MỘT ĐỀ XUẤT MA TRẬN MDS 4×4 AN TOÀN, HIỆU QUẢ CHO TẦNG TUYẾN TÍNH CỦA CÁC MÃ PHÁP DẠNG AES

Nguyễn Ngọc Điệp*

Tóm tắt. *Trong bài báo này, chúng tôi đề xuất và đánh giá tầng tuyến tính có tính chất cài đặt hiệu quả trong phần cứng dựa trên ma trận tựa vòng có thể sử dụng trong thiết kế tầng tuyến tính cho các mã pháp dạng AES, trong khi vẫn đảm bảo được tính chất cài đặt trong phần mềm tương tự như tầng tuyến tính trong AES. Đánh giá số lượng điểm bất động của tầng tuyến tính nhận được và so sánh với tầng tuyến tính trong AES.*

Từ khóa*:* Ma trận MDS, Tầng tuyến tính, AES.

1. GIỚI THIỆU

Từ năm 2000 với việc chuẩn hóa mật mã tiên tiến (AES) [1], một số lượng lớn đáng ngạc nhiên xuất hiện các nguyên thủy mới, trong đó sử dụng các thành phần tương tự như trong AES: LED [2], GOST R 34.11.2012 [3] ... Điều này đa phần là do chiến lược vệt lan rộng được xem xét trong [7] nhằm bảo đảm tính chất khuếch tán tốt cũng như cho phép dễ dàng đưa ra cận an toàn cho khả năng chống lại thám mã lượng sai và tuyến tính cho mã pháp nhận được. Tầng tuyến tính trong AES gồm có 2 biến đổi chính: *MixColumns* và *ShiftRows*. Phép *MixColumns* là phép nhân ma trận với một véc tơ cột còn phép *ShiftRow* là một hoán vị các từ của trạng thái. Để đảm bảo được chiến lược vệt lan rộng, các ma trận phải có tính chất khuếch tán tốt nhất, đó là các ma trận MDS [4].

Tuy nhiên, khi thiết kế ngoài việc phải đảm bảo độ an toàn, cũng cần phải lựa chọn các tham số làm sao tầng tuyến tính nhận được phải dễ dàng cài đặt trong các môi trường khác nhau. Bởi vì tầng tuyến tính là thành phần chậm nhất trong một mã pháp, đây là một vấn đề lớn thu hút các nhà làm mật mã ứng dụng.

Công trình liên quan: Trong [6] nhóm tác giả đề xuất và đánh giá một số ma trận MDS cuộn có dạng Hadamrd hiệu quả trong cài đặt phần cứng. Tuy nhiên, những đánh giá cho tài nguyên cài đặt phần cứng của nhóm tác giả này là chưa chặt, chưa đưa ra chiều sâu thiết kế (số xung nhịp) của sơ đồ phần cứng nhận được. Hơn nữa trận MDS Hadamard cuộn đem lại nhiều điểm bất động cho tầng tuyến tính. Trong [5] đưa ra kết quả số lượng điểm bất động cho tầng tuyến tính của AES bằng 2^{16}, nghiên cứu này cũng trích dẫn một số tấn công tiềm năng liên quan đến điểm bất động của tầng tuyến tính. Các ma trận tựa vòng được Junod và cộng sự nghiên cứu đề xuất trong [8] bởi lợi thế cài đặt của nó vì các ma trận này có nhiều phần tử bằng 1 hơn khi so sánh với ma trận dịch vòng hoặc ma trận Hadamard. Trong [9], tác giả Hoàng Văn Quân đề xuất ma trận dịch vòng hiệu quả sử dụng trong mã pháp dạng AES trên cơ sở khai thác cấu trúc đa thức sinh nguyên thủy của trường hữu hạn nhằm tìm kiếm các phần tử hiệu quả cho ma trận tuyến tính.

Đóng góp của bài báo: Trên cơ sở cách tiếp cận trong [9], bài báo đề xuất các ma trận *tựa vòng* 4x4 an toàn và có tính chất cài đặt hiệu quả hơn. Hơn nữa, bằng lý thuyết bài báo cũng chỉ ra rằng đa thức nguyên thủy trong [9] không phải là duy nhất. Từ đó, cho phép xây dựng được nhiều các ma trận MDS có tính chất mật mã tốt hơn.

Bố cục của bài báo: Phần 1 là giới thiệu tổng quan. Những khái niệm cơ bản cũng như những kiến thức cần thiết được trình bày trong mục 2. Mục 3 là đề xuất một

ma trận MDS tựa vòng an toàn, hiệu quả. Việc phân tích khả năng cài đặt trong phần mềm và đánh giá số lượng điểm bất động của ma trận đề xuất sẽ được trình bày tương ứng trong mục 4 và 5. Mục 6 là kết quả cài đặt mô phỏng phần cứng và cài đặt phần mềm cho ma trận đề xuất. Cuối cùng là phần kết luận ở mục 7 và danh mục tài liệu tham khảo.

