Nghiên cứu các cuộc tấn công hệ mật mã nhẹ PRESENT

Mai Mạnh Trừng
Khoa Công nghệ thông tin
Đại học Kinh tế - Kỹ thuật Công nghiệp Hà nội
Email: mmtrung@uneti.edu.vn

Lê Phê Đô
Khoa Công nghệ thông tin
Đại học Công nghệ - Đại học Quốc gia Hà nội
Email: dolp.cntt@gmail.com

Lê Trung Thực
Khoa Công nghệ thông tin
Đại học Công nghệ - Đại học Quốc gia Hà nội
Email: thuclt12a@gmail.com

Trần Văn Mạnh
Khoa Công nghệ thông tin
Đại học Công nghệ - Đại học Quốc gia Hà nội
Email: manhtv@outlook.com

Lê Thị Len
Khoa Công nghệ thông tin
Đại học Công nghệ - Đại học Quốc gia Hà nội
Email: phalebl@gmail.com

Nguyễn Khắc Hưng
Khoa Công nghệ thông tin
Đại học Công nghệ - Đại học Quốc gia Hà nội
Email: nguyenhung.tlh@gmail.com

Abstract— **Bài báo này, giới thiệu hệ mật mã hạng nhẹ PRESENT và một số kết quả nghiên cứu về tấn công mật mã khối hạng nhẹ PRESENT. Trên cơ sở thảo luận và phân tích này nhóm đề xuất ra giải pháp nâng cao bảo mật hệ mật mã hạng nhẹ PRESENT.**

Keywords: Mật mã nhẹ, mã khối, mã dòng, PRESENT Lightweight, attact PRESENT.

I. GIỚI THIỆU

Trong thế giới đầy biến động về an ninh, an toàn mạng. Trong nhiều trường hợp kẻ tấn công thâm nhập vào hệ thống trái phép, cũng như lắp đặt các thiết bị triển khai trong môi trường cần theo dõi. Tức là, một kẻ tấn công có quyền truy cập vật lý và kiểm soát các thiết bị. Do vậy, với những ứng dụng được bảo mật nhạy cảm (quân sự, tài chính hoặc các ứng dụng tự động…) chúng ta cần phải nêu cao chính sách bảo mật.

Với các thiết bị có tài nguyên hạn chế thì các thuật toán mật mã thông thường là quá lớn, quá chậm và quá tốn năng lượng. Các thuật toán mật mã nhẹ khắc phục được những nhược điểm này. Mục tiêu của mật mã nhẹ là làm cho một loạt các ứng dụng cho các thiết bị hiện đại, như các thiết bị đo thông minh, hệ thống an ninh xe, hệ thống giám sát bệnh nhân không dây, hệ thống giao thông thông minh (ITS) và Internet of Things (IoT), …

