Đặc trưng vi sai hiệu quả cho toàn bộ số vòng của GOST 28147-89

Nguyễn Văn Long, Trần Hồng Thái, Nguyễn Bùi Cương

***Tóm tắt*— GOST 28147-89 là thuật toán mã khối của Liên bang Nga được đưa vào sử dụng từ những năm 90 của thế kỷ 20. Thuật toán này được chứng minh có khả năng chống thám mã vi sai. Trong bài báo này, chúng tôi làm rõ tính chất xáo trộn khóa sử dụng phép cộng modulo 2^{32} theo quan điểm thám mã vi sai. Tiếp theo, chúng tôi xây dựng bộ công cụ tìm kiếm đặc trưng vi sai tốt nhất cho GOST 28147-89 với số vòng rút gọn và đặc trưng vi sai hiệu quả cho GOST 28147-89 đầy đủ.**

***Abstract*— GOST 28147-89 is a well-known block cipher and the official encryption standard of the Russian Federation since 1990s. This algorithm has been approved to be capable of against the differential cryptanalysis. In this paper, we prove theorically property for key mixing with the addition modulo 2^{32} in the view of differential cryptanalysis. Then, we construct a tool-box to search either the best differential characteristic for round-reduced GOST 28147-89 or good differential characteristics for full GOST 28147-89.**

Từ khóa— GOST 28147-89; thám mã vi sai.

I. GIỚI THIỆU

Thám mã vi sai được Biham và Shamir đưa ra năm 1991 trong [1] và đã trở thành một công cụ hiệu quả trong phân tích và thám mã đối với mã khối. Mặc dù GOST 28147-89 sử dụng cấu trúc Feistel giống như chuẩn mã dữ liệu DES, nhưng ta không thể áp dụng trực tiếp thám mã vi sai gốc lên nó. Nguyên nhân là thuật toán này có sự khác biệt ở phép cộng khóa theo modulo 2^{32}. Đây cũng chính là lý do mà sau hơn 20 năm sử dụng, dù đã có nhiều công trình nghiên cứu về độ an toàn của GOST 28147-89 trước thám mã vi sai, nhưng các kết quả nghiên cứu hoặc chưa làm sáng tỏ hoàn toàn cơ sở lý thuyết, hoặc chưa đưa ra được độ đo thực hành cụ thể về khả năng kháng lại thám mã vi sai.

Courtois và cộng sự [6] đã công bố nhiều kết quả về phân tích và thám mã lên GOST 28147-89. Các kết quả này hầu hết dựa trên các "khẳng định" không có căn cứ hoặc chưa được chứng minh rõ ràng. Kết quả tiêu biểu nhất về lĩnh vực này chính là luận án của Isukova về thám mã vi sai lên một số mã pháp, trong đó có GOST 28147-89 [1]. Tuy nhiên, trong cách tiếp cận của nhóm tác giả này có một số điểm chưa được làm tường minh về mặt lý thuyết.

Trong bài báo này chúng tôi làm rõ tính chất xáo trộn khóa sử dụng phép cộng modulo 232 theo quan điểm thám mã vi sai. Sau đó dựa trên cơ sở cách tiếp cận trong [2], chúng tôi xây dựng thuật toán thực hành được cho phép xác định đặc trưng vi sai tốt nhất lên GOST 28147-89, với số vòng rút gọn và đặc trưng vi sai hiệu quả lên thuật toán đầy đủ số vòng.

Bố cục của bài báo như sau: Mục I của bài báo giới thiệu tổng quan về tình hình thám mã lên chuẩn GOST 28147-89. Sơ lược về chuẩn và phân tích các biến đổi cơ bản theo quan điểm thám mã vi sai được trình bày trong mục II. Đặc biệt trong mục này, chúng tôi chứng minh một nhận xét quan trọng mà chưa được làm rõ trong [2]. Mục III phân tích cách tiếp cận trong việc tìm kiếm đặc trưng vi sai hiệu quả và tìm kiếm các đặc trưng này với GOST 28147-89. Kết quả thực nghiệm được trình bày trong mục IV. Cuối cùng là phần kết luận.

II. GOST 28147-89 VÀ THÁM MÃ VI SAI

A. Sơ lược về GOST 28178-89

GOST 28147-89 là một thuật toán mã khối được xây dựng theo cấu trúc Feistel với kích cỡ khối dữ liệu là 64 bit tương tự như thuật toán chuẩn mã dữ liệu DES. Tuy nhiên khác với DES, thuật toán mã hóa GOST dùng khóa có kích cỡ 256 bit, do đó nâng cao đáng kể độ an toàn đối với phương pháp vét cạn. Chi tiết về thuật toán này được trình bày trong các tài liệu [2,5].

Thiết kế của GOST 28147-89 đơn giản hơn so với DES. Thuật toán có 32 vòng mã. Trong mỗi vòng, nửa bên phải của thông báo được cộng với khóa con bí mật (bởi phép cộng modulo 2^{32}), kết quả nhận được sẽ được dịch vòng sang trái 11 bit sau khi qua tầng biến đổi S-hộp. Sau đó, nửa trái và nửa phải của thông báo đổi chỗ cho nhau. Chú ý là các S-hộp sử dụng trong GOST được giữ bí mật và chúng được xem như thành phần khóa bổ sung. Tiêu chuẩn để lựa chọn các S-hộp này cũng chưa được công bố. Do vậy, một câu hỏi đặt ra là liệu thuật toán GOST 28147-89 còn an toàn hay không nếu nội dung của S-hộp được công bố.

Việc phân tích một thuật toán mã hóa bất kỳ luôn được bắt đầu từ việc phân tích các thành phần của nó, chúng có thể ảnh hưởng đến sự thay đổi sai khác trong quá trình thực hiện qua các vòng của thuật toán. Do đó bước đầu cần phải nghiên cứu các đặc tính của các thành phần của thuật toán

GOST 28147-89. Sau đây chúng tôi sẽ phân tích cụ thể những biến đổi này theo quan điểm thám mã vi sai.

