**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

---




**Nguyễn Sơn Lâm**

# ĐỀ ÁN
# NGHIÊN CỨU VÀ ĐÁNH GIÁ CÁC MÔ HÌNH KÝ SỐ ĐA BÊN HẬU LƯỢNG TỬ TRÊN NỀN THUẬT TOÁN CRYSTALS-DILITHIUM

**ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT**

**(Theo định hướng ứng dụng)**

HÀ NỘI - 2025

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Nguyễn Sơn Lâm

# NGHIÊN CỨU VÀ ĐÁNH GIÁ CÁC MÔ HÌNH KÝ SỐ ĐA BÊN HẬU LƯỢNG TỬ TRÊN NỀN THUẬT TOÁN CRYSTALS-DILITHIUM

CHUYÊN NGÀNH: HỆ THỐNG THÔNG TIN

MÃ SỐ: 8.48.01.04

**ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT**

**(Theo định hướng ứng dụng)**

**NGƯỜI HƯỚNG DẪN KHOA HỌC**

**TS Đặng Minh Tuấn**

**PGS.TS Trần Quang Anh**

HÀ NỘI - 2025

# LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của bản thân tôi, dưới sự hướng dẫn của TS. Đặng Minh Tuấn và PGS.TS Trần Quang Anh.

Các số liệu, kết quả nêu trong đề án tốt nghiệp là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác. Tất cả các tham chiếu đều được trích dẫn rõ ràng.

Tác giả đề án tốt nghiệp ký và ghi rõ họ tên



Nguyễn Sơn Lâm

# LỜI CẢM ƠN

Lời đầu tiên em xin được gửi lời cảm ơn chân thành, sâu sắc đến các quý Thầy, Cô giáo tại Học viện Bưu Chính Viễn Thông đã tận tình chỉ dạy và quan tâm em trong suốt quá trình học tập và rèn luyện tại trường.

Con xin gửi tất cả lòng biết ơn chân thành và sự kính trọng của con đến bố, mẹ, ông, bà và gia đình. Đó là những người đã sinh thành, dưỡng dục con để bây giờ con có thể tự tin đi trên đôi chân của mình cho đến ngày hôm nay.

Em trân trọng biết ơn TS. Đặng Minh Tuấn và PGS.TS Trần Quang Anh đã tận tình hướng dẫn, chỉ bảo, góp ý cho em trong suốt quá trình thực hiện đề án. Trong thời gian vừa qua Thầy là người định hướng và giúp đỡ em vượt qua những khó khăn trở ngại trong quá trình này.

Xin chân thành cảm ơn đến người thương yêu, những người bạn đã luôn ở bên, động viên, chia sẻ những lúc khó khăn trong quá trình làm luận văn. Đó là những lời động viên, chia sẻ mà bản thân mãi mãi sẽ ghi nhớ.

Xin chân thành cảm ơn!

# MỤC LỤC

LỜI CAM ĐOAN ..... i

LỜI CẢM ƠN ..... ii

MỤC LỤC ..... iii

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT ..... vi

DANH SÁCH BẢNG ..... viii

DANH SÁCH HÌNH VẼ ..... ix

PHẦN MỞ ĐẦU ..... 1

Chương 1. CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN NGHIÊN CỨU ..... 3

1.1. Khái quát về chữ ký số ..... 3

1.1.1. Cấu trúc thuật toán ký số ..... 3

1.1.2. Ứng dụng trong Môi trường Điện tử ..... 5

1.2. Hệ thống hóa các mô hình ký số đa bên ..... 6

1.2.1. Các mô hình ký số đa bên ..... 6

1.2.2. Ký Ngưỡng(Threshold Signatures – TS) ..... 7

1.2.3. Ký gộp (Aggregate Signatures – AS) ..... 10

1.2.4. Ký độc lập ..... 12

1.3. Mật mã hậu lượng tử (PQC) và CRYSTALS-Dilithium ..... 16

1.3.1. Nguy cơ từ máy tính lượng tử. ..... 16

1.3.2. Giới thiệu về PQC và các họ thuật toán chính. ..... 18

1.3.3. Thuật toán CRYSTALS-Dilithium: Cấu trúc và tầm quan trọng. 21

1.4. Khảo sát và phân tích các nghiên cứu tiêu biểu ..... 34

1.4.1. Nghiên cứu về ký ngưỡng từ mô hình lý thuyết Cozzo & Smart (2019) đến đánh giá thực nghiệm của Roux (2022).....34

1.4.2. Nghiên cứu về đa chữ ký hậu lượng tử của Hamidreza Rahmati và Farhad Rahmati .....36

1.4.3. Nghiên cứu về ký ngưỡng trên thiết bị IOT: Davydov & Bezzateev (2023).....40

1.4.4. Tổng hợp so sánh và xác định hướng triển khai .....43

Chương 2. PHƯƠNG PHÁP NGHIÊN CỨU VÀ THIẾT KẾ MÔ HÌNH THỬ NGHIỆM 45

2.1. Mục tiêu và yêu cầu của thử nghiệm .....45

2.1.1. Mục tiêu: Đo lường các chỉ số hiệu năng. .....45

2.1.2. Yêu cầu: Đảm bảo tính chính xác, khách quan, có thể lặp lại. .46

2.2. Bộ tiêu chí đánh giá hiệu năng.....47

2.2.1. Hiệu năng Tính toán.....47

2.2.2. Hiệu năng Truyền thông.....48

2.2.3. Hiệu quả Lưu trữ.....48

2.2.4. Độ ổn định và Mở rộng.....48

2.3. Thuyết minh mô hình ký ngưỡng (Threshold).....49

2.3.1. Phân tích giao thức.....49

2.3.2. Lưu đồ thuật toán. .....52

2.3.3. Cấu trúc dữ liệu.....55

2.4. Thuyết minh mô hình ký độc lập .....57

2.4.1. Phân tích giao thức.....57

2.4.2. Lưu đồ thuật toán. .....59

2.5. Môi trường và công cụ thử nghiệm.....60

2.5.1. Cấu hình phần cứng, hệ điều hành. ....60
2.5.2. Ngôn ngữ lập trình. ....60
2.5.3. Thư viện sử dụng. ....61
Chương 3. THỰC NGHIỆM, PHÂN TÍCH KẾT QUẢ VÀ KẾT LUẬN .62
3.1. Thiết kế thực nghiệm ....62
3.1.1. Đối tượng và Kịch bản ....62
3.1.2. Môi trường đo lường ....62
3.2. Phân tích kết quả thực nghiệm ....62
3.2.1. Kích thước chữ ký và hiệu quả lưu trữ ....62
3.2.2. Hiệu năng tính toán ....63
3.2.3. Phân tích cơ chế "Bí mật kép" (Dual Secrets) ....64
3.2.4. Phân tích lưu lượng mạng ....65
3.3. Thảo luận ....65
KẾT LUẬN ....66
DANH MỤC CÁC TÀI LIỆU THAM KHẢO ....69

# DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

| Viết tắt | Tiếng Anh | Tiếng Việt |
|---|---|---|
| APK | Aggregate Public Key | Khóa công khai gộp |
| AS | Aggregate Signatures | Chữ ký gộp |
| AVX2 | Advanced Vector Extensions 2 | Tập lệnh mở rộng vector nâng cao thế hệ 2 |
| CPU | Central Processing Unit | Bộ xử lý trung tâm |
| DKG | Distributed Key Generation | Sinh khóa phân tán |
| ECC | Elliptic Curve Cryptography | Mật mã đường cong Elliptic |
| ECDSA | Elliptic Curve Digital Signature Algorithm | Thuật toán chữ ký số đường cong Elliptic |
| e-Gov | Electronic Government | Chính phủ điện tử |
| FFT | Fast Fourier Transform | Biến đổi Fourier nhanh |
| FIPS | Federal Information Processing Standards | Tiêu chuẩn Xử lý Thông tin Liên bang |
| FROST | Flexible Round-Optimized Schnorr Threshold signatures | Chữ ký ngưỡng Schnorr tối ưu vòng linh hoạt |
| GC | Garbled Circuits | Mạch rối |
| IoT | Internet of Things | Internet vạn vật |
| LSSS | Linear Secret Sharing Schemes | Lược đồ chia sẻ bí mật tuyến tính |
| LWE | Learning With Errors | Học có nhiễu |
| ML-DSA | Module-Lattice-Based Digital Signature Standard | Tiêu chuẩn chữ ký số dựa trên lưới module |
| M-LWE | Module Learning With Errors | Module học có nhiễu |
| MPC | Secure Multi-Party Computation | Tính toán đa bên an toàn |
| M-SIS | Module Short Integer Solution | Module nghiệm nguyên ngắn |
| NIST | National Institute of Standards and Technology | Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ |
| NTT | Number Theoretic Transform | Biến đổi số học |
| PK | Public Key | Khóa công khai |
| PQC | Post-Quantum Cryptography | Mật mã hậu lượng tử |
| QFT | Quantum Fourier Transform | Biến đổi Fourier lượng tử |
| RAM | Random Access Memory | Bộ nhớ truy cập ngẫu nhiên |
| ROAST | Robust Asynchronous Schnorr Threshold signatures | Chữ ký ngưỡng Schnorr bền vững bất đồng bộ |
| RSA | Rivest–Shamir–Adleman | Thuật toán Rivest–Shamir–Adleman (Mã hóa RSA) |
| SHA | Secure Hash Algorithm | Thuật toán băm an toàn |
| SHAKE | Secure Hash Algorithm and KECCAK | Thuật toán băm an toàn và KECCAK |
| Sig | Signature | Chữ ký |
| SIMD | Single Instruction Multiple Data | Đơn lệnh đa dữ liệu |

| | | |
|---|---|---|
| SVP | Shortest Vector Problem | Bài toán vector ngắn nhất (Bài toán gần điểm lưới) |
| TLS/SSL | Transport Layer Security / Secure Sockets Layer | Bảo mật tầng giao vận / Lớp socket an toàn |
| TPS | Transactions Per Second | Số giao dịch trên giây |
| TS | Threshold Signatures | Chữ ký ngưỡng |
| XOF | Extensible Output Functions | Hàm băm mở rộng (Hàm đầu ra mở rộng) |
| ZK-rollups | Zero-Knowledge rollups | Giải pháp mở rộng quy mô sử dụng bằng chứng không tiết lộ tri thức |
| ZK-SNARKs | Zero-Knowledge Succinct Non-Interactive Argument of Knowledge | Bằng chứng tri thức không tương tác súc tích và không tiết lộ thông tin |

# DANH SÁCH BẢNG

**Bảng 1.1: Các cấp độ an toàn của Dilithium chuẩn FIPS 204** .....31
**Bảng 1.2: Bảng tóm tắt các đặc điểm kỹ thuật của các nghiên**.....43
**Bảng 3.1: Kích thước chữ ký gộp** .....63
**Bảng 3.2: Tổng quát thời gian giữa các quá trình** .....63
**Bảng 3.3: Phân rã thời gian thực thi giai đoạn Ký (Signing Breakdown)** .....63
**Bảng 3.4: Trung bình các lần lặp từ chối** .....64
**Bảng 3.5: Lưu lượng mạng**.....65

# DANH SÁCH HÌNH VẼ

Hình 1.1. Minh họa cấu trúc thuật toán ký số .....3
Hình 1.2: Mô hình các giai đoạn  ký ngưỡng .....7
Hình 1.3: Mô hình các giai đoạn ký gộp.....10
Hình 1.4: Mô hình các giai đoạn ký độc lập .....13
Hình 1.5: Các tham số đầu vào của Razhi-ms .....37

# PHẦN MỞ ĐẦU

Chữ ký số đóng vai trò then chốt trong việc đảm bảo an toàn thông tin và giá trị pháp lý cho các giao dịch điện tử, từ ký kết hợp đồng, hóa đơn điện tử, phê duyệt báo cáo nội bộ cho đến các ứng dụng trong hệ thống phân tán và công nghệ blockchain. Trong nhiều thập kỷ qua, các thuật toán mật mã khóa công khai như RSA và ECDSA đã được sử dụng rộng rãi và trở thành nền tảng cho hầu hết các hệ thống chữ ký số hiện nay. Tuy nhiên, sự phát triển nhanh chóng của máy tính lượng tử, cùng với sự ra đời của thuật toán Shor, đã đặt ra mối đe dọa nghiêm trọng đối với các hệ mật mã truyền thống, khi chúng có thể bị phá vỡ trong thời gian đa thức trên máy tính lượng tử, làm suy giảm đáng kể mức độ an toàn của chữ ký số hiện hành.[1], [2]

Trước bối cảnh đó, mật mã hậu lượng tử (Post-Quantum Cryptography – PQC)[3] đã trở thành một hướng nghiên cứu quan trọng nhằm xây dựng các thuật toán mật mã có khả năng chống lại cả các cuộc tấn công từ máy tính cổ điển và máy tính lượng tử. Khác với RSA và ECDSA vốn dựa trên các bài toán toán học dễ bị giải bởi thuật toán lượng tử Shor, các thuật toán hậu lượng tử được xây dựng dựa trên những bài toán được cho là khó giải ngay cả đối với máy tính lượng tử, tiêu biểu như bài toán mạng (lattice-based), bài toán mã hóa (code-based) và bài toán dựa trên hàm băm (hash-based). Trong số đó, CRYSTALS-Dilithium là một thuật toán chữ ký số tiêu biểu thuộc nhóm lattice-based và đã được Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (NIST) chính thức lựa chọn làm tiêu chuẩn chữ ký số hậu lượng tử. Việc chuẩn hóa này mở ra cơ hội quan trọng cho nghiên cứu, đánh giá và triển khai các hệ thống chữ ký số an toàn trong kỷ nguyên hậu lượng tử.

Song song với quá trình chuẩn hóa chữ ký hậu lượng tử, nhu cầu ký số đa bên – trong đó nhiều thực thể cùng tham gia ký và xác thực một tài liệu – ngày càng gia tăng trong các lĩnh vực như quản lý văn bản điện tử, hợp đồng số, tài chính – kế toán và các hệ thống phân tán như blockchain. Mô hình ký đa bên không chỉ nâng cao tính minh bạch và độ tin cậy, mà còn giúp phân tán quyền kiểm soát và giảm thiểu rủi ro từ một thực thể đơn lẻ. Tuy nhiên, phần lớn các nghiên cứu hiện nay về chữ ký số hậu

lượng tử vẫn chủ yếu tập trung vào mô hình ký đơn. Các nghiên cứu về ký số đa bên dựa trên CRYSTALS-Dilithium mới chỉ dừng lại ở mức đề xuất lý thuyết hoặc thử nghiệm giới hạn, và đặc biệt, nhiều công trình chưa công bố mã nguồn triển khai, gây khó khăn cho việc kiểm chứng, đánh giá độc lập và ứng dụng thực tiễn.

Xuất phát từ khoảng trống nghiên cứu nêu trên, đề tài "Nghiên cứu và đánh giá các mô hình ký số đa bên hậu lượng tử trên nền CRYSTALS-Dilithium" được thực hiện nhằm tái hiện và triển khai thử nghiệm các mô hình ký số đa bên dựa trên Dilithium, đồng thời tiến hành đo đạc và phân tích các chỉ số hiệu năng quan trọng như thời gian sinh khóa, kích thước khóa, thời gian ký, thời gian xác minh, kích thước chữ ký, dung lượng dữ liệu trao đổi và khả năng mở rộng theo số lượng bên tham gia. Kết quả nghiên cứu kỳ vọng sẽ cung cấp dữ liệu thực nghiệm có giá trị, góp phần đánh giá tiềm năng ứng dụng và đề xuất các hướng cải tiến kỹ thuật cho các hệ thống chữ ký số đa bên hậu lượng tử trong tương lai.

# Chương 1.CƠ SỞ LÝ THUYẾT VÀ TỔNG QUAN NGHIÊN CỨU

## 1.1.Khái quát về chữ ký số

Định nghĩa

Chữ ký số (Digital Signature) là một cơ chế mật mã được sử dụng để xác thực tính toàn vẹn và nguồn gốc của dữ liệu điện tử, cũng như cung cấp khả năng chống chối bỏ (non-repudiation) [4].

Các đặc tính cơ bản của chữ ký số bao gồm:

- Tính toàn vẹn (Integrity): Đảm bảo rằng dữ liệu không bị thay đổi sau khi được ký. Bất kỳ sự thay đổi nào đối với tài liệu sẽ làm cho chữ ký không hợp lệ khi xác minh [4].
- Tính xác thực (Authenticity): Đảm bảo rằng chữ ký được tạo ra bởi người sở hữu khóa bí mật tương ứng, qua đó xác định danh tính của người ký [4].
- Tính chống chối bỏ (Non-repudiation): Ngăn chặn người ký phủ nhận việc mình đã ký tài liệu, vì chỉ có họ mới sở hữu khóa bí mật để tạo ra chữ ký đó [4].

### *1.1.1. Cấu trúc thuật toán ký số*

Một sơ đồ chữ ký số được định nghĩa chính thức bởi một bộ ba thuật toán xác suất đa thức (PPT) hoạt động cùng nhau [4]:




*Hình 1.1. Minh họa cấu trúc thuật toán ký số*

Bước 1. Sinh khóa

- Quy trình: Thuật toán xác suất này nhận đầu vào là tham số an toàn $1^{\lambda}$ (với λ là độ dài khóa) và sinh ra một cặp khóa.
- Đầu ra:

$$\text{KeyGen}(1^{\lambda}) \rightarrow (pk, sk)$$

  + Khóa công khai $pk$ được công bố công khai, cho mọi bên cần xác minh sử dụng để xác nhận đúng người ký với khóa bí mật tương ứng và xác nhận thông điệp không bị sửa đổi.
  + Khóa bí mật $sk$ được giữ bí mật tuyệt đối bởi người ký, thành phần duy nhất có khả năng tạo chữ ký số hợp lệ trên thông điệp $M$.

Bước 2. Ký

- Quy trình: Thuật toán (xác suất hoặc xác định) nhận đầu vào là khóa bí mật $sk$ và thông điệp $m \in \{0,1\}^{*}$.
- Đầu ra: Chữ ký σ

$$\text{Sign}(sk, M) \rightarrow \sigma$$

Bước 3. Xác minh

- Quy trình: Thuật toán xác định nhận đầu vào khóa công khai $pk$, thông điệp $m$, và chữ ký σ.
- Đầu ra: Một bit $\{0,1\}$ thể hiện tính hợp lệ của chữ ký.

$$\text{Verify}(pk, M, \sigma) \rightarrow 0,1$$

- Tính đúng đắn: Sơ đồ phải đảm bảo rằng, với xác suất gần như 1, mọi chữ ký hợp lệ được tạo ra bởi thuật toán Sign đều phải được thuật toán Verify chấp nhận (trả về 1).

### *1.1.2. Ứng dụng trong Môi trường Điện tử*

Tính ứng dụng rộng rãi của chữ ký số bắt nguồn từ khả năng cung cấp các đảm bảo về an ninh phi vật lý tương đương hoặc vượt trội so với chữ ký tay truyền thống. Trong môi trường điện tử, chữ ký số là nền tảng cho nhiều lĩnh vực quan trọng [3], [4]:

- Giao dịch tài chính và ngân hàng: Chữ ký số được sử dụng để xác thực các lệnh chuyển tiền, phê duyệt hợp đồng vay vốn hoặc ủy quyền giao dịch, đảm bảo rằng chỉ chủ tài khoản hợp pháp mới có quyền thực hiện các thao tác này. Việc áp dụng chữ ký số góp phần giảm thiểu rủi ro gian lận và tăng tốc độ xử lý giao dịch nhờ loại bỏ quy trình giấy tờ thủ công[5], [6].
- An ninh mạng và truyền thông: Các chứng chỉ số (như X.509) trong giao thức TLS/SSL dựa vào chữ ký số để xác thực máy chủ và thiết lập kênh truyền thông an toàn trên Internet, đảm bảo tính bí mật và toàn vẹn của dữ liệu truyền tải[5]. Chữ ký số cũng được ứng dụng để xác minh tính toàn vẹn của các bản cập nhật phần mềm và mã nguồn (code signing), giúp ngăn chặn việc phát tán mã độc.
- Hệ thống Blockchain và tiền mã hóa: Trong các hệ thống phân tán như Blockchain, chữ ký số (ví dụ: ECDSA, hoặc trong tương lai là các thuật toán PQC) là cơ chế chủ đạo để chứng minh quyền sở hữu tài sản. Mỗi giao dịch được ký bằng khóa bí mật của người gửi, cho phép mạng lưới xác minh tính hợp lệ của giao dịch mà không cần bên trung gian[5].
- Chính phủ điện tử và thủ tục hành chính: Chữ ký số cho phép cá nhân và doanh nghiệp thực hiện các thủ tục hành chính như khai thuế, đăng ký kinh doanh hoặc ký các hồ sơ trực tuyến, đồng thời đảm bảo giá trị pháp lý của tài liệu điện tử theo quy định của pháp luật[4]

