TCVN **TIÊU CHUẨN QUỐC GIA**

TCVN 12854-2 :2020

ISO/IEC 29192-2: 2012

Xuất bản lần 1

CÔNG NGHỆ THÔNG TIN – CÁC KỸ THUẬT AN TOÀN – MẬT MÃ HẠNG NHẸ – PHẦN 2: MÃ KHỐI

Information technology – Security techniques – Lightweight cryptography – Part 2: Block ciphers

HÀ NỘI – 2020

Mục Lục

Lời nói đầu

TCVN 12854-2 : 2020 hoàn toàn tương đương với ISO/IEC 29192-2:2012.

TCVN 12854-2 : 2020 do Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã biên soạn, Ban Cơ yếu Chính phủ đề nghị, Tổng cục Tiêu chuẩn Đo lường Chất lượng thẩm định, Bộ Khoa học và Công nghệ công bố.

Bộ tiêu chuẩn TCVN 12854 (ISO/IEC 29192) *Công nghệ thông tin – Các kỹ thuật an toàn – Mật mã hạng nhẹ* gồm các tiêu chuẩn sau:

- TCVN 12854-1 : 2020 (ISO/IEC 29192-1:2012) Phần 1: Tổng quan
- TCVN 12854-2 : 2020 (ISO/IEC 29192-2:2012) Phần 2: Mã khối
- TCVN 12854-3 : 2020 (ISO/IEC 29192-3:2012) Phần 3: Mã dòng
- TCVN 12854-4 : 2020 (ISO/IEC 29192-4:2013) Phần 4: Các cơ chế sử dụng kỹ thuật phi đối xứng.

TIÊU CHUẨN QUỐC GIA TCVN 12854-2 : 2020

Công nghệ thông tin - Các kỹ thuật an toàn – Mật mã hạng nhẹ - Phần 2: Mã khối

Information technology – Security techniques – Lightweight cryptography – Part 2: Block ciphers

1 Phạm vi áp dụng

Tiêu chuẩn này quy định ba mã khối phù hợp với các yêu cầu cài đặt ứng dụng mật mã hạng nhẹ:

- PRESENT: Một mã khối hạng nhẹ với kích thước khối là 64 bit và kích thước khóa là 80 hoặc 128 bit.
- CLEFIA: Mã khối hạng nhẹ với kích thước khối là 128 bit, kích thước khóa là 128, 192, hoặc 256 bit.
- LEA:Mã khối hạng nhẹ với kích thước khối là 128 bit và kích thước khóa là 128, 192 hoặc 256 bit.

2 Tài liệu viện dẫn

Tiêu chuẩn này không có tài liệu viện dẫn.

3 Thuật ngữ và định nghĩa

Tiêu chuẩn này áp dụng các thuật ngữ và định nghĩa sau đây.

3.1

Khối (block)

Xâu bit có độ dài xác định

[TCVN 11367-1 (ISO/IEC 18033-1)]

3.2

Mã Khối (block cipher)

Hệ mật đối xứng với tính chất là thuật toán mã hóa thao tác trên một khối của bản rõ, nghĩa là xâu bit có độ dài xác định, kết quả cho ra là một khối của bản mã.

[TCVN 11367-1 (ISO/IEC 18033-1)]

3.3

Bản mã (ciphertext)

Dữ liệu được biến đổi để che giấu nội dung thông tin trong đó.

[TCVN 11817-1 (ISO/IEC 9798-1)]

3.4

Khóa (key)

Dãy các kí tự điều khiển sự vận hành của một biến đổi mật mã (ví dụ, mã hóa và giải mã).

CHÚ THÍCH Lấy từ TCVN 7817-1.

3.5

Mã khối n-bit (n-bit block cipher)

Mã khối với tính chất là các khối của bản rõ và bản mã đều có độ dài n bit.

[TCVN 12213 (ISO/IEC 10116)]

3.6

Bản rõ (plaintext)

Thông tin chưa được mã hóa.

CHÚ THÍCH Lấy từ TCVN 11495-1(ISO/IEC 9797-1).

3.7

Khóa vòng (round key)

Dãy các kí tự nhận được từ khóa sử dụng lược đồ khóa, và được sử dụng để điều khiển việc biến đổi thông tin trên mỗi vòng của mã khối.

4 Các ký tự

0x	Tiền tố cho xâu nhị phân trong ký hiệu hệ thập lục phân
‖	Phép ghép nối các xâu bit
$a \leftarrow b$	Cập nhật giá trị của a bằng giá trị của b
$\oplus$	Phép toán cộng XOR theo từng bit

5 Mã khối hạng nhẹ với kích thước khối 64 bit

5.1 Tổng quan

Điều này đặc tả một mã khối hạng nhẹ 64 bit là PRESENT trong 5.2.

Phụ lục A xác định các định danh đối tượng sử dụng làm định danh thuật toán được đặc tả trong Điều 5. Phụ lục B cung cấp các ví dụ số của các mã khối được mô tả trong tiêu chuẩn này. Phụ lục C tổng hợp các đặc tính hạng nhẹ của các mã khối được mô tả trong tiêu chuẩn này. Phụ lục D đưa ra một giới hạn số lượng các hoạt động của mã khối sử dụng một khóa đơn.

5.2 PRESENT

5.5.2 Thuật toán PRESENT

Thuật toán PRESENT [10] là một mã khối đối xứng xử lý các khối dữ liệu 64 bit, sử dụng khóa có độ dài 80 hoặc 128 bit. Mã khối được gọi là PRESENT-80 hoặc PRESENT-128 khi sử dụng khóa 80 bit hoặc 128 bit tương ứng.

