Giới thiệu về thuật toán hàm băm Whirlpool

Đinh Tiến Thành, Cao Minh Tuấn, Vũ Bá Linh

Tóm tắt: Hàm băm Whirlpool [10] được đề xuất trong dự án NESSIE vào năm 2000, hàm băm này dựa trên mã khối và được đánh giá là an toàn. Hàm nén sử dụng mã khối W (được xem như là một biến thể của AES) được thiết kế dành cho hàm băm và không sử dụng cho hàm mã hóa nhằm khắc phục điểm yếu của các hàm băm dựa trên mã khối. Trong bài báo này sẽ trình bày các đặc trưng an toàn của hàm băm Whirlpool dựa trên cấu trúc, các phép biến đổi và một số tấn công lên hàm băm.

I. GIỚI THIỆU CHUNG

Whirlpool là một hàm băm mật mã được thiết kế bởi Vincent Rijmen và Paulo Barreto vào năm 2000. Whirlpool được đề xuất trong dự án NESSIE (New European Schemes for Signatures, Intergrity, and Encryption) và được công bố trong chuẩn ISO/IEC 10118-3 [7, 10].

Hàm băm Whirlpool được thiết kế dựa trên cấu trúc Merkle – Damgard sửa đổi. Hàm nén dựa trên mã khối tựa AES với mỗi khối mã có độ dài 512 bit và khóa 512 bit. Đầu vào của thuật toán là một thông báo với độ dài tối đa (2^{256}-1) bit và tạo đầu ra là giá trị tóm lược thông báo 512 bit. Whirlpool có sử dụng một mã khối được thiết kế đặc biệt để sử dụng cho các hàm băm và không sử dụng như một hàm mã hóa độc lập [8, 10]. Hàm vòng và lược đồ khóa được thiết kế theo chiến lược vệt lan rộng (Wide Trail Strategy) [7], một thiết kế được chứng minh là có khả năng chống lại các thám mã tuyến tính và vi sai nổi tiếng.

Bố cục của bài báo: Sau mục giới thiệu chung, mục II trình bày về cấu trúc hàm băm Whirlpool. Mục III trình bày về độ an toàn của hàm băm Whirlpool. Mục IV thống kê kết quả của một số tấn công lên hàm băm Whirlpool, kết quả cho thấy sự phức tạp và độ khó của các tấn công lên hàm băm này, mục V trình bày hiệu suất thực thi trên phần mềm của hàm băm Whirlpool và cuối cùng là Kết luận.