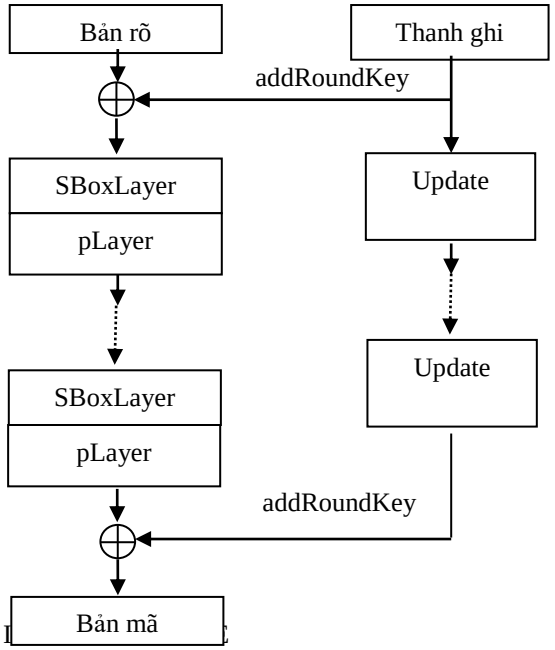
Trong thiết kế của mật mã hạng nhẹ sự cân bằng giữa chi phí, an ninh và hiệu suất phải được đảm bảo. Vì các mã khối, độ dài khóa đưa ra sự thỏa hiệp giữa độ an toàn và giá thành, trong khi đó số vòng đưa ra thỏa hiệp giữa hiệu suất và độ an toàn. Thông thường, ta có thể dễ tối ưu hóa được hai tiêu chí bất kỳ trong ba tiêu chí trên, nhưng việc tối ưu hóa cả ba mục tiêu là việc rất khó. Bên cạnh đó, cài đặt bằng phần cứng có hiệu suất cao cũng cần tính tới giải pháp để tránh các tấn công kênh kề. Điều này thường dẫn tới các yêu cầu về diện tích cao, đồng nghĩa với chi phí cao. Do vậy, mật mã hạng nhẹ phát triển là xu thế tất yếu. Trong mật mã hạng nhẹ chia là 4 nhóm [1, 9] đó là mật mã khối, mật mã dòng, hàm băm và mã xác thực thông báo. Với mật mã khối có khoảng 36 thuật toán[2], kiến trúc của chúng dựa theo kiến trúc mạng thay thế và hoán vị SPN(Substitution Permutation Network) và kiến trúc Feistel. Hệ mật mã hạng nhẹ PRESENT theo kiến trúc SPN. Hệ mật này đã được công nhận là hệ mật tiêu chuẩn do ISO/IEC công nhận.

Trong bài báo này, trên cơ sở thảo luận về thuật toán PRESENT và các cách tấn công hệ mật này. Qua đó, nhóm nghiên cứu đề xuất giải pháp nâng cao hệ mật mã hạng nhẹ PRESENT.

II. TRÌNH BÀY THUẬT TOÁN PRESENT

Hệ mật PRESENT với kích thước khối là 64 bit và độ dài khóa có thể sử dụng là 80 hoặc 128 bit. PRESENT có cấu trúc SPN với số vòng 31, mỗi vòng thực hiện phép cộng XOR để đưa vào khóa vòng, tầng phi tuyến sử dụng một S-hộp 4 bit duy nhất được áp dụng 16 lần song song trong mỗi vòng, tầng tuyến tính sử dụng hoán vị bit đơn giản. Cuối cùng là một phép cộng khóa $K32$ cho việc làm trắng sau (post-whitening). Phép toán giải mã sử dụng các biến đổi ngược của các thành phần mật mã trên. Lược đồ khóa của PRESENT có hai phiên bản khóa là 80 bit và 128 bit có thể được thực hiện dưới dạng tính toán trên đường truyền trên một thanh ghi dịch nhằm tiết kiệm tài nguyên sử dụng[2].

Nhận xét: Từ nghiên cứu tìm hiểu kỹ thuật toán PRESENT, chúng tôi thấy rằng, đối với các thuật toán mã khối hạng nhẹ, mã pháp thường có những đặc điểm sau: 64 bit là lựa chọn chung cho kích thước khối.



Hình 1. Mô tả thuật toán phép mã hóa trong thuật toán PRESENT

Thuật toán mã hóa:

generateRoundKeys()
for i= 1 **to** 31 **do**
addRoundKey(STATE, K_i)
SBoxLayer(STATE)
pLayer(STATE)
end for
addRoundKey(STATE, K_{32})

Trong đó:

