TỐI ƯU HÓA MÃ HÓA ĐỒNG HÌNH TRÊN THIẾT BỊ DI ĐỘNG VỚI CÁC PHƯƠNG PHÁP BẢO MẬT TRONG ỨNG DỤNG HỌC MÁY

Thái Thanh Tùng[1*], Vũ Xuân Hạnh[1], Lê Hữu Nam[1],
Trần Duy Hùng[1], Đặng Thùy Linh[1]
[*]Tác giả liên hệ, email: tttung@hou.edu.vn. ORCID: 0009-0004-2910-568X

Ngày tòa soạn nhận được bài báo: 15/01/2026
Ngày phản biện đánh giá: 17/03/2026
Ngày bài báo được duyệt đăng: 14/04/2026

DOI: 10.59266/houjs.2026.1172

Tóm tắt: *Mã hóa đồng hình (Homomorphic Encryption - HE) cho phép thực hiện tính toán trực tiếp trên dữ liệu được mã hóa, mở ra hướng tiếp cận hiệu quả cho học máy bảo vệ quyền riêng tư (PPML). Tuy nhiên, chi phí tính toán cao của HE là rào cản lớn đối với việc triển khai trên các thiết bị di động có tài nguyên hạn chế. Bài báo này đánh giá tính khả thi của lược đồ CKKS trong suy luận giọng nói được mã hóa trên điện thoại thông minh thông qua một kiến trúc di động-đám mây lai. Chúng tôi triển khai mô hình CNN 3 lớp tương thích HE và thực nghiệm trên tập dữ liệu LibriSpeech. Kết quả cho thấy độ trễ đầu-cuối đạt dưới 1,5 giây (Wifi) và dưới 2 giây (LTE), với mức tiêu thụ năng lượng chấp nhận được. Mặc dù quá trình mã hóa trên thiết bị vẫn là nút thắt cổ chai, các kỹ thuật như lượng tử hóa mô hình, xử lý theo lô và phân bổ tính toán hợp lý cho thấy tiềm năng triển khai thực tế của các ứng dụng AI bảo vệ quyền riêng tư trên thiết bị di động.*

Từ khóa: *mã hóa đồng hình, Mobile-cloud computing, trí tuệ nhân tạo bảo vệ quyền riêng tư, Bảo mật di động, suy luận mã hóa*

I. Đặt vấn đề

Sự phát triển nhanh chóng của điện toán di động và trí tuệ nhân tạo (AI) đã thúc đẩy việc thu thập và xử lý lượng lớn dữ liệu cá nhân trên các thiết bị cầm tay. Tuy nhiên, việc phụ thuộc vào điện toán đám mây để xử lý các tác vụ AI làm gia tăng đáng kể rủi ro về quyền riêng tư và bảo mật dữ liệu, đặc biệt khi dữ liệu được truyền và xử lý trên các hệ thống không đáng tin cậy.

Các phương pháp mã hóa truyền thống như AES và RSA chỉ bảo vệ dữ liệu trong quá trình lưu trữ và truyền tải, nhưng yêu cầu giải mã trước khi xử lý, dẫn đến nguy cơ rò rỉ thông tin (Sahai & Waters,

[1] Trường Đại học Mở Hà Nội, Hà Nội, Việt Nam

2005). Mã hóa đồng hình (Homomorphic Encryption - HE) khắc phục hạn chế này bằng cách cho phép thực hiện tính toán trực tiếp trên dữ liệu đã được mã hóa (Gentry, 2009).

Trong các lược đồ HE hiện đại, CKKS hỗ trợ tính toán xấp xỉ trên số thực, phù hợp với các tác vụ học máy (Cheon và cộng sự, 2017). Tuy nhiên, chi phí tính toán cao và yêu cầu tài nguyên lớn vẫn là rào cản chính khi triển khai trên thiết bị di động (Chen và cộng sự, 2020). Đồng thời, các thư viện HE hiện nay chủ yếu được tối ưu cho hệ thống máy chủ và chưa hỗ trợ hiệu quả cho kiến trúc di động như ARM (Laine và cộng sự, 2017).

Mặc dù đã có nhiều nghiên cứu về HE và học máy bảo vệ quyền riêng tư, phần lớn tập trung vào môi trường máy chủ, chưa xem xét đầy đủ các ràng buộc thực tế trên thiết bị di động như tiêu thụ năng lượng và độ trễ (Chen và cộng sự, 2020; Lu & Zhang, 2019). Do đó, việc đánh giá khả năng triển khai HE trong các hệ thống AI di động vẫn là một vấn đề mở.

