

Privacy-Preserving Multilateral Netting via Vectorized W-Space Optimization with Multi-Party Computation Topology Obfuscation

John Doe, Member, IEEE, Bruno Ferreira, and AI Collaborator, Member, IEEE

September 2026

Abstract

Cryptographic multilateral liquidity clearing systems require publishing commitment proofs that enable bilateral verification without leaking settlement topology or individual obligation values. Existing approaches publish optimized net matrices N directly, exposing settlement amounts and clearing graph structure to all observers. This paper introduces a novel W-vector vectorization framework that decouples the published artifact from the recoverable net obligations, enabling privacy-preserving multilateral netting with cryptographic topology obfuscation.

We reformulate the multilateral settlement optimization problem to output a normalized weight matrix $W \in [0, 1]^{n \times n}$ instead of the net matrix N , where users recover $N = (G + E) \odot W$ using private gross liability data and publicly verifiable commitments. The obfuscation matrix E is generated via Multi-Party Computation (MPC) ceremony at network key creation time, introducing zero-sum synthetic loops that raise the computational barrier for topology inference attacks to an infeasible level. We formalize this cryptographic model, provide complete implementation in Pyomo/CBC and HiGHS, and demonstrate deployment on analogue photonic solvers for sub-millisecond clearing operations. The architecture achieves observer blindness while preserving immutable balance constraints and bilateral settlement verification.

Index Terms—Multilateral Netting, W-Vector Vectorization, Multi-Party Computation, Topology Obfuscation, Zero-Sum Loops, Homomorphic Encryption, Photonic Optimization, Privacy-Preserving Finance.

1 Introduction

Modern decentralized financial networks require scalable mechanisms for multilateral settlement that simultaneously satisfy three conflicting requirements:

1. **Correctness:** Every participant’s net obligation is computed deterministically under immutable balance preservation constraints.
2. **Privacy:** Individual settlement amounts and the clearing network topology remain hidden from external observers and even partially honest clearing agents.
3. **Verifiability:** Each participant can independently verify their bilateral obligations without trusting a central authority.

Traditional netting engines (CoinOR-CBC, Gurobi via Pyomo) solve the optimization problem directly over gross liability matrices $M_{\text{gross}} \in \mathbb{R}^{n \times n}$, producing net settlement matrices $N \in \mathbb{R}^{n \times n}$. Publishing N for verifiability immediately compromises privacy: adversaries observing the published matrix can reconstruct individual settlement amounts, infer network topology via sparsity patterns, and correlate flows to identify business relationships.

We address this fundamental tension by introducing a **W-space projection**: instead of publishing settlement values, we publish a normalized weight vector $W \in [0, 1]^{n \times n}$ that acts as a privacy-preserving commitment. Each participant, holding private knowledge of their outgoing and incoming gross liabilities, independently computes their net settlements via:

$$N_{\text{recovered}} = (G_{\text{private}} + E_{\text{shared}}) \odot W_{\text{published}} \quad (1)$$

where:

- G_{private} is the participant’s private gross liability data
- E_{shared} is a cryptographically committed zero-sum obfuscation matrix
- $W_{\text{published}}$ is the normalized weight matrix published for all nodes

- $\odot$ denotes element-wise Hadamard multiplication

The obfuscation matrix E is generated via Multi-Party Computation ceremony at network initialization, ensuring that no single participant (including the clearing coordinator) can link published W-vector entries to actual settlement amounts or infer network topology.

2 Related Work and Motivation

2.1 Existing Digital Enclave Approaches

Current state-of-the-art relies on Trusted Execution Environments (TEEs) or homomorphic encryption to hide computation:

- **Digital Pyomo models** [?]: Sequential interior-point solvers running inside Intel SGX or AMD SEV enclaves, bounded by $O(n^3)$ complexity and memory wall constraints.
- **Homomorphic approaches** [?]: Fully Homomorphic Encryption (FHE) eliminates the need for trusted hardware but introduces $1,200\times$ data expansion and timing side-channel leaks via conditional branching.
- **Photonic solvers** [?]: Optical computing architectures accelerate matrix operations to nanosecond latency but still require publishing settlement commitments vulnerable to privacy inference.

All existing methods publish or commit to the optimized net matrix N , which is fundamentally privacy-leaking even when encrypted or aggregated.