2. ĐẶT VẤN ĐỀ

Xem xét một số định nghĩa, khái niệm sau:

Định nghĩa 1. Ma trận dịch vòng là ma trận mà các hàng (hoặc các cột) của nó nhận được từ hàng (cột) trước nó bằng cách dịch vòng đi một phần tử.

Một ma trận dịch vòng $d \times d$ được ký hiệu là $Cir\left(a_0, a_1, ..., a_{d-1}\right)$, $a_i \in \mathbb{F}_{2^n}, 0 \le i \le d-1$.

Định nghĩa 2 [8]. Ma trận tựa vòng kích thước $d \times d$ là ma trận có dạng $\begin{pmatrix} a & 1_{d-1} \\ 1_{d-1} & Cir\left(a_1, a_2, ..., a_{d-1}\right) \end{pmatrix}$,

trong đó, $1_{d-1} = \underbrace{(1,...,1)}_{d-1}$ và $a, a_i \in GF\left(2^n\right), 1 \le i \le d-2$. Ma trận này ký hiệu là $C.like\left(a, Cir\left(1, a_1, ..., a_{d-1}\right)\right)$.

Định nghĩa 3 [4]**.** Ma trận vuông kích thước *d*x*d* là một ma trận MDS khi và chỉ khi tất cả các ma trận vuông con của nó không suy biến.

Như đã biết, số nhánh của các ma trận MDS bằng *d* + 1. Ngoài số nhánh, một tham số nữa liên quan đến độ an toàn của tầng tuyến tính đó là số điểm bất động. Trong [5], tác giả đưa ra khái niệm điểm bất động của tầng tuyến tính $\mathcal{L}: \mathbb{F}_{2^n}^m \to \mathbb{F}_{2^n}^m$, $L\left(X\right) = A \times X^T$, trong đó, A là ma trận không suy biến kích thước *d*x*d*. Theo đó số lượng điểm bất động N_L là số nghiệm của hệ phương trình $\left(A - I\right)X = 0$, trong đó, $I_{d \times d}$ là ma trận đơn vị. Như vậy, có $N_L = 2^{n\left(rank(A) - rank(A-I)\right)} = 2^{n\left(d - rank(A-I)\right)}$, số lượng các điểm bất động phụ thuộc vào hạng của ma trận $A - I$.

Trong [9], xem xét đánh giá độ phức tạp cài đặt phần cứng đối với ma trận vòng $Cir\left(1,1,a,b\right)$. Theo đó, cài đặt này yêu cầu số xung nhịp là 2 + *t*, còn tổng số cổng XOR yêu cầu là $16\left(\#\left(a\right) + \#\left(b\right) + 3n\right)$ với #(*a*) là số lượng cổng XOR yêu cầu cho cài đặt phép nhân của phần tử *a*.

Trong trường hợp của ma trận tựa vòng $C.like(c, Cir\left(1, e, f\right))$, phép biến đổi MixColumns $Y = M \times X$ là:

$$\begin{pmatrix} y_{00} & y_{01} & y_{02} & y_{03} \\ y_{10} & y_{11} & y_{12} & y_{13} \\ y_{20} & y_{21} & y_{22} & y_{23} \\ y_{30} & y_{31} & y_{32} & y_{33} \end{pmatrix} = \begin{pmatrix} c & 1 & 1 & 1 \\ 1 & 1 & e & f \\ 1 & f & 1 & e \\ 1 & e & f & 1 \end{pmatrix} \times \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{pmatrix} \quad (1)$$

trong đó, $y_{ij}, x_{ij}, c, e, f \in \mathbb{F}_{2^n}$, $0 \le i, j \le 3$. Ví dụ để tính y_{00}, cần thực hiện $y_{00} = x_{00} \cdot c \oplus x_{10} \oplus x_{20} \oplus x_{30}$, phép tính này yêu cầu $2 + t_1$ xung nhịp, trong đó t_1 là số xung yêu cầu khi thực hiện với phép nhân với *c*. Đối với các phân tử còn lại trong ma trận *Y*, ví dụ đối với y_{10}, ta cần thực hiện $y_{10} = x_{00} \oplus x_{10} \oplus x_{20} \cdot e \oplus x_{30} \cdot f$. Phép tính này yêu cầu $2 + t_2$ xung nhịp, trong đó, t_2 là số xung nhịp lớn nhất yêu cầu khi thực hiện với phép nhân với *e* hoặc *f*.

Như vậy, số xung nhịp khi cài đặt các ma trận tựa vòng sẽ là $2 + max\{t_1, t_2\}$, Còn số cổng XOR yêu cầu sẽ là $4(\#(c) + 3n) + 12(\#(e) + \#(f) + 3n)$. Bảng 1 là tổng hợp độ phức tạp cài đặt phần cứng đối với một số dạng ma trận.

