TỔNG QUAN VỀ CHỮ KÝ SỐ KHÁNG LƯỢNG TỬ DỰA TRÊN HÀM BĂM

Đỗ Thị Bắc*, Bounsaveng Khit

Trường Đại học Công nghệ thông tin và Truyền thông - Đại học Thái Nguyên, Việt Nam

ARTICLE INFORMATION

Journal: *Vinh University Journal of Science*
ISSN: *1859-2228*

Volume: 52
Issue: 3A
****Correspondence:*** *dtbac@ictu.edu.vn*

Received: *31 March 2023*
Accepted: *12 May 2023*
Published: *20 September 2023*

Citation:
Đỗ Thị Bắc, Bounsaveng Khit (2023). Tổng quan về chữ ký số kháng lượng tử dựa trên hàm băm. ***Vinh Uni. J. Sci.*** *Vol. 52 (3A), pp. 40-54 doi:10.56824/vujs.2023a046*

TÓM TẮT

Trước thách thức phát triển máy tính lượng tử, các thuật toán chữ ký số kháng lượng tử đã được ra đời và nhận được quan tâm của nhiều nhà khoa học. Bài báo này tập trung vào đánh giá các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm và khả năng ứng dụng của chúng trong việc bảo vệ thông tin truyền tải. Phân tích, tổng hợp và so sánh những điểm mạnh và hạn chế của các thuật toán là phương pháp nghiên cứu chính của bài báo. Đồng thời cũng chỉ ra thách thức khi ứng dụng chúng trong các lĩnh vực khác nhau như ngân hàng, thương mại điện tử và chính phủ. Bài báo hữu ích cho những người mới bắt đầu nghiên cứu về chữ ký số kháng lượng tử, giúp họ có được cái nhìn tổng quan về lĩnh vực. Nó cũng giúp các nhà quản lý, chính phủ và các tổ chức khác trong việc đưa ra các quyết định và chính sách liên quan đến nền tảng bảo mật bằng chữ ký số.

Từ khóa: Kháng lượng tử; chữ ký số; hàm băm; bảo mật thông tin; mật mã.

1. Giới thiệu

Trong những năm gần đây, với sự phát triển của máy tính lượng tử, các phương pháp truyền thống trong việc bảo mật thông tin như thuật toán chữ ký số dựa trên RSA hay ECC đang dần trở nên không an toàn và cần được thay thế bằng các giải pháp mới khác [1-5]. Một trong những giải pháp đó là sử dụng các thuật toán chữ ký số kháng lượng tử (hậu lượng tử) dựa trên hàm băm [6-10].

Các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm đã được đề xuất từ khá lâu và được chứng minh đạt được độ bảo mật tốt hơn so với các thuật toán chữ ký số truyền thống [1, 10-16]. Các thuật toán chữ ký số này được phát triển dựa trên kỹ thuật chữ ký số một lần và sử dụng hàm băm để tạo ra khóa [7-9, 17-19]. Đến năm 2016, Viện Tiêu chuẩn và Công nghệ Quốc gia Mỹ (National Institute of Standards and Technology - NIST) đã kêu gọi các cá nhân và các tổ chức đề xuất thuật toán được cho là có khả năng kháng lượng tử với thời hạn một năm đến tháng 11 năm 2017 [21]. Tháng 1 năm 2018, NIST đã công bố kết quả của vòng đầu tiên. Đã có 82 thuật toán được đề xuất (gồm cả mã hóa, trao đổi khóa và chữ ký số). Trong đó 23 trong số chúng là các lược đồ chữ ký và chỉ 4 là các lược đồ chữ ký dựa trên hàm băm

(nhưng chỉ có 2 trong số đó được xuất bản). Vào tháng 4 năm 2018, NIST đã tổ chức một hội thảo, nơi người đề xuất trình bày các giải pháp của họ. Tiếp theo là giai đoạn phân tích từ 3 đến 5 năm với báo cáo về các đánh giá. Dự kiến, từ năm 2023 đến năm 2025, một bản thảo chuẩn hóa cho mật mã hậu lượng tử [21] sẽ sẵn sàng. NIST nhấn mạnh rằng đây không phải là một cuộc thi và một số ứng viên có thể được chấp thuận cho một ứng dụng/mục đích duy nhất.

Trong bài báo này, chúng tôi sẽ giới thiệu tổng quan về chữ ký số kháng lượng tử dựa trên hàm băm thông qua các thuật toán như Lamport, Merkle, HORS, Picnic và SPHINCS+ [21]. Đây là các thuật toán được đánh giá cao về khả năng ứng dụng và trong đó có các thuật toán lọt vào vòng 2 hoặc vòng 3 của cuộc thi tìm chuẩn mã hóa do NIST tổ chức đã nêu trên. Chúng tôi sẽ trình bày về cấu trúc, cách thức hoạt động của từng thuật toán. Đồng thời, sẽ so sánh các thông số của các thuật toán như độ dài khóa, độ phức tạp tính toán và độ an toàn. Cuối cùng, chúng tôi sẽ đưa ra đánh giá về khả năng ứng dụng của các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm và những hạn chế của chúng trong thực tế.

Bài báo được cấu trúc thành 4 phần, sau phần giới thiệu là phần trình bày về phương pháp nghiên cứu với 3 nhóm phương pháp chính gồm phương pháp phân tích tài liệu, phân tích nội dung, phân tích hệ thống. Nội dung chính của bài báo nằm trong phần 3 - Kết quả và thảo luận. Phần này bao gồm giới thiệu về các thuật toán, mô tả hoạt động, phân tích, đánh giá và so sánh các thuật toán.

2. Phương pháp nghiên cứu

Mật mã hậu lượng tử là một lĩnh vực khoa học còn rất mới và là không gian mở đối với các nhà khoa học trên cả thế giới và ở Việt Nam. Hiện tại máy tính lượng tử chưa được sử dụng phổ biến mới chỉ được nghiên cứu và sử dụng trong các phòng thí nghiệm của các tập đoàn lớn trên thế giới nên việc nghiên cứu về mật mã kháng lượng tử hay nhỏ hơn là chữ ký số kháng lượng tử là rất thời sự và cấp thiết. Vì vậy trong bài báo tổng quan này chúng tôi sử dụng các phương pháp nghiên cứu chủ yếu sau:

- Phương pháp phân tích tài liệu: Phương pháp này bao gồm đánh giá các tài liệu trước đây đã được xuất bản về chủ đề chữ ký số và chữ kỹ số kháng lượng tử dựa trên hàm băm. Việc này giúp đánh giá tình hình nghiên cứu hiện tại, những hạn chế, điểm mạnh và những đề xuất ứng dụng của chúng.
- Phương pháp phân tích nội dung: Phương pháp này giúp đánh giá các thông số kỹ thuật để phát hiện ra các vấn đề, xu hướng giải pháp chính để phát triển các dòng chữ ký số lượng tử trong tương lai.
- Phân tích hệ thống: bài báo đang đề cập đến một lĩnh vực phức tạp, phương pháp phân tích hệ thống được sử dụng để đưa ra cái nhìn toàn diện về các mối quan hệ giữa các yếu tố khác nhau trong lĩnh vực đó.

