Transport and Communications Science Journal

IMPLEMENTING THE BLAKE3 HASH ALGORITHM ON A RASPBERRY PI DEVELOPMENT BOARD

Dam Duc Trung[1], Dang Tuan Anh[2*]

[1]Vietnam Defense Industry, No. 51 Phu Dien, Phu Dien Ward, Bac Tu Liem, Hanoi, Vietnam

[2]Academy of Cryptography Techniques, No. 141 Chien Thang, Tan Trieu, Thanh Tri, Hanoi, Vietnam

ARTICLE INFO

TYPE: Research Article
Received: 03/03/2025
Revised: 12/04/2025
Accepted: 10/06/2025
Published online: 15/06/20225
https://doi.org/10.47869/tcsj.76.5.5

[*] *Corresponding author*
Email: tuananh4796@gmail.com

Abstract. The rapid development and increasing convergence of blockchain technology and the Internet of Things (IoT) have imposed growing demands on processing efficiency, system reliability, and data security in distributed systems. In this context, the selection and optimization of cryptographic hash functions play a crucial role, with BLAKE3 emerging as a promising candidate due to its high computational speed, low resource consumption, and strong cryptographic security. Designed to optimize performance across various hardware platforms, BLAKE3 demonstrates significant advantages. In this study, we implement the BLAKE3 hashing algorithm on a Raspberry Pi development board running Raspberry Pi OS using the Python programming language. The execution time of the algorithm is evaluated by measuring the hashing duration of input strings of varying lengths. Experimental results indicate that BLAKE3 achieves consistent and superior performance compared to traditional hash functions such as SHA-256 and SHA-3, particularly on devices with higher processing capabilities. This study provides an overview of the feasibility of BLAKE3 in embedded environments and lays the groundwork for future optimization research.

Keywords: BLAKE3, Blockchain, Hash algorithm, IoT, Merkle tree structure, Raspberry Pi

Tạp chí Khoa học Giao thông vận tải

THỰC THI THUẬT TOÁN BĂM BLAKE3 TRÊN BO MẠCH PHÁT TRIỂN RASPBERRY PI

Đàm Đức Trung[1], Đặng Tuấn Anh[2*]

[1]Viện Vũ khí – Tổng cục Công nghiệp Quốc phòng, Số 51 Phú Diễn, Phường Phú Diễn, Bắc Từ Liêm, Hà Nội, Việt Nam

[2]Học viện Kỹ thuật mật mã, Số 141 Chiến Thắng, Tân Triều, Thanh Trì, Hà Nội, Việt Nam

THÔNG TIN BÀI BÁO

CHUYÊN MỤC: Công trình khoa học
Ngày nhận bài: 03/03/2025
Ngày nhận bài sửa: 12/04/2025
Ngày chấp nhận đăng: 10/06/2025
Ngày xuất bản Online: 15/06/2025
https://doi.org/10.47869/tcsj.76.5.5

** Tác giả liên hệ*
Email: tuananh4796@gmail.com

Tóm tắt. Sự phát triển mạnh mẽ và xu hướng hội tụ giữa công nghệ blockchain và Internet vạn vật đã đặt ra những yêu cầu ngày càng cao về hiệu suất xử lý, độ tin cậy và bảo mật trong các hệ thống phân tán. Trong bối cảnh đó, việc lựa chọn và tối ưu hóa thuật toán băm đóng vai trò then chốt, với BLAKE3 được đề xuất như một giải pháp tiềm năng nhờ đặc tính tốc độ tính toán nhanh, tiêu tốn ít tài nguyên và đảm bảo mức độ an toàn mật mã cao, được thiết kế để tối ưu hóa tốc độ trên nhiều nền tảng phần cứng khác nhau. Trong nghiên cứu này, nhóm tác giả tiến hành cài đặt thuật toán băm BLAKE3 trên bo mạch phát triển Raspberry Pi sử dụng hệ điều hành Raspberry Pi OS với ngôn ngữ lập trình Python và đánh giá thời gian thực thi của thuật toán bằng cách đo lường thời gian băm của các chuỗi ký tự có độ dài khác nhau. Kết quả cho thấy BLAKE3 có tốc độ băm ổn định và vượt trội so với các thuật toán băm truyền thống như SHA-256 và SHA-3, đặc biệt trên các thiết bị có khả năng xử lý tốt hơn. Nghiên cứu này cung cấp cái nhìn tổng quan về tính khả thi của BLAKE3 trong môi trường nhúng và có thể là tiền đề cho các nghiên cứu tối ưu hóa tiếp theo.

Từ khóa: BLAKE3, Blockchain, IoT, cấu trúc cây Merkle, Raspberry Pi, thuật toán băm

1. GIỚI THIỆU

Trong kỷ nguyên số và thời đại bùng nổ thông tin, nhu cầu bảo mật dữ liệu trở nên quan trọng hơn bao giờ hết, cùng với sự gia tăng mạnh mẽ của các thiết bị kết nối Internet, từ hệ

thống máy chủ đến các thiết bị nhúng và Internet vạn vật (IoT), các thuật toán mật mã phải đảm bảo tính toàn vẹn và an toàn dữ liệu trong khi vẫn duy trì hiệu suất cao [1, 2]. Đặc biệt, trong các hệ thống tài nguyên hạn chế như thiết bị nhúng, việc lựa chọn thuật toán băm có tốc độ cao nhưng vẫn đảm bảo tính bảo mật là một thách thức quan trọng [2]. Các thuật toán băm (hashing algorithm) là một công cụ quan trọng trong việc bảo vệ tính toàn vẹn và xác thực của dữ liệu [3], được sử dụng rộng rãi trong các ứng dụng như đảm bảo tính toàn vẹn và xác thực, lưu trữ dữ liệu an toàn. Trong bối cảnh này, các thuật toán băm truyền thống như SHA-256 hay SHA-3 vẫn được sử dụng rộng rãi, nhưng chúng bộc lộ những hạn chế về hiệu suất, đặc biệt khi triển khai trên các thiết bị có phần cứng hạn chế tài nguyên [3-5]. Do đó, các thuật toán băm hiện đại như BLAKE3 đã được đề xuất nhằm tối ưu hóa tốc độ và khả năng ứng dụng trong các môi trường tính toán khác nhau.