Bài báo này đề xuất và đánh giá một kiến trúc di động-đám mây lai cho suy luận học máy được mã hóa sử dụng CKKS. Nghiên cứu tập trung phân tích hiệu năng, tiêu thụ năng lượng và độ trễ, từ đó làm rõ sự đánh đổi giữa bảo mật và khả năng triển khai trong môi trường thực tế.

Những đóng góp chính của nghiên cứu này được tóm tắt như sau:

(1) Triển khai thực nghiệm mã hóa đồng hình CKKS trên thiết bị di động thực (Samsung Galaxy S21), bao gồm tối ưu biên dịch thư viện SEAL cho kiến trúc ARM64;

(2) Đề xuất chiến lược tối ưu hóa kết hợp giữa xử lý theo lô (batching), lựa chọn đa thức xấp xỉ và phân bổ tính toán mobile-cloud nhằm giảm độ trễ và tiêu thụ năng lượng trong suy luận mã hóa;

(3) Xây dựng bộ đánh giá toàn diện bao gồm độ trễ, mức tiêu thụ năng lượng, băng thông và độ chính xác mô hình, đồng thời so sánh với phương pháp suy luận không mã hóa (baseline);

(4) Phân tích sự đánh đổi giữa bảo mật (mức 128bit), hiệu năng và khả năng triển khai thực tế trong các hệ thống AI bảo vệ quyền riêng tư trên thiết bị di động.

II. Cơ sở lý thuyết

2.1. Mã hóa đồng hình và lược đồ CKKS

Mã hóa đồng hình (Homomorphic Encryption - HE) cho phép thực hiện các phép toán trực tiếp trên dữ liệu đã được mã hóa mà không cần giải mã, từ đó đảm bảo tính bảo mật trong quá trình xử lý (Gentry, 2009). Các lược đồ HE hiện đại như BGV và BFV dựa trên các bài toán khó trong lý thuyết lattice và hỗ trợ các phép toán số học trên bản mã (Brakerski và cộng sự, 2014; Fan & Vercauteren, 2012).

Trong số đó, lược đồ CKKS nổi bật nhờ khả năng hỗ trợ tính toán xấp xỉ trên số thực, phù hợp với các ứng dụng học máy và xử lý tín hiệu (Cheon và cộng sự, 2017). Tuy nhiên, các phép toán đồng hình làm gia tăng nhiễu trong bản mã, và việc kiểm soát nhiễu, đặc biệt thông qua bootstrapping, đòi hỏi chi phí tính toán đáng kể (Cheon và cộng sự, 2018). Điều này hạn chế khả năng triển khai HE trên các nền tảng có tài nguyên hạn chế như thiết bị di động.

2.2. Thư viện HE và thách thức trên thiết bị di động

Các thư viện HE phổ biến như Microsoft SEAL và HElib cung cấp các công cụ triển khai các phép toán đồng hình và thường được tối ưu hóa cho môi

trường máy chủ (Microsoft Research, 2023; Halevi & Shoup, 2020). Tuy nhiên, khả năng hỗ trợ cho kiến trúc di động như ARM vẫn còn hạn chế, dẫn đến hiệu năng thấp khi triển khai trên thiết bị cầm tay (Laine và cộng sự, 2017).

Các nghiên cứu thực nghiệm cho thấy chi phí tính toán và tiêu thụ năng lượng là những rào cản chính đối với HE trên thiết bị di động (Chen và cộng sự, 2020). Do đó, nhiều hướng tiếp cận đề xuất mô hình di động-đám mây lai, trong đó thiết bị di động thực hiện mã hóa/giải mã, còn máy chủ đảm nhận các phép tính đồng hình (Lu & Zhang, 2019).