3. Kết quả và thảo luận

3.1. Giới thiệu các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm

Các lược đồ chữ ký số dựa trên hàm băm (hash-based signature) gần đây đã thu hút được nhiều sự chú ý như một sự thay thế tiềm năng cho các lược đồ chữ ký số truyền thống khi các máy tính lượng tử quy mô lớn được xây dựng [1-5, 9]. Lý do chính có lẽ

là do các ước tính về độ an toàn - cũng là để an toàn chống lại các cuộc tấn công được hỗ trợ bởi máy tính lượng tử. Điều này phân biệt chữ ký dựa trên hàm băm với các lược đồ chữ ký hậu lượng tử khác [1]. Ngoài ra, chữ ký dựa trên hàm băm không cần các phép toán tính toán phức tạp như số học số nguyên lớn [5-9]. Lược đồ chữ ký số dựa trên hàm băm thì thường có kích thước chữ ký lớn nhưng thời gian thực hiện ký khá là tương đối nhanh. Thêm nữa, các lược đồ chữ ký dựa trên hàm băm có kích thước chữ ký khá lớn nhưng kích thước khóa công khai lại nhỏ. Cho tới nay, các lược đồ chữ ký số dựa trên hàm băm đang mang lại độ tin tưởng lớn nhất do các lược đồ này có độ an toàn chỉ dựa trên độ an toàn của hàm băm được sử dụng thay vì một số bài toán khó như các bài toán khó kháng lượng tử khác [14]. Sau đây sẽ giới thiệu nhanh một số điểm cơ bản về các thuật toán chữ ký số kháng lượng tử.

- Lamport One-Time Signature Scheme (Lamport OTS): Đây là một thuật toán chữ ký số đơn giản và hiệu quả, được đề xuất bởi Leslie Lamport vào năm 1979. Thuật toán này sử dụng một bộ đôi khóa bí mật và khóa công khai cho mỗi bit của thông điệp cần ký [1]. Mặc dù rất hiệu quả, Lamport OTS không thể tái sử dụng và đòi hỏi một bộ khóa khổng lồ.

- Merkle Signature Scheme: Được đề xuất bởi Ralph Merkle vào năm 1987, thuật toán này giải quyết vấn đề về số lượng khóa trong Lamport OTS bằng cách sử dụng một cây Merkle hash. Mỗi lá cây được ký bằng Lamport OTS và các giá trị hash của các lá cây được sử dụng để tạo ra chữ ký [18].

- Hash-based Online Short-term Signatures (HORS): Được giới thiệu bởi Patrick Horster và Joachim von zur Gathen vào năm 2011, HORS sử dụng một bộ khóa bí mật duy nhất để tạo ra chữ ký cho nhiều thông điệp [1].

Ngoài ra, trong cuộc thi tìm kiếm Chữ ký số kháng lượng tử của NIST, đã loại bỏ một số thuật toán sau hai vòng đánh giá, chỉ còn lại 7 thuật toán, và tất cả các thuật toán đó đều được đánh giá cao về độ an toàn và hiệu suất. Các thuật toán đó bao gồm: SPHINCS+, Picnic, Rainbow, SPHINCS-256, GeMSS, Gravity-SPHINCS, qTESLA. Trong số 7 thuật toán chữ ký số kháng lượng tử dựa trên hàm băm được chọn sau hai vòng đánh giá của cuộc thi Chữ ký số Kháng lượng tử của NIST [21], chúng tôi lựa chọn giới thiệu thêm về 2 thuật toán:

- SPHINCS+: SPHINCS+ là một thuật toán chữ ký số kháng lượng tử dựa trên hàm băm, được giới thiệu bởi Johannes Buchmann, Erik Dahmen, Andreas Hülsing, Alexander May và Christof Paar vào năm 2017. SPHINCS+ sử dụng một số kỹ thuật mới như cây hash tùy chỉnh (custom hash tree), tránh sử dụng lặp lại (avoiding repetition), và khử nhiễu (noise reduction) để giảm thiểu kích thước của chữ ký và chứng chỉ, cũng như tăng cường tính bảo mật của thuật toán [13].

- Picnic: Picnic là một thuật toán chữ ký số kháng lượng tử dựa trên hàm băm, được giới thiệu bởi Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, Krzysztof Pietrzak, Guy N. Rothblum, và John Steinberger vào năm 2019 [21]. Thuật toán này sử dụng một hàm băm được gọi là SPHINCS+-Haraka để tạo ra chữ ký. Picnic được thiết kế để đáp ứng các yêu cầu bảo mật khắt khe của các ứng dụng thực tế, nhưng vẫn đảm bảo tính khả dụng và hiệu suất.

3.2. Cách hoạt động của các thuật toán

Để hiểu cách hoạt động của các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm, ta có thể xem xét chi tiết về cách chúng tạo khóa, tạo chữ ký số và cách xác minh chữ ký số. Tất cả sẽ được minh họa thông qua sơ đồ thuật toán với mã giả (pseudocode).

3.2.1. Lược đồ Lamport

Lược đồ chữ ký dựa trên hàm băm đầu tiên được ghi nhận trong một báo cáo kỹ thuật của Lamport [1], mặc dù lược đồ tương tự đôi khi được gọi là "lược đồ chữ ký một lần của Lamport-Diffie". Xem Hình 1 để biết minh họa về một chữ ký được tạo ra và các Thuật toán 3.2.1, 3.2.2 và 3.2.3 cho các mô tả bằng mã giả.