## 1.2. Hệ thống hóa các mô hình ký số đa bên

### *1.2.1. Các mô hình ký số đa bên*

#### 1.2.1.1. Khái quát

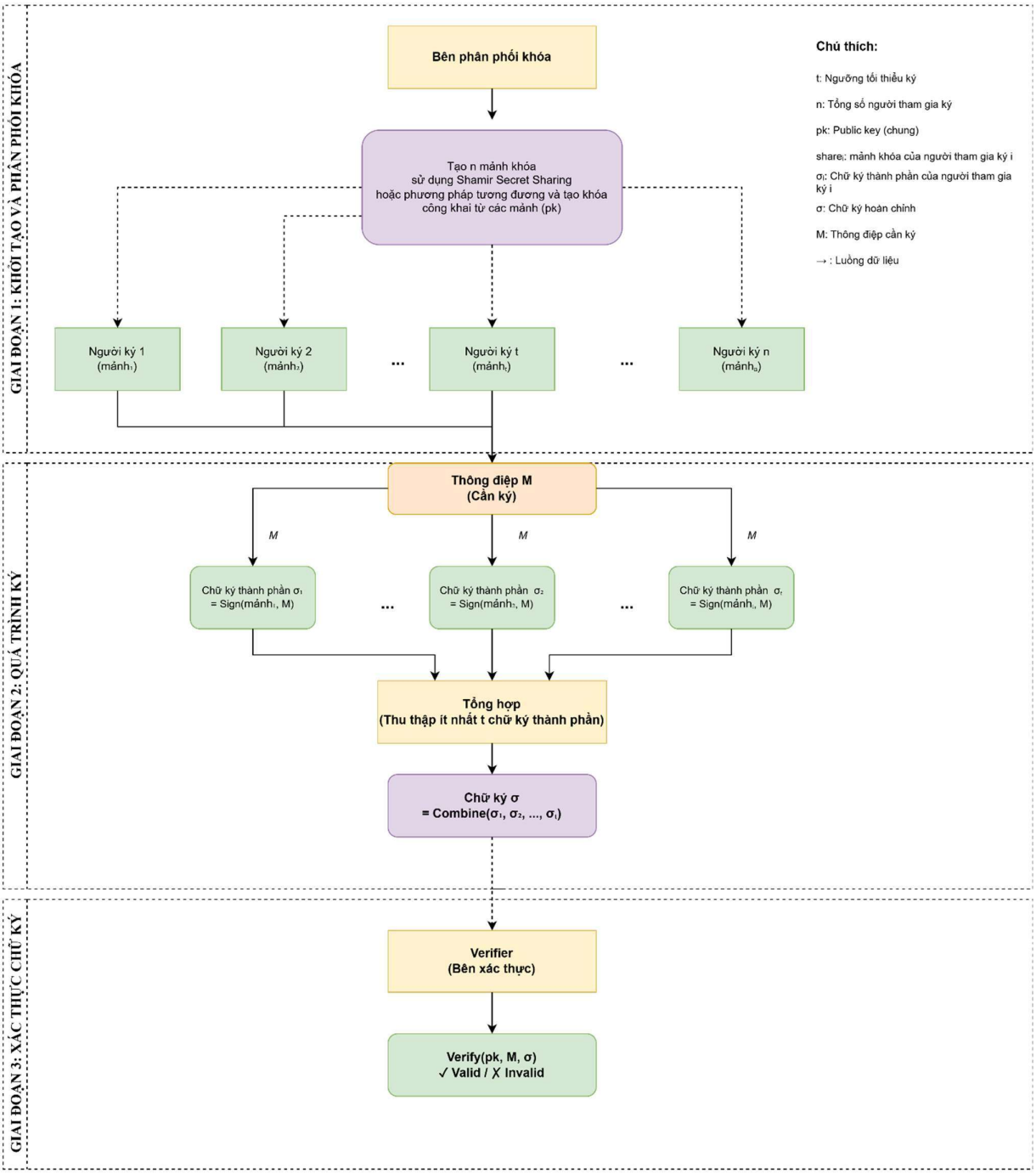
Ký số đa bên là một lớp các giao thức mật mã mở rộng từ chữ ký số truyền thống, cho phép một nhóm các thực thể tham gia ($P_1, P_2, \dots, P_n$) cùng phối hợp để tạo ra một chữ ký hợp lệ cho một thông điệp $M$.[17]

Khác với mô hình ký số đơn lẻ nơi toàn bộ khóa bí mật $sk$ nằm dưới sự kiểm soát của một cá nhân hoặc một máy chủ duy nhất, mô hình ký số đa bên phân tán quyền lực ký. Trong mô hình này, khóa bí mật có thể được chia nhỏ thành các mảnh hoặc tồn tại dưới dạng các khóa độc lập, đảm bảo rằng không một thực thể đơn lẻ nào có thể tự mình giả mạo chữ ký của cả nhóm.[10][16]

#### 1.2.1.2. Các đặc tính

- Chống điểm lỗi đơn lẻ: Nếu khóa bí mật nằm trên một máy chủ và máy chủ đó bị tấn công, toàn bộ hệ thống sụp đổ. Với ký đa bên, kẻ tấn công phải xâm nhập đồng thời vào $t$ máy chủ khác nhau (với $t$ là ngưỡng an toàn) mới có thể giả mạo chữ ký.
- Phân tán niềm tin: Không cần tin tưởng tuyệt đối vào một bên thứ ba trung gian (Trusted Third Party). Quyền lực được chia sẻ, phù hợp với các mô hình phi tập trung như Blockchain hay các bỏ phiếu hội nghị.
- Tính riêng tư: Trong biến thể như ký ngưỡng, cấu trúc nội bộ của nhóm ký được giấu kín. Người bên ngoài chỉ biết nhóm đã ký, nhưng không biết cụ thể các thành viên nào đã tham gia vào phiên ký đó.

### *1.2.2. Ký Ngưỡng(Threshold Signatures – TS)*




*Hình 1.2: Mô hình các giai đoạn ký ngưỡng*

#### 1.2.2.1. Lý thuyết và mục đích

Mô hình ký ngưỡng hướng tới việc bảo vệ khóa bí mật bằng cách phân tách khóa thành nhiều phần sao cho chỉ khi một số lượng tối thiểu $t$ trên tổng số $n$ bên tham gia đồng thuận, chữ ký hợp lệ mới được tạo ra. Điều này được mô tả bởi cặp tham số: $(t, n)$[7]. Kỹ thuật nền tảng của TS là sơ đồ chia sẻ bí mật (Secret Sharing), thường sử dụng sơ đồ Shamir. Khóa bí mật $sk$ sẽ được tách thành $n$ mảnh:

Tạo cặp khóa tổng thể:

$$(sk, pk)$$

- Sau đó chia khóa bí mật thành $n$ phần: $sk \Rightarrow sk_1, sk_2, \dots, sk_n$ và phân phối phần khóa $sk_i$cho từng signer $P_i$. Mỗi signer chỉ biết phần khóa của mình và không suy ra được khóa bí mật tổng thể[7].

Quy trình ký:

- Mỗi signer sẽ ký cục bộ bằng phần khóa của mình:

$$\sigma_i = \text{SignShare}(sk_i, M)$$

- Gửi phần chữ ký $\sigma_i$ đến bộ thu thập (Combiner) và chỉ khi có tối thiểu $t$ mảnh hợp lệ cùng tham gia, chữ ký $\sigma$ mới được sinh ra:

$$\sigma = \text{Combine}(\sigma_1, \sigma_2, \dots, \sigma_t)$$

Quy trình xác thực:

$$\text{Verify}(pk, M, \sigma) \in \text{Valid,Invalid}$$

- Đầu vào sẽ là khóa công khai $pk$, thông điệp $M$, chữ ký ngưỡng $\sigma$. Phần xác thực sẽ không biết được ai đã tham gia ký và không tiết lộ số signer đã ký.
- Quy trình sẽ sử dụng khóa công khai $pk$ để kiểm tra chữ ký có hợp lệ: đúng của người sở hữu $sk$ không và chữ ký $\sigma$ có gắn đúng với thông điệp $M$ hay không, nếu cả hai đều thỏa mãn thì chữ ký là hợp lệ.

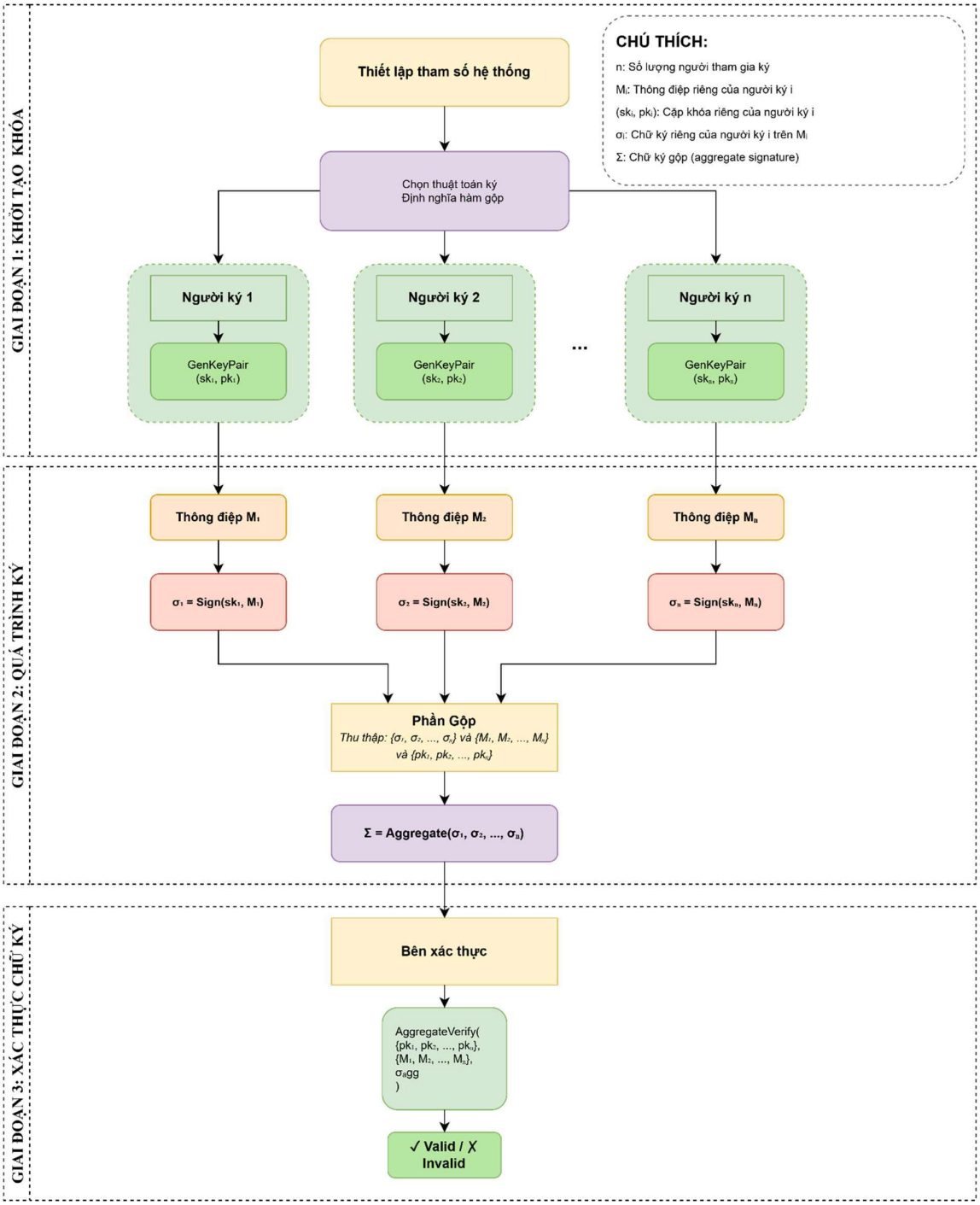
Mục đích chính của TS gồm[7]:

- Phân tán rủi ro: Không bên nào giữ toàn bộ khóa $sk$. Việc một bên bị tấn công không dẫn đến rò rỉ khóa.
- Chống điểm lỗi đơn lẻ: Hệ thống vẫn có thể hoạt động khi một số bên mất quyền truy cập hoặc gặp sự cố.
- Chữ ký đơn gọn: TS tạo một chữ ký duy nhất $\sigma$, có kích thước tương đương chữ ký đơn, giúp giảm băng thông, bộ nhớ và xác minh nhanh chóng.

#### 1.2.2.2. Ứng dụng

- Quản trị tài sản có giá trị cao: Được sử dụng trong các tổ chức tài chính, ví lưu ký tài sản số (custody), yêu cầu sự đồng thuận của một hội đồng quản trị (ví dụ mô hình 3/5 hoặc 5/7).
- Quản trị doanh nghiệp: Các giao dịch lớn hoặc tài liệu quan trọng chỉ hợp lệ khi có đủ chữ ký từ các thành viên có thẩm quyền.
- Hệ thống Blockchain: Là nền tảng cho các mô hình multisig thế hệ mới (ví dụ: FROST, ROAST cho Schnorr; Threshold BLS).
- Tính chịu lỗi cao: Hệ thống có thể hoạt động ngay cả khi một số người dùng ngoại tuyến hoặc mất khóa.
- Hạn chế: TS yêu cầu một số vòng giao thức tương tác giữa các bên để tổng hợp chữ ký. Điều này làm tăng độ trễ và phụ thuộc vào kết nối mạng ổn định, đặc biệt khó triển khai trên quy mô lớn hoặc trong hệ thống phi tập trung.

### *1.2.3. Ký gộp (Aggregate Signatures – AS)*




*Hình 1.3: Mô hình các giai đoạn ký gộp*

#### 1.2.3.1. Lý thuyết và mục đích

Khác với TS, mô hình AS cho phép gộp nhiều chữ ký khác nhau (thường được tạo bởi nhiều bên, trên các thông điệp độc lập) thành một chữ ký duy nhất.[8]

Giả sử có $n$ signer, mỗi signer $P_i$ có cặp khóa:

$$(pk_i, sk_i)$$

Quá trình ký và gộp:

- Mỗi signer $P_i$ cần phải ký một thông điệp riêng $M_i$, quá trình ký được thực hiện và tạo ra chữ ký $\sigma_i$:

$$\sigma_i = \text{Sign}(sk_i, M_i)$$

- Quá trình này kết thúc ta có một tập các chữ ký đơn $\sigma_1, \sigma_2, \dots, \sigma_n$. Tiếp theo sẽ được một đơn vị trung gian (Aggergator) thực hiện việc kết hợp các chữ ký đơn lại thành một chữ ký duy nhất Σ:

$$\Sigma = \text{Aggregate}(\sigma_1, \sigma_2, \dots, \sigma_n)$$

- Cốt lõi của AS dựa trên các cấu trúc đại số tuyến tính, chẳng hạn như: đường cong elliptic (như trong BLS) hoặc không gian véc-tơ của lưới.

Xác thực chữ ký gộp:

- Bên xác thực sẽ nhận đầu vào là chữ ký gộp Σ cùng các cặp khóa công khai và thông điệp tương ứng $(pk_i, M_i)$:

$$\text{AggregateVerify}((pk_i, M_i)_{i=1..n}, \Sigma)$$

- Quy trình xác thực sẽ trả về hợp lệ khi tính chất toán học của chữ ký gộp Σ[8] đảm bảo tính toàn vẹn của toàn bộ tập chữ ký:
  + Tất cả các cặp $(pk_i, M_i)$ từ đầu vào đều đóng góp vào Σ.
  + Không thừa, không thiếu signer.
  + Không có chữ ký nào bị tráo giữa các signer.
  + Không có thông điệp nào bị thay đổi hoặc sai lệch.

Mục đích chính:

- Hiệu quả xác minh: Từ xác minh $n$ chữ ký với chi phí $O(n)$ xuống chỉ còn $O(1)$hoặc $O(\log n)$[8].
- Giảm tải lưu trữ: Toàn bộ nhiều chữ ký được thay bằng một chữ ký duy nhất[8].
- Tối ưu băng thông: Hữu ích khi truyền dẫn lượng dữ liệu lớn qua mạng.

#### 1.2.3.2. Ứng dụng

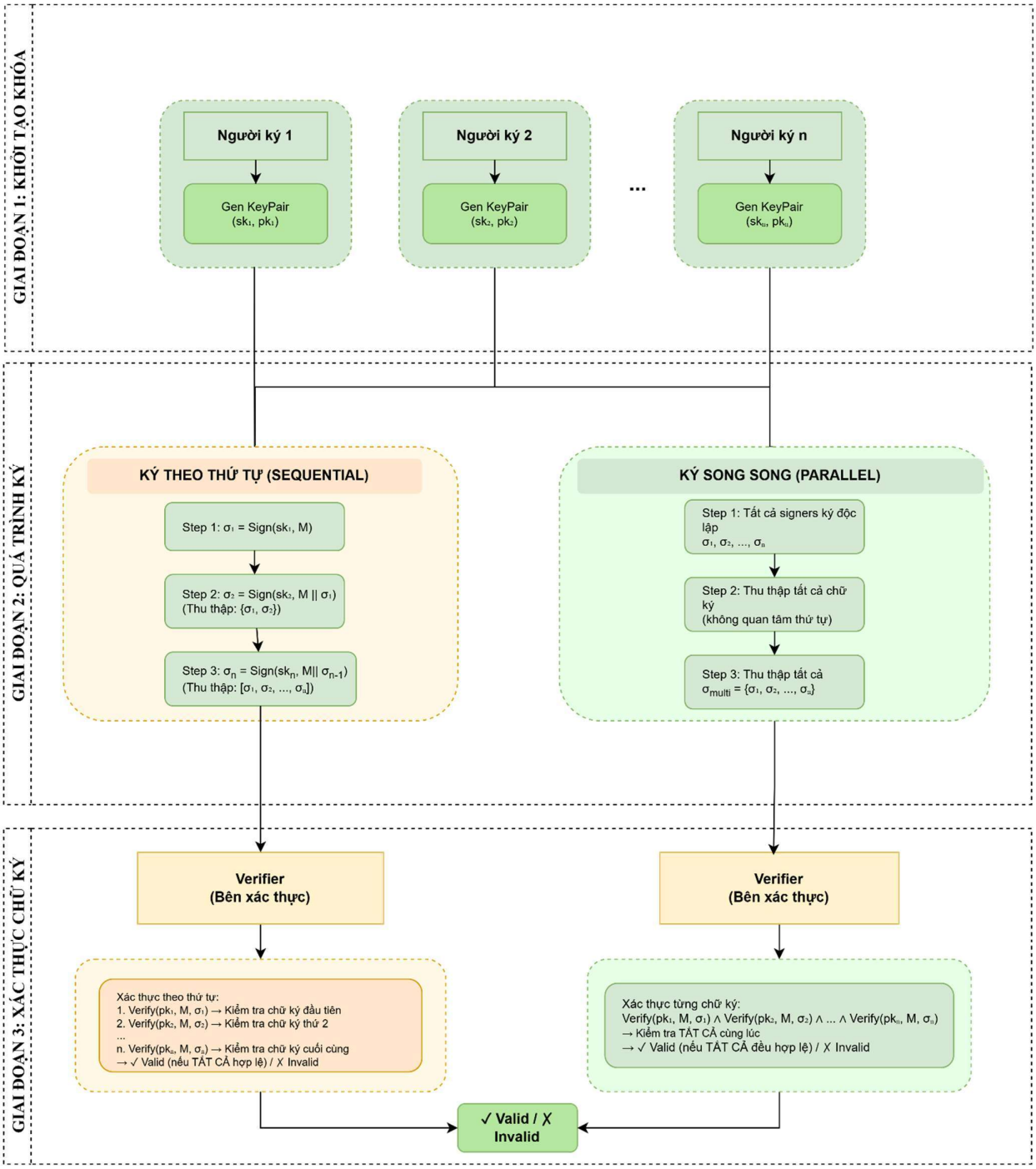
- Nén chứng cứ (Proof Aggregation): Sử dụng trong các hệ thống blockchain hiện đại như Ethereum, ZK-rollups, ZK-SNARKs để gộp hàng trăm hoặc hàng nghìn bằng chứng lại với nhau[9].
- Nhật ký máy chủ: Dùng để giảm kích thước dữ liệu cần ký và cần xác minh.
- Chuỗi khối (Blockchain): Gộp chữ ký trong các block giúp giảm kích thước block, tăng thông lượng giao dịch.
- Các hệ thống IoT: Thiết bị nhỏ có khả năng tính toán hạn chế chỉ cần xác minh một chữ ký gộp thay vì nhiều chữ ký rời rạc[10].

### *1.2.4. Ký độc lập*

Mô hình độc lập là mô hình cơ bản, trong đó mỗi bên $P_i$ sở hữu cặp khóa riêng $(pk_i, sk_i)$ và tự tạo chữ ký $\sigma_i$ trên cùng thông điệp $M$ [11]:

$$\sigma_i = \text{Sign}(sk_i, M)$$

- Phân tán quyền sở hữu: Giao dịch hoặc tài liệu chỉ hợp lệ khi thu thập đủ chữ ký từ các bên liên quan.
- Đơn giản hóa triển khai: Không yêu cầu chia sẻ bí mật hay tạo giá trị ngẫu nhiên phức tạp.
- Không tương tác: Mỗi bên ký độc lập, giảm sự phụ thuộc vào hạ tầng mạng.




*Hình 1.4: Mô hình các giai đoạn ký độc lập*

#### 1.2.4.1. Ký Song Song (Parallel Signing)

Các bên ký trực tiếp lên thông điệp, hoàn toàn độc lập, không phụ thuộc vào nhau. Mỗi signer $P_i$ tự tạo chữ ký[11]:

$$\sigma_i = \text{Sign}(sk_i, M)$$

Kết quả sẽ là một tập các chữ ký:

$$\sigma_1, \sigma_2, \dots, \sigma_n$$

Quy trình xác thực: Mỗi chữ ký được xác minh độc lập: $\text{Verify}(pk_i, M, \sigma_i)$. Lặp qua từng chữ ký trong tập chữ ký. Với mỗi chữ ký $\sigma_i$, kiểm tra xem nó có hợp lệ trên thông điệp $M$với khóa công khai tương ứng $pk_i$ hay không. Nếu tất cả chữ ký đều hợp lệ tài liệu được chấp nhận[11].

- Đặc điểm:
  - + Không yêu cầu liên lạc hoặc trao đổi dữ liệu giữa các signer.
  - + Không có ràng buộc về trình tự.
  - + Người xác minh phải kiểm tra từng chữ ký riêng.
- Hạn chế:
  - + Tăng chi phí xác minh vì phải xác minh từng chữ ký (chi phí tuyến tính theo số signer).
  - + Kích thước chữ ký lớn, vì tập chữ ký tăng theo số lượng người ký, ứng dụng không tối ưu băng thông hoặc bộ nhớ như chữ ký gộp.
  - + Không minh bạch về mức độ đồng thuận, vì không thể xác định được người ký có xem hay đồng ý với chữ ký của người khác hay không.
  - + Không phù hợp quy trình yêu cầu thứ tự phê duyệt, chẳng hạn như phê duyệt hành chính hoặc kiểm soát nội bộ.
- Ứng dụng:

+ Ký hợp đồng nhiều bên: Các bên liên quan ký riêng rẽ một hợp đồng điện tử hoặc văn bản pháp lý.
+ Ký tài liệu PDF nhiều người: Mỗi signer ký vào một trường chữ ký (signature field) riêng trong tài liệu.
+ Chứng thực đồng thuận giữa các thực thể ngang hàng: Nhiều máy chủ hoặc các bên tham gia phê duyệt cùng xác nhận dữ liệu.

#### 1.2.4.2. Ký Theo Thứ Tự (Sequential Signing)

Quy trình ký theo thứ tự là người ký sau chỉ thực hiện ký sau khi đã kiểm tra hoặc chấp nhận chữ ký của người trước đó[11]. Đây là mô hình ký theo chuỗi phê duyệt. Chữ ký của mỗi signer phụ thuộc vào thông điệp và lịch sử ký:

$$\sigma_1 = \text{Sign}(sk_1, M)$$
$$\sigma_2 = \text{Sign}(sk_2, M \parallel \sigma_1)$$
$$\sigma_3 = \text{Sign}(sk_3, M \parallel \sigma_1 \parallel \sigma_2)$$

Kết quả:

$$\Sigma = [\sigma_1, \sigma_2, \sigma_3]$$

Quy trình xác thực phải tuân theo đúng thứ tự ký. Nếu 1 chữ ký sai, toàn bộ chuỗi chữ ký bị bác bỏ.

