

| **Algorithm** | **Standard** | **Purpose** |
|---|---|---|
| ML-KEM-768 | FIPS 203 | Key encapsulation (confidentiality) |
| ML-DSA-65 | FIPS 204 | Digital signatures (authentication) |
| XChaCha20-Poly1305 | IETF draft-xchacha | Symmetric AEAD encryption |
| BLAKE2b-256 | RFC 7693 | Hashing and key derivation |

Table 1: SessionVM cryptographic primitives.

## 2.2 Session Identity

Post-quantum session identities use the prefix `07`:

$$\text{SessionID} = \text{“07”} \| \text{hex}\big(\text{BLAKE2b-256}(\text{pk}_{\text{KEM}} \| \text{pk}_{\text{DSA}})\big)$$

Total length: 66 characters (backward-compatible with classical `05`-prefix identities).

## 2.3 1:1 Message Encryption Protocol

**Algorithm 1** SessionVM Post-Quantum Message Encryption

**Sender** $S$ encrypts message $m$ for recipient $R$:

**Step 1: Encapsulation**
$(\text{ct}_{\text{kem}}, \text{ss}) \leftarrow \text{ML-KEM-768.Encaps}(\text{pk}_R^{\text{KEM}})$

**Step 2: Key Derivation**
$K \leftarrow \text{BLAKE2b-256}(\text{ss} \| \text{pk}_S^{\text{DSA}} \| \text{pk}_R^{\text{KEM}})$

**Step 3: Symmetric Encryption**
$\text{nonce} \leftarrow \text{Random}(24)$
$\text{ct}_{\text{payload}} \leftarrow \text{XChaCha20-Poly1305.Seal}(K, \text{nonce}, m)$

**Step 4: Authentication**
$\sigma \leftarrow \text{ML-DSA-65.Sign}(\text{sk}_S^{\text{DSA}}, \text{ct}_{\text{kem}} \| \text{nonce} \| \text{ct}_{\text{payload}})$

**Wire format**:
$\text{ct}_{\text{kem}}^{1088} \| \text{nonce}^{24} \| \text{ct}_{\text{payload}} \| \sigma^{3309} \| \text{pk}_S^{\text{DSA},1952}$
**Overhead**: 6,373 bytes (vs. 112 bytes classical)

## 2.4 Security Properties

**Theorem 2.1** (Per-Message Forward Secrecy)**.** *Each message uses a fresh ML-KEM encapsulation, generating a unique shared secret. Compromise of the recipient's long-term KEM key does not reveal past shared secrets, as each encapsulation is probabilistic and the shared secret is bound to the specific ciphertext.*

**Theorem 2.2** (Authentication)**.** *Every message is signed with ML-DSA-65, providing EU-CMA authentication. The sender's DSA public key is included in the wire format for verification without prior key exchange.*

## 2.5 Performance

| Operation | Time (M1 Max) |
|---|---|
| GenerateIdentity | 268 $\mu$s |
| Encaps + Decaps | 226 $\mu$s |
| Sign + Verify | 1.08 ms |
| CreateSession | 3.8 $\mu$s |
| SendMessage | 1.9 $\mu$s |

Table 2: SessionVM operation latencies.

## 2.6 Swarm Distribution

Messages are distributed across a swarm of storage nodes:

- Deterministic node assignment from session/message ID.
- ∼10 GB storage per node for swarm participation.
- Automatic rebalancing on node churn.
- 24-hour session TTL, 10,000 message limit, 30-day retention.

# 3 Coercion-Resistant Governance

## 3.1 Encrypted Voting (PIP-0003, PIP-0012)

The Pars Network implements coercion-resistant voting where:

- Votes are encrypted under FHE public keys before submission.
- Tallying occurs homomorphically—no individual vote is ever decrypted.
- Threshold decryption reveals only the aggregate result.
- Voters cannot prove how they voted (receipt-freeness), defeating coercion.

## 3.2 Encrypted CRDTs (PIP-0013)

Conflict-Free Replicated Data Types (CRDTs) operate on encrypted state:

- Merge operations execute on ciphertext without decryption.
- Eventual consistency maintained across disconnected nodes.
- Adversary with network access sees only encrypted CRDT state.

## 3.3 Shadow Governance (PIP-7010)

For censorship-resistant decision-making under oppressive conditions:

- Governance proposals encrypted and distributed via RNS mesh.
- Voting via anonymous ring signatures (LatticeLSAG).
- Results revealed only through threshold decryption ceremony.
- No single authority can block or observe governance activity.