Thuật toán 3.2.1. Lamport.KeyGen () (n, k, F)

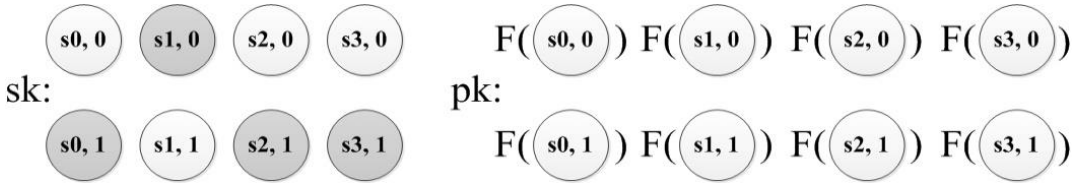
for $i \in \{1, \ldots, n\}$ **do**
  for $j \in \{0, 1\}$ **do**
    $s_{i,j} \leftarrow \{0, 1\}^k$
    $p_{i,j} \leftarrow F(s_{i,j})$
  end for
end for
return pk = $(p_{1,0}, p_{1,1}, \ldots, p_{n,0}, p_{n,1})$; sk = $(s_{1,0}, s_{1,1}, \ldots, s_{n,0}, s_{n,1})$

Thuật toán 3.2.2. Lamport.Sign (m, $s_{i,j} \in k$) (n, F)

($m_1, \ldots, m_n$) = m với $m_i \in \{0, 1\}$
for $i \in \{1, \ldots, n\}$ do
  $\sigma i = si\ m, i$
end for
return $\sigma = (\sigma_1, \ldots, \sigma_n)$

Thuật toán 3.2.3. Lamport.Verify (m, $\sigma_i \in \sigma$, $p_{i,j} \in$ pk) (n, F)

($m_1, \ldots, m_n$) = m
for $i \in \{1, \ldots, n\}$ **do**
  if $F(\sigma_i) \neq p_{i\ m,\ i}$ **then**
    return False
  end if
end for
return True

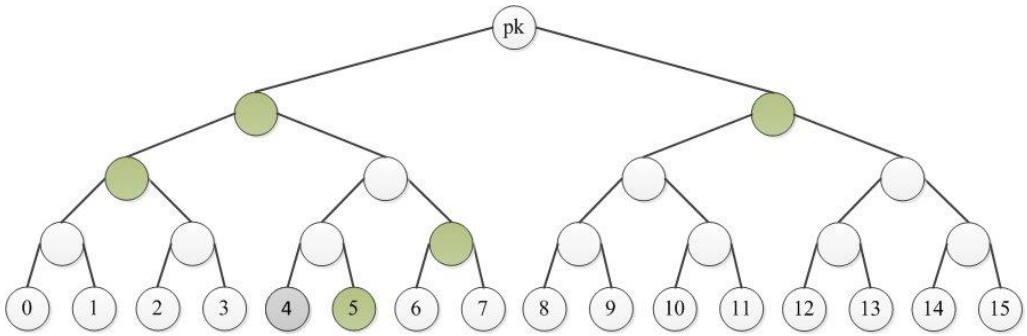


Hình 1: *Lược đồ chữ ký của Lamport. Ở đây, m = 1011 được sử dụng như một thông điệp ví dụ và các nút màu xám chỉ ra những bí mật được tiết lộ*

3.2.2. Lược đồ chữ ký số Merkle

Lược đồ chữ ký Lamport được mô tả trên là đại diện cho các lược đồ chữ ký một lần. Điều này thường không phải là hữu ích trong thực tế. Phần này sẽ trình bày về cách xây dựng lược đồ chữ ký nhiều lần từ lược đồ chữ ký một lần, như đề xuất của Merkle [1]. Ở đây lược đồ chữ ký nhiều lần khá khác với lược đồ chữ ký số truyền thống: số lượng chữ ký tối đa có thể được tạo ra là cố định và khóa bí mật liên tục phát triển khi nó được sử dụng để tạo ra chữ ký. Điều này được giải quyết chính thức trong lược đồ chữ ký số Merkle *với* ngữ cảnh của lược đồ chữ ký dựa trên hàm băm [13].

Mô tả lược đồ chữ ký Merkle (MSS) được trình bày trong các thuật toán 3.2.4a, 3.2.5 và 3.2.6, được tham số hóa bằng lược đồ chữ ký một lần trừu tượng (abstract) OTS. Ở đây h là chiều cao của cây Merkle, cho phép người ký tạo ra 2^h chữ ký (xem Hình 2). Chúng ta định nghĩa một chương trình con để xây dựng cây Merkle trong thuật toán 3.2.4b. Các quy trình sinh khóa và ký được mô tả trong các thuật toán 3.2.4a và 3.2.5 là khá tốn kém để thực thi, cả về tính toán và sử dụng bộ nhớ.



Hình 2: *Minh họa cây băm nhị phân với p = 16*

Thuật toán 3.2.4. Merkle.KeyGen () (*h, OTS, H*)

1: **for** $i \in \{0, \ldots, 2^{h-1}\}$ **do**
2: $\quad pk_i, sk_i \leftarrow OTS.\text{KeyGen}()$
3: **end for**
4: $node_{i,j} \leftarrow \text{Merkle.BuildTree}(pk_0, ..., pk_{2^{h-1}})$ // Với $0 \le i \le h$ và $0 \le j < 2^{h-i}$
5: $pk \leftarrow node_{h,0}$
6: **return** $pk, sk_0 = (0; sk_0, ..., sk_{2^{h-1}})$

Thuật toán 3.2.4b. Merkle.BuildTree ($leaf_0, \ldots, leaf_{2^{h-1}}$) (*h, H*)

1: **for** $j \in \{0, \ldots, 2^{h-1}\}$ **do**
2: $\quad node_{0,j} \leftarrow leaf_j$
3: **end for**
4: **for** $i \in \{1, \ldots, h\}$ **do**
5: $\quad$ **for** $j \in \{0, \ldots, 2^{h-i} - 1\}$ **do**
6: $\quad\quad node_{i,j} \leftarrow H(node_{i-1,2j}, node_{i-1,2j+1})$
7: $\quad$ **end for**
8: **end for**
9: **return** $node_{i,j}$ **for** $i \in \{1, \ldots, h\}, j \in \{0, \ldots, 2^{h-1}\}$

Thuật toán 3.2.5. Merkle.Sign (m, sk_{idx}) (h, *OTS*)