Ví dụ:

– Xác thực chữ ký đầu tiên

$$\text{Verify}(pk_1, M, \sigma_1)$$

– Nếu $\sigma_1$ hợp lệ tạo lại dữ liệu mà signer 2 đã ký: $M_2 = M \parallel \sigma_1$. Xác minh chữ ký thứ hai:

$$\text{Verify}(pk_2, M_2, \sigma_2)$$

– Tạo lại dữ liệu cho signer thứ $n$: $M_n = M \parallel \sigma_1 \parallel .. \parallel \sigma_n$. Xác minh chữ ký thứ $n$:

$$\text{Verify}(pk_n, M_n, \sigma_n)$$

– Đặc điểm:

+ Có thứ tự ký rõ ràng.
+ Người ký sau phụ thuộc chữ ký trước.
+ Toàn bộ lịch sử ký được bảo toàn.
+ Phù hợp quy trình hành chính hoặc phê duyệt nhiều tầng.

– Hạn chế:

+ Độ trễ cao: người sau phải đợi người trước ký xong.
+ Dễ tắc nghẽn nếu một signer bị chậm hoặc không phản hồi.
+ Kích thước dữ liệu tăng theo số lần ký.
+ Không thể ẩn thứ tự hoặc danh tính người ký.
+ Chi phí xác minh cũng tăng tuyến tính theo số người ký.

– Ứng dụng:

+ Quy trình phê duyệt văn bản trong cơ quan nhà nước.
+ Hồ sơ điện tử trong e-Gov.
+ Duyệt hợp đồng trong doanh nghiệp theo thứ tự cấp quản lý.

## 1.3. Mật mã hậu lượng tử (PQC) và CRYSTALS-Dilithium

### *1.3.1. Nguy cơ từ máy tính lượng tử.*

Máy tính lượng tử là một mô hình tính toán dựa trên nguyên lý của cơ học lượng tử. Thay vì sử dụng bit nhị phân (0 hoặc 1) như trong máy tính cổ điển, máy tính lượng tử sử dụng qubit — một đơn vị thông tin lượng tử có thể tồn tại đồng thời trong nhiều trạng thái nhờ hiện tượng chồng chập (superposition). Ngoài chồng chập, máy tính lượng tử còn khai thác rối lượng tử (entanglement) trạng thái khi hai hoặc nhiều qubit trở nên liên kết với nhau sao cho trạng thái của một qubit sẽ ảnh hưởng

đến trạng thái của qubit kia, bất kể khoảng cách không gian. Cả hai tính chất này cho phép máy tính lượng tử thực hiện tính toán song song ở quy mô lớn[6].

Về nguyên lý vận hành, máy tính lượng tử thực hiện các thao tác trên trạng thái của qubit bằng các phép biến đổi đơn vị (unitary transformations), tương đương với các cổng logic trong máy tính cổ điển nhưng phức tạp hơn. Kết quả cuối cùng được thu về sau khi đo trạng thái hệ qubit, và quá trình đo sẽ làm trạng thái lượng tử sụp đổ thành một giá trị xác định. Nhờ đặc điểm này, máy tính lượng tử có thể giải quyết những bài toán cực kỳ khó đối với máy tính cổ điển, ví dụ như phân tích số nguyên lớn, tìm logarit rời rạc, mô phỏng phân tử trong hóa học lượng tử, hoặc tối ưu hóa tổ hợp. Tuy nhiên, việc xây dựng máy tính lượng tử ở quy mô lớn vẫn còn là một thách thức kỹ thuật và vật lý, mặc dù các tiến bộ gần đây của IBM, Google và các hãng công nghệ khác đang rút ngắn khoảng cách đến hiện thực hóa.

Trong lĩnh vực mật mã, sự xuất hiện của máy tính lượng tử đặt ra mối đe dọa nghiêm trọng đến các hệ thống bảo mật hiện nay như RSA và ECC[5]. Điều này là do các bài toán nền tảng mà những hệ thống này dựa vào, như bài toán phân tích số nguyên và bài toán logarit rời rạc, có thể được giải một cách hiệu quả bằng thuật toán Shor - một trong những thuật toán lượng tử nổi bật nhất.

Thuật toán Shor, được đề xuất bởi Peter Shor vào năm 1994, là một thuật toán lượng tử có khả năng phân tích số nguyên lớn và giải bài toán logarit rời rạc trong thời gian đa thức. Cụ thể, với RSA, thuật toán này cho phép tìm nhanh hai thừa số nguyên tố của một số nguyên lớn $n = p \times q$, từ đó có thể tính toán khóa bí mật $d$ và giải mã toàn bộ hệ thống. Shor hoạt động dựa trên việc tìm chu kỳ của một hàm số bằng biến đổi Fourier lượng tử (Quantum Fourier Transform – QFT). Một khi chu kỳ được xác định, người ta có thể sử dụng các phép toán số học để tìm ra các ước số của n. Sức mạnh của QFT giúp tìm chu kỳ nhanh chóng và chính xác hơn nhiều so với các thuật toán cổ điển.

Với máy tính cổ điển, việc phân tích một số nguyên lớn (ví dụ n có 2048 bit) thành hai thừa số nguyên tố là cực kỳ khó. Thời gian cần thiết để thực hiện tấn công

brute-force hoặc thử nghiệm phân tích số sẽ mất hàng tỷ năm. Các thuật toán tốt nhất hiện tại như GNFS (General Number Field Sieve) vẫn có độ phức tạp siêu đa thức, không đủ khả năng xử lý trong thời gian thực.

Tuy nhiên, với thuật toán Shor trên máy tính lượng tử, bài toán này có thể được giải trong thời gian đa thức nhờ khả năng tính toán song song lượng tử. Việc tìm chu kỳ của hàm $f(x) = a^x mod n$ được thực hiện rất nhanh bằng biến đổi QFT. Điều này giúp rút ngắn thời gian từ hàng tỷ năm xuống chỉ còn vài giờ hoặc thậm chí vài phút, nếu máy tính lượng tử có đủ số lượng qubit và độ ổn định (low noise). Do đó, sức mạnh thực sự của thuật toán Shor không chỉ nằm ở bản thân thuật toán, mà còn ở cách mà máy tính lượng tử xử lý thông tin theo cách hoàn toàn khác biệt và vượt trội so với máy tính cổ điển. Đây chính là lý do mà RSA, vốn được xem là một trong những thuật toán an toàn nhất hiện nay, lại trở nên dễ bị phá vỡ nếu máy tính lượng tử quy mô lớn được hiện thực hóa.

### *1.3.2. Giới thiệu về PQC và các họ thuật toán chính.*

Mật mã hậu lượng tử là một nhánh của mật mã học nhằm xây dựng các thuật toán an toàn trước các cuộc tấn công từ máy tính lượng tử, trong khi vẫn duy trì độ an toàn đối với các máy tính cổ điển. Do các hệ mật mã hiện tại như RSA và ECC dễ bị phá vỡ bởi thuật toán lượng tử Shor, việc nghiên cứu và chuẩn hóa các thuật toán mới có khả năng kháng lượng tử là một ưu tiên cấp thiết.

PQC không dựa vào các bài toán như phân tích số nguyên hay logarit rời rạc. Thay vào đó, các thuật toán hậu lượng tử sử dụng các cấu trúc toán học được cho là khó giải ngay cả với máy tính lượng tử. Dưới đây là các nhóm chính trong mật mã hậu lượng tử[13]:

- Lattice-based cryptography (mật mã dựa trên mạng): Dựa vào độ khó của bài toán gần điểm lưới (Shortest Vector Problem – SVP) và Learning With Errors (LWE). Đây là nhóm có nhiều thuật toán nổi bật như CRYSTALS-Dilithium, Falcon.

- Code-based cryptography (mật mã dựa trên lý thuyết mã): Dựa vào các bài toán khó trong lý thuyết mã như bài toán giải mã mã McEliece.
- Hash-based cryptography (mật mã dựa vào hàm băm): Sử dụng các cấu trúc như Merkle tree để tạo chữ ký số, tiêu biểu là SPHINCS+.
- Multivariate cryptography (mật mã đa biến): Dựa trên hệ phương trình bậc hai nhiều biến trên trường hữu hạn, có lịch sử lâu đời nhưng hiệu năng chưa tối ưu.
- sogeny-based cryptography (mật mã dựa trên ánh xạ đẳng cấu elliptic): Một hướng nghiên cứu mới, tiêu biểu là SIKE, tuy nhiên đã bị phá vỡ một số biến thể.

Trong bối cảnh toàn cầu đang gấp rút tìm kiếm các giải pháp mật mã kháng lượng tử, Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (NIST) đã khởi động một chương trình chuẩn hóa các thuật toán mật mã hậu lượng tử từ năm 2016. Mục tiêu của chương trình là đánh giá, lựa chọn và chuẩn hóa các thuật toán đủ an toàn, hiệu quả, có thể thay thế cho RSA và ECC trong tương lai[13].

Chương trình này bao gồm ba vòng tuyển chọn chính, kéo dài từ năm 2017 đến 2022[13]:

- Vòng 1 (2017–2019): NIST tiếp nhận hơn 80 thuật toán ứng viên từ các nhóm nghiên cứu trên toàn thế giới. Các thuật toán này được đánh giá sơ bộ về tính đúng đắn, bảo mật và hiệu suất.
- Vòng 2 (2019–2020): 26 ứng viên được chọn vào vòng hai để đánh giá sâu hơn, bao gồm kiểm tra khả năng chống tấn công lượng tử và hiệu năng thực tế trên phần cứng và phần mềm.
- Vòng 3 (2020–2022): 15 ứng viên cuối cùng được chọn để cạnh tranh vị trí chuẩn hóa. Trong đó, một số được đề xuất chuẩn hóa ngay, số còn lại tiếp tục được nghiên cứu thêm (gọi là "chương trình bổ sung").

NIST đã đặt ra một số tiêu chí chính cho việc lựa chọn thuật toán chuẩn[13]:

- Độ an toàn: Không có lỗ hổng được biết đến trước cả máy tính cổ điển và lượng tử.
- Hiệu suất: Ký và xác minh nhanh, phù hợp với các thiết bị có tài nguyên hạn chế như IoT.
- Kích thước: Khóa công khai và chữ ký nên có kích thước hợp lý.
- Tính linh hoạt: Có thể áp dụng trong nhiều bối cảnh khác nhau như TLS, VPN, blockchain, firmware signing…
- Khả năng triển khai: Dễ lập trình, tránh lỗi, kháng tấn công kênh bên (side-channel attackSau quá trình đánh giá nghiêm ngặt trong 5 năm, trong đó bao gồm các thuật toán ký số và mã hóa.

Đối với nhóm thuật toán ký số, ba thuật toán ký số đã được lựa chọn đưa vào chuẩn cuối cùng gồm[14]:

- CRYSTALS-Dilithium(FIPS 204): Lattice-based, hiệu suất cao, chữ ký và khóa công khai có kích thước trung bình.

- Falcon: Cũng thuộc lattice-based, có kích thước chữ ký nhỏ, tốc độ ký và xác minh nhanh nhưng khó triển khai.
- SPHINCS+(FIPS 205): Hash-based, độ an toàn cao nhưng kích thước chữ ký lớn và tốc độ chậm hơn.

Các thuật toán này được thiết kế không chỉ để chống lại các máy tính lượng tử, mà còn đảm bảo tương thích với phần cứng hiện tại, độ an toàn toán học và tính hiệu quả khi triển khai trong thực tế.

### *1.3.3. Thuật toán CRYSTALS-Dilithium: Cấu trúc và tầm quan trọng.*

#### 1.3.3.1. Các phép toán cốt lõi và cấu trúc toán học của CRYSTALS-Dilithium

Sức mạnh của CRYSTALS-Dilithium đến từ việc kết hợp chặt chẽ giữa độ khó của các bài toán mạng mô-đun và hiệu suất vượt trội của các phép toán trên vành đa thức[15]. Các phép toán nền tảng có thể được phân thành bốn nhóm chức năng sau.

##### *1.3.3.1.1. Nền tảng Đại số*

Đây là lớp cấu trúc toán học nơi mọi phép tính của Dilithium diễn ra. Thuật toán không làm việc trực tiếp trên số nguyên mà trên các đối tượng đại số phức tạp nhằm đảm bảo độ an toàn và tối ưu hiệu năng.

*i. Trường hữu hạn* $\mathbb{Z}_q$

Tầng cơ sở của Dilithium là trường hữu hạn các số nguyên modulo $q$:

$$\mathbb{Z}_q = 0,1, \dots, q-1$$

Thuật toán sử dụng tham số: $q = 8{,}380{,}417$, $n = 256$. Điều kiện lựa chọn tham số $q$ thỏa mãn: $q \equiv 1(\mathrm{mod} 2n)$.

*ii. Vành đa thức (Polynomial Ring)* $R_q$

Dilithium không thực hiện tính toán trực tiếp trên các số nguyên mà trên các đa thức thuộc vành thương:

$$R_q = \mathbb{Z}_q[x] \,/\, (x^n + 1)$$

Trong đó:

- $\mathbb{Z}_q$: trường hữu hạn chứa các hệ số của đa thức,
- $x^n + 1$: đa thức cyclotomic dùng để xác định quy tắc rút gọn bậc và nhân đa thức.

Các phần tử của $R_q$ là các đa thức có bậc nhỏ hơn $n$, với hệ số thuộc $\mathbb{Z}_q$. Việc sử dụng cấu trúc vành này giúp: biểu diễn dữ liệu gọn hơn, hỗ trợ phép nhân đa thức hiệu quả khi kết hợp với NTT, giảm độ phức tạp tính toán từ $O(n^2)$ xuống còn $O(n\log n)$, điều này trực tiếp tăng tốc quá trình ký và xác minh của Dilithium.

*iii.Mạng Mô-đun (Module Lattices)*

Tầng toán học cao nhất trong Dilithium là mạng mô-đun, mở rộng khái niệm mạng (lattice) sang không gian của các vector đa thức. Thay vì sử dụng một đa thức đơn lẻ, Dilithium làm việc với các vector và ma trận đa thức có dạng:

$$\mathrm{t} = \mathrm{A}s_1 + s_2$$

Trong đó:

- A: ma trận $k \times \ell$ gồm các phần tử thuộc $R_q$,
- $s_1, s_2$: các vector bí mật và nhiễu,
- t: vector kết quả quan sát được.

Các bài toán khó bảo mật nền tảng của Dilithium bao gồm:

- M-LWE (Module Learning With Errors):
  + Đây là bài toán nền tảng bảo vệ tính bí mật của khóa. Trong Dilithium, M-LWE đảm bảo rằng khóa bí mật $(s_1, s_2)$không thể bị khôi phục từ khóa công khai $(\mathrm{A}, \mathrm{t})$. Bài toán yêu cầu: Tìm lại vector bí mật s từ phương trình: $\mathrm{t} = \mathrm{A}s_1 + s_2$ trong đó $s_2$ là nhiễu nhỏ được thêm vào cố ý.

+ Việc nhiễu phá vỡ cấu trúc tuyến tính khiến bài toán trở nên khó ngay cả với máy tính lượng tử.

– M-SIS (Module Short Integer Solution)

+ Là bài toán nền tảng đảm bảo tính toàn vẹn và khả năng chống giả mạo của chữ ký Dilithium. Bài toán này chứng minh rằng không ai có thể tạo ra chữ ký hợp lệ mà không biết khóa bí mật, đảm bảo tính chống giả mạo chữ ký.

+ Cho ma trận ngẫu nhiên đều: A ma trận $k \times \ell$gồm các phần tử thuộc $R_q$. Một hằng số giới hạn $\beta$.

Bài toán cần giải: Tìm một vector $\neq 0$: $\mathrm{z} \in R_q^{k+\ell}$ thỏa mãn:

$$[\,\mathrm{A} \mid \mathrm{I}\,] \cdot \mathrm{z} = 0 (\mathrm{mod} q)$$

và đồng thời:

$$\| \mathrm{z} \|_\infty < \beta$$

Bài toán này được xem là rất khó trong cả không gian cổ điển và lượng tử.

*1.3.3.1.2. Công cụ Tăng tốc Tính toán*

Để khắc phục những hạn chế về hiệu năng vốn có của các hệ mật mã dựa trên lưới, CRYSTALS-Dilithium tích hợp nhiều kỹ thuật toán học tối ưu, cho phép chuyển đổi các phép toán đại số phức tạp thành những phép tính số học đơn giản, có thể thực thi hiệu quả và song song trên phần cứng hiện đại. Hai kỹ thuật quan trọng nhất là Biến đổi Số học NTT và Phép nhân ma trận–đa thức[15].

*i. Biến đổi Số học NTT*

NTT là phiên bản rời rạc của Biến đổi Fourier Nhanh (FFT), nhưng hoạt động trong trường hữu hạn $\mathbb{Z}_q$. Đây là phép toán tăng tốc cốt lõi trong Dilithium.

– Thách thức tính toán: Trong vành đa thức: $R_q = \mathbb{Z}_q[x]/(x^n + 1)$, việc nhân hai đa thức bậc $n - 1$ theo cách truyền thống (tích chập – convolution) có

độ phức tạp là $O(n^2)$. Với $n = 256$, mỗi phép nhân đa thức đòi hỏi hơn 65.000 phép toán, gây ảnh hưởng nghiêm trọng tới hiệu năng thực tế.

– Phương pháp tiếp cận: Trong Dilithium chuyển đổi các đa thức từ miền hệ số sang miền tần số (NTT domain) bằng cách đánh giá chúng tại các căn nguyên thủy của đơn vị modulo $q$. Trong miền NTT, phép nhân đa thức phức tạp trở thành: Phép nhân từng điểm. Sau đó áp dụng biến đổi NTT ngược để thu được kết quả. Điều này thay thế phép tích chập có độ phức tạp $O(n^2)$ bằng:

$$O(n \log n)$$

– Hiệu quả và đặc tính tối ưu:

  + Dilithium sử dụng cấu trúc vành $x^n + 1$, cho phép áp dụng NTT mà không cần zero-padding.
  + Điều này giúp giảm 50% bộ nhớ và băng thông so với các hệ thống dùng NTT thông thường.
  + Nhờ lựa chọn $q$ thỏa mãn $q \equiv 1(\mathrm{mod} 2n)$, Dilithium đảm bảo tồn tại căn nguyên thủy phù hợp cho NTT.

*ii. Phép nhân Ma trận–Đa thức*

– Đây là phép toán chiếm phần lớn thời gian trong sinh khóa, ký và xác minh. Công thức tổng quát:

$$\mathrm{w} = \mathrm{Az}$$

Hoặc trong quá trình sinh khóa:

$$\mathrm{t} = \mathrm{As}_1 + \mathrm{s}_2$$

– Trong đó:

  + $\mathrm{A} \in R_q^{k \times \ell}$: ma trận đa thức công khai
  + $\mathrm{s}_1, \mathrm{s}_2$: các vector đa thức bí mật

- Thay vì thực hiện phép nhân ma trận đa thức trực tiếp, Dilithium áp dụng triệt để miền NTT:
  + Lưu trữ ma trận $A$ trực tiếp dưới dạng NTT. Ma trận A được sinh từ seed $\rho$ thông qua SHAKE-128 và Dilithium lưu A dưới dạng NTT:

$$\widehat{\mathrm{A}} = \mathrm{NTT}(\mathrm{A})$$

  + Điều này loại bỏ hoàn toàn chi phí thực hiện Forward NTT cho A khi ký hoặc xác minh.
  + Giúp giảm hơn một nửa chi phí tính toán liên quan đến A.

Tính toán song song trong miền NTT. Các vector bí mật cũng được đưa vào miền NTT: $\hat{\mathrm{s}}_1 = \mathrm{NTT}(\mathrm{s}_1), \hat{\mathrm{s}}_2 = \mathrm{NTT}(\mathrm{s}_2)$. Sau đó phép nhân: $\widehat{\mathrm{A}} \circ \hat{\mathrm{s}}_1$ được thực hiện theo từng điểm, do các phép nhân điểm này độc lập với nhau nên chúng có thể thực hiện hoàn toàn song song.

Tương thích Phần cứng (Hardware Acceleration): Cấu trúc NTT và phép nhân điểm của Dilithium có tính tương thích cao với các tập lệnh vector hóa trên CPU hiện đại:

  + SIMD (Single Instruction Multiple Data): Cho phép một lệnh CPU xử lý đồng thời nhiều hệ số dữ liệu.
  + Triển khai thực tế: Các thư viện tối ưu (như AVX2 trên Intel/AMD hoặc Neon trên ARM) có thể xử lý 8 đến 16 hệ số đa thức cùng một lúc trong một chu kỳ máy, giúp tăng tốc độ ký và xác minh lên nhiều lần so với cài đặt tham chiếu.

*1.3.3.1.3. Cơ chế Sinh Ngẫu nhiên và Bảo mật*

Độ an toàn của CRYSTALS-Dilithium không chỉ dựa vào độ khó của các bài toán mạng mô-đun, mà còn phụ thuộc vào sự vững chắc của các mật mã nguyên thủy hỗ trợ sinh tham số và xử lý dữ liệu. Nhóm chức năng này đảm bảo tính không thể

dự đoán và tính kháng va chạm, vốn là hai yêu cầu cốt lõi của bất kỳ hệ chữ ký hiện đại nào.

*i. Hàm băm mở rộng (Extensible Output Functions – XOF)*

Khác với các hàm băm có đầu ra cố định như SHA-256, Dilithium sử dụng SHAKE-128 và SHAKE-256, hai biến thể của họ Keccak theo tiêu chuẩn FIPS 202. Các hàm băm dạng XOF cho phép tạo ra chuỗi đầu ra giả ngẫu nhiên có độ dài tùy ý từ một đầu vào duy nhất, hoạt động như một Mô hình Tiên tri Ngẫu nhiên (*Random Oracle Model*).

