TLO: Topology-Lattice Obfuscation for Smart Contracts

Anonymous Submission

Abstract

We present TLO (Topology-Lattice Obfuscation), a practical circuit obfuscation framework for smart contracts. Security derives from a two-layer defense: a topology layer using structural mixing defeats structural and statistical attacks (empirically validated), while the LWE layer computes inner products on-chain to hide control functions. Security is based on uniform-secret LWE hardness (∼108-bit classical with n=64, σ=1024; see §5.3) combined with topology properties, providing post-quantum resistance (assuming LWE quantum-resistance).

TLO achieves 6/6 resistance against our attack evaluation matrix at ∼2.58M gas (8.6% of block limit). Control functions are hidden via LWE ciphertexts where the key $s_{\text{enc}} = H(\mathsf{secret})$ is derived from the secret at encryption time; at evaluation, the contract derives $s(x) = H(x)$ from the candidate input—matching only when $x = \mathsf{secret}$. Attackers *can* simulate evaluation with arbitrary keys, but incorrect keys yield garbage outputs. The 1-bit oracle interface limits information leakage. Target applications include predicates with eventually-expiring secrets (honeypots, sealed-bid auctions, lotteries). Deployment requires only a standard smart contract with expiry timestamp.

1 Introduction

Smart contracts are fully transparent. Anyone can read bytecode, analyze logic, and exploit vulnerabilities. This conflicts with applications requiring hidden logic: cryptographic honeypots, MEV-resistant execution, sealed-bid auctions, and private liquidation thresholds.

Indistinguishability obfuscation (iO) provides permanent security but requires impractical overhead ($10^6\times$). We take a different approach: *practical obfuscation* that resists known attack classes through two complementary layers:

Two-Layer Security Model:

1. **Layer 1 (Topology):** Structural mixing defeats structural/statistical attacks. *Security: heuristic.*
2. **Layer 2 (LWE):** On-chain inner products defeat semantic attacks. *Security: computational (∼108-bit with σ=1024).*

Key mechanism: 1-bit oracle + wrong-key-gives-garbage property.

1.1 Contributions

1. **TLO Framework:** Two-layer obfuscation combining topology mixing with on-chain LWE inner products (∼2.58M gas for n=64).
2. **Structural Mixing:** Wire selection defeating structural/statistical attacks (heuristic, empirically validated).
3. **Oracle + Wrong-Key-Garbage:** Attackers *can* simulate with arbitrary keys, but incorrect keys yield garbage; 1-bit oracle limits information leakage.
4. **On-Chain LWE:** Control functions hidden via LWE ciphertexts with full inner product computation.
5. **Post-Quantum Security:** ∼108-bit classical via uniform-secret LWE (n=64, σ=1024); ∼203-bit for n=128; see §5.3.

1.2 Scope

We claim: Security based on LWE hardness + topology heuristics; 6/6 attack resistance in our matrix; post-quantum resistance.

We do NOT claim: iO security; universal security.

2 Preliminaries

2.1 Learning With Errors

Definition 2.1 (LWE [5]). For dimension n, modulus q, and error distribution χ, the LWE problem is: given $(A, As+e \mod q)$ where $A \in \mathbb{Z}_q^{m\times n}$, $s \in \mathbb{Z}_q^n$, $e \leftarrow \chi^m$, distinguish from uniform (A, u).

LWE is believed quantum-resistant and forms the basis for post-quantum cryptography standards (ML-KEM) [1]. Our parameters (n=64, q=65521) are smaller than NIST profiles; see §5.3 for security estimates.

2.2 LWE Control Function Hiding

We hide each gate's control function via LWE ciphertexts. Each CF bit is encoded as (a, b) where $b = \langle a, s_{\text{enc}} \rangle + e + \mathsf{bit} \cdot q/2$. At encryption time, $s_{\text{enc}} = H(\mathsf{secret})$. At evaluation time, the contract derives $s(x) = H(x)$ from the candidate input—decryption succeeds iff $x = \mathsf{secret}$.