Bảng 1. *Tham số cài đặt phần cứng của hai dạng ma trận dịch vòng và tựa vòng.*

Dạng ma trận	Số xung nhịp	Số cổng XOR
$Cir(1,1,a,b)$	$2 + t$	$16(\#(a) + \#(b) + 3n)$
$C.like(c, Cir(1,e,f))$	$2 + max\{t_1, t_2\}$	$4(\#(c) + 3n) + 12(\#(e) + \#(f) + 3n)$
$Had(1,a,b,c)$ [6]	$2 + t$	$16(\#(a) + \#(b) + \#(c) + 3n)$

3. MỘT ĐỀ XUẤT MA TRẬN TUYẾN TÍNH TỰA VÒNG HIỆU QUẢ TRONG CÀI ĐẶT

Rõ ràng một điều rằng độ phức tạp trong cài đặt ma trận tuyến tính chính là việc cài đặt phép nhân với các phần tử của ma trận trong trường hữu hạn. Như vậy, việc lựa chọn các hệ số trong mỗi ma trận sẽ quyết định tính chất cài đặt của nó. Trong bất kỳ trường hữu hạn $\mathbb{F}_{2^n}$, với đa thức sinh có bậc *n* thì phép nhân với phần tử 1, g và g^{-1} là dễ dàng cài đặt, ở đây *g* – là phần tử nguyên thủy của trường (thông thường trên trường hữu hạn với đa thức sinh là đa thức nguyên thủy ta thường chọn $g = x$ [1]). Trong cài đặt phép nhân với phần tử 1 là tầm thường, có nghĩa rằng phép nhân này không tốn tài nguyên, còn phép nhân với g và g^{-1} có độ phức tạp như nhau, và chỉ cần không quá 1 clock cycle. Do vậy, tốc độ cài đặt của chúng là nhanh nhất.

Mặt khác ma trận tựa vòng có dạng như trong (1) phải là một ma trận có tính MDS để đảm bảo tính khuếch tán cực đại, có nghĩa là tất cả các ma trận con vuông của nó phải không suy biến. Từ đây, ta có mệnh đề về sự rằng buộc giữa các hệ số trong ma trận *M* như sau:

Mệnh đề 1. *Ma trận tựa vòng* $C.like(c, Cir(1,e,f))$ *trong* (1) *là một ma trận MDS khi và chỉ khi*

$$\begin{cases} c, e, f \ne \{0,1\}, & c \ne e^{-1}, & c \ne f^{-1}, \\ e \ne f^{-1}, & e \ne f^2, & f \ne e^2 \end{cases}. \tag{2}$$

Việc chứng minh mệnh đề này là đơn giản khi xét điều kiện khác không cho định thức của tất cả các ma trận con có thể của nó. Ngoài ràng buộc trên các hệ số *c, e* và *f* phải được lựa chọn làm sao để dễ dàng cài đặt trong cả phần cứng cũng

như phần mềm. Thông thường những phần tử là bội trên trường hữu hạn của phần tử *g,* hoặc nghịch đảo của nó, hoặc biểu thức "đơn giản" từ hai phần tử này, ví dụ như $g \oplus 1$, hoặc $g^{-1} \oplus 1$,… được ưu tiên lựa chọn, ví dụ như hệ số bằng 3 (tức là bằng $g \oplus 1$, với $g = 2$) như trong AES [1], hệ số bằng 4 (bằng g^2, với $g = 2$) như trong LED [2],… Tuy nhiên, chúng ta yêu cầu làm sao để cho những hệ số này khi thực hiện phép nhân cũng chỉ yêu cầu 1 clock cycle. Khi khai thác biểu thức của phép nhân với những phần tử dạng này trong mối *quan hệ với đa thức sinh của trường* có mệnh đề sau:

Mệnh đề 2. *Cho trường hữu hạn* $\mathbb{F}_{2^n}$, *n chẵn, với đa thức sinh là đa thức nguyên thủy* $f(x) = x^n \oplus \bigoplus_{i=1}^{n-1} \lambda_i x^i \oplus 1$, *phần tử nguyên thủy của trường được chọn là g* = 2. *Phép nhân với* g^2 *chỉ yêu cầu 1 xung nhịp khi và chỉ khi:*

$$\begin{cases} \lambda_7 = 0 \\ \lambda_i \lambda_{i-1} = 0, 2 \le i \le n-1 \end{cases} \tag{3}$$

Trong [9], tác giả có đưa ra phân tích về vấn đề này ở đây chúng tôi chỉ tổng hợp lại và viết dưới dạng mệnh đề cho rõ ràng hơn. Phần chứng minh của mệnh đề có thể tham khảo giải thích trong [9]. Tuy nhiên, trong [9] tác giả kết luận đa thức $f(x) = x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$ là đa thức nguyên thủy duy nhất cho phép xây dưng các ma trận dịch vòng tương tự như trong [9] là chưa đủ. Mệnh đề 3 sau đây sẽ chứng minh cho vấn đề này. Đây chính là mở rộng của bài báo so với kết quả trong [9]. Từ đó cho phép tìm được một tập nhiều hơn các ma trận có tính chất cài đặt tốt.