BLAKE3 là một thuật toán băm thế hệ mới, được thiết kế để có tốc độ xử lý vượt trội trong khi vẫn duy trì tính bảo mật mạnh mẽ [6, 7]. Với cấu trúc băm dạng cây và khả năng tận dụng các kỹ thuật tối ưu hóa như SIMD và đa luồng, BLAKE3 có thể đạt tốc độ cao hơn đáng kể so với các thuật toán băm truyền thống, đặc biệt trên các nền tảng phần cứng hiện đại [7]. So với SHA-256 hay SHA-3, BLAKE3 không chỉ nhanh hơn mà còn có khả năng mở rộng linh hoạt nhờ vào cơ chế xuất đầu ra có độ dài tùy biến. Trong thử nghiệm tạo băm cho tệp 16 KB, BLAKE3 với khóa 256 bit vượt trội hơn 17 lần so với SHA3-256, 14 lần với SHA-256, 9 lần với SHA-512, 6 lần so với SHA-1 và với BLAKE2b đến 5 lần [5, 7-11]. Điều này làm cho BLAKE3 trở thành một lựa chọn hấp dẫn cho các ứng dụng yêu cầu hiệu suất cao như hệ thống nhúng, mã hóa dữ liệu và blockchain. Tuy nhiên, hiệu suất thực tế của BLAKE3 trên các thiết bị có tài nguyên hạn chế như Raspberry Pi vẫn cần được nghiên cứu để đánh giá tính khả thi của nó trong môi trường nhúng.

Trong nghiên cứu này, nhóm tác giả thực hiện cài đặt thuật toán BLAKE3 trên các bo mạch phát triển Raspberry Pi 3B+, Raspberry Pi 4, một máy tính nhỏ gọn có khả năng xử lý đầy đủ các tác vụ cần thiết cho các ứng dụng bảo mật và mã hóa. Mục tiêu chính là đo lường thời gian băm của các chuỗi ký tự có độ dài khác nhau nhằm phân tích hiệu suất của BLAKE3 trên các nền tảng phần cứng khác nhau. Raspberry Pi, với khả năng xử lý hạn chế nhưng lại rất phổ biến trong các ứng dụng nhúng và IoT, sẽ là nền tảng lý tưởng để kiểm tra khả năng triển khai thuật toán băm này. Kết quả thu được sẽ cung cấp cái nhìn tổng quan về khả năng triển khai BLAKE3 trên các hệ thống nhúng, đồng thời làm cơ sở cho các nghiên cứu tiếp theo về tối ưu hóa thuật toán băm trong môi trường tài nguyên hạn chế.

2. CƠ SỞ LÝ THUYẾT CỦA THUẬT TOÁN BĂM BLAKE3

Thuật toán băm BLAKE3 đã có nhiều sự cải tiến trong thuật toán, bằng cách sử dụng cấu trúc cây (Merkle tree structure) và các hàm toán học dựa trên ChaCha để đạt được hiệu suất cao hơn [4]. Trong nghiên cứu này, nhóm tác giả tập trung khai thác các bước chính của thuật toán BLAKE3 bao gồm 4 bước: (1) Bước khởi tạo; (2) Bước phân đoạn; (3) Bước băm cây; (4) Đầu ra.

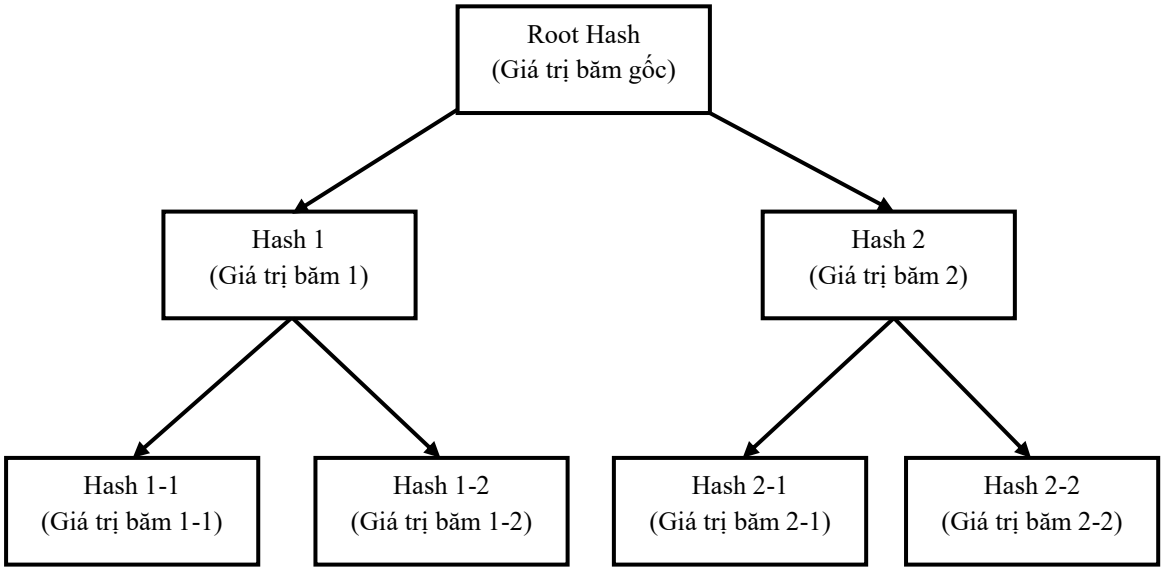
- *Khởi tạo (Initialization):* Bước này sẽ khởi tạo các giá trị băm với các giá trị cố định và các tham số cấu hình h0, h1, h2, ..., h7 trong thuật toán băm BLAKE3.

- *Phân đoạn (Chunking):* Bước thứ hai sẽ chia dữ liệu đầu vào thành các đoạn nhỏ gọi là chunks. Mỗi chunk được xử lý độc lập, giúp tối ưu hóa quá trình song song hóa.

- *Băm cây (Tree Hashing):* Trong bước này, mỗi chunk được băm thành các giá trị băm riêng lẻ và sau đó kết hợp lại trong cấu trúc cây để tạo ra giá trị băm cuối cùng. Quá trình này bao gồm nhiều vòng lặp với các phép toán XOR, cộng, và xoay bit (sử dụng trong quá trình này chính là thuật toán hàm G trong BLAKE2).

