**Họ các thuật toán mã khối mới cho các mạng truyền dữ liệu thời gian thực**

Bài báo này trình bày về một số kết quả xây dựng họ thuật toán mật mã khối mới dựa trên lớp nguyên thủy mật mã **F**2/1. Đây là lớp phần tử điều khiển được (Controlled Elements - CE) đã được nghiên cứu và chứng minh là phù hợp để phát triển các thuật toán mật mã khối có tốc độ và khả năng tích hợp cao trên các nền tảng phần cứng được tích hợp với quy mô lớn (very large scale integrated – VLSI) (như FPGA và ASIC). Điều đó đảm bảo họ thuật toán này sẽ hoạt động hiệu quả và phù hợp cho truyền dữ liệu thời gian thực trên các mạng không dây.

**Abstract-**In this report, a number of results of constructing the algorithm of new block ciphers will be demonstrated based on the primitive class with code F2/1.This is the controllable elements (CEs) which have been studied and proved to be suitable for developing the algorithm of block ciphers with highly integrated speed and outcome on the hardware platform such as VLSI (FPGA and ASIC). Therefore, this algorithm will be sure to operate effectively and appropriate for transmitting real-time data on wireless network.

1. **Mở đầu**

Nhu cầu sử dụng các dịch vụ qua mạng không dây ngày càng lớn, trong đó có rất nhiều dịch vụ yêu cầu truyền dữ liệu theo thời gian thực như: chat voice, video call, …[1]. Tuy nhiên, trên thực tế lại hình thành một mâu thuẫn giữa nhu cầu phát triển các ứng dụng với các nguy cơ mất an toàn thông tin.

Để chống lại các nguy cơ này thì việc sử dụng các kỹ thuật mật mã là giải pháp hiệu quả nhất [2]. Hiện tại, các lớp bảo mật của công nghệ không dây vẫn sử dụng các thuật toán mật mã được đánh giá là chưa hoàn toàn phù hợp với giải pháp phần cứng, đặc biệt là cho thiết bị không dây. Hầu hết, các thuật toán mật mã sử dụng nhiều các biến đổi số học và đại số phức tạp, nhưng không thích hợp cho các giải pháp phần cứng [2]. Đó là nguyên nhân tại sao khi tích hợp các giải pháp mật mã thường chiếm nhiều tài nguyên của hệ thống. Điều bất cập nhất là phải phát triển đồng thời các thuật toán mật mã theo xu hướng đảm bảo tốc độ và có hiệu quả tích hợp cao, phù hợp triển khai trên phần cứng, đáp ứng nhu cầu thay khóa phiên thường xuyên và đảm bảo độ an toàn nhằm đáp ứng yêu cầu ngày càng cao của các ứng dụng.

Với ưu điểm của các phần tử điều khiển được CE, dựa trên lớp nguyên thủy mật mã F2/1 [2], chúng cho phép xây dựng các thuật toán mật mã có độ an toàn và hiệu quả tích hợp cao hơn so với các thuật toán dựa trên DDP (Data Dependent Permutation) [2, 3, 5]. Thông thường, các thuật toán này đảm bảo sự an toàn chống lại thám mã tuyến tính (dựa trên các kết quả đã công bố). Tuy nhiên, chúng có thể bị tấn công (về mặt lý thuyết) trước các kiểu thám mã lượng sai trên lược đồ khóa [5]. Hơn nữa, tất cả các thuật toán mật mã dựa trên DDO (Data Dependent Operation) đã công bố [2, 3], đều hướng tới thực hiện trên môi trường có tài nguyên hạn chế và các mạng tốc độ cao. Vì vậy, tất cả các thuật toán này đều sử dụng lược đồ khóa đơn giản, dẫn đến, hầu hết các thuật toán đều có thể bị tấn công (lý thuyết) dựa trên cơ sở thám mã lượng sai với các khóa có liên kết (Related Key Differential Attacks - RKDA) [6, 7].

Bài báo này sẽ trình bày kết quả xây dựng họ thuật toán mật mã khối tốc độ cao TMN64 và TMN128 dựa trên lớp CE F2/1. Trong đó có chứng minh, đánh giá về độ an toàn và hiệu quả của các thuật toán đề xuất sử dụng trong thực tiễn để phát triển các ứng dụng truyền thông trên các mạng không dây.

Họ thuật toán được phát triển với cấu trúc tương tự nhau, nhưng sử dụng ưu nhược điểm về độ an toàn và hiệu năng thực hiện trên kích thước khóa mật và dữ liệu khác nhau. Với việc lựa chọn kích thước khối dữ liệu được xử lý khác nhau (64 hoặc 128 bit) và độ dài khóa bí mật khác nhau (128 hoặc 256 bit), cho phép chúng có thể ứng dụng trong các mạng có tài nguyên hạn chế (thường sử dụng các thuật toán mật mã khối hạng nhẹ) đến các môi trường mạng yêu cầu đảm bảo độ bảo mật cao.