2.3 Reversible Circuits

Definition 2.2 (Reversible Gate)**.** A gate $g = (a, c_1, c_2, c_f)$ operates on n wires: active wire a is XORed with $c_f(c_1, c_2)$ where $c_f : \{0,1\}^2 \to \{0,1\}$ is one of 16 control functions.

Gates are self-inverse: $g(g(s)) = s$. This enables commit-reveal protocols where the solver demonstrates knowledge without revealing the secret.

3 The Topology Layer

The topology layer is a reversible circuit mixing design that defeats structural and statistical attacks through wire selection, without cryptographic primitives.

3.1 Wire Selection Algorithm

Structural mixing selects wires to defeat pattern detection:

1. **Non-pow2 distances:** Control wires at distances $d \notin \{1, 2, 4, 8, \ldots\}$ from active wire defeats butterfly/FFT pattern detection
2. **Uniform wire usage:** Prefer underused wires; defeats chi-squared statistical attacks
3. **Irregular layers:** Varying gates per layer (e.g., 30–70 for 256 wires) defeats regularity detection
4. **64+ wires:** Sufficient width defeats diagonal correlation (Pearson $r < 0.10$)

Listing 1: Wire Selection (pseudocode)

```
def select_control_wire(active, usage, target):
    # Choose non-pow2 distance
    d = random_choice(non_pow2_distances)
    candidate = (active + d) % num_wires

    # Prefer underused wires (70% prob)
    if random() < 0.7 and usage[candidate] < target:
        return candidate

    # Otherwise, find underused alternative
    return find_underused_wire(usage, target)
```

3.2 Topology Attack Resistance

Attack	Type	Defense	Mechanism
Compression	Structural	Topology	No duplicate gates
PatternMatch	Structural	Topology	Random CF cycling
Structural	Structural	Topology	Non-pow2 distances
Statistical	Statistical	Topology	Uniform wire usage
DiagCorrelation	Statistical	Topology	64+ wires

Table 1: Topology defeats structural/statistical attacks.

Key insight: Unlike butterfly or derangement topologies that only rearrange gates, structural mixing has anti-attack properties *built into wire selection.*

4 LWE for Semantic Attacks

4.1 The RainbowTable Problem

RainbowTable is a *semantic* attack—it matches truth-table behavior, not structure:

1. Extract subcircuit from obfuscated circuit
2. Evaluate subcircuit on sample inputs
3. Match behavior against pre-computed lookup table

Topology cannot defeat this. Any structural transformation preserves semantic behavior of reversible circuits.

4.2 How LWE Blocks RainbowTable

Attackers *can* simulate evaluation with arbitrary keys—this is not a restricted oracle. The defense is the **wrong-key-gives-garbage** property: incorrect keys yield random CF bits, producing garbage outputs.

> **On-Chain Inner Product:** Control functions encoded as LWE ciphertexts (a, b). At encryption: $s_{\text{enc}} = H(\mathsf{secret})$. At evaluation: $s(x) = H(x)$.
> **Key mechanism:** Attackers can simulate with any s', but $s' \neq s_{\text{enc}}$ yields garbage CF bits. Combined with 1-bit output, this limits information leakage.

Proposition 4.1 (LWE Security)**.** *Under* LWE *hardness, no* PPT *adversary can evaluate a subcircuit in isolation.*

Proof sketch: Each CF is hidden via LWE ciphertext. Subcircuit evaluation requires recovering CF bits, which reduces to LWE hardness. □

5 Security Analysis

5.1 Two-Layer Security Model

TLO provides security through complementary layers with different bases:

1. **Topology layer (heuristic):** Defeats structural/statistical attacks through wire selection. *Empirically validated, not proven.*

2. LWE **layer (computational):** Defeats semantic attacks via on-chain inner products. *Based on* LWE *hardness (∼49-bit with n=64; see §5.3).*

3. **Wrong-key-gives-garbage:** Attackers can simulate with arbitrary keys, but incorrect keys yield garbage outputs. Combined with 1-bit oracle.