B. Những biến đổi cơ sở trong GOST 28147-89 theo quan điểm thám mã vi sai

1. Phép dịch vòng sang trái 11 bit

Phép toán dịch vòng trong thuật toán mã hóa GOST 28147-89 là một trong ba phép toán cơ bản tạo nên hàm vòng. Khi xem xét thám mã vi sai, chúng ta thường đề cập tới sai khác của hai thông báo. Giả sử có hai thông báo A và B, khi đó sai khác của chúng là $(A \oplus B)$. Nếu dịch vòng mỗi giá trị A và B đi s vị trí, ta sẽ nhận được sai khác mới có dạng $(A \boxplus s) \oplus (B \boxplus s)$ và chúng có tính chất là $(A \boxplus 11) \oplus (B \boxplus 11) = (A \oplus B) \boxplus 11$. Ở đây $s = 11$, có nghĩa là để nhận được sai khác đúng ở đầu ra, cần phải dịch vòng sai khác đầu vào sang trái 11 vị trí.

2. Phép biến đổi qua tầng hộp thế

Như đã nói ở trên, các hộp thế (S-hộp) trong thuật toán GOST 28147-89 được giữ bí mật. Đây cũng chính là thành phần phi tuyến duy nhất trong mã khối. Trong phạm vi của nghiên cứu này, chúng tôi sử dụng bộ hộp thế được sử dụng trong hệ mật của Ngân hàng Trung ương Nga để phân tích [2]. Đối với mỗi hộp thế ta cần tính bảng phân bố vi sai DDT của chúng. Nghĩa là cần xác định xác suất vi sai đối với mỗi cặp sai khác đầu vào ΔA và đầu ra ΔC bất kỳ. Xác suất này được xác định bởi:

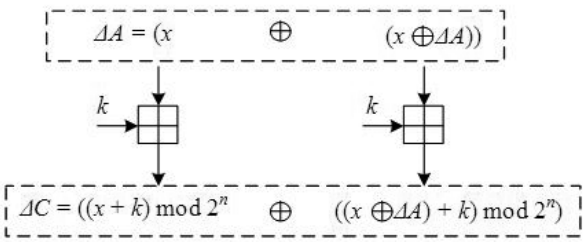
$$p(\Delta A, \Delta C) = \frac{\#\{x : S(x) \oplus S(x \oplus \Delta A) = \Delta C\}}{16} \quad (1)$$

Sau khi tính giá trị cho toàn bộ 8 bảng phân bố vi sai chúng tôi thấy có những quy luật sau:

- Tổng của tất cả các giá trị trong một dòng, tức là số lượng các giá trị khác nhau của sai khác đầu ra ΔC tương ứng với một sai khác đầu vào ΔA, luôn bằng 2^4.
- Đối với sai khác đầu vào ΔA khác 0, không có hộp thế nào cho sai khác đầu ra $\Delta C = 0$
- Trong bộ hộp thế sử dụng trong hệ mật của ngân hàng Trung ương Nga, hộp thế S_7 và S_8 là yếu nhất.

3. Phép cộng với khóa vòng theo modulo 2^{32}

Phương pháp thám mã vi sai dựa trên cơ sở theo dõi theo sự thay đổi giá trị sai khác của hai thông báo. Để xác định được sai khác này thường sử dụng phép toán cộng theo modulo 2 (hay còn gọi là phép XOR). Cũng chính bởi tính chất của phép cộng theo modulo 2 này mà trong thuật toán mã hóa DES, khi xem xét sai khác thông qua hàm vòng F, giá trị khóa không được tính đến, đó là do hai giá trị như nhau sẽ cho kết quả là bit 0 theo tính chất của phép cộng modulo 2. Khác với DES, trong thuật toán GOST 28147-89, phép cộng với khóa được thực hiện theo modulo 2^{32}. Do đó cần phải nghiên cứu ảnh hưởng của phép cộng này tới việc tạo ra các sai khác. Thực hiện vét cạn trên toàn bộ các số 32 bit là không thể. Trước tiên chúng tôi tiến hành phân tích trên các số 2, 3,… chúng được cộng với nhau tương ứng theo các modulo 2^2, 2^3. Kết quả của phân tích được thể hiện trong các Bảng 1 và Bảng 2 tương ứng. Quá trình tính bảng phân bố xác suất vi sai của phép cộng modulo 2^n được minh họa như trong Hình 1, ở đây ký hiệu ΔA là sai khác đầu vào, ΔC là sai khác đầu ra, k là khóa bí mật và p là bản rõ nào đấy, trong đó $\Delta A, \Delta C, k, p \in V_n$ và phép "+" là ký hiệu phép cộng với khóa theo modulo 2^n.



Hình 1. Minh họa tính sai khác đầu ra của phép cộng modulo

BẢNG 1. PHÂN TÍCH PHÉP CỘNG THEO MODULO 2^2

	0	**1**	**2**	**3**
0	16	0	0	0
1	0	8	0	8
2	0	0	16	0
3	0	8	0	8

BẢNG 2. PHÂN TÍCH PHÉP CỘNG THEO MODULO 2^3

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	64	0	0	0	0	0	0	0
1	0	32	0	16	0	0	0	16
2	0	0	32	0	0	0	32	0
3	0	16	0	16	0	16	0	16
4	0	0	0	0	64	0	0	0
5	0	0	0	16	0	32	0	16
6	0	0	32	0	0	0	32	0
7	0	16	0	16	0	16	0	16

Quan sát các bảng ta thấy chúng có cấu trúc giống nhau. Ví dụ, một phần tư phía trên bên trái và một phần tư phía dưới bên phải của bảng 2 lặp lại cấu trúc của bảng 1. Sự lặp lại tương tự có thể

quan sát thấy khi so sánh bảng phân bố đối với những giá trị n lớn hơn (ví dụ n = 4, 5…).