Mệnh đề 3. *Cho trường hữu hạn* $\mathbb{F}_{2^n}$, *n chẵn, với đa thức sinh là đa thức nguyên thủy* $h(x)$, *phần tử nguyên thủy của trường được chọn là g = 2. Phép nhân với* g^{-2} *chỉ yêu cầu 1 xung nhịp khi và chỉ khi:*

$$\begin{cases} \lambda_1 = 0 \\ \lambda_i \lambda_{i-1} = 0, 3 \le i \le n-1 \end{cases} \tag{4}$$

trong đó, $h(x) = x^n \oplus \bigoplus_{i=1}^{n-1} \lambda_{n-i} x^{n-i} \oplus 1$ *là đa thức liên hợp của đa thưc nguyên thủy* $f(x) = x^n \oplus \bigoplus_{i=1}^{n-1} \lambda_i x^i \oplus 1$.

Thấy rằng, nếu đa thức $f(x) = x^n \oplus \bigoplus_{i=1}^{n-1} \lambda_i x^i \oplus 1$ là nguyên thủy, khi đó đa thức liên hợp của nó $h(x) = x^n \oplus \bigoplus_{i=1}^{n-1} \lambda_{n-i} x^{n-i} \oplus 1$ cũng là nguyên thủy [4]. Từ đây, ta có thể thấy các điều kiện (3) và (4) là ngược của nhau, có nghĩa rằng với mỗi đa thức

thỏa mãn điều kiện trong mệnh dề 1, ta luôn tìm được 1 đa thức tương ứng thỏa mãn điều kiện trong mệnh đề 2.

Như đã phân tích ở mục trước, chúng tôi chỉ quan tâm đến các ma trận 4x4 trên trường $\mathbb{F}_{2^8}$, do vậy, chỉ quan tâm đến các đa thức bậc 8. Với giá trị $n = 8$ chỉ có duy nhất một đa thức thỏa mãn yêu cầu trong mệnh đề 1, đó là đa thức $f(x) = x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$. Đây chính là kết quả được đưa ra trong [9]. Từ đây cũng chỉ có 1 đa thức $h(x) = x^8 \oplus x^7 \oplus x^5 \oplus x^3 \oplus 1$ là thỏa mãn điều kiện trong mệnh đề 3. Hai đa thức này là liên hợp của nhau. Chúng cũng chính là 2 trong tổng số $\varphi(2^8 - 1)/8 = 16$ đa thức nguyên thủy có thể có bậc bằng 8, trong đó φ là hàm Euler [1]. Do vậy, chúng có thể được sử dụng trong vai trò đa thức sinh của trường $\mathbb{F}_{2^8}$.

Khi xây dựng trường $\mathbb{F}_{2^8}$ với $f(x) = x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$ và $g = 2$. Ta chọn các hệ số trong ma trận tựa vòng (1) như sau: $c = f = g^{-1}$ và $e = g^2$. Dễ dàng kiểm tra được rằng các hệ số này thỏa mãn điều kiện (2). Do vậy, ma trận

$$C.like_1\left(g^{-1}, Cir\left(1, g^2, g^{-1}\right)\right) = C.like_1\left(149, Cir\left(1, 4, 149\right)\right) \quad (5)$$

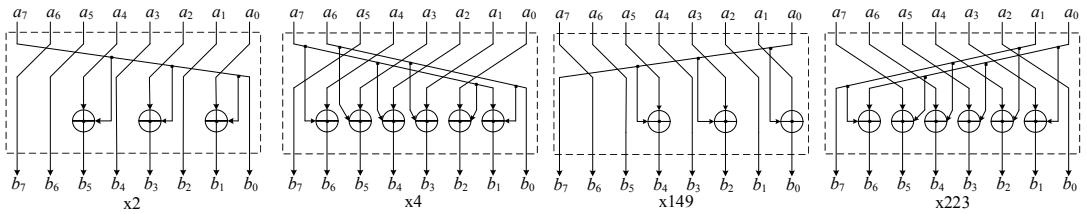
là một ma trận MDS.

Tương tự, khi xây dựng trường với đa thức sinh là $h(x) = x^8 \oplus x^7 \oplus x^5 \oplus x^3 \oplus 1$, phần tử nguyên thủy là $g = 2$. Khi chọn các hệ số $c = f = g$, còn $e = g^{-2}$, ta có thể xây dựng ma trận MDS như sau:

$$C.like_2\left(g, Cir\left(1, g^{-2}, g\right)\right) = C.like_2\left(2, Cir\left(1, 223, 2\right)\right). \quad (6)$$

Khi thực hiện cài đặt theo quan điểm phân tích ở mục trước, hai ma trận (5) hoặc (6) này chỉ yêu cầu $2 + max\{t_1, t_2\} = 2 + 1 = 3$ xung nhịp. Ví dụ hình 1 minh họa dưới đây cho phép nhân với các phần tử trong ma trận (5) và (6).