2.3. HE trong học máy bảo vệ quyền riêng tư (PPML)

HE là một công cụ quan trọng trong học máy bảo vệ quyền riêng tư (PPML), cho phép thực hiện suy luận mô hình mà không tiết lộ dữ liệu đầu vào. CryptoNets là một trong những công trình tiên phong, chứng minh khả năng áp dụng HE trong suy luận mạng nơ-ron (Gilad-Bachrach và cộng sự, 2016). Các nghiên cứu tiếp theo tập trung vào việc tối ưu hóa mô hình để phù hợp với các ràng buộc của HE, chẳng hạn như sử dụng các hàm kích hoạt đa thức nhằm giảm độ sâu tính toán (Bourse và cộng sự, 2018). Ngoài ra, các framework như nGraph-HE đã được đề xuất để cải thiện hiệu năng suy luận trên dữ liệu mã hóa (Boemer và cộng sự, 2019).

Tuy nhiên, phần lớn các nghiên cứu này được triển khai trên môi trường máy chủ, trong khi áp dụng trực tiếp trên thiết bị di động vẫn còn hạn chế do các ràng buộc về tài nguyên và năng lượng.

2.4. Khoảng trống nghiên cứu

Mặc dù đã có nhiều tiến bộ trong HE và PPML, việc triển khai trên thiết bị di động vẫn tồn tại một số thách thức chính: (i) thiếu các thư viện HE được tối ưu hóa cho kiến trúc di động, (ii) hạn chế về các khung đánh giá hiệu năng toàn diện, đặc biệt liên quan đến năng lượng và độ trễ, và (iii) thiếu các nghiên cứu thực nghiệm trên các tác vụ thực tế.

Ngoài ra, đánh đổi giữa mức độ bảo mật, hiệu năng và khả năng triển khai trong các hệ thống AI di động sử dụng HE vẫn chưa được phân tích đầy đủ. Nghiên cứu này nhằm giải quyết các khoảng trống trên thông qua đánh giá thực nghiệm lược đồ CKKS trong kiến trúc di động-đám mây.

III. Phương pháp nghiên cứu

3.1. Tổng quan về kiến trúc hệ thống

Nghiên cứu đề xuất một kiến trúc di động-đám mây lai nhằm triển khai suy luận học máy trên dữ liệu được mã hóa bằng CKKS. Trong kiến trúc này, thiết bị di động thực hiện thu thập dữ liệu, tiền xử lý và mã hóa, trong khi máy chủ đảm nhận suy luận trên bản mã. Kết quả được trả về dưới dạng mã hóa và giải mã cục bộ tại thiết bị người dùng.

Cách tiếp cận này phù hợp với mô hình tính toán thuê ngoài an toàn, trong đó dữ liệu nhạy cảm không bao giờ xuất hiện ở dạng bản rõ trên máy chủ, giúp giảm thiểu rủi ro rò rỉ thông tin (Popa và cộng sự, 2011). Quy trình xử lý gồm các bước: (i) trích xuất đặc trưng từ tín hiệu giọng nói, (ii) mã hóa bằng CKKS, (iii) truyền dữ liệu, (iv) suy luận đồng hình và (v) giải mã kết quả.

Hệ thống bao gồm ba thành phần chính:

Ứng dụng di động (client-side): thu thập dữ liệu giọng nói, thực hiện tiền xử lý, mã hóa và giải mã kết quả;

Máy chủ suy luận (server-side): thực hiện các phép toán đồng hình trên mô hình học sâu;

Lớp truyền thông (communication layer): đảm bảo truyền tải dữ liệu mã hóa an toàn thông qua các giao thức như TLS.

Hình minh họa kiến trúc hệ thống trong đó thiết bị di động thực hiện thu thập dữ liệu, mã hóa CKKS và giải mã kết quả, trong khi máy chủ thực hiện suy luận trên bản mã thông qua tính toán đồng hình. Dữ liệu được truyền qua lớp truyền thông bảo mật TLS.

Hình 1: Kiến trúc hệ thống suy luận mã hóa di động - đám mây

3.2. Thiết lập thực nghiệm và cấu hình hệ thống

3.2.1. Cấu hình phía thiết bị di động

Thiết bị: Samsung Galaxy S21

Bộ xử lý: Qualcomm Snapdragon 888 (8 nhân, 5nm)

RAM: 8GB

Hệ điều hành: Android 13

Thư viện HE: Microsoft SEAL v4.1, được biên dịch chéo cho ARM64 bằng Android NDK r25 và Clang 14.

Kiến trúc mô hình: CNN 3 lớp với phép tích chập 1D, được lượng tử hóa hoàn toàn và ánh xạ tới các phép toán đa thức tương thích HE. Ví dụ: thay thế ReLU bằng các phép xấp xỉ bậc thấp như hàm bình phương hoặc spline đa thức (Bourse và cộng sự, 2018).

