Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng, ngày 17-18/08/2017
DOI: 10.15625/vap.2017.000114

VỀ MỘT THUẬT TOÁN MÃ HOÁ KHÓA ĐỐI XỨNG CẢI TIẾN

Nguyễn Đức Toàn[1], Bùi Thế Hồng[2], Nguyễn Văn Tảo[3]

[1] Trường Cao đẳng Công nghiệp Thực phẩm

[2] Trường Đại học Sư phạm Kỹ thuật Hưng Yên

[3] Trường Đại học Công nghệ Thông tin và Truyền thông Thái Nguyên

ductoanndt9@gmail.com, hongbuithe@gmail.com, nvtao@ictu.edu.vn

TÓM TẮT: *Bài báo này đề xuất một thuật toán mã hoá khóa đối xứng cải tiến, trong đó chỉ sử dụng ba phép toán cơ bản, đó là phép cộng modulo, phép dịch vòng bit và phép cộng bit loại trừ XOR. Khác với các thuật toán mã hoá khối đã có trước đây, trong thuật thuật toán này, độ dài của khối bản rõ, độ dài của khóa bí mật và số vòng lặp mã hoá có thể thay đổi, còn độ dài của bản mã lại dài gấp đôi độ dài của bản rõ. Một điểm đặc biệt nữa của thuật toán là sử dụng một phép dịch vòng không được định trước mà hoàn toàn phụ thuộc vào dữ liệu cần mã hoá. Phép toán này giúp tăng thêm độ mật của thuật toán. Quy trình tạo khóa và các lược đồ mã hoá và giải mã chạy nhanh vì chỉ phải thực hiện các phép toán đơn giản.*

Từ khóa: *Mật mã khóa đối xứng, phép quay phụ thuộc dữ liệu, mã hoá khối.*

I. GIỚI THIỆU

Từ xa xưa và nhất là hiện nay, để bảo vệ những thông điệp quan trọng và nhạy cảm, người ta đều mã hoá chúng trước khi truyền đi trên các phương tiện công cộng. Những nhu cầu này của thực tế đã thúc đẩy hình thành một ngành khoa học gọi là khoa học mật mã nhằm nghiên cứu các kỹ thuật biến đổi một cách bí mật các thông điệp có nghĩa thành các chuỗi kí tự vô nghĩa và ngược lại, qua đó những thông tin nhạy cảm có thể được giữ bí mật trước bất kỳ người thám mã nào.

Hiện tại đã có một số kỹ thuật mã hoá để đảm bảo an toàn cho các thông tin quan trọng dựa vào hệ mã hoá khóa đối xứng và hệ mã hoá khóa công khai.

Trong hệ mã hoá khóa công khai, người gửi dùng khóa công khai của người nhận mã hoá thông điệp, còn người nhận lại dùng khóa bí mật của mình để giải mã thông điệp đã được mã hoá. RSA là một trong những thuật toán khóa công khai nổi tiếng được ứng ứng dụng rộng rãi nhất.

Trong hệ mã hoá khóa đối xứng, người gửi và người nhận dùng cùng một khóa để mã hoá và giải mã. Hai bên đều phải có trách nhiệm giữ bí mật khóa này. DES (Data Encryption Standards) là mã khối sử dụng rộng rãi nhất trên thế giới trong thời gian vừa qua. Tuy nhiên, hiện tại DES được xem là không đủ an toàn cho nhiều ứng dụng. Nguyên nhân chủ yếu là độ dài 56 bit của khóa là quá nhỏ. Thuật toán được tin tưởng là an toàn trong thực tiễn có dạng Triple DES (thực hiện DES ba lần), mặc dù trên lý thuyết phương pháp này vẫn có thể bị phá. Gần đây DES đã được thay thế bằng AES (Advanced Encryption Standard).