Từ đây ta có thể tính được số cổng XOR cần thiết để thực hiện biến đổi MixColumns khi sử dụng ma trận (5) là $4(\#(c) + 3n) + 12(\#(e) + \#(f) + 3n) = 4(3 + 3 \times 8) + 12(6 + 3 + 3 \times 8) = 504$, và số xung nhịp là 3. Do tính liên hợp của hai hai đa thức $f(x)$ và $h(x)$ cho nên đây cũng chính là độ phức tạp khi thực hiện cài đặt phần cứng cho ma trận (6). Dưới đây là so sánh tính chất cài đặt cho ma trận trong bài báo này và những đề xuất trước đó.



Hình 1. *Minh họa phép nhân với 2, 4, 149 và 223 trên* $\mathbb{F}_{2^8}$ *với* $f(x) = x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$ *và* $h(x) = x^8 \oplus x^7 \oplus x^5 \oplus x^3 \oplus 1$.

Bảng 2. *Độ phức tạp cài đặt của một số tầng tuyến tính trên cơ sở các ma trận tuyến tính 4x4.*

Dạng ma trận	Số cổng XOR	Số xung nhịp	Đa thức sinh của trường $\mathbb{F}_{2^8}$	Ghi chú
Cir(1,1,2,3)	576	4	$x^8 \oplus x^4 \oplus x^3 \oplus x \oplus 1$	[1]
Cir(4, 149, 1, 1)	520	3	$x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$	[9]
Had(1,2,4,195)	560	4	$x^8 \oplus x^7 \oplus x^6 \oplus x \oplus 1$	[6]
C.like$_1$(195,Cir(1,4,195))	504	3	$x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$	Đề xuất
C.like$_2$(2,Cir(1,223,2))	504	3	$x^8 \oplus x^7 \oplus x^5 \oplus x^3 \oplus 1$	Đề xuất

Bảng phân tích thấy rằng ma trận chúng tôi đề xuất là hiệu quả hơn so với những đề xuất trước đó. Tiếp theo chúng tôi sẽ phân tích cài đặt theo quan điểm phần mềm của ma trận đề xuất so với những ma trận trước đó.

4. PHÂN TÍCH CÀI ĐẶT THEO QUAN ĐIỂM PHẦN MỀM

Tài liệu mô tả gốc của chuẩn AES trình bày phương pháp cài đặt trên môi trường 32 bit sử dụng kỹ thuật tra bảng [1]. Theo đó, phương pháp này là có thể áp dụng với ma trận MixColumns tuyến tính 4x4 bất kỳ. Bộ nhớ yêu cầu khi lưu toàn bộ 4 bảng tra cho cài đặt biến đổi MixColumns này là 4096 Bytes (4KB). Trong hệ thống mạng với năng lực tính toán của các máy tính hiện nay thì bộ nhớ 4 KB là hoàn toàn không đáng kể. Tuy nhiên, trên quan điểm người thiết kế chúng tôi còn mong muốn hướng đến những môi trường hạng nhẹ, nơi mà quan tâm nhiều hơn đối với yêu cầu về tài nguyên cài đặt. Do vậy, để có thể phân tích cài đặt một cách tổng thể theo quan điểm phần mềm, trong mục này chúng tôi chỉ tập trung phân tích độ phức tạp, hay nói chính xác hơn là số phép toán cần thiết khi cài đặt mà không sử dụng kỹ thuật tra bảng (trong nhiều tài liệu gọi đây là phương pháp cài đặt bit-slice [1]). Bản chất kỹ thuật này là thực hiện biến đổi tuyến tính khi xem xét phép nhân trực tiếp với các hệ số của ma trận tuyến tính sử dụng trong biến đổi MixColumns. Ta chỉ quan tâm đến biến đổi MixColumns bởi phép ShiftRows thực tế cài đặt là không tốn tài nguyên.

Không giống như trong mục 3 khi mà phép XOR được hiểu là cộng 2 bit theo modulo 2 với nhau, ở đây phép XOR được hiểu là cộng từng bit theo modulo 2 của hai số 8 bit.

Đối với hai ma trận đề xuất, ví dụ với *C.like*$_1$(195, *Cir*(1,4,195)) xét biểu thức:

$$\begin{pmatrix} y_{00} & y_{01} & y_{02} & y_{03} \\ y_{10} & y_{11} & y_{12} & y_{13} \\ y_{20} & y_{21} & y_{22} & y_{23} \\ y_{30} & y_{31} & y_{32} & y_{33} \end{pmatrix} = \begin{pmatrix} 149 & 1 & 1 & 1 \\ 1 & 1 & 4 & 149 \\ 1 & 149 & 1 & 4 \\ 1 & 4 & 149 & 1 \end{pmatrix} \times \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{pmatrix} \quad (7)$$