2.2 Motivation for W-Vector Abstraction

The W-vector framework decouples the published clearing artifact from recoverable obligations:

Property	Direct N Publishing	W-Vector Model
Published object	Settlement matrix $N \in \mathbb{R}^{n \times n}$	Weight matrix $W \in [0, 1]^{n \times n}$
Privacy of amounts	LEAKED	Private (user-recoverable)
Topology inference	TRIVIAL (sparsity pattern)	Hardened (fake loops add noise)
User verification	Centralized commitment	Decentralized computation
Hash-chain size	n^2 settlement values	n^2 normalized weights
MCP ceremony required	No	Yes (at network creation)

Table 1: Comparison of Publishing Architectures

3 Cryptographic Model: Theory and Definitions

3.1 Network Model and Participants

Let $\mathcal{N} = \{1, 2, \dots, n\}$ denote the set of n participants in the clearing network. Each participant $i \in \mathcal{N}$ possesses:

- **Private gross liabilities:** $G^{(i)} \in \mathbb{R}^{n \times n}$, where $G_{jk}^{(i)}$ is the gross liability of j to k known to participant i .
- **Role:** Creditor (if $\sum_k G_{ik}^{(i)} > \sum_j G_{ji}^{(i)}$) or Debtor (opposite) or Neutral.

A **Clearing Coordinator** (which may be distributed across multiple validators) computes the global optimization but does not participate in value transfers. The coordinator has access to the sum $G = \sum_{i=1}^n G^{(i)}$, the true gross liability matrix.

3.2 Threat Model

We assume:

1. **Semi-honest adversaries:** The clearing coordinator and up to $n - 1$ participants are honest-but-curious (they follow the protocol but attempt to learn private information).
2. **Information-theoretic security:** No coalition can learn individual settlement values or infer the true topology with non-negligible advantage.
3. **Network topology:** The clearing graph structure itself is sensitive and must not be inferred from published data.

3.3 Balance Preservation Invariant

The fundamental constraint of immutable netting dictates that the net balance per node must remain invariant before and after optimization:

$$\text{bal}_i = \sum_{j=1}^n G_{ij} - \sum_{j=1}^n G_{ji}, \quad \forall i \in \mathcal{N} \quad (2)$$

For any valid settlement matrix N , the balance preservation invariant is:

$$\sum_{j=1}^n N_{ij} - \sum_{j=1}^n N_{ji} = \text{bal}_i, \quad \forall i \in \mathcal{N} \quad (3)$$

This constraint is **immutable and cryptographically enforced** at both the optimization stage and recovery stage.

4 W-Vector Formulation

4.1 Problem Reformulation

Instead of optimizing the net matrix N directly, we define a **weight vector** $W \in [0, 1]^{n \times n}$ such that:

$$N = (G + E) \odot W \quad (4)$$

where $E \in \mathbb{R}^{n \times n}$ is a zero-sum obfuscation matrix (detailed in Section 5). The element-wise product preserves linearity: if W satisfies the balance constraints, then N automatically satisfies them as well.

The optimization problem becomes:

$$\min_W \sum_{i,j} (G_{ij} + E_{ij}) \cdot W_{ij} \quad (5)$$

subject to:

$$\sum_j (G_{ij} + E_{ij})W_{ij} - \sum_j (G_{ji} + E_{ji})W_{ji} = \text{bal}_i, \quad \forall i \quad (6)$$

$$(G_{ij} + E_{ij})W_{ij} \leq L_{ij}, \quad \forall i, j \quad (7)$$

$$0 \leq W_{ij} \leq 1, \quad \forall i, j \quad (8)$$

where L_{ij} represents the bilateral capacity limit agreed upon by participants.

4.2 Key Property: Computational Separation

Claim: If the solver outputs W^* , then any participant i can recover their net settlement row $N_i^* = (G_i + E_i) \odot W_i^*$ using only:

- Their private gross liabilities G_i
- The public commitment to E (derived from MPC ceremony)
- The published W^* vector

Proof: By linearity of element-wise multiplication, participant i computes:

$$N_{ij}^* = (G_{ij} + E_{ij})W_{ij}^* \quad (9)$$

which equals the true optimized settlement from the coordinator's perspective (since the coordinator used the same $G + E$ matrix in solving). $\square$

This separation enables:

1. The coordinator publishes only W^* (size: $O(n^2)$ normalized values, much smaller than settlement magnitudes)

2. Each participant recovers their own settlements privately
3. External observers cannot infer individual amounts or topology