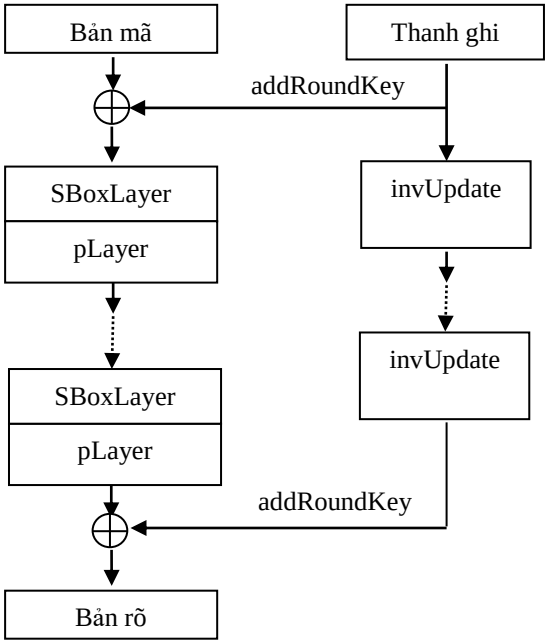
- **addRoundKey(STATE, K_i):** Với vòng khóa $K_i = k^i_{63} \ldots k^i_0$, i chạy từ 1 đến 32 và trạng thái hiện thời là $b_{63} \ldots b_0$, **addRoundKey** gồm toán tử j từ 0<=j<=63. $b_j \longrightarrow b_j \oplus K^i_j$
- **sBoxlayer:** Nhóm sử dụng S-Boxs 4 bit đơn tới 4 bit đơn $f^4_2 \longrightarrow f^4_2$ trong PRESENT. Sử dụng chuỗi biến đổi fourier s:

$$S_b^w(a) = \sum_{x \in F_2^4} (-1)^{(b,S(x))+(a,x)}$$

- **Player:** Khi lựa chọn trộn các lớp, họ tập trung vào hiệu quả phần cứng đòi hỏi một lớp tuyến tính có thể được thực hiện với một số lượng tối thiểu của các yếu tố xử lý.

$$P(i) = \begin{cases} i.16 \mod 63, & i \in \{0, \ldots, 62\} \\ 63, & i = 63, \end{cases}$$

Thuật toán giải mã:

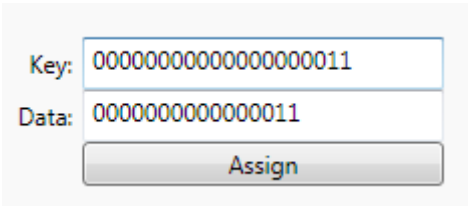


Hình 2. Mô tả thuật toán phép giải mã trong thuật toán PRESENT

Biểu diễn thuật toán giải mã:

generateRoundKeys()
addRoundKey(STATE, K_{32})
for i= 31 **to** 1 **do**
addRoundKey(STATE, K_i)
invPLayer(STATE)
invSBoxLayer(STATE)
addRoundKey(STATE, K_i)
end for

Chương trình minh họa:



Hình 3. Giao diện nhập khóa và dữ liệu

Lưu đồ khóa (PRESENT Key schedule)

```
Input Key:  0000000000000000 0011
Subkey Round 1: >>0000000000000000<<

...after Shift: 0002200000000000 0000
...after S-Box: c002200000000000 0000
Subkey Round 2 (after Salt): >>c002200000000000<< 8000
```

……

```
...after Shift: b0e27a0eb8783ac6 6d59
...after S-Box: 80e27a0eb8783ac6 6d59
Subkey Round 31 (after Salt): >>80e27a0eb8783ac9<< 6d59

...after Shift: 2dab301c4f41d70f 0759
...after S-Box: 6dab301c4f41d70f 0759
Subkey Round 32 (after Salt): >>6dab301c4f41d700<< 8759
```

Hình 4. Lưu đồ khóa PRESENT

Mã hóa PRESENT:

```
Input Key:  0000000000000000 0011
Input Data: 0000000000000011

Round 1
  Subkey: >>0000000000000000<<
Text after...
...Key XOR: 0000000000000011
...  S-Box: cccccccccccccc55
...P-Layer: fffcffff00000003

Round 2
  Subkey: >>c002000000000000<<
Text after...
...Key XOR: 3ffedfff00000003
...  S-Box: b2217222cccccccb
...P-Layer: 80ff08feef019801

          ……..

Round 31
  Subkey: >>80e27a0eb8783ac9<<
Text after...
...Key XOR: f43c6f9c835cf309
...  S-Box: 29b4a2e43b042bce
...P-Layer: 6a471313aecd60c4

Final Round (32):
  Subkey: >>6dab301c4f41d700<<
Text after...
...Key XOR: 07ec230fe18cb7c4
```