- *Đầu ra (Output):* Bước thứ tư giá trị băm cuối cùng được tạo ra từ các giá trị băm của các chunks, đảm bảo tính toàn vẹn và bảo mật của dữ liệu.

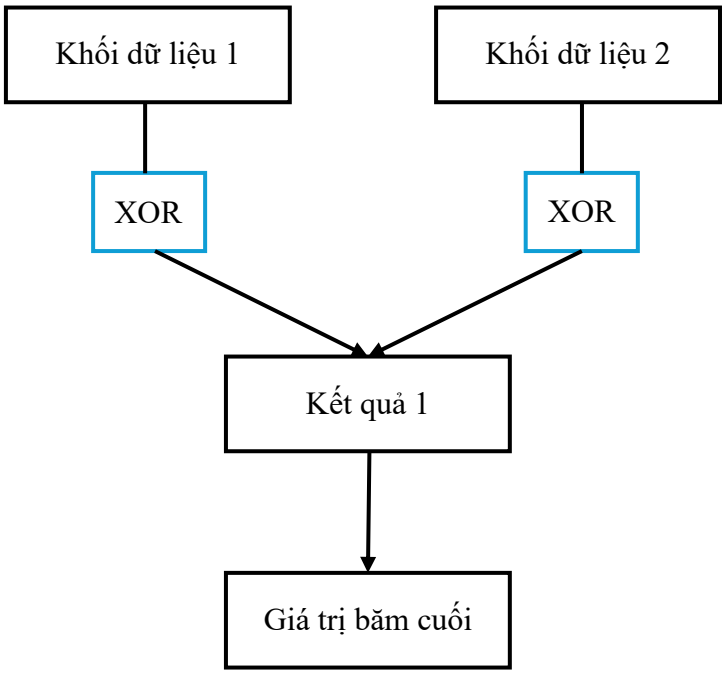
Cây Merkle trong thuật toán băm BLAKE3 là một cấu trúc dữ liệu cây trong mật mã học được sử dụng để tổng hợp và xác minh tính toàn vẹn của dữ liệu [6]. Mỗi lá của cây chứa một khối dữ liệu, và mỗi nút không phải lá là giá trị băm của các con của nó. Cây Merkle là cơ sở cho cách BLAKE3 tổ chức và xử lý dữ liệu một cách song song. Điều này cho phép tận dụng toàn bộ tài nguyên của các hệ thống đa nhân (multicore) và đa luồng (multithread), từ đó tăng tốc độ tính toán băm. Sau khi tất cả các khối dữ liệu đã được băm, các giá trị băm được kết hợp lại từng bước ở các cấp độ cao hơn của cây Merkle, giúp cải thiện hiệu suất đáng kể so với các thuật toán băm truyền thống xử lý theo chuỗi tuần tự. Hình 1 thể hiện biểu đồ của cây Merkle.



Hình 1. Biểu đồ một cấu trúc cây Merkle đơn giản.

Trong thuật toán băm BLAKE3 sử dụng phép toán XOR (Exclusive OR) để kết hợp các khối dữ liệu. XOR là một phép toán quan trọng trong mật mã học vì tính chất đối xứng của nó. Việc phép XOR hai giá trị có thể đảo ngược quá trình bằng cách sử dụng XOR lại với giá trị ban đầu. BLAKE3 sử dụng phép toán XOR, cộng số học, và quay vòng các bit (bitwise rotation) [12] trong quá trình băm. Cơ chế chính của BLAKE3 dựa trên cấu trúc cây Merkle, cho phép băm từng khối nhỏ của dữ liệu và kết hợp lại thành một giá trị băm cuối cùng. Hình 2 thể hiện phép XOR được thực hiện trong thuật toán.

Thuật toán băm BLAKE3 hoạt động dựa trên một loạt các bước tính toán như sau [7]: (1) Bước đầu tiên thuật toán sẽ *phân chia dữ liệu thành các khối*, dữ liệu đầu vào được chia thành các khối nhỏ có kích thước cố định, thường là 64 byte (512 bit); (2) Thuật toán sẽ *tính toán hàm nén (Compression Function)*, hàm nén này sử dụng nhiều vòng tính toán bao gồm các phép XOR, phép cộng modulo 32 và các phép dịch vòng, các phép toán này được thực hiện trên các từ 32-bit trong các khối dữ liệu, giúp tạo ra tính phi tuyến và khuếch tán trong giá trị băm; (3) *Sử dụng cấu trúc cây Merkle*, ở đây giá trị băm của từng khối dữ liệu được kết hợp lại thành cây Merkle để tạo ra giá trị băm cuối cùng của toàn bộ dữ liệu, cho phép xử lý song song và giúp tăng tốc độ tính toán trong thuật toán băm BLAKE3.