## 3.4 Fractal Governance (PIP-7007)

Hierarchical governance with compartmentalization:

- Cell-based structure where each level knows only adjacent levels.
- Compromise of one cell does not expose the governance tree.
- Zero-knowledge proofs verify authority without revealing identity.

# 4 Confidential Computing: ThresholdVM and FHE

## 4.1 ThresholdVM Architecture

ThresholdVM provides fully homomorphic encryption as a native virtual machine:

| Scheme | Domain | Use Case |
|---|---|---|
| BFV | Integer | Exact integer arithmetic on encrypted data |
| BGV | Modular | Modular arithmetic for cryptographic operations |
| CKKS | Approximate real | ML inference on encrypted features |
| TFHE | Boolean | Arbitrary Boolean circuits on encrypted bits |

Table 3: FHE schemes supported by ThresholdVM.

## 4.2 Threshold Decryption

No single party ever sees plaintext:

1. Smart contract encrypts data under FHE public key.
2. ThresholdVM evaluates computation homomorphically.
3. $t$-of-$n$ MPC parties produce decryption shares.
4. Result assembled from shares; individual shares reveal nothing.

## 4.3 GPU Acceleration

The Hanzo GPU kernel library provides Metal and CUDA kernels for FHE operations:

- NTT/iNTT forward and inverse transforms.
- Polynomial multiplication with modular arithmetic.
- TFHE bootstrap and blind rotate operations.
- Batch parallel ciphertext operations.

## 4.4 EVM Precompile Integration

| Address | Name | Gas |
|---|---|---|
| `0x0700` | FHE Operations (Fheos) | 500,000 |
| `0x0701` | Access Control (ACL) | 25,000 |
| `0x0702` | Input Verifier | 50,000 |
| `0x0703` | Gateway (Warp relay) | 100,000 |

Table 4: FHE EVM precompile addresses.

### 4.5 TFHE-KMS Bridge

Encrypted policy evaluation without plaintext exposure:

1. Client encrypts query under TFHE public key.
2. KMS evaluates access policy on encrypted query.
3. Returns encrypted Boolean result.
4. Client decrypts to learn only the policy decision.

## 5 Lattice Ring Signatures for Anonymity

### 5.1 LatticeLSAG Construction

LatticeLSAG extends Linkable Spontaneous Anonymous Group signatures to ML-DSA-65:

**Definition 5.1** (LatticeLSAG)**.** Given a ring $R = \{pk_1, \ldots, pk_n\}$ of ML-DSA-65 public keys and a signer with index $\pi$ and secret key $sk_\pi$:

1. **Anonymity**: For any verifier, $\Pr[\text{identify } \pi] = 1/n$.
2. **Linkability**: Signer produces key image $I = H(sk_\pi)$; same signer always produces same $I$.
3. **Spontaneous**: Ring formed from any public keys without coordination.
4. **PQ-safe**: Security reduces to Module-LWE hardness.

### 5.2 Applications

- **Anonymous intelligence reporting**: Source signs report with ring of authorized personnel; recipient verifies authenticity without identifying source.
- **Whistleblower protection**: Key images detect duplicate reports while preserving anonymity.
- **Covert operations**: Operational orders authenticated without revealing which commander issued them.
- **Double-action detection**: Linkability prevents same agent from submitting contradictory reports.

## 6 Zero-Knowledge Proof Systems

### 6.1 Privacy Pools

Confidential UTXO model with nullifiers:

- Transactions hide sender, recipient, and amount.
- Nullifiers prevent double-spending without revealing which UTXO was consumed.
- Range proofs verify amounts are positive without revealing values.
- ZK proofs of solvency without balance disclosure.

### 6.2 Proof Systems

| System | Gas | Setup | Proof Size |
|---|---|---|---|
| Groth16 | 200K | Trusted | 128 B |
| PLONK | 250K | Universal | 384 B |
| fflonk | 250K | Universal | 320 B |
| Halo2 | 300K | None | 1–10 KB |
| KZG4844 | 50K | Trusted | 48 B |

Table 5: Zero-knowledge proof systems.

## 7 Content Provenance and Data Integrity

### 7.1 Data Integrity Seal (PIP-0010)

Cryptographic seals bind content to its creator:

- ML-DSA-65 signature over content hash + metadata + timestamp.
- Anchored to blockchain for immutable timestamping.
- Verifiable chain of custody from creation through distribution.