- SHAKE-128
  - + Mục đích: Dùng trong các tác vụ yêu cầu hiệu năng và tốc độ cao, chủ yếu liên quan đến dữ liệu công khai.
  - + Ứng dụng chính: function ExpandA(), mở rộng hạt giống (seed) $\rho \in \{0,1\}^{256}$ thành ma trận đa thức kích thước lớn $\mathrm{A} \in R_q^{k \times \ell}$.
  - + Ý nghĩa kỹ thuật: Cho phép tiết kiệm băng thông truyền tải. Thay vì phải gửi toàn bộ ma trận A(khoảng hàng chục KB), chỉ cần truyền seed $\rho$; người nhận có thể tái tạo lại chính xác A bằng SHAKE-128.
- SHAKE-256
  - + Mục đích: Dùng trong các tác vụ cần mức bảo mật cao nhất (Security Level 5).
  - + Ứng dụng chính: Tạo vector thử thách trong sơ đồ Fiat–Shamir $\mathrm{c} = \mathrm{H}(\mu, \mathrm{w}_1)$, đồng thời dùng để băm thông điệp cần ký.
  - + Ý nghĩa kỹ thuật: Đảm bảo tính kháng va chạm mạnh, ngăn chặn khả năng tìm hai thông điệp khác nhau nhưng sinh cùng thử thách c, từ đó chống lại tấn công giả mạo.

*ii. Các kỹ thuật lấy mẫu*

Lựa chọn phân phối xác suất cho các tham số của Dilithium là yếu tố quyết định đến độ an toàn đối với Module-LWE và Module-SIS. Đây cũng là điểm khác biệt quan trọng giữa Dilithium và các thuật toán dùng phân phối Gaussian như Falcon.

- Lấy mẫu đều (Uniform Sampling)
  - + Đối tượng: Ma trận công khai
    $$\mathrm{A} \in R_q^{k \times \ell}.$$
  - + Cơ chế: Mỗi hệ số của đa thức được lấy đều trong trường hữu hạn:
    $$a_{i,j}(x) \sim U(\mathbb{Z}_q).$$
  - + Mục đích: Đảm bảo không tồn tại cấu trúc ẩn nào trong A để kẻ tấn công khai thác; bài toán Module-LWE/M-SIS giữ được độ khó trường hợp trung bình (average-case hardness).
- Lấy mẫu trong khoảng đều
  - + Đối tượng: Các vector bí mật
    $$\mathrm{s}_1, \mathrm{s}_2 \in R^n, s_{i,j} \in [-\eta, \eta].$$
  - + Cơ chế: Mỗi hệ số được chọn độc lập và đều trong khoảng nhỏ:
    $$s_{i,j} \sim U([-\eta, \eta]).$$
  - + Ưu điểm so với Gaussian (Falcon):
  - + Dễ triển khai, không yêu cầu số thực hay bảng tra cứu phức tạp.
  - + Dễ bảo đảm thời gian hằng số, giảm nguy cơ rò rỉ thông tin qua kênh kề (Timing Attacks). Đây là yếu tố đặc biệt quan trọng cho triển khai phần cứng nhúng và thiết bị IoT.
- Lấy mẫu loại bỏ (Rejection Sampling) – Cơ chế "Zero-Knowledge"
  - + Đây là kỹ thuật tinh vi nhất dùng trong quá trình ký để che giấu khóa bí mật. Một trong các vấn đề bảo mật là chữ ký ban đầu có dạng:

$$z = y + cs_1.$$

+ Vì y và $c$ là giá trị công khai hoặc có thể suy ra, nên z chứa thông tin thống kê về $s_1$. Nếu tái sử dụng nhiều lần, kẻ tấn công có thể áp dụng các phương pháp hồi quy để ước lượng $s_1$.
+ Để giải quyết vấn đề trên, thuật toán quy định rằng z phải nằm trong miền an toàn:

$$\| z \|_\infty < \gamma_1 - \beta.$$

+ Nếu điều kiện trên không thỏa mãn hoặc có nguy cơ rò rỉ, chữ ký bị loại bỏ và quá trình ký được thực hiện lại với vector $y$ mới. Kết quả ta thu được, phân phối của chữ ký: $\sigma = (z, h, c)$ trở nên độc lập hoàn toàn với khóa bí mật $s_1$. Điều này mang lại tính chất tương tự bằng chứng không tiết lộ tri thức (*Zero-Knowledge Proof*), đảm bảo chữ ký không rò rỉ bất kỳ thông tin nào về khóa riêng.

*1.3.3.1.4. Kỹ thuật Nén và Ổn định*

Đây là một trong các đặc điểm kỹ thuật của Dilithium, giúp giảm kích thước khóa và chữ ký mà không làm mất tính chính xác.

*i. Phép Lượng tử hóa*

– Hàm Power2Round (Sử dụng trong Sinh khóa) Hàm này được dùng để nén vector $t = As1 + s2$. Với mỗi hệ số $t$, hàm thực hiện tách giá trị theo cơ số $2^d$:

$$t = t_1 \cdot 2^d + t_0$$

+ Trong Dilithium đều dùng $d = 14$. Tách một giá trị $t$ thành:

$$(t_1, t_0) = \text{Power2Round}(t)$$

+ $t_1$: phần bit cao (được đưa vào khóa công khai),
+ $t_0$: phần dư bị loại bỏ hoặc giữ trong khóa bí mật.

– Hàm Decompose (Sử dụng trong Ký số):

Được áp dụng để nén vector chữ ký và vector trung gian. Mỗi hệ số $r$ được phân rã dựa trên tham số $\alpha$(thường chọn $\alpha = 2^{\gamma_2}$):

$$r = r_1 \cdot 2\alpha + r_0$$

– Trong đó:

+ $r_1$(High Bits): phần thông tin quan trọng, được sử dụng khi sinh thử thách trong hàm băm hoặc trong chữ ký rút gọn.
+ $r_0$(Low Bits): phần dư nhỏ, biểu diễn nhiễu cho phép.

Mục đích: Đảm bảo rằng việc xác minh chữ ký không phụ thuộc vào các dao động nhỏ của nhiễu, đồng thời giảm kích thước dữ liệu cần gửi.

*ii. Cơ chế gợi ý*

Do quá trình lượng tử hóa (loại bỏ bit thấp) ở bước trên, giữa Người Ký và Người Xác Minh sẽ xuất hiện một độ lệch số học nhỏ. Nếu độ lệch này xảy ra ở ranh giới của các khoảng làm tròn, nó có thể dẫn đến kết quả xác minh sai lệch hoàn toàn (ví dụ: 3.9 làm tròn thành 4, 4.1 làm tròn thành 4, nhưng nếu cộng thêm nhiễu nhỏ có thể nhảy sang 5).

– MakeHint (bên Ký): Khi tính toán chữ ký, thuật toán kiểm tra xem liệu việc cộng nhiễu có làm thay đổi "bit cao" của các hệ trong $\mathrm{w} = \mathrm{Ay}$ hay không. Nếu có sự thay đổi ranh giới, một "gợi ý" (hint) 1-bit được tạo ra:

$$\mathrm{h} = \mathrm{MakeHint}(\ldots)$$

Vector gợi ý h có trọng số Hamming thấp và được gửi kèm chữ ký.

– UseHint (bên Xác Minh): bên Xác Minh sử dụng vector gợi ý h để điều chỉnh kết quả làm tròn của mình:

$$\mathrm{w}'_{\mathrm{high}} = \mathrm{UseHint}\ (\mathrm{h},\ \mathrm{Az} - c\,\mathrm{t}_1 \cdot 2^d)$$

- Nhờ đó, bên Xác Minh khôi phục được chính xác phần bit cao của w(giá trị mà bên Ký đã cam kết) bất chấp sự tồn tại của nhiễu và việc lượng tử hóa trước đó.

*1.3.3.1.5. Cơ chế Đóng gói dữ liệu*

Để tối ưu hóa việc lưu trữ và truyền tải, Dilithium không lưu trữ các hệ số dưới dạng số nguyên 32-bit thông thường mà sử dụng các kỹ thuật đóng gói bit (Bit Packing) chuyên dụng.

Mục đích: Biến đổi các vector đa thức (cấu trúc toán học) thành chuỗi byte (mảng uint8_t) để truyền qua mạng hoặc lưu vào đĩa cứng.

Kỹ thuật:

+ Các hệ số trong trường $Z_q$ (cần 23 bit) được nén khít lại với nhau, loại bỏ các bit 0 dư thừa.
+ Các vector nén (như $t_1$, chữ ký $z$) được cắt bớt số bit biểu diễn dựa trên giới hạn biên độ của chúng (ví dụ: chỉ dùng 10 bit hoặc 4 bit cho mỗi hệ số thay vì 23 bit).

Ý nghĩa: Kỹ thuật này giúp Dilithium đạt được kích thước khóa và chữ ký nhỏ nhất có thể (ví dụ: Dilithium2 có chữ ký chỉ 2.4KB), tạo lợi thế cạnh tranh lớn so với các thuật toán PQC khác như SPHINCS+ (chữ ký lên tới hàng chục KB).

*1.3.3.1.6. Các bộ tham số chuẩn*

Cấu trúc của Dilithium cho phép linh hoạt điều chỉnh độ an toàn bằng cách thay đổi kích thước của mạng mô-đun $(k, \mathrm{l})$. Theo chuẩn FIPS 204, có ba mức tham số chính:

***Bảng 1.1: Các cấp độ an toàn của Dilithium chuẩn FIPS 204***

| Bộ tham số | Mức an toàn (NIST) | Kích thước Mạng (k,ℓ) | Tương đương cổ điển |
|---|---|---|---|
| Dilithium2 | Level 2 (Medium) | (4,4) | AES-128 / SHA-256 |
| Dilithium3 | Level 3(Recommended) | (6,5) | AES-192 / SHA-384 |
| Dilithium5 | Level 5 (High) | (8,7) | AES-256 / SHA-512 |

#### 1.3.3.2. Quy trình hoạt động tổng quan (Algorithmic Workflow)

Dựa trên nền tảng toán học đã phân tích, CRYSTALS-Dilithium vận hành thông qua ba thuật toán chính. Dưới đây là mô tả tóm tắt quy trình (chi tiết kỹ thuật sẽ được trình bày ở Chương 2):

##### *1.3.3.2.1. Thuật toán Sinh khóa (Key Generation - KeyGen)*

Quá trình tạo cặp khóa công khai ($pk$) và bí mật ($sk$):

Sử dụng SHAKE-128 để mở rộng một seed ngẫu nhiên thành ma trận đa thức $A \in R_q^{k\times l}$ (dạng NTT).

Lấy mẫu các vector bí mật $s_1, s_2$trong khoảng nhỏ.

Tính toán $t = As_1 + s_2$.

Áp dụng Power2Round để tách $t$ thành $t_1$(công khai) và $t_0$ (bí mật).

Kết quả: $pk = (\rho, t_1)$, $sk = (\rho, K, tr, s_1, s_2, t_0)$.

##### *1.3.3.2.2. Thuật toán Ký số (Signing - Sign)*

Quá trình ký thông điệp $M$ trong Dilithium dựa trên biến thể Fiat–Shamir với kỹ thuật "hủy bỏ" (abort/rejection sampling). Các bước thực hiện như sau:

Bước 1. Cam kết: Sinh ngẫu nhiên vector che $y \leftarrow \mathcal{D}_\eta$. Tính $w = Ay$.

Bước 2. Tạo thử thách: Băm thông điệp và phần cao của của w, ký hiệu $w_1$ để tạo thử thách $c = H(w1, M)$ trong đó $c \in R_q$ có trọng số Hamming nhỏ.

Bước 3. Tính chữ ký tiềm năng: Tính $z = y + cs_1$.

Bước 4 Rejection Sampling: Kiểm tra xem:

+ z phải thuộc miền nhỏ (không vượt ngưỡng quy định),
+ Việc công bố z không được làm lộ thông tin về $s_1$.

Nếu không đạt, quay lại Bước 1 chọn $y$ khác.

Bước 5 (Tạo gợi ý): Nếu đạt, tạo gợi ý $h$ để sửa lỗi làm tròn.

Kết quả: Chữ ký là:$\sigma = (z, h, c)$.

*1.3.3.2.3. Thuật toán Xác minh (Verification - Verify)*

Quá trình kiểm tra tính hợp lệ của chữ ký $\sigma = (z, h, c)$trên thông điệp $M$được thực hiện như sau:

Tái tạo ma trận công khai:Từ seed $\rho$trong khóa công khai $pk$, khởi tạo lại ma trận:

$$A \in R_q^{k \times \ell}.$$

Tính toán vector phục hồi:Tính giá trị xấp xỉ của vector cam kết:

$$w' = Az - c\, t_1 \cdot 2^d.$$

Sau đó sử dụng gợi ý hđể điều chỉnh phần bit cao:

$$w_1' = \text{UseHint}(h, w').$$

Kiểm tra tính nhất quán của thử thách:Kiểm tra xem thử thách được tái tạo có trùng khớp với thử thách trong chữ ký hay không:

$$c = H(w_1', M).$$

Kiểm tra ràng buộc trên z: Đảm bảo rằng tất cả các hệ số của z đều nằm trong miền giới hạn quy định (nhỏ hơn ngưỡng cho phép), để không phá vỡ độ khó của bài toán M-SIS.

#### 1.3.3.3. Ưu điểm và Tầm quan trọng (Significance)

Phần này đặc biệt quan trọng để biện minh lý do bạn chọn Dilithium cho đề tài Thạc sĩ.

##### *1.3.3.3.1. Vai trò trong Tiêu chuẩn hóa NIST (Standardization Status)*

CRYSTALS-Dilithium đã chính thức được NIST lựa chọn là tiêu chuẩn chữ ký số chính (Primary Standard) cho mức độ bảo mật PQC, được ban hành dưới mã số FIPS 204 (ML-DSA) vào tháng 8/2024.

Nó được khuyến nghị sử dụng mặc định cho hầu hết các ứng dụng bảo mật thông tin thay thế cho RSA và ECDSA.

##### *1.3.3.3.2. Cân bằng Hiệu năng và Kích thước (Performance Balance)*

Tốc độ: Nhờ sử dụng NTT và lấy mẫu đều, Dilithium có tốc độ Xác minh (Verify) cực nhanh, nhanh hơn cả ECDSA và Falcon. Tốc độ Sinh khóa và Ký cũng rất ấn tượng, phù hợp cho các hệ thống yêu cầu độ trễ thấp (TLS handshakes).

Kích thước: Mặc dù lớn hơn ECC (Dilithium2 có chữ ký ~2.4KB so với 64 bytes của Ed25519), nhưng nó vẫn nhỏ hơn nhiều so với các hệ chữ ký dựa trên hàm băm (SPHINCS+ ~7.8KB) và đủ nhỏ để truyền tải qua gói tin TCP/IP thông thường mà không gây phân mảnh (fragmentation) nghiêm trọng.

##### *1.3.3.3.3. Khả năng mở rộng cho Ký đa bên (Suitability for MPC - Quan trọng cho luận văn)*

Đây là lý do chính khiến Dilithium được chọn cho đề tài này.

Do dựa trên cấu trúc tuyến tính của Mạng (Lattices)$(z = y + cs)$, Dilithium có tính chất cộng tính (additive homomorphism) tự nhiên.

Điều này giúp việc thiết kế các giao thức Ký ngưỡng (Threshold) và Ký gộp (Aggregation) dễ dàng và hiệu quả hơn nhiều so với các thuật toán dựa trên hàm băm (SPHINCS+) hoặc thuật toán có cấu trúc phức tạp như Falcon.

## 1.4. Khảo sát và phân tích các nghiên cứu tiêu biểu

Việc triển khai các giao thức ký đa bên trên nền tảng mật mã lưới (Lattice-based) gặp phải thách thức lớn hơn nhiều so với RSA hay ECDSA do đặc thù của kỹ thuật "Lấy mẫu loại bỏ" (Rejection Sampling) và sự phức tạp của việc quản lý nhiễu. Dưới đây là phân tích các công trình nền tảng và cập nhật nhất liên quan đến vấn đề này.

### *1.4.1. Nghiên cứu về ký ngưỡng từ mô hình lý thuyết Cozzo & Smart (2019) đến đánh giá thực nghiệm của Roux (2022)*

#### 1.4.1.1. Cơ sở lý thuyết và Đề xuất mô hình (Cozzo & Smart - 2019)

Trong công trình nghiên cứu mang tên *"Sharing the LUOV: Threshold Post-Quantum Signatures"*, Cozzo và Smart đã đặt nền móng lý thuyết quan trọng cho việc chuyển đổi các thuật toán ký số hậu lượng tử (PQC) sang dạng ký ngưỡng (threshold). Đối với thuật toán CRYSTALS-Dilithium, nhóm tác giả nhận định rằng thách thức lớn nhất nằm ở sự pha trộn giữa các phép toán tuyến tính và phi tuyến tính[16].

Để giải quyết vấn đề này, họ đề xuất một kiến trúc Tính toán đa bên an toàn (Secure Multi-Party Computation - MPC) lai ghép:

+ Sử dụng Mạch số học (Arithmetic Circuits) hay LSSS (Linear Secret Sharing Schemes) cho các phép toán tuyến tính (như phép nhân ma trận - vector).
+ Sử dụng Mạch rối (Garbled Circuits - GC) cho các phép so sánh phi tuyến tính (như kiểm tra chuẩn vô cùng hoặc lấy các bit cao/thấp).

Một đóng góp quan trọng của nghiên cứu là quy trình xử lý bước Lấy mẫu từ chối (Rejection Sampling) trong môi trường phân tán. Cozzo và Smart nhấn mạnh rằng các bên tham gia phải giữ bí mật tuyệt đối các giá trị trung gian (như vector $z$ hoặc $r_0$). Các giá trị này chỉ được công khai khi và chỉ khi phép thử Rejection Sampling thành công, nhằm ngăn chặn việc rò rỉ thông tin về khóa bí mật. Dựa trên

phân tích lý thuyết về số lượng cổng AND và chi phí chuyển đổi giữa LSSS và GC, nhóm tác giả ước lượng thời gian thực thi để tạo ra một chữ ký Dilithium ngưỡng là khoảng 12 giây[16].

#### 1.4.1.2. Hiện thực hóa và Đánh giá hiệu năng (Roux - 2022)

Trong luận văn thạc sĩ *"Implementation of a Threshold Post-Quantum Signature Scheme"* , Dominique Roux đã tiến hành hiện thực hóa mô hình lý thuyết của Cozzo và Smart sử dụng framework MP-SPDZ. Kết quả thực nghiệm đã chỉ ra khoảng cách lớn giữa dự báo lý thuyết và thực tế triển khai, đồng thời bộc lộ những rào cản kỹ thuật nghiêm trọng khi áp dụng MPC tổng quát (Generic MPC) cho Dilithium[3].

Cụ thể, các hạn chế chính được ghi nhận bao gồm:

- Hiệu năng thực tế thấp: Trái ngược với dự báo 12 giây của Cozzo và Smart, thời gian trung bình để sinh một chữ ký trong mô hình an toàn thụ động (semi-honest, honest majority) lên tới 87 giây. Nghiêm trọng hơn, trong mô hình an toàn chủ động (active/malicious security) với ngưỡng 2-out-of-2, thời gian này tăng vọt lên tới 41 phút, khiến giải pháp trở nên bất khả thi cho các ứng dụng thực tế đòi hỏi độ trễ thấp.
- Tiêu tốn tài nguyên phần cứng: Việc triển khai đòi hỏi tài nguyên hệ thống rất lớn. Mã nguồn sau khi biên dịch (bytecode) có kích thước lên tới 144 MiB và quá trình biên dịch yêu cầu hơn 14 GB RAM. Điều này đồng nghĩa với việc giải pháp không thể triển khai trên các thiết bị IoT, thiết bị di động hoặc các phần cứng có tài nguyên hạn chế[3].
- Độ trễ do đồng bộ hóa Rejection Sampling: Mặc dù số vòng lặp trung bình để tạo ra một chữ ký hợp lệ chỉ khoảng 1.4 vòng, phù hợp với lý thuyết, nhưng cơ chế đồng bộ hóa trạng thái giữa các bên sau mỗi lần lặp lại (restart) trong môi trường MPC đã gây ra độ trễ mạng tích lũy lớn.

Từ đó kết luận rằng nghiên cứu của Cozzo & Smart và thực nghiệm của Roux đã chứng minh rằng: mặc dù việc xây dựng chữ ký ngưỡng Dilithium dựa trên các kỹ thuật MPC tổng quát là khả thi về mặt toán học, nhưng lại chưa tối ưu về mặt hiệu năng cho các ứng dụng thực tiễn. Điều này đặt ra bài toán cần giải quyết cho đề tài này là tìm kiếm các phương pháp tối ưu hơn hoặc các mô hình ký gộp/ký độc lập để giảm thiểu chi phí tính toán và truyền thông.

### *1.4.2. Nghiên cứu về đa chữ ký hậu lượng tử của Hamidreza Rahmati và Farhad Rahmati*

Trong bối cảnh nhu cầu tối ưu hóa dung lượng lưu trữ và tốc độ xác thực cho các mạng lưới Blockchain — đặc biệt là Bitcoin — ngày càng trở nên cấp thiết, công trình của Hamidreza Rahmati và Farhad Rahmati (2024) với tiêu đề *"Quantum resistant multi-signature scheme with optimal communication round: A Blockchain-based approach"* đã đề xuất một lược đồ ký số đa bên hậu lượng tử mang tên Razhi-ms[17].

Razhi-ms là một lược đồ ký đa bên (Multi-signature) dựa trên mật mã lưới (Lattice-based cryptography), được thiết kế nhằm khắc phục các hạn chế cốt lõi của các mô hình ký đa bên hiện hữu, đặc biệt trong môi trường Blockchain.

#### 1.4.2.1. Tổng quan nghiên cứu và bài toán đặt ra

Nhóm tác giả chỉ ra ba thách thức chính cản trở việc áp dụng rộng rãi chữ ký đa bên trong kỷ nguyên hậu lượng tử:

- Mối đe dọa từ máy tính lượng tử: Các lược đồ chữ ký truyền thống dựa trên bài toán phân tích số nguyên hoặc logarit rời rạc (như RSA, ECDSA) được chứng minh là không an toàn trước thuật toán Shor. Điều này đòi hỏi sự chuyển dịch sang các bài toán khó trên lưới, tiêu biểu là Module-LWE (MLWE) và Module-SIS (MSIS)
- Độ trễ giao tiếp: Phần lớn các lược đồ ký đa bên hậu lượng tử hiện nay (ví dụ: MuSig-L, DualMS) yêu cầu ít nhất hai vòng tương tác giữa các bên tham gia

(commit–challenge), gây ra độ trễ đáng kể trong các ứng dụng Blockchain thời gian thực.

- Hiệu quả lưu trữ trên chuỗi: Một số lược đồ tối ưu về số vòng giao tiếp lại không hỗ trợ gộp khóa công khai (Aggregate Public Key – APK), buộc hệ thống phải lưu trữ danh sách đầy đủ các khóa công khai của người ký. Điều này làm tăng đáng kể chi phí lưu trữ *on-chain*.