II. CẤU TRÚC HÀM BĂM WHIRLPOOL

A. Cấu trúc logic

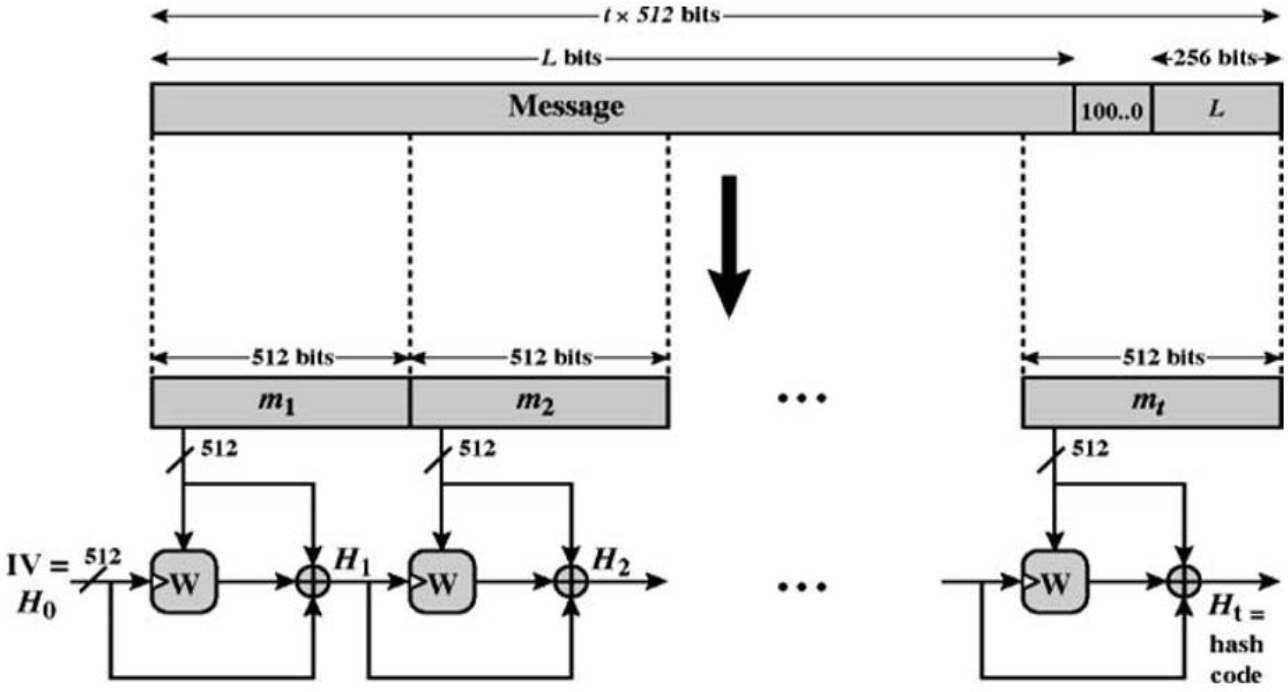
Với một thông báo bao gồm một chuỗi các khối $m_1;m_2;...;m_t$, hàm băm Whirlpool được thực hiện như sau [7]:

$H_0 \leftarrow IV$

$H_i = E(H_{i-1}, m_i) \oplus H_{i-1} \oplus m_i, i = 1,2,...,t$

Giá trị mã băm thu được cuối cùng là H_t

Hàm băm Whirlpool được thiết kế dựa trên cấu trúc Merkle – Damgard sửa đổi và được mô tả như Hình 1:



Hình 1: Cấu trúc hàm băm Whirlpool

Đầu vào của thuật toán là một thông báo với độ dài $\leq (2^{256}-1)$ bit và tạo đầu ra là mã băm 512-bit. Các đầu vào được xử lý trong khối 512-bit. Hình 1 mô tả quá trình biến đổi một thông báo đầu vào để cho mã băm đầu ra. Việc xử lý bao gồm các bước sau đây:

Bước 1: Thông báo được đệm thêm bit để độ dài là $(2t-1)\times 256$ bit.