Bố cục của bài báo như sau: Mục 2 trình bày tổng quan về các lớp CE F2/1; Mục 3 trình bày kết quả xây dựng họ thuật toán TMN; Mục 4 và 5, trình bày về các kết quả đánh giá độ an toàn và hiệu quả tích hợp của họ thuật toán trên FPGA; Phần còn lại trình bày các kết quả đạt được của bài báo.

1. **Mô tả lớp nguyên thủy mật mã F2/1**

Các lớp CE F2/1 [2, 3] được mô tả trên hình 1,trong đó:

* F2/1 được biểu diễn ở dạng tổng quát (Hình 1a);
* F2/1 được biểu diễn như một cặp hàm boole (Boole Function–HB) *f*1, *f*2 với 3 biến vào, và *y*1=*f*1(*x*1, *x*2, *v*); *y*2=*f*2(*x*1, *x*2, *v*) (Hình 1b). Có thể xây dựng được rất nhiều cặp HB (*f*1, *f*2) khác nhau từ 3 biến vào;
* F2/1 được biểu diễn như là cặp hai hộp thế F(0)2/1 và F(1)2/1. Ở đây, nếu *v*=0 thì CE thực hiện phép thế F(0)2/1 và nếu *v*=1 thì CE thực hiện phép thế F(1)2/1. Hay nói cách khác là giá trị của véc tơ điều khiển sẽ lựa chọn phép thế cơ bản tương ứng.

Các phương trình biểu diễn mối quan hệ giữa của biến của CE F2/1 được mô tả như sau:

(1)

(2)

Trong [2], đã đưa ra các tiêu chí để thực hiện việc phân loại các CE F2/1 và từ đó lựa chọn phần tử sao cho phù hợp với các ứng dụng mật mã. Tiêu chuẩn dưới đây cho phép lựa chọn các toán tử phi tuyến F2/1 phù hợp với việc thiết kế phần tử mở rộng F*n*/*m* cho ứng dụng cho mật mã:

1. Các hàm *f*1, *f*2 của F2/1 là hàm logic 3 biến cân bằng và có độ phi tuyến (NL- Non Linearity) lớn nhất;
2. Phép biến đổi của F2/1 là song ánh;
3. Tổ hợp tuyến tính của hai đầu ra của F2/1 tức là hàm cũng là hàm logic cân bằng và có độ phi tuyến lớn nhất;
4. Mỗi CE đều thỏa mãn tính xoắn.



*Hình 1. Cấu trúc biểu diễn của CE* **F**2/1

*a. Dạng tổng quát; b. Dạng HB 3 biến vào; c. Dạng 2 phép thế;*

*d. CE thuộc loại* **P**2/1*; e. Đặc trưng vi sai của* **F**2/1

1. **Phát triển họ thuật toán mã khối tốc độ cao**

Họ thuật toán mã khối mới được xây dựng dựa trên CE F2/1 với cấu trúc giống nhau, trong đó thiết kế của nhánh xử lý dữ liệu bên phải là hoàn toàn mới so với các kiến trúc đã biết. Để đảm bảo chống lại khả năng tấn công dựa trên khóa yếu (vì các thuật toán đã công bố dựa trên cơ sở mạng chuyển vị - thay thế điều khiển được (Controlled Substitution Permutation Network - CSPN) đều sử dụng lược đồ khóa đơn giản [2, 5-7]), họ thuật toán sẽ được thiết kế sử dụng nguyên tắc sinh khóa vòng “*khi đang chạy*” (on-the-fly).

Nhờ các nguyên tắc xây dựng này, các họ thuật toán đề xuất sẽ có tốc độ xử lý và hiệu năng cao trên FPGA (Field Programmable Gate Array) và có khả năng chống lại các tấn công kiểu RKDA. Họ thuật toán đề xuất được ký hiệu tương ứng như sau: (2/1)TMN64 và (2/1)TMN128 là các thuật toán được phát triển dựa trên lớp CE F2/1 được ký hiệu tương ứng là TMN64 và TMN128.

***3.1.Mô tả thuật toán***

TMN64 vàTMN128 là họ thuật toán được phát triển nhằm mục tiêu hoạt động hiệu quả trên nền tảng FPGA. Chúng là các thuật toán mật mã khối có kích thước khối 64 hoặc128 bit và khóa bí mật 128 hoặc 256 bit tương ứng với *R* vòng mã hóa (*R* sẽ phụ thuộc vào kết quả đánh giá độ an toàn của từng thuật toán).

*Tổng quát, họ thuật toán TMN được mô tả như sau*:

***Thuật toán:***

1. For *j* = 1 to *R* -1 do {(*A*, *B*) 🠠 Crypt(*e*)(*A*, *B*, *Q′j* ); (*A*, *B*) 🠠 (*B*, *A*)}
2. {(*A*, *B*) 🠠 Crypt(*e*)(*A*, *B*, *Q′R* )}
3. {(*A*, *B*) 🠠 (*A* ⊕ *Q′R*+1, *B* ⊕ *Q′R*+1)}.