Hình 5. Kết quả 32 vòng lặp trao đổi khóa, hộp S-Box, P- Layer.

```
Input:
       Key: 0000000000000000 0011
      Data: 0000000000000011
Output:
    Cipher: 07ec230fe18cb7c4
```

Hình 6. Kết quả của quá trình mã hóa với thuật toán mật mã nhẹ PRESENT

Do các hệ mật này được thiết kế cho nhu cầu có mức độ an toàn trung bình, nên độ dài khóa dao động từ 64-128 bit nhằm đạt độ an toàn 264 đến 280. Hàm vòng được thiết kế đơn giản sao cho không tốn nhiều không gian. Lựa chọn chung đối với tầng xáo trộn (đối với các chiến lược thiết kế) là các S-hộp có kích thước 4 x 4.

Hoán vị các bit dễ dàng đạt được trong phần cứng, do đó, lựa chọn cho tầng tuyến tính là một phép hoán vị bit (ví dụ như hệ mật PRESENT). Các mã pháp thiết kế mới như KLEIN và LED đã đề xuất một dạng khác cho tầng tuyến tính. Nó tương tự như AES, tức là nhận trạng thái với một ma trận phân tách có khoảng cách cực đại MDS. Ưu điểm của việc sử dụng các ma trận này là giúp cho các nhà thiết kế chứng minh độ an toàn chặt chẽ hơn, cùng với tính chất khuếch tán rất tốt.

III. MỘT SỐ PHƯƠNG PHÁP TẤN CÔNG PRESENT

Để tấn công một hệ mật trong mật mã nhẹ có các phương pháp tấn công như tấn công vét cạn, tấn công thống kê bão hòa, tấn công đại số, tấn công khôi phục khóa.

A. Tấn công đại số

Tấn công một thuật toán mật mã bằng cách phân tích đại số[6] thường bao gồm hai bước: đầu tiên, mô hình hóa các phép toán của thuật toán mật mã bằng một hệ đa thức đa biến phi tuyến tính $f_1,\ldots f_s \in P$, trong đó P = K [x1,…., xn] là một vòng đa thức trong n xác định $x_1,\ldots, x_n$ trên một mặt phẳng K. Trong trường hợp mật mã hoá đối xứng, K thường là trường F_{2^r} đặc trưng của 2 với $r \geq 1$. Thứ hai, giải phương trình kết quả của phương trình S được cho bởi:

$$f_1(x_1,\ldots x_n) = 0$$
$$\vdots$$
$$f_S(x_1,\ldots x_n) = 0$$

Cho các biến quan trọng, do đó phá vỡ mật mã. Mặc dù ý tưởng cơ bản nghe đơn giản, vấn đề giải quyết các hệ thống chung của các phương trình đa phi tuyến tính trên các trường phổ biến được biết đến là NP-hard [3]. Tuy nhiên, đây là sự phức tạp chung và cũng có những trường hợp có vấn đề mà giải pháp có thể được phục hồi một cách chính xác. Ví dụ, mật mã Crypto-1 [4] đã bị tấn công thành công bằng cách sử dụng các cuộc tấn công đại số. Hơn nữa, các kỹ thuật đại số hóa ra được khá hiệu quả khi được sử dụng kết hợp với các cuộc tấn công kênh phụ [5] và thường được sử dụng trong bối cảnh thứ hai.

Lưu ý rằng mô hình hoá một thuật toán thông qua đa thức $f_1,\ldots,f_s$ có thể được thực hiện theo những cách khác nhau và các mô tả khác biệt có thể có một tác động đáng kể đến thời gian giải quyết của hệ thống [6]. Trong nhiều trường hợp, hệ thống đa thức có thể được biểu diễn hoàn toàn như là một hàm của các điểm không xác định chính. Tuy nhiên, các đa thức kết quả thường có một mức độ rất lớn và bao gồm một số lượng lớn các đơn thức mà nói chung là không hiệu quả để giải quyết hệ thống. Điều này dẫn đến một hệ thống quá mức, tức là có nhiều phương trình hơn các biến. Hơn nữa, vì hệ đa thức có thể có các giải pháp trong

việc đóng mở đại số $\overline{f_{r^2}}$ của f_{r^2} , một trong những các phương trình ô $x_i^{2^r}$ - x_i = 0, cho i ∈ {1,…,n} tới S đảm bảo rằng tất cả các giải pháp được tìm thấy bởi người giải quyết được trong f_{r^2}.