### 7.2 Content Provenance (PIP-0011)

Track content origin and modification history:

- Each transformation produces a provenance record (signed).
- Merkle tree of provenance records for efficient verification.
- Detect AI-generated content manipulation via provenance gaps.

### 7.3 Attestation Protocol

Domain-typed attestations with equivocation detection:

- **OracleCommit**: Data feed commitments.
- **SessionComplete**: Session lifecycle events.
- **EpochBeacon**: Epoch transition markers.
- Equivocation (signing conflicting statements) is automatically detected and penalized.

## 8 Network-Layer Countermeasures

### 8.1 Zero-Trust Micro-Segmentation

Hanzo ZT provides identity-first networking:

- Every connection requires cryptographic identity verification.

- No implicit trust from network position or IP address.
- Policy enforcement at every hop in the overlay mesh.
- Lateral movement from compromised nodes is impossible without valid identity.

### 8.2 Resilience Against Targeting

- **Circuit breakers**: Prevent cascading failure exploitation (configurable thresholds, automatic recovery).
- **Rate limiting**: Token bucket per-endpoint prevents resource exhaustion.
- **RNS mesh**: No centralized infrastructure to target; mesh reforms around destroyed nodes.
- **Dilithium transport**: Adaptive flow control maintains throughput under degraded conditions.

## 9 AI-Powered Counter-Intelligence

### 9.1 Multi-Agent Adversarial Framework

- **83 specialized agents**: Security-focused variants for vulnerability assessment, code audit, and threat modeling.
- **Red/blue pairs**: Automated adversarial testing where red agents attack and blue agents defend.
- **15 orchestrators**: Incident response workflows coordinating multiple agents.
- **Extended reasoning**: 200–300 sequential tool calls for deep analysis.
- **Anomaly detection**: O11y platform with real-time behavioral analytics for intrusion indicators.

### 9.2 Agent Consensus for Threat Validation

- Multiple agents independently analyze a potential threat.
- Threshold voting (67%) required to confirm threat classification.
- W3C DID authentication prevents agent impersonation.
- PQ-secure communication via ZAP protocol.

## 10 Counter Harvest-Now-Decrypt-Later

### 10.1 Defense-in-Depth Against HNDL

1. **Hybrid PQ transport**: All network traffic uses ML-KEM-768 + X25519; adversary must break both to decrypt.
2. **Epoch-based key rotation**: Consensus keys rotate every 10 minutes; even a quantum break reveals only 10 minutes of data.

3. **Forward secrecy**: Ephemeral keys destroyed after every session; past sessions cannot be decrypted.
4. **Quantum-safe symmetric**: XChaCha20-Poly1305 (256-bit) and BLAKE2b-256 provide 128-bit post-Grover security.
5. **Per-message encapsulation**: SessionVM generates fresh ML-KEM ciphertext per message, not per session.

### 10.2 Key Hierarchy Protection

| Layer | Key Type | Rotation | Protection |
|---|---|---|---|
| Root | Shamir shares | Manual ceremony | HSM, $t$-of-$n$ |
| Organization | AES-256 per-org | Policy-based | KMS, encrypted |
| Session | Ephemeral ML-KEM | Per-session | Destroyed after use |
| Message | Ephemeral ML-KEM | Per-message | Destroyed after use |

Table 6: Key hierarchy with rotation and protection mechanisms.

## 11 Social Recovery and Key Escrow

### 11.1 Shamir-Based Recovery

- Root keys split into $n$ shares; $t$ required for reconstruction.
- Shares distributed to geographically separated custodians.
- Break-glass emergency tokens with time-limited validity.
- WORM audit log prevents undetected recovery operations.

### 11.2 Regulator Escrow

For compliance with lawful intercept requirements:

- Escrow shards held by designated authorities.
- Cooperative reconstruction requires $t$ parties (no unilateral access).
- Audit trail records all access attempts.
- 6–7 year retention per SEC 17a-4 requirements.

## 12 Naval Countermeasure Scenarios

### 12.1 Electronic Warfare Environment

- RNS mesh reforms around jammed nodes; LoRa provides backup.
- PQ encryption prevents exploitation of intercepted traffic.
- Circuit breakers prevent adversary-induced cascading failures.

### 12.2 Contested Communications

- SessionVM messaging over RNS mesh: no centralized infrastructure to target.
- ZAP protocol minimizes bandwidth (91% reduction vs. JSON-RPC).
- Store-and-forward ensures message delivery during intermittent connectivity.

### 12.3 Insider Threat

- Zero-trust prevents lateral movement from compromised endpoints.
- Ring signatures enable anonymous reporting of suspicious activity.
- Threshold MPC ensures no single insider can compromise keys.
- Behavioral analytics detect anomalous access patterns.

