AUTHENTICATION AND ENCRYPTION SOLUTION FOR DOCUMENT STORAGE USING MERKLE HASH TREE, ED25519 DIGITAL SIGNATURE SCHEME AND AES BLOCK CIPHER

Nguyen Van Nghi[1*], Nguyen Van Duan[1], Vo Xuan Luych[1], Vu Ba Linh[2]
[1] *Academy of Cryptography Techniques*
[2]*People's Police University of Technology and Logistics*

ARTICLE INFO

Received: **10/8/2022**
Revised: **16/9/2022**
Published: **16/9/2022**

KEYWORDS

Document storage
Merkle tree
Ed25519
AES
SHA-256

ABSTRACT

The processes of authenticating and encrypting documents are tasks that need to be completed immediately and are of the utmost importance in document storage systems that are currently being utilized by corporations and other organizations. This article proposes a solution to authenticate and secure stored documents using a combination of cryptographic techniques, including Merkle tree, Ed25519 elliptic curve digital signature scheme, and the AES block cipher. The purpose of the suggested solution is to achieve the same level of security as the traditional method, which consists of digitally signing and encrypting each document independently, while also providing improved performance. A theoretical and practical comparison of the Python programming language with well-established methods for cryptography and digital signing has provided the evidence necessary to demonstrate these results. Therefore, it is possible to reach the conclusion that this approach has a significant amount of potential for implementing the security of stored documents in practice.

GIẢI PHÁP XÁC THỰC, BẢO MẬT VĂN BẢN LƯU TRỮ SỬ DỤNG CÂY BĂM MERKLE, LƯỢC ĐỒ KÝ SỐ ED25519 VÀ MÃ KHỐI AES

Nguyễn Văn Nghị[1*], Nguyễn Văn Duẩn[1], Võ Xuân Luých[1], Vũ Bá Linh[2]
[1]*Học viện Kỹ thuật mật mã*
[2]*Trường Đại học Kỹ thuật – Hậu cần Công an nhân dân*

THÔNG TIN BÀI BÁO

Ngày nhận bài: **10/8/2022**
Ngày hoàn thiện: **16/9/2022**
Ngày đăng: **16/9/2022**

TỪ KHÓA

Lưu trữ văn bản
Cây băm Merkle
Ed25519
AES
SHA-256

TÓM TẮT

Xác thực và bảo mật văn bản là các nhiệm vụ quan trọng và cấp thiết trong hệ thống lưu trữ văn bản của doanh nghiệp, tổ chức hiện nay. Bài viết này đề xuất giải pháp xác thực, bảo mật văn bản lưu trữ sử dụng kết hợp các kỹ thuật mật mã là cây băm Merkle, lược đồ ký số trên đường cong elliptic Ed25519 và mã khối AES. Mục tiêu giải pháp đề xuất là có độ an toàn ngang bằng và hiệu năng tốt hơn so với giải pháp ký số, mã hóa từng văn bản truyền thống. Các kết quả này được chứng minh dựa trên việc so sánh cơ sở lý thuyết và thực nghiệm bằng ngôn ngữ lập trình python so với giải pháp ký số và mã hóa truyền thống. Qua đó kết luận rằng đây là một giải pháp có tiềm năng tốt để có thể áp dụng bảo mật văn bản lưu trữ trên thực tế.

DOI: https://doi.org/10.34238/tnu-jst.6352

[*] Corresponding author. *Email: nghinv@actvn.edu.vn*

1. Giới thiệu

Chuyển đổi số đang được quan tâm và trở thành xu hướng bắt buộc trong tương lai gần của các doanh nghiệp, chính phủ tại Việt Nam hiện nay. Để tiến hành chuyển đổi số thì công việc đầu tiên cần thực hiện là việc số hoá các tài liệu truyền thống sang các tài liệu, văn bản điện tử.

Các văn bản điện tử được tạo ra và lưu trữ trong hệ thống công nghệ thông tin của các cơ quan, doanh nghiệp. Những văn bản điện tử có thể chứa các thông tin nhạy cảm liên quan đến cá nhân, cơ quan, doanh nghiệp. Và yêu cầu đặt ra cần có giải pháp bảo mật, xác thực các văn bản điện tử lưu trữ này.

Giải pháp bảo mật, xác thực các văn bản điện tử đang được sử dụng phổ biến hiện nay là áp dụng các thuật toán mật mã, bao gồm thuật toán mã hoá và chữ ký số [1], [2]. Quá trình mã hoá văn bản sẽ đảm bảo tính bí mật của văn bản và quá trình ký số đảm bảo tính toàn vẹn và chống chối bỏ đối với văn bản đã ký. Hai quá trình có thể được thực hiện riêng lẻ hoặc kết hợp tuần tự với nhau. Tuy nhiên với số lượng văn bản của cơ quan, tổ chức cần lưu trữ có dung lượng lớn theo từng năm thì việc mã hóa và ký số từng văn bản sẽ cần rất nhiều dung lượng lưu trữ bản mã và giá trị ký số. Vấn đề thứ hai là việc kiểm tra văn bản lưu trữ có bị thay đổi hoặc chỉnh sửa hay không phải tiến hành theo từng văn bản mất nhiều công sức. Như vậy, đối với các văn bản lưu trữ với tính chất có rất ít sự chỉnh sửa thì yêu cầu đặt ra cần có giải pháp bảo mật văn bản theo từng nhóm (lô) văn bản tương ứng với người dùng sở hữu và có thể dễ dàng thực hiện và kiểm tra sự thay đổi chỉnh sửa những văn bản lưu trữ này.

