xPrivFi Technical Specification

Version 1.0 — Full Protocol Specification

Written by:
BF

bf@xprivfi.info

With documentation support from the XPF Assistant

December 5, 2025

Contents

Abstract

XPrivFi is a minimal Layer-1 blockchain paired with a structured Layer-2 mining system called `HexGrid`. The protocol introduces round-based mining with deterministic reward settlement, predictable emission, and a security model combining RandomHash proof-of-work with off-chain computation rounds. This document provides the full technical specification for developers, security researchers, and auditors.

1 Executive Overview

XPrivFi is built on three fundamental principles:

- **Minimal Layer-1**: A simple, predictable RandomHash blockchain.
- **Round-Based Layer-2 Mining**: `HexGrid` coordinates 6-minute mining rounds, selecting a single winner per cycle.
- **Deterministic Emission**: 500,000 `XPF` mined as 1 `XPF` per round over ∼6.6 years.

The goal is not to compete with large multi-team blockchain foundations, but to introduce a transparent, auditable architecture with minimal attack surface and a new model for mining participation.

The document defines:

1. Full `Layer-1` consensus, block format, networking, and difficulty.
2. Full `Layer-2` round engine specification.
3. Reward settlement and deterministic emission.
4. Threat model (Sybil, 51%, timing attacks, fairness constraints).
5. Future privacy extensions (CP-Shield).
6. Vision for optional RootBaseLayer interoperability.

2 Architecture Summary

XPrivFi consists of two cooperating layers:

2.1 Layer-1 Blockchain

- RandomHash (iPoW) consensus
- Simple UTXO/account hybrid (spec detailed in Section 3)
- 6-minute target block time
- Deterministic reward transactions from `HexGrid`
- Predictable emission supply

2.2 Layer-2 HexGrid Engine

- Off-chain computation of mining rounds
- 6-minute work cycles
- 30-second transparency phase
- 30-second lobby/commit phase
- Fairness-oriented deterministic winner selection
- Winner broadcasts a verified proof to L1

2.3 Data Flow Between Layers

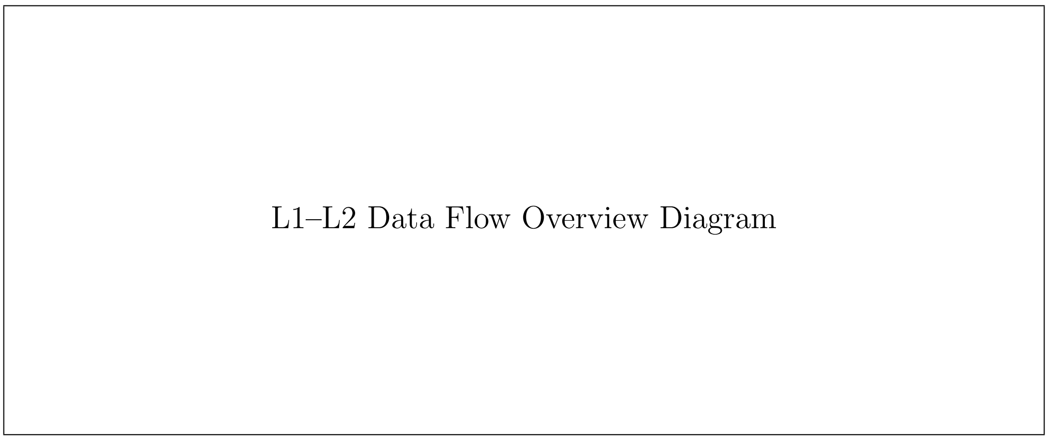

Figure 1: L1–L2 Data Flow Overview Diagram

3 Glossary

XPF — Native currency of XPrivFi.

HexGrid — The Layer-2 mining framework coordinating round-based work.

Round — A 6-minute mining cycle consisting of:

1. Work phase
2. Transparency phase
3. Lobby phase

Deterministic Reward — A guaranteed output of 1 XPF per round from the mining pool.