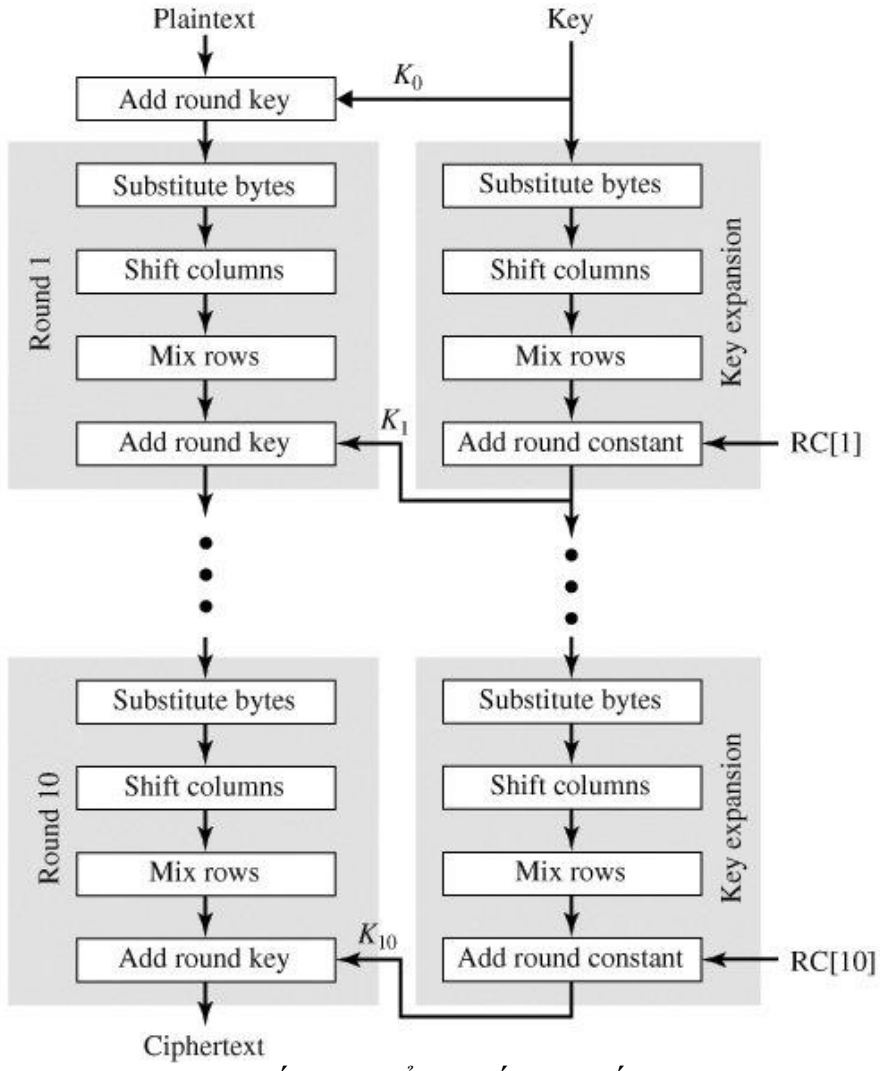
Bước 2: Một khối 256 bit được nối thêm vào thông báo. Khối này là một số nguyên không dấu 256-bit và chứa độ dài bit của thông báo ban đầu.

Kết quả của hai bước đầu mang lại một thông báo là bội số của 512 bit. Trong Hình 1, thông báo mở rộng được biểu diễn như chuỗi khối 512-bit $m_1; m_2; ...; m_t$, do đó tổng độ dài mở rộng của thông báo là t×512 bit.

Bước 3: Khởi tạo ma trận băm: Một ma trận 8×8 byte được sử dụng để lưu giữ kết quả trung gian và kết quả cuối cùng của hàm băm. Ma trận được khởi tạo gồm toàn bit-0.

Bước 4: Quá trình băm thông báo với khối 512-bit (64-byte) như Hình 2.

B. Mã khối W



Hình 2: Cấu trúc tổng thể mã khối W

Thuật toán mã khối *W* thực hiện với khối bản rõ 512-bit, khóa 512-bit và tạo ra một khối bản mã 512 bit [7]. Thuật toán này sử dụng bốn hàm biến đổi khác nhau:

- Subbytes (SB): Thay thế byte bằng một bảng thay thế S-box
- ShiftColumns (SC): Dịch chuyển cột của trạng thái trung gian theo các offset khác nhau.
- MixRows (MR): Trộn dữ liệu trong mỗi hàng của bảng trạng thái
- AddroundKey(AK): Cộng khóa.

Khối bản rõ đầu vào đầu tiên được biến đổi qua hàm Addroundkey, sau đó được biến đổi qua 4 hàm biến đổi tuần tự SubBytes, ShiftColumn, MixRows, AddRoundkey liên tục trong 10 vòng. Các khóa trong các vòng mã hóa được sinh bởi lược đồ tạo khóa, ở đây AddRoundConstant (AC) là phép cộng hằng số vòng RC (Round Constant) vào trạng thái khóa. Trạng thái cuối cùng được chuyển thành đầu ra.

