A Robust, Decentralized Architecture for Quantum Randomness Generation Leveraging the Kirk Network, PQ-VRFs, and Threshold Cryptography

Paraxiom

December 12, 2024

Abstract

Quantum Random Number Generators (QRNGs) produce high-quality randomness derived from quantum phenomena, offering strong security assurances. However, relying on a single QRNG device centralizes trust and creates a single point of failure. In this paper, we present a decentralized architecture that integrates multiple geographically distributed QRNG nodes using the Kirk network's quantum and classical channels. We incorporate Post-Quantum Verifiable Random Functions (PQ-VRFs), threshold secret sharing, and Byzantine Fault Tolerant (BFT) consensus protocols to ensure unbiased, verifiable, and tamper-resistant randomness. We address previously overlooked issues, including security assumptions, QRNG independence, PQ-VRF implementation, timing constraints, QKD integration details, scalability, error handling, entropy monitoring, and side-channel considerations. We also introduce continuous behavioral monitoring, adaptive key refresh policies, and fallback mechanisms.

Finally, we detail a blockchain oracle system that transforms quantum-secure keys into various blockchain-compatible formats (e.g., Ethereum, Bitcoin, Polkadot), creat-

ing a future-proof, monetizable service. This integrated approach provides a holistic, robust solution for quantum-era blockchain infrastructures, federated learning policies, and quantum-secure consensus mechanisms.

1 Introduction

Quantum Random Number Generators (QRNGs) derive randomness from inherently unpredictable quantum events. This can significantly enhance cryptographic robustness, especially as quantum computing challenges classical assumptions. However, a single QRNG source centralizes trust and can become a vulnerability if compromised.

To eliminate single points of failure, we propose a decentralized architecture: multiple QRNG devices distributed across different physical locations (*noeuds* and *pôles*) connected by the Kirk network. The Kirk network provides QKD-based authentication and secure classical channels. This setup enables distributed randomness generation, where no single node can bias the final output.

We integrate PQ-VRFs for verifiable correctness, threshold cryptography to ensure at least $(n/2 + 1)$ honest nodes are needed to reconstruct randomness, BFT protocols to handle malicious participants, continuous behavioral monitoring, and adaptive QKD refresh policies. Additionally, we cover scalability, error handling, entropy assessment, side-channel protection, and conclude with a blockchain oracle that adapts quantum-safe keys for multiple blockchain formats.

2 Background

2.1 QRNG Centralization Problem

A single QRNG device is a potential single point of compromise. By employing multiple QRNG sources, each independently verified and monitored, and combining outputs securely,

we distribute trust and prevent one compromised device from controlling the system.

2.2 Kirk Network and Quantum Infrastructure

The Kirk network supports quantum key distribution (QKD) and secure classical channels, enabling:

- Reliable authentication using QKD-generated keys.
- Geographic and administrative diversity by leveraging multiple *pôles* and *noeuds*.
- High-bandwidth, low-latency classical connections for coordination and proof exchange.

2.3 PQ-VRF and Threshold Cryptography

Post-Quantum VRFs ensure that randomness contributions are verified and not tampered with. Threshold schemes, like Shamir Secret Sharing, require a majority of honest nodes to reconstruct the final randomness, preventing small colluding groups from biasing results.

3 Revised Architecture and Solutions

3.1 Security Assumptions and Node Compromise

We define a clear threshold: $(n/2+1)$ honest nodes are required for security. BFT protocols (e.g., PBFT) ensure consensus even if up to half the nodes are malicious. Continuous behavioral monitoring—statistical tests, correlation checks, and timing analysis—flags suspicious nodes. Persistent anomalies lead to reduced trust or temporary node exclusion.

3.2 QRNG Source Independence

Each node uses a distinct QRNG vendor or quantum phenomenon, audited by independent labs. Regular statistical testing (NIST SP 800-90B, Dieharder) checks entropy quality. This

diversity prevents common-mode failures and reduces the chance that multiple nodes can be identically compromised.

3.3 PQ-VRF Implementation Details

We select PQ-VRFs from post-quantum candidates endorsed by NIST. These VRFs have:

- Security proofs linking their hardness to well-studied assumptions.
- Performance benchmarks ensuring scalability.
- Formal composition arguments confirming that mixing quantum randomness with PQ-VRF outputs maintains overall security.

3.4 Timing and Synchronization

The protocol tolerates asynchrony:

- Nodes contribute randomness when available.
- Timeouts handle offline nodes; rounds proceed with remaining participants.
- Fallback modes ensure progress if too many nodes are offline, possibly at a reduced threshold.