RandomHash (iPoW) — A CPU-oriented proof-of-work algorithm emphasizing memory hardness and anti-GPU parallelism.

CP-Shield — Optional future privacy layer based on ZK commitments.

CP-Diff — Difficulty model used by L1 block production.

RootBaseLayer (RBL) — An independent experimental settlement concept that may interoperate with XPrivFi in future research.

4 System Goals

- Predictable monetary supply without halving schedules.
- Fairness-oriented mining for regular devices.
- Minimalistic and auditable layer separation.
- High transparency and low protocol complexity.
- Expandability through optional future modules.

High-Level System Architecture Diagram

Figure 2: High-Level System Architecture Diagram

5 Layer-1 Blockchain Specification

Layer-1 (L1) of XPrivFi is a minimal blockchain designed to provide:

- deterministic settlement of Layer-2 mining rewards,
- a predictable block structure,
- a CPU-oriented proof-of-work consensus,
- a stable, human-readable chain model.

L1 is intentionally small in scope. It does not perform smart contract execution or complex state transitions.

5.1 Consensus: RandomHash (iPoW)

XPrivFi uses RandomHash, a memory-hard CPU-oriented hashing algorithm. The purpose is to reduce GPU/ASIC advantages by demanding sequential, memory-bound execution.

5.1.1 RandomHash Properties

- High memory pressure per hash attempt.
- Unfriendly to parallel GPU execution.
- Designed to limit hardware inequality.
- Deterministic evaluation.
- Resistant to ASIC optimization due to irregular memory access patterns.

Let H denote the RandomHash function.

Given block header B_h, mining solves:

$$H(B_h) < D_t$$

where D_t is the active difficulty target.

5.1.2 Difficulty Target

D_t is calibrated using CP-Diff v1 (Section 5.7).

5.2 Block Structure

Each block consists of:

$$\text{Block} = \{\text{Header},\ \text{Body}\}$$

5.2.1 Block Header

Field	**Description**
Version	Protocol version integer.
PreviousHash	Hash of previous block header.
MerkleRoot	Merkle root of body transactions.
Timestamp	Unix time (UTC).
DifficultyTarget	Current target D_t.
Nonce	Value miners vary to find valid hash.
L2CommitHash	Hash summarizing the Round Winner (if present).

Table 1: Block Header Fields

5.2.2 Block Body

The body contains:

- Ordinary user transactions.
- The L2 reward settlement transaction (if produced this round).

5.3 Transaction Model

XPrivFi uses a hybrid model:

- accounts for balances,
- transaction-level Merkle inclusion,
- deterministic nonce ordering to prevent replay.

5.3.1 Transaction Format

Field	Description
From	Sender public key or address.
To	Recipient address.
Amount	Amount in `XPF`.
Nonce	Incrementing integer per-account.
Fee	Transaction fee (0 during initial phase).
Signature	Ed25519 signature.

Table 2: Transaction Structure

5.4 Signature Scheme

XPrivFi uses Ed25519 signatures:

$$\mathrm{sig} = \mathrm{Ed25519}(\mathrm{sk},\ \mathrm{txhash})$$

5.5 Transaction Hash

$$\mathrm{txhash} = H(\text{serialized transaction})$$

5.6 Mempool Rules

A node accepts a transaction if:

- Signature is valid.
- Nonce equals account's next required nonce.
- Sender has sufficient balance.
- Transaction size is within limits.

Transactions are ordered by:

1. nonce,
2. timestamp,
3. (future) fee.

5.7 Block Validation

Given block B, node verifies:

1. **Header Validation**
   - PreviousHash matches known chain tip.
   - Timestamp is not in future.
   - Hash meets difficulty target.
2. **Body Validation**
   - All transactions are valid.
   - No double spending.
   - Nonces increase strictly.
   - Reward settlement transaction matches L2 hash.
3. **Merkle Root**
   - Recompute MerkleRoot from body.
   - Must match header field.