III. MỘT SỐ ĐẶC TRƯNG AN TOÀN TRONG THIẾT KẾ CÁC THÀNH PHẦN CỦA HÀM BĂM WHIRLPOOL

A. Mục tiêu an toàn trong thiết kế

Với đầu ra của Whirlpool là một mã băm có chiều dài cố định là *n*-bit, các nhà thiết kế Whirlpool [10] đã đặt ra các mục tiêu an toàn sau:

- Cho đầu vào bất kỳ, độ phức tạp tính toán dự kiến để tạo ra một va chạm của mã băm đầu ra là $2^{\frac{n}{2}}$.

- Độ phức tạp tính toán để tìm ra thông báo đầu vào của mã băm là 2^n.

- Độ phức tạp tính toán để tìm ra một thông báo thứ hai có cùng mã băm là 2^n.

- Không thể phát hiện mối tương quan hệ thống giữa bất kỳ sự kết hợp tuyến tính của các bit đầu vào với sự kết hợp tuyến tính của các bit trong giá trị băm hoặc không thể dự đoán các bit trong giá trị băm sẽ thay đổi như thế nào khi các bit đầu vào nhất định được đảo lộn (nghĩa là có thể chống lại các tấn công tuyến tính và tấn công lượng sai).

Một tấn công lên Whirlpool chỉ được xem là thành công, nếu tấn công này chứng minh được một trong những mục tiêu an toàn theo thiết kế trên bị phá vỡ.

B. Độ an toàn của mã khối W

Để đánh giá độ an toàn của mã khối *W*, ta so sánh một số tính chất mật mã của S-box được sử dụng trong thuật toán hàm băm Whirlpool với S-box trong thuật toán AES như Bảng 1:

Bảng 1: So sánh đặc trưng S-box 8bit của mã khối W và AES [5,6]

Đặc trưng	Whirlpool	AES
Cân bằng (Balanced)	Có	Có
Bậc đại số (Degree)	7	7
Độ lượng sai (Diff(*S*))	8	4
Độ tuyến tính (Lin(*S*))	28	16

So sánh đặc trưng S-box 8bit của Whirlpool và S-box 8bit của AES theo Bảng 1, ta thấy:

- Về đặc trưng cân bằng của hàm Bool: cả hai S-box đều có các hàm Bool cân bằng, nghĩa là trong bảng chân lý của mỗi hàm Bool thành phần số lượng giá trị 1 bằng số lượng giá trị 0.

- Bậc đại số: cả S-box của Whirlpool và AES đều có bậc đại số lớn nhất đối với S-hộp 8 bit là 7. Do đó S-hộp trong Whirlpool kháng lại tấn công đại số.

- Đặc trưng lượng sai: độ lượng sai của S-box Whirlpool (Diff(*S*) = 8) không tốt bằng độ lượng sai của S-box AES (Diff(*S*) = 4).

- Đặc trưng tuyến tính: độ tuyến tính của S-hộp của Whirlpool không tốt bằng độ tuyến tính của S-hộp của AES. Độ tuyến tính của S-box AES nhỏ hơn so với S-box Whirpool.

- Mã khối W có tất cả tính phi tuyến của S-Box 8-bit đã được thiết kế để chống lại phương pháp thám mã cổ điển. Độ lượng sai Diff(S) = 8 và độ tuyến tính Lin(S) = 28. Hiện tại, chưa có công bố tìm ra điểm yếu đại số của S-box này.

Ngoài ra, hàm băm Whirlpool không giống với tất cả các đề xuất cho hàm băm dựa trên mã khối, Whirlpool sử dụng một mã khối tựa AES dành cho hàm băm. Lý do của việc này là các nhà thiết kế muốn sử dụng một mã khối với độ an toàn và hiệu quả như AES nhưng cung cấp một độ an toàn tương đương với SHA-512. Kết quả là mã khối W có cấu trúc tương đương và sử dụng các hàm cơ bản như AES nhưng kích thước khối và khóa là 512 bit. Bảng 2 so sánh AES và W.