Việc phân tích các bảng nhận được cho phép xác định xác xuất để sai khác không bị thay đổi khi thực hiện phép toán cộng theo modulo. Bởi vì tất cả những kết luận trong quá trình khảo sát không chỉ đúng với phép cộng các số theo modulo 2^{32} mà nó còn đúng với giá trị 2^n bất kỳ. Như vậy với phép cộng $(a + b) \bmod 2^n$ các tác giả trong [2] đưa ra những kết luận sau:

- Mỗi giá trị sai khác đầu vào ánh xạ về chính nó với xác suất xác định bởi:

Nếu sai khác đầu vào $\Delta A < 2^{n-1}$, khi đó

$$p = 2^{-t} \quad (2)$$

Nếu $\Delta A \geq 2^{n-1}$, thì $p = 2^{-(t-1)}$ (3)

Ở đây t là số bit khác 0 trong sai khác đầu vào ΔA.

- Nếu $\Delta A = 0$, thì $\Delta C = 0$ với với xác xuất $p = 1$.
- Nếu $\Delta A = 2^{n-1}$, thì $\Delta C = 2^{n-1}$ với xác xuất $p = 1$.
- $p(\Delta A \to \Delta A) \geq p(\Delta A \to \Delta C),\ \forall \Delta C \neq \Delta A$

Những tổng kết ở trên là rất có ý nghĩa trong những phân tích tiếp theo, bởi vì đầu tiên cần biết được xác suất mà ở đó giá trị sai khác không bị thay đổi dưới tác động của một nguyên thủy mật mã.

Trong luận án của Isukova [2] tác giả không chứng minh các công thức xác suất (2) và (3) mà chỉ diễn giải những kết luận rút ra từ việc phân tích những bảng phân bố xác suất nói trên. Để giải thích rõ hơn công thức này, chúng tôi sẽ chứng minh bằng lý thuyết tổng quát cho hai công thức (2) và (3).

Xét phép cộng khóa theo modulo 2^n. Ký hiệu $x = (x_n \cdots x_1 x_0), k = (k_n \cdots k_1 k_0)$ và $\Delta A = (d_n \cdots d_1 d_0)$, trong đó $x_i, k_i, a_i \in \{0,1\}$. Ta cần xác định xác suất để sai khác đầu vào và đầu ra bằng nhau qua phép cộng khóa theo modulo, nghĩa là ta phải có

$$\Delta A = (x + k) \oplus ((x \oplus \Delta A) + k), \quad (4)$$

ở đây phép "+" là ký hiệu phép cộng theo modulo 2^n. Và xác suất để thỏa mãn (4) được xác định bởi biểu thức $p = 2^{-t}$, nếu $\Delta A < 2^{n-1}$ và $p = 2^{-(t-1)}$, nếu $\Delta A \geq 2^{n-1}$. Chúng tôi sẽ làm rõ hơn bằng phân tích lý thuyết cho hai công thức này.

Từ (4) ta có:

$$(x + k) \oplus \Delta A = (x \oplus \Delta A) + k. \quad (5)$$

Ta cần phải xác định được số cặp x và k thỏa mãn (5), có nghĩa là xác định được xác suất tương ứng với giá trị sai khác đầu vào ΔA. Để thỏa mãn (5) trước tiên ta phải có:

$$x_0 \oplus k_0 \oplus d_0 = (x_0 \oplus d_0) \oplus k_0. \quad (6)$$

Biểu thức (6) xảy ra với mọi cặp bit (x_0, k_0). Tiếp theo ta xét đến bit thứ 2 trong phương trình (5). Để thỏa mãn ta phải có:

$$x_1 \oplus k_1 \oplus d_1 \oplus c_1 = (x_1 \oplus d_1) \oplus k_1 \oplus c_1' \Leftrightarrow c_1 = c_1', \quad (7)$$

trong đó c_1 và c_1' là bit nhớ tương ứng ở vế trái và vế phải của (5) khi cộng theo modulo với khóa của bit thứ nhất. Từ đó ta có

$$c_1 = c_1' \Leftrightarrow x_0 k_0 = (x_0 \oplus d_0) k_0 \Leftrightarrow d_0 k_0 = 0. \quad (8)$$

Trong (8), nếu $d_0 = 0$, thì (8) sẽ thỏa mãn với mọi cặp (x_0, k_0) và số lượng của chúng bằng $\#(x_0, k_0) = 4$. Còn nếu $d_0 = 1$, thì chỉ có 2 cặp (x_0, k_0) thỏa mãn, có nghĩa là $\#(x_0, k_0) = 2$.

Xét sang bit thứ 3, từ (5) ta có:

$$x_2 \oplus k_2 \oplus d_2 \oplus c_2 = (x_2 \oplus d_2) \oplus k_2 \oplus c_2' \Leftrightarrow c_2 = c_2'. \quad (9)$$

Từ đây, ta có

$$\begin{aligned} & c_2 = c_2' \\ & \Leftrightarrow x_1 k_1 \oplus (x_1 \oplus k_1) c_1 = (x_1 \oplus d_1) k_1 \oplus (x_1 \oplus d_1 \oplus k_1) c_1' \\ & \Leftrightarrow d_1 (k_1 \oplus c_1) = 0 \end{aligned} \quad (10)$$

Trong biểu thức (10) thấy rằng, nếu $d_1 = 0$, thì (10) sẽ thỏa mãn với mọi cặp (x_1, k_1) và số lượng của chúng bằng $\#(x_1, k_1) = 4$. Còn khi $d_1 = 1$, khi đó $k_1 = c_1$. Như vậy $\#(x_1, k_1) = 2$ cả trong hai trường hợp có nhớ $(c_1 = 1)$ hoặc không có nhớ $(c_1 = 0)$.