Tuy thế, quá trình nâng cấp và cải tiến mã hoá khối vẫn còn tiếp diễn. Năm 1991, các tác giả của bài báo [9] đã đề xuất một Thuật toán mã hoá dữ liệu quốc tế mới (IDEA- International Data Encription Algorithm). Thuật toán này sử dụng ba phép toán cơ bản là cộng modulo, XOR và nhân modulo để đạt được độ khuếch tán và rối loạn cao. Tuy vậy, IDEA được cho là yếu về khóa. Các tác giả của bài báo [1] đã đưa ra một số lớp khóa yếu của IDEA. Tiếp theo, các tác giả của bài báo [4] đã xây dựng được một tấn công mới chống lại IDEA với 6 vòng lặp.

Bài báo này đề xuất một thuật toán mã hoá khối cải tiến nhằm nâng cao độ an toàn của bản mã bằng cách tăng độ dài khóa và cho phép người sử dụng có thể tùy biến theo nhu cầu của mình mà tự xác định được mức độ an ninh và thời gian thực hiện của thuật toán. Những đặc điểm cơ bản sau đây của thuật toán đã giúp cho kỹ thuật này đạt được những mục tiêu trên:

- Đơn giản vì chỉ sử dụng ba phép toán cơ bản, đó là cộng modulo, dịch vòng bit và cộng bit loại trừ XOR;
- Quy trình mã hoá và giải mã được thực hiện theo kiểu lặp với số vòng lặp có thể thay đổi tùy thuộc vào sự lựa chọn về mức an ninh của người sử dụng;
- Độ dài của mỗi khối bản rõ có thể thay đổi tùy vào sự lựa chọn của người sử dụng phù hợp với nhu cầu và mức độ an ninh mà họ mong muốn;
- Độ dài của khóa sử dụng trong Quy trình mã hoá phụ thuộc trực tiếp vào độ dài của khối bản rõ.

Phần còn lại của bài báo này được sắp xếp như sau. Mục II trình bày các ý tưởng cải tiến và thuật toán cải tiến. Mục III trình bày chi tiết Quy trình mã hoá và giải mã. Mục IV là các kết luận và tài liệu tham khảo.

II. Ý TƯỞNG VÀ THUẬT TOÁN CẢI TIẾN

2.1. Ý tưởng

Thuật toán mã hoá khóa đối xứng cải tiến này chỉ sử dụng ba phép toán cơ bản, đó là phép cộng modulo, phép dịch vòng bit và phép cộng bit loại trừ XOR. Ngoài ra, độ dài của khóa bí mật và số vòng lặp mã hoá có thể thay đổi tùy thuộc vào yêu cầu về độ mật của người gửi. Hơn nữa, độ dài của bản mã dài gấp đôi độ dài của bản rõ sẽ tăng thêm tính an toàn. Các phép dịch vòng phụ thuộc dữ liệu được đề xuất trong [6] cũng được sử dụng trong các quy trình mã hoá, giải mã và sinh khóa để tăng thêm tính rối loạn và khuếch tán của các bit dữ liệu.

Một vài thông số và ký hiệu sau đây sẽ được sử dụng trong thuật toán.

- Các thông số do người sử dụng tự chọn:

r: Số vòng lặp dùng để mã hoá và giải mã một khối

s: Cỡ của khối khối bản rõ do người sử dụng lựa chọn

k: Khóa được dùng trong hàm hoán đổi hai pha và bằng $log_2 s$

K_M: Khóa chính dùng để sinh các khóa con cho các vòng lặp và có độ dài bằng $2s$

$K_1 \dots K_r$: Các khóa được sử dụng trong các vòng lặp của quy trình mã hoá và giải mã. Chúng được sinh ra từ khóa chính K_M;

- Các ký hiệu:

R: Phép quay phụ thuộc dữ liệu (phép dịch vòng)

$\oplus$: Phép cộng logic loại trừ

<<<: Phép dịch vòng (quay) trái

>>>: Phép dịch vòng (quay) phải

+: Phép cộng modulo

$\sim$: Phép lấy phần bù 1.

2.2. Thuật toán cải tiến

2.2.1. Quy trình sinh khóa

Khóa hoán đổi k có độ dài thay đổi theo chiều dài của khối bản rõ và được tính bằng công thức $k = log_2 s$, trong đó s là chiều dài của khối bản rõ.

Ví dụ: Nếu $s = 128$ thì $k = log_2 s = log_2 2^7 = 7$.