1: $(idx; sk_0, ..., sk_{2^h-1}) = sk_{idx}$
2: **if** $idx > 2^h - 1$ **then**
3: **return** Fail
4: **end if**
5: $sk_{idx+1} \leftarrow (idx + 1; sk_0, ..., sk_{2^h-1})$
6: $\sigma_{OTS} \leftarrow OTS.Sign(m, sk_{idx})$
7: **for** $j \in \{0, \ldots, 2^h - 1\}$ **do**
8: $pk_j \leftarrow OTS.\text{RecoverPK}(sk_j)$
9: **end for**
10: $node_{i,j} \leftarrow \text{Merkle.BuildTree}(pk_0,..., pk_{2^h-1})$ // Với $0 \le i \le h$ và $0 \le j < 2^{h-i}$
11: **for** $i \in \{0, \ldots, h - 1\}$ **do**
12: $auth_i \leftarrow node_{i,\lfloor \frac{idx}{2^i} \rfloor \oplus 1}$
13: **end for**
14: $\sigma = (idx, pk_{idx}, \sigma_{OTS}, auth)$
15: **return** σ, sk_{idx+1}

Thuật toán 3.2.6. Merkle.Verify (m, σ, pk) (h, *OTS*, *H*)

1: $(idx, pk_{idx}, \sigma_{OTS}, auth) = \sigma$
2: **if** ¬ OTS.Verify (m, σ_{OTS}, pk_{idx}) **then**
3: **return** False
4: **end if**
5: $node_0 \leftarrow pk_{idx}$
6: **for** $i \in \{0, \ldots, h - 1\}$ **do**
7: **if** $\left\lfloor \frac{idx}{2^i} \right\rfloor \bmod 2 = 0$ **then**
8: $node_{i+1} \leftarrow H(node_i, auth_i)$
9: **else**
10: $node_{i+1} \leftarrow H(auth_i, node_i)$
11: **end if**
12: **end for**
13: **return** $node_h$ = pk

3.2.3. Lược đồ chữ ký số HORS

HORS sử dụng một bản địa phương của hàm băm SHA-3 và một tập hợp các chuỗi ngẫu nhiên đã được tạo ra trước đó. Các chuỗi này được sử dụng để tạo ra một chuỗi khóa riêng tư, một chuỗi khóa công khai và một chuỗi chữ ký [1]. Thuật toán này tạo ra chữ ký bằng cách ký một thông điệp với khóa riêng tư và sau đó mã hóa các giá trị ký bằng cách sử dụng khóa công khai.

HORST là một biến thể của HORST [1], đưa thêm một cây Merkle trên đỉnh khóa công khai HORS để đạt được độ nén khóa.

Giống như trong Lamport OTS, khóa bí mật HORST bao gồm một mảng các giá trị ngẫu nhiên (có thể được tạo ra từ một mầm). Định nghĩa sk = (s_0, …, s_{t-1}) bao gồm t giá trị k bit ngẫu nhiên, trong đó t là lũy thừa của hai. Như trước đây, chúng ta áp dụng hàm

một chiều F để thu được $p_i = F(s_i)$. Các giá trị p_i sau đó được đặt trên các nút lá của cây Merkle, và kết hợp để lấy ra nút gốc pk.

Các thuật toán 3.2.7, 3.2.8 và 3.2.9 mô tả về HORST. Ở đây HORST thực hiện xác minh bằng khôi phục khóa công khai. Dựa vào chương trình con Merkle.BuildTree được giới thiệu trong thuật toán 3.2.4b, đặt h = *log*(t).

Thuật toán 3.2.7. HORST.KeyGen () (k, t, F, H)

for $i \in \{0, \ldots, t-1\}$ **do**
  $s_i \leftarrow \{0,1\}^k$
  $p_i \leftarrow F(s_i)$
end for
$node_{i,j} \leftarrow$ MerkleBuildTree($p_0, \ldots, p_{t-1}$) // Với $0 \le i \le \log(t)$ và $0 \le j < (t/2^i)$
pk $\leftarrow$ node$_{log(t),0}$
return pk, sk = ($s_0, \ldots, s_{t-1}$)

Thuật toán 3.2.8. HORST.Sign (m, s_i sk) (x, m, t, F, H)

for $i \in \{0, \ldots, t-1\}$ **do**
  $p_i \leftarrow F(s_i)$
end for
$node_{i,j} \leftarrow$ MerkleBuildTree($p_0, \ldots, p_{t-1}$) // Với $0 \le i \le \log(t)$ và $0 \le j < (t/2^i)$
$\left(m_0, \ldots, m_{\frac{\|m\|}{\log(t)}}\right) = m$ // Chia m thành các chuỗi log(t) bit
for idx $\in \left\{0, \ldots, \frac{\|m\|}{\log(t)} - 1\right\}$ ***do***
  for $i \in \{0, \ldots, \log(t) - x - 1\}$ **do**
    $auth_{idx\,i,} \leftarrow node_{i,\left\lfloor \frac{m_{idx}}{2^i} \right\rfloor \oplus 1}$
  end for
  $\sigma_{idx} \leftarrow s_{midx}, auth_{idx}$
end for
$\sigma_x = node_{x,j}$ for $j \in \{0, \ldots, 2^x - 1\}$
return σ_{idx} **for** idx $\in \in \left\{0, \ldots, \frac{\|m\|}{\log(t)} - 1\right\}, \sigma_x$

Thuật toán 3.2.9. HORST.Verify (m, σ, pk)

σ_{idx} **for** $idx \in \left\{0, \ldots, \frac{\|m\|}{\log(t)} - 1\right\}, \sigma_x \leftarrow \sigma$
$\left(m_0, \ldots, m_{\frac{\|m\|}{\log(t)}}\right) = m$ // Chia m thành các chuỗi log(t) bit
$node_{i,j} \leftarrow$ MerkleBuildTree($p_0, \ldots, p_{t-1}$) // Với $0 \le i \le \log(t)$ và $0 \le j < (t/2^i)$
for $idx \in \left\{0, \ldots, \frac{\|m\|}{\log(t)} - 1\right\}$ **do**
  $sm_{idx}, auth_{idx} \leftarrow \sigma_{idx}$
  $node_{idx,0} \leftarrow F(sm_{idx})$
  for $i \in \{0, \ldots, \log(t) - x - 1\}$ **do**

