Cryzene: Autovectorized, Tensor-Driven Byzantine Consensus with Post-Quantum Privacy Guarantees under Minimal Synchrony

Kritagya Jha
Secure Distributed Systems Lab, India

Presented: July 2025

1 Introduction

The evolution of distributed systems, zero-trust infrastructure, and post-quantum cryptography has made consensus protocols a foundational yet vulnerable pillar. Traditional Byzantine Fault Tolerant (BFT) protocols like PBFT and Tendermint assume synchrony and trusted setups, which collapse under adversarial or quantum-aware settings.

Cryzene is introduced as a software-native, cryptographic consensus framework designed to operate under conditions of minimal synchrony, Byzantine adversaries, and quantum-level threats. It bridges the cryptographic rigidity of hybrid post-quantum primitives with the computational softness of software-only environments.

2 System Design Overview

Cryzene integrates five key technical domains:

1. **Hybrid Post-Quantum Cryptography:** Kyber for secure key encapsulation (based on Module-LWE), and SQIsign for isogeny-based, compact digital signatures.

2. **Tensorized Computation:** Polynomial and ciphertext structures are encoded as tensors aligned with SIMD memory layouts to enable batch cryptographic execution.

3. **Autovectorizing Compiler:** A compiler pass transforms encrypted tensor operations, minimizing rotations, cache thrashing, and memory overhead using ApplyRoll-like semantics.

4. **Doubly-Efficient PIR:** Inspired by SimplePIR, Cryzene implements matrix-vector accelerated Private Information Retrieval with sublinear server computation.

5. **Minimal Synchrony Byzantine Agreement:** A protocol with $O(f)$ rounds and $O(f^2)$ messages, tolerant to Byzantine faults, assuming only a single synchronous communication path.

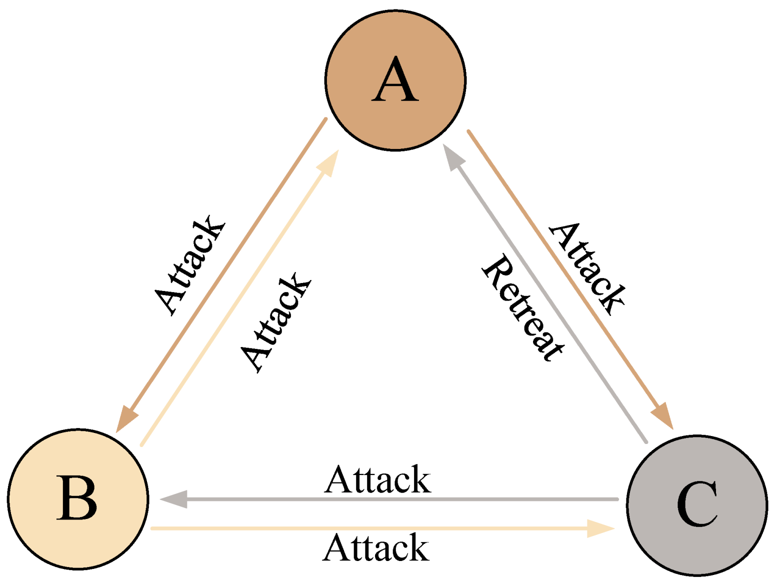

Figure 1: Illustration of the classic Byzantine Generals Problem: Inconsistent message propagation between nodes leads to conflicting decisions (e.g., "Attack" vs "Retreat") — motivating the need for Byzantine Agreement under adversarial fault conditions.

3 Protocol Execution Flow

Phase 1: Secure Channel Initialization — Each node uses Kyber encapsulation to generate session keys. Messages are signed using SQIsign.

Phase 2: PIR Query Issuance — Clients issue LWE-based PIR queries to an encrypted server-side dataset (e.g., environmental sensor logs, configuration metadata).

Phase 3: Homomorphic Processing — The server responds with HE-encrypted values which are tensor-aligned and transformed using the autovectorizing compiler to reduce evaluation depth.