Trong đó:

*R* là số vòng mã hóa của thuật toán;

*R* + 1 tương ứng với phép biến đổi cuối (FT) (biến đổi cuối thực hiện bằng cách thực hiện phép XOR trên nửa khối dữ liệu trái và phải với khóa vòng tương ứng đã chọn);

*e*∈{0, 1} là tham số định nghĩa với (*e*=0) là mã hóa và (*e*=1) là giải mã;

Crypt(*e*) là hàm mô tả sự biến đổi của vòng mã hóa cơ sở của các thuật toán. Với các thuật toán này, quá trình mã hóa và giải mã cùng sử dụng chung một sơ đồ.

Mô tả của hai thuật toán được chỉ ra trên Hình 2.



*Hình 2. Sơ đồ của hai thuật toán;*

*(a).Tổng quát; (b).Sơ đồ một vòng mã hóa cơ sở của TMN64*

*(c). Sơ đồ một vòng mã hóa cơ sở của TMN128*

Trên Hình 2b và 2c, mô tả chi tiết một vòng mã hóa cơ sở (hàm biến đổi Crypt(*e*)) của họ thuật toán TMN64 và TMN128 tương ứng.

Trong họ thuật toán sử dụng CE kiểu (*h*, *g*). Bộ phần tử này được biểu diễn như sau:

*y*1=*x*2*v**x*1*x*2;*y*2= *x*1*v**x*2; *y*3=*x*1*v**x*1*x*2*v* (3)

**Thuật toán TMN64**

Trong nhánh trái của thuật toánTMN64 sử dụng hai CSPN: F32/96, F−132/96. Trong đó F32/96 là sự kết nối của tám khối F8/12và F-18/12 (hình 3a), còn F−132/96 là sự kết nối của tám khối F-18/12và F8/12 (Hình 3b).



*Hình 3. Cấu trúc* F32/96*(a) và* F−132/96*(b)*

Các mạng này được thiết kế dựa trên các CE F2/1 được thiết kế như chỉ ra trong Hình 4a và 4b.



*Hình 4. Cấu trúc* F8/12*(a) và* F−18/12 *(b)*

Hoán vị **I**1 được thiết kế như sau: (1)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,29)(10)(11,18)(12,26)(14)(15,22)(16,30)(19)(20,27)(23)(24,31)(28)(32)

Nhánh phải của thuật toán TMN64 sử dụng Si được thiết kế như mô tả trên Hình 5.



*Hình 5. Cấu trúc Si(a); khối Sd(b); khối S-1d (c)*

Hoán vị I2 được thiết kế như sau: (1)(2,5)(3,9)(4,13)(6)(7,10)(8,14)(11)(12,15)(16)

Khối toán tử điều khiển được P16/8 được thiết kế như mô tả tại Hình 6.



*Hình 6. Cấu trúc* P2/1 *(a);* P16/8 *(b);*P32/16 *(c)*

Trong đó, véc tơ điều khiển *U*lo sử dụng 8 bit thấp của dữ liệu các nhánh để điều khiển các khối Sd và S-1d.

Khối mở rộng E được thiết kế như sau:

E(*X*) = (*X*, *X* <<8, *X* <<16),

trong đó <<<*b* là phép dịch trái quay vòng *b* bit.

**Thuật toán TMN128**

Trong nhánh trái của thuật toán TMN128 sử dụng F64/192, F−164/192, ở đây F64/192 là sự kết nối của 16 khối F8/12và F-18/12(hình 7a), còn F−132/192 là sự kết nối của 16 khối F-18/12và F8/12 (hình 7b).

Các CSPN này được thiết kế dựa trên các CE F2/1 như chỉ ra trong Hình 4a và 4b.

Hoán vị I1 được thiết kế như sau: (1)(2,9)(3,17)(4,25)(5,33)(6,41)(7,49)(8,57)(10)(11,18)(12,26)(13,34)(14,42)(15,50)(16,58)(19)(20,27)(21,35)(22,43)(23,51)(24,59)(28)(29,36)(30,44)(31,52)(32,60) (37)(38,45)(39,53)(40,61)(46)(47,54)(48,62)(55)(56,63) (64).



*Hình 7. Cấu trúc* F64/192*(a) và* F−164/192*(b)*

Hoán vị I2 được thiết kế như sau: (1)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,29)(10)(11,18) (12,26)(14)(15,22)(16,30)(19)(20,27)(23)(24,31)(28)(32)

Khối P32/16 được thiết kế như mô tả trên hình 6c, trong đó, véc tơ điều khiển *U*lo sử dụng 16 bit thấp của dữ liệu các nhánh để điều khiển các khối Sd và S-1d.

Khối mở rộng E được thiết kế như sau:

**E**(*X*)=(*X*, *X* <<12, *X* <<24)



*Hình 8. Cấu trúc* Si*(a);* Sd*(b);* S-1d*(c)*

Các hộp S4x4: các thuật toán đề xuất sẽ sử dụng các hộp S4x4 đã được chứng minh là an toàn trong thuật toán Serpent-1 [8].