8: **if** $\frac{\|m\|}{\log(t)} \bmod 2 = 0$ **then**
9: $node_{idx,\, i+1} \leftarrow H(node_{idx,i}, auth_{idx,i})$
10: **else**
11: $node_{idx,i+1} \leftarrow H(auth_{idx,i}, node_{idx,i})$
12: **end if**
13: **end for**
14: $if\ node_{idx,\log(t)-x} \neq node'_{0,\frac{m_{idx}}{2^{log(t)-x}}}$ ***then***
15: **return** False
16: **end if**
17: **end for**
18: **return** $node'_{x,0}$ = pk // σ_x gồm 2^x nút, vì vậy $node'_{x,0}$ là gốc

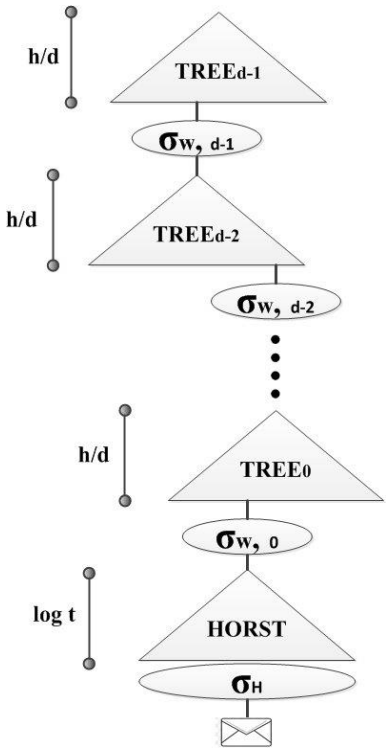
Điều khiển HORST trở thành một lược đồ chữ ký vài lần chứ không phải là OTS là tỷ lệ giữa số nút lá *t* và thông điệp *m*. Miễn là sự khác biệt đủ lớn, chỉ một tập con nhỏ của khóa bí mật được tiết lộ cho mỗi chữ ký. Điều quan trọng, một kẻ giả mạo sẽ không thể tự do lựa chọn các thông điệp. Để đạt được tính không thể giả mạo (có thể có trong phạm vi "một vài" truy vấn thông điệp), trước tiên cần phải ngẫu nhiên hóa thông điệp và lấy ra một bản tóm tắt thông điệp *n* bit; đây là "thông điệp" được ký bằng HORST [1]. Điều này có thể được thực hiện bằng cách cho thêm một giá trị giả ngẫu nhiên *R* cùng thông điệp làm đầu vào cho hàm một chiều, tạo ra *R* một cách tất định từ khóa bí mật. Giá trị *R* sau đó cần phải được gửi kèm trong chữ ký, để người xác minh có thể tính toán lại giá trị băm từ thông điệp. Điều này được đề cập chi tiết hơn trong [12].

3.2.4. Lược đồ chữ ký số SPHINCS+:

Tất cả các cấu trúc chữ ký Merkle mà chúng ta đã xét đại diện ở phần 3.2.2 đều có trạng thái. Để đảm bảo rằng cặp khóa chữ ký một lần trên nút lá không được sử dụng lại, ít nhất sẽ phải duy trì chỉ mục của nút lá được sử dụng lần cuối [4]. Ngoài ra, trạng thái có thể được sử dụng để tối ưu hóa việc tạo chữ ký bằng cách sử dụng các thuật toán duyệt cây [8,10].

Trạng thái cũng đi kèm với những nhược điểm đáng chú ý. Đặc biệt, nó làm phức tạp rất nhiều việc quản lý khóa [16]: chia sẻ khóa giữa các thiết bị đột nhiên trở thành một vấn đề đồng bộ hóa phức tạp và việc khôi phục từ các bản sao lưu có thể dễ dàng dẫn đến khôi phục không mong muốn. Điều này mâu thuẫn với các định nghĩa API điển hình cũng như nhiều giả thiết mà người triển khai và người thực hiện có thể đưa ra, đến mức Adam Langley đã gọi các chữ ký dựa trên hàm băm trạng thái là "một khẩu súng thần công khổng lồ". Trên thực tế, một giao diện như vậy với một khóa bí mật thay đổi sẽ xung đột trực tiếp với định nghĩa của lược đồ chữ ký số [1].

Trong phần này, sẽ xem xét SPHINCS (post-quantum stateless hash based digital signature scheme): một lược đồ chữ ký dựa trên băm [12] không trạng thái hay SPHINCS$^+$ (được cải tiến nhiều lần từ lược đồ gốc SPHINCS) (xem Hình 3). Lược đồ này chứng minh rằng, với cái giá phải trả về hiệu năng, có thể làm cho các lược đồ dựa trên hàm băm không trạng thái trở nên thực tế [5].



Hình 3: *Cấu trúc của lược đồ chữ ký số nhiều lần SPHINCS.*

SPHINCS+ sử dụng một cặp khóa công khai - bí mật và một hàm băm SHA-256 [21]. Thuật toán này tạo ra một chuỗi chữ ký ngẫu nhiên bằng cách chọn ngẫu nhiên một số lượng hợp lệ [21] các cây Merkle, mỗi cây có một tập hợp khóa riêng tư độc lập và được kết hợp với các cây khác thông qua một chức năng băm khối cuối cùng của cây đó. Khi chữ ký được tạo ra, người ký sử dụng khóa riêng tư của mình để ký một thông điệp và sau đó ký tiếp các giá trị bổ sung để tạo thành chữ ký hoàn chỉnh. Do có nhiều phiên bản cải tiến ở mỗi bước thực hiện nên trong khuôn khổ bài báo review này sẽ trình bày vắn tắt các bước thực hiện [14, 21]. Các đánh giá và so sánh các dòng (phiên bản cải tiến khác nhau của SPHINCS) sẽ được thảo luận trong một nghiên cứu khác sâu hơn.

Cách tạo chữ ký số thực hiện các bước sau:

- ✓ Tạo một cặp khóa công khai - bí mật
- ✓ Tạo một chuỗi chữ ký ngẫu nhiên
- ✓ Sử dụng khóa riêng tư để ký thông điệp và các giá trị bổ sung

Cách xác minh chữ ký số, thực hiện các bước sau;

- ✓ Kiểm tra tính hợp lệ của các giá trị bổ sung
- ✓ Sử dụng khóa công khai để xác minh chữ ký

3.2.5. Lược đồ chữ ký số Picnic

Picnic được đánh giá là một trong những thuật toán chữ ký số kháng lượng tử tốt nhất hiện nay [21]. Điều này được chứng minh bởi việc nó đạt được nhiều giải thưởng và được sử dụng rộng rãi trong các ứng dụng bảo mật. Với mức độ bảo mật và hiệu suất tốt, Picnic được coi là một lựa chọn lý tưởng cho các ứng dụng yêu cầu bảo mật cao như ngân hàng trực tuyến, chứng khoán và các ứng dụng IoT [21].