#### 1.4.2.2. Đề xuất kỹ thuật và cơ chế hoạt động của Razhi-ms

Nhằm giải quyết đồng thời các hạn chế về số vòng giao tiếp, dung lượng lưu trữ và khả năng kháng lượng tử của các lược đồ ký đa bên hiện hữu, Rahmati và Rahmati (2024) đã đề xuất lược đồ Razhi-ms là một cơ chế ký số đa bên hậu lượng tử dựa trên mật mã lưới , được xây dựng trên nền tảng toán học của CRYSTALS-Dilithium và tối ưu hóa cho môi trường Blockchain[17].

##### *1.4.2.2.1. Nền tảng kỹ thuật và bộ tham số cấu hình*

Razhi-ms kế thừa cấu trúc toán học của Dilithium, dựa trên các bài toán khó Module-LWE và Module-SIS, kết hợp với kỹ thuật Fiat–Shamir with Aborts nhằm đảm bảo an toàn trước các tấn công lượng tử. Để đạt mức an toàn 128-bit (cổ điển) trong khi tối ưu kích thước chữ ký và khóa công khai gộp, tác giả sử dụng một bộ tham số được tinh chỉnh riêng, thể hiện trong Bảng 3 của bài báo[17].

**Table 3.** Proposed parameters for $Razhi - ms$ scheme

| Parameter | Description | Value |
|---|---|---|
| $n$ | Degree of $f(x)$ that is a power of 2 | 256 |
| $q$ | Prime modulus | 8397313 |
| $(k, l)$ | Dimensions of matrix $A$ | (4,4) |
| $\gamma_1$ | Coefficient range of y | $2^{17}$ |
| $\gamma_2$ | low-order rounding range | 63616 |
| $\tau$ | # of $\pm 1's$ in polynomial $c$ | 41 |
| $\eta$ | Private key range | 5 |
| $\beta$ | $\tau.\eta$ | 205 |
| $\lambda$ | Security parameter | 128 |

*Hình 1.5: Các tham số đầu vào của Razhi-ms*

Cụ thể, lược đồ hoạt động trên vành đa thức bậc $n = 256$, với module nguyên tố $q = 8{,}397{,}313$, khác biệt so với chuẩn FIPS 204 nhằm cải thiện biên an toàn trong quá trình tổng hợp chữ ký. Ma trận công khai $A$ có kích thước $(k, l) = (4{,}4)$, đảm bảo cân bằng giữa chi phí tính toán và độ an toàn.

Các tham số phân phối và làm tròn được lựa chọn gồm $\gamma_1 = 2^{17}$, điều khiển biên của vector ngẫu nhiên $y$; $\gamma_2 = 63{,}616$, dùng trong quá trình tách bit cao/thấp; trọng số thử thách $\tau = 41$, xác định số lượng hệ số $\pm 1$trong đa thức thử thách; giới hạn khóa bí mật $\eta = 5$; và tham số $\beta = \tau \cdot \eta = 205$, được sử dụng làm ngưỡng an toàn trong kiểm tra rejection sampling.

Việc lựa chọn bộ tham số này cho phép kích thước chữ ký và khóa công khai gộp xấp xỉ với một chữ ký đơn lẻ, đồng thời duy trì đầy đủ các chứng minh an toàn dựa trên MSIS và MLWE.

*1.4.2.2.2. Cơ chế ký đa bên một vòng*

Đóng góp kỹ thuật cốt lõi của Razhi-ms1 nằm ở việc loại bỏ hoàn toàn mô hình commit–challenge–response truyền thống vốn yêu cầu nhiều vòng tương tác. Thay vào đó, lược đồ áp dụng cơ chế mã hóa mầm ngẫu nhiên (seed encryption), cho phép tất cả các bên tham gia ký chỉ cần trao đổi một gói tin duy nhất.

*i. Giai đoạn tạo chữ ký thành phần*

Tại mỗi người ký $BS_i$, quá trình ký diễn ra độc lập như sau:

- Trước hết, người ký tự sinh một mầm ngẫu nhiên cục bộ $\rho_i'$có độ dài 256 bit. Thông qua hàm mở rộng ngẫu nhiên (XOF), mầm này được sử dụng để tạo ra vector ngẫu nhiên $y_i$. Từ thông điệp cần ký $m$ và giá trị $y_i$, người ký tính toán thử thách cục bộ $c_i$.
- Để các bên khác có thể tái tạo chính xác trạng thái ngẫu nhiên phục vụ tổng hợp chữ ký, người ký $BS_i$ mã hóa mầm $\rho_i'$ bằng khóa công khai

của từng người ký còn lại $BS_j$, tạo ra các cặp giá trị mã hóa $(u_{ij}, v_{ij})$. Đây là điểm khác biệt then chốt so với các lược đồ đa bên truyền thống.

- Sau đó, người ký tạo chữ ký thành phần: $z_i = y_i + c_i s_i$, và phát đi một gói tin duy nhất: $\sigma_i = (z_i, c_i, u_{ij}, v_{ij})$,đây cũng là bước giao tiếp duy nhất trong toàn bộ giao thức.

*ii. Giai đoạn giải mã và tổng hợp chữ ký*

Khi nhận được gói tin từ các bên khác, mỗi người ký hoặc nút tổng hợp sử dụng khóa bí mật của mình để giải mã các cặp $(u_{ij}, v_{ij})$, từ đó khôi phục lại mầm $\rho_i'$tương ứng. Nhờ đó, vector ngẫu nhiên $y_i$và thành phần $c_i s_i$của từng người ký có thể được tái tạo một cách nhất quán mà không cần thêm bất kỳ trao đổi nào. Chữ ký chung z được tổng hợp như sau:

$$\mathrm{z} = \mathrm{y} + \left( \bigotimes_{i=1}^{n} c_i \mathrm{s}_i \right) mod\, \beta$$

Tương ứng, khóa công khai gộp được xác định bởi: $APK = \mathrm{A}\left( \bigotimes_{i=1}^{n} c_i \mathrm{s}_i \right) mod\, \beta$ trong đó y được xác định bởi: $y = \sum_{i=1}^{n} y_i$, trong đó Trong đó $\otimes$ là phép nhân từng thành phần giữa các vector. Kết quả cuối cùng là chữ ký đa bên: $\sigma = (z, c)$, có cấu trúc và kích thước tương đương một chữ ký Dilithium đơn lẻ.

### 1.4.2.3. Nhận xét

Mặc dù Razhi-ms mang lại những cải tiến lý thuyết đáng kể về số vòng giao tiếp và kích thước dữ liệu khi kế thừa cấu trúc toán học của Dilithium, quá trình hiện thực hóa lược đồ này đối mặt với những thách thức sau:

- Không khớp với tham số nền tảng của Dilithium: Bài báo sử dụng module nguyên tố q = 8,397,313 để tối ưu hóa cho thiết kế riêng biệt của họ. Trong khi đó, chuẩn FIPS 204 của NIST đã cố định q = 8,380,417. Việc này có thể khiến việc sử dụng các thư viện mật mã chuẩn hiện nay trở nên vô dụng. Việc

phát triển mã nguồn không thể tái sử dụng các hàm nhân đa thức tốc độ cao (NTT) đã được tối ưu hóa mà buộc phải tự viết lại từ đầu, dẫn đến hiệu năng thấp và rủi ro lỗi cao.

- Trong bài báo công bố, nhóm tác giả chỉ tập trung so sánh về Kích thước (Khóa gộp ~3KB, Chữ ký ~2.2KB) và Số vòng giao tiếp (1 vòng). Hoàn toàn không có số liệu đo đạc cụ thể về thời gian ký (Signing time) và thời gian xác minh (Verification time) tính bằng mili-giây (ms) trên một cấu hình phần cứng tiêu chuẩn. Mặc dù giảm được thời gian chờ mạng (latency), nhưng Razhi-ms lại thêm vào các bước tính toán phức tạp cục bộ mà Dilithium gốc không có như việc mỗi người ký phải thực hiện thêm các phép nhân ma trận để mã hóa/giải mã tham số ngẫu nhiên cho N-1 người khác. Với nhóm ký lớn, chi phí này có thể tăng tuyến tính, làm chậm đáng kể.
- Giới hạn về phạm vi ứng dụng: Lược đồ Razhi-ms được thiết kế để tạo ra một chữ ký chung σ cho một thông điệp duy nhất $m$. Nó hoạt động theo cơ chế $(n, n)$ trên cùng một thông điệp. Điều này khác biệt hoàn toàn với Aggregate Signature cho phép gộp nhiều chữ ký trên nhiều tài liệu khác nhau thành một chữ ký duy nhất. Do đó, Razhi-ms chưa giải quyết được bài toán gộp chữ ký trong các kịch bản như Blockchain nơi cần xác thực nhiều giao dịch khác nhau cùng lúc.

### *1.4.3. Nghiên cứu về ký ngưỡng trên thiết bị IOT: Davydov & Bezzateev (2023)*

Tên công trình: *"Threshold Lattice-Based Signature Scheme for Authentication by Wearable Devices "* .

Sự bùng nổ của các thiết bị đeo thông minh (wearable devices) như đồng hồ, kính, vòng đeo tay sức khỏe đã mở ra nhu cầu xác thực người dùng đa yếu tố để tăng cường bảo mật. Mô hình xác thực ngưỡng cho phép người dùng chỉ cần sử dụng một tập hợp con $(t)$ các thiết bị trong tổng số $(n)$ thiết bị đang có để đăng nhập, mang lại sự tiện lợi và tính sẵn sàng cao[10].

#### 1.4.3.1. Tổng quan nghiên cứu và bài toán đặt ra

##### *1.4.3.1.1. Vấn đề cần giải quyết*

Mối đe dọa lượng tử: Các giải pháp xác thực hiện hành trên thiết bị đeo chủ yếu dựa trên mật mã đường cong Elliptic (ECC). Tuy nhiên, ECC dễ bị tổn thương trước thuật toán Shor trên máy tính lượng tử.

Hạn chế hiện tại:

+ Các lược đồ ký ngưỡng dựa trên lưới (Lattice-based) đã tồn tại nhưng thường gặp vấn đề như kích thước khóa quá lớn, hoặc thiếu tính linh hoạt về ngưỡng.
+ Thiết bị đeo có năng lực tính toán và pin thấp, đòi hỏi giải pháp phải là mật mã hạng nhẹ.

Mục tiêu nghiên cứu: Xây dựng một lược đồ chữ ký ngưỡng dựa trên lưới linh hoạt (t-of-n), an toàn trước máy tính lượng tử và đủ nhẹ để chạy trên các thiết bị đeo[10].

#### 1.4.3.2. Đề xuất kỹ thuật và cơ chế hoạt động

Bài báo đề xuất một lược đồ mới cải tiến từ công trình của Damgard (2022), kết hợp giữa mật mã lưới và chia sẻ bí mật Shamir.

*i. Cơ sở toán học:*

- Dựa trên tính khó của hai bài toán lưới module: Module Learning With Errors (M-LWE) và Module Short Integer Solution (M-SIS).
- Sử dụng mô hình Fiat-Shamir with Aborts (tương tự như thuật toán Dilithium chuẩn NIST) để đảm bảo tính an toàn và không lộ khóa bí mật qua chữ ký.
- Các tham số đầu vào tuân thủ hoàn toàn theo bộ tham số chuẩn của FIPS 204.

*ii. Kiến trúc giao thức quy trình hoạt động gồm 4 giai đoạn chính:*

- Thiết lập và Tạo khóa: Hệ thống sinh tham số công khai. Các thiết bị phối hợp để tạo ra một khóa công khai chung (*pk*) và một khóa bí mật (*sk*) chung, nhưng không thiết bị nào giữ toàn bộ khóa bí mật này.
- Chia sẻ bí mật (Secret Sharing): Áp dụng lược đồ Shamir's Secret Sharing. Khóa bí mật được chia nhỏ thành *n* các phần (shares) và phân phối cho thiết bị đeo. Việc khôi phục yêu cầu tối thiểu *t* phần[10].
- Tạo chữ ký (Signature Generation):
  + Khi người dùng cần xác thực, thiết bị sẽ tham gia.
  + Mỗi thiết bị tạo một cam kết (commitment) và tính toán một phần chữ ký (partial signature) dựa trên phần bí mật của mình.
  + Sử dụng kỹ thuật Rejection Sampling (lấy mẫu từ chối) để đảm bảo phân phối xác suất của chữ ký không tiết lộ thông tin về khóa bí mật.
  + Các phần chữ ký sau đó được gộp lại bằng nội suy Lagrange để tạo ra chữ ký cuối cùng.
  + Xác minh: Hệ thống kiểm tra tính hợp lệ của chữ ký dựa trên khóa công khai chung và thông điệp, đảm bảo rằng ít nhất thiết bị hợp lệ đã tham gia ký.

#### 1.4.3.3. Nhận xét và Đánh giá

Ưu điểm:

- Hiệu năng tính toán: Thử nghiệm thực tế cho thấy tốc độ thực thi rất nhanh. Ở mức bảo mật cao nhất (Level 5), tổng thời gian cho toàn bộ quy trình (tạo khóa, chia sẻ, ký, xác minh) chưa đến 40 ms. Điều này chứng minh tính khả thi cao khi triển khai trên các vi xử lý yếu của thiết bị đeo[10].
- Tính kháng lượng tử: Lược đồ cung cấp sự bảo vệ lâu dài trước các cuộc tấn công từ máy tính lượng tử, điều mà các giải pháp ECC hiện tại không làm được.

- Tính linh hoạt: Khắc phục được nhược điểm của lược đồ Damgard gốc bằng cách cho phép ngưỡng linh hoạt thay vì cố định , phù hợp với thực tế sử dụng (ví dụ: người dùng có thể quên 1 thiết bị ở nhà nhưng vẫn đăng nhập được).

Hạn chế và Thách thức:

- Kích thước dữ liệu lớn: Đây là điểm yếu cố hữu của mật mã lưới. Kích thước chữ ký (~11.7 KB) và dữ liệu truyền tải (~15.7 KB). Điều này có thể gây tốn băng thông Bluetooth/Wi-Fi và năng lượng truyền tin của thiết bị đeo.
- Rủi ro từ Shamir Secret Sharing: Lược đồ Shamir cơ bản không có tính năng xác minh (verifiable). Nếu một thiết bị bị tấn công và gửi phần chia sẻ sai, quá trình tạo chữ ký chung sẽ thất bại mà không dễ phát hiện nguồn lỗi.

### *1.4.4. Tổng hợp so sánh và xác định hướng triển khai*

***Bảng 1.2: Bảng tóm tắt các đặc điểm kỹ thuật của các nghiên***

| Tiêu chí so sánh | Mô hình MPC Tổng quát (Cozzo & Smart / Roux) | Mô hình Ký gộp (1-Round) (Razhi-ms 2024) | Mô hình Ký ngưỡng (t,n) (Leevik et al. 2023) |
|---|---|---|---|
| Kỹ thuật nền tảng | Generic MPC (Garbled Circuits) | Seed Encryption + Aggregation | Shamir Secret Sharing + Fiat-Shamir with Aborts |
| Mô hình ngưỡng | (t, n) | (n, n) (Cố định) | (t, n) (Linh hoạt) |
| Thời gian ký | ~87 giây (Rất chậm) | Trung bình | ~26.5 ms (Rất nhanh) |
| Số vòng tương tác | Nhiều vòng (phụ thuộc Rejection) | 1 vòng | 2 vòng |
| Kích thước chữ ký | Lớn (do cấu trúc mạch MPC) | Rất nhỏ: ~2.2 KB | Lớn: ~11.7 KB |
| Tính Chuẩn hóa | Tuân thủ tham số chuẩn | Sử dụng các tham số khác so với chuẩn FIPS 204 | Tuân thủ tham số chuẩn |
| Nhận định | Khả thi về lý thuyết nhưng không thực | Tối ưu nhất cho Blockchain nhờ kích | Phù hợp nhất cho IoT/Wearables nhờ tốc |

| | | | |
|---|---|---|---|
| | tế cho triển khai diện rộng. | thước nhỏ, nhưng khó triển khai do lệch chuẩn. | độ nhanh, dù tốn băng thông. |

Từ bảng so sánh, ta nhận thấy mô hình của Cozzo & Smart (2019) đã lỗi thời về mặt hiệu năng. Mô hình của Leevik (2023) cho thấy tiềm năng ứng dụng thực tế lớn nhờ tốc độ xử lý miligiây.

Đề án quyết định không triển khai mô hình Ký gộp (Razhi-ms) mà tập trung vào ký ngưỡng kết hợp với ký độc lập làm đối trọng so sánh. Các lý do cụ thể bao gồm:

- Bài toán Ký ngưỡng $(t, n)$ là bài toán tổng quát hơn so với Ký gộp $(n, n)$. Nếu triển khai thành công hệ thống ngưỡng $(t, n)$, hệ thống đương nhiên hỗ trợ trường hợp $t = n$ (tương đương ký gộp). Do đó, giá trị khoa học và phạm vi ứng dụng của đề án sẽ rộng hơn (áp dụng được cho cả Blockchain, IoT, và quản trị doanh nghiệp).
- Mô hình Ký ngưỡng theo hướng tiếp cận của Leevik et al. cho phép sử dụng lại các nguyên thủy mật mã (cryptographic primitives) đã được kiểm chứng của chuẩn FIPS 204 (ML-DSA). Việc này giảm thiểu rủi ro sai sót toán học và lỗi bảo mật so với việc tự cài đặt lại hệ thống tham số lạ của Razhi-ms.
- Có thể xem xét và thử nghiệm việc tích hợp kỹ thuật nén dữ liệu của FIPS 204 vào mô hình ký ngưỡng, giúp giải quyết bài toán kích thước chữ ký mà không cần hy sinh tính chuẩn hóa.
- Việc so sánh giữa ký ngưỡng và ký độc lập sẽ làm nổi bật sự đánh đổi giữa: *Kích thước chữ ký/Băng thông* và *Tài nguyên tính toán CPU* (Ký độc lập có thể nhẹ hơn ở phía người ký nhưng nặng hơn ở phía xác thực).

# Chương 2. PHƯƠNG PHÁP NGHIÊN CỨU VÀ THIẾT KẾ MÔ HÌNH THỬ NGHIỆM

## 2.1.Mục tiêu và yêu cầu của thử nghiệm

Để giải quyết các khoảng trống nghiên cứu đã xác định ở Chương 1, đặc biệt là sự thiếu hụt các đánh giá thực nghiệm toàn diện về hiệu năng của Dilithium trong môi trường đa bên, quá trình thử nghiệm cần đạt được các mục tiêu và yêu cầu cụ thể sau.

### *2.1.1. Mục tiêu: Đo lường các chỉ số hiệu năng.*

Mục tiêu cốt lõi của phần thực nghiệm là cung cấp số liệu định lượng chính xác để đánh giá hiệu quả của mô hình Ký ngưỡngđược xây dựng dựa trên tiếp cận của Leevik et al. (2023) và chuẩn FIPS 204 thông qua việc đối sánh với mô hình Ký độc lập (Independent Signing). Các chỉ số cần đo lường bao gồm:

- Đo lường chi phí tính toán:
  - + Xác định thời gian thực thi trung bình cho các pha: Sinh khóa, Ký và Xác minh.
  - + Đánh giá sự chênh lệch thời gian xử lý giữa mô hình đơn lẻ (Standard Dilithium) và mô hình đa bên để làm rõ chi phí cộng thêm của các thuật toán chia sẻ bí mật và cộng gộp.
- Đo lường chi phí truyền thông :
  - + Đây là mục tiêu quan trọng nhất nhằm kiểm chứng các nhận định lý thuyết về nhược điểm băng thông của mật mã lưới.
  - + Đo tổng dung lượng dữ liệu (tính bằng Kilobytes) cần truyền tải/lưu trữ để hoàn tất một phiên ký thành công.
  - + Kiểm tra tính hiệu quả của kỹ thuật nén dữ liệu (Bit-packing) khi áp dụng vào chữ ký ngưỡng.

+ Đếm tổng số vòng tương tác cần thiết để hoàn thành giao thức, đặc biệt chú trọng vào số vòng lặp phát sinh do cơ chế *Rejection Sampling* phân tán.

– Đánh giá khả năng mở rộng:

+ Phân tích sự biến thiên của hiệu năng khi số lượng bên tham gia ($N$) tăng lên (ví dụ: $N = 3, 5, 10, 20$).
+ Xác định ngưỡng chịu đựng của hệ thống (điểm uốn mà tại đó độ trễ mạng trở nên không thể chấp nhận được cho các ứng dụng thực tế).

### *2.1.2. Yêu cầu: Đảm bảo tính chính xác, khách quan, có thể lặp lại.*

Để kết quả thực nghiệm có giá trị khoa học và tham khảo cao, quá trình thiết kế và triển khai phải tuân thủ nghiêm ngặt các nguyên tắc sau:

– Tính chính xác và Tin cậy:

+ Các phép đo thời gian được thực hiện với miligiây(ms).
+ Kết quả phải được lấy trung bình từ một số lượng lớn lần chạy thử nghiệm (ví dụ: 100 hoặc 1.000 lần ký) để loại bỏ các yếu tố nhiễu ngẫu nhiên của hệ điều hành và phương sai của quá trình *Rejection Sampling*.
+ Phải kiểm tra tính đúng đắn sau mỗi lần chạy: Chữ ký tạo ra phải vượt qua hàm Verify chuẩn.

– Tính khách quan và Công bằng:

+ Môi trường đồng nhất: Tất cả các mô hình phải được triển khai và đo đạc trên cùng một nền tảng phần cứng, hệ điều hành và phiên bản trình biên dịch.
+ Thư viện cơ sở: Sử dụng chung một thư viện lõi (ví dụ: liboqs hoặc PQClean) cho các phép toán cơ sở (NTT, SHAKE) để đảm bảo so sánh công bằng về mặt thuật toán giao thức, tránh sự chênh lệch do tối ưu hóa code ở tầng thấp (low-level optimization).

– Khả năng lặp lại:

+ Bộ tham số thử nghiệm phải được công bố rõ ràng, tuân theo các mức độ an toàn chuẩn của NIST (Dilithium2, Dilithium3, Dilithium5).
+ Mã nguồn thử nghiệm, kịch bản đo và dữ liệu đầu vào giả lập cần được đóng gói hoặc mô tả chi tiết để các nghiên cứu sau có thể tái lập kết quả.