Lập luận tương tự ta cũng sẽ nhận được kết quả cho các cặp bit từ (x_3, k_3) cho đến (x_{n-2}, k_{n-2}). Điều đó có nghĩa rằng:

$$\begin{cases} \#(x_j, k_j) = 4, & d_j = 0 \\ \#(x_j, k_j) = 2, & d_j = 1 \end{cases}, 1 \leq j \leq n-2. \quad (11)$$

Xét bit thứ n có phương trình:

$$\begin{aligned} & x_{n-1} \oplus k_{n-1} \oplus d_{n-1} \oplus c_{n-1} = (x_{n-1} \oplus d_{n-1}) \oplus k_{n-1} \oplus c_{n-1}' \Leftrightarrow \\ & \Leftrightarrow c_{n-1} = c_{n-1}' \end{aligned} \quad (12)$$

Trong đó, c_{n-1} và c'_{n-1} bit nhớ khi thực hiện phép cộng có nhớ giữa các bit x_{n-2}, k_{n-2} và d_{n-2}. Ta thấy rằng để thỏa mãn (4) thì phải đảm bảo tất cả $c_j = c'_j$, với mọi $1 \le j \le n-1$. Như vậy (12) luôn luôn thỏa mãn, có nghĩa rằng lực lượng cặp bit của khóa và thông báo bằng $\#(x_{n-1}, k_{n-1}) = 4$ mà không phụ thuộc vào việc d_{n-1} bằng 0 hay bằng 1.

Như vậy nếu gọi $t = wt(\Delta A)$, khi đó nếu $d_{n-1} = 0$ thì số lượng cặp bit khóa và thông báo thỏa mãn (5) sẽ là:

$$\#(x_*, k_*) = 4 \times \left(\underbrace{2 \times \cdots \times 2}_{t} \right) \times \left(\underbrace{4 \times \cdots \times 4}_{n-1-t} \right) = .$$

$$= 4 \times 2^t \times 4^{n-1-t} = 2^{2n-t}$$

Do đó, xác suất vi sai trong trường hợp này là

$$p = \frac{\#(x_*, k_*)}{4^n} = \frac{2^{2n-t}}{2^{2n}} = \frac{1}{2^t}.$$

Ngược lại, nếu $d_{n-1} = 1$ thì số lượng cặp bit khóa và thông báo thỏa mãn (5) sẽ là

$$\#(x_*, k_*) = 4 \times \left(\underbrace{2 \times \cdots \times 2}_{t-1} \right) \times \left(\underbrace{4 \times \cdots \times 4}_{n-t} \right) = .$$

$$= 4 \times 2^{t-1} \times 4^{n-t} = 2^{2n-t+1}$$

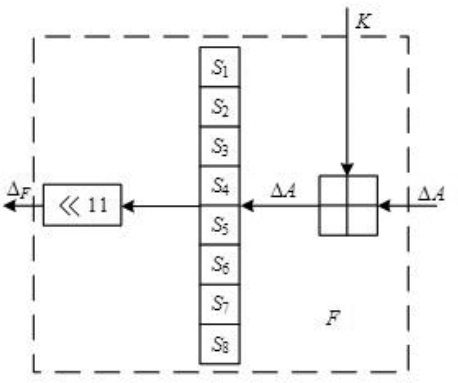
Tương ứng, xác suất vi sai trong trường hợp này xác định bởi công thức $p = \frac{\#(x_*, k_*)}{4^n} = \frac{2^{2n-t+1}}{2^{2n}} = \frac{1}{2^{t-1}}$. Như vậy chúng tôi đã giải thích hoàn toàn bằng lý thuyết cho biểu thức xác suất theo công thức (2) và (3).

III. TÌM KIẾM ĐẶC TRƯNG VI SAI HIỆU QUẢ CHO GOST 28147-89

A. Phân tích tiếp cận tìm kiếm đặc trưng vi sai hiệu quả cho GOST 28147-89

Để phân tích thuật toán GOST chúng tôi xem xét một vài giá trị sai khác đầu vào, sau đó tính toán xác suất thay đổi của sai khác này qua các vòng mã hóa khác nhau. Minh họa các biến đổi tác động lên sai khác trong hàm vòng được cho trong Hình 2. Nguyên tắc lựa chọn những sai khác này như sau:

- Sai khác đầu vào ΔA cần phải lựa chọn sao cho có ít nhất có thể các bó chủ động 4 bit đối với các hộp thế, bởi vì số hộp thế chủ động càng ít thì xác xuất vi sai ở mỗi vòng càng cao.
- Sai khác đầu vào ΔA cần chứa ít nhất có thể các bit khác 0, bởi vì theo công thức (2) và (3) xác xuất để sai khác đầu ra không bị thay đổi dưới tác động của phép cộng khóa theo modulo 2^{32} sẽ giảm khi tăng số bit khác 0 trong sai khác đang xét.
- Khi đi qua các hộp thế, sai khác đầu vào ΔA cần phải cho sai khác đầu ra ΔC chứa số lượng ít nhất có thể các bit khác 0 (lý tưởng là 1). Cần lựa chọn giá trị đầu ra sao cho có thể nhận được xác suất lớn nhất có thể. Với một đầu vào cố định mà cho 2 sai khác đầu ra có cùng trọng số Hamming và có cùng xác suất, thì ưu tiên lựa chọn sai khác mà bit thấp nhất của nó bằng 0.



Hình 2. Hàm vòng *F* của GOST 28147-89

Với một giá trị đầu vào cố định, việc lựa chọn giá trị sai khác đầu ra với xác suất lớn nhất khi qua tầng hộp thế và xác suất lớn nhất khi qua tầng cộng khóa sẽ cho xác suất lớn nhất qua mỗi vòng mã hóa. Tuy nhiên, việc này không đảm bảo xác suất lớn nhất khi tính toán sai khác đầu ra qua nhiều vòng mã hóa liên tiếp. Việc tối ưu xác suất qua từng vòng cục bộ sẽ cho giá trị xác suất qua các vòng gần hơn với cận của xác suất lớn nhất (tương ứng với đặc trưng vi sai tốt nhất). Chính vì thế, chúng tôi gọi đặc trưng vi sai nhân được trong phân tích ở đây là cặp đặc trưng vi sai hiệu quả. Với số vòng nhỏ chúng tôi có thể chỉ ra được giá trị xác suất tương ứng đặc trưng vi sai tốt nhất như trong Khẳng định 1 và Khẳng định 2 dưới đây.