Khóa chính K_M có cỡ là $2s$

Ví dụ: Nếu $s = 128$ bit thì cỡ của K_M: là 256 bit

Các khóa vòng lặp $K_1 \dots K_r$ có độ dài bằng nhau. Mỗi vòng lặp dùng một khóa, việc sinh các khóa vòng lặp được thực hiện bởi một quy trình lặp. Mỗi vòng lặp sinh một khóa, số khóa con được sinh ra bằng đúng số vòng lặp của quy trình mã hoá. Khóa chính K_M là đầu vào cho vòng lặp thứ nhất của chu trình sinh khóa và đầu ra của lần lặp trước là đầu vào của vòng lặp sau.

+ Đầu tiên, khóa chính K_M được biến đổi qua một phép hoán đổi 4 bit bằng cách hoán đổi liên tục từng khối 4 bit với khối 4 bit tiếp theo.

+ Sau đó, chia K_M (sau khi đã hoán đổi) thành hai nửa trái và phải. Tráo đổi nửa trái thành nửa phải và nửa phải thành nửa trái. Tiếp theo, nửa trái được quay vòng trái, còn nửa phải được quay vòng phải và lấy phần bù 1. Ghép hai nửa lại với nhau và tiến hành phép quay phụ thuộc dữ liệu sẽ thu được khóa vòng lặp K_1.

+ Lại lấy K_1 làm đầu vào và tiến hành biến đổi như trên sẽ thu được khóa K_2. Tiếp tục như vậy cho đến khi thu được K_r.

Phép quay phụ thuộc dữ liệu R dựa vào $k = log_2 s$ bit ít ý nghĩa nhất (các bit này không xác định chiều quay) của kết quả biến đổi khóa chính K_M và các khóa con K_i. Trong đó bit đầu tiên xác định chiều quay, các bit còn lại xác định số bit cần quay.

Quy trình sinh các khóa vòng K_i được mô tả trong Hình 1.

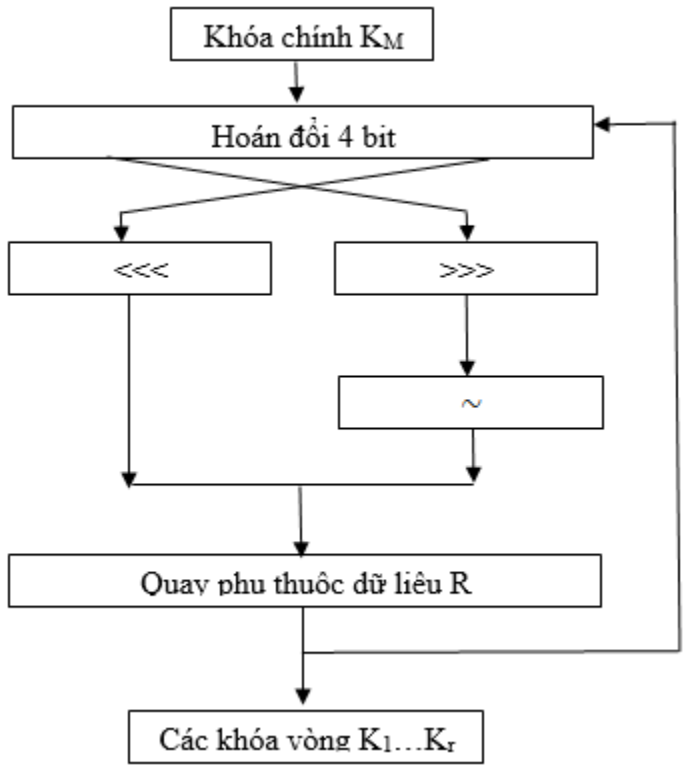
Khóa tích lũy: Khóa này dùng chung giữa người nhận và người gửi và có thể được phân phối bằng một lược đồ phân phối khóa nào đó. Khóa này gồm 4 phần và được minh họa trong Hình 2.

+ Phần đầu có độ dài 6 bit chứa số vòng lặp r. Trong thuật toán mã hoá này, số vòng lặp tối thiểu là 1 và tối đa là 63 vòng ($1 \leq r \leq 63$).

