Lux Teleport: Sovereign Omnichain Liquidity via Threshold Cryptography and Per-Chain Governance

Zach Kelling

Lux Industries

2022 (Revised December 2025)

Abstract

We present **Lux Teleport**, a non-custodial omnichain bridge and liquidity protocol that natively connects 18 execution environments and 270 registered chain IDs through a unified MPC threshold signature network. Unlike existing bridges that rely on trusted intermediaries, multisigs, or chain-specific messaging layers, Teleport achieves trustless cross-chain asset transfer through FROST [1] and CGGMP21 [2] threshold cryptography with MPC group key verification (a single aggregate public key verified via `ecrecover` on EVM chains), per-chain nonce tracking, and automatic undercollateralization detection.

Teleport introduces three innovations: (1) *sovereign chain governance* where each connected chain's native token holders control their own bridge fees, yield strategies, and risk parameters; (2) a *yield-bearing bridge token* system where locked assets on source chains are deployed to yield strategies and the yield is reflected in bridge token share prices on destination chains; and (3) a *Shariah-compliant mode* that programmatically filters yield sources to enable the first protocol-level Islamic finance DeFi infrastructure.

The protocol supports 18 native bridge programs across every major smart contract runtime: EVM (Solidity), Solana (Anchor/Rust), TON (FunC), Sui and Aptos (Move), Cosmos/IBC (CosmWasm), Bitcoin L1 (FROST Taproot), OP_NET (AssemblyScript), XRPL (Hooks), Polkadot (ink!), NEAR (Rust/WASM), Stellar (Soroban), Stacks (Clarity), Cardano (Aiken/Plutus), StarkNet (Cairo), ICP (Rust Canisters), Algorand (PyTeal), Fuel (Sway), and Tezos (CameLIGO). All bridge tokens implement the vendor-neutral `IBridgeToken` interface (not tied to any specific bridge vendor). The OmnichainRouter contract is non-upgradeable, with a hard-coded 1% maximum bridge fee, configurable rotation delay (default 24h, minimum 0 for instant rotation, maximum 7 days; operational parameter, not a security invariant), and automatic pause on undercollateralization below 98.5%. The `batchMintDeposit` function enables GPU-optimized batch processing of multiple deposits in a single transaction.

This paper supersedes Lux Teleport Protocol (2022) [13] and Lux Bridge (2023) [14].

Contents

1 Introduction

Cross-chain interoperability remains the critical unsolved problem in blockchain infrastructure. The total value locked across bridges exceeds $15B, yet bridge exploits account for over $2.8B in losses since 2020 [12]. We identify four classes of existing solutions and their fundamental limitations:

1. **Trusted multisig**: Custodial risk, single points of failure (WBTC, most centralized bridges). Failure mode: compromise m-of-n keys $\rightarrow$ drain all funds.
2. **Optimistic**: Long withdrawal delays (7–14 days), fraud proof windows (Across, Hop). Failure mode: censored fraud proofs $\rightarrow$ invalid state accepted.
3. **Light client**: Chain-specific implementation per pair, high complexity ($O(n^2)$ implementations for n chains). Failure mode: consensus bug in light client.
4. **MPC threshold**: Distributed key management, no single custodian, but key management complexity. Failure mode: $\geq t$ node compromise.

Teleport addresses the MPC approach's key management complexity through two complementary threshold signature protocols:

- **FROST** [1]: Flexible Round-Optimized Schnorr Threshold for Ed25519 and BIP-340 Taproot, providing 64-byte signatures in 2 rounds. Covers 14 chain families.
- **CGGMP21** [2]: Threshold ECDSA with identifiable aborts for secp256k1. Covers 6 chain families. Standard `ecrecover` on all EVM chains.

Contributions. This paper makes the following contributions:

1. Formal description of the OmnichainRouter contract with provable security properties (§3).
2. Sovereign governance model separating security, economic, and protocol authority domains (§5).
3. Yield-bearing bridge token architecture with 29 deployed yield strategies across 9 categories (§6).
4. First protocol-level Shariah compliance framework for DeFi (§7).
5. 18 native bridge programs across 270 registered chain IDs (§8).

6. Formal security analysis proving conservation, uniqueness, liveness, safety, governance isolation, and exit guarantee properties (§9).