Xuất phát từ ý tưởng lưu trữ tập tin hệ thống với cây băm Merkle trong [3], thì trong bài viết này, chúng tôi đề xuất giải pháp bảo mật, xác thực văn bản sử dụng nguyên thủy mật mã là cây băm Merkle [4] – [9], lược đồ ký số Ed25519 [10], [11] và mã khối AES [12]. Mục tiêu của giải pháp này sẽ mang lại hiệu năng tốt so việc mã hoá, ký số văn bản một cách tuần tự đơn lẻ và mang lại độ bảo mật cao do sử dụng các nguyên thuỷ mật mã hiện đại. Các mục tiêu này được chứng minh dựa trên việc so sánh cơ sở lý thuyết và thực nghiệm bằng ngôn ngữ lập trình python so với giải pháp ký số và mã hóa truyền thống.

Bài viết này được bố cục theo 5 mục chính: sau phần 1 giới thiệu, phần 2 đưa ra cơ sở lý thuyết liên quan. Phần 3 trình bày chi tiết về giải pháp bảo mật, xác thực văn bản sử dụng cây băm Merkle, lược đồ ký số Ed25519 và mã khối AES mà chúng tôi đề xuất. Phần 4, bài viết trình bày về vấn đề an toàn, hiệu năng của giải pháp đề xuất dựa trên cơ sở lý thuyết mô hình giải pháp và kết quả triển khai thực nghiệm của giải pháp. Cuối cùng là phần kết luận và tài liệu tham khảo được tham chiếu trong bài viết.

2. Cơ sở lý thuyết liên quan

2.1. Cây băm Merkle

Cây băm Merkle được giới thiệu bởi Merkle vào năm 1979 [4]. Về cơ bản, cây băm Merkle là một cây nhị phân có thứ tự được xây dựng từ một dãy các đối tượng dữ liệu (d_1, d_2, ..., d_n) sử dụng một hàm băm mật mã H. Các nút lá của cây là các giá trị băm $H(d_i)$ đối với $1 \leq i \leq n$. Các nút khác của cây băm Merkle đều sẽ bao gồm hai nút con, một nút con bên trái (left) và một nút con bên phải (right). Các nút dưới cùng của cây là các nút lá. Các nút tiếp theo sẽ chứa giá trị băm H(left||right), trong đó left là giá trị băm chứa ở nút bên trái, right là giá trị băm lưu ở nút bên phải và được ghép nối với nhau (||). Nút cuối cùng của cây được gọi là nút gốc hay root [2] – [9].

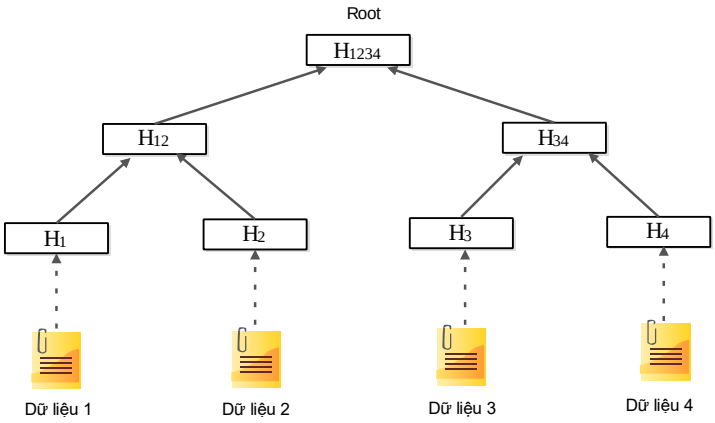
Quá trình xây dựng cây băm Merkle được trình bày thông qua ví dụ trong hình 1.

Đầu vào chúng ta có dữ liệu 1, 2, 3, 4: đây có thể là các tập tin dữ liệu hoặc các khối dữ liệu. Cách thức xây dựng cây băm Merkle được thực hiện theo các bước sau:

- Tiến hành băm các dữ liệu 1, 2, 3, 4 với một hàm băm mật mã H để thu được các giá trị băm là H_1, H_2, H_3, H_4.

- Lưu các giá trị băm vào các nút lá cây Merkle. Với hai nút lá thì thành lập được một nút tiếp theo của cây băm Merkle. Các nút này chứa nội dung là giá trị băm H(left || right) trong đó thì left là giá trị nút bên trái và right là giá trị nút bên phải của nút vừa thành lập.

- Tiến hành xây dựng các nút còn lại của cây băm Merkle với nội dung là giá trị băm H(left || right) để thu được nút cuối cùng gọi là nút gốc hay root.



Hình 1. *Mô tả cấu trúc cây băm Merkle*

Thấy rằng với số lượng dữ liệu là lũy thừa của 2 thì cây Merkle được xây dựng tuần tự như các bước đã nêu trên. Trường hợp số lượng dữ liệu đầu vào không phải là lũy thừa của 2 thì các nút lá còn thiếu của cây Merkle sẽ nhận giá trị băm của dữ liệu cuối cùng.

2.2. Lược đồ ký số Ed25519

Ed25519 là lược đồ ký số dựa trên đường cong elliptic dạng Edwards sử dụng hàm băm SHA-512 và Curve25519 được giới thiệu bởi Daniel J. Bernstein và các cộng sự năm 2012 trong [10], [11].