Bảng 2: So sánh đặc trưng Whirlpool và AES [9]

	W	AES
Kích thước khối	512	128
Kích thước khóa	512	128, 192 hoặc 256
Định hướng ma trận	Đầu vào được ánh xạ hàng	Đầu vào được ánh xạ cột
Số vòng	10	10, 12 hoặc 14
Mở rộng khóa	Hàm vòng W	Thuật toán mở rộng riêng
Đa thức đặc trưng	$x^8+x^4+x^3+x^2+1$ (011D)	$x^8+x^4+x^3+x+1$ (011B)
Nguồn gốc của S-Box	Cấu trúc đệ quy, xây dựng từ 3 mini-box	Nhân nghịch đảo trên trường $GF\left(2^8\right)$ cộng với biến đổi Affine
Nguồn gốc hằng số vòng	Giá trị tiếp theo của S-Box	Số nguyên tố 2^i của $GF\left(2^8\right)$
Tầng khuếch tán	Nhân phải với ma trận MDS 8x8	Nhân trái với ma trận MDS 4x4
Phép hoán vị	Dịch cột	Dịch hàng

**** Nhận xét:***

- Mã khối W trong thuật toán hàm băm Whirlpool làm việc với các khối đầu vào là 512 bit, lớn hơn so với kích thước khối được sử dụng trong thuật toán AES.
- Khóa được sử dụng trong thuật toán Whirlpool là 512 bit, trong khi thuật toán AES có thể làm việc với khóa có độ dài là 128 bit, 192 bit hoặc 256 bit.
- Ma trận trong Whirlpool được làm đầy theo hàng, còn trong AES được làm đầy theo cột.
- Thuật toán Whirlpool và AES đều có có thuật toán mở rộng khóa.
- Đa thức đặc trưng được sử dụng trong 2 thuật toán là gần giống như nhau.
- Có 2 phép biến đổi khác nhau được sử dụng trong 2 thuật toán trên là phép ShiftColumns-MixRows và ShiftRows-Mixcolumns.

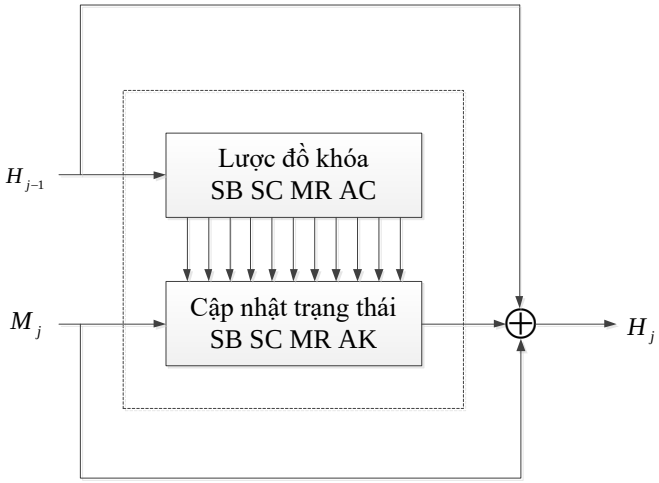
Thuật toán hàm băm Whirlpool có mã khối W được thiết kết theo chiến lược vệt lan rộng tương tự như AES. Chiến lược vệt lan rộng đã được sử dụng trong thiết kế một số nguyên thủy mật mã đối xứng như các hàm băm, mã dòng và mã

khối. Ứng dụng nổi tiếng nhất của nó là thiết kế của AES. Chiến lược vệt lan rộng bao gồm hai phép biến đổi: một phép biến đổi phi tuyến cục bộ (có nghĩa là bất kỳ bit đầu ra nào chỉ phụ thuộc vào một số bit đầu vào giới hạn và các bit đầu ra lân cận phụ thuộc vào các bit đầu vào lân cận) và một phép biến đổi khuếch tán tuyến tính (cung cấp sự khuếch tán cao).

Mã khối W có tính thác đổ từng vòng như AES [7] nên chống lại được tấn công Square. Theo lý thuyết, tấn công Square đạt hiệu quả tốt nhất khi số vòng là 6, mà số vòng của Whirlpool là 10 vòng nên thuật toán là an toàn trước tấn công này.