B . Tấn công khôi phục khóa

Mục đích cuối cùng của kẻ tấn công không chỉ để phân biệt một mật mã từ một sự hoán vị ngẫu nhiên mà còn để khôi phục lại khoá bí mật. Một cuộc tấn công phân biệt có thể được chuyển thành một cuộc tấn công phục hồi quan trọng như sau: Giả sử mã hóa ε của một mật mã khối n-bit khóa con K_j cho $j \in \{0, \dots, r\}$, với K=K_0 ||…..|| K_n và có thể được viết như sau:

$C= \varepsilon(K, M)=f_{r-1}(f_{r-2}(\dots f_1(f_0(M \oplus K_0) \oplus K_1)\dots \oplus K_{r-2} \oplus K_{r-1}) + K_r$

Nói cách khác, ε có thể được mô hình hóa như thể hiện trong hình 6. Hơn nữa giả định rằng kẻ tấn công đã tìm thấy sự khác biệt (α, β) với xác suất P trải dài thứ nhất r-1 nhiều vòng $f_{r-2} \dots . f_0$. Kẻ tấn công khởi tạo các biến đếm $T_0, \dots, T_{n-1}$ như Hình 6. Với mỗi giá trị 0. Chọn ngẫu nhiên bản rõ M và M' = M ⊕ α. Sau đó yêu cầu mã hóa C=$O_{\varepsilon_k}(M)$ và C' = $O_{\varepsilon_k}(M')$. Ta lặp lại tất cả các giá trị có thể $K \epsilon \{0, \dots, 2^{n-1}\}$ của K_r và kiểm tra nếu

$$\beta = f_{r-1}^{-1}(C \oplus k) \oplus f_{r-1}^{-1}(C' \oplus k)$$

Thuật toán khôi phục khóa[6]:

recover_key($(\alpha, \beta), O_{\varepsilon_k}$)

Inputs:

(r-1) - round differentialc (α, β) of probability p, encryption oracle O_{ε_k}

Output:

round key K_r

Algorithm:

($T_0, \dots, T_{n-1}$)← $(0, \dots, 0)$

for I ∈ $\{0, \dots, l-1\}$ ***do***

$M \xleftarrow{s} F_2^n$

C ← $O_{\varepsilon_k}(M)$

C ← $O_{\varepsilon_k}(M \oplus \alpha)$

for k ∈ $\{0, \dots, n-1\}$ ***do***

if $(f_{r-1}^{-1}(C \oplus k) \oplus f_{r-1}^{-1}(C' \oplus k) = \beta)$ **then**

$T_k \leftarrow T_k + 1$

end

end

end

$T_j \leftarrow \max(T_0, \dots, T_{n-1})$

return j

C. Tấn công thống kê bão hòa

a) Nguyên tắc tấn công

Cuộc tấn công của chúng tôi dựa trên một điểm yếu trong lớp khuyếch tán của PRESENT. Xem xét kỹ hơn về hoán vị (permutation) cho thấy, ví dụ: Cho các hộp S-5,6,9 và 10 (tính từ hộp S-box 0 ở bên phải), chỉ có 8 trong số 16 bit đầu vào được chuyển đến các hộp S-box khác. Quan sát Hình 1 minh hoạ này. Lưu ý rằng có tồn tại nhiều Ví dụ sự khuếch tán kém trong hoán vị (nghĩa là với 8 bit trong số 16 còn lại trong cùng 4 hộp S sau khi hoán vị). Do đó, nếu chúng ta xác định 16 bit tại đầu vào của các hộp S-5-6-9-10, thì 8 bit sẽ được biết đến ở đầu vào tương tự cho vòng tiếp theo[7]. Chúng ta có thể lặp lại quá trình này vòng lặp lại và quan sát hành vi không đồng nhất ở đầu ra của các hộp S-5-6-9-10 như Hình 7.