+ Phần 2 có độ dài 12 bit, biểu diễn độ dài của khối bản rõ. Số này phải chia hết cho 8, không nhỏ hơn 128 bit và không lớn hơn 2048 bit ($128 \leq s \leq 2048$). Nếu độ dài của khối bản rõ nhỏ hơn cỡ của khối thì phải chèn thêm cho đủ.

+ Phần 3 chứa khóa hoán đổi k và phần 4 chứa khóa chính K_M.

Hình 2 mô tả các thành phần của khóa tích lũy còn bảng 1 liệt kê kích thước của một số khóa tích lũy theo kích thước của khối bản rõ.



Hình 1. Quy trình sinh các khóa cho các vòng lặp K_i

R	s	k	K_M

Hình 2. Định dạng khóa tích lũy

Bảng 1. Liệt kê độ dài (tính theo bit) của khóa tích lũy tính theo một vài độ dài điển hình của khối bản rõ. Khóa tích lũy được dùng chung giữa người gửi và người nhận.

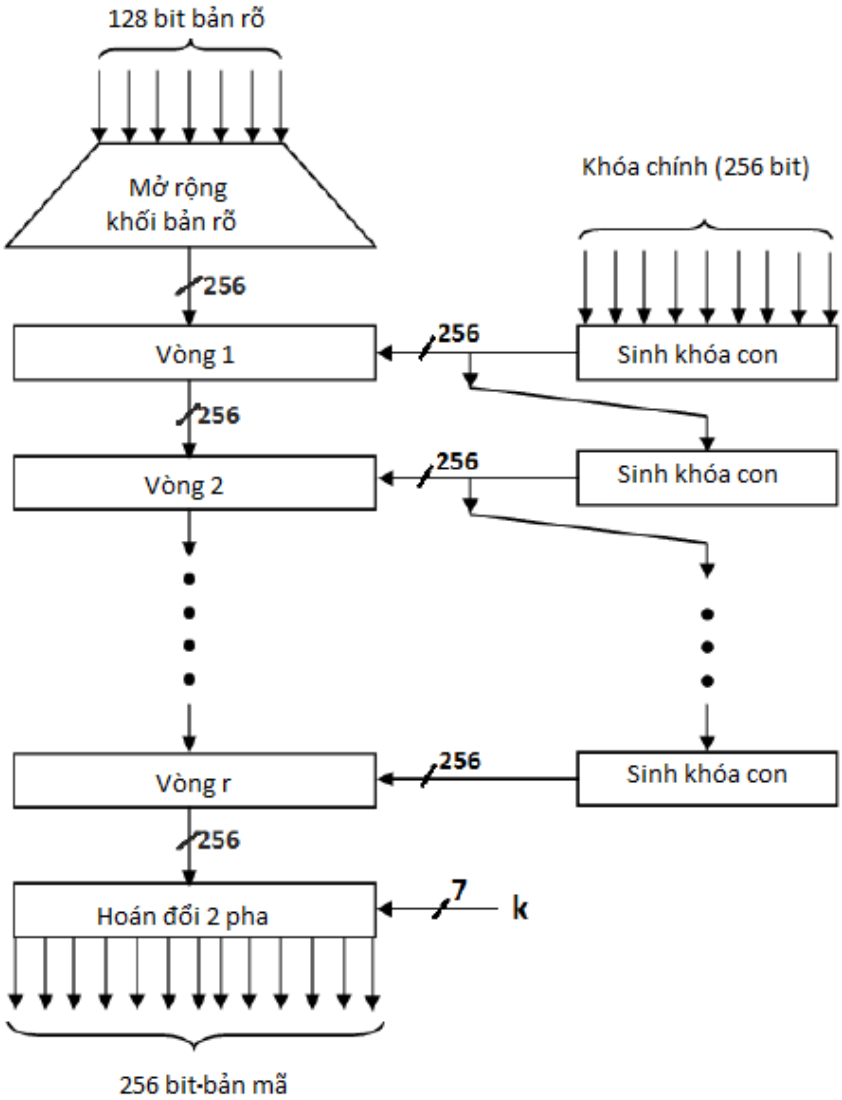
Bảng 1. Độ dài khóa tích lũy tính theo độ dài khối bản rõ