2 System Model

2.1 Entities

Definition 1 (MPC Signer Set)**.** *A set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ of n nodes, each holding a share of a distributed secret key generated through distributed key generation (DKG) with no trusted dealer. A threshold $t \leq n$ defines the minimum number of honest nodes required to produce a valid signature. Default: $n = 3, t = 2$. On-chain verification uses a single* MPC group address*—the aggregate public key derived from the DKG. Individual signer identities exist for accountability and identifiable abort but are not exposed on-chain: $\texttt{ecrecover}(digest, \sigma) = \texttt{mpcGroupAddress}$.*

Definition 2 (OmnichainRouter)**.** *An immutable (non-upgradeable, non-proxied) smart contract deployed on each connected chain that:*

1. *Verifies MPC threshold signatures for mint/release operations*
2. *Maintains per-source-chain nonce bitmaps for replay prevention*
3. *Distributes bridge fees to sovereign stakeholders*
4. *Auto-pauses when backing ratio falls below 98.5%*

Definition 3 (Sovereign Governor)**.** *The native chain's DAO timelock address G_C for chain C. Controls economic parameters:*

- *Bridge fee rate: $0 \leq f_b \leq 100$ bps (hard-capped, immutable maximum)*
- *Stakeholder share: $0 \leq f_s \leq 10{,}000$ bps*
- *Token registration and daily mint limits*
- *Shariah compliance mode activation*

G_C cannot*: mint tokens, modify nonce bitmaps, bypass collateralization checks, or shorten timelock durations.*

2.2 Threat Model

We assume a computationally bounded adversary $\mathcal{A}$ who may compromise up to $t - 1$ MPC signer nodes, control the governor address on any single chain, observe and reorder transactions on any chain, and deny service to individual relay nodes.

We assume $\mathcal{A}$ *cannot* simultaneously compromise $\geq t$ MPC signer nodes, break the Discrete Logarithm Problem (DLP) or the Elliptic Curve Discrete Logarithm Problem (ECDLP), or rewrite confirmed blocks on any connected chain (finality assumption).

3 Protocol Description

3.1 Bridge Operations

The protocol supports four atomic operations, each protected by threshold signature verification and nonce uniqueness.

Lock. User deposits asset A of amount a on source chain C_s into the YieldBridgeVault. The vault emits:

$$\texttt{Lock}(C_s, C_d, \nu, A, a, r)$$

where C_d is the destination chain, ν is a monotonically increasing nonce, and r is the recipient address on C_d.

Mint. MPC signer set $\mathcal{S}$ observes the Lock event, reaches quorum ($\geq t$ signers agree), and produces threshold signature σ over:

$$m = H\big(\texttt{"DEPOSIT"} \,\|\, C_s \,\|\, \nu \,\|\, T \,\|\, r \,\|\, a\big) \tag{1}$$

where H is keccak256 for EVM chains or SHA-256 for non-EVM, and T is the bridge token address. Any relayer submits $(C_s, \nu, T, r, a, \sigma)$ to the OmnichainRouter on C_d.

The router:

1. Verifies σ against the MPC public key
2. Checks $\mathcal{N}(C_s, \nu) = 0$ (nonce not processed)
3. Checks daily mint limit: $\texttt{dailyMinted}[T] + a \leq \texttt{dailyMintLimit}[T]$
4. Sets $\mathcal{N}(C_s, \nu) \leftarrow 1$ (marks nonce processed)
5. Computes fee: $f = a \cdot f_b / 10{,}000$
6. Mints $(a - f)$ tokens to recipient r
7. Distributes fee: $f \cdot f_s / 10{,}000$ to stakeholder vault, remainder to treasury

Burn. User burns bridge tokens on C_d, emitting:

$$\texttt{Burn}(C_d, C_s, \nu', T, a, r')$$

This is fully permissionless—no MPC signature required.

Release. MPC produces σ' over:

$$m' = H\big(\texttt{"RELEASE"} \,\|\, C_d \,\|\, \nu' \,\|\, T \,\|\, r' \,\|\, a\big) \tag{2}$$

Note the distinct prefix (`"RELEASE"` vs `"DEPOSIT"`) for domain separation, preventing cross-operation signature reuse.