Dữ liệu đầu vào: Tín hiệu giọng nói từ kho ngữ liệu LibriSpeech ASR (Panayotov và cộng sự, 2015), được xử lý trước thành các đặc trưng hệ số cepstral tần số mel (MFCC) và được chuẩn hóa thành các vectơ giá trị thực phù hợp cho mã hóa CKKS.

Các thao tác mã hóa và giải mã được triển khai bằng C++ thông qua các liên kết JNI. Phương pháp xử lý theo lô (vectorized batching) cho phép mã hóa 16 khung đặc trưng MFCC cho mỗi bản mã. Kích thước bản mã trung bình khoảng 80 KB mỗi mẫu, và độ trễ mã hóa dao động từ 600ms đến 1,1 giây mỗi mẫu, tùy thuộc vào cài đặt tham số.

3.2.2. Cấu hình phía máy chủ:

Máy chủ suy luận sử dụng CPU đa lõi và thực thi các phép toán đồng hình thông qua thư viện Microsoft SEAL với hỗ trợ đa luồng. Mô hình học máy được triển khai là mạng nơ-ron tích chập (CNN) 3 lớp, được điều chỉnh để tương thích với HE bằng cách thay thế các hàm kích hoạt phi tuyến như ReLU bằng các hàm đa thức xấp xỉ bậc thấp (Bourse và cộng sự, 2018). Các phép toán tích chập được ánh xạ sang các phép toán đồng hình tương ứng, bao gồm phép quay (rotation), phép

nhân và phép cộng trên bản mã. Nhờ thiết kế mạng nông và tối ưu hóa tham số, quá trình bootstrapping không được sử dụng nhằm giảm chi phí tính toán (Cheon và cộng sự, 2018).

CPU **:** Intel Xeon Gold 5218 (16 lõi, 2.3 GHz)

RAM **:** 128GB

Khung HE **:** Microsoft SEAL 4.1 với phép toán đa thức đa luồng sử dụng Intel AVX2

Hệ điều hành **:** Ubuntu 22.04 LTS

3.3. Cấu hình tham số mã hóa

Lược đồ CKKS được cấu hình nhằm đạt mức bảo mật tương đương 128-bit dựa trên độ khó của bài toán LWE (Albrecht và cộng sự, 2015). Các tham số chính bao gồm:

Bậc môđun đa thức **:** 8192

Chuỗi mô đun hệ số **:** [60, 40, 40, 40, 40, 60] bit

Tỷ lệ (Δ) **:** 2242^{24}224

Kích thước lô **:** 16 vectơ đặc trưng được mã hóa

Mức độ bảo mật : ≥ 128 bit

Kỹ thuật batching được sử dụng để mã hóa nhiều đặc trưng trong một bản mã, giúp cải thiện hiệu năng xử lý. Các khóa phụ trợ được tạo trước nhằm giảm chi phí tính toán trong quá trình suy luận.

3.4. Các chỉ số đánh giá

Các chỉ số sau đây đã được đo lường để đánh giá hiệu suất hệ thống:

1. Thời gian mã hóa/giải mã (EDT) trên thiết bị di động;

2. Thời gian suy luận được mã hóa (IT) trên máy chủ;

3. Độ trễ đầu cuối (EEL) bao gồm độ trễ mạng qua Wifi;

4. Tốc độ hao pin (BDR) được đo bằng Qualcomm Trepn Profiler;

5. Mức sử dụng băng thông (BW) cho việc truyền dữ liệu được mã hóa.

Mỗi thí nghiệm được lặp lại 20 lần và lấy giá trị trung bình trên hai cấu hình mạng: Wi-Fi 6 (mạng LAN tại nhà, 300Mbps) và 4G LTE (mạng của nhà mạng, 30 Mbps). Các mẫu giọng nói có độ dài từ 2 đến 4 giây cho mỗi câu nói.

3.5. Mô hình mối đe dọa

Hệ thống giả định một máy chủ trung thực, thực hiện chính xác các phép tính mã hóa nhưng có thể cố gắng suy luận thông tin nhạy cảm từ văn bản mã hóa. Các thiết bị di động được coi là an toàn và giữ quyền truy cập độc quyền vào các khóa giải mã riêng tư. Dữ liệu được truyền độc quyền qua các kênh được mã hóa TLS với tính bảo mật chuyển tiếp. Kết quả là, tính bảo mật dữ liệu được duy trì ngay cả khi máy chủ bị xâm phạm hoặc bị giám sát mạng thụ động, phù hợp với mục tiêu tính toán thuê ngoài an toàn (Popa và cộng sự, 2011).

