Ứng dụng Zero Knowledge Proofs và thuật toán hậu lượng tử(Post Quantum Cryptography) vào mạng Blockchain liên ngân hàng

GIẢNG VIÊN HƯỚNG DẪN: TS.NGUYỄN NGỌC TỰ
SINH VIÊN: NGUYỄN HOÀNG QUÝ - 24521494
SINH VIÊN: HUỲNH NHẬT DUY - 24520375

Ngày 3 tháng 10 năm 2025

Mục lục

1 Introduction

Blockchain đã trở thành một nền tảng quan trọng trong lĩnh vực tài chính, đặc biệt đối với các hệ thống ngân hàng và liên ngân hàng, nhờ vào các đặc tính minh bạch, toàn vẹn và phi tập trung. Tuy nhiên, sự phát triển của [1]máy tính lượng tử đang đe dọa phá vỡ các thuật toán mật mã hiện tại như RSA và ECDSA, từ đó đặt ra nhu cầu áp dụng các thuật toán mật mã hậu lượng tử (PQC) nhằm đảm bảo an toàn cho các giao dịch.

Bên cạnh đó, các thủ tục ngân hàng truyền thống, chẳng hạn như rút tiền hay sao kê, vẫn còn khá phức tạp. Ví dụ, để thực hiện giao dịch rút tiền, khách hàng thường phải cung cấp các giấy tờ tùy thân như CCCD hoặc hộ chiếu, làm gia tăng nguy cơ rò rỉ thông tin cá nhân.

Do đó, xuất hiện nhu cầu phát triển một giải pháp vừa đảm bảo khả năng kháng lượng tử, duy trì an toàn cho các giao dịch, vừa tăng cường tính riêng tư (privacy), qua đó củng cố niềm tin của khách hàng trong việc sử dụng các dịch vụ ngân hàng và liên ngân hàng.

2 Project Objective

2.1 Nhu cầu

Khách hàng và các tổ chức liên ngân hàng ngày càng yêu cầu bảo vệ thông tin nhạy cảm, muốn thực hiện giao dịch một cách riêng tư, đồng thời duy trì tính minh bạch và toàn vẹn của hệ thống.[2]

2.2 Vấn đề

Các phương pháp truyền thống chưa đáp ứng đồng thời được tính riêng tư và khả năng kháng lượng tử, đặc biệt trong các giao dịch liên ngân hàng, nơi số lượng validator ít và chi phí bảo mật có thể cao hơn.

2.3 Đề xuất giải pháp

1. [3]Kết hợp Post-Quantum Cryptography (Dilithium) và Advanced Encryption Standards (AES) hoặc chuyển sang sử dụng Post-Quantum Cryptography (Dilithium) tích hợp trong HSM để ký các giao dịch, đảm bảo an toàn trước tấn công lượng tử.

2. Áp dụng [4]Zero-Knowledge Proofs (ZSTARK) để xác minh các giao dịch mà không tiết lộ thông tin nhạy cảm, duy trì privacy cho khách hàng và bảo mật dữ liệu tổ chức.

3. Kết hợp PQC + ZKP trên mạng blockchain liên ngân hàng (cross-ledger settlement) giúp cân bằng giữa tính minh bạch, bảo mật cao và chi phí chấp nhận được cho hệ thống có số lượng validator hạn chế.

3 Weakness analysis

3.1 Kích thước chữ ký & bloat chain

Các chữ ký PQC thường lớn hơn ECC (Elliptic Curve Cryptography). Ví dụ, SPHINCS+ có thể vài KB cho một chữ ký, trong khi ECDSA chỉ vài chục byte.
Kích thước lớn dẫn tới blockchain "bloat": block nặng hơn, chi phí lưu trữ trên các node cao hơn, băng thông truyền tải tăng, đồng thời làm giảm TPS.