B. Đặc trưng vi sai tốt nhất cho số vòng rút gọn của GOST 28147-89

Trong mục này, dựa trên cơ sở ý tưởng phân tích trong nghiên cứu của Isukova [2], chúng tôi sẽ xác định giá trị xác suất vi sai tốt nhất cho số vòng rút gọn của GOST 28147-89, cụ thể là đối với 3 và 4 vòng.

Khẳng định 1. *Xác suất vi sai lớn nhất cho 3 vòng mã hóa của GOST 28147-89 là p, với p là xác suất vi sai lớn nhất của vòng mã đầu tiên.*

Chứng minh. Như phân tích phép cộng khóa trong mục 3 ở trên, xác suất vi sai mà ở đó sai khác

đầu vào không bị thay đổi qua phép cộng khóa là lớn nhất, có nghĩa là:

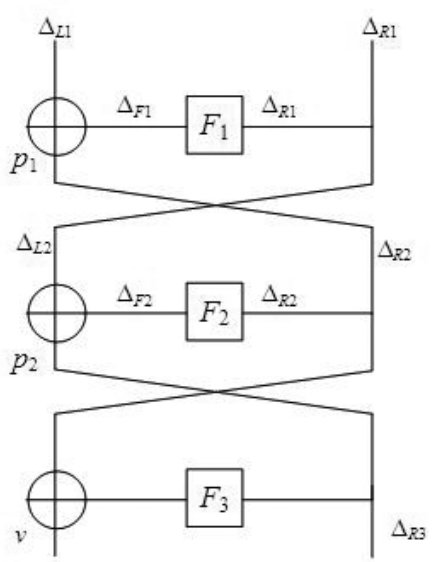
$$p(\Delta A \to \Delta A) \geq p(\Delta A \to \Delta C),\ \forall \Delta C \neq \Delta A,$$

trong đó $\Delta A, \Delta C$ tương ứng là sai khác đầu vào và đầu ra sau phép cộng khóa theo modulo 2^n. Như vậy sai khác đầu vào của tầng cộng khóa cũng chính là sai khác đầu vào của S-hộp. Mặt khác, theo công thức (2) hoặc (3), giá trị xác suất vi sai giảm khi trọng số hamming của sai khác đầu vào tăng. Hơn nữa, xác suất qua tầng cộng khóa cũng giảm khi số lượng các S-hộp chủ động tạo lên bởi sai khác tăng. Bằng phân tích lý thuyết và thực nghiệm chúng tôi chứng tỏ được rằng giá trị xác suất vi sai lớn nhất của vòng mã hóa đầu tiên là bằng $p = 2^{-2.41}$. Để chứng tỏ giá trị xác suất này cũng chính bằng giá trị xác suất vi sai lớn nhất cho 3 vòng mã hóa của GOST 38147-89, ta xét sơ đồ minh họa trên Hình 3.

Ta có $p_1 = p$ là giá trị xác suất vi sai lớn nhất của vòng 1. Xét trường hợp nếu $\Delta_{L1} \neq \Delta_{F1}$ khi đó $\Delta_{R2} = \Delta_{L1} \oplus \Delta_{F1} \neq 0$, có nghĩa rằng xác suất vi sai của vòng thứ hai $p_2 < 1$. Trong trường hợp ngược lại, nếu $\Delta_{L1} = \Delta_{F1}$, khi đó $\Delta_{R2} = \Delta_{L1} \oplus \Delta_{F1} = 0$, do đó $p_2 = 1$. Như vậy

$$p_{3r(\Delta_{L1} \neq \Delta_{F1})} = p \cdot p_2 < p_{3r(\Delta_{L1} = \Delta_{F1})} = p \cdot p_2 = p \cdot 1 = p.$$

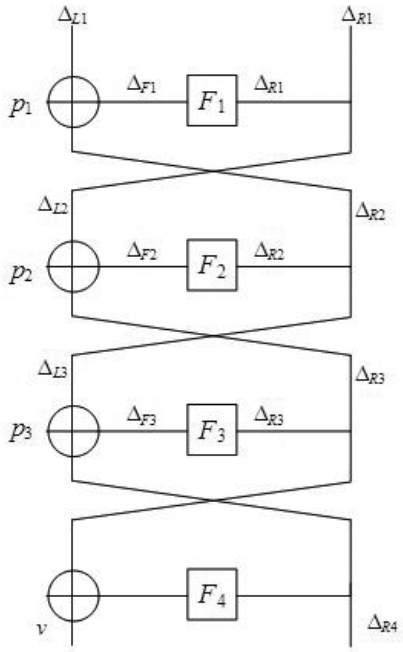
□



Hình 3. Minh họa biến đổi qua 3 vòng của GOST 28147-89

Khẳng định trên chứng tỏ một điều xác suất vi sai lớn nhất cho 3 vòng của GOST 28147-89 chính bằng *p* trong trường hợp khi mà nửa bên trái của sai khác đầu vào bằng giá trị đầu ra của hàm vòng ở vòng thứ nhất. □

Khẳng định 2. *Xác suất vi sai lớn nhất cho 4 vòng mã hóa của GOST 28147-89 là p^2, với p là xác suất vi sai lớn nhất của vòng mã đầu tiên.*



Hình 4. Minh họa biến đổi qua 4 vòng của GOST 28147-89

Chứng minh. Xét sơ đồ minh họa biến đổi qua 4 vòng mã hóa của GOST 28147-89 (như biểu diễn trên Hình 4). Bằng cách lập luận tương tự như trong chứng minh Khẳng định 1 khi xét hai trường hợp $\Delta_{L1} \neq \Delta_{F1}$ và $\Delta_{L1} = \Delta_{F1}$, chúng ta sẽ xác định được xác suất vi sai lớn nhất qua 4 vòng mã hóa của GOST 28147-89 là p^2. □

C. Một thuật toán tìm kiếm đặc trưng vi sai hiệu quả cho toàn bộ số vòng của GOST 28147-89

Trên cơ sở những phân tích ở trên trong mục này chúng tôi xây dựng một thuật toán thực hành được cho phép tìm kiếm đặc trưng vi sai tốt nhất cho 2, 3 và 4 vòng và đặc trưng vi sai hiệu quả cho toàn bộ số vòng mã hóa còn lại của GOST 28147-89. Trong thuật toán, ký hiệu "dòng *i*" là chỉ dòng thứ *i* của thuật toán mô tả được cho dưới dạng giả mã (pseudo-code).