Các tham số lược đồ ký số Ed25519 bao gồm:

- Phương trình đường cong: $-x^2 + y^2 = 1 - \frac{121665}{121666}x^2y^2$
- Số nguyên tố $p = 2^{255} - 19$
- Bậc của đường cong:

$$l = 2^{252} + 27742317777372353535851937790883648493$$

- Điểm cơ sở B có tọa độ:

X_B = 15112221349535400772501151409588531511454012693041857206046113283949847762202

Y_B = 46316835694926478169428394003475163141307993866256225615783033603165251855960

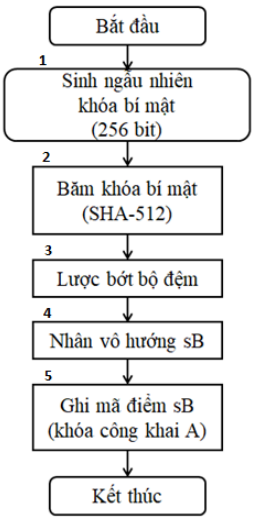
- Hàm băm SHA-512
- Hàm PH(x) = x là hàm tính giá trị băm đầu vào x.

Quá trình sinh khóa lược đồ Ed25519 được mô tả trong lược đồ hình 2 bao gồm các bước:

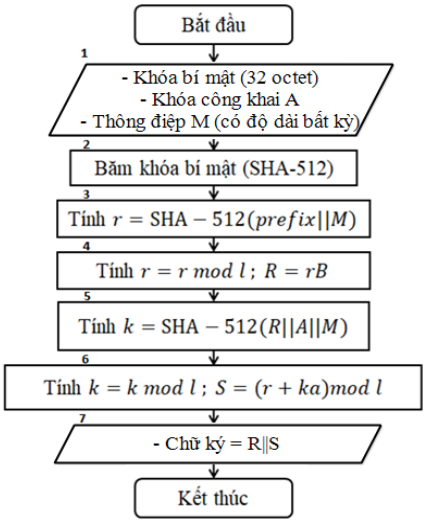
- Bước 1: Khóa bí mật người dùng (Private key) gồm 32 octet (256 bit) được sinh ngẫu nhiên.
- Bước 2: Băm khóa bí mật 32-byte bằng SHA-512, lưu dưới dạng tóm lược trong một bộ đệm dài 64 octet.
- Bước 3: Tiến hành lược bộ đệm chỉ sử dụng 32 byte trọng số thấp và kết quả là được thể hiện dạng số nguyên s.
- Bước 4, 5: Thực hiện nhân vô hướng số nguyên s với điểm cơ sở B. Kết quả là một điểm trên đường cong Curve25519 và được thực hiện quá trình ghi mã để chuyển điểm về dạng số nguyên dài 32 octet và đây chính là khóa công khai A của người dùng.

Quá trình ký Ed25519 được mô tả trong hình 3 bao gồm các bước chính sau:

- Bước 1: Băm 32 octet khóa riêng sử dụng hàm băm SHA-512. Xây dựng thành phần *a* vô hướng bí mật từ nửa đầu tiên của giá trị băm, và tương ứng với khóa công khai A, như mô tả ở phần trước.



Hình 2. *Quá trình sinh khóa Ed25519*



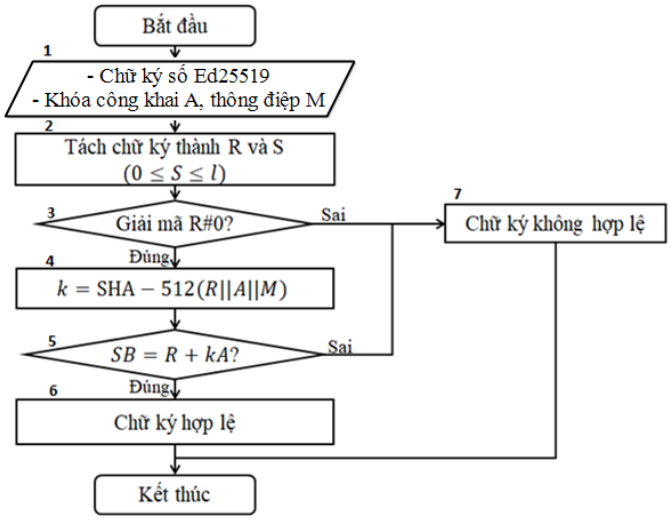
Hình 3. *Quá trình ký số Ed25519*

- Bước 2: Tính toán SHA-512(prefix|| M) với M là thông điệp được ký. Chuỗi tóm lược 64 octet được thể hiện như một số nguyên r.

- Bước 3: Tính điểm $R = rB$, để bước này hiệu quả, thay thế r bằng $r \bmod l$ trước khi tính.

- Bước 4: Tính SHA-512($R||A||M$), đầu ra là bản tóm lược 64 octet được biểu diễn dạng số nguyên k.

- Bước 5: Chữ ký được tạo là sự ghép nối của chuỗi R (32 octet) và bản ghi mã của S (32octet, 3 bit có ý nghĩa cao nhất của octet cuối cùng bằng 0).