## 2.2. Bộ tiêu chí đánh giá hiệu năng

Để đảm bảo tính khách quan và khả năng so sánh giữa các mô hình (Ký ngưỡng, Ký gộp, Ký độc lập), đề tài sử dụng bộ tiêu chí gồm 4 nhóm chỉ số chính:

### *2.2.1. Hiệu năng Tính toán*

Nhóm này đo lường tốc độ xử lý của CPU, ảnh hưởng trực tiếp đến độ trễ của hệ thống.

- Thời gian sinh khóa :
  + Ký ngưỡng: Đo thời gian thực hiện giao thức *Sinh khóa phân tán (Distributed Key Generation - DKG)*. Thời gian tính từ lúc bắt đầu đến khi mỗi bên nhận được mảnh khóa bí mật ($sk_i$) và khóa công khai chung ($pk$).
  + Ký độc lập: Đo thời gian sinh cặp khóa $(pk, sk)$ đơn lẻ chuẩn.
  + *Đơn vị:* Miligiây (ms).
- Thời gian ký:
  + Ký ngưỡng: Thời gian trung bình ký thông điệp của 1 bên và tổng thời gian từ khi nhận thông điệp đến khi tạo ra chữ ký hợp lệ (bao gồm cả thời gian tính toán cục bộ và thời gian chờ đồng bộ giữa các bên).
  + Ký độc lập: Thời gian trung bình để tạo một chữ ký đơn lẻ.
  + *Đơn vị:* Miligiây (ms).
- Thời gian Xác minh:

+ Đo thời gian xác minh một chữ ký cuối cùng khi đủ ngưỡng (Batch Verification). Chỉ số này cần so sánh với tổng thời gian xác minh $N$ chữ ký độc lập để chứng minh hiệu quả.
+ Chứng minh rằng thời gian xác minh một chữ ký ngưỡng nhỏ hơn nhiều so với tổng thời gian xác minh $N$ chữ ký độc lập. *Mục tiêu:* $T_{Verify_Agg} < N \times T_{Verify_Single}$.

### *2.2.2. Hiệu năng Truyền thông*

Đây là nhóm chỉ số quan trọng đối với đề tài, vì kích thước dữ liệu trao đổi lớn, dễ gây nghẽn mạng.

– Dung lượng dữ liệu trao đổi:
  + Định nghĩa: Tổng kích thước (tính bằng KB) của tất cả các gói tin (Commitments, Shares, Partial Signatures) được gửi đi giữa các bên để hoàn thành một chữ ký.
  + *Tham chiếu:* Nghiên cứu của Leevik (2023) cho thấy chỉ số này lên tới ~15KB/user, cần đo đạc lại.

### *2.2.3. Hiệu quả Lưu trữ*

– Kích thước chữ ký:
  + Ký ngưỡng: Kích thước chữ ký cuối cùng ~11.5 KB theo bài báo công bố.
  + Ký độc lập: Tổng kích thước của $N$ chữ ký riêng biệt.
– Kích thước Khóa công khai (pk): So sánh kích thước của một Khóa công khai chung $(PK_{group})$ so với việc phải lưu trữ danh sách $N$ khóa công khai rời rạc $(PK_1, PK_2, \dots, PK_N)$.

### *2.2.4. Độ ổn định và Mở rộng*

– Số vòng lặp Rejection Sampling :
  + Định nghĩa: Số vòng lặp trung bình, nếu một bên (signer) tạo ra chữ ký vi phạm biên độ an toàn, toàn bộ giao thức phải khởi động lại (Restart).

Bài nghiên cứu của Leevik ghi nhận tỷ lệ chấp nhận khoảng 20-50% (tương đương cần 2-5 lần thử để thành công), số vòng lặp càng cao tương đương với việc bắt buộc user phải tương tác với lược đồ ký nhiều lần để được chấp nhận. Chỉ số này càng thấp thì hệ thống càng ổn định.

+ *Mục tiêu:* Kiểm chứng xem mô hình triển khai có đạt được con số tối ưu được công bố là 2-5 vòng lặp hay không? Đây là thước đo quan trọng về độ ổn định của hệ thống phân tán, khác biệt hoàn toàn so với ký đơn (nơi việc lặp lại chỉ diễn ra cục bộ).

– Khả năng mở rộng (Scalability Factor): Phân tích xu hướng tăng của thời gian ký và dung lượng dữ liệu khi số lượng người tham gia tham gia ($N$) tăng dần (ví dụ: thử nghiệm với $N = 3, 5, 10, 20$). Mục tiêu là xác định giới hạn chịu đựng của hệ thống.

## 2.3. Thuyết minh mô hình ký ngưỡng (Threshold)

### *2.3.1. Phân tích giao thức.*

Giao thức là sự kết hợp giữa độ khó của bài toán M-LWE/M-SIS trên lưới và tính chất bảo mật thông tin của lược đồ chia sẻ bí mật Shamir.

#### 2.3.1.1. Pha Thiết lập và Chia sẻ khóa

Để đảm bảo tính đơn giản và tập trung vào hiệu năng ký, đề tài sử dụng mô hình Đại lý tin cậy (Trusted Dealer) cho giai đoạn khởi tạo ( theo thiết kế của Leevik):

– Sinh khóa chủ: Dealer thực hiện quy trình sinh khóa Dilithium tiêu chuẩn để tạo cặp khóa $(pk, sk)$, Mỗi người ký $P_i$ phải lưu trữ hai loại bí mật:

+ $s_i$ (Small Secret): Giữ lại từ lúc sinh khóa ban đầu. Dùng để kiểm tra điều kiện an toàn: .
+ $x_i$ (Large Share): Được tạo thành bằng phân rã bí mật (Shamir's Secret Sharing) bằng cách xây dựng một đa thức ngẫu nhiên $f(x) = \mathrm{s}_1 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$, trên một trường hữu hạn phù hợp, sao cho $f(0) = \mathrm{s}_1, sau$ đó gửi mảnh chia sẻ $f_i(\mathrm{uid}_i)$cho các bên khác. Bên

$P_i$ tổng hợp tất cả các mảnh nhận được để tạo Large Share:$x_i = \sum_{j=1}^{n} f_j\,(\mathrm{uid}_i)$.Trong đó $\mathrm{uid}_i$là định danh duy nhất của $P_i$.

- Kết quả:
  + Không bên nào (ngoại trừ Dealer tại thời điểm khởi tạo) nắm giữ toàn bộ khóa bí mật, sau khi pha kết thúc khóa cũng bị tiêu hủy.
  + Chỉ khi có tối thiểu $t$ bên hợp tác thì khả năng ký hợp lệ mới được khôi phục.
  + Khóa công khai $pk$ được công bố cho tất cả các bên và người xác minh.

#### 2.3.1.2. Pha Ký số phân tán

Quy trình ký diễn ra qua 2 vòng tương tác chính giữa các bên tham gia và một nút đóng vai trò tổng hợp:

- Vòng 1: Cam kết: Mỗi bên sẽ tính toán tạo ra một cam kết và gửi cam kết để tổng hợp một thử thách chung(Challenge) duy nhất cho toàn phiên ký này
- Vòng 2: Dựa trên thử thách chung vừa nhận + mảnh khóa bí mật $sk_i$ đang nắm giữ, mỗi bên tạo ra phần chữ ký của riêng mình. Trước khi gửi đi, mỗi bên tự kiểm tra biên độ dữ liệu của phần chữ ký vừa tạo:
  + Nếu dữ liệu vi phạm các ngưỡng an toàn hoặc có nguy cơ làm lộ thông tin thống kê về khóa bí mật, quá trình ký sẽ bị hủy và phải khởi động lại từ đầu.
  + Nếu kiểm tra hợp lệ, các bên gửi phần chữ ký của mình cho nút Tổng hợp.
- Nút Tổng hợp thực hiện phép cộng tuyến tính tất cả các phần chữ ký nhận được để tạo thành vector chữ ký tổng.

#### 2.3.1.3. Hiện thực hóa các Tham số và Hàm chuẩn FIPS 204

Để đảm bảo mô hình ký ngưỡng không chỉ hoạt động đúng về mặt logic chia sẻ bí mật mà còn tuân thủ các tiêu chuẩn an toàn, quá trình triển khai hệ thống được

xây dựng bám sát đặc tả kỹ thuật của FIPS 204 (ML-DSA). Việc hiện thực hóa được chia thành hai tầng: tầng tham số cấu hình và tầng hàm chức năng toán học.

*i. Cấu hình Tham số An toàn*

Hệ thống được thiết lập để hoạt động ở mức độ an toàn ML-DSA-44 (tương đương NIST Security Level 2) hoặc ML-DSA-65 (Level 3), tùy biến thông qua file cấu hình *dilithium_math.py*. Các hằng số này được định nghĩa chính xác theo chuẩn để đảm bảo tính tương thích với các bộ kiểm tra công khai:

Trường hữu hạn : Sử dụng module nguyên tố $q = 8{,}380{,}417$. Đây là tham số cốt lõi cho phép sử dụng thuật toán nhân đa thức nhanh NTT (Number Theoretic Transform).

Bậc đa thức : Cố định $N = 256$. Mọi phép toán chia sẻ bí mật Shamir đều được thực hiện trên 256 hệ số của các đa thức này.

Giới hạn biên độ:

+ $\gamma_1 = 2^{19} = 524{,}288$: Dùng để kiểm soát biên độ của vector nhiễu $y$.
+ $\eta = 2$ hoặc 4: Dùng để sinh khóa bí mật và nhiễu $e$ nhỏ.
+ $\tau = 39$ (Level 2) hoặc 49 (Level 3): Số lượng hệ số $\pm 1$ trong đa thức thử thách $c$.

*ii. Các Hàm chức năng Tiêu chuẩn*

Thay vì tự cài đặt lại các phép toán cơ sở (vốn dễ mắc lỗi và hiệu năng thấp), hệ thống tích hợp các hàm chuẩn của thư viện liboqs / PQClean thông qua cơ chế wrapper (Ctypes), kết hợp với logic điều khiển bằng Python:

Biến đổi Số học (NTT & Inverse NTT):

+ Sử dụng hàm *poly_ntt()* và *poly_invntt_tomont()* được tối ưu hóa bằng C/Assembly.

+ Việc này giúp giảm độ phức tạp của phép nhân đa thức – thao tác tốn kém nhất trong quá trình nội suy Lagrange – từ $O(N^2)$ xuống $O(N\ log\ N)$ đảm bảo thời gian ký ngưỡng chỉ ở mức miligiây.

Sinh Ma trận từ Seed:

+ Thay vì lưu trữ và truyền tải toàn bộ ma trận $A$ (kích thước lớn), hệ thống triển khai hàm ExpandA() theo chuẩn FIPS 204.
+ Đầu vào là một seed $\rho$ (32 bytes). Sử dụng hàm băm SHAKE-128 để bung ra ma trận đa thức giả ngẫu nhiên. Điều này giúp giảm băng thông truyền tải trong pha thiết lập khóa xuống hơn 99%.

### *2.3.2. Lưu đồ thuật toán.*

#### 2.3.2.1. Lưu đồ Sinh khóa phân tán (Distributed Key Generation – DKG)

Quy trình sinh khóa phân tán nhằm đảm bảo rằng không tồn tại bất kỳ thực thể đơn lẻ nào nắm giữ toàn bộ khóa bí mật $S$. Mỗi bên chỉ giữ một phần khóa (key share), và chỉ khi có đủ ngưỡng $t$bên hợp tác thì mới có thể thực hiện ký hợp lệ.

– Bước 1: Khởi tạo: Các bên tham gia $P_1, \dots, P_n$ thống nhất bộ tham số chung của hệ mật mã lưới $(q, N, k, l)$và seed công khai $\rho$. Từ $\rho$, tất cả các bên có thể tái tạo nhất quán ma trận công khai $A \in R_q^{k\times l}$ bằng cách sử dụng hàm expand_a() trong thư viện chuẩn FIPS 204. Mỗi bên $P_i$ độc lập tạo các vector bí mật $s_i$ từ phân phối Gaussian rời rạc, và vector nhiễu $e_i$, theo phân phối xác định trước. $s_i$ được $P_i$ giữ bí mật tuyệt đối.
– Bước 2: Tính khóa công khai phần (Partial Public Key): Mỗi bên tính khóa công khai cục bộ:

$$t_i = A \cdot s_i + e_i (\mathrm{mod} q) (2.1)$$

– Bước 3: Chia sẻ bí mật (Shamir Secret Sharing):

+ Để phân tán khóa bí mật, mỗi bên $P_i$ áp dụng lược đồ chia sẻ bí mật Shamir lên $s_i$, tạo ra $n$ mảnh chia sẻ bằng cách xây dựng một đa thức

ngẫu nhiên $f(x) = s_1 + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1}$, trên một trường hữu hạn phù hợp, sao cho $f(0) = s_1, sau$ đó gửi mảnh chia sẻ $f_i(\text{uid}_i)$cho các bên khác. Bên $P_i$ tổng hợp tất cả các mảnh nhận được để tạo Large Share:$x_i = \sum_{j=1}^{n} f_j(\text{uid}_i)$.Trong đó $\text{uid}_i$là định danh duy nhất của $P_i$.

– Bước 4: Tổng hợp khóa:

+ Khóa công khai chung:

$$A, T = \sum_{i=1}^{n} t_i.$$

+ Khóa bí mật phần của bên $P_i$: $sk_i(x_i, s_i)$.

Cuối cùng hệ thống thu được khóa công khai chung $pk = T$ và tập các khóa bí mật phần $\{sk_i\}$. Không bên nào có thể suy ra khóa bí mật toàn cục.

#### 2.3.2.2. Lưu đồ Ký ngưỡng

Ký ngưỡng là giai đoạn phức tạp của hệ thống, yêu cầu tối thiểu $t$ bên tham gia và bao gồm 4 vòng tương tác, kết hợp cơ chế Fiat–Shamir with Aborts (Rejection Sampling) để đảm bảo an toàn.

Đầu vào là thông điệp cần ký $M$ và tập con gồm $t$ người ký hợp lệ.

– Bước 1: Cam kết (Commitment)

+ Mỗi bên $P_i$ sẽ sinh vector nhiễu ngẫu nhiên $y_i, r_i$ và tính: $w_i = A \cdot y_i$.

+ Tạo cam kết: $com_i = \text{Commit}(w_i, r_i)$, và gửi $com_i$ cho các bên còn lại (hoặc cho bên điều phối).

– Bước 2: Sinh thử thách

+ Sau khi nhận đủ cam kết, tổng hợp: $com = \sum_{i=1}^{t} com_i$.

+ Sinh thử thách toàn cục: $c = H_0(com \parallel M \parallel pk)$,trong đó $c$ là một đa thức được lấy mẫu theo hàm SampleInBall với trọng số Hamming nhỏ.

– Bước 3: Phản hồi và kiểm tra cục bộ (Response)

+ Người ký $P_i$sử dụng $s_i$ để kiểm tra biên chuẩn: $z_i' = y_i + c \cdot s_i$, chỉ khi thỏa mãn:$\| z_i' \| < \beta$.

+ Sau khi vượt qua bước kiểm tra, người ký sử dụng bí mật lớn $x_i$để tính thành phần chữ ký: $z_i = y_i + c \cdot x_i$ Giá trị $z_i$ có độ lớn rất lớn, không thực hiện kiểm tra biên, và được gửi trực tiếp tới nút tổng hợp. Các bên nhận sẽ thực hiện kiểm tra bằng cách tính lại giá trị băm: $H(\mathrm{z_i} \parallel r_\mathrm{i}) = ? h_\mathrm{i}$ nhằm xác thực tính toàn vẹn của dữ liệu nhận được.

– Bước 4: Tổng hợp chữ ký (Aggregation)

+ Tại phần điều phối, khi nhận đủ $t$ chữ ký thành phần $z_j$ cần tổng hợp lại:

$$z = \sum_{j=1}^{t} \lambda_i z_j.$$

Và cộng dồn các $r$ cục bộ để tạo ra chìa khóa mở cam kết tổng:

$$r = \sum_{j=1} r_j$$

+ Rejection Sampling toàn cục sẽ kiểm tra: $\| z \|_\infty < t \cdot \mathrm{Bound}$. Nếu vi phạm sẽ Restart toàn bộ quy trình.
+ Nén chữ ký: Dùng thuật toán Bit-Packing để nén vector $z$ (vốn là các số nguyên lớn) thành chuỗi bytes gọn nhẹ $z_{\mathrm{packed}}$.

$$z_{\mathrm{packed}} = \mathrm{BitPack}(z).$$

+ Đầu ra: Chữ ký ngưỡng cuối cùng:

$$\sigma = (z_{packed}, c, com, r).$$

2.3.2.3. Lưu đồ Xác minh chữ ký

Quy trình xác minh tuân thủ chuẩn FIPS 204 (Dilithium) và không phân biệt chữ ký được tạo bởi một người hay bởi nhiều người theo mô hình ngưỡng.

Giải nén: Từ chữ ký $\sigma = (z_{packed}, c, com, r)$và khóa công khai *T*, giải nén $z_{packed}$ để thu được vector $z$. Kiểm tra biên, nếu: $\| z \| \geq t \cdot$ Bound, thì chữ ký không hợp lệ. Tái tạo tham số công khai bằng cách sử dụng seed $\rho$ để dựng lại ma trận $A$.

Tính toán giá trị kiểm tra:

$$w = A \cdot z - c \cdot T.$$

Kiểm tra $w$ có đúng là giá trị được cam kết trong *com* hay không qua hàm *commit_opening*$(w, r, com)$ mục đích là tái tạo *com_new* và so sánh với *com*, nếu khớp thì tiếp tục với hàm băm:

$$c' = SampleInBall(H(M \parallel w \,||\, pk)).$$

So sánh:

+ Nếu $c' = c$, chúng ta có chữ ký hợp lệ (Valid).
+ Ngược lại thì không hợp lệ (Invalid).

### *2.3.3. Cấu trúc dữ liệu.*

Hệ thống được thiết kế hướng đối tượng (OOP) với các cấu trúc dữ liệu chính ánh xạ trực tiếp từ toán học mật mã lưới sang Python objects.

*i. Đối tượng Đa thức*

Đây là cấu trúc dữ liệu nguyên tử (atomic data structure) của hệ thống.

Class: Poly, thuộc tính:

+ coeffs: Mảng số nguyên (Integer Array) độ dài $N = 256$. Chứa các hệ số thuộc $Z_q$.

+ in_ntt: Boolean. Đánh dấu đa thức đang ở miền hệ số hay miền tần số (NTT domain) để tránh tính toán sai.

– Phương thức chính: add(), sub(), mul_ntt(), to_bytes(), from_bytes().

*ii. Khóa Công khai*

Lưu trữ thông tin cần thiết để xác minh chữ ký. Để tối ưu băng thông, ma trận *A* không được lưu trực tiếp.

– JSON:

```
{
 "scheme": "dilithium-threshold-fips204",
 "rho": "Base64_String",    // 32 bytes seed để sinh ma trận A
 "t1": "Base64_List",       // Vector t1 (đã nén)
 "k": 6,                    // Tham số kích thước ma trận (Dilithium3)
 "l": 5,                    // Tham số kích thước ma trận
}
```

*iii.Mảnh Khóa Bí mật*

Mỗi bên tham gia lưu giữ một bản ghi này. Nó chứa phần bí mật của riêng họ sau khi đã chạy Shamir Secret Sharing.

– JSON

```
{
 "party_id": 1,             // Định danh người dùng (Index x trong Shamir)
"small_s1": [Poly_Obj],     // Vector L
 "small_s2": [Poly_Obj],    // Vector K
 "large_x1": [Poly_Obj],    // Shamir share của s1
 "large_x2": [Poly_Obj],    // Shamir share của s2
```

"threshold": 3 // Ngưỡng t yêu cầu

}

*iv. Chữ ký Số*

Cấu trúc chữ ký cuối cùng, được tối ưu hóa kích thước bằng Bit-packing.

- Tổng kích thước dự kiến: ~4 KB (so với 11.7 KB nếu không nén).
- Thành phần:
  - + Challenge ($c$): ~32 Bytes (Seed) hoặc ~2.5KB (nếu để dạng Poly thô). Trong cài đặt này dùng dạng Seed hoặc Poly nén.
  - + Response ($z$): Chiếm phần lớn dung lượng. Là vector gồm $L$ đa thức. Mỗi hệ số được nén (Bit-packed) từ 32-bit xuống còn ~17-20 bit.
  - + Hints ($h$): (Tùy chọn) Vector gợi ý để khôi phục phần cao của $w$, giúp giảm kích thước chữ ký hơn nữa (theo chuẩn FIPS 204).

## 2.4. Thuyết minh mô hình ký độc lập

Mô hình này được thiết kế như một nhóm đối chứng. Thay vì áp dụng các kỹ thuật chia sẻ bí mật hay gộp chữ ký phức tạp, mô hình thực hiện ký tuần tự truyền thống, nơi chữ ký của mỗi cá nhân được tạo và lưu trữ độc lập.

### *2.4.1. Phân tích giao thức.*

2.4.1.1. Giai đoạn Khởi tạo

Không tồn tại thực thể trung tâm. Mỗi người ký $S_i$ độc lập sinh cặp khóa riêng của mình: $(pk_i, sk_i)$ theo chuẩn thuật toán CRYSTALS-Dilithium. Các khóa công khai $pk_i$ được công bố để phục vụ cho quá trình xác minh sau này.

2.4.1.2. Giai đoạn Ký

+ Thông điệp hoặc tài liệu cần ký $M$được chuyển tuần tự qua danh sách người ký đã định trước:

$$L = S_1, S_2, \dots, S_N.$$

+ Người ký đầu tiên $S_1$sử dụng khóa bí mật $sk_1$để tạo chữ ký $\sigma_1$trên $M$. Người ký thứ $i(i > 1)$ nhận gói tin: $(M, \sigma_1, \dots, \sigma_{i-1})$,sau đó tạo chữ ký của riêng mình: $\sigma_i = \text{Sign}(sk_i, M)$,và chuyển tiếp gói tin: $(M, \sigma_1, \dots, \sigma_i)$ cho người ký tiếp theo.
+ Sau khi hoàn tất vòng ký, kết quả thu được là tập hợp chữ ký riêng biệt:

$$\Sigma = \sigma_1, \sigma_2, \dots, \sigma_N.$$

#### 2.4.1.3. Giai đoạn Xác minh

Quá trình xác minh được thực hiện một cách tuần tự:

– Với mỗi chữ ký $\sigma_i \in \Sigma$, hệ thống sử dụng khóa công khai tương ứng $pk_i$ để kiểm tra tính hợp lệ.
– Văn bản $M$ chỉ được chấp nhận nếu toàn bộ $N$ chữ ký đều hợp lệ.