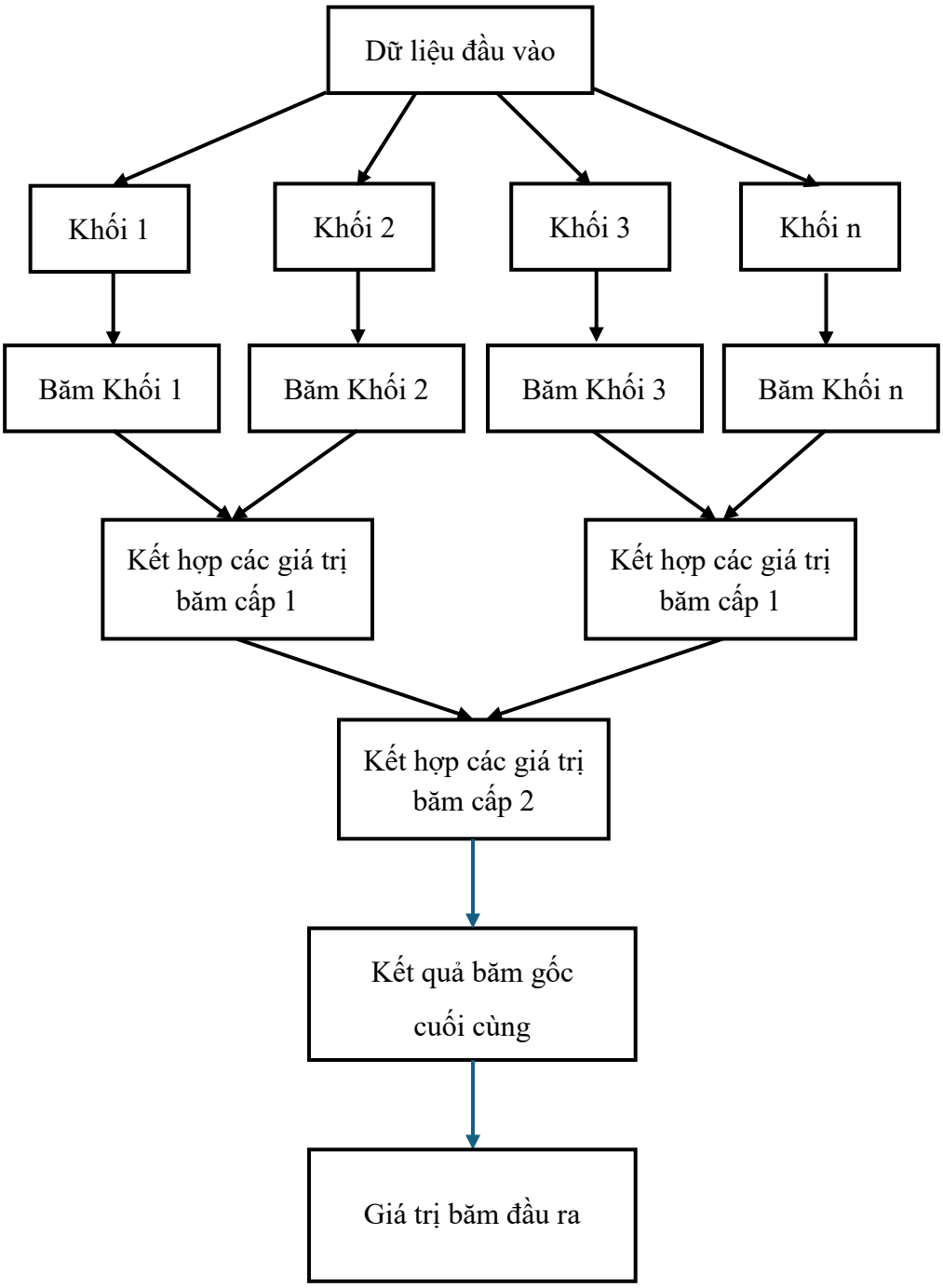


Hình 2. Phép XOR trong thuật toán băm.

Một số vấn đề an toàn và hiệu suất của thuật toán băm BLAKE3 được thể hiện thông qua một số điểm sau [6-8, 10-13]: (1) *Tính kháng va chạm:* BLAKE3 sử dụng cây Merkle và các phép toán như XOR, cộng modulo, và quay bit, giúp giảm khả năng va chạm so với SHA-256, SHA-3, và BLAKE2, đồng thời hỗ trợ tính toán song song; (2) *Tính chống dịch ngược:* Các phép quay bit và cộng số học trong BLAKE3 tạo ra các biến đổi không thể dự đoán, giúp chống lại tấn công tìm kiếm ngược. So với SHA-256 và SHA-3, BLAKE3 đạt tính xác thực tương tự nhưng với hiệu suất cao hơn; (3) *Tính băm hướng đối tượng dữ liệu:* Nhờ cấu trúc cây Merkle, BLAKE3 có thể xử lý dữ liệu lớn nhanh chóng qua song song hóa, vượt trội hơn SHA-256 và SHA-3 vốn xử lý tuần tự; (4) *Tính toàn vẹn dữ diệu:* Mọi thay đổi nhỏ trong dữ liệu đều ảnh hưởng đến giá trị băm cuối cùng, giúp phát hiện thay đổi dễ dàng, đặc biệt hiệu quả nhờ cấu trúc cây Merkle; (5) *Tính hiệu quả:* BLAKE3 đạt hiệu suất cao, vượt trội so với SHA-256 và SHA-3 với cấu trúc tối ưu và giảm số vòng tính toán giúp BLAKE3 phù hợp cho các ứng dụng yêu cầu hiệu suất cao như blockchain và bảo mật mạng.

Hàm băm BLAKE3 được thiết kế để tận dụng tối đa khả năng của các hệ thống đa lõi, cho phép tính toán giá trị băm nhanh chóng và hiệu quả. Hơn nữa, các phép toán mật mã học

như XOR, cộng modulo, và quay bit trong BLAKE3 được tối ưu hóa để thực hiện nhanh chóng trên các phần cứng hiện đại, từ đó giảm thiểu độ trễ trong tính toán. Hình 3 thể hiện quá trình thực hiện băm của thuật toán băm BLAKE3.



Hình 3. Minh họa quá trình thuật toán băm BLAKE3.

3. MỘT SỐ CÔNG BỐ ĐÁNH GIÁ AN TOÀN CỦA THUẬT TOÁN BĂM BLAKE3

Thuật toán BLAKE3 đã thu hút sự chú ý trong cộng đồng nghiên cứu về mật mã học nhờ vào hiệu suất vượt trội và khả năng bảo mật mạnh mẽ so với các thuật toán băm truyền thống. Công bố [6] giới thiệu BLAKE3 như một thuật toán băm mới với cấu trúc cây và các kỹ thuật tối ưu hóa như SIMD và đa luồng, giúp nâng cao tốc độ xử lý. Các tác giả cho rằng BLAKE3 có thể vượt qua các thuật toán băm cũ như SHA-256 và SHA-3 về tốc độ mà không làm giảm tính bảo mật. Đây là nghiên cứu cơ sở cho việc ứng dụng BLAKE3 trong các hệ thống đòi hỏi hiệu suất cao.