5.2.2 Ký hiệu cụ thể cho PRESENT

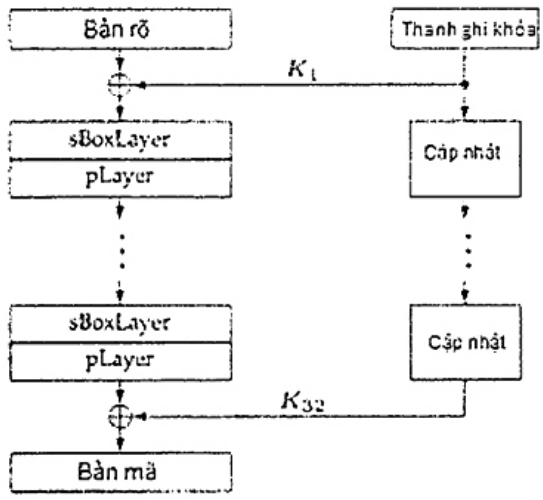
$K_i = k_{63}^i \ldots k_0^i$	khóa vòng 64 bit được sử dụng trong vòng i
k_b^i	bit thứ b của khóa vòng K_i
$K = k_{79} \ldots . k_0$	thanh ghi khóa 80 bit
k_b	bit thứ b của thanh ghi khóa K
STATE	trạng thái bên trong 64 bit
b_i	bit thứ i của *STATE* hiện tại

w_i từ 4 bit với $0 \leq i \leq 15$

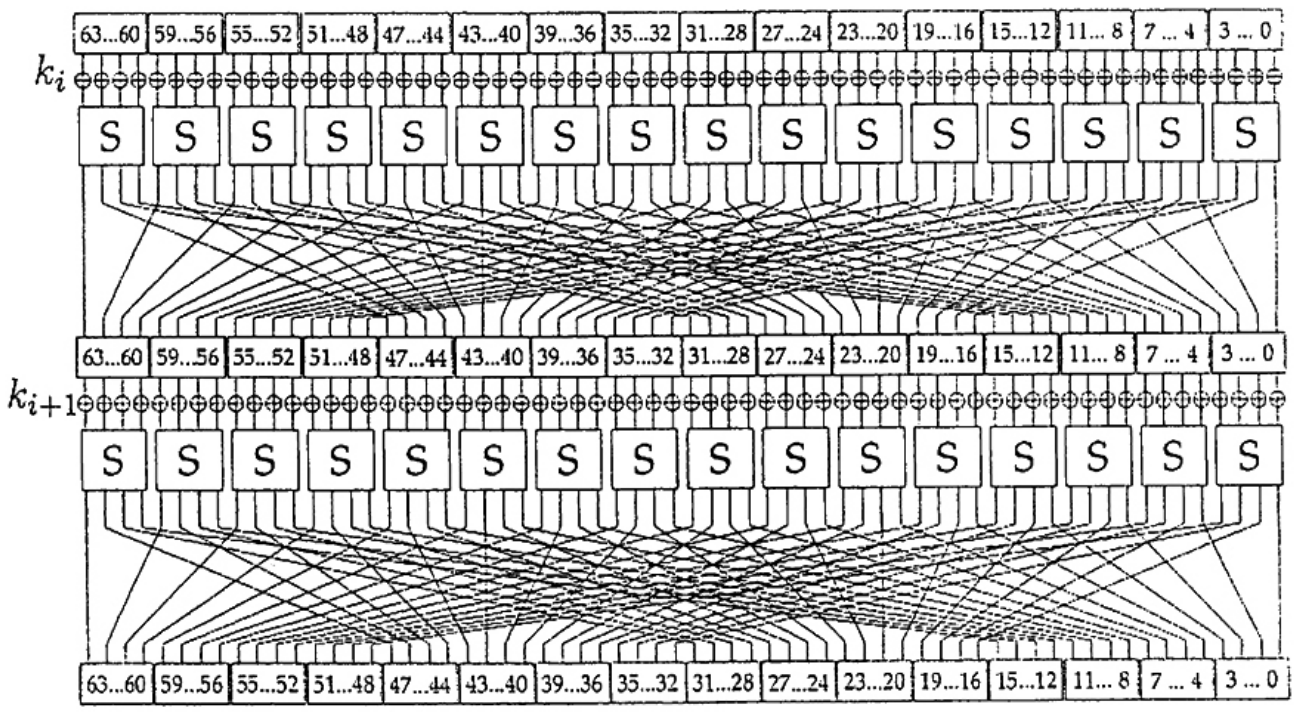
5.2.3 Mã hóa PRESENT

Mã khối PRESENT gồm 31 vòng, tức là 31 lần áp dụng của một dãy các biến đổi đơn giản. Một mô tả ở dạng giả mã của thuật toán mã hóa đầy đủ được đưa ra trong Hình 1, với *STATE* ký hiệu trạng thái bên trong. Các biến đổi cụ thể của thuật toán được xác định trong 5.2.5. Mỗi vòng của thuật toán sử dụng một khóa vòng khác nhau K_i ($1 \leq i \leq 31$), nhận được như mô tả trong 5.2.6. Hai vòng liên tiếp của thuật toán được chỉ rõ để minh họa trong Hình 2.

```
generateRoundKeys()
for i = 1 to 31 do
    addRoundKey(STATE,K_i)
    sBoxLayer(STATE)
    pLayer(STATE)
end for
addRoundKey(STATE,K_32)
```



Hình 1 - Thủ tục mã hóa của PRESENT



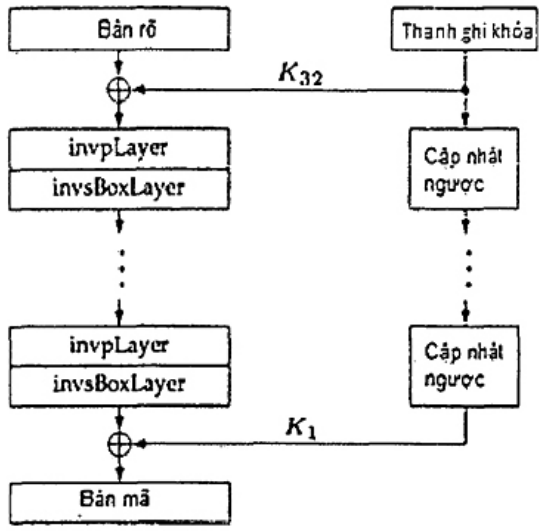
Hình 2 – Hai vòng của PRESENT

5.2.4 Giải mã PRESENT

Thuật toán giải mã PRESENT đầy đủ được đưa ra trong Hình 3. Các biến đổi cụ thể của thuật toán được xác định trong 5.2.5. Mỗi vòng của thuật toán sử dụng một khóa vòng K_i ($1 \leq i \leq 31$) khác nhau, nhận được như trong 5.2.6.