Cơ chế hoạt động của lược đồ này bao gồm hai giai đoạn chính: tạo chữ ký (signing) và xác minh chữ ký (verifying). Trong giai đoạn tạo chữ ký, người sử dụng cần phải thực hiện các bước sau:

- Tạo ra một khóa bí mật (secret key) và một khóa công khai (public key) bằng cách sử dụng một thuật toán sinh khóa riêng biệt [21].
- Tạo ra một bản rõ (plaintext) của thông điệp cần ký (message), sau đó sử dụng một hàm băm (hash function) để chuyển đổi thông điệp này thành một giá trị băm (hash value) [21].
- Sử dụng các phép toán trên đường cong elip (elliptic curve operations) [21] để tạo ra chữ ký số.

Trong giai đoạn xác minh chữ ký, người nhận cần phải thực hiện các bước sau:

- Tạo ra một bản rõ của thông điệp cần xác minh.
- Sử dụng khóa công khai của người gửi để giải mã chữ ký số và tạo ra một giá trị băm mới từ thông điệp này.
- So sánh giá trị băm mới này với giá trị băm được gửi kèm theo chữ ký. Nếu hai giá trị này giống nhau, chữ ký được coi là hợp lệ.

Với tính năng kháng lượng tử, lược đồ chữ ký số Picnic được đánh giá là có khả năng chống lại các cuộc tấn công từ máy tính lượng tử [21], vì nó dựa trên các bài toán tính toán khó giải trên các hệ thống đường cong elip và các hàm băm chứ không dựa trên việc giải các bài toán trên mạng lưới lượng tử.

3.3. So sánh các thông số kỹ thuật của các thuật toán

Trong khuôn khổ nghiên cứu này, chúng tôi định hướng tổng hợp so sánh các thuật toán theo 3 hướng tiếp cận: tiếp cận theo hướng đánh giá độ phức tạp thuật toán, hiệu suất và độ tin cậy và cùng thực thi trên cơ sở phần cứng hoặc phần mềm.

3.3.1. Độ phức tạp tính toán

Tính toán độ phức tạp của các thuật toán trong từng bước có thể phụ thuộc vào nhiều yếu tố như cấu trúc của hệ thống, kích thước khóa và thông điệp, cách thức cài đặt thuật toán [1]. Sau đây là tổng hợp thông tin cơ bản về độ phức tạp tính toán của các thuật toán này thông qua kích thước khóa và độ dài thông điệp ở bước tạo chữ ký và xác thực chữ ký. Dữ liệu được tổng hợp trong Bảng 1.

Bảng 1: *Tổng hợp độ phúc tạp tính toán của các thuật toán*

Thuật toán	**Tạo chữ ký**	**Xác thực chữ ký**	**Ghi chú**
Lamport [1]	O(kl)	O(kl)	k là kích thước của khóa, l là độ dài của thông điệp
Merkle [1]	O(nlogn)	O(nlogn)	n là số lượng khóa
HORS [1]	O(nl)	O(nl)	n là số lượng khóa, l là độ dài của thông điệp
SPHINCS+[21]	O(nk^2)	O(nk^2)	n là số lượng khóa, k là kích thước của khóa
Picnic [21]	O(nlogn)	O(nlogn)	n là số lượng khóa

Tuy nhiên đây chỉ là độ phức tạp tính toán lý thuyết dựa trên kích thước khóa và thông điệp, trong thực tế sẽ phụ thuộc vào nhiều yếu tố như nền tảng phần cứng, cách triển khai. Đồng thời thông tin về độ phức tạp tính toán của các thuật toán được đưa ra trong các bài báo khác nhau tùy thuộc vào từng version của thuật toán tham khảo cụ thể.

3.3.2. Độ dài chữ ký, độ dài khóa, tính linh hoạt

Để thực hiện so sánh các thuật toán Merkle, Lamport, HORS, SPHINCS+ và Picnic, ta dựa trên các thông số quan trọng sau đây:

- Độ dài chữ ký (Signature length): Độ dài của chữ ký tạo ra bởi thuật toán, được tính bằng bit.
- Độ dài khóa (Key length): Độ dài của khóa được sử dụng bởi thuật toán, được tính bằng bit.
- Tính linh hoạt (Flexibility): Khả năng sử dụng với nhiều loại dữ liệu và ứng dụng khác nhau.

Bảng 2, tổng hợp các thông số mô tả trên của các thuật toán Merkle, Lamport, HORS, SPHINCS+ và Picnic.

Thông qua bảng cho thấy Merkle có độ dài chữ ký và khóa tương đối dài, tính linh hoạt cao. Lamport có độ dài chữ ký rất dài, độ dài khóa ngắn, tính linh hoạt thấp. HORS có độ dài chữ ký và khóa ngắn, tính linh hoạt cao. SPHINCS+ và Picnic có độ dài chữ ký và độ dài khóa khá ngắn, nên sẽ phù hợp với nhiều ứng dụng khác nhau. Trên cơ sở các thống kê trên cũng là một quan sát hỗ trợ các nhà phát triển ứng dụng lựa chọn.

Bảng 2: *Tổng hợp thông số về độ dài chữ ký, độ dài khóa và độ linh hoạt*

Thuật toán	Độ dài chữ ký	Độ dài khóa	Tính linh hoạt
Lamport [1]	1024-4096	512-1024	Thấp
Merkle [1]	256-512	256-512	Cao
HORS [1]	40-100	40-200	Cao
SPHINCS+[21]	128-512	128-256	Cao
Picnic [21]	128-512	128-256	Cao

3.3.3. Thực thi trên phần mềm, phần cứng

Để có thêm những góc nhìn mới hỗ trợ các nhà phân tích lựa chọn giải pháp ký số phù hợp và hiệu quả cho từng loại ứng dụng cụ thể, chúng tôi tổng hợp thêm những dữ liệu đã được công bố rải rác trong một số nghiên cứu của nhiều tác giả khác nhau. So sánh này dựa trên số phép tính trên 1 đơn vị thời gian là giây. Đồng thời để đảm bảo tính đồng nhất và có nhiều góc nhìn, các thuật toán cùng được cài đặt trên một cấu hình máy với 2 phương án sau đây và kết quả được tổng hợp trong Bảng 3.