Hàm *n_Round*(*L_in*[], *R_in*[], *L_out*[], *R_out*[], *max_prob_round*[]) xác định cặp sai khác đầu vào, đầu ra có xác suất vi sai lớn nhất. Sai khác đầu vào cho *n* vòng ($2 \leq n \leq 32$) được lưu trong hai mảng *L_in*[], *R_in*[] tương ứng với hai nửa trái, phải. Sai khác đầu ra sau *n* vòng được lưu trong hai mảng tương ứng *L_out*[], *R_out*[], còn xác suất vi sai cực đại tương ứng được lưu trong mảng *max_prob_round*[].

Thuật toán sẽ xác định toàn bộ sai khác có thể tương ứng với toàn bộ $2^{32} - 1$ sai khác đầu vào bên

phải (dòng 4), trong đó nửa bên trái của vòng đầu tiên được xác định đúng bằng đầu ra của hàm vòng *F* (dòng 9, 10, 13). Xác suất sau *i* vòng được xác định bởi biểu thức 2^{prob_round}, trong đó giá trị *prob_round* được xác định ở dòng 8.

Để duyệt toàn bộ cho nửa bên phải của sai khác đầu vào cần $2^n - 1$ phép kiểm tra, trong đó *n* là kích thước theo bit của nửa bên phải. Tính xác suất sau *r* vòng cần độ phức tạp *rO*(.), trong đó *O*(.) là độ phức tạp xác định xác suất vi sai cho một vòng mã hóa. Như vậy, độ phức tạp của thuật toán mô tả là $(2^n - 1)rO(.)$. Với $n = 32$, $r = 32$, giá trị độ phức tạp xấp xỉ bằng $2^{37}O(.)$.

Thuật toán xác định đặc trưng vi sai hiệu quả cho số vòng bất kỳ của chuẩn GOST 28147-89:

void *n_Round*(*L_in*[], *R_in*[], *L_out*[], *R_out*[], *max_prob_round*[])

1. **for all** $i \in [0,30]$ **do**
2. *max_prob_round*[*i*] ← –500
3. **end for**
4. **for all** $x \in \left[0, 2^{32} - 1\right]$ **do**
5. *prob_round* ← 0
6. *R_in_i* ← *x*
7. **for all** $i \in [0,30]$ **do**
8. *prob_round* ← *prob_round* + *Round_i*(*i* + 1, *L_in_i*, *R_in_i*, *L_out_i*, *R_out_i*, *diff_out_F*)
9. **if** (*i* = 0) **then**
10. *temp_sbox* ← *diff_out_F*
11. **end if**
12. **if** (*max_prob_round*[*i*] < *prob_round*) **then**
13. *L_in*[*i*] ← *temp_sbox*
14. *R_in*[*i*] ← *x*
15. *L_out*[*i*] ← *L_out_i*
16. *R_out*[*i*] ← *R_out_i*
17. *max_prob_round*[*i*] ← *prob_round*
18. **end if**
19. *L_in_i* ← *L_out_i*
20. *R_in_i* ← *R_out_i*
21. **end for**
22. **end for**

Hàm *Round_i*(*i* + 1, *L_in_i*, *R_in_i*, *L_out_i*, *R_out_i*, *diff_out_F*) (dòng 8) nhận 3 tham số đầu vào là *i* + 1, hai nửa sai khác đầu vào *L_in_i* và *R_in_i*, hàm trả về 4 giá trị là hai nửa sai khác đầu ra *L_out_i* và *R_out_i*, đầu ra của hàm vòng *F* là *diff_out_F* và xác suất vi sai tương ứng. Cụ thể, hàm sẽ thực hiện việc xác định xác suất vi sai cực đại cho một vòng mã hóa tương ứng với sai khác đầu vào $\Delta_{in\_i} = L_{in\_i} \| R_{in\_i}$. Nếu là vòng đầu tiên (tương ứng với *i* = 1) thì nửa trái của sai khác đầu vào được xác định bằng đầu ra của hàm vòng *F* ở vòng này (giá trị này lưu trong biến *diff_out_F*). Giả mã cho hàm *Round_i*(.) được mô tả như sau:

double *Round_i*(*i* + 1, *L_in_i*, *R_in_i*, *L_out_i*, *R_out_i*, *diff_out_F*)

1. *prob* ← 0
2. *diff_out_F* ← 0
3. **if** (*R_in* ≠ 0) **then**
4. **if** ((*R_in* >> 31)&0x1 = 0) **then**
5. $prob \leftarrow -wt[R\_in\_i]$
6. **else**
7. $prob \leftarrow -\left(wt[R\_in\_i] - 1\right)$
8. **end if**
9. **for all** $i \in [0,\ldots,7]$ **do**
10. $prob \leftarrow prob + \max_{1 \le j \le 2^4 - 1} DDT[i]\left[\left(R\_in\_i \gg 4 \cdot (7 - i)\right) \,\&\, 0xF\right][j]$
11. $diff\_out\_F \leftarrow diff\_out\_F \gg 4$
12. $diff\_out\_F = diff\_out\_F \oplus j_{max}$
13. **end for**
14. $diff\_out\_F = \left(diff\_out\_F \gg 21\right) \oplus \oplus \left(diff\_out\_F \gg 11\right)$
15. **end if**
16. *L_out_i* ← *R_in_i*
17. **if** ((*i* + 1) = 1) **then**
18. *R_out_i* ← 0
19. **else** $R\_out\_i \leftarrow diff\_out\_F \oplus L\_in\_i$
20. **end if**
21. **if** (*R_in_i* = 0) **then** *prob* ← 0
22. **return** *prob*

Trong mô tả hàm *Round_i*(.) nói trên, dòng 4, 5, 6 và 7 là xác định xác suất qua phép cộng khóa, chính là giá trị xác suất mà ở đó sai khác đầu ở hàm vòng *F* vào không bị thay đổi dưới tác động của phép cộng khóa theo modulo 2^{32}.