5.8 Networking Protocol

XPrivFi uses a simple P2P gossip network.

5.8.1 Message Types

- `NEW_BLOCK`
- `TX_BROADCAST`
- `REQUEST_BLOCK`
- `REQUEST_TX`
- `PEER_HELLO`

Nodes maintain a peer list and use epidemic broadcasting.

5.8.2 Block Propagation

Nodes broadcast new blocks immediately upon validation.

5.8.3 Transaction Propagation

Transactions are propagated with:

$$\text{Random Delay} \in [0, 120 \text{ ms}]$$

to prevent relay-pattern fingerprinting.

5.9 Difficulty Adjustment: CP-Diff v1

5.9.1 Goal

Maintain approximately:

$$T_{\text{block}} \approx 6 \text{ minutes.}$$

5.9.2 Formula

Let:

- D_t — current difficulty target,
- T_{actual} — actual block time,
- $T_{\text{target}} = 360$ seconds.

Adjustment:

$$D_{t+1} = D_t \cdot \left(1 + \alpha \cdot \frac{T_{\text{actual}} - T_{\text{target}}}{T_{\text{target}}}\right)$$

Where:

$$\alpha = 0.12$$

Clamped between:

$$D_{t+1} \in [D_t \cdot 0.75,\ D_t \cdot 1.25]\,.$$

5.10 Retarget Window

Difficulty is recomputed every block. This reduces temporal variance while keeping consensus stable.

5.11 State Machine

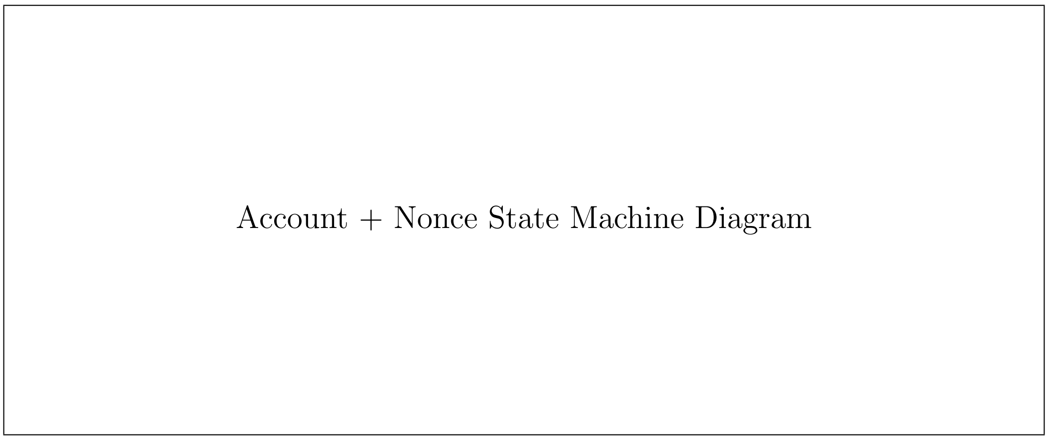

Figure 3: Account + Nonce State Machine Diagram

State transitions:

- **INIT** → account created
- **BALANCE_UPDATE** → after valid transaction
- **NONCE_INCREMENT** → after each broadcasted transaction

5.12 Block Reward Handling

XPrivFi does *not* mint block rewards at L1. Instead, L2 produces the reward transaction.

5.12.1 Reward Transaction Format

Field	Description
From	`DexPoolFund (Pool Address)`
To	Winning miner's address
Amount	1 `XPF`
RoundID	Deterministic round index
Proof	HexGrid winner proof

Table 3: Reward Settlement Transaction

L1 must validate:

1. pool has remaining balance;
2. proof matches L2 commit hash in block header;
3. amount always equals 1 XPF.