Trong công trình [14], các tác giả đã phân tích tổn thất công suất động được tạo ra bởi các triển khai phần cứng khác nhau của hàm băm BLAKE3, được tính toán bằng các công cụ từ kết quả mô phỏng thời gian sau triển khai của các thiết kế được tích hợp trên thiết bị FPGA Xilinx Spartan-7. Công trình chỉ ra hiệu suất tiêu thụ điện năng của thuật toán này được so sánh với các kết quả tương đương của các triển khai lặp của BLAKE2 (thuật toán tiền nhiệm trực tiếp của BLAKE3) và Keccak, lõi của tiêu chuẩn SHA-3. Kết quả của nghiên cứu này chỉ ra sự khác biệt về nhu cầu công suất của ba thuật toán băm trên khi chúng được triển khai trên nền tảng FPGA, đồng thời minh họa mức tiết kiệm đáng kể có thể đạt được thông qua việc sử dụng kỹ thuật đường ống trong logic vòng lặp.

Trong nghiên cứu [15] các tác giả trình bày một thiết kế phần cứng mới nhằm tăng tốc thuật toán băm BLAKE3 trên kiến trúc RISC-V, tận dụng cả phần mở rộng vector (V1.0) và một tập lệnh BLAKE3 được thiết kế riêng. Cách tiếp cận này khai thác tính song song và sự đơn giản của BLAKE3 để đạt được thông lượng cao và độ trễ thấp, bằng cách triển khai hàm nén lõi thông qua một kiến trúc tập lệnh hỗ trợ phép toán vector và tính toán hằng số. Sự kết hợp giữa phần cứng và phần mềm giúp tăng tốc đáng kể hàm nén, đạt hiệu suất cao hơn gấp 10 lần so với một lõi Xeon 8280 2.7GHz và nhanh hơn 10% so với phiên bản sử dụng AVX512.

Trong công trình của Sugier [16], nhóm tác giả đã triển khai thuật toán BLAKE3 trên FPGA, cụ thể là trên thiết bị Xilinx Spartan-7, nhằm đánh giá hiệu suất của hàm nén cốt lõi của thuật toán. Những thử nghiệm này giúp phân tích sự đánh đổi giữa tốc độ và kích thước phần cứng trong các kiến trúc đề xuất. Ngoài ra, công bố còn so sánh hiệu suất của BLAKE3 với các triển khai phần cứng của BLAKE2 và SHA3-256 (Keccak). Công trình này nhấn mạnh vào tính hiệu quả của BLAKE3 khi được triển khai trên FPGA với kết quả nghiên cứu có thể là cơ sở quan trọng để tối ưu hóa việc triển khai BLAKE3 trên các hệ thống phần cứng, đặc biệt là trong bối cảnh nhu cầu về mã hóa tốc độ cao trên các nền tảng nhúng và IoT ngày càng tăng. Công trình tiếp theo [17] của nhóm tập trung vào việc tối ưu hóa mức tiêu thụ điện năng của thuật toán băm BLAKE3 khi được triển khai trên FPGA, đánh giá một cách tiếp cận mới nhằm giảm tiêu thụ điện năng bằng cách sử dụng các khối DSP chuyên dụng (DSP48E1) để thực hiện phép cộng nhị phân, thay vì sử dụng các phần tử logic thông thường trong mảng lập trình của FPGA. Kết quả cho thấy việc thay thế các bộ cộng trong mảng lập trình bằng các phần tử DSP có thể giảm đáng kể mức tiêu thụ điện năng động của kiến trúc BLAKE3 tiêu chuẩn, đặc biệt là trong các thiết kế không sử dụng pipeline. Tuy nhiên, cách tiếp cận này cũng có những hạn chế khi áp dụng cho một số kiến trúc nhất định.

Một công trình khác [18] tiếp tục phân tích tiêu thụ điện năng của các thuật toán băm BLAKE3, BLAKE2 và Keccak khi được triển khai trên FPGA. Nghiên cứu sử dụng cùng một phương pháp thiết kế và triển khai để đảm bảo điều kiện thử nghiệm đồng nhất, giúp so sánh

chính xác hiệu suất năng lượng của ba thuật toán trên tám dự án khác nhau. Kết quả chỉ ra rằng sự ổn định của tín hiệu có ảnh hưởng đáng kể đến mức tiêu thụ điện năng, tương đương với các yếu tố như kích thước và số lượng tài nguyên logic được sử dụng. Đặc biệt, nghiên cứu phát hiện ra rằng các thuật toán họ BLAKE (bao gồm BLAKE3 và BLAKE2) có mức độ nhiễu tín hiệu (glitches) rất cao trong các vòng băm không sử dụng pipeline. Điều này dẫn đến yêu cầu điện năng cao, làm cho kiến trúc lặp tiêu chuẩn của các thuật toán này trở nên kém hiệu quả khi vận hành ở tần số xung nhịp cao. Công trình cũng nhấn mạnh rằng phân tích được thực hiện trong điều kiện hoạt động liên tục với tải tối đa, tức là pipeline luôn đầy và tạo ra đầu ra mới sau mỗi NR chu kỳ xung nhịp.

Nghiên cứu [19] đã giới thiệu một phương pháp tích hợp nhằm tăng cường bảo mật tệp tin bằng cách kết hợp kỹ thuật loại bỏ dữ liệu trùng lặp (deduplication) với thuật toán mã hóa AES (Advanced Encryption Standard). Cụ thể, nhóm tác giả đề xuất thuật toán băm BLAKE3 được sử dụng để xác định và loại bỏ các khối dữ liệu trùng lặp, giúp tối ưu hóa không gian lưu trữ, đồng thời AES được áp dụng để mã hóa và giải mã các tệp duy nhất, đảm bảo tính bảo mật cao trong quá trình lưu trữ và truyền tải dữ liệu. Ứng dụng của nghiên cứu này đặc biệt phù hợp với các hệ thống lưu trữ đám mây và truyền tải dữ liệu an toàn, nơi cần tối ưu hóa băng thông và không gian lưu trữ mà vẫn đảm bảo mức độ bảo mật cao.