3.2 Nonce Management

Each OmnichainRouter maintains a per-source-chain nonce bitmap:

$$\mathcal{N} : \mathbb{Z}_{\text{chainId}} \times \mathbb{Z}_{\text{nonce}} \to \{0, 1\} \tag{3}$$

The check-and-set operation $\big(\text{assert } \mathcal{N}(C_s, \nu) = 0;\ \text{set } \mathcal{N}(C_s, \nu) \leftarrow 1\big)$ executes atomically within a single transaction. Combined with the deterministic message hash, this ensures each deposit is processed exactly once across all destination chains.

3.3 Backing Attestation

Every 12 hours, the MPC signer set attests to the total backing on each source chain:

$$\sigma_b = \text{Sign}\big(H(\texttt{"BACKING"} \,\|\, T \,\|\, B \,\|\, \tau)\big) \tag{4}$$

where B is the total backing amount and τ is a monotonically increasing timestamp. If:

$$\frac{B}{\texttt{totalMinted}[T]} < 0.985 \tag{5}$$

the OmnichainRouter automatically sets $\texttt{paused} \leftarrow \texttt{true}$. Unpausing requires a separate MPC signature.

4 Threshold Cryptography

4.1 FROST Protocol

For chains supporting Ed25519 or Schnorr (Solana, TON, Sui, Aptos, Cosmos, Bitcoin Taproot, Polkadot, NEAR, Stellar, Cardano, Algorand, Tezos), we employ FROST [1].

Algorithm 1 FROST t-of-n Threshold Signing

Require: Message m, participant set $P \subseteq \mathcal{S}$, $|P| \geq t$
Ensure: Threshold Schnorr signature $\sigma = (R, s)$

Round 1 (Commitment):
for all $p_i \in P$ **do**
    Sample nonce pair $(d_i, e_i) \xleftarrow{\$} \mathbb{Z}_q^2$
    Broadcast commitment $(D_i, E_i) = (g^{d_i}, g^{e_i})$
end for
Round 2 (Signature):
Compute binding factor: $\rho_i = H_1(i, m, \{(D_j, E_j)\}_{j \in P})$
Compute group commitment: $R = \prod_{i \in P} D_i \cdot E_i^{\rho_i}$
Compute challenge: $c = H_2(R, Y, m)$
for all $p_i \in P$ **do**
    Compute partial: $z_i = d_i + e_i \rho_i + \lambda_i s_i c$
end for
Aggregate: $s = \sum_{i \in P} z_i$
return $\sigma = (R, s)$

where λ_i are Lagrange interpolation coefficients and s_i are Shamir secret shares of the group private key.

Theorem 1 (FROST Unforgeability [1])**.** *Under the Discrete Logarithm assumption in the Random Oracle Model, FROST is existentially unforgeable under chosen-message attack for any adversary corrupting at most $t - 1$ of n participants.*

4.2 CGGMP21 Protocol

For EVM chains, TRON, XRPL, and StarkNet (secp256k1 ECDSA), we employ CGGMP21 [2]. The protocol produces standard (r, s, v) ECDSA signatures verifiable via `ecrecover` on all EVM chains.

Theorem 2 (CGGMP21 Security [2])**.** *CGGMP21 UC-realizes ideal threshold ECDSA signing with identifiable abort under the Strong RSA, DDH, and Paillier assumptions, tolerating $t - 1$ malicious corruptions.*

4.3 Taproot Integration

For Bitcoin L1 custody, FROST operates in Taproot mode via `frost.KeygenTaproot()`, producing an x-only public key $\tilde{X}$ (32 bytes, BIP-340 [3]). The Taproot address `bc1p`$\langle$Bech32m$(\tilde{X})\rangle$ serves as the bridge vault. No on-chain smart contract is required; Bitcoin Script's native Schnorr verification validates threshold signatures.

4.4 Signature Verification by Chain Family

Family	Scheme	Sig.	Gas/Cost
EVM (30+)	ECDSA	65 B	3,000
Solana	Ed25519	64 B	0 (sysvar)
TON	Ed25519	64 B	∼0.01 TON
Sui	Ed25519	64 B	∼0.001 SUI
Aptos	Ed25519	64 B	∼100 gas
Cosmos	Ed25519	64 B	∼50,000 gas
Bitcoin	Schnorr	64 B	1 vbyte
Polkadot	Ed25519	64 B	∼0.01 DOT
NEAR	Ed25519	64 B	∼0.001 N
Stellar	Ed25519	64 B	100 stroops
Cardano	Ed25519	64 B	∼0.17 ADA
StarkNet	Stark ECDSA	64 B	∼1,000 gas
Tezos	Ed25519	64 B	∼0.01 XTZ
Algorand	Ed25519	64 B	0.001 ALGO

Table 1: Signature verification across chain families. FROST produces a single 64-byte Schnorr/Ed25519 signature regardless of the t-of-n threshold.