***3.2. Thiết kế lược đồ khóa***

Phương án mở rộng khóa mật là giải pháp đơn giản và hiệu quả để ngăn ngừa các điểm yếu của khóa theo lý thuyết của mật mã khối [2]. Tuy nhiên, để thực hiện mở rộng khóa trong trường hợp thiết bị có tài nguyên hạn chế sẽ cần thêm tài nguyên (khi thực hiện bằng thiết bị) hoặc phải tính toán trước tất cả khóa vòng trước khi bắt đầu mã hóa, điều này sẽ làm giảm tốc độ mã khi thường xuyên phải thay đổi khóa mật. Để khắc phục các vấn đề này, đề xuất kỹ thuật sinh khóa vòng theo kiểu “*khi đang chạy*”. Với phương án này, sơ đồ thuật toán mã sẽ đưa vào một khối mở rộng khóa (key expand), để sinh khóa vòng tiếp theo trong thời gian thực hiện vòng mã hóa (giải mã) hiện thời.

Sơ đồ chung của các thuật toán mã hóa khi sử dụng lược đồ khóa “*khi đang chạy*” được mô tả trong Hình 9.



*Hình 9. Sơ đồ chung của họ thuật toán đề xuất với lược đồ khóa vòng “khi đang chạy”*

*Mô tả chi tiết về thủ tục sinh khóa vòng* (cho họ thuật toán TMN):

1. Phía trước vòng mã hóa thứ nhất sử dụng toán tử chuyển mạch T(*e*), thực hiện biến đổi trên 32 bit (hoặc 64 bit) khóa (*k*1, *k*2) để sinh ra khóa vòng *Q*1=(*X*, *Y*) (tương ứng với các thuật toán), trong đó khóa mật có dạng: *K*=(*k*1, *k*2, *k*3, *k*4, *k*5, *k*6, *k*7, *k*8).

Chú ý: trong chế độ mã hóa (*e*=0), thì T(*e*)(*X*, *Y*)=(*X*, *Y*) và trong chế độ giải mã (*e*=1), thì T(*e*)(*X*, *Y*)=(*Y*, *X*).

Vì vậy, trong chế độ mã hóa *Q*1=(*k*1, *k*2) và trong chế độ giải mã *Q*1=(*k*2, *k*1).

1. Vòng (*R/*2+1) không sử dụng toán tử chuyển mạch T(*e*).
2. Các vòng còn lại (1... *R/*2, *R/*2+2 ... *R*+1) sẽ sử dụng toán tử chuyển mạch T(*e*).
3. Sau vòng (*R*/2+1) sử dụng toán tử chuyển mạch T(1) (toán tử chuyển mạch cố định và không đổi trong cả trường hợp mã hóa và giải mã). Đầu vào của T(1) nhận giá trị đầu ra của khối Q(*e*)\_exp (tại *zh*/2), còn đầu ra của khối T(1) đặt vào đầu vào của khối Q\_exp (tại *zh*/2+1).

Chú ý: Khối Q\_exp không thay đổi khi thay đổi từ chế độ mã hóa sang giải mã, nghĩa là nó không phụ thuộc vào giá trị *e*=1hay*e*=0.

1. Trong vòng mã hóa Crypt(*e*),với khối dữ liệu A và B sẽ được cộng với khóa vòng *Qj*=(*X*, *Y*).

Nghĩa là: Với khối dữ liệu А: А←А⊕(*X, Y*).

Với khối dữ liệu В: B←B⊕(*X, Y*).

1. Các giá trị khóa *zh* (không phụ thuộc vào việc nó sinh ra như thế nào) khi mã/giải mã sẽ được sử dụng theo trình tự ngược nhau.

Phương án hiệu quả thực hiện lược đồ của khóa “*khi đang chạy*” để sinh ra khóa mở rộng nhờ sử dụng toán tử chuyển mạch. Phương án này được biểu diễn trên Hình 10.

Ở đây, đối với thuật toán TMN64 thì *m*=32 và *n*=16; Đối với thuật toán TMN128 thì *m*=64 và *n*=32.



*Hình 10. Thủ tục Q\_exp để mở rộng khóa mật thành các khóa vòng “khi đang chạy”*

Các thủ tục lặp để nhận được khóa mở rộng Q\_exp bao gồm các khối Sd và S-1d, còn nhánh phải sử dụng các vòng mã hóa (Hình 9). Các khối Sd và S-1d được thiết kế sử dụng các hộp thế S*i*, S-1*i* kích thước 4×4 và các CE P2/1. Các khối Sd và S-1d được mô tả trên các hình 5b, 5c và 8b, 8c, tương ứng với các thuật toán TMN64 và TMN 128.

Thủ tục Q\_exp không có tính xoắn (involution), nhưng là toán tử chuyển mạch, tức là: Q\_exp=(Q\_exp)-1.