Công bố [20] đề xuất một phương pháp cải tiến thuật toán dẫn xuất khóa dựa trên mật khẩu (PBKDF) bằng cách thay thế hàm nén của Argon2i bằng BLAKE3 và một phiên bản tinh chỉnh của nó. Dựa trên ý tưởng rằng Argon2 có thể được điều chỉnh để hỗ trợ các hàm nén khác, nghiên cứu tập trung vào việc tối ưu hóa hiệu suất của Argon2i trong phần mềm mà vẫn duy trì tính bảo mật. Kết quả cho thấy việc thay thế hàm nén mặc định của Argon2i bằng BLAKE3 giúp tăng tốc đáng kể quá trình tính toán trong phần mềm. Ngoài ra, nhóm tác giả lập luận rằng cách tiếp cận này có thể phù hợp hơn khi triển khai trên phần cứng, đặc biệt là trong các thiết bị có yêu cầu cao về hiệu suất và bảo mật, như các bộ tạo khóa dựa trên mật khẩu.

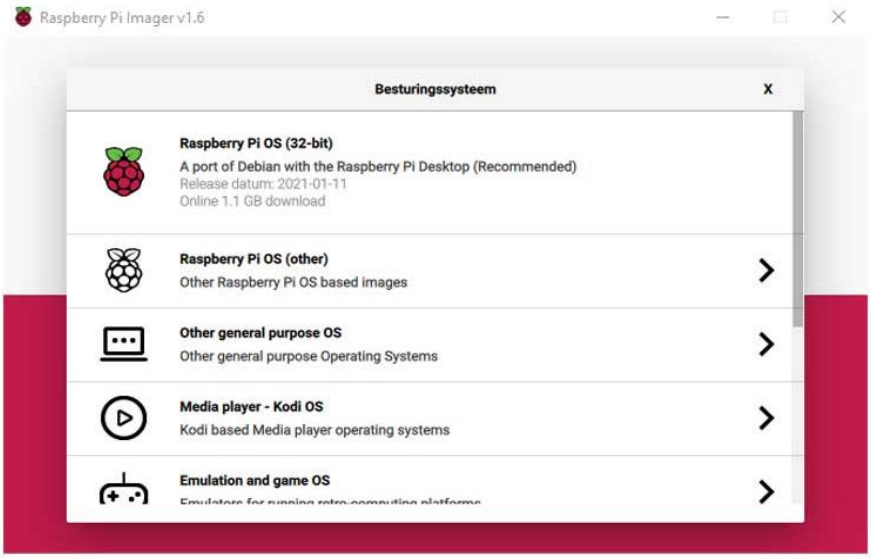
4. KẾT QUẢ VÀ THẢO LUẬN

Để triển khai và thực hiện cài đặt thực thi hàm băm BLAKE3 trên nền tảng phần cứng, nhóm tác giả sử dụng môi trường trên Bo mạch phát triển Raspberry Pi, sử dụng ngôn ngữ lập trình Python, cụ thể như sau: CPU bộ vi xử lý ARM Cortex-A72 64-bit với tốc độ xung nhịp 1.5 GHz, RAM 2GB, sử dụng thẻ nhớ microSD để cài đặt hệ điều hành và lưu trữ các tệp tin liên quan đến mã nguồn và kết quả, hệ điều hành sử dụng Raspberry Pi OS, ngôn ngữ lập trình Python 3 là ngôn ngữ lập trình chính hỗ trợ xử lý hình ảnh và mã hóa, trong đó bao gồm BLAKE3 (Thư viện triển khai thuật toán băm BLAKE3 cho Python); Pillow (Hỗ trợ thao tác với hình ảnh (nếu cần dùng trong quá trình phân tích dữ liệu); Python IDE.

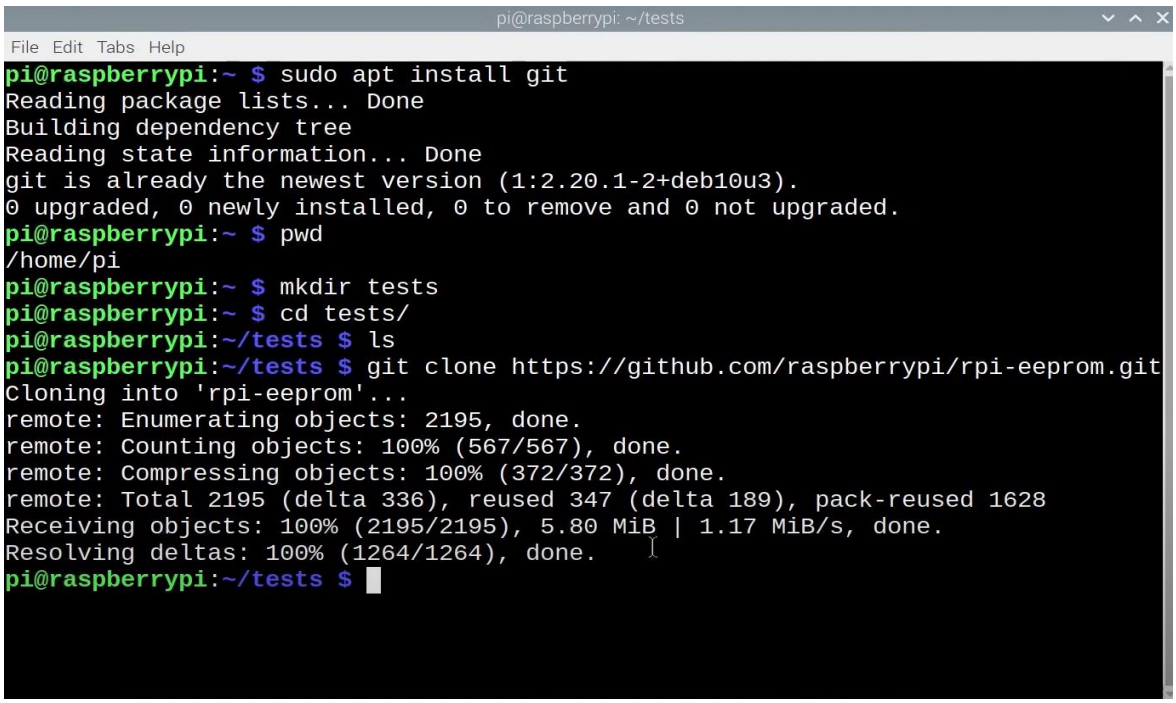
4.1. Luồng hoạt động của thuật toán băm BLAKE3 trên Raspberry Pi