Hình 4. *Quá trình kiểm tra chữ ký số Ed25519*

Quá trình kiểm tra chữ ký Ed25519 được mô tả trong hình 4 bao gồm các bước chính sau:

- Bước 1, 2, 3: Nhập thông tin chữ ký số, khóa công khai A và bản rõ M. Đầu tiên phải tách chữ ký thành 2 phần, mỗi phần 32 octet. Giải mã phần octet đầu tiên thành một điểm R, và phần còn lại thành số nguyên S $(0 \le S \le l)$. Tiến hành giải ghi mã khóa công khai A, tức là chuyển khóa công khai thành điểm A'. Nếu bất kỳ quá trình nào ở trên thất bại thì chữ ký là không hợp lệ.

- Bước 4: Tính SHA-512($R||A||$PH(M)) và bản tóm lược 64 octet như số nguyên k.

- Bước 5: Kiểm tra phương trình $SB = R + kA'$. Nếu phương trình trên là đúng thì chữ ký được xác nhận là chính xác ngược lại chữ ký không hợp lệ.

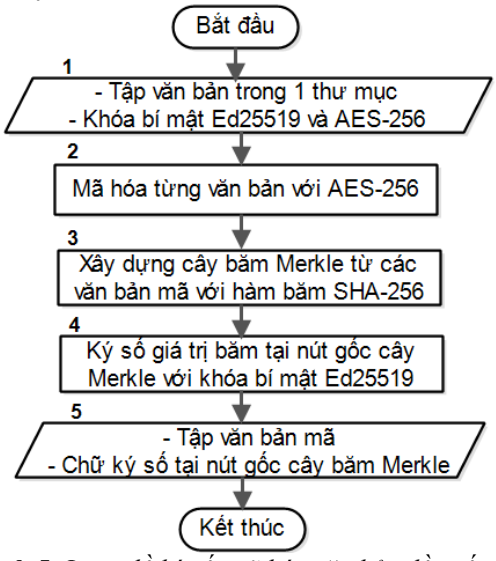
- Bước 6: Thông báo chữ ký hợp lệ.
- Bước 7: Thông báo chữ ký không hợp lệ.

2.3. Chuẩn mã hóa tiên tiến AES

AES là tiêu chuẩn mã hóa tiên tiến của Hoa Kỳ công bố năm 2001 với mục đích sử dụng để mã hóa dữ liệu số trong các lĩnh vực dân sự khác nhau [12]. Đây là thuật toán mã khối có ba phiên bản khác nhau tương ứng 3 kích thước khóa là 128 bit, 192 bit, 256 bit.

3. Giải pháp xác thực, bảo mật văn bản lưu trữ sử dụng cây băm Merke, lược đồ ký số ED25519 và mã khối AES

Trong bài viết này các tác giả đưa ra đề xuất giải pháp, xác thực bảo mật văn bản lưu trữ sử dụng cây băm Merkle, lược đồ ký số Ed25519 và mã khối AES theo lược đồ hình 5.



Hình 5. *Lược đồ ký số, mã hóa văn bản đề xuất*

Giải pháp đề xuất được thực thi theo 5 bước trong lược đồ hình 5. Để làm rõ 5 bước này thì chúng ta xem xét hoạt động giải pháp với tập 4 văn bản rõ trong thư mục như hình 6:

- Bước 1, 2: Các văn bản rõ trong thư mục sẽ được tiến hành mã hóa lần lượt với hệ mật AES có độ dài khóa 256 bit.
- Bước 3: Xây dựng cây băm Merkle với các nút lá là các giá trị băm của văn bản mã, sử dụng hàm băm SHA-256 [13].

+ Trường hợp này số văn bản mã m = 4 và bằng 2^n với n = 4 thì cây băm được xây dựng bình thường như hình 6.

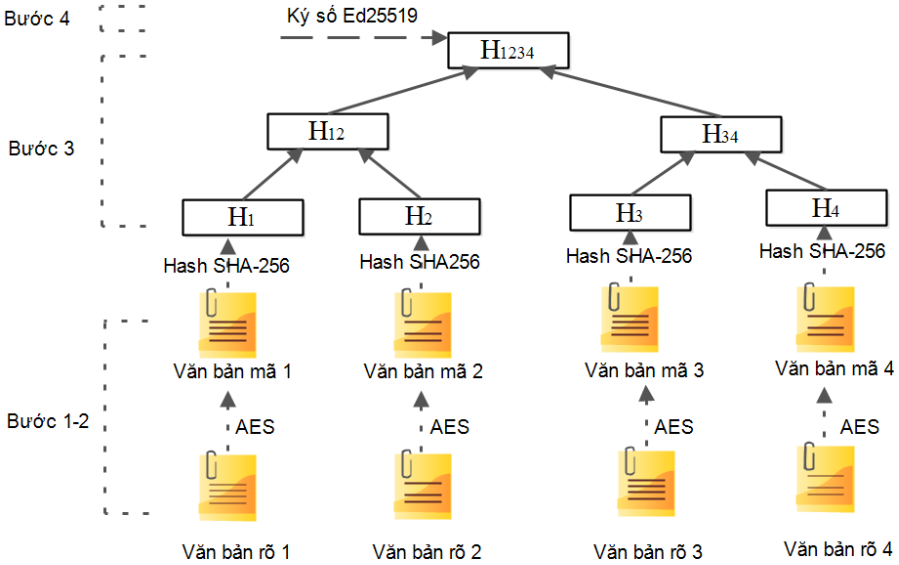
+ Trường hợp $2^{n+1} > m > 2^n$ thì cây băm Merkle được xây dựng với 2^{n+1} nút lá và giá trị nút lá từ m+1 tới 2^{n+1} giống với nút lá m.