6 Layer-2: HexGrid Mining Engine

HexGrid is the deterministic off-chain execution environment responsible for XPrivFi's round-based mining. It is not a rollup or a smart contract engine. It is a specialized computation layer with strict inputs, outputs, and timing rules.

L2 executes the computation-heavy, fairness-oriented mining cycle, while L1 validates only the final reward.

6.1 Core Responsibilities

HexGrid performs:

- round coordination (timing, phases, participation window),
- miner workload enforcement,
- deterministic path selection,
- randomness extraction,
- winner commitment hashing,
- reward transaction generation.

It is fully deterministic:

$$\text{Identical round inputs} \rightarrow \text{Identical outputs across all honest nodes.}$$

6.2 Round Structure

Each round R_i consists of three phases:

6.2.1 1. Work Phase (6 minutes)

Miners compute the full deterministic workload defined in Section 6.3.

All miners execute:

$$W_i = \mathsf{ComputeRoundWork}(R_i,\ \text{address},\ \text{seed},\ \text{params})$$

This must be completed independently by every participant.

6.2.2 2. Transparency Phase (30 seconds)

HexGrid broadcasts:

- Loop iteration count
- Round seed
- Local winner hash candidates
- Participation count

Nodes verify consistency.

6.2.3 3. Lobby Phase (30 seconds)

Participants register for the next round:

$$\text{Register}(address)$$

Registrations produce:

$$C_i = \text{Commitment}(address,\ \text{timestamp})$$

6.3 Round Duration

Total:

$$T_{\text{round}} = 6 \text{ minutes} + 30\text{s} + 30\text{s} =\approx 7 \text{ minutes}.$$

6.4 Participation Rules

Each round accepts:

$$200 \leq n \leq 890$$

where n is the number of active miners.

6.4.1 Identity Constraint

Each identity (address) must:

- submit a valid lobby commitment,
- have a valid public key,
- perform the full workload.

6.4.2 Sybil Ineffectiveness

Creating k identities requires:

$$k \cdot W_i$$

work.

There is no shortcut.

6.5 Workload Definition

The workload is deterministic and identical for all participants.

Let:

- R_i — round index,
- A — miner address,
- S_i — round seed,
- M — workload multiplier,
- H — RandomHash.

6.5.1 Phase 1: Initialization

$$X_0 = H(R_i \parallel A \parallel S_i)$$

6.5.2 Phase 2: Sequential Computation Loop

For $j = 1 \ldots M$:

$$X_j = H(X_{j-1} \oplus j)$$

6.5.3 Phase 3: Final Output

$$W_i(A) = X_M$$

This output becomes the miner's *score*.

6.6 Winner Selection

Each miner submits:

$$\text{Score}_i(A) = W_i(A)$$

HexGrid selects:

$$A^\star = \arg\min_A \; W_i(A)$$

Tie-breaking rule:

$$A^\star = \min(\text{AddressLexicographical})$$

The selected winner is deterministic once all scores are known.

6.6.1 Winner Proof

The winner must submit:

$$\text{Proof} = (A^\star,\; W_i(A^\star),\; \text{loop outputs})$$

HexGrid commits:

$$C_i = H(\text{Proof})$$

which is embedded in the L1 block header.

6.7 Randomness Model

6.7.1 Round Seed

Seed generation:

$$S_{i+1} = H(B_i \parallel C_i)$$

where:

- B_i — L1 block hash,
- C_i — HexGrid commitment.

Thus, seeds are unpredictable but verifiable.

6.7.2 Entropy Sources

- previous block hash,
- winner proof,
- number of participants.

6.8 Settlement to L1

HexGrid generates the reward transaction:

$$\mathsf{RewardTx}(A^\star) = (From:\ \text{Pool},\ To: A^\star,\ Amount = 1\ \texttt{XPF})$$

Then broadcasts:

- RewardTx
- HexGrid commit hash C_i
- Round index R_i

L1 validates:

1. balance available in pool,
2. amount equals 1 XPF,
3. proof matches commitment.