3.6. Chiến lược tối ưu hóa cho HE trên thiết bị di động

Để giảm chi phí tính toán và tiêu thụ năng lượng, nghiên cứu áp dụng một số chiến lược:

Batching: khả năng đóng gói của CKKS để xử lý nhiều dữ liệu trong một bản mã;

Xấp xỉ đa thức: thay thế các hàm kích hoạt phi tuyến bằng đa thức bậc thấp để giảm độ sâu mạch;

Phân bổ tính toán: chuyển phần suy luận sang máy chủ để giảm tải cho thiết bị di động;

Nén dữ liệu: giảm kích thước bản mã trước khi truyền tải.

Các chiến lược này giúp cải thiện đáng kể hiệu năng và khả năng triển khai thực tế của hệ thống.

3.7. Phân tích bảo mật

Hệ thống đạt mức bảo mật tương đương 128-bit dựa trên giả định độ khó của bài toán RLWE, một biến thể của LWE (Albrecht và cộng sự, 2015). Điều này đảm bảo rằng các tấn công vét cạn là không khả thi trong thực tế. Tuy nhiên, một số rủi ro vẫn tồn tại, bao gồm rò rỉ thông tin thông qua thời gian xử lý hoặc kích thước bản mã, cũng như các tấn công suy luận mô hình. Do đó, trong các nghiên cứu tương lai, cần kết hợp HE với các kỹ thuật bổ sung như differential privacy hoặc secure aggregation để tăng cường bảo mật.

IV. Kết quả và thảo luận

Nghiên cứu đánh giá tính khả thi của suy luận học máy được mã hóa trên thiết bị di động thông qua các chỉ số về hiệu năng, tiêu thụ năng lượng và chi phí truyền thông. Các kết quả được thu thập từ hệ thống triển khai thực tế, với dữ liệu giọng nói từ LibriSpeech (Panayotov và cộng sự, 2015).

4.1. Hiệu năng mã hóa và giải mã

Kết quả cho thấy thời gian mã hóa tăng gần tuyến tính theo độ dài vectơ đầu vào, dao động từ khoảng 600 ms đến hơn 1 giây đối với các tín hiệu từ 2 đến 4 giây. Ngược lại, thời gian giải mã thấp hơn đáng kể do không yêu cầu các phép toán phức tạp.

Bảng 1: Thời gian mã hóa và giải mã trên thiết bị di động (Galaxy S21, CKKS)

Thời lượng đầu vào	Độ dài vectơ	Thời gian mã hóa (ms)	Thời gian giải mã (ms)
2 giây	128	615	142
3 giây	192	878	203
4 giây	256	1123	266

Xu hướng này phù hợp với các nghiên cứu trước, trong đó mã hóa được xác định là thành phần tốn kém nhất khi triển khai HE trên thiết bị di động (Chen và cộng sự, 2020). Điều này cho thấy mã hóa là nút thắt cổ chai chính trong hệ thống.

4.2. Hiệu năng suy luận (Phía máy chủ)

Thời gian suy luận trên máy chủ duy trì dưới 400 ms trong mọi trường hợp, cho thấy khả năng xử lý hiệu quả của mô hình CNN đã được tối ưu hóa cho HE. Việc sử dụng các hàm kích hoạt đa thức giúp giảm độ sâu tính toán và tránh nhu cầu bootstrapping.

Bảng 2: Thời gian suy luận được mã hóa trên máy chủ

Độ dài vectơ	Các phép toán đồng cấu	Thời gian suy luận (ms)	Mức sử dụng CPU (%)
128	~5k	198	42
192	~7.4k	272	54
256	~10 nghìn	347	63

Kết quả này phù hợp với các nghiên cứu trước đó, trong đó các mô hình tương thích HE có thể đạt hiệu năng gần thời gian thực khi triển khai trên hạ tầng máy chủ (Bourse và cộng sự, 2018; Boemer và cộng sự, 2019).

4.3. Chi phí mạng và mức độ sử dụng băng thông

Chúng tôi đã đo kích thước bản mã trước và sau khi nén Brotli, và ước tính độ trễ truyền tải mạng qua Wi-Fi 6 (300 Mbps) và LTE (30 Mbps).