generateRoundKeys()
addRoundKey(STATE,K_{32})
for $i = 31$ downto 1 **do**
  invpLayer(STATE)
  invsBoxLayer(STATE)

  addRoundKey(STATE,K_i)
end for



Hình 3 – Thủ tục giải mã của PRESENT

5.2.5 Các biến đổi của PRESENT

5.2.5.2 addRoundKey

Cho trước khóa vòng $K_i = k^i_{63} \ldots k^i_0$ với $1 \leq i \leq 32$ và *STATE* hiện tại $b_{63} \ldots b_0$, **addRoundKey** gồm các phép toán $b_j \leftarrow b_j \oplus k^i_j$, với $0 \leq j \leq 63$.

5.2.5.2 sBoxLayer

Tầng phi tuyến **sBoxLayer** của quá trình mã hóa của PRESENT sử dụng một S-hộp S 4 bit sang 4 bit, nó được áp dụng 16 lần song song trong mỗi vòng. S-hộp biến đổi đầu vào x thành đầu ra $S(x)$ như mô tả theo hệ thập lục phân được chỉ ra trong Bảng 1.

Bảng 1 – S-hộp PRESENT

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Đối với **sBoxLayer** *STATE* hiện tại $b_{63} \ldots b_0$ được xem như 16 từ 4 bit $w_{15} \ldots w_0$ với $w_i = b_{4*i+3} \| b_{4*i+2} \| b_{4*i+1} \| b_{4*i}$ ($0 \leq i \leq 15$) và 4 bit đầu ra $S(w_i)$ tạo ra các giá trị cập nhật trạng thái là phép ghép $S(w_{15}) \| S(w_{14}) \| \cdots \| S(w_0)$.

5.2.5.3 invsBoxLayer

S-hộp sử dụng trong thủ tục giải mã của PRESENT là nghịch đảo của S-hộp S 4 bit sang 4 bit mà được mô tả trong 5.2.5.2. S-hộp nghịch đảo biến đổi đầu vào x thành đầu ra $S^{-1}(x)$ như được mô tả theo hệ thập lục phân trong Bảng 2.

Bảng 2 – S-hộp nghịch đảo của PRESENT

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S^{-1}(x)$	5	E	F	8	C	1	2	D	B	4	6	3	0	7	9	A

5.2.5.4 pLayer

Hoán vị bit **pLayer** sử dụng trong thủ tục mã hóa của PRESENT được cho trong Bảng 3. Bit i của *STATE* được chuyển đến vị trí bit $P(i)$.

Bảng 3 – Tầng hoán vị pLayer của PRESENT

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P(i)$	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P(i)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P(i)$	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

5.2.5.5 invpLayer

Tầng hoán vị nghịch đảo **invpLayer** sử dụng trong thủ tục giải mã của PRESENT được cho trong Bảng 4. Bit i của *STATE* được chuyển tới vị trí bit $P^{-1}(i)$.

Bảng 4 – Tầng hoán vị nghịch đảo invpLayer của PRESENT

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P^{-1}(i)$	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P^{-1}(i)$	1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P^{-1}(i)$	2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P^{-1}(i)$	3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63

5.2.6 Lược đồ khóa PRESENT

5.2.6.2 PRESENT-80 và PRESENT-128

PRESENT có thể lấy khóa hoặc 80 hoặc 128 bit. Phiên bản khóa 80 bit (PRESENT-80) được mô tả trong 5.2.6.2 và phiên bản khóa 128 bit (PRESENT-128) được mô tả trong 5.2.6.3.

5.2.6.2 Khóa 80 bit cho PRESENT-80

Khóa của người dùng cung cấp được lưu trữ trong thanh ghi khóa K và được biểu diễn là $k_{79}k_{78} \dots k_0$. Tại vòng i, khóa vòng 64 bit $K_i = k_{63}^i \dots k_0^i$ bao gồm 64 bit trái nhất của nội dung thanh ghi K hiện tại. Do vậy, tại vòng i chúng ta có:

$$K_i = k_{63}^i \dots k_0^i = k_{79}k_{78} \dots k_{16}.$$

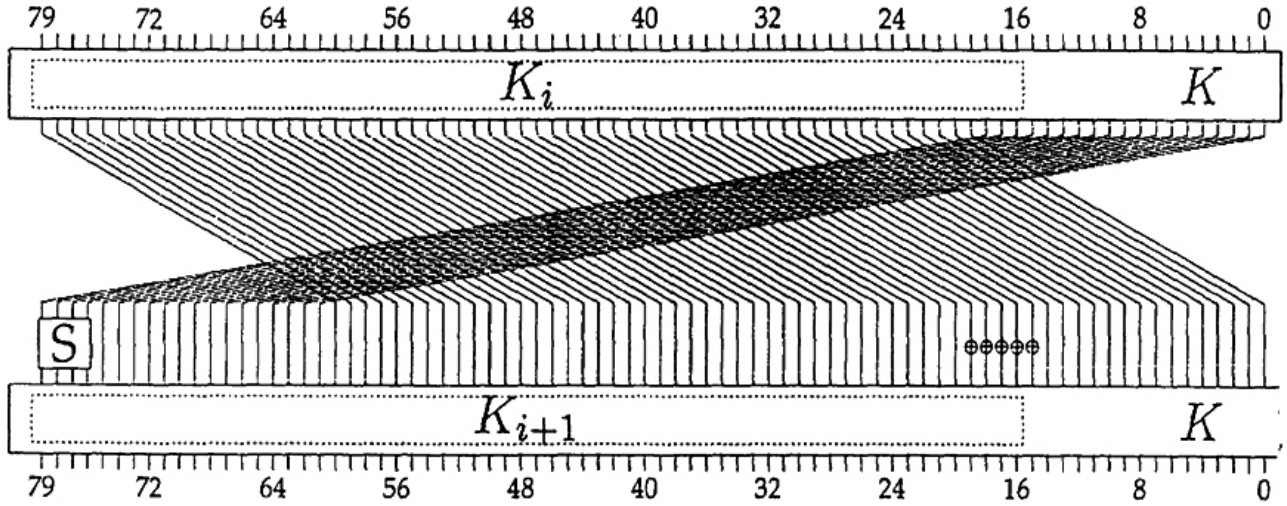
Sau khi trích khóa vòng K_i, thanh ghi khóa $K = k_{79}k_{78} \dots k_0$ được cập nhật như sau:

1) $k_{79}k_{78} \dots k_1k_0 \leftarrow k_{18}k_{17} \dots k_{20}k_{19}$

2) $k_{79}k_{78}k_{77}k_{76} \leftarrow S[k_{79}k_{78}k_{77}k_{76}]$

3) $k_{19}k_{18}k_{17}k_{16}k_{15} \leftarrow k_{19}k_{18}k_{17}k_{16}k_{15} \oplus round\_counter$

Diễn giải bằng lời, thanh ghi khóa được quay vòng 61 bit vị trí sang bên trái, 4 bit trái nhất được đưa qua S-hộp của PRESENT, và giá trị *round_counter* i được cộng XOR với các bit $k_{19}k_{18}k_{17}k_{16}k_{15}$ bit của K trong đó bit có trọng số thấp nhất của *round_counter* nằm bên phải. Các vòng được đánh số từ $1 \leq i \leq 31$ và *round_counter* = i. Hình 4 là minh họa lược đồ khóa của PRESENT-80.



Hình 4 – Lược đồ khóa PRESENT-80

5.2.6.3 khóa 128 bit cho PRESENT-128

Tương tự như biến thể 80 bit, khóa do người dùng cung cấp ban đầu được lưu trữ trong thanh ghi khóa K và được biểu diễn là $k_{127}k_{126} \dots k_0$. Tại vòng i, khóa vòng 64 bit $K_i = k_{63}^i \dots k_0^i$ gồm 64 bit trái nhất của nội dung hiện tại của thanh ghi K. Do vậy, tại vòng i ta có:

$$K_i = k_{63}^i \dots k_0^i = k_{127}k_{126} \dots k_{64}$$

Sau khi trích khóa vòng K_i, thanh ghi khóa $K = k_{127}k_{126} \dots k_0$ được cập nhật như sau:

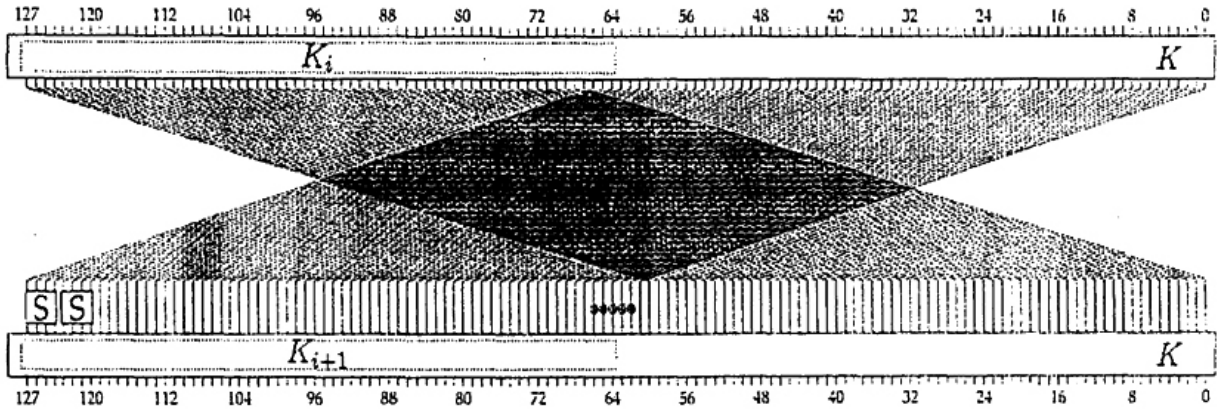
1) $k_{127}k_{126} \dots k_1k_0 \leftarrow k_{66}k_{65} \dots k_{68}k_{67}$

2) $k_{127}k_{126}k_{125}k_{124} \leftarrow S[k_{127}k_{126}k_{125}k_{124}]$

3) $k_{123}k_{122}k_{121}k_{120} \leftarrow S[k_{123}k_{122}k_{121}k_{120}]$

4) $k_{66}k_{65}k_{64}k_{63}k_{62} \leftarrow k_{66}k_{65}k_{64}k_{63}k_{62} \oplus round\_counter$

Diễn giải bằng lời, thanh ghi khóa được quay vòng 61 bit vị trí sang bên trái, 8 bit trái nhất được đưa qua S-hộp của PRESENT, và giá trị *round_counter i* là được cộng XOR với các bit $k_{66}k_{65}k_{64}k_{63}k_{62}$ của K bit có trọng số thấp nhất của *round_counter* nằm bên phải. Các vòng được đánh số từ $1 \leq i \leq 31$ và *round_counter* = i. Hình 5 là đồ họa mô tả lược đồ khóa PRESENT-128.



Hình 5 – Lược đồ khóa PRESENT-128

6 Mã khối hạng nhẹ với kích thước khối 128 bit

6.1 Tổng quan

Điều này đặc tả hai mã khối hạng nhẹ 128 bit: CLEFIA tại 6.2 và LEA tại 6.3.

Phụ lục A xác định các định danh đối tượng được sử dụng làm định danh các thuật toán được đặc tả trong Điều 6. Phụ lục B cung cấp các ví dụ số của các mã khối được mô tả trong tiêu chuẩn này. Phụ lục C tổng hợp các đặc tính hạng nhẹ của các mã khối được mô tả trong tiêu chuẩn này. Phụ lục D đưa ra một giới hạn số lượng các hoạt động của mã khối sử dụng một khóa đơn.

