QuantumHarmony Light Paper

Version 1.2 — January 2025

QuantumVerse Protocols

Changelog v1.2: Added MEV protection documentation. Corrected finality description—QuantumHarmony provides deterministic BFT finality via the Coherence Gadget, not probabilistic finality.

Abstract

QuantumHarmony is a Layer 1 blockchain built on Substrate that replaces quantum-vulnerable cryptographic components with post-quantum alternatives. This document describes what the system does, how it works, and its current state.

1 Problem Statement

1.1 Quantum Computing Threat

Current blockchains rely on cryptographic primitives that quantum computers can break:

Primitive	Algorithm	Quantum Attack
Signatures	ECDSA, Ed25519	Shor's Algorithm
Finality	BLS (GRANDPA)	Shor's Algorithm
Hashing	Blake2b	Grover's Algorithm

Fact: NIST estimates cryptographically relevant quantum computers could exist within 10–15 years. Blockchain addresses and signed transactions recorded today become vulnerable once such computers exist.

1.2 What QuantumHarmony Changes

Component	Standard Substrate	QuantumHarmony
Signatures	Ed25519 / ECDSA	SPHINCS+ (NIST PQC)
Block Hashing	Blake2b	Keccak-256 (SHA-3)
Finality Gadget	GRANDPA (BLS)	Coherence Gadget (Falcon1024)
Randomness	VRF	Quantum-enhanced VRF (optional QKD)

2 Technical Implementation

2.1 SPHINCS+ Signatures

SPHINCS+ is a stateless hash-based signature scheme standardized by NIST in 2024. Its security relies solely on hash function properties, not discrete logarithms or elliptic curves.

Trade-offs:

- Signature size: approximately 8–50 KB
- Slower signing compared to Ed25519
- Verification time comparable to classical schemes

Implementation is provided via `pallet-sphincs-keystore`.

2.2 Keccak-256 Hashing

Keccak-256 (SHA-3) replaces Blake2 throughout the runtime.

- 256-bit output provides 128-bit post-Grover security
- 1600-bit sponge state
- Standardized and widely audited

2.3 Consensus and Finality

Block production: Aura (Authority Round).

Finality: Deterministic BFT finality via the **Coherence Gadget**, a post-quantum replacement for GRANDPA.

2.4 Coherence Gadget

The Coherence Gadget provides GRANDPA-equivalent deterministic finality:

GRANDPA	**Coherence Gadget**
BLS signatures	Falcon1024 signatures
Prevote / Precommit	STARK verification + coherence scoring
2/3 supermajority	2/3 supermajority
Finality proof	Finality Certificate

Protocol flow:

1. New block produced by Aura
2. Proof collection (entropy / coherence inputs)
3. Proof verification and scoring

4. Falcon1024 signing
5. Vote broadcast (encrypted)
6. Supermajority aggregation
7. Finality certificate generation

2.5 Proof of Coherence (PoC)

With quantum hardware:

- QRNG / QKD entropy sources (Toshiba, Crypto4A, IdQuantique)
- STARK proofs verified with Winterfell
- QBER-based coherence scoring (threshold: 11%)

Without hardware (fallback):

- Mock entropy sources
- Full BFT execution preserved
- Falcon1024 signatures
- Deterministic finality still guaranteed

Quantum hardware improves security guarantees but is not required for correctness or finality.

2.6 MEV Protection

QuantumHarmony provides native Maximal Extractable Value (MEV) protection at the protocol level.

The problem: In traditional blockchains, validators can reorder transactions (frontrunning), insert their own transactions (sandwich attacks), or censor specific transactions.

Solution:

1. Leader elected via quantum-seeded VRF (unpredictable)
2. Leader maintains qVRF-ordered priority queue
3. Leader compares priority queue against public mempool
4. Discrepant transactions are deleted

Reporter requirements: Every report must include a randomly generated nonce:

$$\texttt{tx\_hash} = \texttt{Hash}(\texttt{payload} \| \texttt{random\_nonce})$$

This ensures unique transaction hashes, prevents replay attacks, and enforces deterministic ordering.

Attack	Mitigation
Frontrunning	qVRF ordering is unpredictable
Sandwich attacks	Discrepancy detection removes injected txs
Transaction censorship	Leader rotation
Replay attacks	Random nonce per report

3 Governance System

QuantumHarmony includes standard Substrate governance pallets:

- Democracy
- Collective
- Treasury
- Scheduler

3.1 Academic Vouching

Allows verified academics to vouch for applicants using on-chain voting and thresholds.

3.2 Ricardian Contracts

Human-readable legal contracts stored on-chain with lifecycle management and signatures.

3.3 Notarial Services

Document timestamping and attestation with revocation support.

4 Current State

Testnet: Operational (3 validators)
Block time: 6 seconds
Consensus: Aura + Coherence Gadget

What works:

- Block production with Aura + SPHINCS+
- Deterministic BFT finality via Coherence Gadget
- All governance and legal pallets
- STARK proof verification path
- Docker deployment

In progress:

- Production QKD hardware integration
- Multi-region validator expansion

Not done:

- Security audit
- Mainnet launch

5 Limitations

- Large post-quantum signatures (29 KB for SPHINCS+, 1.3 KB for Falcon1024)
- No BLS-style signature aggregation
- Non-standard Substrate tooling compatibility

6 Comparison

	QuantumHarmony	**Substrate**	**QRL**
Signatures	SPHINCS+ / Falcon	Ed25519 / BLS	XMSS
Finality	Deterministic BFT	GRANDPA (BFT)	PoW
MEV Protection	Native (qVRF)	No	No
Quantum HW	Optional	No	No

7 References

1. NIST Post-Quantum Cryptography Standardization (2024)
2. NIST FIPS 205: SPHINCS+
3. NIST FIPS 206: Falcon
4. Substrate Developer Documentation
5. Grover, L. "A Fast Quantum Mechanical Algorithm for Database Search" (1996)
6. Shor, P. "Algorithms for Quantum Computation" (1994)

Contact

Project: QuantumHarmony (QuantumVerse Protocols)
Technical Lead: Sylvain Cormier
Repository: `https://github.com/QuantumVerseProtocols/quantumharmony`

This document describes the system as implemented. No forward-looking claims are made.