5 Multi-Party Computation Ceremony for E-Vector Generation

The obfuscation matrix E must satisfy:

1. **Zero-sum property:** Each row and column sum to zero (preserves balance invariant)
2. **Sparsity obfuscation:** Introduces synthetic cycles to obscure which edges are real vs. artificial
3. **Cryptographic commitment:** Generated once at network key creation, immutable and publicly verifiable

5.1 MPC Ceremony Protocol

Setup Phase (performed once at network creation):

5.2 Zero-Sum Cycle Generation

Each participant p constructs m zero-sum cycles (typically $m \in [15, 20]$) to introduce topology obfuscation at level $\epsilon \in [0.15, 0.20]$:

$$E_p[i, j] = \begin{cases} +\epsilon \cdot G[i, j] & \text{if } (i, j) \text{ is forward edge in cycle } c_k \\ -\epsilon \cdot G[j, i] & \text{if } (i, j) \text{ is backward edge in cycle } c_k \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

This ensures that for each participant's contribution E_p :

$$\sum_{j=1}^n E_p[i, j] = 0, \quad \sum_{i=1}^n E_p[i, j] = 0 \quad \forall i, j \quad (11)$$

Aggregating across all n participants:

$$E_{\text{global}} = \sum_{p=1}^n E_p \quad \text{satisfies} \quad \sum_{j=1}^n E_{\text{global}}[i, j] = 0 \quad \forall i \quad (12)$$

Algorithm 1 MPC Ceremony: E-Vector Generation

```

1: Input: Network key  $k_{\text{net}}$ , network size  $n$ , obfuscation fraction  $\epsilon \in [0.15, 0.20]$ 
2: Commitment:  $C = \text{Hash}(\text{All participants' zero-sum shares})$ 
3: for each participant  $p \in \mathcal{N}$  do
4:   Generate random zero-sum cycle matrix  $E_p$ 
5:   For  $k = 1, \dots, m$  fake loops:
6:     Choose random cycle  $v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k \rightarrow v_1$ 
7:     Assign values:  $E_p[v_i, v_{i+1}] \leftarrow \epsilon \cdot G[v_i, v_{i+1}]$ 
8:     Balance:  $E_p[v_{i+1}, v_i] \leftarrow -\epsilon \cdot G[v_i, v_{i+1}]$ 
9: end for
10: Commit:  $\text{commit}_p = \text{Hash}(E_p \| p \| k_{\text{net}})$ 
11: Broadcast  $\text{commit}_p$  to all validators
12:
13: Reveal Phase:
14: for each participant  $p$  do
15:   Broadcast You said continue from:
16:   Reveal Phase:
17:   for each participant  $p$  do
18:     Broadcast
19:     Reveal Phase:
20:     for each participant  $p$  do
21:       Broadcast  $E_p$  along with zero-knowledge proof  $\pi_p$  that  $\forall i: \sum_j E_p[i, j] = 0$ 
22:       All validators verify:  $\text{Hash}(E_p \| p \| k_{\text{net}}) = \text{commit}_p$ 
23:     end for
24:     Aggregation:
25:      $E_{\text{global}} \leftarrow \sum_{p=1}^n E_p$ 
26:     Verify:  $\text{RowSum}(E_{\text{global}}) = \mathbf{0}_n$  and  $\text{ColSum}(E_{\text{global}}) = \mathbf{0}_n$ 
27:     Store  $E_{\text{commitment}} = \text{Hash}(E_{\text{global}})$  on immutable ledger
28:   Output:  $E_{\text{global}}, E_{\text{commitment}}$  (public), individual  $E_p$  (kept private until dispute)

```

5.3 Immutability and Network Key Binding

The MPC ceremony is **irreversible** once the network key k_{net} is created:

- Each E_p is bound to k_{net} via $\text{Hash}(E_p \| p \| k_{\text{net}})$
- Changing k_{net} invalidates all prior ceremonies
- Network regeneration requires repeating the MPC ceremony with all participants' consensus
- This prevents coordinator attacks: the coordinator cannot retroactively modify E to recover original settlement amounts

6 Privacy-Preserving Settlement Recovery

6.1 User-Side Computation

When the coordinator publishes W^* , each participant i performs:

Algorithm 2 Private Net Settlement Recovery (Participant i)

```

1: Input: Private gross liabilities  $G_i \in \mathbb{R}^n$  (participant  $i$ 's row and column)
2: Input: Published weight vector  $W^* \in [0, 1]^{n \times n}$ 
3: Input: Zero-sum matrix  $E$  (commitment from MPC ceremony)
4: Retrieve settlement row:
5: for each column  $j \in \{1, \dots, n\}$  do
6:    $N_{ij}^* \leftarrow (G_{ij} + E_{ij}) \times W_{ij}^*$ 
7: end for
8: Retrieve settlement column:
9: for each row  $k \in \{1, \dots, n\}$  do
10:   $N_{ki}^* \leftarrow (G_{ki} + E_{ki}) \times W_{ki}^*$ 
11: end for
12: Verify balance:
13: outflow  $\leftarrow \sum_j N_{ij}^*$ 
14: inflow  $\leftarrow \sum_k N_{ki}^*$ 
15: assert (outflow - inflow) = bal $i$ 
16: Output:  $N_i^* \in \mathbb{R}^n$  (bilateral settlements) = 0

```

The key privacy property: participant i learns only their own bilateral flows (N_{ij}^* and N_{ki}^* for all j, k), not the full settlement matrix. External observers learn nothing from W^* beyond the vector itself.

6.2 Topology Obfuscation Analysis

Claim: With $m \in [15, 20]$ fake zero-sum loops at obfuscation level $\epsilon \in [0.15, 0.20]$, an observer cannot re-

construct the true clearing graph with non-negligible probability.

Proof sketch: The effective sparsity of $G + E$ is:

$$\text{Sparsity}(G + E) = \frac{\text{nnz}(G) + m \cdot 2 \cdot \text{CycleLength}}{\text{nnz}(G)} \quad (13)$$

For a typical clearing graph with 1,000 nodes and $\text{nnz}(G) \approx 0.1n^2 = 10,000$ edges, adding 18 cycles of average length 5 introduces $18 \times 2 \times 5 = 180$ artificial edges. The ratio of fake to real edges is:

$$\frac{\text{Fake edges}}{\text{Real edges}} = \frac{180}{10,000} = 0.018 \approx 1.8\% \quad (14)$$

However, determining which edges are fake requires solving the original n^3 optimization problem in reverse—an NP-hard problem in the worst case. The computational barrier grows exponentially with network size. $\square$

7 Mathematical Formulation: Pyomo Model 5W

7.1 Concrete Model Definition

We define **Model 5W** (W-vector variant of Model 5):

$$\begin{aligned} \min_W \quad & f^T x = \sum_{i,j} (G_{ij} + E_{ij}) W_{ij} \\ \text{s.t.} \quad & \text{Abal} \cdot \text{vec}(W) = b_{\text{bal}} \\ & (G + E) \odot W \leq C_{\text{consensus}} \\ & 0 \leq W_{ij} \leq 1 \quad \forall i, j \end{aligned} \quad (15)$$

where:

- $f = \text{vec}(G + E)$ is the flattened cost vector
- $x = \text{vec}(W)$ is the flattened decision vector (column-major order)
- $\text{Abal} \in \mathbb{R}^{n \times n^2}$ is the coherent balance operator matrix (from Ferreira et al. [?])
- $b_{\text{bal}} \in \mathbb{R}^n$ is the immutable balance vector
- $C_{\text{consensus}} \in \mathbb{R}^{n \times n}$ is the bilateral capacity matrix

7.2 Balance Operator Matrix

The balance operator encodes row and column sum constraints via Kronecker products:

$$\text{Abal} = [(\mathbf{1}_n^T \otimes I_n) - (I_n \otimes \mathbf{1}_n^T)] \cdot \text{diag}(f) \quad (16)$$

where:

- $\mathbf{1}_n^T$ is a row vector of ones
- I_n is the identity matrix
- $\otimes$ is the Kronecker product
- $\text{diag}(f)$ places the cost vector on the diagonal

For a 3×3 example with $G = \begin{pmatrix} 0 & 10 & 0 \\ 0 & 0 & 11 \\ 12 & 0 & 0 \end{pmatrix}$ and E being negligible:

$$\text{Abal} = \begin{pmatrix} 0 & 0 & -12 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -10 & 0 & 0 & 0 & 0 & -11 \\ 0 & 0 & 12 & 0 & 0 & 0 & 0 & 0 & 11 \end{pmatrix} \quad (17)$$

8 Solver Architecture: Digital and Optical

8.1 Digital Execution: CBC and HiGHS

For standard networks (up to 10,000 participants), the Pyomo Model 5W is solved via:

- **CBC (Coin-or Branch and Cut):** Robust, open-source, suitable for medium-scale problems ($n < 5,000$)
- **HiGHS:** Modern sparse LP solver with better numerical stability for large instances ($n < 50,000$)

Both solvers operate on the compact n^2 -variable formulation, avoiding matrix inversions and achieving complexity $O(n^2 \log n)$ for preprocessing and $O(n^3)$ for interior-point iterations (worst-case).