#### 2.4.1.4. Đặc điểm kỹ thuật

Ưu điểm:

+ Cấu trúc đơn giản, dễ triển khai trong thực tế.
+ Không yêu cầu sửa đổi thuật toán Dilithium gốc.
+ Không cần cơ chế chia sẻ khóa hay kênh truyền bảo mật giữa các người ký.

Nhược điểm:

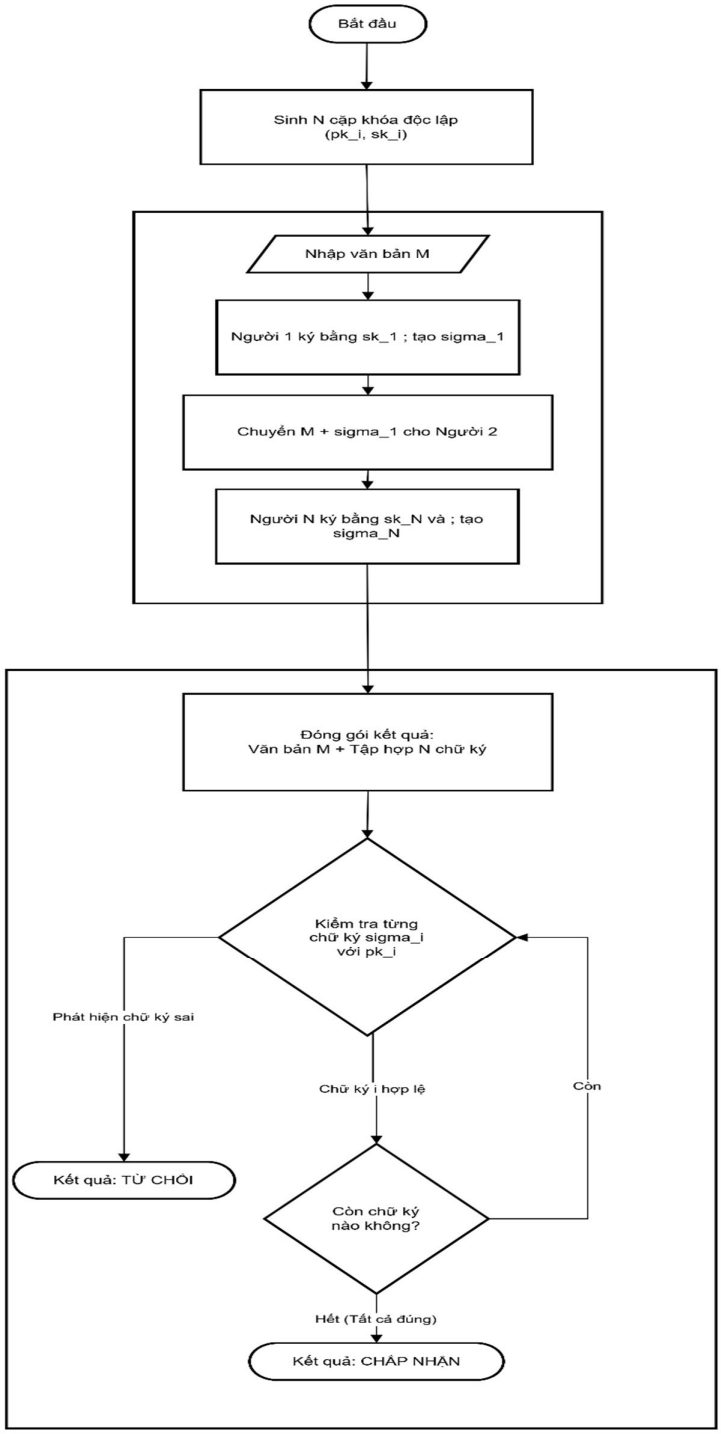
+ Kích thước dữ liệu chữ ký tăng tuyến tính theo số lượng người ký:

$$| \Sigma | = \mathcal{O}(N).$$

+ Thời gian xác minh cũng tăng tuyến tính theo $N$, gây tốn tài nguyên tính toán và lưu trữ. Không phù hợp với các hệ thống Blockchain hoặc môi trường on-chain, nơi chi phí lưu trữ và xác minh là yếu tố quan trọng, đặc biệt khi số lượng người ký lớn.

### *2.4.2. Lưu đồ thuật toán.*

Do tính chất tuần tự, lưu đồ thuật toán được thiết kế theo dạng tuyến tính như sau:




*Hình 2.1: Lưu đồ thuật toán ký độc lập*

## 2.5.Môi trường và công cụ thử nghiệm

Để đảm bảo tính khách quan và khả năng tái lập của kết quả nghiên cứu, hệ thống thử nghiệm được thiết lập trên môi trường máy chủ (server-grade) với các thông số kỹ thuật và công cụ phần mềm cụ thể như sau.

### *2.5.1. Cấu hình phần cứng, hệ điều hành.*

Thử nghiệm được triển khai trên một node tính toán hiệu năng cao với cấu hình chi tiết:

Bộ xử lý trung tâm (CPU):

+ Model: Intel® Xeon® E5-2697 v4.
+ Thông số: 18 nhân vật lý, 36 luồng (Threads), xung nhịp cơ bản 2.30 GHz (Turbo Boost lên 3.60 GHz).
+ Kiến trúc: x86_64, hỗ trợ tập lệnh AVX2 (Advanced Vector Extensions 2). Đây là yếu tố quan trọng giúp tăng tốc các phép toán vector hóa và nhân đa thức trong thuật toán lưới (Lattice-based).

Bộ nhớ trong (RAM): Dung lượng: 62 GB DDR4.

Hệ điều hành:

+ OS: Ubuntu 22.04.5 LTS (Jammy Jellyfish).
+ Kernel: Linux 6.8.0-65-generic.

### *2.5.2. Ngôn ngữ lập trình.*

Đề án sử dụng Python (phiên bản 3.10.12) làm ngôn ngữ chính để phát triển và kết nối các thành phần của hệ thống.

Lý do lựa chọn: Python có hệ sinh thái mạnh mẽ cho tính toán khoa học, cú pháp rõ ràng giúp mô phỏng nhanh các giao thức phức tạp.

### *2.5.3. Thư viện sử dụng.*

Các thư viện mã nguồn mở đóng vai trò cốt lõi trong việc triển khai các thuật toán mật mã và đo lường hiệu năng:

- NumPy (v2.2.6): Sử dụng để biểu diễn và thao tác trên các vector đa thức thuộc vành $R_q$.
- Numba (v0.62.1):
  + Dùng để biên dịch các hàm tính toán cốt lõi của Dilithium (như phép nhân đa thức, lấy mẫu Rejection Sampling) thành mã máy (machine code) ngay tại thời điểm chạy (runtime).
  + Giúp loại bỏ overhead của trình thông dịch Python trong các vòng lặp tính toán dày đặc.
- Hashlib & Cryptography: Cung cấp các hàm băm chuẩn SHAKE-128 và SHAKE-256 (thuộc họ SHA-3) cần thiết cho việc mở rộng ma trận $A$ và tạo các thử thách (challenge) ngẫu nhiên an toàn.
- Ctypes:
  + Sử dụng để gọi các thư viện liên kết động (.so) được viết bằng C (ví dụ: libdilithium_ntt.so) nhằm tối ưu hóa phép biến đổi NTT, đảm bảo tính chính xác theo chuẩn FIPS 204.

# Chương 3.THỰC NGHIỆM, PHÂN TÍCH KẾT QUẢ VÀ KẾT LUẬN

## 3.1.Thiết kế thực nghiệm

### *3.1.1. Đối tượng và Kịch bản*

Hệ thống thử nghiệm được triển khai để đánh giá hiệu năng của ba chiến lược ký số trên nền tảng CRYSTALS-Dilithium:

- Ký Độc lập (Independent): $N$ chữ ký rời rạc.
- Ký Toàn phần ($t = n$): Yêu cầu 100% sự đồng thuận, tương đương với lược đồ ký đa của Hamidreza Rahmati và Farhad Rahmati.
- Ký Ngưỡng($t \approx 2/3n$): Yêu cầu đa số đồng thuận, cho phép chịu lỗi.

### *3.1.2. Môi trường đo lường*

- Phần cứng: CPU Intel Xeon E5-2697 v4, 62GB RAM.
- Dữ liệu đo: Thời gian xử lý (CPU time), Dung lượng lưu trữ (Disk usage), và Lưu lượng mạng (Network I/O).
- Quy mô thử nghiệm: $N \in \{3, 5, 10, 20\}$ người tham gia.

## 3.2.Phân tích kết quả thực nghiệm

Thực nghiệm được tiến hành trên mô hình chữ ký ngưỡng Gaussian Threshold 3-of-5, gồm 5 người tham gia và yêu cầu tối thiểu 3 người ký để tạo chữ ký hợp lệ. Hệ thống sử dụng tham số phân phối Gaussian với σ = 261 và được đánh giá trên ba cấp độ an toàn của Dilithium (Level 2, 3, 5). Mỗi kịch bản được chạy 100 lần và lấy giá trị trung bình nhằm đảm bảo độ tin cậy thống kê.

### *3.2.1. Kích thước chữ ký và hiệu quả lưu trữ*

Kết quả cho thấy chữ ký ngưỡng có kích thước lớn hơn chữ ký Dilithium đơn lẻ, nhưng không phụ thuộc vào số lượng người tham gia, duy trì đặc tính O(1).

**Bảng 3.1: Kích thước chữ ký gộp**

| Cấp độ | Kích thước chữ ký | So với Dilithium chuẩn (~2.4 KB) |
|---|---|---|
| Level 2 | 9.03 KB | ~3.7 lần |
| Level 3 | 11.28 KB | ~3.5 lần |
| Level 5 | 14.28 KB | ~3.1 lần |

Phân tích: Sự gia tăng kích thước chủ yếu đến từ các thành phần đặc thù của ký ngưỡng gồm vector ngẫu nhiên *r* và cam kết *com*, vốn là yêu cầu bắt buộc trong giao thức Leevik et al. để đảm bảo tính an toàn.Mặc dù lớn hơn chữ ký đơn, chữ ký ngưỡng vẫn nhỏ hơn đáng kể so với việc lưu trữ nhiều chữ ký độc lập, và đặc biệt không tăng kích thước khi mở rộng số người tham gia. So với mô hình Ký độc lập (kích thước tăng tuyến tính ), chữ ký ngưỡng có kích thước cố định. Với kích thước trung bình 2.4 KB/chữ ký đơn, chỉ cần nhóm ký có từ 4 người trở lên ( KB), mô hình ký ngưỡng đã bắt đầu tiết kiệm dung lượng lưu trữ hơn. Đây là lợi thế lớn cho các ứng dụng Blockchain nơi phí lưu trữ (gas fee) rất đắt đỏ.

### *3.2.2. Hiệu năng tính toán*

**Bảng 3.2: Tổng quát thời gian giữa các quá trình**

| Cấp độ | KeyGen (DKG) | Signing | Verification |
|---|---|---|---|
| Level 2 | 0.118 s | 0.070 s | 0.006 s |
| Level 3 | 0.144 s | 0.110 s | 0.009 s |
| Level 5 | 0.207 s | 0.158 s | 0.016 s |

**Bảng 3.3: Phân rã thời gian thực thi giai đoạn Ký**

| Cấp độ An toàn | Thời gian Ký cục bộ () | Thời gian Gộp () | Tổng độ trễ () | Thành phần phi tính toán (Network/Overhead) |
|---|---|---|---|---|
| Level 2 | 23 ms | 2 ms | 70 ms | ~45 ms (64%) |
| Level 3 | 37 ms | 2 ms | 110 ms | ~71 ms (64%) |
| Level 5 | 53 ms | 3 ms | 158 ms | ~102 ms (65%) |

Kết quả:

+ Ký: Thời gian tạo chữ ký dao động từ 70 ms đến 158 ms, nhanh hơn đáng kể so với các mô hình MPC truyền thống.

+ Xác minh: Thời gian xác minh rất thấp (6–16 ms), gần tương đương xác minh chữ ký đơn, giúp giảm tải cho bên xác thực.
+ DKG: Quá trình sinh khóa phân tán chỉ mất dưới 0.21 s, cho phép khởi tạo hệ thống gần như tức thời.

Phân tích:

+ Sự gia tăng đột biến trong mô hình Ký ngưỡng xuất phát từ cơ chế đồng bộ hóa Rejection Sampling, nếu một bên $P_i$ tính toán ra vector không thỏa mãn điều kiện an toàn, quy trình phải khởi động lại (restart). Với $n$ bên, xác suất để cả $n$ bên cùng thành công trong một lần thử giảm mạnh theo hàm mũ ($P_{total} \approx p^n$). Do đó, số vòng lặp lại tăng lên, kéo theo độ trễ mạng tích lũy.
+ Ngược lại, ở mô hình Ký độc lập, các bên thực hiện Rejection Sampling cục bộ và độc lập. Bên $P_1$ làm lại không ảnh hưởng đến tiến độ của bên $P_2$. Do đó, thời gian tổng chỉ phụ thuộc vào bên chậm nhất, không bị cộng dồn.

### *3.2.3. Phân tích cơ chế "Bí mật kép" (Dual Secrets)*

**Bảng 3.4: Trung bình các lần lặp từ chối**

| Tham số | Giá trị |
|---|---|
| Tỷ lệ chênh lệch $x_i/s_i$ | ~1.7 × $10^6$ |
| Số lần thử trung bình | 5.5 |
| Tỷ lệ chấp nhận | 54.25% |

Phân tích: Mảnh khóa Shamir $x_i$có giá trị rất lớn (~77 triệu), vượt xa biên an toàn chuẩn $B_{BOUND} \approx 14{,}173$. Nếu kiểm tra điều kiện an toàn trực tiếp trên $x_i$, chữ ký sẽ luôn bị từ chối. Việc tách riêng bí mật nhỏ $s_i$để kiểm tra biên cho phép hệ thống vượt qua bước rejection sampling với xác suất thành công trên 50%, xác nhận tính đúng đắn và ổn định của cơ chế Dual Secrets.

### *3.2.4. Phân tích lưu lượng mạng*

**Bảng 3.5: Lưu lượng mạng**

| Cấp độ | Lưu lượng/User | Tổng lưu lượng |
|---|---|---|
| Level 2 | 9.03 KB | 27.09 KB |
| Level 3 | 11.28 KB | 33.84 KB |
| Level 5 | 14.28 KB | 42.84 KB |

Phân tích: Do phải truyền tải các tham số trung gian kích thước lớn của mật mã lưới (ma trận , vector , các cam kết , mảnh chia sẻ ). Để tạo ra một chữ ký ngưỡng nhỏ gọn, hệ thống cần trao đổi lượng dữ liệu lớn hơn kích thước chữ ký cuối cùng, bao gồm các cam kết và chữ ký thành phần. Mức tiêu thụ băng thông này phù hợp với mạng Internet/LAN, nhưng cần cân nhắc tối ưu khi triển khai trên các hệ thống IoT băng thông thấp.

## 3.3. Thảo luận

Mô hình đề xuất đã giảm kích thước xuống còn 9.03 KB nhờ áp dụng kỹ thuật Bit-packing (nén các hệ số đa thức 32-bit xuống ~17-20 bit) theo chuẩn FIPS 204, giúp tiết kiệm băng thông và dung lượng lưu trữ đáng kể. (Theo nghiên cứu của Leevik et al. là ~11.7 KB). Mặc dù chữ ký ngưỡng lớn hơn một chữ ký đơn lẻ (2.4 KB), nhưng nó thể hiện ưu thế vượt trội về lưu trữ khi số lượng người ký tăng lên. Với mô hình ký độc lập, dung lượng lưu trữ tăng tuyến tính; chỉ cần nhóm ký vượt quá 4 người ( 4*2.4KB), mô hình ký ngưỡng đã bắt đầu tiết kiệm không gian lưu trữ hơn,. Điều này đặc biệt có ý nghĩa đối với các ứng dụng Blockchain, nơi chi phí lưu trữ on-chain là cực kỳ đắt đỏ.

Thời gian để tạo ra một chữ ký ngưỡng khá lớn (70ms - 158ms) và chịu thêm ảnh hưởng bởi độ trễ mạng. Điều này khẳng định rằng điểm nghẽn của mật mã lưới phân tán không nằm ở tính toán CPU (vốn rất nhanh nhờ NTT) mà nằm ở sự phức tạp của giao thức tương tác (2 vòng) và đồng bộ hóa trạng thái. Đổi lại, bên xác thực chỉ cần tốn 6ms để kiểm tra một chữ ký ngưỡng duy nhất, thay vì phải tốn thời gian để kiểm tra từng chữ ký lẻ. Đây là lợi thế trong các hệ thống IoT hoặc xử lý giao dịch tần suất cao, việc chuyển gánh nặng tính toán cho bên ký là hợp lý.

Khả năng mở rộng (Scalability Assessment): Từ kết quả thực nghiệm, có thể phân loại khả năng ứng dụng của mô hình theo quy mô nhóm:

- Nhóm nhỏ (n=5): Hoạt động tối ưu. Thời gian tổng < 0.3s, băng thông thấp, tỷ lệ thành công cao. Phù hợp cho ví đa chữ ký hoặc phê duyệt văn bản cấp cao.
- Nhóm trung bình (n=10): Chấp nhận được. Thời gian < 1s, nhưng cần đường truyền mạng tốt để xử lý ~400KB dữ liệu. Phù hợp cho cơ chế đồng thuận nhóm nhỏ.
- Nhóm lớn (n=20): Gặp nút thắt cổ chai. Thời gian sinh khóa > 3s và tỷ lệ thành công vòng đầu < 1% đòi hỏi cơ chế thử lại phức tạp. Không khuyến nghị sử dụng trực tiếp mô hình này mà nên chuyển sang mô hình phân cấp hoặc chọn đại diện nhóm.

# KẾT LUẬN

## 1. Tổng kết các kết quả đạt được

Đề án *"Nghiên cứu và đánh giá các mô hình ký số đa bên hậu lượng tử trên nền thuật toán CRYSTALS-Dilithium"* đã hoàn thành các mục tiêu đề ra:

- Về mặt lý thuyết: Hệ thống hóa thành công cơ sở lý thuyết về PQC và chữ ký ngưỡng, làm rõ các thách thức kỹ thuật của mật mã lưới trong môi trường phân tán.
- Về mặt kỹ thuật: Thiết kế và cài đặt hoàn thiện hệ thống Ký ngưỡng với khả năng tùy biến cao (t-of-n). Hệ thống đã giải quyết thành công các vấn đề khó như Rejection Sampling phân tán và đồng bộ hóa trạng thái ngẫu nhiên.
- Về mặt hiệu năng: Đạt được các chỉ số hiệu năng ấn tượng:
- Tốc độ: Ký ~23ms, Xác minh ~6ms.
- Kích thước: Chữ ký nhỏ gọn cố định ~11KB.
- Độ ổn định: Hoạt động tin cậy trong các bài kiểm tra tải cao.

## 2. Hạn chế của đề tài

Bên cạnh những thành quả đạt được, đề án vẫn còn tồn tại một số hạn chế cần khắc phục trong các nghiên cứu tiếp theo:

- Chi phí băng thông: Để đổi lấy sự tối ưu về lưu trữ, mô hình ký ngưỡng tiêu tốn lượng băng thông nội bộ khá lớn. Điều này có thể gây trở ngại khi triển khai trên các mạng diện rộng có độ trễ cao hoặc băng thông thấp.
- Nền tảng triển khai: Mã nguồn hiện tại được viết bằng Python để phục vụ mục đích kiểm thử ý tưởng. Mặc dù đã tối ưu hóa, nhưng hiệu năng chưa thể so sánh với các ngôn ngữ cấp thấp (như C/C++ hoặc Rust) trên các thiết bị phần cứng hạn chế tài nguyên.
- Môi trường thử nghiệm: Các kịch bản kiểm thử chủ yếu được thực hiện trong môi trường mạng giả lập lý tưởng, chưa đánh giá hết được các tác động của packet loss hay timeout trong môi trường mạng thực tế không ổn định.

## 3. Hướng phát triển trong tương lai

Dựa trên nền tảng vững chắc đã xây dựng, các hướng phát triển tiếp theo bao gồm:

- Tối ưu hóa giao thức truyền thông: Nghiên cứu áp dụng các kỹ thuật nén dữ liệu tiên tiến hoặc cải tiến giao thức để giảm số vòng tương tác và kích thước gói tin trao đổi, nhằm giảm thiểu chi phí băng thông.
- Triển khai trên thiết bị nhúng: Chuyển đổi mã nguồn sang ngôn ngữ Rust hoặc C để tích hợp vào các vi điều khiển, Smartcards hoặc thiết bị IoT, hướng tới các ứng dụng thực tế như ví phần cứng.
- Tích hợp vào Blockchain: Ứng dụng mô hình Ký ngưỡng linh hoạt ($t < n$) vào các nền tảng Blockchain (như Hyperledger Fabric hoặc Ethereum Layer 2) để giải quyết bài toán mở rộng quy mô và tiết kiệm phí lưu trữ (Gas cost).

- Nâng cao khả năng bảo mật: Nghiên cứu các biện pháp chống tấn công kênh kề đối với các thao tác lấy mẫu ngẫu nhiên trong môi trường phân tán.