3.2 Xác minh & hiệu năng

Xác minh PQC signatures hoặc [5]zk-STARK proofs tốn nhiều CPU, đặc biệt với các node nhẹ.
Trên smart contract EVM, mỗi operation bị giới hạn gas, nên verify PQC hoặc STARK code trực tiếp trên chain có thể rất tốn kém và giới hạn khả năng mở rộng.

3.3 Aggregation & threshold signing khó khăn

Nhiều cơ chế multi-signature hoặc threshold signature hiện nay (ví dụ BLS) dựa trên pairing groups và không tương thích với hầu hết PQC schemes.
Để xây dựng threshold PQC (ví dụ cho consortium blockchain) cần nghiên cứu thêm, hoặc dùng MPC/hybrid solution, tăng độ phức tạp và khả năng lỗi.

3.4 Proof sizes & prover resources

zk-STARK proofs thường lớn, tốn CPU để sinh proof.
Để giảm tải, có thể cần prover pools (nhiều máy prover song song) hoặc trusted prover farms.
Tuy nhiên, nếu dùng trusted prover $\rightarrow$ giảm trust model (cần tin prover không gian lận).

3.5 Crypto-agility implementation bugs

Khi triển khai migration logic (dual-sig, key rotation, replay protection) dễ xảy ra lỗi.
Ví dụ: chấp nhận transaction với signature cũ, hoặc chữ ký mới sai lệch → có thể tạo lỗ hổng replay/phishing.
Điều này đặc biệt nguy hiểm khi phải duy trì song song nhiều thuật toán (ECDSA + PQC).

4 Methodology

Đề tài sẽ được chia làm ba track dựa vào đặc điểm của từng phần trong thiết kế.

4.1 TRACK A - PQC Signatures for transactions

Trong track A này, nhóm sẽ triển khai node và wallet có khả năng ký giao dịch bằng thuật toán Dilithium (Post-Quantum Signature).

4.1.1 Benchmark

Sử dụng Hyperledger Caliper để thực hiện benchmark, bao gồm các chỉ số chính:

- Tx size delta (bytes): so sánh kích thước giao dịch khi dùng chữ ký PQC so với chữ ký ECDSA truyền thống.
- CPU cost for sign/verify (ms): đo chi phí tính toán khi ký và xác minh chữ ký.
- Mempool throughput and TPS impact: đánh giá tác động của PQC signatures đến hiệu năng xử lý giao dịch (throughput, transactions per second).

4.2 TRACK B - ZK proof module

Trong phần này, nhóm sẽ xây dựng một **minimal rollup prover** có khả năng:

- Gom nhiều giao dịch **(batch transactions)**.
- Sinh ra **zk-STARK proof**.

Sử dụng thư viện WINTERFELL (Rust, phát triển bởi Facebook/Polygon zk research) để tạo và xác minh zk-STARK proofs.

4.2.1 Verifier (Smart Contract)

- Được triển khai dưới dạng **smart contract** trên **EVM-compatible chain** (ví dụ: Hyperledger Besu private chain).
- Chức năng chính:
  - Nhận zk-STARK proof từ prover.
  - Thực hiện xác minh proof.
  - Áp dụng **state transition** nếu proof hợp lệ.

4.2.2 Benchmark

- Proof generation time per batch: thời gian trung bình để prover sinh proof cho một batch giao dịch.
- Proof size (bytes): kích thước proof, ảnh hưởng đến chi phí truyền tải on-chain.
- Verifier cost: do sử dụng Hyperledger Besu, đo **gas cost** khi verify proof để xác định chi phí verifier.

4.3 TRACK C - Deployment and Integration

Nhóm sẽ **tích hợp Track A (PQC signatures)** và **Track B (ZK rollup module)** vào một **private consortium blockchain network** chạy bằng **Hyperledger Besu**.

4.3.1 Thành phần triển khai

- **Node/Validator**
  - Chạy consensus (IBFT2 hoặc QBFT) trên Besu.
  - Xác minh giao dịch đã được ký bằng **PQC signatures**.
- **Wallet**
  - Sinh giao dịch với chữ ký **Dilithium**.
  - Gửi giao dịch đến mạng Besu.
- **Prover & Verifier**
  - **Prover:** gom nhiều giao dịch (batch) và sinh **zk-STARK proof**.