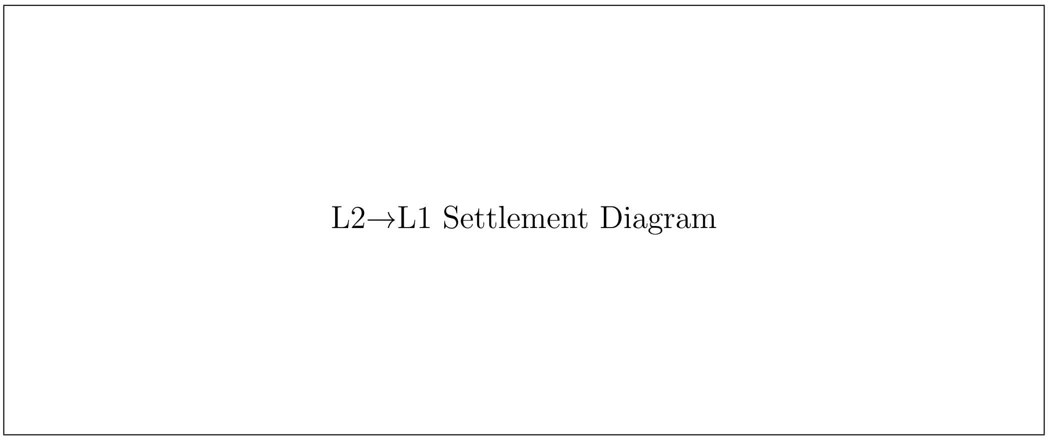

Figure 4: L2→L1 Settlement Diagram

7 Fairness Constraints

7.1 No Shortcut Rule

Sequential computation loop prevents:

- parallel skipping,
- caching shortcuts,
- GPU-style batched vectorization.

7.2 Hardware Range Normalization

The workload is designed such that:

$$\frac{\text{fastest CPU}}{\text{slowest mobile device}} \approx \text{bounded factor.}$$

Phones can participate meaningfully.

7.3 Identity Neutrality

Number of identities does not increase chance without proportional work.

8 Layer-2 Threat Model

8.1 Sybil Attacks

Creating k identities costs k workloads. Thus, Sybil is ineffective.

8.2 Hardware Dominance

Adversaries with unrealistic hardware may reduce relative fairness, but do not gain infinite advantage or the ability to skip work.

8.3 Commitment Withholding

If a winner withholds proof:

$$\text{Round is invalidated and repeated.}$$

8.4 Timing Attacks

Round phases enforce fixed timings to reduce early-disclosure attacks.

8.5 Equivocation

All honest nodes recompute:

$$C_i = H(\text{Proof})$$

and compare.

9 CP-Shield (Future Privacy Layer)

CP-Shield is a research-stage privacy extension that introduces:

- ZK-protected transfers,
- note commitments,
- shielded balances,
- optional private mode.

9.1 Shielded Notes

$$\text{Note} = H(\text{pk},\ v,\ r)$$

9.2 Nullifiers

$$\text{Nullifier} = H(\text{sk},\ \text{Note})$$

9.3 ZK Circuit (Conceptual)

- proves note ownership,
- proves amount conservation,
- hides sender and receiver.

CP-Shield is *not* activated at mainnet launch.

10 RootBaseLayer Interoperability (Vision)

RootBaseLayer (RBL) is a separate experimental settlement and exchange model conceived by the creator.

10.1 Non-binding Outlook

XPrivFi does not depend on RBL. No integration is promised.

Possible future interoperability research includes:

- atomic swaps,
- shared entropy sources,
- L2 settlement parallels.

This remains conceptual.

11 Legal and Security Disclaimers

11.1 Informational Status

This technical specification is provided solely for:

- developers,
- researchers,
- cryptographers,
- security auditors,
- early testers.