Để khai thác điểm yếu này, chúng tôi đánh giá về mặt lý thuyết sự phân bố của 8 bit theo đường đậm của hình 1 ở đầu ra của lớp S-box, cho một giá trị cố định của cùng 8 bit của bản rõ. Điều này đòi hỏi phải đoán 8 khóa con tham gia vào đường mòn. Một cũng cần phải giả định rằng các bit không nằm trong đường mòn được phân phối đồng nhất. Đây là một giả định hợp lý ngay khi 56 bit còn lại của bản rõ (không bao gồm 8 bit trong đường mòn) được tạo ngẫu nhiên. Sau đó, với sự phân bố đường mòn 8-bit ở đầu vào của một vòng, có thể tính toán phân bố 8-bit ở đầu ra của vòng. Bằng cách lặp lại việc áp dụng thuật toán này, chúng ta có thể tính toán sự phân bố cho một số vòng tùy ý. Đối với mỗi dự đoán chính(key guess), công việc cần thiết để tính toán sự phân bố lý thuyết của đường mòn đích sau r vòng tương đương với r. 216 mã hóa từng phần. Một khi chúng ta đã tính được các phân bố lý thuyết của đường mòn cho mỗi lần dự đoán chính có thể, chúng ta có thể tấn công các thuật toán bằng cách đơn giản so sánh chúng với một sự phân bố thực tế thu được bằng cách mã hóa một số lượng lớn các bản rõ với khoá bí mật. Dự đoán chính giảm thiểu khoảng cách giữa các phân bố lý thuyết và thực tế được chọn là khoá chính xác. Như trong [3], chúng ta có thể xây dựng một danh sách các ứng cử viên khóa được sắp xếp theo khoảng cách giữa lý thuyết và thực hành. Vị trí của khóa bên phải tốt hơn trong danh sách, thì cuộc tấn công càng tốt.

b) Tấn công mở rộng

(Mở rộng 1) Tăng phần cố định trong bản rõ. Người ta có thể dễ dàng đạt được một vòng trong cuộc tấn công bằng cách đơn giản xác định 16 bit bản rõ tương ứng với 4 hộp S-đầu vào tích cực của đường mòn. Bằng cách này, dấu vết 8 bit ở vòng thứ hai cũng được cố định và sự khuếch tán được hoãn lại một vòng. Bằng cách lấy ra 32 bit trong 64 (tương ứng với hộp S-4-5-6-7-8-9-10-11), người ta cũng có thể mở rộng cuộc tấn công bằng 2 vòng. Tuy nhiên, chúng tôi đã giới hạn trong thế hệ của

không quá 2^{32} văn bản. Hạn chế này có thể được giảm nhẹ với sự gia hạn sau.

(Mở rộng 2) Sử dụng nhiều giá trị bản thô. Phân tích tương tự có thể được thực hiện nhiều lần, sử dụng các giá trị khác nhau cho phần 8 bit (hoặc 16 hoặc 32 bit) của các bản rõ và sau đó kết hợp các kết quả (ví dụ: lấy tổng số khoảng cách đồng nhất với các khoảng đo được tương ứng đến các bản thô khác nhau). Điều này cho phép khai thác nhiều văn bản hơn và chuyển sang một ngữ cảnh được biết rõ. Cuộc tấn công kết quả tương tự như nhiều phân tích tuyến tính: mỗi phần cố định của bản rõ có thể được xem như tương tự như một phép xấp xỉ bổ sung trong [7].