Đặc tính chuyển mạch của thủ tục Q\_exp đạt được do sử dụng khối chuyển vị T(*e*) trên sơ đồ hình 9 (thực chất là khối chuyển vị  hoặctương ứng với các thuật toán TMN64 và TMN128).

Phần tử chuyển mạch T(*e*) thực hiện theo công thức sau: T(*e*)(*X*, *Y*)=(*X*, *Y*), nếu *e*=0 và T(*e*)(*X*, *Y*)=(*Y*, *X*), nếu *e*=1, ở đây: *X, Y*∈{0, 1}16 hoặc *X, Y*∈{0, 1}32, *e*=0 – mã hóa, *e*=1 – giải mã (tương ứng với các thuật toán).

*Trường hợp sử dụng khóa mật 128 bit (TMN64)*: thủ tục sinh ra khóa vòng phức tạp sử dụng khóa mật trong dạng: *K*=(*k*1, *k*2, *k*3, *k*4, *k*5, *k*6, *k*7, *k*8).

Các giá trị *z*1, *z*2, …, *zh*, sử dụng trong thủ tục sinh khóa vòng phức tạp thể hiện trong các bảng sau (ở đây các khóa con (*k*1, *k*2, …, *k*8) có kích thước 16 bit.

BẢNG 1. LƯỢC ĐỒ KHÓA CỦA THUẬT TOÁN **TMN64** (Với R=10)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Vòng *j* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Giá trị *h* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | - |
| Mã hóa *zh* | *k*7 | *k*3 | *k*5 | *k*1 | *k*6 | *k*6 | *k*1 | *k*5 | *k*3 | *k*7 | - |
| Giải mã *zh* | *k*7 | *k*3 | *k*5 | *k*1 | *k*6 | *k*6 | *k*1 | *k*5 | *k*3 | *k*7 | - |

*Trường hợp sử dụng khóa mật 256 bit (TMN128)*: thủ tục sinh ra khóa vòng phức tạp sử dụng khóa mật trong dạng: *K*=(*k*1, *k*2, *k*3, *k*4, *k*5, *k*6, *k*7, *k*8).

Các giá trị *z*1, *z*2, …, *zh*, sử dụng trong thủ tục sinh khóa vòng phức tạp thể hiện trong các bảng sau (ở đây các khóa con (*k*1, *k*2, …, *k*8) có kích thước 32 bit.

BẢNG 2. LƯỢC ĐỒ KHÓA CỦA THUẬT TOÁN TMN128(VỚI R=14)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Vòng *j* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Giá trị *h* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | - |
| Mã hóa *zh* | *k*7 | *k*3 | *k*5 | *k*1 | *k*6 | *k*4 | *k*2 | *k*2 | *k*4 | *k*6 | *k*1 | *k*5 | *k*3 | *k*7 | - |
| Giải mã *zh* | *k*7 | *k*3 | *k*5 | *k*1 | *k*6 | *k*4 | *k*2 | *k*2 | *k*4 | *k*6 | *k*1 | *k*5 | *k*3 | *k*7 | - |

1. **Phân tích độ an toàn của họ thuật toán**

***4.1. Đánh giá các tiêu chuẩn thống kê***

Để đánh giá các tiêu chuẩn thống kê họ thuật toán dựa trên CE F2/1, phần này sẽ thực hiện các thử nghiệm theo phương pháp đã được mô tả của NESSIE [9]. Theo đó độ an toàn của các thuật toán mã khối được xem xét dựa trên 4 tiêu chí sau đây:

* Số lượng trung bình các bit đầu ra thay đổi khi thay đổi một bit đầu vào (kí hiệu: *d*1).
* Mức độ biến đổi hoàn toàn (kí hiệu: *d*c).
* Mức độ của hiệu ứng thác lũ (kí hiệu: *d*a).
* Mức độ phù hợp với tiêu chuẩn hiệu ứng thác lũ chặt chẽ (kí hiệu: *dsa*).

Đồng thời căn cứ vào đó thì độ an toàn hay các phép biến đổi trong các thuật toán mã khối sẽ đảm bảo khi các điều kiện sau đồng thời xảy ra: *dc*=1, *da*≈1, *dsa*≈1 và *d*1≈ ½ *n*.