Bảng 3: Kích thước bản mã và thời gian truyền tải

Độ dài vectơ	Kích thước văn bản mã hóa (thô)	Kích thước nén	Thời gian Wi-Fi (ms)	Thời gian LTE (ms)
128	81 KB	41 KB	~1.1	~11.1
192	121 KB	62 KB	~1,6	~17.0
256	161 KB	81 KB	~2.1	~22,2

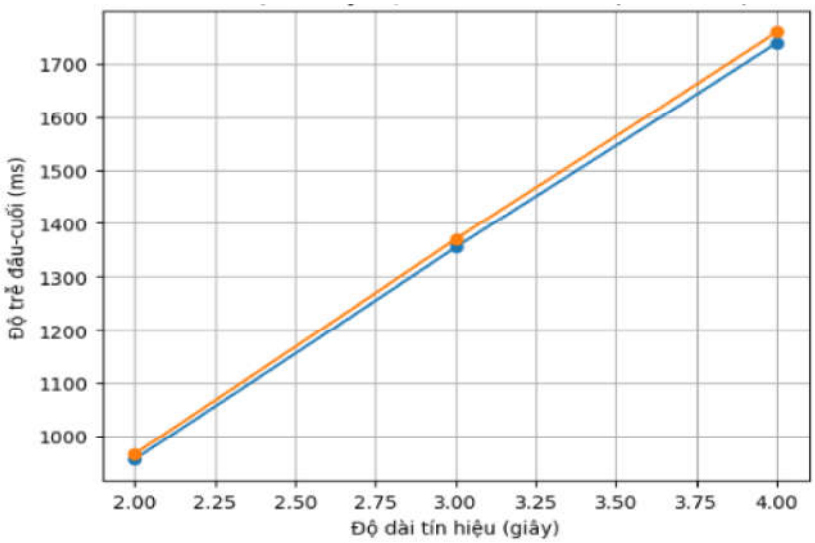
Các bản mã đã nén vẫn nằm trong giới hạn băng thông chấp nhận được cho việc suy luận thời gian thực, ngay cả trong điều kiện LTE. Độ trễ khứ hồi tổng thể-bao gồm mã hóa, truyền tải, suy luận và giải mã-trung bình dưới 1,5 giây trên Wi-Fi cho tất cả các độ dài đầu vào.

4.4. Đánh giá tổng thể hiệu năng hệ thống

Để đánh giá toàn diện hiệu năng của hệ thống suy luận mã hóa, chúng tôi tiến hành tổng hợp các thành phần độ trễ chính trong toàn bộ pipeline xử lý, bao gồm: (i) thời gian mã hóa trên thiết bị di động, (ii) thời gian truyền tải dữ liệu qua mạng, (iii) thời gian suy luận đồng hình trên máy chủ, và (iv) thời gian giải mã kết quả. Tổng độ trễ đầu-cuối (end-to-end latency - EEL) được xác định theo công thức:

$$EEL = T_{enc} + T_{net} + T_{inf} + T_{dec}$$

trong đó T_{enc}, T_{net}, T_{inf}, và T_{dec} lần lượt là thời gian mã hóa, truyền tải, suy luận và giải mã.



Hình 2: Độ trễ đầu-cuối theo độ dài tín hiệu với điều kiện mạng Wifi và LTE

Kết quả này cho thấy rằng các thành phần chính ảnh hưởng đến hiệu năng là thời gian mã hóa và suy luận, trong khi độ trễ mạng chỉ đóng vai trò thứ yếu. Điều này củng cố giả thuyết rằng việc tối ưu hóa thuật toán HE và thiết kế mô hình là yếu tố quyết định trong việc cải thiện hiệu năng hệ thống. Sự suy giảm này chủ yếu do việc sử dụng các hàm kích hoạt xấp xỉ đa thức, phù hợp với các ràng buộc của HE (Bourse và cộng sự, 2018). Tuy nhiên, mức giảm này được xem là chấp nhận được trong bối cảnh bảo vệ quyền riêng tư.

4.5. Mức tiêu thụ năng lượng và khả năng sử dụng

Sử dụng Qualcomm Trepn Profiler, chúng tôi đã theo dõi mức tiêu thụ năng

lượng trong quá trình mã hóa, giải mã và trạng thái chờ.