  - **Verifier (smart contract on-chain):**
    * Nhận proof từ Prover.
    * Kiểm chứng proof.
    * Áp dụng thay đổi trạng thái **(state transition)** vào ledger nếu hợp lệ.

- **Ledger**
  - Shared ledger của consortium blockchain, nơi lưu giữ state đã cập nhật.

- **KSM (Key Simulation Module)**
  - Mô phỏng **HSM (Hardware Security Module)** để quản lý PQC keys.
  - Chức năng:
    * Sinh và lưu trữ khóa PQC an toàn.
    * Thực hiện ký số bằng PQC trong môi trường giả lập HSM.
    * Hỗ trợ test: key rotation, emergency rollback, dual-sig migration.

4.3.2 Benchmark tổng hợp cần đo lường

- **End-to-end latency:** thời gian từ khi wallet ký PQC transaction đến khi ledger cập nhật sau proof.
- **Chain throughput (TPS):** đánh giá hiệu năng của toàn hệ thống khi tích hợp PQC + zk-STARK.
- **Security evaluation:**
  - Backward compatibility.
  - Replay protection.
  - Validator consensus với PQC keys (mô phỏng qua KSM).

5 Timeline

Timeline này nhóm đã chỉnh sửa sao cho phù hợp thời gian còn lại dựa trên timeline thầy đề xuất.

5.1 Tuần 1–2: Literature review & Infrastructure

- **Track A — PQC Signatures**
  - Nghiên cứu thuật toán PQC (Dilithium).
  - Chuẩn bị module ký/verify cơ bản cho wallet.
- **Track B — ZK Rollup**
  - Nghiên cứu zk-STARK, chọn thư viện (Winterfell).
  - Chuẩn bị contract mẫu verifier trên Besu/EVM.
- **Track C — Deployment**
  - Cài đặt private consortium blockchain (Hyperledger Besu).
  - Chuẩn bị công cụ benchmark (Caliper).

5.2 Tuần 3–4: PQC & ZK Prototype

- **Track A**
  - Tích hợp PQC signing vào wallet + node.
  - Benchmark ban đầu: Tx size delta, CPU cost cho ký/xác minh.
- **Track B**
  - Xây dựng minimal prover sinh proof cho batch nhỏ.
  - Benchmark proof generation time & proof size.
- **Track C**
  - Kết nối wallet → Besu network (PQC tx).
  - Node validator xác minh PQC signatures.

5.3 Tuần 5–6: Module nâng cao & KSM

- **Track A**
  - Test backward compatibility với ECDSA.
  - Triển khai replay protection.
- **Track B**

- Hoàn thiện verifier contract (state transition).
- Đo gas cost khi verify proof.

- **Track C**
  - Phát triển KSM + Migration Layer (Dual-sig, rollback, key rotation).
  - Test: key rotation, emergency rollback.

5.4 Tuần 7–8: Stress test & Consensus

- **Track A**
  - TPS benchmark (mempool throughput với PQC).

- **Track B**
  - Stress test prover với batch lớn.
  - Benchmark prover pool scaling (song song nhiều prover).

- **Track C**
  - Validator ký block bằng PQC keys qua HSM/KSM; test consensus (IBFT/QBFT) với PQC signatures.
  - End-to-end throughput (TPS) khi chạy PQC + zk-STARK.

5.5 Tuần 9–10: Tổng hợp & Báo cáo

- **Track A + B**
  - Báo cáo benchmark riêng (PQC, zk-STARK).

- **Track C**
  - Đánh giá cơ chế rollback/migration: khả năng quay về ECDSA khi PQC lỗi
    - Latency từ wallet ký → ledger update.
    - Security evaluation: backward compatibility, replay protection, validator consensus.
  - Deliverables:
    - Repo reproducible (Docker + scripts).
    - Migration playbook.
    - Báo cáo cuối + slides + demo video.