• Phương án 1: Chạy trên phần mềm với cấu hình máy là CPU Intel Core i7-6700K với mã nguồn triển khai trong OpenSSL và kích thước khóa 256-bit:

• Phương án 2: Chạy trên phần cứng mô phỏng FPGA Zynq-7020 với mã nguồn triển khai trong Verilog và kích thước khóa 256-bit:

Tuy nhiên đây chỉ là các giá trị tham khảo và các số liệu thực tế có thể khác nhau tùy thuộc vào các yếu tố như nền tảng phần cứng, mã nguồn triển khai, kích thước khóa, số lượng vòng lặp, và kích thước thông điệp.

Bảng 3: *Tổng hợp số phép tính của các thuật toán trong các phương án đề xuất*

Thuật toán	Phương án 1	Phương án 2
Lamport [16, 17]	650	1,500
Merkle [16, 17]	10,000	30,000
HORS [16, 17]	8,000	7,000
SPHINCS+ [15]	20	180
Picnic [19]	4	20

3.4. Phân tích những hạn chế của các thuật toán và đề xuất các giải pháp để cải thiện hiệu quả và khả năng ứng dụng của chúng

Các hạn chế của các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm được thảo luận nhiều trong các nghiên cứu như [1-5], [16-19] và được khái quát như sau:

• Độ dài khóa: Một số thuật toán yêu cầu khóa rất dài để đảm bảo tính kháng lại các cuộc tấn công kháng lượng tử. Điều này đòi hỏi bộ nhớ lớn hơn và tốn kém hơn trong việc lưu trữ và truyền tải khóa.

• Tính hiệu quả: Các thuật toán có tính toán phức tạp cao hơn, làm cho việc tạo và xác minh chữ ký trở nên chậm hơn. Điều này đặc biệt quan trọng đối với các ứng dụng yêu cầu tốc độ xử lý nhanh, chẳng hạn như trong các giao dịch tài chính trực tuyến.

• Độ tin cậy: Các thuật toán kháng lượng tử dựa trên hàm băm đều dựa trên giả thuyết rằng hàm băm không thể bị tấn công bởi các cuộc tấn công kháng lượng tử. Tuy nhiên, giả thuyết này có thể không được chứng minh hoàn toàn đối với các hàm băm hiện tại, do đó độ tin cậy của thuật toán có thể không được đảm bảo.

Để cải thiện hiệu quả và khả năng ứng dụng của các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm, trên cơ sở các nghiên cứu và đánh giá trên và tổng hợp các kết quả từ các nghiên cứu cho thấy có thể sử dụng các giải pháp sau:

• Tối ưu hóa độ dài khóa: Các nghiên cứu có thể tìm cách tối ưu hóa độ dài khóa để giảm tải cho việc lưu trữ và truyền tải khóa [16, 19].

• Tối ưu hóa tính toán: Các thuật toán mới có thể được phát triển để tối ưu hóa tính toán và tăng tốc quá trình tạo và xác minh chữ ký [1, 3].

• Nghiên cứu các hàm băm mới: Các hàm băm mới có thể được nghiên cứu để đảm bảo tính kháng lại các cuộc tấn công kháng lượng tử [16].

• Sử dụng các thuật toán kết hợp: Sử dụng các thuật toán kết hợp khác nhau có thể cải thiện tính hiệu quả và độ tin cậy của hệ thống chữ ký số kháng lượng tử [14, 20].

3.5. Đánh giá khả năng ứng dụng của các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm

Dựa trên cơ sở các đặc trưng riêng của từng loại ứng dụng và tính chất nổi bật cũng như nhược điểm của mỗi thuật toán, các đánh giá về tiềm năng ứng dụng của các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm trong việc bảo vệ thông tin truyền tải được thảo luận trong nhiều kết quả nghiên cứu như [1-5], [10-16]… Cụ thể như sau:

• Các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm như Merkle, Lamport, HORS, SPHINCS+, Picnic đều có khả năng bảo vệ thông tin truyền tải trong nhiều lĩnh vực khác nhau, bao gồm ngân hàng, thương mại điện tử và chính phủ.

• Trong lĩnh vực ngân hàng, các thuật toán này có thể được sử dụng để bảo vệ thông tin nhạy cảm như thông tin tài khoản và giao dịch tài chính của khách hàng.

• Trong lĩnh vực thương mại điện tử, các thuật toán này có thể được sử dụng để xác thực chữ ký số của các bên tham gia trong các giao dịch trực tuyến, đảm bảo tính toàn vẹn và bảo mật của thông tin truyền tải.

• Trong lĩnh vực chính phủ, các thuật toán này có thể được sử dụng để bảo vệ các thông tin nhạy cảm như thông tin quân sự và chính sách quốc gia.

Tóm lại, các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm có thể được sử dụng trong nhiều ứng dụng bảo mật khác nhau, nơi mà tính bảo mật và độ tin cậy cao là rất quan trọng.

4. Kết luận

Bài báo tổng quan về các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm nói chung mà đặc biệt là các thuật toán đại diện như Merkle, Lamport, HORS, SPHINCS+ và Picnic đều có những ưu điểm và hạn chế riêng. Tuy nhiên, chúng đều là những phương pháp bảo mật hiệu quả trong việc chống lại các cuộc tấn công của kẻ xâm nhập và tấn công kháng lượng tử.

Các thuật toán này có khả năng ứng dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là trong lĩnh vực ngân hàng, thương mại điện tử và chính phủ. Chúng có thể được sử dụng để bảo vệ thông tin truyền tải, đảm bảo tính toàn vẹn và bảo mật dữ liệu. Vì vậy, các thuật toán này đang được sử dụng rộng rãi trong thực tế và tiếp tục được phát triển để cải thiện hiệu quả và độ tin cậy của chúng.

Tuy nhiên, với sự phát triển nhanh chóng của công nghệ, các kẻ tấn công cũng không ngừng cải tiến kỹ thuật tấn công của mình. Vì vậy, cần tiếp tục nghiên cứu và phát triển các thuật toán bảo mật mới để đáp ứng được các thách thức của tương lai khi máy tính lượng tử trở thành thiết bị phổ dụng.