3.5 Threshold Schemes and Secret Sharing

We choose $(n/2 + 1)$-threshold secret sharing for strong resilience. Proactive secret sharing periodically refreshes shares to thwart adaptive adversaries. XOR-based methods are simpler but less robust; Shamir's secret sharing is preferred for stronger security.

3.6 QKD Integration Details

To enhance QKD integration:

- **Adaptive Key Refresh:** Adjust key lifetimes based on threat conditions or entropy quality.
- **Multi-Path QKD Links:** Employ multiple QKD links or MDI-QKD to reduce trust in any single device.
- **Key Pooling:** Maintain pre-distributed key buffers for short-term QKD outages.

QKD performance dashboards, automated alerts, and interoperability tests with Toshiba and ID Quantique ensure transparency and reliability.

3.7 Scalability Considerations

For large-scale deployments:

- Hierarchical architectures let clusters handle local verification before aggregating globally.
- Caching PQ-VRF proofs on a distributed ledger reduces communication overhead.
- Parallelization of VRF checks and secret sharing maintains throughput.

3.8 Error Handling and Recovery

Defined procedures cover:

- **QRNG Failures:** Quarantine faulty nodes; continue with remaining honest nodes.
- **Node Replacement:** New nodes undergo stricter initial checks. Compromised nodes are gradually removed.
- **Partial Outages:** Store keys and rely on fallback KEM if QKD fails, ensuring uninterrupted operation.

3.9 Entropy Assessment and Side-Channels

Continuous entropy checks and alarms detect declining quality. Third-party audits enhance trust. Physical security, constant-time implementations, and environmental monitoring mitigate side-channels.

4 Blockchain Integration and Key Translation

The blockchain oracle system adapts quantum-secure keys for various ecosystems:

- Ethereum/Bitcoin (secp256k1)
- Polkadot (SR25519)
- Ed25519 and post-quantum formats (e.g., SPHINCS+)

As PQC adoption grows, the oracle can supply PQ-native keys, future-proofing against quantum attacks. Governance policies in smart contracts dictate rotation, revocation, and usage. This service provides a premium, monetizable function—secure key material for multiple chains.

5 Validation, Strengths, and QKD Enhancements

A validation analysis confirms strengths in the security model, implementation details, and practical considerations. To further improve QKD integration:

- Dynamically adjust QKD key refresh rates.
- Use multi-path QKD links and device-independent protocols.
- Deploy QKD performance dashboards and alerts.

These refinements complement the existing model, enhancing trust, adaptability, and resilience as quantum networks mature.

6 Technical Implementation Specifications

Below are concrete performance, hardware, security, and blockchain integration parameters.

6.1 Performance Requirements

QKD Key Generation

- Minimum Rates: 1000 secure bits/second per QKD link
- Key Storage Buffer: 10GB per node
- Latency: Max 100ms for key retrieval

QRNG Output

- Raw Entropy: Min 1Gbps generation
- Response Time: Max 1ms for randomness requests
- Uptime: 99.999

Network Parameters

- Inter-pôle Latency: Max 10ms
- Intra-pôle Latency: Max 2ms
- Verification Time: Max 50ms

6.2 Hardware Specifications

QRNG Requirements

- Entropy Quality: Min 0.98 bits/bit
- Diverse Quantum Sources per Node
- Self-Test and Health Monitoring: Required

Server Requirements

- CPU: 64 cores @ 3.2GHz
- RAM: 256GB
- Storage: 2TB NVMe
- Network: 40Gbps

6.3 Security Thresholds

Statistical Tests

- P-value threshold: 0.01
- Sample size: 1M bits
- Test frequency: Every 10 minutes

Monitoring Thresholds

- Latency Alert: >50
- Entropy Drop: >2
- Cross-Correlation: >0.1
- Failed Verifications: >0.01

6.4 Blockchain Oracle Parameters

Key Generation Formats

- secp256k1 (Ethereum/Bitcoin)
- ed25519, SR25519 (Substrate-based chains)
- SPHINCS+ (Post-quantum)

Performance Metrics

- Key Generation Time: Max 100ms
- Throughput: 1000 keys/second
- Verification Latency: Max 200ms

7 Conclusion

Our integrated solution addresses previously identified gaps and introduces robust measures for decentralized quantum randomness generation. By using multiple QRNG sources, PQ-VRFs, BFT consensus, threshold cryptography, continuous monitoring, and adaptive QKD integration, we ensure a secure, scalable, and fault-tolerant system.

This platform not only strengthens security against quantum threats but also creates economic opportunities by providing quantum-secure keys and randomness to multiple blockchain ecosystems, offering a premium service for the post-quantum era.