# DANH MỤC CÁC TÀI LIỆU THAM KHẢO


[1] **"Bảo mật PKI với các giải pháp chống lượng tử - SSL.com". Truy cập: 14 Tháng 11 2025. [Online]. Available at: https://www.ssl.com/vi**

[2] **Hostragons, "Mật mã hậu lượng tử: Bảo mật trong thời đại máy tính lượng tử". Truy cập: 14 Tháng 11 2025. [Online]. Available at: https://www.hostragons.com/vi/blog/bao-mat-mat-ma-luong-tu-sau-trong-ky-nguyen-may-tinh-luong-tu/**

[3] **TS. Nguyễn Quốc Toàn, "Quá trình dịch chuyển mật mã hậu lượng tử và an ninh quốc gia | An toàn thông tin". Truy cập: 16 Tháng 12 2025. [Online]. Available at: https://antoanthongtin.vn/tin/qua-trinh-dich-chuyen-mat-ma-hau-luong-tu-va-an-ninh-quoc-gia**

[4] **L. M. K. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, và I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance", 2001.**

[5] **D. Roux, C. Cachin, và I. Amores Sesar, "Implementation of a Threshold Post-Quantum Signature Scheme Master Thesis", 2022.**

[6] **"Digital Signature Standard (DSS)", tháng 2 2023. doi: 10.6028/NIST.FIPS.186-5.**

[7] **L. T. A. N. Brandão, N. Mouha, và A. Vassilev, "Threshold schemes for cryptographic primitives":, Gaithersburg, MD, tháng 3 2019. doi: 10.6028/NIST.IR.8214.**

[8] **D. Boneh, B. Lynn, và H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps".**

[9] **H. Rahmati và F. Rahmati, "Quantum resistant multi-signature scheme with optimal communication round: A Blockchain-based approach". [Online]. Available at: https://www.fortunebusinessinsights.com/industry-reports/blockchain-market-100072**

**[10]** **A. Leevik, V. Davydov, và S. Bezzateev, "Threshold Lattice-Based Signature Scheme for Authentication by Wearable Devices", *Cryptography*, vol 7, số p.h 3, tháng 9 2023, doi: 10.3390/cryptography7030033.**

**[11]** **ETSI, "EN 319 142-2 - V1.2.1 - Electronic Signatures and Trust Infrastructures (ESI); PAdES digital signatures; Part 2: Additional PAdES signatures profiles", 2025.**

**[12]** **C. Gidney và M. Ekerå, "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits", 2021, doi: 10.22331/q-2021-04-15-433.**

**[13]** **G. Alagic *và c.s.*, "Status report on the third round of the NIST Post-Quantum Cryptography Standardization process", tháng 9 2022. doi: 10.6028/NIST.IR.8413-upd1.**

**[14]** **"Post-Quantum Cryptography | CSRC". Truy cập: 16 Tháng Chạp 2025. [Online]. Available at: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/selected-algorithms**

**[15]** **S. Bai *và c.s.*, "CRYSTALS-Dilithium Algorithm Specifications and Supporting Documentation (Version 3.1)", 2021. [Online]. Available at: https://pq-crystals.org/dilithium/**

**[16]** **D. Cozzo và N. P. Smart, "Sharing the LUOV: Threshold Post-Quantum Signatures", 2019. [Online]. Available at: https://www.nist.gov/news-events/events/2019/03/nist-threshold-cryptography-workshop-2019.**

**[17]** **H. Rahmati và F. Rahmati, "Quantum resistant multi-signature scheme with optimal communication round: A Blockchain-based approach". [Online]. Available at: https://www.fortunebusinessinsights.com/industry-reports/blockchain-market-100072**

## BẢN CAM ĐOAN

Tôi cam đoan đã thực hiện việc kiểm tra mức độ tương đồng nội dung đề án/ qua phần mềm Turnitin một cách trung thực và đạt kết quả mức độ tương đồng 15% toàn bộ nội dung đề án/luận văn/luận án. Bản đề án kiểm tra qua phần mềm là bản cứng đề án đã nộp để bảo vệ trước hội đồng. Nếu sai tôi xin chịu các hình thức kỷ luật theo quy định hiện hành của Học viện.

*Hà Nội, ngày* 09 *tháng* 02 *năm* 2026

HỌC VIÊN CAO HỌC

**Nguyễn Sơn Lâm**

# KẾT QUẢ KIỂM TRA TRÙNG LẶP TÀI LIỆU

turnitin

## 15% Tỉnh tương đồng nói chung

**Đã lọc khỏi Báo cáo**

- Mục lục tham khảo
- Văn bản được trích dẫn
- Kết quả trùng khớp nhỏ (ít hơn 8 từ)

**Nguồn hàng đầu**

15% Nguồn Internet
4% Ấn bản
4% Bài tập được nộp (bởi các học sinh)

*Hà Nội, ngày* 09 *tháng* 03 *năm* 2026

GIẢNG VIÊN HƯỚNG DẪN

**TS. ĐẶNG MINH TUẤN**

HỌC VIÊN CAO HỌC

**NGUYỄN SƠN LÂM**

BỘ KHOA HỌC VÀ CÔNG NGHỆ
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**
**Độc lập - Tự do - Hạnh phúc**

# BÁO CÁO GIẢI TRÌNH
# SỬA CHỮA, HOÀN THIỆN ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Họ và tên học viên: Nguyễn Sơn Lâm

Chuyên ngành: Hệ thống thông tin

Khóa: 2024 đợt 1

Tên đề tài: Nghiên cứu và đánh giá các mô hình ký số đa bên hậu lượng tử trên nền thuật toán Crystals-Dilithium

Người hướng dẫn khoa học: TS. Đặng Minh Tuấn

Ngày bảo vệ: 24/01/2026

Các nội dung học viên đã sửa chữa, bổ sung trong đề án tốt nghiệp theo ý kiến đóng góp của Hội đồng chấm đề án tốt nghiệp:

| TT | Ý kiến hội đồng | Sửa chữa của học viên |
|---|---|---|
| 1 | Chỉnh sửa lại hình vẽ trong đề án: việt hóa, trích dẫn hoặc thực hiện vẽ lại. | Học viên đã thực hiện vẽ lại sửa các từ tiếng anh thành tiếng việt tại trang 7, 10, 13 |
| 2 | Làm rõ các đánh giá, tính chất đa bên. | Học viên đã thêm mô tả về mô hình ký số đa bên tại mục 1.2.1 trang 6.<br>Học viên đã bổ sung các phân tích tại mục 3.2 trang 62 về kích thước chữ ký, hiệu năng tính toán và lưu lượng mạng. Bổ sung thảo luận về hiệu quả mô hình tại mục 3.3 trang 65. |

*Hà Nội, ngày..... tháng 02 năm 2026*

**Ký xác nhận của**

| CHỦ TỊCH HỘI ĐỒNG CHẤM ĐỀ ÁN | THƯ KÝ HỘI ĐỒNG | NGƯỜI HƯỚNG DẪN KHOA HỌC | HỌC VIÊN |
|---|---|---|---|
| PGS.TS. Trần Đình Quế | TS. Đào Thị Thúy Quỳnh | TS. Đặng Minh Tuấn | Nguyễn Sơn Lâm |

BỘ KHOA HỌC VÀ CÔNG NGHỆ
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**
**Độc lập - Tự do - Hạnh phúc**

# BIÊN BẢN
# HỌP HỘI ĐỒNG CHẤM ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

Căn cứ quyết định số Quyết định số 3299/QĐ-HV ngày 26 tháng 12 năm 2025 của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ. Hội đồng đã họp vào hồi ...9... giờ ..30.. phút, ngày 24 tháng 01 năm 2025 tại Học viện Công nghệ Bưu chính Viễn thông để chấm đề án tốt nghiệp thạc sĩ cho:

Học viên: **Nguyễn Sơn Lâm**

Tên đề án tốt nghiệp: **Nghiên cứu và đánh giá các mô hình ký số đa bên hậu lượng tử trên nền thuật toán CRYSTALS-Dilithium**

Ngành: **Hệ thống thông tin** Mã số: **8.48.01.04**

Các thành viên của Hội đồng chấm đề án tốt nghiệp có mặt: ...5.../ 05

| TT | HỌ VÀ TÊN | TRÁCH NHIỆM TRONG HĐ | GHI CHÚ |
|---|---|---|---|
| 1 | **PGS.TS. Trần Đình Quế** | Chủ tịch | |
| 2 | **TS. Đào Thị Thúy Quỳnh** | Thư ký | |
| 3 | **PGS.TS. Nguyễn Hà Nam** | Phản biện 1 | |
| 4 | **TS. Nguyễn Quang Hưng** | Phản biện 2 | |
| 5 | **PGS.TS. Trần Đăng Hưng** | Uỷ viên | |

**Các nội dung thực hiện:**

1. Chủ tịch Hội đồng điều khiển buổi họp. Công bố quyết định của Giám đốc Học viện Công nghệ Bưu chính Viễn thông về việc thành lập Hội đồng chấm đề án tốt nghiệp thạc sĩ.
2. Người hướng dẫn khoa học hoặc thư ký đọc lý lịch khoa học và các điều kiện bảo vệ đề án tốt nghiệp của học viên (Có bản lý lịch khoa học và kết quả các môn học cao học của học viên kèm theo).
3. Học viên trình bày tóm tắt đề án tốt nghiệp.
4. Phản biện 1 đọc nhận xét (có văn bản kèm theo)
5. Phản biện 2 đọc nhận xét (có văn bản kèm theo)
6. Các câu hỏi của thành viên Hội đồng:

1) Học viên giải thích kỹ lời ích của loại chữ ký phương pháp ký đa bên, chữ ký ngưỡng?
2) Tính hiệu lượng khi thực hiện như? Phân tích được và sau khi được hiệu lượng đó?
3) Khả năng thực hiện ứng dụng ở thiết bị đầu?

7. Trả lời của học viên:

Chưa hiểu thường hay trong các giao dịch blockchain.
Học viên sẽ về tìm hiểu thêm.

- Học viên chưa thực nghiệm nên sẽ thực hiện thêm.

8. Thư ký đọc nhận xét về quá trình thực hiện đề án tốt nghiệp của học viên *(có văn bản kèm theo).*

9. Hội đồng họp riêng:

- Bầu Ban kiểm phiếu:
  1. Trưởng Ban kiểm phiếu: PGS.TS. Trần Đăng Hưng
  2. Uỷ viên Ban kiểm phiếu: TS. Nguyễn Quang Hưng
  3. Uỷ viên Ban kiểm phiếu: TS. Đào Thị Thúy Quỳnh
- Hội đồng chấm đề án tốt nghiệp bằng bỏ phiếu kín.
- Ban kiểm phiếu làm việc:
- Trưởng Ban kiểm phiếu báo cáo kết quả kiểm phiếu *(có Biên bản họp Ban kiểm phiếu kèm theo)*
- **Điểm trung bình của đề án tốt nghiệp:** 7,8

**Kết luận:**

1. Các nội dung cần chỉnh sửa, hoàn thiện sau bảo vệ đề án tốt nghiệp:

- Chỉnh sửa lại hình vẽ trong đề án: viết hoa, trình bày hoặc thực hiện vẽ lại;
- Làm rõ các cách đánh giá, tính chất đa bên.

2. Đề nghị Học viện công nhận (hoặc không) và cấp bằng (hoặc không) thạc sĩ cho học viên: Đề nghị

3. Đề án tốt nghiệp có thể phát triển thành đề tài nghiên cứu cho NCS……

Buổi làm việc kết thúc vào…… cùng ngày.

| **Chủ tịch** | **Thư ký** |
|---|---|
| **PGS.TS. Trần Đình Quế** | **TS. Đào Thị Thúy Quỳnh** |

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Độc lập – Tự do – Hạnh phúc**

# BẢN NHẬN XÉT ĐỀ ÁN TỐT NGHIỆP THẠC SĨ

*(Dùng cho người phản biện)*

Tên đề tài đề án tốt nghiệp: Nghiên cứu và đánh giá các mô hình ký số đa bên hậu lượng tử trên nền thuật toán CRYSTALS-Dilithium
Chuyên ngành: Hệ thống thông tin; Mã chuyên ngành: 8.48.01.04
Họ và tên học viên: Nguyễn Sỉn Lõm

Họ và tên người nhận xét: Nguyễn Quang Hưng
Học hàm, học vị: Tiến sỹ; Chuyên ngành: Mạng và kênh thông tin liên lạc
Cơ quan công tác: Học viện CN Bưu chính viễn thông
Số điện thoại: .....E-mail: hungnq1@ptit.edu.vn

## NỘI DUNG NHẬN XÉT

**I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:**

- Đề tài có tính thời sự cao khi tập trung vào mật mã hậu lượng tử (PQC), cụ thể là thuật toán CRYSTALS-Dilithium vừa được NIST chuẩn hóa thành FIPS 204 vào tháng 8/2024.
- Việc chuyển đổi từ mô hình ký đơn sang ký số đa bên đáp ứng nhu cầu thực tế trong các hệ thống phân tán, Blockchain và Chính phủ điện tử, nơi quyền kiểm soát cần được phân tán để tránh rủi ro từ một thực thể đơn lẻ.

**II/ Nội dung của đề án tốt nghiệp, các kết quả đã đạt được:**

- Đề án đã thực hiện đo đạc thực nghiệm chi tiết các chỉ số: thời gian sinh khóa (DKG), thời gian ký, thời gian xác minh và lưu lượng mạng trên nhiều cấp độ an toàn (Level 2, 3, 5).
- Một điểm sáng là việc áp dụng kỹ thuật Bit-packing giúp giảm kích thước chữ ký ngưỡng xuống còn khoảng 9.03 KB (Level 2), ưu việt hơn so với các nghiên cứu trước đó như của Leevik (~11.7 KB).
- Học viên đã hệ thống hóa tốt các mô hình ký đa bên (Ký ngưỡng, Ký gộp, Ký độc lập) và lựa chọn hướng triển khai Ký ngưỡng (t, n) kết hợp với Ký độc lập làm đối trọng so sánh là hoàn toàn hợp lý.
- Việc sử dụng ngôn ngữ Python kết hợp với thư viện Numba và Ctypes để tối ưu hóa hiệu năng tính toán (NTT, SHAKE) cho thấy sự đầu tư kỹ thuật bài bản nhằm đạt tới tốc độ xử lý ở mức miligiây.

**III/ Những vấn đề cần giải thích thêm:**

***1. Về cơ chế bảo mật***

Đề án sử dụng mô hình Đại lý tin cậy (Trusted Dealer) cho giai đoạn khởi tạo khóa. Mặc dù giúp đơn giản hóa việc thử nghiệm hiệu năng, nhưng trong thực tế, đây lại là

"điểm lỗi đơn lẻ" (single point of failure), vì Dealer nắm giữ toàn bộ khóa bí mật tại thời điểm sinh khóa.

***2. Về khả năng mở rộng***

Kết quả thực nghiệm cho thấy khi số người tham gia tăng lên (N=20), hệ thống gặp nút thắt cổ chai với tỷ lệ thành công vòng đầu rất thấp (< 1%) và thời gian sinh khóa tăng mạnh. Điều này cho thấy giới hạn của mô hình khi triển khai ở quy mô lớn.

***3. Về hiệu năng mạng***

Chi phí băng thông nội bộ của mô hình ký ngưỡng vẫn còn lớn (từ 27 KB đến hơn 42 KB mỗi người dùng), có thể gây trở ngại cho các thiết bị IoT có kết nối không ổn định

**IV/ Kết luận:**

Nhìn chung, đề án là một công trình nghiên cứu định hướng ứng dụng có tính thực tiễn cao, nhưng cần được tối ưu hóa thêm về mặt giao tiếp mạng và nền tảng phần cứng để có độ chín phù hợp cho triển khai thực tế.

Đồng ý cho học việc được bảo vệ trước Hội đồng chấm đề án.

**CÂU HỎI:**

***Câu 1: Về mô hình Trusted Dealer***

Trong phần thuyết minh, học viên sử dụng Dealer để tạo và phân phối các mảnh khóa bí mật. Làm thế nào để loại bỏ hoàn toàn thực thể trung gian này trong thực tế (triển khai DKG phi tập trung hoàn toàn) mà vẫn đảm bảo hiệu năng tính toán không bị sụt giảm quá sâu như mô hình MPC của Roux?

***Câu 2: Về cơ chế "Bí mật kép" (Dual Secrets)***

Học viên phân tích rằng mảnh khóa Shamir $x_i$ có giá trị rất lớn và phải tách riêng bí mật nhỏ $s_i$ để kiểm tra biên an toàn (rejection sampling). Nếu một bên tham gia cố tình gửi sai phần chữ ký dựa trên bí mật lớn $x_i$, hệ thống của học viên có cơ chế nào để xác thực (Verifiable Secret Sharing) và loại bỏ bên gian lận ngay lập tức mà không phải khởi động lại toàn bộ quy trình ký hay không?

***Câu 3: Về khả năng ứng dụng trên thiết bị hạn chế tài nguyên***

Một trong các mục tiêu của đề án là hướng tới thiết bị IoT/Wearables. Tuy nhiên, mã nguồn hiện tại được viết bằng Python. Dựa trên kết quả đo đạc lưu lượng mạng (~15KB/user) và bộ nhớ, học viên đánh giá đâu là thách thức lớn nhất khi chuyển đổi (porting) thuật toán này sang các ngôn ngữ cấp thấp (C/Rust) để chạy trên các vi điều khiển có RAM rất nhỏ (ví dụ < 64KB)?

Hà Nội, ngày 23 tháng 1 năm 2026

NGƯỜI NHẬN XÉT



TS. Nguyễn Quang Hưng

CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
**Độc lập - Tự do - Hạnh phúc**

# BẢN NHẬN XÉT LUẬN VĂN TỐT NGHIỆP THẠC SĨ
(*Dùng cho người phản biện*)

Tên đề tài luận văn: *Nghiên cứu và đánh giá các mô hình ký số đa bên hậu lượng tử trên nền thuật toán CRYSTALS-Dilithium*
Chuyên ngành: Hệ thống thông tin Mã số: 8.48.01.04
Tên học viên: Nguyễn Sơn Lâm
Họ và tên người nhận xét: Nguyễn Hà Nam.....
Học hàm, học vị: PGS. TS Chuyên ngành: CNTT .....
Cơ quan công tác: Ban Khoa học và Đổi mới sáng tạo, ĐHQGHN.....

## NỘI DUNG NHẬN XÉT

**I/ Cơ sở khoa học và thực tiễn, tính cấp thiết của đề tài:**
Đề tài có cơ sở khoa học rõ ràng và tính thời sự cao, xuất phát từ mối đe dọa hiện hữu của máy tính lượng tử đối với các hệ mật mã khóa công khai truyền thống như RSA và ECDSA. Việc lựa chọn CRYSTALS-Dilithium – thuật toán chữ ký số hậu lượng tử đã được NIST chuẩn hóa làm nền tảng nghiên cứu là phù hợp và có giá trị thực tiễn cao.
Bên cạnh đó, đề tài gắn với nhu cầu thực tế về ký số đa bên trong các hệ thống quản lý văn bản điện tử, tài chính – kế toán, blockchain và chính phủ điện tử. Việc kết hợp hai hướng nghiên cứu: *ký số đa bên* và *mật mã hậu lượng tử* cho thấy học viên có định hướng tiếp cận vấn đề đúng đắn.
Tuy nhiên, tính cấp thiết khoa học chưa được làm sắc nét. Phần mở đầu mới dừng ở việc mô tả xu hướng và bối cảnh chung, chưa chỉ ra rõ khoảng trống nghiên cứu cụ thể: đâu là vấn đề chưa được giải quyết trong các nghiên cứu hiện có về ký đa bên trên Dilithium, và đề tài này giải quyết khoảng trống đó theo cách nào. Điều này khiến động cơ nghiên cứu mang tính "hợp lý" nhưng chưa đủ "thuyết phục" về mặt học thuật.

**II/ Về nội dung, chất lượng của luận văn, các kết quả đã đạt được (so với đề cương đã được duyệt):**
Luận văn được xây dựng đúng cấu trúc của một đề án thạc sĩ định hướng ứng dụng, bám sát đề cương đã được phê duyệt. Phần cơ sở lý thuyết và tổng quan nghiên cứu được trình bày công phu, hệ thống hóa khá đầy đủ các mô hình ký số đa bên (ký ngưỡng, ký gộp, ký độc lập) cũng như nền tảng toán học của CRYSTALS-Dilithium.
Tuy nhiên, xét dưới góc độ phản biện chuyên ngành CNTT, nội dung nghiên cứu còn nặng về trình bày và diễn giải lý thuyết, nhẹ về đóng góp kỹ thuật riêng. Phần lớn nội dung chương 1 mang tính tổng hợp tài liệu, trong khi phần phân tích so sánh và rút ra nhận định mang tính phản biện khoa học còn hạn chế.
Ở chương 2 và chương 3, học viên đã xây dựng mô hình thử nghiệm và tiến hành đo đạc các chỉ số hiệu năng (thời gian ký, kích thước chữ ký, lưu lượng mạng, khả năng mở rộng). Tuy nhiên, thực nghiệm chủ yếu dừng ở mức đánh giá mô tả, chưa đủ chiều sâu để khẳng định ưu – nhược điểm của từng mô hình ký đa bên trong các kịch bản ứng dụng khác nhau. Việc lựa chọn tham số, kịch bản thử nghiệm và cách diễn giải kết quả chưa được liên hệ chặt chẽ với các yêu cầu thực tế (ví dụ: môi trường IoT, blockchain hay hệ thống hành chính điện tử).

**III/ Những vấn đề cần giải thích thêm:**

Thứ nhất, học viên cần làm rõ đóng góp khoa học cốt lõi của luận văn. Hiện tại, đề án thiên về tái hiện và đánh giá các mô hình có sẵn hơn là đề xuất cải tiến hay phương án tối ưu mới dựa trên Dilithium.

Thứ hai, cần giải thích rõ hơn ý nghĩa thực tiễn của các chỉ số đánh giá. Các kết quả đo đạc (kích thước chữ ký, thời gian ký, lưu lượng mạng) cần được gắn với ngưỡng chấp nhận cụ thể trong từng bối cảnh ứng dụng, thay vì chỉ so sánh tương đối giữa các mô hình.

Thứ ba, phần thảo luận kết quả còn ngắn, chưa phân tích sâu nguyên nhân dẫn đến sự khác biệt về hiệu năng giữa các mô hình ký ngưỡng, ký gộp và ký độc lập, đặc biệt trong mối liên hệ với đặc tính toán học và cơ chế nội tại của Dilithium.

Tuy nhiên, để nâng cao chất lượng học thuật, luận văn cần được chỉnh sửa theo hướng làm rõ đóng góp nghiên cứu, tăng chiều sâu phân tích kỹ thuật và mở rộng phần thảo luận kết quả. Nếu học viên tập trung thu hẹp phạm vi để đào sâu một mô hình cụ thể, đồng thời tăng cường lập luận phản biện và liên hệ thực tiễn, luận văn sẽ đạt chất lượng tốt hơn và đáp ứng rõ ràng hơn yêu cầu của một luận văn thạc sĩ chuyên ngành CNTT tại PTIT.

**IV/ Kết luận:**

Luận văn là một công trình được thực hiện nghiêm túc, lựa chọn đề tài đúng hướng và phù hợp với định hướng ứng dụng của chương trình đào tạo thạc sĩ tại PTIT. Học viên thể hiện năng lực tổng hợp kiến thức tốt và khả năng triển khai thử nghiệm trên nền tảng mật mã hậu lượng tử hiện đại. Tôi đồng ý để học viên được bảo vệ luận văn trước Hội đồng chấm luận văn thạc sĩ

Ngày tháng năm 2026

NGƯỜI NHẬN XÉT

(Ký và ghi rõ họ tên)

Nguyễn Hà Nam