Definition 5.1 (Extraction Resistance)**.** An obfuscator $\mathcal{O}$ is extraction resistant if no PPT adversary can extract exploitable information from $\mathcal{O}(C)$ with non-negligible probability.

Theorem 5.2 (TLO Attack Resistance)**.** *Under* LWE *hardness, topology empirical security, and the wrong-key-gives-garbage property (1-bit on-chain oracle), TLO achieves extraction resistance against our 6-class attack matrix.*

Proof: Structural/statistical attacks are defeated by the topology layer (empirical). RainbowTable requires subcircuit evaluation, blocked by LWE CF hiding. □

5.2 Attack Evaluation Matrix

Attack	Defense	Basis	Status
Compression	Topology	Structural	Blocked
PatternMatch	Topology	Structural	Blocked
Structural	Topology	Structural	Blocked
Statistical	Topology	Statistical	Blocked
DiagCorrelation	Topology	Statistical	Blocked
RainbowTable	LWE	Semantic	Blocked

Table 2: TLO attack resistance (empirical, not universal).

5.3 Security Estimates

Uniform-Secret LWE**.** TLO derives the LWE secret as $s_{\mathsf{enc}} = H(\mathsf{secret})$, producing a *uniform* secret over $\mathbb{Z}_q^n$ rather than a small-coefficient secret. This variant is *harder* to attack: primal (uSVP) attacks fail when $\|s\| \approx \sqrt{n} \cdot q/2$. We validated this via BKZ attacks using fpylll—BKZ-50 on n=16 failed after 200+ iterations.

Using the official `lattice-estimator` with our parameters (n=64, q=65521, σ=1024), we obtain ∼108-bit classical security via BDD attack analysis. The larger noise (σ=1024 vs. $\sqrt{q}/4 \approx 64$) is safe because $\sigma \ll q/4$=16380, ensuring negligible decryption error. For n=32: ∼51-bit; for n=128: ∼203-bit.

Hash-Compare Baseline: A simple `H(secret) == H(input)` check costs ∼45K gas but provides *no* obfuscation—the predicate structure is visible on-chain. TLO hides control functions at 57× gas cost.

Multi-Bit Output: The Key Distinction. Hash-compare returns a 1-bit output (true/false). TLO circuits compute an N-bit output that can encode hidden parameters, computed results, or payloads revealed only on correct input. Both implement *point functions*—predicates meaningful only at $x = \mathsf{secret}$—but TLO provides a hidden payload, not just confirmation.

Approach	Output	What's Hidden
Hash-compare	1 bit	Secret value only
TLO	N bits	Secret + hidden computation

The 57× gas premium buys multi-bit hidden computation, not stronger unlocking security. Use TLO when the payload matters; use hash-compare for simple confirmation.

5.4 Post-Quantum Security

TLO is post-quantum resistant (assuming LWE quantum-resistance):

- **Topology layer:** No cryptographic assumptions
- **Lattice layer:** LWE is believed quantum-resistant

5.5 Assumptions

1. LWE **hardness:** Learning With Errors is computationally hard.

2. **Topology empirical security:** Wire selection defeats structural attacks in our evaluation (heuristic, not proven).

3. **Contract correctness:** Expiry logic is correctly implemented.

6 Implementation

6.1 Contract Architecture

TLO requires no external infrastructure—just a standard smart contract with timestamp-based expiry:

Deployment: Set `secretExpiry` at deployment.

6.2 Gas Costs

Measured on 64-wire/640-gate circuits (Tenderly-confirmed):

Listing 2: TLOHoneypot Contract

```
contract TLOHoneypot {
  uint256 public secretExpiry;
  bytes32 public commitHash;

  function check(bytes32 s) external view
      returns (bool) {
    require(block.timestamp < secretExpiry);
    return evaluate(s);
  }

  function commit(bytes32 h) external {
    commitHash = h;
  }

  function reveal(bytes32 s) external {
    require(keccak256(abi.encode(s,
        msg.sender)) == commitHash);
    require(evaluate(s));
    // Transfer reward
  }
}
```

LWE n	Security (σ=1024)	Gas	Block %
32	~51-bit	1.27M	4.2%
64	**~108-bit**	**2.58M**	**8.6%**
96	~178-bit	3.0M	10.0%
128	~203-bit	3.8M	12.7%

Table 3: Gas costs by LWE dimension (σ=1024, validated via `lattice-estimator`).