Từ biểu thức của ma trận ta thấy 4 giá trị y_{00}, y_{01}, y_{02} và y_{03} được tính giống nhau khi nhân lần lượt hàng thứ nhất của ma trận tuyến tính với lần lượt các cột của ma trận dữ liệu. Ví dụ với y_{00} có:

$$y_{00} = 149 \cdot x_{00} \oplus x_{10} \oplus x_{20} \oplus x_{30} = 2^{-1} \cdot x_{00} \oplus x_{10} \oplus x_{20} \oplus x_{30}.$$

Như vậy, để tính được toàn bộ y_{00}, y_{01}, y_{02} và y_{03} cần 4x1 = 4 phép Xtime và 4x3 = 12 phép XOR. Mặt khác ta thấy 12 giá trị y_{ij}, với $1 \le i \le 3, 0 \le j \le 3$ còn lại được tính giống nhau bởi 3 hàng còn lại trong ma trận tuyến tính có các số hạng giống nhau. Ví dụ với y_{10} có:

$$y_{10} = x_{00} \oplus x_{10} \oplus 4 \cdot x_{20} \oplus 149 \cdot x_{30} = x_{00} \oplus x_{10} \oplus 2 \cdot 2 \cdot x_{20} \oplus 2^{-1} \cdot x_{30}.$$

Biểu thức này yêu cầu 3 phép Xtime và 3 phép XOR. Như vậy, việc tính toàn bộ 12 giá trị y_{ij}, với $1 \le i \le 3, 0 \le j \le 3$ sẽ yêu cầu 12x3 = 36 phép Xtime và 12x3 = 36 phép XOR.

Từ đây, ta có độ phức tạp để thực hiện toàn bộ biến đổi MixColumns là 36 + 4 = 40 phép Xtime và 36 + 12 = 48 phép XOR.

Tương tự, độ phức tạp cho ma trận $C.like_2(2,Cir(1,223,2))$ cũng là 40 phép Xtime và 48 phép XOR. Bảng 3 dưới đây là so sánh khả năng cài đặt trên môi trường 8 bit của các ma trận chúng tôi đề xuất, ma trận trong AES, trong [6] và trong [9].

Bảng 3. *So sánh cài đặt kiểu bit-slice các ma trận MDS 4x4.*

Ma trận	XOR	Xtime	Ghi chú
Cir(2, 3, 1, 1)	60	16	[1]
Cir(4, 149, 1, 1)	48	48	[9]
Had(1, 2, 4, 145)	80	112	[6]
$C.like_1(149,Cir(1,4,149))$	48	40	Đề xuất
$C.like_2(2,Cir(1,223,2))$	48	40	Đề xuất

Qua bảng phân tích ta thấy ma trận MDS trong AES yêu cầu số phép toán ít nhất, còn ma trận MDS Hadamard trong [6] là không hiệu quả khi cài đặt theo kiểu bit-slice và không thích hợp khi sử dụng trong cài đặt trên các môi trường hạn chế khi so sánh với ma trận MixColumns trong AES và các ma trận MDS tựa vòng mà chúng tôi đề xuất. Cần phải nói thêm rằng ma trận sử dụng trong biến đổi MixColumns AES là ma trận tối ưu nhất trong tất cả các ma trận MDS 4x4 trên $\mathbb{F}_{2^8}$ khi cài đặt theo kiểu bit-slice, tuy nhiên khi cài đặt phần cứng như phân tích ở mục trước thì ma trận chúng tôi đề xuất lại là ma trận tối ưu nhất. Ma trận tựa vòng chúng tôi đề xuất có độ phức tạp cài đặt theo kiểu bit-slice tốt hơn nghiên cứu trong [9] của tác giả Hoàng Văn Quân.

Trên thực tế khi lựa chọn ma trận ta phải xét tất cả các tính chất có thể của nó. Do vậy, trong phần tiếp theo chúng tôi sẽ phân tích thêm một tính chất nữa, đó là số lượng điểm bất động của tầng tuyến tính.

5. ĐIỂM BẤT ĐỘNG CỦA TẦNG TUYẾN TÍNH

Như đã phân tích ở mục trước số nhánh là tham số quan trọng nhất của tầng tuyến tính, khi xây dựng tầng tuyến tính theo kiểu AES mà sử dụng ma trận MDS 4x4 trong biến đổi MixColumns ta sẽ đảm bảo được tính chất theo chiến lược vệt lan rộng [7]. Khái niệm số lượng điểm bất động của tầng tuyến tính đưa ra trong

[5] sẽ là một tham số quan trọng khi lựa chọn tầng tuyến tính, nó ảnh hưởng đến độ an toàn và được xem như là một tham số bổ sung cho khái niệm số nhánh của tầng tuyến tính.

Tầng tuyến tính trong AES gồm 2 biến đổi: ShiftRows và MixColumns. Khi kết hợp ta có thể biểu diễn bởi phép nhân một ma trận 16x16 [5].