C. Độ an toàn của hàm nén

Hàm nén của Whirlpool hoạt động ở chế độ Miyaguchi – Preneel.



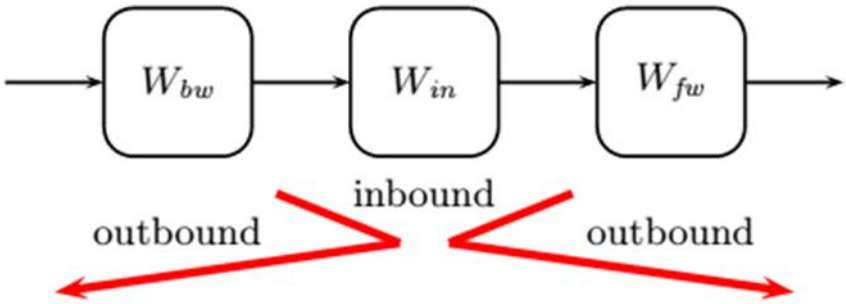
Hình 3: Chế độ Miyaguchi - Preneel

Chế độ Miyaguchi – Preneel là một trong số các chế độ an toàn dùng để xây dựng hàm băm từ mã khối [1]. Đặc biệt, chế độ này có thể an toàn nếu một số thuộc tính lý tưởng được giữ vững cho các mã khối cơ bản. Các kết quả nghiên cứu của Black, Rogaway và Shrimpton [2] đã phân tích các đặc tính an toàn của chế độ Miyaguchi – Preneel.

IV. KẾT QUẢ MỘT SỐ TẤN CÔNG LÊN HÀM BĂM WHIRLPOOL

A. Tấn công Rebound

Tấn công Rebound là một tấn công phân tích lượng sai tìm va chạm dựa trên đầu ra hoặc bắt đầu từ giữa phép toán gần đúng sử dụng trong tấn công lặp. Ý tưởng chính của kỹ thuật này là xây dựng một vệt lượng sai bằng các bậc tự do với xác suất thấp. Tấn công Rebound được tạo từ 3 giai đoạn: 1 giai đoạn inbound và 2 giai đoạn outbound. Giai đoạn inbound tìm các giải pháp cho giai đoạn outbound với độ phức tạp trung bình thấp. Trong giai đoạn outbound, mỗi giải pháp trong giai đoạn inbound được lan truyền theo hai hướng, đồng thời kiểm tra xem có hay không các đặc điểm phù hợp với giai đoạn này. Xác suất của các đặc tính trong giai đoạn outbound càng cao càng tốt.



Hình 4: Hai giai đoạn inbound và outbound

Tấn công Rebound lên hàm băm Whirlpool được giới thiệu bởi nhóm tác giả Mario Lamberger, Florian Mendel, Christian Rechberger, Vincent Rijmen, Martin Schlaffer [4]. Đây là một tấn công lượng sai sử dụng vệt lượng sai với số S-box active tối thiểu theo chiến lược vệt lan rộng. Trọng tâm của tấn công Rebound lên Whirlpool là một vệt lượng sai 4 vòng.

$$1 \xrightarrow{r_1} 8 \xrightarrow{r_2} 64 \xrightarrow{r_3} 8 \xrightarrow{r_4} 1$$

Trong tấn công Rebound, khối bản mã W được chia thành các phần như sau: $W = W_{fw} \circ W_{in} \circ W_{bw}$, sao cho trạng thái hoạt động đầy đủ của vệt lượng sai được bao gồm bởi các giai đoạn inbound.

Chi tiết các bước tấn công Rebound lên hàm băm Whirlpool được trình bày trong [4]. Kết quả được tổng hợp trong Bảng 3.

B. Tấn công tiền ảnh thứ hai

Tấn công tiền ảnh thứ hai lên hằm băm Whirlpool được giới thiệu bởi Yu Sasaki [11]. Đây là một tấn công lên các hàm băm được xây dựng dựa trên mã khối tựa AES.