It does *not* represent:

- financial advice,
- investment solicitation,
- securities offering,
- a guarantee of future performance,
- a promise of software behavior under all circumstances.

11.2 No Guarantees

XPrivFi is an experimental protocol. All specifications may evolve based on:

- security research,
- testnet feedback,
- real-world constraints,
- implementation optimizations.

11.3 No Liability

The creator and contributors assume no responsibility for:

- loss of funds,

- network instability,
- chain reorganizations,
- cryptographic failures,
- implementation bugs,
- third-party software defects.

11.4 License Boundary

This specification does not override:

- software licensing terms,
- branding restrictions,
- protocol naming rights.

Nothing in this document grants rights to:

- use the "xPrivFi" name,
- claim affiliation,
- market derivative products,
- republish portions without attribution.

11.5 No Trademark Claims

"XPrivFi" and associated terms are project identifiers, not registered trademarks. Nothing in this document implies trademark registration.

12 Security Considerations

12.1 L1 Considerations

The L1 blockchain inherits standard PoW security properties:

- susceptibility to 51% attacks,
- block reorganization threats,
- timestamp manipulation within bounds,
- network partitioning risks.

12.2 L2 Considerations

HexGrid introduces:

- deterministic fairness constraints,
- sequential workload enforcement,
- anti-shortcut computational loops.

Potential attack vectors:

1. **Workload falsification** — mitigated via deterministic recomputation.
2. **Commit withholding** — mitigated via round invalidation.
3. **High-end hardware dominance** — mitigated but not eliminated.
4. **Seed manipulation** — mitigated via L1 anchoring.

12.3 Sybil vs Hardware Attacks

Sybil attacks are ineffective because:

$$k \text{ identities } \Rightarrow k \text{ full workloads}$$

Hardware dominance is possible but bounded. No system can guarantee equality under extreme hardware asymmetry.

12.4 Privacy Risks

CP-Shield is not active. Until implementation:

- all balances are public,
- all transfers are linkable,
- miner identities are visible.

13 Future Work

13.1 Protocol Extensions

- Full CP-Shield privacy layer,
- Optimized RandomHash variants,
- Mobile-optimized verification modes,
- Adaptive round difficulty,
- Optional transaction fee markets (post-listing).

13.2 Developer Platform

- SDK for wallet integration,
- Reference explorer API,
- Tools for verifying HexGrid proofs,
- L1 node RPC improvements.

13.3 RootBaseLayer Research

While independent from XPrivFi, RootBaseLayer (RBL) offers conceptual future opportunities:

- parallel settlement bridges,
- shared randomness sources,
- experimental fee-less exchange mechanisms.

No integration is guaranteed or implied.

14 Extended Glossary

Account Model — Balance-based ledger entries with nonces.

Block Reorganization — A chain fork replacing recent blocks.

Commit Hash — The L2 hash included in L1 block headers to prove round finalization.

Deterministic Mining — Mining where reward outcomes are calculable from deterministic workloads.

Lobby Phase — A time window for miners to commit to the next round.

Transparency Phase — A window where round computations are revealed.

RandomHash — CPU-oriented proof of work used by L1.

Mining Pool Address — Reserved address supplying 500,000 XPF for L2 rewards.

Seed — Deterministic value that initializes each round.

Winner Proof — Verifiable proof that the selected miner completed the required work.

ZK Proof — A cryptographic proof allowing verification without revealing information.