Dòng 9 và 10 xác định xác suất vi sai cực đại qua tầng S-hộp. Ở đây 8 bảng phân bố vi sai *DDT* được tính trước, mỗi bảng có kích thước 16x16 tương ứng với mỗi S-hộp 4 bit. Độ phức tạp của việc xác định giá trị xác suất vi sai qua tầng S-hộp này là $O(.) = 2^N \times \left(2^m - 1\right)$, trong đó *N* là số lượng S-hộp, còn *m* là kích thước (bit) S-hộp. Trên thực tế với mỗi một hộp thế sau khi tính bảng phân bố vi sai tương ứng cho nó, chúng ta luôn luôn có thể

xác định được một vài giá trị sai khác đầu ra tương ứng với một sai khác đầu vào cố định mà ở đó xác suất vi sai là cực đại. Trong những sai khác đầu ra nhận được này chúng tôi chỉ quan tâm đến những đầu ra mà ở đó trọng số Hamming của nó là nhỏ nhất. Như vậy, từ bảng phân bố vi sai DDT ta sẽ lập được hai mảng các giá trị đầu ra tương ứng với từng đầu vào và bảng xác suất tương ứng. Phân tích này có nghĩa rằng không cần phải duyệt toàn bộ các giá trị sai khác đầu ra ở dòng 10 của giả mã nói trên để tìm giá trị cực đại. Kết quả là độ phức tạp để tính xác suất khi qua tầng hộp thế sẽ chỉ là $O(.) = 2^N$ phép truy cập vào địa chỉ mảng. Như vậy độ phức tạp để xác định xác suất vi sai tốt nhất cho r vòng mã hóa là $(2^n - 1)rO(.) = (2^n - 1)r2^N = 2^N(2^n - 1)r$. Với $n = 32$, $N = 8$ và $r = 32$. Độ phức tạp để xác định đặc trưng vi sai tốt nhất xấp xỉ 2^{41} phép toán cơ bản. Bộ nhớ yêu cầu để lưu giá trị xác suất sau mỗi vòng và giá trị sai khác đầu vào/ra tương ứng là không đáng kể.

IV. KẾT QUẢ THỰC NGHIỆM

Chúng tôi tiến hành cài đặt thuật toán tìm kiếm đặc trưng vi sai tốt nhất theo như phân tích ở trên. Kết quả chạy đối với 32 vòng mã của GOST 28147-89 được cho trong bảng 3, ở đây r là số thứ tự vòng mã, còn #Δ là số cặp có cùng giá trị xác suất trong vòng thứ r. Trong bảng thống kê, những đặc trưng cho 2, 3 và 4 vòng là những đặc trưng tốt nhất tương ứng với xác suất vi sai của chúng. Còn lại từ 5 đến 32 vòng là giá trị xác suất cho những đặc trưng vi sai hiệu quả cho GOST 28147-89. Trong bảng thống kê, chúng tôi cũng xác định số lượng các cặp sai khác mà ở đó xác suất vi sai qua một số lượng vòng mã hóa là bằng nhau.

BẢNG 3. ĐẶC TRƯNG VI SAI TỐT NHẤT CHO 2, 3 VÀ 4 VÒNG VÀ ĐẶC TRƯNG VI SAI HIỆU QUẢ CHO SỐ VÒNG CÒN LẠI CỦA GOST 28147-89

r	#Δ	L_{in}	R_{in}	L_{out}	R_{out}	p
2	1	00006800	00000001	v	00000000	$2^{-2.4}$
	2	00006800	00000100	v	00000000	
	3	00280000	00000800	v	00000000	
	4	00700000	00010000	v	00000000	
	5	50000000	00080000	v	00000000	
	6	60000000	00400000	v	00000000	
3	1	00006800	00000001	v	00000001	$2^{-2.4}$
	2	00280000	00000100	v	00000100	
	3	00700000	00000800	v	00000800	
	4	50000000	00010000	v	00010000	
	5	60000000	00080000	v	00080000	
	6	80000007	00400000	v	00400000	
4	1	00006800	00000001	v	00006800	$2^{-4.8}$
	2	00280000	00000100	v	00280000	
	3	00700000	00000800	v	00700000	
	4	50000000	00010000	v	50000000	
	5	60000000	00080000	v	60000000	
	6	80000007	00400000	v	80000007	
5	1	60000000	00080000	v	00080200	$2^{-8.2}$
6	1	50000000	00010000	v	00280000	$2^{-13.6}$
	2	60000000	00080000	v	00300000	
7	1	60000000	00080000	v	80080204	$2^{-17.1}$
8	1	00020000	00000020	v	80020004	$2^{-27.2}$
9	1	00020000	00000020	v	003020A0	$2^{-36.4}$
	2	00000040	04000000	v	74000010	
10	1	00020000	00000020	v	00838000	$2^{-46.2}$
11	1	00000050	02000000	v	E2028100	$2^{-51.1}$
	2	00000050	08000000	v	E8028100	
12	1	00000040	04000000	v	80060044	$2^{-65.8}$
13	1	E0010000	002800D0	v	50000050	$2^{-69.9}$
	2	E0010000	002800E0	v	50000060	
14	1	E0010000	002800D0	v	00000100	$2^{-77.9}$
	2	E0010000	002800E0	v	00000100	
15	1	E0010000	002800D0	v	50280050	$2^{-80.3}$
	2	E0010000	002800E0	v	50280060	
16	1	E0010000	002800D0	v	E0010000	$2^{-94.7}$
	2	E0010000	002800E0	v	E0010000	
17	1	00010000	00000050	v	00000050	$2^{-102.1}$
	2	00010000	00000060	v	00000060	
	3	E0010000	002800D0	v	002800D0	
	4	E0010000	002800E0	v	002800E0	
18	1	00010000	00000050	v	00000000	$2^{-106.1}$
	2	00010000	00000060	v	00000000	
19	1	00010000	00000050	v	00000050	$2^{-106.1}$
	2	00010000	00000060	v	00000060	
20	1	00010000	00000050	v	00010000	$2^{-110.1}$
	2	00010000	00000060	v	00010000	
21	1	00010000	00000050	v	50000050	$2^{-112.5}$
	2	00010000	00000060	v	50000060	
22	1	00010000	00000050	v	00000100	$2^{-120.5}$
	2	00010000	00000060	v	00000100	
23	1	00010000	00000050	v	50280050	$2^{-122.9}$
	2	00010000	00000060	v	50280060	