Độ dài của khối bản rõ (s)	Độ dài của khóa hoán đổi (k)	Độ dài của khóa chính (K_M)	Độ dài của khóa tích lũy
128	7	256	281
256	8	512	538
512	9	1024	1051
1024	10	2048	2076
2048	11	4096	4125

III. THUẬT TOÁN MÃ HOÁ ĐỀ XUẤT

3.1. Quy trình mã hoá

Quy trình mã hoá được đề xuất trong bài báo này gồm 3 giai đoạn như được mô tả trong Hình 2. Để tiện lợi cho quá trình mô tả, chúng tôi sẽ giả thiết độ dài s của khối bản rõ là 128 bit. Đầu tiên khối bản rõ cần mã hoá sẽ được mở rộng lên gấp đôi, sau đó thực hiện các vòng lặp và cuối cùng thực hiện một phép hoán đổi hai pha.



Hình 3. Lược đồ khối của Quy trình mã hoá 128 bit bản rõ

Giải thích lược đồ mã hoá:

- Do khối bản rõ có độ dài 128 bit nên khóa chính K_M phải có độ dài bằng 256 bit. Qua quy trình sinh khóa vòng ta có khóa K_1 (256 bit) và được chuyển vào cho vòng lặp 1.

- Vòng 1 sẽ sinh ra một bản mã trung gian gồm 256 bit từ 256 bit đã mở rộng và khóa K_1 và được chuyển thành đầu vào cho vòng lặp thứ 2. Đồng thời quy trình sinh khóa cũng sinh ra một khóa mới là K_2 để kết hợp với bản mã trung gian tạo thành bản mã trung gian thứ 2. Kết quả mã hoá của vòng trên sẽ được kết hợp với khóa vòng của bước hiện tại qua các hàm vòng sẽ tạo ra bản mã trung gian của vòng hiện tại. Cứ tiếp tục thực hiện như vậy cho đến vòng lặp thứ r.

- Cuối cùng thực hiện một hàm hoán đổi 2 pha để thu được bản mã dài 256 bit (dài gấp đôi bản rõ).

Sau đây chúng tôi sẽ mô tả chi tiết 3 giai đoạn của Quy trình mã hoá.

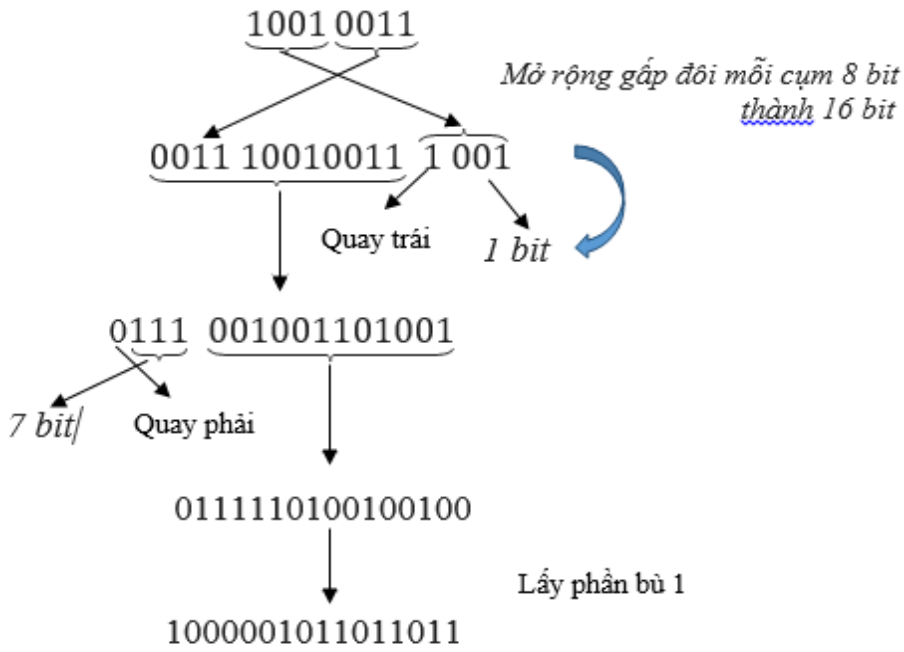
Giai đoạn 1: Mở rộng khối bản rõ. Khối này được mở rộng gấp đôi thành 256 bit và được chuyển làm đầu vào cho vòng lặp thứ nhất. Hình 3 là một ví dụ mở rộng một cụm 8 bit thành 16 bit. Cách mở rộng được thực hiện như sau:

- Chia bản rõ thành các cụm 8 bit. Mở rộng 8 bit lên thành 16 bit bằng cách giữ nguyên 8 bit ban đầu rồi tiến hành đặt 4 bit trái sang bên phải và đặt thêm 4 bit phải sang bên trái.

- Lấy 4 bit phải để xác định phép quay của 12 bit bên trái. Nếu bit trái là 1 thì quay vòng 12 bit sang trái một số bit bằng giá trị của 3 bit còn lại. Nếu là 0 thì quay phải.

- Sau đó lấy 4 bit bên trái để quay 12 bit còn lại bên phải tương tự như trên.

- Cuối cùng lấy phần bù 1 để thu được khối mở rộng của bản rõ.



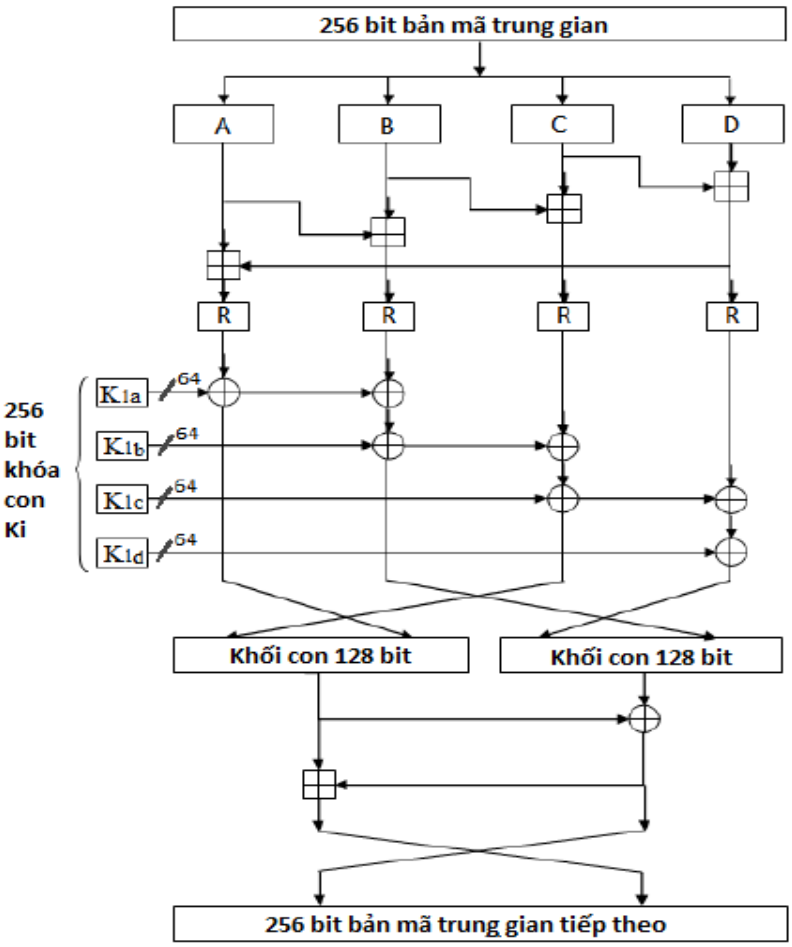
Hình 4. Mở rộng bản rõ lên gấp đôi bằng phép quay phụ thuộc dữ liệu

Giai đoạn 2: Phép quay phụ thuộc dữ liệu

Các hàm trong vòng lặp: Mỗi vòng lặp i đều có đầu vào là khối dữ liệu đang mã hoá và một khóa vòng lặp K_i. Sau đây là giả mã cho một vòng lặp:

1. $Chia\ khối\ bản\ mã\ trung\ gian\ thành\ 4\ phần\ bằng\ nhau\ A, B\ , C, D$
2. $D := C + D$
3. $C := B + C$
4. $B := A + B$
5. $A := D + A$
6. $Quay\ A, B, C, D$
7. $Chia\ khóa\ con\ K_i\ thành\ 4\ phần\ bằng\ nhau\ K_{1a}, K_{1b}, K_{1c}, K_{1d}$
8. $A := A \oplus K_{1a}$
9. $B := B \oplus A \oplus K_{1b}$
10. $C := C \oplus B \oplus K_{1c}$
11. $D := D \oplus C \oplus K_{1d}$
12. $Ghép\ C\ và\ A, gọi\ là\ E$
13. $Ghép\ D\ và\ B, gọi\ là\ F$
14. $F := E \oplus F$
15. $E := E + F$
16. $Ghép\ F\ với\ E$

Kết thúc vòng lặp r ta thu được bản mã của khối bản rõ. Kích thước của bản mã gấp đôi kích thước của khối bản rõ ban đầu. Hình 5 mô tả một vòng lặp trong r vòng mã hoá.



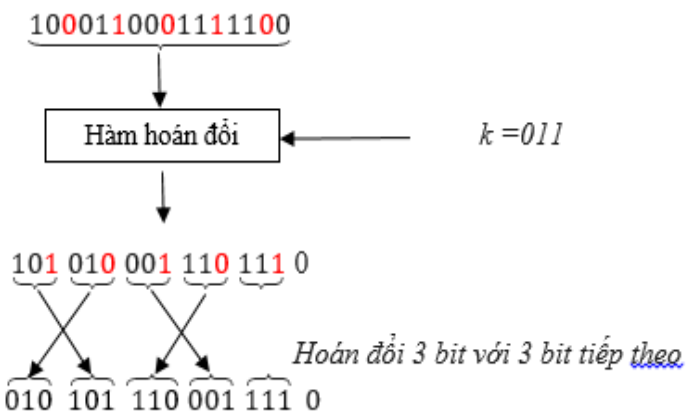
Hình 5. Một vòng lặp trong Quy trình mã hoá

Giai đoạn 3: Hàm hoán đổi 2 pha

Hàm này dùng để hoán vị các bit với khóa k do người dùng nhập vào. Hàm này có 2 pha:

- Cứ k bit lại hoán đổi bit thứ k từ 0 thành 1 hoặc từ 1 thành 0.
- Hoán đổi mỗi nhóm k bit với nhóm k bit tiếp theo.

Hình 6 minh họa cách thực hiện của hàm hoán đổi 2 pha. Người sử dụng nhập khóa dưới dạng nhị phân ví dụ là $k = 011$. Như vậy ở pha thứ nhất, cứ 3 bit thì bit thứ 3 sẽ được hoán đổi từ 1 thành 0 hoặc từ 0 thành 1. Sau đó, hoán đổi mỗi nhóm 3 bit với nhóm 3 bit tiếp theo. Nếu còn lại một số bít lẻ không đủ ghép lại thành một cặp 3 bit thì giữ nguyên.



Hình 6. Hàm hoán đổi 2 pha

3.2. Quy trình giải mã

Quy trình giải mã là quy trình ngược của quy trình mã hoá. Do đó độ phức tạp của giải mã cũng tương tự như độ phức tạp của mã hoá. Độ an toàn của quy trình mã hoá và giải mã đạt được ở mức độ cao vì tính phi tuyến được tạo ra nhờ phép quay phụ thuộc dữ liệu và số vòng lặp không cố định và các vòng lặp không bảo toàn chẵn, lẻ. Tính phi tuyến ở trong quy trình này là không thỏa mãn nguyên tắc xếp chồng - nghĩa là đầu ra không tỷ lệ thuận với đầu vào. Đây là một dạng hỗn độn cổ điển tức là không thể đoán trước được chiều quay và số vòng quay.