15 Appendices

15.1 Appendix A: Block Diagram

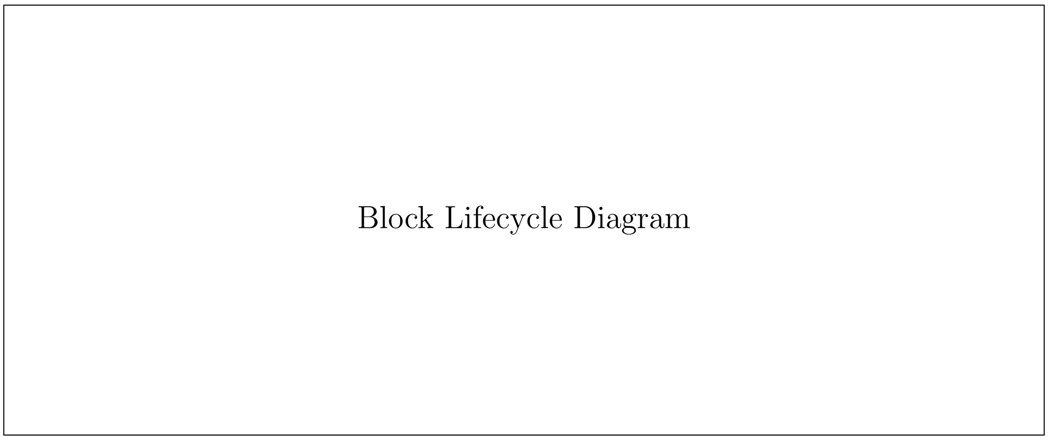

Figure 5: Block Lifecycle Diagram

15.2 Appendix B: Round Diagram

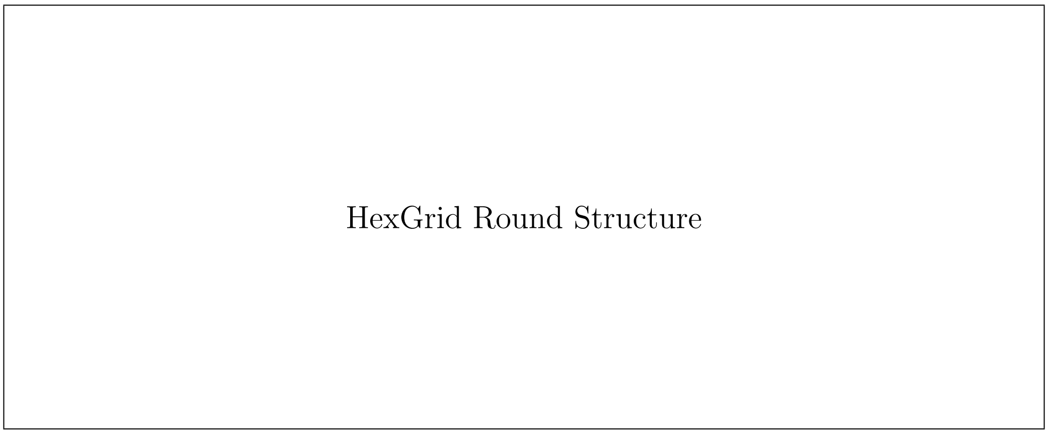

Figure 6: HexGrid Round Structure

15.3 Appendix C: Proof Structure

Winner Proof Structure Diagram

Figure 7: Winner Proof Structure Diagram

15.4 Appendix D: Network Topology

Peer-to-Peer Topology

Figure 8: Peer-to-Peer Topology

15.5 Appendix E: Possible CP-Shield Circuits

ZK Circuit Placeholder

Figure 9: ZK Circuit Placeholder

16 References

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[2] A. Researcher, "RandomHash: A Memory-Hard, CPU-Oriented Hashing Algorithm," 2019.

[3] E. Ben-Sasson et al., "Zerocash: Decentralized Anonymous Payments," IEEE Symposium on Security and Privacy, 2014.

[4] M. Rosenfeld, "Analysis of Hashrate-Based Double-Spending," 2014.

[5] D. J. Bernstein et al., "Ed25519: High-speed high-security signatures," 2011.

17 Document Versioning

- **Version:** 1.0
- **Date:** November 2025
- **Status:** Draft (Technical Specification)
- **Scope:** L1+L2 core rules, mining model, security, future modules

17.1 Change Log

- v1.0 — Initial full specification suite.