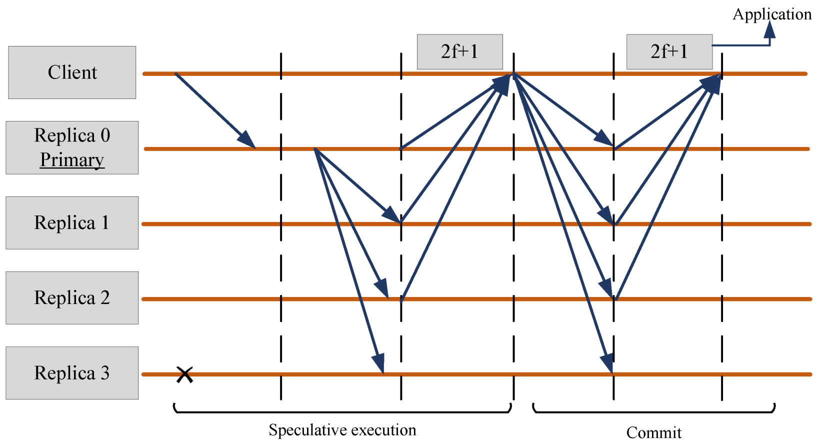

Figure 2: Autovectorized homomorphic execution flow: Cryzene's compiler restructures encrypted tensor operations to minimize ciphertext rotations and enable SIMD-aligned speculative execution followed by commit phases across Byzantine-replicated nodes.

Phase 4: Consensus Rounds — Using message digests and signed encrypted messages, nodes participate in Byzantine Agreement under partial synchrony. Output ciphertexts are verified and recorded.

4 Technical Challenges and Innovations

Autovectorization in Encrypted Tensor Programs

The HE engine uses a domain-specific intermediate representation (IR) to abstract over CKKS/BFV operations. Loop unrolling, tiling, and rotation minimization strategies (e.g., ApplyRoll) are applied. SIMD-aware packing is performed to align with tensor layouts.

Cryptographic Integration

Kyber's NTT-friendly polynomial layout is reused for homomorphic multiplication pipelines. This avoids redundant memory reshuffling between PQC and HE phases.

Minimal Synchrony Model

Cryzene tolerates asynchronous links across the network, requiring only one minimal synchronous path to guarantee safety and liveness. Fault-tolerant progress is ensured under partial network partitions.

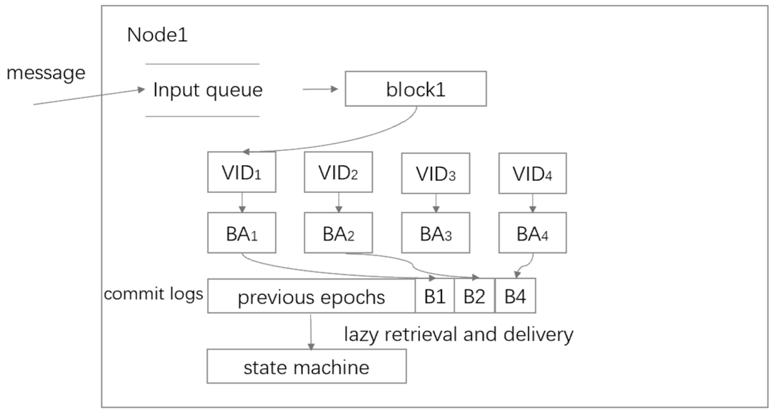

Figure 3: Cryzene architectural overview: A layered cryptographic flow combining Kyber-encrypted client input, LWE-based PIR, homomorphic tensor computation, and fault-tolerant consensus through a minimal synchrony Byzantine Agreement protocol.

5 Applications

- **Encrypted Collective Voting:** Voter data is queried via PIR, aggregated homomorphically, and consensus reached under post-quantum security guarantees.
- **Federated Analytics:** Enables secure multi-party analytics on encrypted data without revealing query patterns or intermediate states.
- **Quantum-Safe Blockchain Layer:** Cryzene can act as a consensus sublayer in ZK-enabled or quantum-hardened blockchain systems.

6 Conclusion

Cryzene is not just a cryptographic toolkit but a cohesive protocol stack for consensus in adversarial, asynchronous, and quantum-capable environments. It stands at the intersection of theoretical cryptography, compiler optimizations, and distributed system pragmatism — ready to simulate, extend, or deploy as a cryptographic backbone in post-quantum digital infrastructure.