Tấn công sử dụng cách tiếp cận theo kiểu tấn công gặp nhau ở giữa – MITM (Meet in the Middle). Theo cách tiếp cận này các trạng thái ở các vòng sẽ được chia thành hai khối độc lập *Forward chunk* và *Backward chunk*. Các byte chỉ ảnh hưởng đến một khối được gọi là byte trung lập. Các tính toán từ hai khối này sẽ được nối tại một điểm (trạng thái) thích hợp.

Ý tưởng chính trong tấn công này đó là cố định khóa như một hằng số và sau đó tìm kiếm các bản rõ đầu vào đáp ứng các yêu cầu cụ thể. Các khối trạng thái trong quá trình mã được chia thành hai phần và có chung điểm bắt đầu. Các byte *trung lập* được chọn trước trong trạng thái bắt đầu. Các phép tính toán trên hai khối *Forward* và *Backward* làm thay đổi giá trị, vị trí cũng như số lượng byte *trung lập* trên một trạng thái, từ đây có thể lấy được thông tin từ các byte còn lại. Kết quả được nối lại tại một điểm gọi là *điểm nối*.

Chi tiết về tấn công lên hàm băm Whirlpool được mô tả trong [11]. Kết quả tấn công được tổng hợp trong Bảng 3.

Bảng 3: Tổng hợp kết quả tấn công lên Whirlpool

Mục tiêu tấn công	Số vòng	Độ phức tạp thời gian/ bộ nhớ	Kiểu tấn công	Tác giả
Hàm băm	5.5	$2^{120+s}/2^{64-s}$	Va chạm	Tấn công Rebound - Mario Lamberger, Florian Mendel, Christian Rechberger Vincent Rijmen, Martin Schlaffer [4]
Hàm băm	7.5	$2^{128+s}/2^{64-s}$	Va chạm gần	
Hàm nén	7.5	$2^{184}/2^{8}$	Va chạm	
Hàm nén	9.5	$2^{176}/2^{8}$	Va chạm gần	
Hàm nén	10	$2^{188}/2^{8}$	Dựa trên bộ phân biệt	
Hàm nén	5	$2^{504}/2^{8}$	Tiền ảnh thứ 2	Tấn công MITM-YuSasaki [11]
Hàm băm	5	$2^{504}/2^{8}$	Tiền ảnh thứ 2	

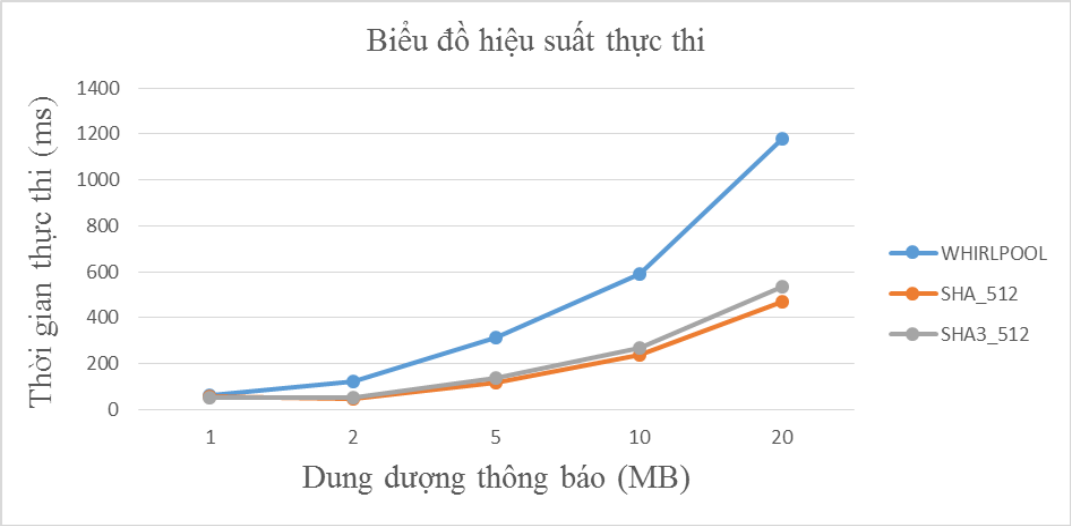
(*s là hệ số điều chỉnh sự tăng/giảm giữa độ phức tạp thời gian và độ phức tạp bộ nhớ* [4])

Bảng kết quả về các tấn công lên hàm băm Whirlpool, cho thấy việc tấn công lên hàm băm là phức tạp, tiêu tốn nhiều tài nguyên cũng như thời gian thực hiện.