6.2 CLEFIA

6.2.1 Thuật toán CLEFIA

Thuật toán CLEFIA [15] là một mã khối đối xứng, có thể xử lý các khối dữ liệu 128 bit sử dụng một khóa mật mã có độ dài 128, 192, 256 bit. Số vòng cho CLEFIA là 18, 22 và 26 với các khóa có kích thước 128 bit, 192 bit và 256 bit tương ứng. Tổng số khóa vòng phụ thuộc vào độ dài khóa. Các hàm mã hóa và giải mã của CLEFIA yêu cầu 36, 44 và 52 khóa vòng cho các khóa 128 bit, 192 bit và 256 bit tương ứng.

6.2.2 Các ký hiệu riêng cho CLEFIA

$a_{(b)}$	Xâu bit có độ dài b bit
$\{0,1\}^n$	Tập các xâu nhị phân có độ dài n
$\cdot$	Phép nhân trên trường GF(2^n)
$\lll i$	Phép dịch vòng sang trái i bit
~a	Phép lấy bù theo từng bit của xâu bit a
Σ^n	n lần phép toán của hàm DoubleSwap Σ

6.2.3 Mã hóa CLEFIA

Quá trình mã hóa của CLEFIA dựa trên cấu trúc Feistel tổng quát hóa 4 nhánh r- vòng $GFN_{4,r}$. Cho $P, C \in \{0,1\}^{128}$ là bản rõ và bản mã. Cho $P_i, C_i \in \{0,1\}^{32}$ $(0 \leq i < 4)$ là các phần được chia từ các bản rõ và bản mã với $P = P_0 \| P_1 \| P_2 \| P_3$ và $C = C_0 \| C_1 \| C_2 \| C_3$. Cho $WK_0, WK_1, WK_2, WK_3 \in \{0,1\}^{32}$ là các khóa làm

trắng và $RK_i \in \{0,1\}^{32}$ $(0 \leq i < 2r)$ là các khóa vòng được cung cấp bởi lược đồ khóa. Khi đó, hàm mã hóa r-vòng ENC_r được định nghĩa như sau:

ENC_r:

1) $T_0 \| T_1 \| T_2 \| T_3 \leftarrow P_0 \| (P_1 \oplus WK_0) \| P_2 \| (P_3 \oplus WK_1)$

2) $T_0 \| T_1 \| T_2 \| T_3 \leftarrow GFN_{4,r}(RK_0, ..., RK_{2r-1}, T_0, T_1, T_2, T_3)$

3) $C_0 \| C_1 \| C_2 \| C_3 \leftarrow T_0 \| (T_1 \oplus WK_2) \| T_2 \| (T_3 \oplus WK_3)$

6.2.4 Giải mã CLEFIA

Hàm giải mã DEC_r được định nghĩa như sau:

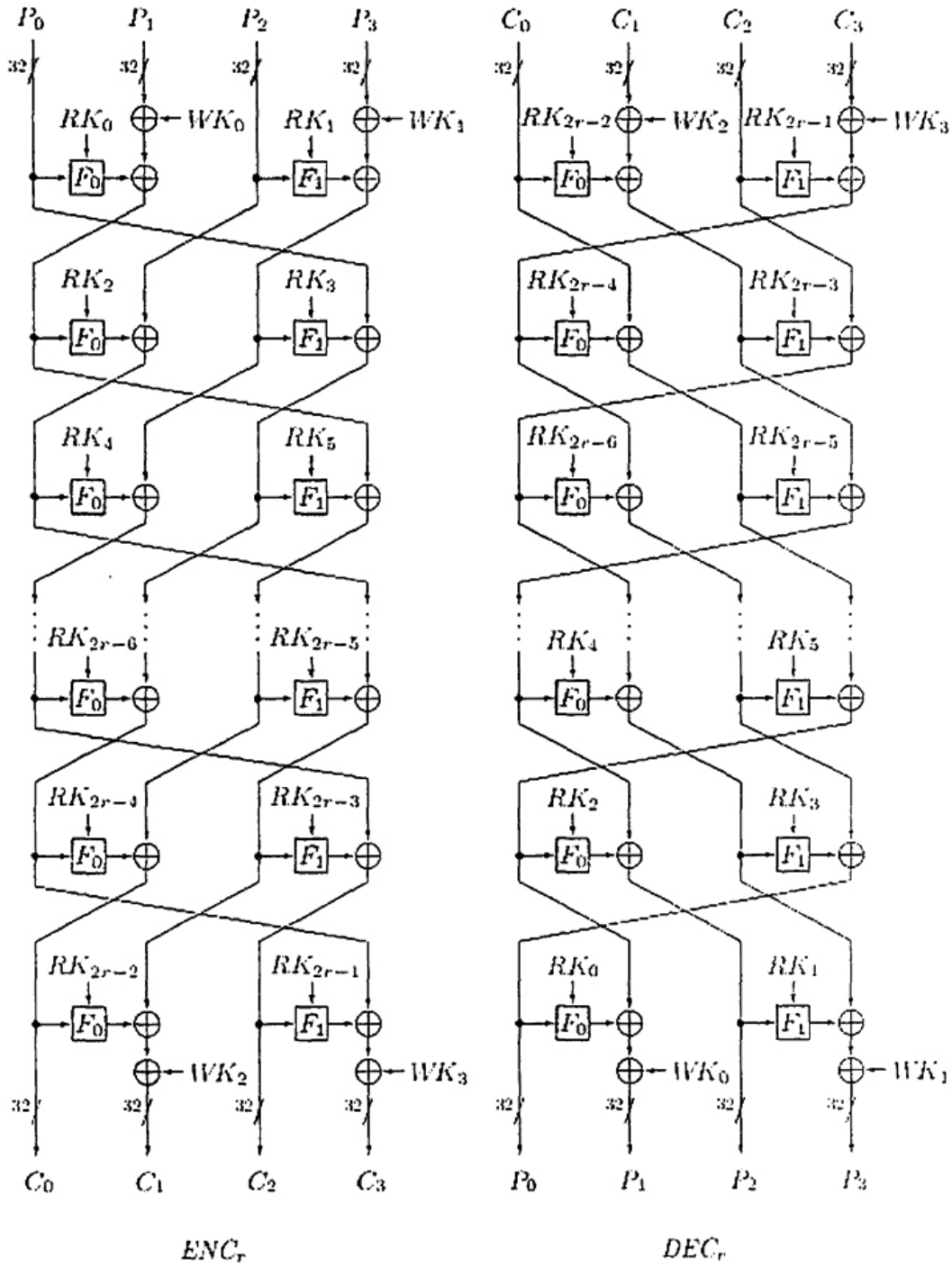
DEC_r:

1) $T_0 \| T_1 \| T_2 \| T_3 \leftarrow C_0 \| (C_1 \oplus WK_2) \| C_2 \| (C_3 \oplus WK_3)$

2) $T_0 \| T_1 \| T_2 \| T_3 \leftarrow GFN_{4,r}^{-1}(RK_0, ..., RK_{2r-1}, T_0, T_1, T_2, T_3)$

3) $P_0 \| P_1 \| P_2 \| P_3 \leftarrow T_0 \| (T_1 \oplus WK_0) \| T_2 \| (T_3 \oplus WK_1)$

Hình 6 minh họa cho cả ENC_r và DEC_r.



Hình 6 – Thủ tục mã hóa và thủ tục giải mã của CLEFIA

6.2.5 Các khối cơ sở của CLEFIA

6.2.5.2 $GFN_{d,r}$

Cấu trúc cơ bản của CLEFIA là cấu trúc Feistel tổng quát. Cấu trúc này dùng được trong cả phần xử lý dữ liệu và phần lược đồ khóa.

CLEFIA sử dụng 4 nhánh và mạng Feistel tổng quát 8-nhánh. Mạng Feistel tổng quát 4-nhánh được dùng trong phần xử lý dữ liệu và lược đồ khóa với khóa 128 bit. Mạng Feistel tổng quát 8-nhánh được áp dụng trong lược đồ khóa với các khóa 192/256 bit. Chúng ta ký hiệu mạng Feistel tổng quát d-nhánh r-vòng dùng trong CLEFIA là $GFN_{d,r}$. $GFN_{d,r}$ sử dụng hai F-hàm 32 bit khác nhau là F_0 và F_1.

Với d cặp của đầu vào X_i và đầu ra Y_i $(0 \le i < d)$ 32 bit và dr/2 khóa vòng 32 bit RK_i $(0 \le i < dr/2)$, $GFN_{d,r}$ $(d = 4, 8)$ và hàm ngược $GFN_{d,r}^{-1}$ $(d = 4)$ được xác định như sau:

$GFN_{4,r}$:

1) $T_0 \| T_1 \| T_2 \| T_3 \leftarrow X_0 \| X_1 \| X_2 \| X_3$

2) Với i = 0 đến r-1:

2.1) $T_1 \leftarrow T_1 \oplus F_0(RK_{2i}, T_0)$

$T_3 \leftarrow T_3 \oplus F_1(RK_{2i+1}, T_2)$

2.2) $T_0 \| T_1 \| T_2 \| T_3 \leftarrow T_1 \| T_2 \| T_3 \| T_0$

3) $Y_0 \| Y_1 \| Y_2 \| Y_3 \leftarrow T_3 \| T_0 \| T_1 \| T_2$

$GFN_{8,r}$:

1) $T_0 \| T_1 \| \ldots \| T_7 \leftarrow X_0 \| X_1 \| \ldots \| X_7$

2) Với i = 0 đến r-1:

2.1) $T_1 \leftarrow T_1 \oplus F_0(RK_{4i}, T_0)$

$T_3 \leftarrow T_3 \oplus F_1(RK_{4i+1}, T_2)$

$T_5 \leftarrow T_5 \oplus F_0(RK_{4i+2}, T_4)$

$T_7 \leftarrow T_7 \oplus F_1(RK_{4i+3}, T_6)$

2.2) $T_0 \| T_1 \| \ldots \| T_6 \| T_7 \leftarrow T_1 \| T_2 \| \ldots \| T_7 \| T_0$

3) $Y_0 \| Y_1 \| \ldots \| Y_6 \| Y_7 \leftarrow T_7 \| T_0 \| \ldots \| T_5 \| T_6$

Hàm ngược $GFN_{4,r}^{-1}$ thu được bằng cách đổi thứ tự của RK_i và hướng của phép dịch vòng từ tại 2.2) và 3) trong $GFN_{4,r}$.

$GFN_{4,r}^{-1}$:

1) $T_0 \| T_1 \| T_2 \| T_3 \leftarrow X_0 \| X_1 \| X_2 \| X_3$

2) Với i = 0 đến r-1:

2.1) $T_1 \leftarrow T_1 \oplus F_0(RK_{2(r-i)-2}, T_0)$

$T_3 \leftarrow T_3 \oplus F_1(RK_{2(r-i)-1}, T_2)$

2.2) $T_0 \| T_1 \| T_2 \| T_3 \leftarrow T_3 \| T_0 \| T_1 \| T_2$

3) $Y_0 \| Y_1 \| Y_2 \| Y_3 \leftarrow T_1 \| T_2 \| T_3 \| T_0$

6.2.5.2 Các F-hàm

Hai F-hàm F_0 và F_1 sử dụng trong $GFN_{d,r}$ được xác định như sau:

F_0: $(RK_{(32)}, x_{(32)}) \mapsto y_{(32)}$

1) $V \leftarrow RK \oplus x$

2) Cho $V = V_0 \| V_1 \| V_2 \| V_3$, $V_i \in \{0,1\}^8$.

$V_0 \leftarrow S_0(V_0)$

$V_1 \leftarrow S_1(V_1)$

$V_2 \leftarrow S_0(V_2)$

$V_3 \leftarrow S_1(V_3)$

3) Cho $y = y_0 \| y_1 \| y_2 \| y_3$, $y_i \in \{0,1\}^8$.

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} \leftarrow M_0 \begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{pmatrix}$$

F_1: $(RK_{(32)}, x_{(32)}) \mapsto y_{(32)}$

1) $V \leftarrow RK \oplus x$

2) Cho $V = V_0 \| V_1 \| V_2 \| V_3$, $V_i \in \{0,1\}^8$.