5 Sovereign Chain Governance

5.1 Separation of Concerns

Teleport cleanly separates three authority domains:

Domain	Controller	Scope
Security	MPC signer set	Mint/release authorization
Economics	Chain governor	Fees, limits, strategies
Protocol	Immutable code	Nonce checks, verification

Table 2: Authority domain separation. No domain can override another.

Property 1 (Governance Isolation)**.** *For any governor G_C of chain C:*

1. *G_C cannot call `bridgeMint` without valid MPC signature σ*
2. *G_C cannot modify $\mathcal{N}$ (nonce bitmap) directly*
3. *G_C cannot set $f_b > 100$ bps (hard-coded maximum)*
4. *G_C cannot change stakeholder vault without 48-hour timelock*
5. *G_C cannot change MPC signers (MPC-controlled, configurable delay up to 7 days)*

5.2 Per-Chain Deployment

Each chain deploys an independent OmnichainRouter with its own governance:

Chain	Governor	Fee Recipient
Lux C-Chain	LUX DAO	xLUX (LiquidLUX)
Zoo EVM	ZOO DAO	ZOO Stakers
Third-party L1	Their DAO	Their staking vault

The MPC signer set can be shared across chains or each chain can maintain its own. In either case, the fee flow is sovereign to each chain.

5.3 Signer Rotation

The rotation delay is a configurable operational parameter set by the governor (default 24 hours, minimum 0 for instant rotation, maximum 7 days). The delay is an operational convenience—not a security invariant—because the MPC threshold *is* the trust model. A compromised governor cannot rotate signers; only the current t-of-n MPC set can initiate rotation.

1. Current t-of-n MPC set signs rotation proposal with new MPC group address.
2. Configurable delay begins (default 24h, governor-settable $\in [0, 7\text{d}]$). Pending signers visible on-chain.
3. During delay: users can exit (burn tokens $\rightarrow$ receive underlying).
4. After delay expires: anyone calls `executeSignerRotation()`.
5. Current signers can cancel during delay via `cancelSignerRotation()`.

6 Yield-Bearing Bridge Tokens

6.1 Share-Based Accounting

Yield-bearing bridge tokens (yLETH, yLBTC, yLUSD) use ERC-4626-like [6] share accounting:

$$\text{sharePrice}(t) = \frac{\text{totalBacking}_{\text{source}}(t)}{\text{totalShares}_{\text{dest}}(t)} \tag{6}$$

When yield accrues on the source chain, the MPC attests to new backing via Equation (1) with `"BACKING"` prefix. The share price increases automatically—holders earn yield without transactions.

6.2 Strategy Categories

29 deployed strategies span 9 categories:

Category	Protocols	APY
Liquid Staking	Lido, Rocket Pool	3–5%
Restaking	EigenLayer, Symbiotic, Karak	3–8%
Lending	Aave, Compound, Morpho, Euler, Spark, Fluid	2–12%
Stablecoin	MakerDAO/Sky, Ethena, Frax	4–15%
LP/DEX	Curve, Convex, Pendle, L2 DEXs	5–20%
BTC Native	Babylon, Lombard, SolvBTC, CoreDAO	3–12%
Solana	Marinade, Jito, Kamino	5–10%
TON	Tonstakers, Bemo, STON.fi	5–15%
Perps LP	LPX fee-based LP	10–30%

Table 3: Yield strategy taxonomy. 29 strategies across 9 categories.

6.3 Yield Composition

Bridge token holders earn layered yield:

$$\text{APY}_{\text{total}} = \underbrace{\text{APY}_{\text{base}}}_{\text{staking}} + \underbrace{\text{APY}_{\text{re}}}_{\text{restaking}} + \underbrace{\text{APY}_{\text{bridge}}}_{\text{fee share}} + \underbrace{\text{APY}_{\text{DeFi}}}_{\text{LP/lending}} \tag{7}$$

For yLBTC on Lux: $\sim$5% (Babylon) + $\sim$2% (EigenLayer) + $\sim$1% (xLUX bridge fees) + $\sim$5–15% (DeFi) = 13–23% total.