7 Evaluation

7.1 Attack Resistance

We evaluated TLO against 14 attack implementations across 6 attack classes. All configurations achieve 6/6 resistance:

LWE n	Score	Gas	Security (σ=1024)
32	6/6	1.27M	~51-bit
64	**6/6**	**2.58M**	**~108-bit**
96	6/6	3.0M	~178-bit
128	6/6	3.8M	~203-bit

7.2 Comparison with Alternatives

Property	TLO	iO	TEE
Attack resistance	6/6	6/6	6/6
Secret keys	None	None	Required
Gas (check)	2.58M	$10^6\times$	$1\times$
Infrastructure	None	None	Hardware
Post-quantum	Yes	Depends	No
Practical	Yes	No	Yes

8 Applications

8.1 Valid Applications

TLO is designed for predicates with *eventually-expiring* secrets:

- **Cryptographic honeypots:** Reward condition is burned once triggered
- **Sealed-bid auctions:** Bids revealed at settlement
- **Lotteries/prediction markets:** Outcomes revealed after close
- **MEV protection:** Order flow is short-lived
- **Dark pools:** Trade conditions expire quickly

8.2 Invalid Applications

TLO is *not* intended for long-lived static secrets:

- Long-term decryption keys
- Permanent signing keys
- Static liquidation thresholds

9 Limitations

Theoretical: Topology security is empirical (heuristic, not proven). We do not claim iO-level indistinguishability.

Practical: TLO with n=64 LWE requires ~2.58M gas (8.6% of block limit). Lower security configurations available for cost-sensitive applications.

What TLO does NOT provide:

- Indistinguishability: Two circuits have distinguishable obfuscations
- Universal security: Only resists our 6 attack classes
- Forward secrecy: Expired secrets may be analyzed retroactively
- Security after LWE compromise: If CF bits are recovered, the reversible circuit can be inverted in linear time. Topology only hardens *pre-compromise* attacks

10 Related Work

Indistinguishability Obfuscation: Theoretical iO [2, 4] provides strong security but requires impractical overhead.

Compute-and-Compare: Goyal-Koppula-Waters [3] and Wichs-Zirdelis [6] introduced C&C for evasive functions. We apply it to control function hiding.

Smart Contract Privacy: Previous work uses ZK-SNARKs (Tornado Cash) or TEEs (Secret Network). TLO provides a new point in the design space: on-chain obfuscation without trusted hardware.

11 Conclusion

TLO provides practical circuit obfuscation for smart contracts through two-layer defense: a topology layer (heuristic) defeats structural/statistical attacks, while on-chain LWE inner products defeat semantic attacks.

TLO achieves 6/6 resistance against our attack matrix at ∼2.58M gas (n=64, ∼108-bit security with σ=1024). Post-quantum resistant (assuming LWE quantum-resistance). Deployment requires only a standard smart contract with timestamp expiry.

Key contributions: On-chain LWE inner products for true CF hiding; wrong-key-gives-garbage property combined with 1-bit oracle interface; discovery that larger noise (σ=1024 vs. $\sqrt{q}/4$) provides dramatically higher security at zero additional cost.

Code and attack suite: https://github.com/igor53627/tlo

References

[1] National Institute of Standards and Technology. Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM). NIST FIPS 203, 2024.

[2] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS*, 2013.

[3] R. Goyal, V. Koppula, and B. Waters. Lockable obfuscation. In *FOCS*, 2017.

[4] A. Jain, H. Lin, and A. Sahai. Indistinguishability obfuscation from well-founded assumptions. In *STOC*, 2021.

[5] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.

[6] D. Wichs and G. Zirdelis. Obfuscating compute-and-compare programs under LWE. In *FOCS*, 2017.

[7] M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.