Các kết quả phân tích thống kê được thực hiện thông qua đánh giá sự ảnh hưởng (lan truyền) của các bit bản rõ lên các bit bản mã. Thử nghiệm được thực hiện cho hai trường hợp:

- 1 khóa và 10000 bản rõ (\*)

- 100 khóa và 100 bản rõ (\*\*)

BẢNG 3. CÁC KẾT QUẢ ĐÁNH GIÁ ẢNH HƯỞNG CỦA CÁC BIT BẢN RÕ LÊN BẢN MÃ (THUẬT TOÁN **TMN64**)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | \* | \* | \* | \* | \*\* | \*\* | \*\* | \*\* |
| *r* | (1) | (2)=*dc* | (3)=*da* | (4)=*dsa* | (1) | (2)=*dc* | (3)=*da* | (4)=*dsa* |
| 10 | 32,011 | 1,0000 | 0,9992 | 0,9921 | 32,004 | 1,0000 | 0,9995 | 0,9910 |
| 4 | 32,248 | 1,0000 | 0,9942 | 0,9884 | 32,120 | 1,0000 | 0,9945 | 0,9891 |
| 3 | 26,425 | 1,0000 | 0,8580 | 0,8259 | 26,456 | 1,0000 | 0,8196 | 0,8126 |
| 2 | 13,468 | 0,9543 | 0,5234 | 0,5123 | 13,556 | 1,0000 | 0,5150 | 0,5149 |
| 1 | 6,356 | 0,5875 | 0,1275 | 0,1497 | 6,429 | 0,4078 | 0,1373 | 0,1266 |

Các kết quả đánh giá thống kê của TMN64 (Bảng 3) và TMN128 (Bảng 4) được thực hiện tương tự với các thử nghiệm trên các thuật toán tham dự vòng chung khảo AES [9].

Các kết quả nhận được đã chỉ ra rằng, sau 4 vòng ảnh hưởng của các bit bản rõ lên bản mã đã đáp ứng tiêu chuẩn đòi hỏi. Vì vậy, các vòng mã hóa sử dụng trong thuật toán TMN64 và TMN128 là đủ an toàn đối với các phân tích thống kê.

BẢNG 4. CÁC KẾT QUẢ ĐÁNH GIÁ ẢNH HƯỞNG CỦA CÁC BIT BẢN RÕ LÊN BẢN MÃ (THUẬT TOÁN **TMN128**)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | \* | \* | \* | \* | \*\* | \*\* | \*\* | \*\* |
| *r* | (1) | (2)=*dc* | (3)=*da* | (4)=*dsa* | (1) | (2)=*dc* | (3)=*da* | (4)=*dsa* |
| 14 | 64,001 | 1,0000 | 0,9992 | 0,9921 | 64,004 | 1,0000 | 0,9994 | 0,9920 |
| 4 | 64,048 | 1,0000 | 0,9942 | 0,9884 | 64,120 | 1,0000 | 0,9937 | 0,9881 |
| 3 | 52,355 | 1,0000 | 0,8180 | 0,8159 | 52,456 | 1,0000 | 0,8196 | 0,8176 |
| 2 | 26,458 | 0,9853 | 0,4134 | 0,4133 | 26,556 | 1,0000 | 0,4150 | 0,4149 |
| 1 | 6,244 | 0,4885 | 0,0975 | 0,0897 | 6,229 | 0,5078 | 0,0973 | 0,0966 |

Bởi vì họ thuật toán sử dụng lược đồ sinh khóa “*khi đang chạy*”, nên trong bài báo này sẽ không thực hiện các đánh giá ảnh hưởng của các bit khóa lên các bit bản mã.

* 1. ***Đánh giá độ an toàn qua thám mã lượng sai***

Khi xây dựng, các thuật toán mật mã khối cần phải được đánh giá độ an toàn trước các tấn công vi sai (Differential Attack - DA) [2, 3] và tấn công tuyến tính (Linear Attack - LA) [2, 4]. Với các thuật toán dựa trên các toán tử kiểu DDO, chúng sử dụng các CE được thiết kế theo các tiêu chí đảm bảo độ phi tuyến lớn nhất, điều này không có nghĩa là chống được tấn công tuyến tính. Vì vậy, đối với các thuật toán kiểu này, sẽ chỉ thực hiện các đánh giá độ an toàn theo tiêu chuẩn thám mã lượng sai.

BẢNG 5. ĐẶC TRƯNG VI SAI CỦA CE F2/1 SỬ DỤNG TRONG HỌ THUẬT TOÁN ĐỀ XUẤT

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***ijk*** | **001** | **101** | **201** | **011** | **111** | **211** | **110** | **210** | **120** | **021** | **121** | **221** |
| *P* | 0,25 | 0,5 | 0,25 | 0,25 | 0,5 | 0,25 | 0,5 | 0,5 | 1 | 0,25 | 0,5 | 0,25 |



*Hình 11. Đặc trưng vi sai hai vòng của TMN64*

Kết quả đánh giá thám mã lượng sai của thuật toán TMN64 chỉ ra rằng, đặc trưng vi sai tốt nhất tương ứng với giá trị vi sai hai vòng với một bit chủ động. Đặc trưng như vậy có xác suất *P*(2) ≤ 2-30.

*Nó được tính toán cụ thể như sau*:

Theo sơ đồ hình thành đặc trưng vi sai hai vòng với vi sai (Δ*A*1, Δ*B*0) → (Δ*A*1, Δ*B*0) trên hình 11.