Mức tiêu thụ điện năng trung bình của CPU trong quá trình mã hóa: 3,2 W

Mức hao pin mỗi lần mã hóa mẫu: 1,4%

Ước tính mức sử dụng pin cho 10 lần suy luận: ~15% pin (khi sạc đầy)

Kết quả đo lường cho thấy quá trình mã hóa tiêu thụ năng lượng đáng kể trên thiết bị di động, nhưng ở mức chấp nhận được. Những phát hiện này phù hợp với các nghiên cứu trước đây, trong đó HE được xác định là kỹ thuật tiêu tốn tài nguyên và năng lượng khi triển khai trên thiết bị cầm tay (Chen và cộng sự, 2020). Do đó, việc tối ưu hóa năng lượng hoặc sử dụng phần cứng chuyên dụng là cần thiết để cải thiện khả năng ứng dụng thực tế.

4.6. Tóm tắt và ý nghĩa

Kết quả thực nghiệm cho thấy kiến trúc di động-đám mây kết hợp với CKKS là khả thi đối với các mô hình học máy quy mô nhỏ. Tuy nhiên, vẫn tồn tại một số hạn chế, bao gồm chi phí mã hóa cao, tiêu thụ năng lượng lớn và ràng buộc thiết kế mô hình.

Những kết quả này cho thấy sự đánh đổi rõ ràng giữa bảo mật, hiệu năng và độ chính xác trong các hệ thống AI bảo vệ quyền riêng tư. Trong tương lai, các cải tiến về thuật toán HE, tối ưu hóa mô hình và hỗ trợ phần cứng có thể giúp giảm các hạn chế này và mở rộng khả năng ứng dụng thực tế (Boemer và cộng sự, 2019).

V. Kết luận

Bài báo đã đánh giá tính khả thi của việc triển khai mã hóa đồng hình CKKS trong suy luận học máy bảo vệ quyền riêng tư trên thiết bị di động thông qua kiến trúc di động-đám mây lai. Kết quả thực nghiệm cho thấy suy luận mã hóa có thể đạt độ trễ gần thời gian thực (dưới 1,5 giây trên Wi-Fi và dưới 2 giây trên LTE) với mức tiêu thụ năng lượng chấp nhận được, khẳng định tiềm năng ứng dụng của HE trong các hệ thống AI di động.

Tuy nhiên, quá trình mã hóa trên thiết bị vẫn là nút thắt cổ chai, cùng với các hạn chế về năng lượng và ràng buộc thiết kế mô hình. Các kết quả cũng cho thấy sự đánh đổi giữa bảo mật, hiệu năng và độ chính xác.

Trong tương lai, các nghiên cứu cần tập trung vào tối ưu hóa thuật toán HE, thiết kế mô hình thân thiện với HE và khai thác tăng tốc phần cứng, nhằm thúc đẩy triển khai các ứng dụng AI bảo vệ quyền riêng tư trên thiết bị di động.

Lời cảm ơn. Công trình này được hỗ trợ một phần bởi dự án khoa học mã số MHN-2025-02.23, do Đại học Mở Hà Nội tài trợ.

Tài liệu tham khảo

Sahai, A., & Waters, B. (2005). Fuzzy identity-based encryption. In R. Cramer (Ed.), *Advances in Cryptology - EUROCRYPT 2005* (Lecture Notes in Computer Science, Vol. 3494, pp. 457-473). Springer. https://doi.org/10.1007/11426639_27

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC)* (pp. 169-178). ACM. https://doi.org/10.1145/1536414.1536440

Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In T. Takagi & T. Peyrin (Eds.), *Advances in Cryptology - ASIACRYPT 2017* (Lecture Notes in Computer Science, Vol. 10624, pp. 409-437). Springer. https://doi.org/10.1007/978-3-319-70694-8_15

Kim, M., Song, Y., Wang, S., Xia, Y., & Wang, X. (2020). Secure and practical linear programming outsourcing in cloud computing. *IEEE Transactions on Cloud Computing*, 8(1), 1-14. https://doi.org/10.1109/TCC.2018.2822720

Microsoft Research. (2023). *Microsoft SEAL (Version 4.x) [Computer software]*. https://github.com/microsoft/SEAL

Laine, K., Lauter, K., & Naehrig, M. (2017). *Microsoft SEAL: A simple encrypted arithmetic library* (MSR-TR-2017-39). Microsoft Research.

Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2014). (Leveled) fully homomorphic encryption without bootstrapping. *ACM Transactions on Computation Theory*, 6(3), Article 13. https://doi.org/10.1145/2633600

Fan, J., & Vercauteren, F. (2012). Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012, 144. https://eprint.iacr.org/2012/144

Cheon, J. H., Kim, A., Kim, M., & Song, Y. (2018). Bootstrapping for approximate homomorphic encryption. In J. B. Nielsen & V. Rijmen (Eds.), *Advances in Cryptology - EUROCRYPT 2018* (Lecture Notes in Computer Science, Vol. 10820, pp. 360-384). Springer. https://doi.org/10.1007/978-3-319-78381-9_13

Halevi, S., & Shoup, V. (2020). *HElib: An implementation of homomorphic encryption [Computer software]*. IBM Research. https://github.com/homenc/HElib

Polyakov, Y., Rohloff, K., & Ryan, G. (2021). *PALISADE lattice cryptography library [Computer software]*. Duality Technologies. https://gitlab.com/palisade/palisade-release

Lattigo Project. (2023). *Lattigo: Lattice-based cryptography library in Go [Computer software]*. https://github.com/ldsec/lattigo

Chen, Z., Sahai, A., & Song, S. (2020). Evaluating homomorphic encryption on mobile devices. In *Proceedings of the IEEE Symposium on Security and Privacy Workshops (SPW)*. https://doi.org/10.1109/SPW50608.2020.00030

Lu, Y., & Zhang, Y. (2019). Privacy-preserving deep learning inference on mobile devices. In *Proceedings of the ACM Workshop on Privacy in the Electronic Society (WPES)* (pp. 71-80). ACM. https://doi.org/10.1145/3338498.3358649

Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2016). CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)* (pp. 201-210).

Bourse, F., Minelli, M., Minihold, M., & Paillier, P. (2018). Fast homomorphic evaluation of deep discrete neural networks. In H. Shacham & A. Boldyreva (Eds.), *Advances in Cryptology - CRYPTO 2018* (Lecture Notes in Computer Science, Vol. 10991, pp. 483-512). Springer. https://doi.org/10.1007/978-3-319-96878-0_16

Boemer, F., Cammarota, R., & Wierzynski, R. C. (2019). nGraph-HE2: A high-throughput framework for neural network inference on encrypted data. In *Proceedings of the 7th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). LibriSpeech: An ASR corpus based on public domain audio books. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 5206-5210). https://doi.org/10.1109/ICASSP.2015.7178964

Albrecht, M. R., Player, R., & Scott, S. (2015). On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3), 169-203. https://doi.org/10.1515/jmc-2015-0016

Popa, R. A., Redfield, C., Zeldovich, N., & Balakrishnan, H. (2011). CryptDB: Protecting confidentiality with encrypted query processing. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP)* (pp. 85-100). https://doi.org/10.1145/2043556.2043566

MOBILE DEVICE-BASED ENCRYPTION-ASSISTED SCREEN WITH SECURITY METHODS IN APPLIED MACHINE LEARNING

Thai Thanh Tung[1], Vu Xuan Hanh[1], Le Huu Nam[1],
Tran Duy Hung[1], Dang Thuy Linh[1]

Abstract**:** *Homomorphic encryption (HE) enables secure computation on encrypted data without exposing sensitive information, making it a promising approach for privacy-preserving machine learning (PPML). However, its high computational cost remains a major barrier to deployment on resource-constrained mobile devices. This paper evaluates the feasibility of the CKKS homomorphic encryption scheme for encrypted speech inference on smartphones within a hybrid mobile-cloud architecture. A three-layer convolutional neural network (CNN) compatible with HE is implemented and evaluated using the LibriSpeech dataset. Experimental results show that end-to-end encrypted inference can be achieved with latency below 1.5 seconds over Wi-Fi and under 2 seconds over LTE, with acceptable energy consumption. Although on-device encryption remains the primary bottleneck, optimization strategies such as batching, model approximation, and computation offloading demonstrate the practical potential of HE-based AI applications on mobile devices.*

Keywords: *Homomorphic encryption, CKKS, Privacy-preserving machine learning, Encrypted inference, Mobile devices, Mobile-cloud computing*

[1] Hanoi Open University, Hanoi, Vietnam