Luồng dữ liệu đầu vào của thuật toán bao gồm 5 bước như sau: (1) ***Nhập dữ liệu đầu vào:*** Nhóm tác giả thực hiện nhập các chuỗi dữ liệu khác nhau nhiều lần để so sánh các kết quả đầu ra; (2) ***Phân đoạn dữ liệu:*** Các dữ liệu đầu vào được chia thành các đoạn nhỏ (chunks) 64 byte (512 bit) để xử lý riêng biệt, tối ưu hóa cho xử lý song song; (3) ***Áp dụng thuật toán băm BLAKE3:*** Mỗi đoạn chunk thông qua nhiều vòng lặp với các phép toán XOR, cộng modulo và quay bit để tạo giá trị băm tạm thời cho từng đoạn; (4) ***Kết hợp dữ liệu với cây Merkle:*** Các giá trị băm của từng đoạn được kết hợp qua cây Merkle, xử lý song song để

tạo giá trị băm cuối cùng; (5) ***Xuất kết quả:*** Giá trị băm cuối cùng được xuất ra, đại diện duy nhất cho dữ liệu đầu vào, đảm bảo tính toàn vẹn và xác thực dữ liệu.



Hình 4. Công cụ Rasberry Pi Imager.



Hình 5. Nạp code vào trong bo mạch Raspberry Pi.

4.2 Kết quả thực nghiệm

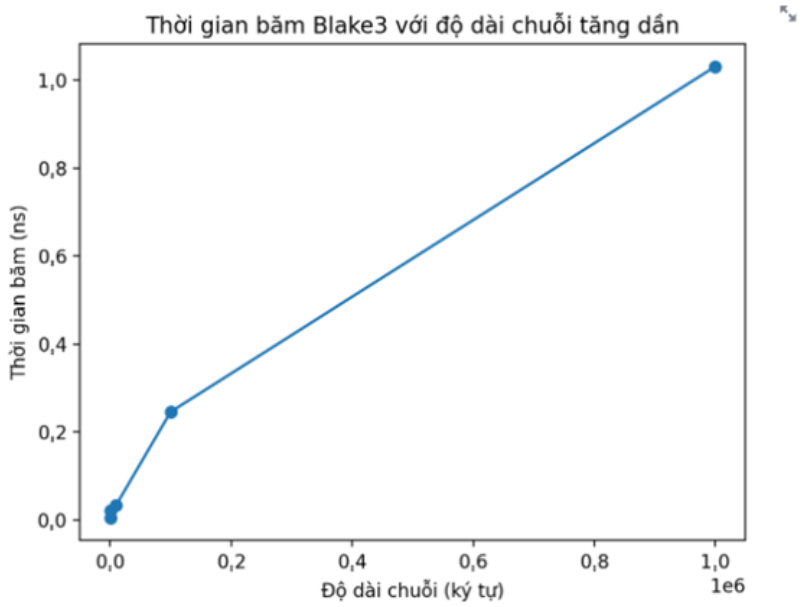
Quá trình thử nghiệm được nhóm tác giả thực hiện với các chuỗi dữ liệu có độ dài khác nhau, từ 100 ký tự đến 1.000.000 ký tự, nhằm đánh giá hiệu suất của thuật toán BLAKE3 khi áp dụng trên bo mạch phát triển Raspberry Pi.

Bảng 1 tóm tắt kết quả đo lường thời gian băm (tính bằng nano giây) cho các chuỗi với độ dài khác nhau.

Bảng 1: Kết quả thực thi của thuật toán băm BLAKE3 trên nền tảng Bo mạch phát triển Raspberry Pi.

Độ dài chuỗi (ký tự)	**Thời gian băm (nano giây)**
100	0,00219
1000	0,0054
10.000	0,0345
100.000	0,2460
1.000.000	1,0304

Kết quả cho thấy thuật toán BLAKE3 hoạt động với tốc độ băm cực nhanh ngay cả khi xử lý chuỗi dữ liệu có kích thước lớn, chẳng hạn với chuỗi dài 1.000.000 ký tự, thời gian băm chỉ mất 1,0304 nano giây. Điều này thể hiện hiệu suất cao của thuật toán, đặc biệt là khi được triển khai trên một nền tảng nhỏ gọn như Raspberry Pi. Với chuỗi có độ dài ngắn hơn (100 ký tự), thời gian băm chỉ mất 0,00219 nano giây, cho thấy thuật toán có khả năng xử lý hiệu quả ngay cả đối với các dữ liệu nhỏ lẻ.



Hình 6. Kết quả thực hiện kiểm thử của BLAKE3.

Qua việc phân tích kết quả, nhóm tác giả nhận định thấy rằng thời gian băm tăng gần như tuyến tính khi độ dài chuỗi tăng dần. Điều này cho thấy tính mở rộng tốt của BLAKE3, đảm bảo rằng thuật toán có thể xử lý các khối dữ liệu lớn mà không gây ra tình trạng tăng đột biến về thời gian thực thi. BLAKE3 đặc biệt hữu ích khi xử lý khối lượng dữ liệu lớn, ví dụ như các hệ thống IoT, blockchain, hoặc các hệ thống yêu cầu băm khối lượng dữ liệu lớn trong thời gian ngắn. Kết quả này là tiền đề cho các công bố nghiên cứu tiếp theo của nhóm tác giả sử dụng thuật toán băm BLAKE3 trên các nền tảng phần cứng khác với số lượng các trường hợp thực thi hàm băm tăng lên cùng độ dài chuỗi đa dạng hơn.

5. KẾT LUẬN

Trong bài báo này, nhóm tác giả đã trình bày nghiên cứu về việc thực thi cài đặt và đánh giá thuật toán băm BLAKE3 trên bo mạch phát triển Raspberry Pi, Kết quả thực nghiệm cho thấy BLAKE3 có thể đạt được hiệu suất băm vượt trội khi so sánh với các thuật toán băm truyền thống như SHA-256 và SHA-3, đặc biệt khi xử lý các chuỗi ký tự có độ dài khác nhau. Các phép đo thời gian băm được thực hiện trên nền tảng Raspberry Pi cho thấy thuật toán này có khả năng tối ưu hóa đáng kể khi triển khai trên các hệ thống phần cứng với tài nguyên hạn chế. Điều này khẳng định rằng BLAKE3 là một lựa chọn khả thi cho các ứng dụng yêu cầu tính hiệu quả và bảo mật cao, đặc biệt trong các hệ thống nhúng và IoT, nơi tài nguyên tính toán và băng thông thường bị giới hạn. Trong thời gian tới, nhóm tác giả sẽ tập trung vào việc cải thiện và tối ưu hóa thuật toán này trong môi trường nhúng, đồng thời mở rộng các ứng dụng của BLAKE3 trong các lĩnh vực như blockchain, bảo mật đám mây và các hệ thống phân tán. Phân tích các tấn công lên BLAKE3 nhằm đảm bảo sự an toàn của các hệ thống sử dụng thuật toán này.