### 12.4 Supply Chain Integrity

- Content provenance (PIP-0011) tracks software components from source to deployment.
- Data integrity seals (PIP-0010) verify component authenticity.
- Attestation protocol detects tampering via equivocation detection.

## 13 Strategic Context

Counter-exploitation is where ownership of the full stack matters most: every layer described here—the messaging, the governance, the confidential compute, and the AI-driven adversarial defense—is built and controlled by Hanzo AI, an American-owned applied-AI lab and venture studio (Techstars '17, founded 2014) behind more than 100 venture-funded startups and several unicorns, led by a founder expert in both AI/ML and cryptography. The AI-powered components of this suite—the multi-agent red/blue framework, agent-consensus threat validation—depend on frontier models, and that is the strategic fault line. U.S. frontier AI has consolidated into a closed commercial duopoly, while the only broadly open frontier weights available today are largely of foreign origin; an organization whose mission is to deny adversary exploitation cannot reasonably build its active defenses on models it can neither inspect nor run in a denied environment. Hanzo is the American open-source frontier alternative: it owns both the Zen model family and the post-quantum cryptography underneath, so the adversarial-defense agents and the coercion-resistant primitives share one auditable, sovereign foundation that runs offline and at the edge—no rented API, no foreign dependency, nothing the adversary controls. This is operational rather than aspirational: the same agentic engineering stack delivered a FINRA/SEC-approved, SOC2 securities platform—a full ATS/BD/TA stack consolidated from 200+ microservices into 3 compiled binaries—to production with a small team, demonstrating that the agent orchestration invoked throughout this paper performs under real regulatory and reliability constraints.

## 14 Conclusion

We have presented a comprehensive counter-exploitation suite combining post-quantum secure messaging, coercion-resistant governance, confidential computing, anonymous authentication, and AI-powered adversarial defense. Every component uses NIST-standardized post-quantum algorithms, ensuring long-term security against quantum adversaries.

The integration of these capabilities into a unified architecture—where SessionVM messaging, ThresholdVM computation, LatticeLSAG anonymity, and multi-agent threat detection share the same cryptographic foundation and consensus infrastructure—provides defense-in-depth that individual point solutions cannot achieve.

## References

[1] NIST, "ML-KEM Standard," FIPS 203, 2024.

[2] NIST, "ML-DSA Standard," FIPS 204, 2024.

[3] NIST, "SLH-DSA Standard," FIPS 205, 2024.

[4] NSA, "CNSA Suite 2.0," 2022.

[5] Y. Nir, A. Langley, "ChaCha20 and Poly1305 for IETF Protocols," RFC 8439, 2018.

[6] M-J. Saarinen, J-P. Aumasson, "The BLAKE2 Cryptographic Hash and MAC," RFC 7693, 2015.

[7] I. Chillotti *et al.*, "TFHE: Fast Fully Homomorphic Encryption," JoC, 2020.

[8] "Practical Two-Round Threshold Signature from LWE," IACR ePrint 2024/1113.

[9] J. Liu *et al.*, "Linkable Spontaneous Anonymous Group Signature," ACISP 2004.

[10] J. Groth, "On the Size of Pairing-Based Non-interactive Arguments," EUROCRYPT 2016.

[11] A. Gabizon *et al.*, "PLONK: Permutations over Lagrange-bases," IACR ePrint 2019/953.

[12] Pars Network, "PIP-0003: Coercion-Resistant Voting," 2026.

[13] Pars Network, "PIP-0013: Encrypted CRDTs," 2026.

[14] Pars Network, "PIP-7010: Shadow Governance," 2026.

[15] Pars Network, "PIP-0010: Data Integrity Seal," 2026.

[16] Pars Network, "PIP-0011: Content Provenance," 2026.

[17] A. Shamir, "How to Share a Secret," CACM, 1979.

[18] J. Kindervag, "Build Security Into Your Network's DNA: The Zero Trust Architecture," Forrester, 2010.

## Reproducibility

**Source code.** `github.com/hanzoai/kms, hanzoai/iam` (commit `TBD`).
**Build.** make build per component; integrated K8s deploy via kubectl kustomize.
**Hardware tested.** Linux amd64/arm64; HSM integration tested with PKCS#11.
**Standards referenced.** NIST SP 800-207 (Zero Trust), Shamir SSS, FIPS 140-3.
**Contact.** research@hanzo.ai.