7 Shariah Compliance

Islamic finance law (Shariah) prohibits *riba* (interest/usury), affecting ∼1.8 billion Muslims. Teleport introduces the `ShariaFilter` contract for protocol-level compliance:

Definition 4 (Compliance Classification)**.** *Each yield strategy Y is assigned a status* $\mathcal{C}(Y) \in \{\textsc{Halal}, \textsc{Haram}, \textsc{Conditional}, \textsc{UnderReview}\}$.

Status	Examples	Rationale
Halal	DEX fees, bridge fees	Service fee
	Validator staking	Network security
	LP provision	Liquidity service
	Babylon BTC staking	Fee-based security
Haram	Aave/Compound	Riba (interest)
	MakerDAO DSR	Savings interest
	sDAI, USDY	Interest-bearing
Conditional	Lido, EigenLayer	Structure-dependent

Table 4: Shariah compliance classification of yield sources

A Shariah Advisory Board (SAB) multisig updates classifications. When `shariahMode` is enabled, only $\mathcal{C}(Y) = \textsc{Halal}$ strategies receive deposits.

8 Supported Execution Environments

18 native bridge programs are deployed across the following execution environments:

#	Runtime	Lang.	Sig.	Chains
1	EVM	Solidity	ECDSA	180+
2	SVM	Rust	Ed25519	3
3	TVM (TON)	FunC	Ed25519	2
4	Sui Move	Move	Ed25519	3
5	Aptos Move	Move	Ed25519	3
6	CosmWasm	Rust	Ed25519	20+
7	OP_NET	AS	Schnorr	1
8	Bitcoin	FROST	Schnorr	3
9	XRPL Hooks	C	ECDSA	3
10	Substrate	Rust	Ed25519	14
11	NEAR	Rust	Ed25519	2
12	Soroban	Rust	Ed25519	2
13	Clarity	Clarity	secp256k1	2
14	Plutus	Aiken	Ed25519	3
15	Cairo	Cairo	Stark	1
16	ICP	Rust	t-ECDSA	1
17	AVM	PyTeal	Ed25519	2
18	FuelVM	Sway	ECDSA	2
19	Michelson	LIGO	Ed25519	2
20	TVM (TRON)	Solidity	ECDSA	3

Table 5: 18 native bridge programs, 270 registered chain IDs. All bridge tokens implement the vendor-neutral `IBridgeToken` interface.

9 Formal Security Properties

Property 2 (Conservation)**.** *For every bridge token T at time t:*

$$\texttt{totalMinted}[T](t) \leq \texttt{totalBacking}[T](t) \tag{8}$$

Enforced by automatic pause when $B/M < 0.985$.

Property 3 (Uniqueness)**.** *For all source chains C_s and nonces ν:*

$$\left|\{tx : tx \text{ processes } (C_s, \nu)\}\right| \leq 1 \tag{9}$$

Enforced by atomic check-and-set on nonce bitmap.

Property 4 (Liveness)**.** *If $\geq t$ of n MPC nodes are online and at least one relayer is operational, the bridge processes new deposits within one source-chain confirmation period plus one destination-chain block time.*

Property 5 (Safety)**.** *Under the DLP/ECDLP assumptions:*

$$|\mathcal{A} \cap \mathcal{S}| < t \implies \Pr[\mathit{forge}(\sigma)] \leq \mathit{negl}(\lambda) \tag{10}$$

Property 6 (Exit Guarantee)**.** *Users can burn bridge tokens and receive underlying assets within* $\max(\mathit{rotationDelay}, \mathit{confirm\_time})$ *under all conditions, including governor compromise and single MPC node failure. The rotation delay is configurable (default 24h, max 7 days).*

10 Fee Economics

10.1 Fee Structure

$$\text{mintAmount} = \text{deposit} \cdot \left(1 - \frac{f_b}{10{,}000}\right) \tag{11}$$

$$\text{toStakeholders} = f_b \cdot \text{deposit} \cdot \frac{f_s}{10{,}000^2} \tag{12}$$

$$\text{toTreasury} = f_b \cdot \text{deposit}/10{,}000 - \text{toStakeholders} \tag{13}$$

Default: $f_b = 10$ bps (0.1%), $f_s = 9{,}000$ bps (90% to stakeholders).