- Bước 4: Sử dụng khóa bí mật Ed25519 của người dùng để ký lên giá trị băm nút gốc cây băm Merkle. Chữ ký thu được lưu trữ cùng thư mục hoặc nơi an toàn thuận tiện cho quá trình kiểm tra.

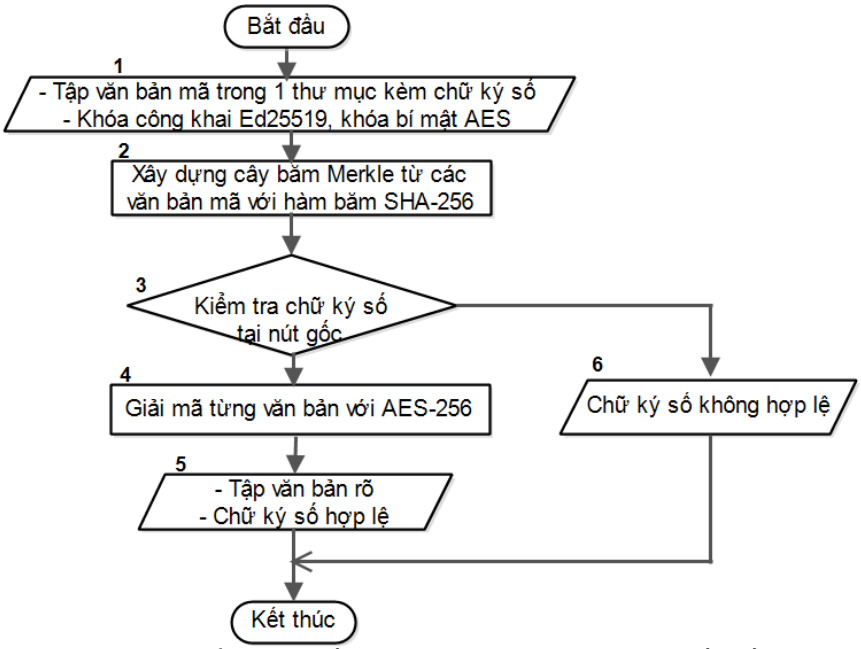
Quá trình kiểm tra các văn bản được mô tả như trong lược đồ thuật toán hình 7 bao gồm các bước:

- Bước 1: Nhập các tham số đầu vào: các văn bản mã kèm chữ ký số, khóa công khai Ed25519 và khóa bí mật AES-256 của người dùng.
- Bước 2: Xây dựng cây băm Merkle với các nút là các giá trị băm của văn bản mã, sử dụng hàm băm SHA-256. Thực hiện tương tự quá trình ký số.
- Bước 3: Sử dụng khóa công khai để kiểm tra chữ ký số nút gốc. Nếu đúng chuyển sang bước 4, sai thì chuyển sang bước 6.

- Bước 4: Giải mã từng văn bản với khóa bí mật AES-256 nhập vào.
- Bước 5: Kết quả nhận được tập văn bản rõ và thông báo chữ ký số hợp lệ.
- Bước 6: Thông báo chữ ký số không hợp lệ.



Hình 6. *Chi tiết quá trình ký số, mã hóa văn bản đề xuất*



Hình 7. *Kiểm tra ký số, giải mã văn bản của giải pháp đề xuất*

Lý giải về các lựa chọn thuật toán mật mã trong giải pháp đề xuất

Việc lựa chọn cây băm Merkle giúp cho giải pháp có thể ký số một hoặc một tập văn bản lưu trữ. Điều này dẫn tới hiệu năng giải pháp cải thiện đáng kể so với ký số từng văn bản (xem mục 4.2).

Ngoài ra, giải pháp lựa chọn lược đồ ký số Ed25519 vì đây là lược đồ ký số mới có ưu điểm: Độ bảo mật tương đương với lược đồ ký số ECDSA và tốt hơn RSA [14]. Hiệu năng hơn nhiều lần so với ECDSA [10]. Mã khối AES được sử dụng rất phổ biến hiện nay trong các giải pháp phần mềm, giao thức bảo mật và chưa có tấn công thực tế nào hiệu quả để phá mã.

4. Vấn đề an toàn, hiệu năng của giải pháp đề xuất

4.1. Vấn đề an toàn

Việc phá vỡ khả năng bảo mật dữ liệu của giải pháp đề xuất thì kẻ tấn công có thể thực hiện các tấn công như sau: Tấn công vào lược đồ ký số Ed25519 nhằm thay đổi hoặc giả mạo chữ ký tại nút gốc của cây băm Merkle; Tấn công vào cây băm Merkle nhằm thay đổi hoặc giả mạo các nút lá hoặc nút root của cây băm. Từ đó thay đổi dữ liệu lưu trữ cần bảo mật trong giải pháp. Trong giải pháp cây băm Merkle được xây dựng từ hàm băm SHA-256, như vậy việc tấn công cây băm Merkle chính là tấn công hàm băm SHA-256; Tấn công vào mã khối AES nhằm giải mã văn bản lưu trữ.