LỜI CẢM ƠN

Nhóm tác giả xin chân thành cảm ơn Học viện Kỹ thuật mật mã đã hỗ trợ trong nghiên cứu này.

TÀI LIỆU THAM KHẢO

[1]. X. Hou, J. Breier, Cryptography and Embedded Systems Security, Springer, 2024. https://doi.org/10.1007/978-3-031-62205-2

[2]. C. Silva, V. A. Cunha, J. P. Barraca, R. L. Aguiar, Analysis of the cryptographic algorithms in IoT communications, Information Systems Frontiers, 26 (2024) 1243-1260. https://doi.org/10.1007/s10796-023-10383-9

[3]. S. R. Prasanna, B. S. Premananda, Performance Analysis of MD5 and SHA-256 Algorithms to Maintain Data Integrity, 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), IEEE, (2021) 246-250. https://doi.org/10.1109/RTEICT52294.2021.9573660

[4]. H. Choi, S. C. Seo, Fast Implementation of SHA-3 in GPU Environment, IEEE Access, 9 (2021) 144574-144586. https://doi.org/10.1109/ACCESS.2021.3122466

[5]. Z. Abdullah Jasim, A. Kadhim Hadi, Optimizing Blockchain Network Performance Using Blake3 Hash Function in POS Consensus Algorithm, IEEE Access, 13 (2025) 44760-44774, https://doi.org/10.1109/ACCESS.2025.3546723

[6]. J. O'Connor, J. P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, BLAKE3: one function, fast everywhere, URL https://github.com/BLAKE3-team/BLAKE3-specs/blob/master/blake3.pdf, 2020, Truy cập tháng 3 năm 2025

[7]. M. Pandya, Performance Evaluation of Hashing Algorithms on Commodity Hardware, arXiv preprint (2024). https://doi.org/10.48550/arXiv.2407.08284

[8]. A. Hłobaż, Analysis of the possibility of using selected hash functions submitted for the SHA-3 competition in the SDEx encryption method, International Journal of Electronics and Telecommunications, 68 (2022) 57-62. https://doi.org/10.24425/ijet.2022.139848

[9]. K. Jang, S. Lim, Y. Oh, H. Kim, A. Baksi, S. Chakraborty, H. Seo, Quantum implementation and analysis of SHA-2 and SHA-3, IEEE Transactions on Emerging Topics in Computing (2025) 1-15. https://doi.org/10.1109/TETC.2025.3546648

[10]. S. Chouhan, G. Sharma, A New Era of Elections: Leveraging Blockchain for Fair and Transparent Voting, arXiv preprint arXiv:2502.16127 (2025). https://doi.org/10.48550/arXiv.2502.16127
[11]. G. Al-Asad, M. Al-Husainy, M. Bani-Hani, A. Al-Zu'bi, S. Albatienh, H. Abuoliem, Comparative Assessment of Hash Functions in Securing Encrypted Images, Engineering, Technology & Applied Science Research, 14 (2024) 18750-5. https://doi.org/10.48084/etasr.8961
[12]. N. Kolomeec, I. Sutormin, D. Bykov, M. Panferov, T. Bonich, On additive differential probabilities of the composition of bitwise exclusive-or and a bit rotation, Cryptography and Communication, 17 (2025) 541-570. https://doi.org/10.1007/s12095-025-00773-y
[13]. N. Soffer, E. Waisbard, An Efficient Hash Function Construction for Sparse Data, Proceedings of the 21st International Conference on Security and Cryptography – SECRYPT, 1 (2024) 698-703. https://doi.org/10.5220/0012764500003767
[14]. J. Sugier, Power Analysis of BLAKE3 Pipelined Implementations in FPGA Devices, International Conference on Dependability and Complex Systems, Springer Nature Switzerland, Springer, Cham (2023) 295-308. https://doi.org/10.1007/978-3-031-37720-4_27
[15]. X. Zou, Y. Peng, T. Li, L. Kong, Z. Pang, L. Zhang, Accelerating Blake3 in RISC-V, 2nd International Conference on Computing, Communication, Perception and Quantum Technology (CCPQT), Xiamen, China, (2023) 296-330. https://doi.org/10.1109/CCPQT60491.2023.00057
[16]. J. Sugier, FPGA Implementations of BLAKE3 Compression Function with Intra-Round Pipelining, International Conference on Dependability and Complex Systems, Cham: Springer International Publishing (2022) 319-330. https://doi.org/10.1007/978-3-031-06746-4_31
[17]. J. Sugier, Dedicated FPGA Resources in Improving Power Efficiency of Implementations of BLAKE3 Hash Function, International Conference on Dependability of Computer Systems, Cham: Springer Nature Switzerland (2024) 283-295. https://doi.org/10.1007/978-3-031-61857-4_28
[18]. J. Sugier, Comparison of power consumption in pipelined implementations of the BLAKE3 cipher in FPGA devices, International Journal of Electronics and Telecommunications, 70 (2024) 23-30. https://doi.org/10.24425/ijet.2023.147710
[19]. K. Tajane, R. Pitale, S. Zambre, H. Huda, A. Utage, V. Dhar, Efficient Cloud Data Deduplication with Blake3 and Secure Transfer using AES, 2024 4th International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, (2024) 572-579. https://doi.org/10.24425/ijet.2023.147710
[20]. I. T. Ciocan, E. A. Kelesidis, D. Maimuț, L. Morogan, A Modified Argon2i Using a Tweaked Variant of Blake3, 26th IEEE Asia-Pacific Conference on Communications (APCC), IEEE (2021) 271-274. https://doi.org/10.1109/APCC49754.2021.9609933