24	1	00010000	00000050	*v*	E0010000	$2^{-137.3}$
	2	00010000	00000060	*v*	E0010000	
25	1	00010000	00000050	*v*	002800D0	$2^{-144.8}$
	2	00010000	00000060	*v*	002800E0	
26	1	00010000	00000050	*v*	00000000	$2^{-157.2}$
	2	00010000	00000060	*v*	00000000	
27	1	00010000	00000050	*v*	002800D0	$2^{-157.2}$
	2	00010000	00000060	*v*	002800E0	
28	1	00010000	00000050	*v*	E0010000	$2^{-169.6}$
	2	00010000	00000060	*v*	E0010000	
	3	E0010000	002800D0	*v*	00010000	
	4	E0010000	002800E0	*v*	00010000	
29	1	E0010000	002800D0	*v*	50000050	$2^{-172.1}$
	2	E0010000	002800E0	*v*	50000060	
30	1	E0010000	002800D0	*v*	00000100	$2^{-180.1}$
	2	E0010000	002800E0	*v*	00000100	
31	1	E0010000	002800D0	*v*	50280050	$2^{-182.4}$
	2	E0010000	002800E0	*v*	50280060	
32	1	E0010000	002800D0	*v*	E0010000	$2^{-196.8}$
	2	E0010000	002800E0	*v*	E0010000	

V. KẾT LUẬN

Trong bài báo này chúng tôi đã trình bày một kết quả cho phép đánh giá độ an toàn của GOST 28147-89 chống lại thám mã vi sai. Về mặt lý thuyết, làm tường minh thêm công thức (2) và (3). Việc giải thích được những công thức này có vai trò quan trọng cho thám mã vi sai đối với những mã pháp tựa GOST 28147-89 khi sử dụng phép cộng khóa theo modulo 2^{32}. Ngoài ra cũng chứng minh bằng lý thuyết xác suất vi sai tốt nhất cho số vòng rút gọn của GOST 28147-89. Về mặt thực nghiệm, chúng tôi đã xây dựng thuật toán cho phép xác định đặc trưng vi sai tốt nhất cho số vòng rút gọn và đặc trưng vi sai hiệu quả cho toàn bộ số vòng còn lại của GOST 28147-89. Đặc biệt, chúng tôi cũng xác định được số lượng các cặp đặc trưng có cùng xác suất cho mỗi vòng của thuật toán này. Những kết quả đạt được của nghiên cứu này khẳng định độ an toàn của GOST 28147-89 chống lại thám mã vi sai. Hướng nghiên cứu tiếp theo đặt ra là tìm xác định đặc trưng vi sai tốt nhất cho số lượng vòng lớn hơn bốn nhằm đánh giá chính xác độ an toàn của GOST 28147-89.

TÀI LIỆU THAM KHẢO

[1]. Biham, E. and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems". Journal of CRYPTOLOGY, 1991. 4(1), pp. 3-72.

[2]. Isukova E. A. "Construct and Evaluate an Algorithm for Estimating the Security of Block ciphers by Differential Cryptanalysis". Thesis, Taganrov tecnichcal institute of south federation universty. 2007, pp. 207.

[3]. Shneier B. "Applied Cryptography: Protocols, Algorithms and Open sources in C" – M.: TRIUMF, 2002.

[4]. Babenko L. K., Isukova E. A.. "Modern Block ciphers and Cryptanalysis – Matxcova", «Gelios ARB», 2006.

[5]. Trmora A. L., "Modern Applied Cryptography". 2th edit., - M.: Gelios ARB, 2002.

[6]. Nicolas T. Courtois, Theodosis Mourouzis, Michal Misztal, Jean-Jacques Quisquater, Guangyan Song: "Can GOST Be Made Secure Against Differential Cryptanalysis?" Cryptologia 39(2), pp. 145-156 (2015).

SƠ LƯỢC VỀ TÁC GIẢ

TS. Nguyễn Văn Long

Đơn vị công tác: Viện Khoa học – Công nghệ Mật mã, Ban Cơ yếu Chính phủ.

E-mail: nvlong.bcy@gmail.com.

Tốt nghiệp chuyên nghành An toàn thông tin các Hệ thống viễn thông, năm 2008 và nhận bằng Tiến sĩ chuyên ngành Các phương pháp bảo vệ thông tin, Học viện FSO, Liên bang Nga năm 2015.

Hướng nghiên cứu hiện nay: Khoa học mật mã, mã hóa đối xứng

ThS. Trần Hồng Thái

Đơn vị công tác: Viện Khoa học – Công nghệ Mật mã, Ban Cơ yếu Chính phủ.

E-mail: ththai@bcy.gov.vn.

Nhận bằng Kỹ sư năm 2000 và Thạc sĩ năm 2007 chuyên ngành Kỹ thuật mật mã, Học viện Kỹ thuật Mật mã.

Hướng nghiên cứu hiện nay: Nghiên cứu đánh giá độ an toàn của mã khối và hàm băm mật mã.

ThS. Nguyễn Bùi Cương

Đơn vị công tác: Viện Khoa học - Công nghệ Mật mã, Ban Cơ yếu Chính phủ, Hà Nội.

E-mail: nguyenbuicuong@gmail.com

Tốt nghiệp chuyên ngành Toán học, Đại học Sư phạm Hà Nội - Đại học Quốc gia Hà Nội năm 2004. Tốt nghiệp Thạc sĩ Toán học, Đại học Khoa học Tự nhiên - Đại học Quốc gia Hà Nội năm 2008.

Hướng nghiên cứu hiện nay: Khoa học mật mã, mã hóa đối xứng.