Trong tương lai, các thuật toán chữ ký số kháng lượng tử dựa trên hàm băm sẽ tiếp tục được ứng dụng rộng rãi trong các lĩnh vực bảo mật thông tin. Tuy nhiên, để đáp ứng được các yêu cầu của thị trường và phát triển của công nghệ, các nhà nghiên cứu mật mã cần tập trung vào việc nghiên cứu và phát triển các thuật toán mới, đồng thời cải tiến các thuật toán hiện tại để đảm bảo tính toàn vẹn và bảo mật dữ liệu trong tương lai.

TÀI LIỆU THAM KHẢO

[1] Johannes Buchmann, Erik Dahmen, and Andreas Hülsing, "Post-quantum signatures," In *Advances in Cryptology - ASIACRYPT 2016, TU Darmstadt (in Germany)*, pp. 121-149, 2016.

[2] Daniel J. Bernstein, Nadia Heninger, Paul Lou, Luke Valenta, "Post-quantum RSA," *Philadelphia (in USA)*, pp. 1-20, 2017. [Online], Available: https://ia.cr/2017/351.

[3] Daniel J. Bernstein, Nadia Heninger, Paul Lou, and Luke Valenta. "Post-quantum RSA in a world of quantum attacks," *In Advances in Cryptology - CRYPTO 2017, Philadelphia, PA 19103 (in USA)*, pp. 293-322, 2017.

[4] Bernstein, D. J., Hopwood, D., Hülsing, A., Lange, T., Niederhagen, R., Papachristodoulou, L., Wilcox-O'Hearn, Z."SPHINCS, "Practical Stateless Hash-Based Signatures," *Lecture Notes in Computer Science*, pp. 368-397, 2015. https://doi.org/10.1007/978-3-662-46800-5_15. DOI: 10.1007/978-3-662-46800-5_15

[5] A. Langley, D. O. Martin, and B. Moeller. ''Post-quantum elliptic curve cryptography for the internet," *Proceedings of the 2017, ACM SIGSAC Conference on Computer and Communications Security*, pp. 1217-1231, 2017. DOI: 10.1145/3133956.3136065

[6] I. Dinur and N. Livne, "Breaking the merkle signature scheme using fault attacks," *Proceedings of the 2014 IEEE Symposium on Security and Privacy*, pp. 131-145, 2014.

[7] Michael Naehrig and Patrick Longa, "Speeding up hash-based signatures with the dangling pointers technique," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 51-62, 2016.

[8] Johannes Buchmann, Erik Dahmen, Michael Szydlo, "Hash-based Digital Signature Schemes," *Nature*, vol. 549, pp. 35-91, 2017. DOI: 10.1038/nature23461

[9] Boneh, D., Zhandry, M., "Secure Signatures and Chosen Ciphertext Security in a Quantum Computing World," *Lecture Notes in Computer Science*, pp. 361-379, 2013. https://doi.org/10.1007/978-3-642-40084-1_21. DOI: 10.1007/978-3-642-40084-1_21

[10] Yevgeniy Dodis, Jonathan Katz, Leonid Reyzin, Adam Smith, "HORS with Applications to Verifiable Random Functions," 2008.

[11] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Zooko Wilcox-O'Hearn, "Practical Stateless Hash-Based Signatures," *Elisabeth Oswald and Marc Fischlin*, vol. 9056, pp. 368-397, 2015. DOI: 10.1007/978-3-662-46800-5_15

[12] Aumasson, J.-P., Endignoux, G., "Improving Stateless Hash-Based Signatures," *Topics in Cryptology - CT-RSA 2018*, pp. 219-242, 2018. DOI: 10.1007/978-3-319-76953-0_12

[13] Andreas Hülsing, "XMSS: eXtended Merkle Signature Scheme", *TU Eindhoven-Netherlands*, ISSN: 2070-1721, pp. 20-40, 2018. DOI: 10.17487/RFC8391

[14] Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen, "Introduction to Post-Quantum Cryptography," *Chicago and Darmstadt*, pp. 1-234, 2009. DOI: 10.1007/978-3-540-88702-7

[15] L. Chen, Y. Chen, X. Wang, and H. Li, "SPHINCS+: robust stateless hash-based signatures," *in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[16] D. J. Bernstein, T. Lange, and R. Niederhagen, "Post-quantum cryptography," Springer, 2019.

[17] D. J. Bernstein, "Introduction to post-quantum cryptography," *Department of Computer Science, University of Illinois at Chicago*, pp. 1-18, 2009. DOI: 10.1007/978-3-540-88702-7_1

[18] Daniel J. Bernstein, Tanja Lange, "Post-quantum cryptography," *Nature*, vol. 549, pp. 188-194, 14 September 2017. DOI: 10.1038/nature23461

[19] A. Couvreur, A. Otmani, and J-P. Tillich, "Polynomial time attack on Wild McEliece over quadratic extensions," *IEEE Trans. Information Theory*, vol. 63(1), pp. 404- 427, 2017. DOI: 10.1109/TIT.2016.2574841

[20] NISTIR7977, "NIST Cryptographic Standards and Guidelines Development Process," *NIST* (in US. Department of Commerce), pp. 1- 22, 2016.

[21] Elaine Barker, William Barker, William Burr, William Polk, Miles Smid, "*NIST special publication 800-57,* Guidance on the selection and use of cryptographic algorithms," *NIST* (in US. Department of Commerce), pp.62-71, 2012.

ABSTRACT

AN OVERVIEW OF QUANTUM RESISTANCE DIGITAL SIGNATURES BASED ON HASH FUNCTIONS

Do Thi Bac, Bounsaveng Khit

University of Information and Communication Technology, Thai Nguyen University, Vietnam

Received on 31/3/2023, accepted for publication on 12/5/2023

Facing the challenge of developing quantum computers, quantum-resistant digital signature algorithms have been developed based on hash functions and have received the attention of many scientists. This paper focuses on evaluating hash function-based quantum-resistant digital signature algorithms and their applicability in protecting transmitted information. Analyzing, synthesizing and comparing the strengths and limitations of the algorithms is the main research method of the paper. It also highlights the challenge of applying them in different fields such as banking, e-commerce, and government. The article is useful for those who are just starting to study quantum-resistant digital signatures, helping them get an overview of the field. It also assists regulators, governments and other organizations in making decisions and policies regarding digital signature security platforms.

Keywords: Quantum-resistant; digital signature; hash function; information security; cryptography.