6 Glossary

PQC (Post-Quantum Cryptography) : Thuật toán mật mã có khả năng chống lại các cuộc tấn công từ máy tính lượng tử.

Dilithium : Thuật toán chữ ký số hậu lượng tử, thuộc chuẩn CRYSTALS-PQC.

ECDSA (Elliptic Curve Digital Signature Algorithm) Thuật toán chữ ký số truyền thống dựa trên đường cong elliptic.

SPHINCS+ : Thuật toán chữ ký dựa trên hàm băm, không trạng thái, hậu lượng tử, chữ ký lớn nhưng chống lượng tử.

zk-STARK : Hệ thống chứng minh không tiết lộ dữ liệu (Zero-Knowledge Scalable Transparent ARgument of Knowledge).

BLS (Boneh–Lynn–Shacham) : Thuật toán chữ ký hỗ trợ tổng hợp chữ ký và threshold signature, dựa trên cặp đồng dạng (pairing groups).

MPC (Multi-Party Computation) : Giao thức cho phép nhiều bên cùng tính toán một hàm mà không tiết lộ dữ liệu riêng.

HSM (Hardware Security Module) : Thiết bị phần cứng bảo mật để lưu trữ khóa và thực hiện các phép toán mật mã.

KSM (Key Simulation Module) : Module phần mềm mô phỏng HSM để quản lý khóa PQC trong môi trường thử nghiệm.

TPS (Transactions Per Second) : Số lượng giao dịch blockchain có thể xử lý trong một giây.

Blockchain "bloat" : Hiện tượng blockchain tăng kích thước lớn, làm tăng yêu cầu lưu trữ và băng thông.

Gas cost : Chi phí tính toán khi thực thi các lệnh trên smart contract EVM.

Replay protection : Cơ chế chống gửi lại các giao dịch cũ.

Dual-sig / Migration Layer : Cơ chế chạy đồng thời hai thuật toán chữ ký (ví dụ ECDSA + PQC) để chuyển đổi an toàn.

Threshold signature : Chữ ký số yêu cầu nhiều bên cùng ký để xác thực.

Multi-signature : Chữ ký số yêu cầu nhiều khóa riêng để phê duyệt giao dịch.

Prover : Thực thể sinh ra chứng minh không tiết lộ (zero-knowledge proof) cho một batch giao dịch.

Verifier : Thực thể (trên chain hoặc off-chain) kiểm tra tính hợp lệ của chứng minh không tiết lộ.

Batch : Nhóm các giao dịch được xử lý cùng lúc để tối ưu hiệu năng.

State transition : Việc cập nhật trạng thái ledger của blockchain sau khi giao dịch hoặc chứng minh hợp lệ.

Crypto-agility : Khả năng chuyển đổi thuật toán hoặc khóa mật mã mà không làm gián đoạn hoạt động hệ thống.

Tài liệu

[1] V. Mavroeidis, K. Vishi, M. D. Zych, and A. Jøsang, "The impact of quantum computing on present cryptography," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 9, no. 3, pp. 405–414, 2018.

[2] J. F. McDermott and R. H. McDermott, "Obligations of trust for privacy and confidentiality in distributed transactions," *Information Management Computer Security*, vol. 19, no. 2, pp. 153–162, 1999.

[3] M. O. Gbadebo, "Integrating post-quantum cryptography and advanced encryption standards to safeguard sensitive financial records from emerging cyber threats," 2025. Accessed: 2025-10-03.

[4] J. Kurmi and A. Sodhi, "A survey of zero-knowledge proof for authentication," 2015. Accessed: 2025-10-03.

[5] Y. Gong, Y. Jin, Y. Li, Z. Liu, and Z. Zhu, "Analysis and comparison of the main zero-knowledge proof scheme," in *2022 International Conference on Big Data, Information and Computer Network (BDICN)*, pp. 366–372, 2022.