$V_0 \leftarrow S_1(V_0)$

$V_1 \leftarrow S_0(V_1)$

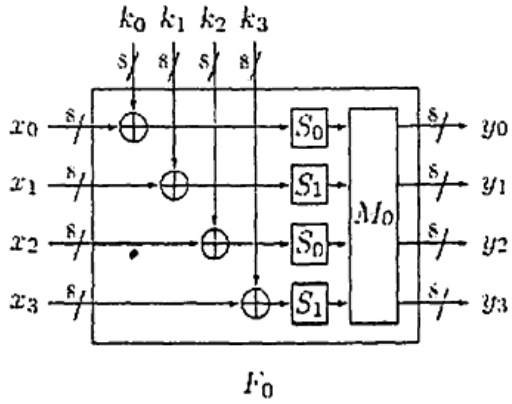
$V_2 \leftarrow S_1(V_2)$

$V_3 \leftarrow S_0(V_3)$

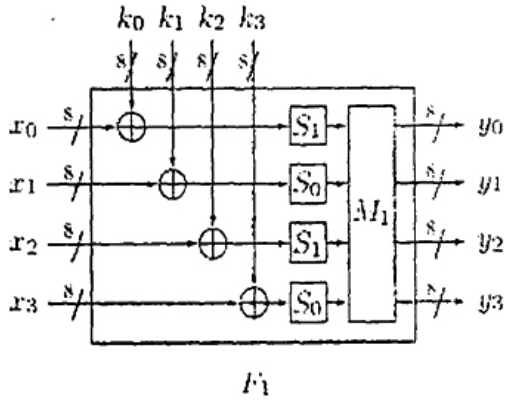
3) Cho $y = y_0 \| y_1 \| y_2 \| y_3$, $y_i \in \{0,1\}^8$.

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} \leftarrow M_1 \begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \end{pmatrix}$$

S_0 và S_1 là các S-hộp phi tuyến 8 bit, M_0 và M_1 là các ma trận khuếch tán được mô tả trong các tiểu mục ở dưới. Trong mỗi F-hàm sử dụng hai S-hộp và một ma trận, nhưng các S-hộp được sử dụng theo thứ tự khác nhau và các ma trận khác nhau. Hình 7 là đồ họa mô tả các F-hàm.



F_0



F_1

Hình 7 – Các F-hàm

6.2.5.3 Các S-hộp

CLEFIA sử dụng hai kiểu S-hộp 8 bit khác nhau S_0 và S_1; S_0 dựa trên bốn S-hộp 4 bit ngẫu nhiên, và S_1 dựa trên hàm nghịch đảo trên GF(2^8).

Bảng 5 và bảng 6 chỉ ra các giá trị đầu ra của S_0 và S_1 tương ứng. Tất cả các giá trị trong Bảng được biểu diễn ở hệ thập lục phân. Với một đầu vào 8 bit của S-hộp, 4 bit cao biểu diễn hàng và 4 bit thấp biểu diễn cột. Ví dụ, nếu một giá trị 0xab là đầu vào, 0x7e là đầu ra của S_0 vì nó cùng nằm trên đường giao của hàng chỉ số 'a.' và cột có chỉ số là '.b'

Bảng 5 - S_0

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.a	.b	.c	.d	.e	.f
0.	57	49	d1	c6	2f	33	74	fb	95	6d	82	ea	0e	b0	a8	1c
1.	28	d0	4b	92	5c	ee	85	b1	c4	0a	76	3d	63	f9	17	af
2.	bf	a1	19	65	f7	7a	32	20	06	ce	e4	83	9d	5b	4c	d8
3.	42	5d	2e	e8	d4	9b	0f	13	3c	89	67	c0	71	aa	b6	f5
4.	a4	be	fd	8c	12	00	97	da	78	e1	cf	6b	39	43	55	26
5.	30	98	cc	dd	eb	54	b3	8f	4e	16	fa	22	a5	77	09	61
6.	d6	2a	53	37	45	c1	6c	ae	ef	70	08	99	8b	1d	f2	b4
7.	e9	c7	9f	4a	31	25	fe	7c	d3	a2	bd	56	14	88	60	0b
8.	cd	e2	34	50	9e	dc	11	05	2b	b7	a9	48	ff	66	8a	73
9.	03	75	86	f1	6a	a7	40	c2	b9	2c	db	1f	58	94	3e	ed
a.	fc	1b	a0	04	b8	8d	e6	59	62	93	35	7e	ca	21	df	47
b.	15	f3	ba	7f	a6	69	c8	4d	87	3b	9c	01	e0	de	24	52
c.	7b	0c	68	1e	80	b2	5a	e7	ad	d5	23	f4	46	3f	91	c9
d.	6e	84	72	bb	0d	18	d9	96	f0	5f	41	ac	27	c5	e3	3a
e.	81	6f	07	a3	79	f6	2d	38	1a	44	5e	b5	d2	ec	cb	90
f.	9a	36	e5	29	c3	4f	ab	64	51	f8	10	d7	bc	02	7d	8e

Bảng 6 - S_1

	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	.a	.b	.c	.d	.e	.f
0.	6c	da	c3	e9	4e	9d	0a	3d	b8	36	b4	38	13	34	0c	d9
1.	bf	74	94	8f	b7	9c	e5	dc	9e	07	49	4f	98	2c	b0	93
2.	12	eb	cd	b3	92	e7	41	60	e3	21	27	3b	e6	19	d2	0e
3.	91	11	c7	3f	2a	8e	a1	bc	2b	c8	c5	0f	5b	f3	87	8b
4.	fb	f5	de	20	c6	a7	84	ce	d8	65	51	c9	a4	ef	43	53
5.	25	5d	9b	31	e8	3e	0d	d7	80	ff	69	8a	ba	0b	73	5c
6.	6e	54	15	62	f6	35	30	52	a3	16	d3	28	32	fa	aa	5e
7.	cf	ea	ed	78	33	58	09	7b	63	c0	c1	46	1e	df	a9	99
8.	55	04	c4	86	39	77	82	ec	40	18	90	97	59	dd	83	1f
9.	9a	37	06	24	64	7c	a5	56	48	08	85	d0	61	26	ca	6f
a.	7e	6a	b6	71	a0	70	05	d1	45	8c	23	1c	f0	ee	89	ad
b.	7a	4b	c2	2f	db	5a	4d	76	67	17	2d	f4	cb	b1	4a	a8
c.	b5	22	47	3a	d5	10	4c	72	cc	00	f9	e0	fd	e2	fe	ae
d.	f8	5f	ab	f1	1b	42	81	d6	be	44	29	a6	57	b9	af	f2
e.	d4	75	66	bb	68	9f	50	02	01	3c	7f	8d	1a	88	bd	ac
f.	f7	e4	79	96	a2	fc	6d	b2	6b	03	e1	2e	7d	14	95	1d