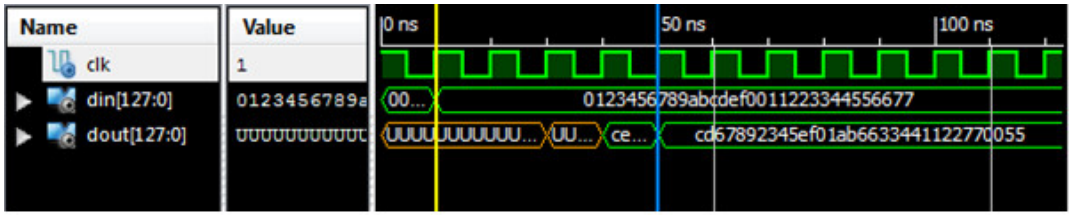
Gọi ma trận tuyến tính 16x16 này là A. Đối với ma trận này có $rank(A)=16$ và $rank(A-I)=14$. Theo đó, số lượng điểm bất động của tầng biến đổi tuyến tính trong AES là $N_{Cir\_AES}=2^{n(rank(A)-rank(A-I))}=2^{8(16-14)}=2^{16}$. Thực hiện tương tự đối với ma trận Had(1, 2, 4, 145) [6], ma trận Cir(4, 149, 1, 1) [9] và hai ma trận chúng tôi đề xuất là C.like$_1$(149,Cir(1,4,149)) và C.like$_2$(2,Cir(1,223,2)) chúng tôi nhận được:

$$N_{Had(1,2,4,145)}=1,\ N_{Cir(4,149,1,1)}=1,\ N_{C.like_1(149,Cir(1,4,149))}=1 \text{ và } N_{C.like_1(2,Cir(1,223,2))}=1.$$

Như vậy, nếu sử dụng hai ma trận này để thay thế ma trận trong biến đổi MixColumns trong AES thì sẽ nhận được tầng tuyến tính mà chỉ có một điểm bất động, đây chính là điểm véc tơ 0 tầm thường.

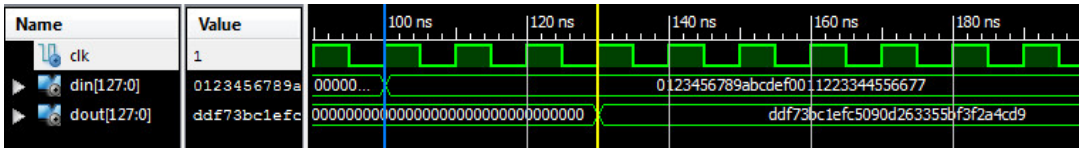
6. KẾT QUẢ THỰC NGHIỆM

Chúng tôi thực hiện các thiết kế các khối chức năng của biến đổi MixColumns khi sử dụng ma trận đề xuất và một số ma trận đề xuất trước đó. Chương trình được viết trên ngôn ngữ VHDL cho Virtex6 XC6VLX240T–2FF1156. Công cụ thiết kế là Xilinx ISE phiên bản 14.3. Ví dụ minh họa mô phỏng Isim cho ma trận trong Cir(2, 3, 1, 1) sử dụng trong AES như trong hình 2 ta thấy rằng cài đặt này yêu cầu 4 xung nhịp.



Hình 2. *Kết quả mô phỏng đối với ma trận trong AES Cir(2, 3, 1, 1).*

Đối với ma trận đề xuất C.like$_1$(149,Cir(1,4,149)). Hình 3 là mô phỏng Isim đối với ma trận đề xuất này. Trong cài đặt này ta thấy sau 3 clock cycle ta sẽ nhận được kết quả đầu ra tính từ thời điểm xuất hiện tín hiệu đầu vào.



Hình 3. *Kết quả mô phỏng với ma trận đề xuất C.like$_1$(149,Cir(1,4,149)).*

Thực hiện tương tự với những ma trận khác trong bảng 3 chúng tôi có thống kê tài nguyên và tốc độ trong bảng 4.

Bảng 4. *Tham số cài đặt phần cứng trên Virtex6 XC6VLX240T–2FF1156.*

Ma trận	Slice Registers	LUT-FF pairs	LUTs	Clock cycles	Maximum Frequency (MHz)	Speed (Mbps)
Cir(2, 3, 1, 1) [1]	688	530	544	4	1225.415	39213
Cir(4, 149, 1, 1) [9]	656	520	528	3	1225.415	52284
Had(1, 2, 4, 145) [6]	702	546	560	4	1225.415	39213
C.like$_1$(149,Cir(1,4,149))	632	493	504	3	1225.415	52284
C.like$_2$(2,Cir(1,223,2))	632	493	504	3	1225.415	52284

Trong bảng này, ta thấy ma trận chúng tôi đề xuất có tham số cài đặt phần cứng tốt nhất, tốc độ xử lý dữ liệu vượt trội so với những ma trận còn lại và tương đương với ma trận trong [9].