Để xem xét khả năng chống tấn công của giải pháp đề xuất thì có thể xem xét hai vấn đề: Thứ nhất là độ bảo mật của từng nguyên thủy mật mã trong giải pháp (lược đồ ký số Ed25519, hàm băm SHA-256 và mã khối AES), đây là thông số thể hiện khả năng kháng tấn công vét cạn của các nguyên thủy mật mã này; Thứ hai là khả năng kháng các tấn công phổ biến khác của các nguyên thủy mật mã sử dụng trong giải pháp đề xuất.

Để đánh giá độ bảo mật các thuật toán mật mã là hàm băm SHA-256, lược đồ ký số Ed25519, mã khối AES, chúng tôi tham khảo khuyến nghị theo hình 8 dưới đây của NIST trong việc lựa chọn kích thước tham số cho các thuật toán mật mã cơ bản tương ứng thời điểm sử dụng [14].

Từ số liệu trong khuyến nghị NIST trong hình 8 thì thấy rằng độ bảo mật của SHA-256 và lược đồ ký số Ed25519 là 128 bit, còn AES-256 là 256 bit. Cũng theo khuyến nghị thì mức bảo mật này có thể sử dụng tốt cho tới năm 2030.

Date	Security Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash (A)	Hash (B)
Legacy (1)	80	2TDEA	1024	160	1024	160	SHA-1 (2)	
2019 - 2030	112	(3TDEA) (3) AES-128	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2019 - 2030 & beyond	128	AES-128	3072	256	3072	256	SHA-256 SHA-512/256 SHA3-256	SHA-1 KMAC128
2019 - 2030 & beyond	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224 SHA3-224
2019 - 2030 & beyond	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-256 SHA3-384 SHA3-512 KMAC256

Hình 8. *Khuyến nghị NIST về độ bảo mật và thời gian sử dụng các thuật toán mật mã cơ bản năm 2020*

Tiếp theo, chúng ta xem xét khả năng kháng các tấn công phổ biến của lược đồ ký số Ed25519, hàm băm SHA2-256 và mã khối AES.

- Đối với lược đồ ký số Ed25519 là tấn công vào quá trình sinh giả ngẫu nhiên giá trị r để tìm khóa riêng trong quá trình ký số và tấn công phá vỡ công thức chữ ký số. Trong [10], các tác giả đã chứng minh Ed25519 đều đảm bảo an toàn bởi các tấn công này: với tấn công vào quá trình sinh giá trị *r* thì Ed25519 sử dụng quá trình sinh số *r* tất định dựa trên hàm băm với đầu vào là mã băm khóa bí mật và thông điệp; công thức chữ ký số Ed25519 đã đưa ra đủ an toàn trước cuộc tấn công phá vỡ theo năng lực tính toán của các máy tính hiện tại.

- Đối với hàm băm SHA2-256 thì tấn công phổ biến nhất là tấn công tìm va chạm. Tấn công tìm va chạm truyền thông tối ưu đối với hàm băm này được công bố gần đây là trong [15] thực hiện với 31/64 bước có độ phức tạp $2^{65.5}$, và trong [16] các tác giả có đưa ra tấn công tìm va chạm lượng tử lên tới 38-39/64 bước với độ phức tạp tính toán $2^{122/\sqrt{s}}$ với s < 2^{12}. Với các tấn công chỉ hiệu quả tới 39/64 bước thì việc kháng va chạm của SHA-256 an toàn gần như tuyệt đối trước các tấn công này.

- Đối với mã khối AES hai tấn công ghi nhận phổ biến là lượng sai và Boomerang. Phiên bản tấn công lượng sai hiệu quả nhất đối với 7 vòng AES-256 được công bố là của các tác giả trong [17] có độ phức tạp tính toán $2^{250.5}$ với số lượng cặp rõ/mã cần thiết $2^{92.5}$ và bộ nhớ 2^{153}. Tấn công Boomerang hiệu quả đối với 6 vòng AES-128 của các tác giả trong [18] có độ phức tạp tính toán 2^{71} với số lượng cặp rõ/mã 2^{71} và bộ nhớ 2^{33}. Rõ ràng, các tấn công chỉ hiệu quả với 7 vòng thì AES-256 đầy đủ 14 vòng thực thi vẫn có được độ an toàn rất tốt hiện nay.

4.2. Về hiệu năng theo lý thuyết

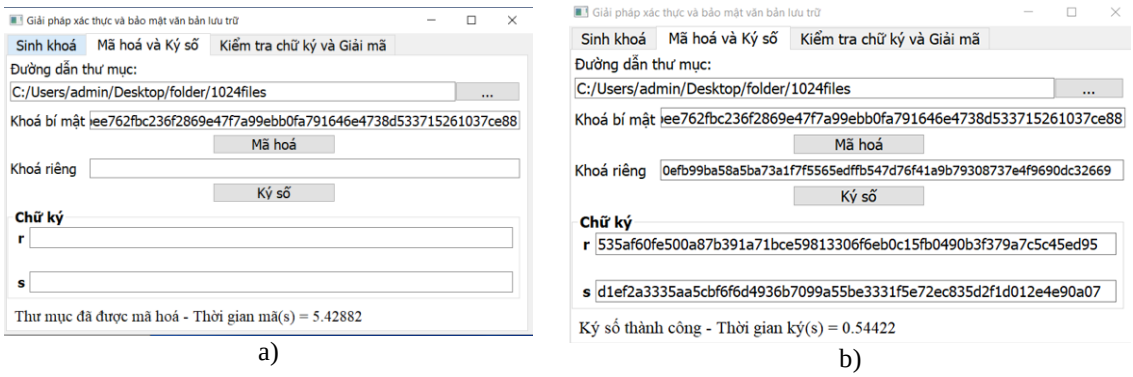
Để đánh giá hiệu năng giải pháp đề xuất, chúng tôi thực hiện so sánh với giải pháp truyền thống được sử dụng phổ biến là ký số, mã hóa từng văn bản trong thư mục.