10.2 xLUX Flywheel

On Lux, the stakeholder vault is LiquidLUX (xLUX). All fees compound:

$$\text{APY}_{\text{xLUX}} = \frac{\sum_p \text{fees}_p}{\text{totalStaked}_{\text{LUX}}} \tag{14}$$

where p ranges over bridge, DEX, lending, perps, and NFT AMM protocols. This creates a positive feedback loop: more bridged assets → more yield → more fees → higher xLUX APY → more LUX staked → deeper liquidity.

11 Related Work

Wormhole [7] uses a 13-of-19 guardian multisig. Teleport's 2-of-3 MPC is more capital-efficient but trades off decentralization breadth for operational simplicity and faster key rotation.

LayerZero [8] provides generic messaging (Oracle + Relayer). Teleport is bridge-specific, optimizing for asset transfer with yield composition.

IBC [9] achieves trust-minimized Cosmos-to-Cosmos transfers via light client verification. Teleport's CosmWasm bridge integrates with IBC for inter-Cosmos transfers while using MPC for non-IBC chains.

tBTC [10] pioneered threshold ECDSA for Bitcoin bridging. Teleport generalizes this to 18 native bridge programs across 270 chain IDs with yield-bearing tokens and sovereign governance.

Axelar [11] runs a dedicated PoS chain for cross-chain communication. Teleport avoids the overhead of a separate consensus chain by embedding MPC signing in existing infrastructure.

12 Conclusion

Lux Teleport provides the infrastructure for a sovereign omnichain liquidity layer: non-custodial, non-upgradeable, per-chain governed, optionally Shariah-compliant, and natively deployed across every major blockchain execution environment. By separating attestation (MPC), governance (per-chain DAO), and protocol logic (immutable code), Teleport enables any blockchain to join the liquidity network while retaining full sovereignty over its economic parameters.

The yield-bearing bridge token system makes bridged capital productive across 29 strategies, the Shariah compliance mode opens DeFi to 1.8 billion Muslims worldwide, and the 270-chain registry covers every blockchain in the top 200 by TVL.

All source code is open-source at `github.com/luxfi`.

References

[1] C. Komlo and I. Goldberg, "FROST: Flexible Round-Optimized Schnorr Threshold Signatures," in *Selected Areas in Cryptography (SAC)*, 2020.

[2] R. Canetti, R. Gennaro, S. Goldfeder, N. Makriyannis, and U. Peled, "UC Non-Interactive, Proactive, Threshold ECDSA with Identifiable Aborts," in *ACM CCS*, 2020.

[3] P. Wuille, J. Nick, and T. Ruffing, "BIP-340: Schnorr Signatures for secp256k1," Bitcoin Improvement Proposal, 2020.

[4] P. Wuille, J. Nick, and A.J. Towns, "BIP-341: Taproot: SegWit version 1 spending rules," Bitcoin Improvement Proposal, 2020.

[5] R. Floersch and J. Baylina, "EIP-712: Typed Structured Data Hashing and Signing," Ethereum Improvement Proposal, 2017.

[6] J. Bebis *et al.*, "EIP-4626: Tokenized Vault Standard," Ethereum Improvement Proposal, 2022.

[7] Wormhole Foundation, "Wormhole: A Generic Message Passing Protocol," Whitepaper, 2022.

[8] R. Zarick, B. Pellegrino, and C. Banister, "LayerZero: An Omnichain Interoperability Protocol," Whitepaper, 2022.

[9] C. Goes *et al.*, "The Interblockchain Communication Protocol," Cosmos Documentation, 2020.

[10] M. Luongo and C. Ream, "tBTC: A Decentralized Redeemable BTC-backed ERC-20 Token," Keep Network, 2020.

[11] S. Layr *et al.*, "Axelar: Connecting Web3," Whitepaper, 2022.

[12] Rekt News, "Leaderboard," https://rekt.news/leaderboard/, 2024.

[13] Z. Kelling and V. Seesahai, "Lux Teleport Protocol," Lux Industries, Inc., 2022. *(Superseded by this paper.)*

[14] Z. Kelling and V. Seesahai, "Lux Bridge: Threshold-Secured Cross-Chain Communication with MPC Custody," Lux Industries, Inc., 2023. *(Superseded by this paper.)*

[15] Z. Kelling, "Lux Universal Threshold Signatures," Lux Industries, Inc., 2021. *(Companion paper: threshold crypto details.)*