IV. KẾT LUẬN

Thuật toán mã hoá khóa đối xứng cải tiến được đề xuất có thể thực hiện với kích thước bất kỳ kích thước dữ liệu của bản rõ. Nhưng trong bài báo này thuật toán được tính toán trên khối bản rõ 128 bit và kích thước khóa tích lũy 281 bit để tạo ra bản mã có kích thước 256 bit. Thuật toán mã hoá khóa đối xứng cải tiến này bằng cách sử dụng các phép tính toán như: cộng modulo, phép quay vòng bit và phép cộng bit loại trừ XOR đã cung cấp một sự mềm dẻo cho người dùng để lựa chọn kích thước khối khối bản rõ và số vòng lặp cho quá trình mã hoá. Phép quay phụ thuộc dữ liệu là tính năng quan trọng nhất của thuật toán mã khóa đối xứng cải tiến. Nó giúp tạo ra sự khuếch tán mạnh mẽ trên khối bản rõ (tức là sự thay đổi 1 bit trong khối bản rõ dẫn tới sự thay đổi hoàn toàn trong khối bản mã tạo ra). Thuật toán đề xuất này có tốc độ mã hoá và giải mã nhanh hơn so với các thuật toán mã hoá khác vì chỉ phải thực hiện các phép toán cơ bản và đơn giản.

TÀI LIỆU THAM KHẢO

[1] Biryukov, J. Nakahara Jr, B. Preneel, J. Vandewalle, "*New Weak-Key Classes of IDEA*", 4th International Conference Information and Communications Security, ICICS 2002, Lecture Notes in Computer Science 2513, Springer-Verlag, pp. 315-326, 2002.

[2] D. Hong, J. K. Lee, D. C. Kim, D. Kwon, K. H. Ryu, and D. G. Lee, "*LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors*", WISA 2013, LNCS 8267, Springer International Publishing Switzerland, 2014.

[3] D. Khovratovich, G. Leurent, and C. Rechberger, "*Narrow-Bicliques: Cryptanalysis of Full IDEA*", Advances in Cryptology, EUROCRYPT 2012, LNCS 7237, Springer-Verlag, pp. 392–410, 2012.

[4] E. Biham, O. Dunkelman, and N. Keller, *"A New Attack on 6-Round IDEA"*, Proceedings of Fast Software Encryption, Lecture Notes in Computer Science, Springer-Verlag, 2007.

[5] G. Álvarez, D. de la Guía, F. Montoya, and A. Peinado, "*Akelarre: a new Block Cipher Algorithm*", Third Annual Workshop on Selected Areas in Cryptography, SAC 96, Kingston, Ontario, 1996.

[6] R. L. Rivest, "*The RC5 Encryption Algorithm*", Proceedings of the Second International Workshop on Fast Software Encryption, pp. 86-96, 1994.

[7] Rajul Kumar, K. K. Mishra, Ashish Tripathi, Abhinav Toma, Surendra Singh, *"Modified Symmetric Encryption Algorithm",* Motilal Nehru National Institute of Technology Allahabad Allahabad, India, 2014.

[8] W. Diffie and M. Hellman, "*New directions in cryptography*", IEEE Transaction on Information Theory, pp. 644–654,1976.

[9] X. Lai and J. Massey, "*A Proposal for a New Block Encryption Standard*", Advance in Cryptography, EUROCRYPT 90, Springer Verlag, Berlin, pp. 389-404, 1991.

ABOUT ONE MODIFIED SYMMETRIC BLOK CIPHER ALGORITHM

Nguyen Duc Toan, Bui The Hong, Nguyen Van Tao

ABSTRACT: *This article proposed about one modified symmetric block cipher Algorithm, in which only three elementary operations including modular addition, bit-wise rotation and bit-wise XOR are used. Unlike previous block ciphers, in this algorithm, the length of the plaintext block, the length of the secret key, and the number of encrypted loops are variable, while the length of the ciphertext is double of length of plain text. Most special feature of the algorithm is the use of the data dependant rotation through which the unpredictability of encrypted text is increasing. Key formation and encryption/decryption schemes of the proposed cipher are significantly fast.*