V. HIỆU SUẤT THỰC THI WHIRLPOOL TRÊN PHẦN MỀM

Whirlpool có thể được thực thi rất hiệu quả trên phần mềm với các nền tảng khác nhau. Trong [7] đưa ra 3 gợi ý về thực thi thuật toán hàm băm Whirlpool trên phần mềm. Ở đây chúng tôi sử dụng mã nguồn trong đệ trình của tác giả trong dự án NESSIE, tiến hành thực hiện cài đặt thuật toán hàm băm Whirlpool bằng ngôn ngữ C++. Từ đó, đánh giá hiệu suất thực thi của hàm băm Whirlpool qua kết quả so sánh với tốc độ thực thi của các hàm băm SHA_512 và SHA3_512 của NIST.

Các chương trình cài đặt được thực hiện trên cùng một nền tảng phần cứng là máy tính xách tay với vi xử lý Core i3 - 3110M xung nhịp 2.4Ghz, bộ nhớ RAM 4GB:



Hình 5: Hiệu suất thực thi trên phần mềm

Nhận xét: kết quả thực nghiệm cho thấy hiệu suất thực thi của hàm băm Whirlpool là chậm hơn so với hàm băm SHA3_512 và SHA_512.

VI. KẾT LUẬN

Các phân tích về hàm băm Whirlpool cho thấy đây là một hàm băm an toàn. Ngoài các thiết kế an toàn cơ bản đối với một hàm băm dựa trên mã khối với cấu trúc Merkle – Damgard, chế độ Miyaguchi – Preneel, Whirlpool được thiết kế với mã khối W tựa AES. Điều này làm tăng tính ngẫu nhiên, khắc phục được các điểm yếu của các hàm băm dựa trên mã khối. Các phép biến đổi của mã khối W tương tự như AES nên đảm bảo an toàn cho hàm băm trước các tấn công lên mã khối như tấn công lượng sai, tấn công Square.

TÀI LIỆU THAM KHẢO

[1]. B. Preneel, Analysis and design of cryptographic hash functions, Ph.D. thesis,Katholieke Universiteit Leuven, January 1993.

[2]. J. Black, P. Rogaway, T. Shrimpton, Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV, May 31, 2002.

[3]. Joan Daernn, Vincent Rijrnn, The Design of Rijndael AES - The Advanced Encryption Standard, 2002.

[4]. Mario Lamberger, Florian Mendel, Christian Rechberger, Vincent Rijmen, and Martin Schlaffer, Rebound Distringuishers: Results on the Full Whirlpool Comperssion Function, 2009.

[5]. Markku-Juhani O. Saarinen and Billy B. Brumley, Stribobr2: “Whirlbob” Second Round CAESAR Algorithm Tweak Specification, August 28, 2015.

[6]. Oleksandr Kazymyrov, Prototype of Russsian Hash Fucntion “Stribog”, 2012.

[7]. Paulo S.L.M. Barreto and Vincent Rijmen, The Whirlpool Hashing Function, May 24, 2003.

[8]. Rajeev Sobti and G.Geetha, Cryptographic Hash Functions: A Review, March 2012.
[9]. William Stallings, Appendix N Whirlpool, 2010.
[10]. William Stallings, The Whirlpool Secure Hash Function, 2006.
[11]. Yu Sasaki, Meet-in-the-Middle Preimage Attacks on AES Hashing Modes and an Application to Whirlpool, in Proceedings of FSE, LNCS, Vol. 6733, pp. 378- 396, 2011.