Một bit chủ động (bên nhánh trái) Δ*A*1 sẽ đi qua vòng thứ nhất với xác suất: *p*1(Δ*A*1→Δ*A*1)= *p*1*p*2=(*P*(*ijk*))12=(*P*(110))12 =0.5122-12. Nó sẽ hình thành tại đầu ra giá trị Δ*A*1 đưa sang nhánh bên phải của vòng thứ hai.

Nhánh bên phải của vòng thứ nhất sẽ thực hiện biến đổi qua Si với vi sai Δ*B*0→Δ*B*0 với xác suất *p*2 =1.

Khi một bit chủ động Δ*B*1 đi qua vòng thứ hai (đi qua Si) sẽ sinh ra Δ*B*1 với xác suất:

*p*4(Δ*B*1→Δ*B*1)=*P*(Δ*B*Δ*B*1) 2-7.

Các khối F32/96 và F-132/96 (tại vòng thứ hai) sẽ được điều khiển bởi ba bit chủ động (sinh ra qua khối mở rộng E). Vì vậy, xác suất của vi sai Δ*A*0→Δ*A*0 qua khối F32/96 sẽ là: *p*3(Δ*A*0→Δ*A*0)=(*P*(001))3=2-6.

Tượng tự, xác suất của vi sai qua khối F-132/96 là Δ*A*0→Δ*A*0 sẽ là: *p*5(Δ*A*0→Δ*A*0) = (*P*(001))3 = 2-6.

Như vậy, xác suất của đặc trưng vi sai sau hai vòng của TMN64 sẽ là: *P*(2)= *p*1.*p*2.*p*3.*p*4.*p*5 2-31.

*Vì vậy để đảm bảo an toàn trước các kiểu tấn công vi sai, thuật toán TMN64 cần sử dụng 10 vòng mã hóa.*

Đối với thuật toán TMN128: các chứng minh về thám mã vi sai cũng được thực hiện tương tự. Chỉ khác là, xác suất vi sai đi qua Si sẽ  2-8. Do đó xác suất của đặc trưng vi sai sau hai vòng của TMN128 sẽ là: *P*(2)=*p*1.*p*2.*p*3.*p*4.*p*5 2-32.

*Vì vậy để đảm bảo an toàn trước các kiểu tấn công vi sai, thuật toán TMN128 cần sử dụng 14 vòng mã hóa.*

Tổng kết giá trị đặc trưng vi sai của họ thuật toán đề xuất được chỉ ra trên bảng 6.

Nhận xét: dựa trên các kết quả được mô tả trên bảng 6, có thể nhận thấy rằng đặc trưng vi sai của họ thuật toán dựa trên CE F2/1 với số vòng đủ nhỏ là đủ để chống lại kiểu tấn công thám mã lượng sai.

BẢNG 6. ĐẶC TRƯNG VI SAI CỦA HỌ CÁC THUẬT TOÁN PHÁT TRIỂN DỰA TRÊN F2/1 và F2/2.

|  |  |  |  |
| --- | --- | --- | --- |
| **Thuật toán** | **Đặc trưng**  **vi sai** | **Xác suất** | **Số vòng đảm bảo**  **an toàn** |
| (2/1)TMN64 | (Δ*A*1, Δ*B*0) → (Δ*A*1, Δ*B*0) | 2-31 | 10 |
| (2/1)TMN128 | (Δ*A*1, Δ*B*0) → (Δ*A*1, Δ*B*0) | 2-32 | 14 |

1. **Phân tích hiệu quả thực hiện của họ thuật toán đề xuất trên FPGA**

Các mô hình đánh giá hiệu quả tích hợp của họ các thuật toán TMN được thực hiện dựa trên các mô hình trong [2, 4]. Họ thuật toán này được cài đặt trên FPGA theo cấu trúc vòng lặp cơ sở và đường ống toàn phần (trong chế độ làm việc ECB). Các kết quả nhận được được so sánh với hiệu quả cài đặt của các thuật toán tham dự vòng chung khảo AES [9]. Chúng đã chứng tỏ rằng, các họ thuật toán đề xuất có tốc độ hoạt động và hiệu quả tích hợp cao hơn so với các thuật toán chung khảo AES.

Điều đó chứng tỏ rằng họ các thuật toán TMN rất phù hợp cho tích hợp trên nền tảng FPGA và chúng hoàn toàn phù hợp để sử dụng trong xây dựng các ứng dụng truyền dữ liệu thời gian thực.