Bảng 5. *Kết quả thực nghiệm tốc độ mã hóa của AES sử dụng ma trận đề xuất.*

Phiên bản AES	Quá trình	Tốc độ	
		MB/s	cpb
AES-128	Mã hóa	204	12
	Giải mã	221	11
AES-192	Mã hóa	189	13
	Giải mã	185	13
AES-256	Mã hóa	165	15
	Giải mã	162	15

Chúng tôi cũng tiến hành cài đặt phần mềm cho thuật toán AES, trong đó, sử dụng ma trận C.like$_1$ để thay thế cho ma trận tuyến tính của AES. Cách tiếp cận ở đây là sử dụng bảng tra trên môi trường với thanh ghi 32 bit. Chương trình được viết trên ngôn ngữ C chuẩn, biên dịch sử dụng Visual Studio v10 trên máy Intel(R) Core(TM) i5-6200U CPU 2.4GHz, Ram 4.00GB, Windows 10. Trong chương trình không sử dụng bất kỳ một lệch assembler hay các hộ trợ SSE nào. Tốc độ được đo bằng MegaBytes/s (MB/s).

Kết quả cài đặt trong bảng 5 là tương đương với cài đặt của AES [1].

7. KẾT LUẬN

Trong bài báo này, chúng tôi đề xuất hai ma trận tuyến tính có tính chất mật mã tốt có thể thay thế ma trận trong biến đổi MixColumns trong AES. Ma trận này là một ma trận MDS tựa vòng trên trường hữu hạn. Khi sử dụng trong biến đổi MixColumns của AES sẽ nhận được tầng tuyến tính đảm bảo được số nhánh theo chiến lược vệt lan rộng mà không có điểm bất động như trường hợp ma trận gốc của AES. Ma trận đề xuất có cài đặt kiểu bit-slice có thể so sánh với ma trận gốc trong AES, và tốt hơn nhiều so với ma trận đề xuất trong [6] và [9]. Theo quan điểm cài đặt phần cứng ma trận của chúng tôi không những yêu cầu tài nguyên ít hơn mà còn có tốc độ xử lý dữ liệu cao hơn cả ma trận gốc trong AES, ma trận đề xuất trong [6] và ma trận trong [9].

Bằng lý thuyết tìm được nhiều đa thức sinh hơn thỏa mãn điều kiện trong mệnh đề 2 và 3. Từ đó, cho phép tìm được nhiều hơn các ma trận MDS có tính chất cài đặt tốt.

Về mặt an toàn, ma trận của chúng tôi khi sử dụng trong tầng tuyến tính của AES không có điểm bất động, trong khi tầng tuyến tính của AES có 2^{16} điểm bất động.

TÀI LIỆU THAM KHẢO

[1]. Зензин, O. and M. Иванов, "*Стардарт криптографической защиты-AES. Конечные поля"*. 2002: КУДРИЦ-ОБРАЗ М.

[2]. Guo, J., et al., "*The LED block cipher"*, in Cryptographic Hardware and Embedded Systems–CHES 2011. 2011, Springer. p. 326-341.

[3]. *ГОСТ Р 34.11-2012: "Криптографическая защита информации"*. Функция хиширования. 2012: p. 25.

[4]. Мак-Вильямс, Ф.Д., "*Теория кодов, исправляющих ошибки"*. 1979.

[5]. Z'aba, M.R., "*Analysis of linear relationships in block ciphers"*. Luận án Tiến sĩ của Queensland University of Technology, 2010.

[6]. Sim, S.M., et al. "*Lightweight MDS Involution Matrices"*. in *FSE*. 2015.

[7]. Daemen, J. and V. Rijmen, "*The wide trail design strategy"*, in Cryptography and Coding. 2001, Springer. p. 222-238.

[8]. P. Junod and S. Vaudenay, "*Perfect diffusion primitives for block cipher – Building efficient MDS matrices"*. In Selected Areas in Cryptology (SAC 2004), LNCS 3357, pp. 84-99, Springer-Verlag, 2004.

[9]. Hoàng Văn Quân. "*Đề xuất ma trận MDS đạt hiệu năng cao khi cài đặt trên phần cứng cho các mã pháp dạng AES"*. Tạp chí Nghiên cứu KH&CN quân sự, Số 40, 12 – 2015.

ABSTRACT

A PROPOSITION FOR THE SECURE AND EFFECTIVE 4×4 MDS MATRICES IN THE LINEAR LAYER OF AES-LIKE BLOCK CIPHERS

In this paper, we proposed and evaluated the linear layer which have effective implemented property on the hardware based on circulant matrices that can be used in designing a linear layer for AES-like block ciphers, while still guarantee implemented property on software as the liner layer in AES. We have evaluated the number of fixed points in the obtained liner layer and compared with the linear layer in AES.

Key words: MDS matrix, The linear layer, AES.

Nhận bài ngày 30 tháng 9 năm 2016
Hoàn thiện ngày 19 tháng 10 năm 2016
Chấp nhận đăng ngày 14 tháng 12 năm 2016

Địa chỉ: Viện Khoa học – Công nghệ mật mã, Ban Cơ yếu Chính phủ;
* Email: diepnngoc@yahoo.com.