(Mở rộng 3) Giải mã một phần của hai vòng thay vì một. Trong trường hợp này, 8 hộp S-box đang hoạt động trong vòng cuối thay vì 4. Vì vậy, chúng ta phải giữ một bảng phân phối 32-bit trong bộ nhớ. Ngoài ra, 38 bit của khoá phải được đoán cho việc giải mã một phần (32 bit cho vòng cuối cùng + 16 bit cho vòng áp chót – 10 bit dư thừa) sử dụng thủ thuật, hay phân biệt r-2 vòng cho khóa chính xác từ r + 2 vòng phân bố cho ứng cử sai. Thời gian phức hợp sẽ là $(32.\ 2^{32}).(16.2^{16})=2^{57}$ dùng lại cho kết quả [8].

IV. GIẢI PHÁP NÂNG CAO CHỐNG TẤN CÔNG BẰNG HỘP S-BOX

Nguồn gốc của cuộc tấn công thống kê bão hòa được đề xuất chống lại PRESENT chủ yếu nằm ở điểm yếu của tầng khuếch tán. Một biện pháp đối phó đơn giản là sửa đổi hoán vị để tránh sự khuếch tán không tốt trong bất kỳ tập con của hộp S-box. Nhưng cuộc tấn công đề xuất liên quan đến tính chất khuếch tán tổng thể của mật mã. Do đó các hộp S-box cũng có một tác động với sự đặc biệt này mà chúng tôi nghiên cứu ngắn gọn trong phần này. Để làm được điều này, chúng ta cần tạo ra 5000 hộp S khác nhau, với bốn điều kiện cố định trong việc tạo ra hộp PRESENT S-box. Theo nhóm nghiên cứu, những khó khăn này đảm bảo rằng PRESENT là chống lại các cuộc tấn công khác biệt và tuyến tính. Để khắc phục tác động của một hộp S-box từ yếu đến mạnh trong phân tích mật mã, chúng ta cần chạy các thí nghiệm mới chống lại một tiện ích trong đó hộp S-box gốc đã được thay thế bằng nhiều S-boxes (tức là tương ứng với tốt nhất và xấu nhất trong số 5000 hộp S-box tạo ra). Hình 8 cho phép đạt được của các cuộc tấn công này cho số vòng tròn khác nhau (mỗi cuộc tấn công sử dụng 2^{30} bản rõ đã chọn). Như mong đợi, cuộc tấn công chống lại phiên bản yếu của mật mã cho kết quả tốt nhất. Phương pháp này nhấn mạnh rằng cuộc tấn công đề xuất không liên quan trực tiếp đến việc phân tích mật mã tuyến tính(nghĩa là có thể có một mật mã không thể tấn công đối với việc phân tích mật mã tuyến tính và riêng biệt, nhưng không chống lại cuộc tấn công thống kê bão hòa được đề xuất).

V. KẾT LUẬN

Trong bài viết này, nhóm nghiên cứu đã trình bày thuật toán và các phương pháp tấn công hệ mật mã nhẹ PRESENT cụ thể với các phương pháp đó là tấn công đại số, tấn công khôi phục khóa, tấn công thống kê bão hòa. Mỗi phương pháp tấn công có những đặc điểm giống nhau và khác nhau. Giống nhau là chúng đều khai thác kẽ hở của hệ mật PRESENT để tấn công. Khác nhau của từng phương pháp phương pháp với phương pháp tấn công đại số là dựa vào hệ đa thức phi tuyến của hệ mật này để tấn công. Còn với phương pháp tấn công khôi phục khóa, dựa vào đặc điểm khóa của PRESENT để tấn công. Với tấn công thống kê bão hòa. Dựa theo đặc điểm điểm yếu trong lớp khuyếch tán của PRESENT.

TÀI LIỆU THAM KHẢO

[1] International standard ISO/IEC 29192, “Information Technology - Security Techniques – Lightweight cryptography”.

[2] Axel York Poschmann, Bochum, “Lightweight cryptography”, February 2009, Faculty of Electrical Engineering and Information Technology Ruhr - University Bochum, Germany.

[3] M. Garey and D. Johnson. Computers and Intractability – A Guide to the Theory
of NP-Completeness. W.H. Freeman and Company, 1979.