Trong giải pháp truyền thống với m văn bản tương ứng $2^{n+1} \geq m \geq 2^n$ hay có thể cho rằng m ~ 2^{n+1}. Thì thời gian thực thi chính là thời gian mã hóa và ký số k văn bản. Trong giải pháp đề xuất thì với k văn bản, thời gian thực thi là thời gian mã hóa m văn bản, thời gian xây dựng cây băm và thời giản ký số một lần nút gốc.

So sánh hai giải pháp thấy rằng: Thời gian mã hóa văn bản là tương đương ở cả hai giải pháp. Thời gian xây dựng cây băm trong giải pháp đề xuất sẽ thực hiện $\sum_{i=1}^{n+1} 2^i$ với i là số nguyên dương và n+1 ≥ i ≥ 1. Thực chất với quá trình ký số k văn bản trong giải pháp truyền thống thì cũng thực hiện băm k lần. Điều này dẫn tới giải pháp đề xuất sẽ thực thi hơn giải pháp truyền thống ~ $\sum_{i=1}^{n} 2^i$ lần.

Như vậy khi so sánh hai giải pháp, chúng tôi so sánh thời gian băm $\sum_{i=1}^{n} 2^i$ lần trong giải pháp đề xuất với thời gian ký m-1 văn bản trong giải pháp truyền thống. Hai giá trị $\sum_{i=1}^{n} 2^i$ và m-1 không có sự chênh lệch nhiều và có thể coi là xấp xỉ nhau. Trong [19], các tác giả tiến hành đo hiệu năng của hàm băm SHA-256 và lược đồ ký số Ed25519 với cùng kích cỡ dữ liệu từ 512B, 1KB, 2KB trên cùng một nền tảng phần cứng chip 64 bit. Kết quả cho thấy rằng hiệu năng của hàm băm SHA-256 cho tốc độ nhanh hơn khoảng 15 lần so với lược đồ ký số Ed25519. Với kích cỡ dữ liệu băm và ký số trong giải pháp đề xuất và truyền thống đều là 256 bit và kết hợp kết quả trong [19] thì có cơ sở khẳng định rằng giải pháp đề xuất nhanh hơn nhiều lần so với giải pháp truyền thống.

4.3. Về hiệu năng trong thực nghiệm

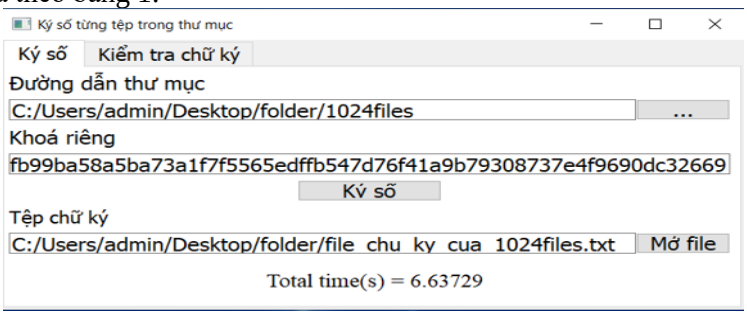


Hình 9. *Thực nghiệm giải pháp ký số, bảo mật văn bản đề xuất*

Trên nền tảng phần cứng Laptop Vostro 15 3510 với RAM 8GB, HĐH Windows 10, 64 bit, Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz. Chúng tôi triển khai thực nghiệm đánh giá hiệu năng giải pháp đề xuất bằng ngôn ngữ lập trình Python với giao diện đồ họa quá trình mã hóa (hình 9a), quá trình ký số (hình 9b) các văn bản tập tin trong một thư mục.

Ngoài ra, trên cùng nền tảng phần cứng thì chúng tôi cũng cài đặt đánh giá hiệu năng giải pháp mã hóa, ký số từng văn bản truyền thống để có kết quả so sánh với giải pháp chúng tôi đề xuất (hình 10).

Quá trình thực nghiệm được thực thi với các tập tin văn bản có kích thước 512 KB trong 4 thư mục chứa có số lượng văn bản lần lượt là 128 files, 256 files, 512 files và 1024 files. Thực hiện với cùng khóa được sinh ra ngẫu nhiên theo ứng dụng cài đặt giải pháp đề xuất. Kết quả đo được thực tế được đưa ra theo bảng 1.