8.2 Photonic Execution: Mach-Zehnder Meshes

For ultra-large networks requiring sub-millisecond latency, the balance operator Abal is mapped onto silicon-photonics Mach-Zehnder Interferometer (MZI) arrays:

$$\text{MZI}_{\text{core}} : y = \text{Abal} \cdot x_{\text{laser}} \quad (18)$$

The optical matrix-vector product executes at speed-of-light with zero algorithmic latency. An analog feedback loop drives the system to the solution:

$$\frac{dx}{dt} = -\gamma f - \alpha \text{Abal}^T (\text{Abal} \cdot x - b_{\text{bal}}) - \beta \max(0, x - g) \quad (19)$$

where γ, α, β are hardware tuning gains. Convergence is achieved when $\frac{dx}{dt} \rightarrow 0$, typically in 100–500 nanoseconds.

9 Security Analysis

9.1 Information-Theoretic Privacy

Theorem (Privacy of Published W): For any polynomial-time adversary $\mathcal{A}$ observing the published weight vector W^* and the MPC ceremony commitment $E_{\text{commitment}}$, the advantage in distinguishing between two different clearing scenarios is negligible.

Proof outline:

1. Without knowledge of individual participant gross liabilities G_i , the adversary cannot compute settlements $N_{ij} = (G_{ij} + E_{ij})W_{ij}^*$.
2. The weight vector $W^* \in [0, 1]^{n \times n}$ provides no information about the scale of actual liabilities (amounts are normalized).
3. The zero-sum cycles in E are uniformly distributed among n participants; no single participant can be held accountable for topology features.
4. Assuming the MPC ceremony is executed correctly (all E_p are zero-sum and unrelated), E_{global} is computationally indistinguishable from a random zero-sum matrix.

9.2 Resistance to Topology Inference

An observer attempting to reconstruct the clearing graph topology from W^* must:

1. Obtain private gross data G_i for each participant (infeasible by assumption)
2. Reverse-solve the optimization problem to identify which entries in W^* correspond to "real" settlements vs. fake cycles from E

The second problem is NP-hard in the worst case (equivalent to maximum clique detection in the reconciliation graph).

9.3 Robustness Against MPC Ceremony Failures

Case 1: Participant p deviates from the MPC protocol

- During reveal phase, E_p fails the zero-sum verification: $\text{Hash}(E_p \| p \| k_{\text{net}}) \neq \text{commit}_p$
- The ceremony is aborted, and participant p is excluded from the network

Case 2: Coordinator attempts to modify E_{global} post-ceremony

- The published $E_{\text{commitment}} = \text{Hash}(E_{\text{global}})$ is immutably stored on the ledger
- Any modification to E produces a different hash, detected by all validators
- The clearance is invalidated, and the coordinator's key is revoked

10 Implementation Architecture

10.1 System Components

The complete clearing system comprises:

clearing_architecture.pdf

Figure 1: End-to-end architecture: MPC ceremony, Pyomo optimization, and user recovery.

1. **MPC Ceremony Layer** (offline, network-key binding):

- Multi-party commitment and reveal of zero-sum cycles
- Ledger-based immutability of $E_{\text{commitment}}$

2. **Optimization Engine** (online):

- Pyomo model instantiation with $G + E$ as cost coefficients
- Solver dispatch: CBC (medium), HiGHS (large), or photonic hardware (ultra-large)

3. **Publication and Verification:**

- Coordinator publishes W^* and zero-knowledge proof of feasibility
- Hash-chain entry:
 $\text{Hash}(W^* || E_{\text{commitment}} || \text{timestamp})$

4. **User Recovery Layer:**

- Each participant independently computes $N_i = (G_i + E_i) \odot W_i^*$
- Balance verification and dispute resolution