[4] N. T. Courtois, K. Nohl, and S. O’Neil. Algebraic Attacks on the Crypto-1 Stream
Cipher in MiFare Classic and Oyster Cards. Cryptology ePrint Archive, Report
2008/166. 2008. http://eprint.iacr.org/2008/166.

[5] A. Bogdanov, I. Kizhvatov, and A. Pyshkin. Algebraic Methods in Side-Channel Collision Attacks and Practical Collision Detection. Progress in Cryptology —
INDOCRYPT 2008. Volume 5365, Lecture Notes in Computer Science. Springer, 2008.

[6] Philipp Jovanovic, Analysis and Design of Symmetric Cryptographic Algorithms, thesis, 2015

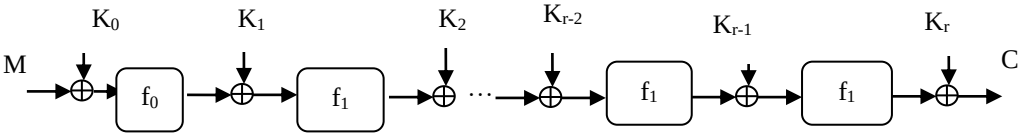
[7] B. Collard, F.-X. Standaert, A Statistical Saturation Attack against the Block Cipher PRESENT, Springer, 2015

[8]. B. Collard, F.-X. Standaert, J.-J. Quisquater, Improving the Time Complexity of Matsui's Linear Cryptanalysis, in the proceedings of The International Conference on Information Security and Cryptology - ICISC 2007, Lecture Notes in Computer Science, vol 4817, pp 77-88, Seoul, Korea, November 2007.

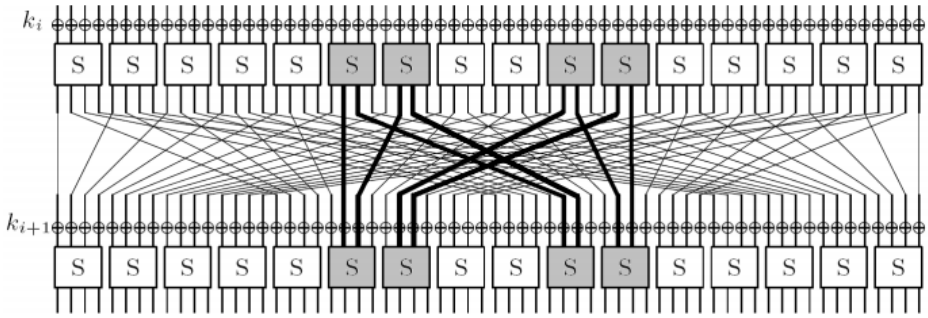
[9]. Orr Dunkelman, “Does Lightweight Cryptography Imply Slightsecurity”, Computer Science Department University of Haifa, 2014

BẢNG 1. S-BOX CỦA PRESENT

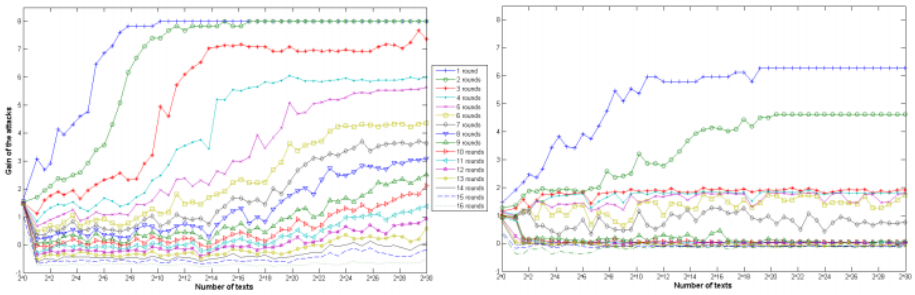
x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S[x]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2



Hình 6. Hàm mã hóa ε_K của mật mã khối



Hình 7. Lớp hoán vị của PRESENT các đường nét đậm nét nhấn mạnh tính chất khuếch tán kém.



Hình 8. So sánh giữa yếu (left) và khỏe (right) S-boxes