BẢNG 7. KẾT QUẢ ĐÁNH GIÁ HIỆU QUẢ TÍCH HỢP CỦA HỌ CÁC THUẬT TOÁN (2/1)TMN trên FPGA

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Thuật toán | Khối dữ liệu (bit) | Số vòng  thuật toán | Tài nguyên (CLBs) | Tần số (MHz) | *Thông lượng (Mbps)* | *Hiệu quả* | |
|  |  |
| Cấu trúc vòng lặp cơ sở (Iterative) | | | | | | | |
| TMN64 | 64 | 12 | 97 | 176 | 939 | 9.68 | 54.98 |
| TMN128 | 128 | 14 | 186 | 179 | 1,637 | 8.80 | 49.16 |
| AES | 128 | 10 | *598* | *225* | *2,880* | *4.82* | *21.4* |
| MARS | 128 | 32 | *2,105* | *22* | *88* | *0.042* | *1.9* |
| Twofish | 128 | 16 | *604* | *80* | *640* | *1.06* | *13.25* |
| Serpent | 128 | 32 | *2,108* | *272* | *1,088* | *0.516* | *1.9* |
| RC6 | 128 | 20 | *1,102* | *90* | *576* | *0.523* | *5.81* |
| Cấu trúc đường ống toàn phần (Pipeline) | | | | | | | |
| TMN64 | 64 | 12 | *479* | *195* | 12,480 | 25.52 | 130.88 |
| TMN128 | 128 | 14 | *914* | *194* | 24,832 | 25.73 | 132.64 |
| AES | 128 | 10 | *3,572* | *58* | *7,424* | *2.1* | *35.8* |
| MARS | 128 | 32 | *14,639* | *1.37* | *175* | *0.012* | *8.7* |
| Twofish | 128 | 16 | *5,355* | *85* | *10,880* | *2* | *23.9* |
| Serpent | 128 | 32 | *15,459* | *44* | *5632* | *0.36* | *8.28* |
| RC6 | 128 | 20 | *8,001* | *1.53* | *196* | *0.025* | *16* |

1. **Kết luận**

Bài báo đã đạt được các kết quả chính như sau: xây dựng họ thuật toán mật mã khối tốc độ cao TMNdựa trên lớp CE **F**2/1 và đề xuất phương án xây dựng lược đồ sinh khóa vòng “*khi đang chạy*”. Với lược đồ này, họ các thuật toán đề xuất vẫn đảm bảo hiệu năng cao khi thực hiện thay đổi khóa thường xuyên và chống lại điểm yếu lớn nhất của các kiểu thuật toán dựa trên CSPN do sử dụng lược đồ sinh khóa đơn giản; đã thực hiện đánh giá độ an toàn của họ các thuật toán này thông qua tiêu chuẩn thống kê NESSIE và thám mã lượng sai.

Các kết quả nhận được đã chứng minh rằng, đây là họ các thuật toán đủ an toàn theo các tiêu chí đánh giá phổ biến trên thế giới. Thực hiện đánh giá hiệu quả tích hợp của họ các thuật toán đề xuất trên FPGA. Các kết quả nhận được đã chứng minh, chúng đủ hiệu quả trên phần cứng FPGA và có thể sử dụng truyền dữ liệu thời gian thực trên các mạng không dây.

**Tài liệu tham khảo**

1. Jiejun Kong, Mario Gerla, “Adaptive Security Support for Real-time Transmissions in Ad Hoc Networks”, <http://netlab.cs.ucla.edu/wiki/files/MEDHOCNET-jkong.pdf>.
2. N. Moldovyan and A. Moldovyan., “Data-driven block ciphers for fast telecommunication systems,”, Auerbach Publications, 2008.
3. N. Moldovyan and A. Moldovyan, “Innovative Cryptography”, Charles River Media, 2007.
4. Nguyen Hieu Minh, Do Thi Bac, Ho Ngoc Duy (2010), “New SDDO-Based Block Cipher for Wireless Sensor Network Security,”, International Journal of Computer Science and Network Security, pp.54-60, VOL.10 No.3.
5. A.V. Bodrov, A.A. Moldovyan, P.A. Moldovyanu, “DDP-based Cipers: Differential Analysis of SPECTR-H64”, Computer Science Journal of Moldova. 2005. Vol. 13, Number 3(39), pp.268-291.
6. Jinkeon Kang, Kitae Jeong, Sang-Soo Yeo, Changhoon Lee (2012), Related-Key Attack on the MD-64 Block Cipher Suitable for Pervasive Computing Environments, Advanced Information Networking and Applications Workshops (WAINA), 26th IEEE International Conference, IEEE Xplore, 726-731.
7. Jinkeon Kang, KitaeJeong, Changhoon Lee, Seokhie Hong (2013), “Distinguishing attack on SDDO-based block cipher BMD-128”, In Proceedings of the 8th International Conference on Ubiquitous Information Technologies and Applications (CUTE 2013), Springer Lecture Notesin Electrical Engineering, 595-602.
8. Ross Anderson, Eli Biham, Lars Knudsen, “Serpent: A Proposal for the Advanced Encryption Standard,", <http://cryptosoft.net/docs/Serpent.pdf>.
9. “NESSIE. New European Schemes for Signatures, Integrity, and Encryption”, <https://www.cosic.esat.kuleuven.ac.be/nessie/>

**Phạm Mạnh Tuấn, Đỗ Thị Bắc, Đỗ Thành Nam, Nguyễn Hiếu Minh**

Theo bài báo “New Block Ciphers for Wireless Mobile Network” trong ICTA 2016, pp.393-402