Hình 10. *Thực nghiệm đo hiệu năng giải pháp truyền thống*

Bảng 1. *Bảng so sánh hiệu năng thực nghiệm của giải pháp đề xuất và giải pháp truyền thống*

Thư mục	Thời gian mã hóa	Thời gian giải mã	Thời gian ký số theo giải pháp đề xuất	Thời gian kiểm tra ký số theo giải pháp đề xuất	Thời gian ký số từng văn bản	Thời gian kiểm tra chữ ký từng văn bản	So sánh thời gian ký giải pháp đề xuất với ký từng văn bản (nhanh hơn bao nhiêu lần)	So sánh thời gian kiểm tra chữ ký giải pháp đề xuất với ký từng văn bản (nhanh hơn bao nhiêu lần)
128 files	0,53399	0,6399	0,07667	0,07575	0,81562	0,85245	10,64	11,25
256 files	0,94234	1,2133	0,13655	0,13752	1,70572	1,80322	12,49	13,11
512 files	2,32886	2,67813	0,2811	0,27109	3,34004	3,352763	11,88	12,37
1024 files	5,71338	5,82134	0,55727	0,55414	6,73405	6,79268	12,08	12,26

Nhận xét:

- Thời gian mã hóa của hai giải pháp là như nhau với thuật toán AES-256 mã từng tập tin trong thư mục.
- Thời gian ký số và kiểm tra ký số ở hai giải pháp có sự chênh lệch lớn. Kết quả đo được kết quả giải pháp đề xuất nhanh hơn giải pháp truyền thống xấp xỉ 12 lần.

5. Kết luận

Bài viết đã đưa ra đề xuất một giải pháp xác thực, bảo mật văn bản sử dụng cây băm Merkle, lược đồ ký số Ed25519 và mã khối AES. Với những thông số về độ an toàn và hiệu năng đã được đưa ra trong bài viết thì giải pháp này có tiềm năng và rất khả thi để áp dụng trong thực tế các hệ thống lưu trữ văn bản lớn.

TÀI LIỆU THAM KHẢO/ REFERENCES

[1] W. Stalling, *Cryptography and Network Security: Principles and Practice*, 6th edition, Pearson Publishing, 2014, pp. 08-24.

[2] S. Nakov, *Practical cryptography for developers*, SoftUni, 2018, pp. 144 – 172.
[3] O. Rodeh, J. Bacik, and C. Mason, "BTRFS: The Linux B-tree filesystem," *ACM Transactions on Storage (TOS)*, vol. 9, no. 3, pp. 1-32, 2013.
[4] M. R. Secrecy, "Authentication, and public key systems. Electrical Engineering," PhD. Thesis, Stanford University, 1979.
[5] B. Carminati, "Merkle Trees," in *Encyclopedia of Database Systems*, Springer Reference, 2009, pp. 1714–1715.
[6] D. Koo, Y. Shin, J. Yun, and J. Hur, "Improving Security and Reliability in Merkle tree-based Online Data Authentication with Leakage Resilience," *Applied Sciences*, vol. 8, no. 12, 2018, Art. no. 2532.
[7] B. Sharma, C. N. Sekharan, and F. Zuo, "Merkle-tree based approach for ensuring integrity of electronic medical records," *Proceedings of 2018 9th IEEE Annual Ubiquitous Computing Electronics & Mobile Communication Conference (UEMCON)*, 2018, pp. 983-987.
[8] Q. Li, "Research on E-Commerce User Information Encryption Technology Based on Merkle Hash Tree," *Proceedings of 2019 International Conference on Robots & Intelligent System (ICRIS)*, 2019, pp. 365-369.
[9] S. Dhumwad, M. Sukhadeve, C. Naik, K. N. Manjunath, and S. Prabhu, "A peer to peer money transfer using SHA256 and Merkle tree," *Proceedings of 2017 23RD Annual International Conference in Advanced Computing and Communications (ADCOM)*, 2017, pp. 40-43.
[10] J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B. Yang, "High-speed high- security signatures," *Journal of Cryptographic Engineering*, vol. 2, pp. 77-89, 2012.
[11] S. Josefsson, "Edwards curve Digital Signature Algorithm (EdDSA)," IETF, 2017. [Online]. Available: https://tools.ietf.org/html/rfc-8032. [Accessed Jun. 19, 2022].
[12] United States National Institute of Standards and Technology, "Announcing the ADVANCED ENCRYPTION STANDARD (AES)," Federal Information Processing Standards Publication 197. November 26, 2001.
[13] M. Lamberge and F. Mendel, *Higher-Order Differential Attack on Reduced SHA-256*, IACR Cryptology ePrint Archive, 2011.
[14] D. Giry, "Recommendation for Key Management," Special Publication 800-57 Part 1 Rev. 5, NIST, 05/2020. [Online]. Available: https://www.keylength.com/en/4/. [Accessed Jul. 04, 2022].
[15] F. Mendel, T. Nad, and M. Schlaffer, "Improving local collisions: New attacks on reduced SHA-256," *LNCS*, Springer, vol. 7881, pp. 262–278, 2013.
[16] A. Hosoyamada and Y. Sasaki, *Quantum Collision Attacks on Reduced SHA-256 and SHA-512*, Cryptology ePrint Archive, 2021.
[17] M. R. Z'aba and M. A. Maarof, "A survey on the cryptanalysis of the advanced encryption standard," in *Proceedings of the Postgraduate Annual Research Seminar*, 2006, pp. 97–102.
[18] K. Li, L. Qu, B. Sun, and C. Li, "New results about the boomerang uniformity of permutation polynomials," *IEEE Transactions on Information Theory*, vol. 65, no. 11, pp. 7542–7553, 2019.
[19] P. Wang, C. Hua, and M. Zochowski, "Benchmarking Hash and Signature Algorithms, medium.com," 2019. [Online]. Available: https://medium.com/logos-network/benchmarking-hash-and-signature-algorithms-6079735ce05. [Accessed Jul. 04, 2022].