a) S-hộp S_0

$S_0 : \{0,1\}^8 \to \{0,1\}^8 : x \mapsto y = S_0(x)$ được sinh bởi sự kết hợp của 4 S-hộp 4 bit SS_0, SS_1, SS_2, và SS_3 theo cách dưới đây. Giá trị của mỗi S-hộp được xác định trong Bảng 7.

1) $t_0 \leftarrow SS_0(x_0), t_1 \leftarrow SS_1(x_1)$, với $x = x_0 \parallel x_1, x_i \in \{0,1\}^4$

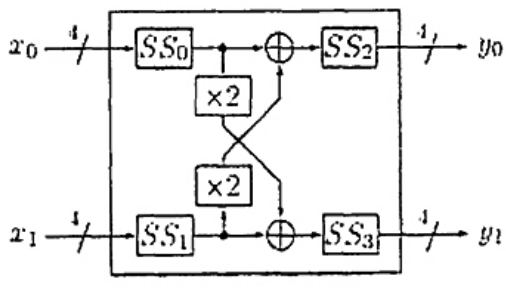
2) $u_0 \leftarrow t_0 \oplus 0 \times 2 \cdot t_1, u_1 \leftarrow 0 \times 2 \cdot t_0 \oplus t_1$

3) $y_0 \leftarrow SS_2(u_0), y_1 \leftarrow SS_3(u_1)$, với $y = y_0 \parallel y_1, y_i \in \{0,1\}^4$

Phép nhân $0 \times 2 \cdot t_1$ được thực hiện trong trường GF(2^4) xác định bởi đa thức nguyên thủy $z^4 + z + 1$. Hình 8 đưa ra cách xây dựng của S_0.

Bảng 7 - SS_i $(0 \le i < 4)$

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
$SS_0(x)$	e	6	c	a	8	7	2	f	b	1	4	0	5	9	d	3
$SS_1(x)$	6	4	0	d	2	b	a	3	9	c	e	f	8	7	5	1
$SS_2(x)$	b	8	5	e	a	6	4	c	f	7	2	3	1	0	d	9
$SS_3(x)$	a	2	6	d	3	4	5	e	0	7	8	9	b	f	c	1



Hình 8 - S_0

b) S-hộp S_1

$1 : \{0,1\}^8 \to \{0,1\}^8 : x \mapsto y = S_1(x)$ được xác định như sau:

$$y = \begin{cases} g((f(x))^{-1}) & if \quad f(x) \neq 0 \\ g(0) & if \quad f(x) = 0 \end{cases}$$

Hàm nghịch đảo được thực hiện trong trường GF(2^8) xác định bởi đa thức nguyên thủy $z^8 + z^4 + z^3 + z^2 + 1$ (= 0x11d). f và g là các phép biến đổi affine trên trường GF(2), mà được xác định như sau:

$f : \{0,1\}^8 \to \{0,1\}^8 : x \mapsto y = f(x)$,

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

$g : \{0,1\}^8 \to \{0,1\}^8 : x \mapsto y = g(x)$,

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Với $x = x_0 \| x_1 \| x_2 \| x_3 \| x_4 \| x_5 \| x_6 \| x_7$ và $y = y_0 \| y_1 \| y_2 \| y_3 \| y_4 \| y_5 \| y_6 \| y_7$, $x_i, y_i \in \{0,1\}$. Các hằng số trong f và g có thể được biểu diễn như 0x1e và 0x69 tương ứng.

6.2.5.4 Các ma trận khuếch tán

Các ma trận M_0 và M_1 được xác định như sau:

$$M_0 = \begin{pmatrix} 01 & 02 & 04 & 06 \\ 02 & 01 & 06 & 04 \\ 04 & 06 & 01 & 02 \\ 06 & 04 & 02 & 01 \end{pmatrix}, \quad M_1 = \begin{pmatrix} 01 & 08 & 02 & 0a \\ 08 & 01 & 0a & 02 \\ 02 & 0a & 01 & 08 \\ 0a & 02 & 08 & 01 \end{pmatrix}.$$

Các phép nhân của một ma trận và một véc tơ được thực hiện trên trường GF(2^8) xác định bởi đa thức nguyên thủy $z^8 + z^4 + z^3 + z^2 + 1$ (= 0x11d).

6.2.6 Lược đồ khóa CLEFIA

6.2.6.2 Cấu trúc chung

Lược đồ khóa CLEFIA hỗ trợ các khóa 128, 192 và 256 bit và tạo ra các khóa trắng $WK_i (0 \le i < 4)$ và các khóa vòng $RK_j (0 \le j < 2r)$ cho phần xử lý dữ liệu. Gọi K là khóa và L là khóa trung gian. Lược đồ khóa bao gồm hai bước sau:

1) Sinh L từ K.

2) Mở rộng K và L (sinh WK_i và RK_j)

Để sinh L từ K, lược đồ khóa cho khóa 128 bit sử dụng một phép hoán vị 128 bit $GFN_{4,12}$, trong đó lược đồ khóa cho các khóa 192/256 bit sử dụng một phép hoán vị 256 bit $GFN_{8,10}$.

6.2.6.2 Lược đồ khóa cho khóa 128 bit

Khóa trung gian 128 bit L được sinh ở bước 1 bằng cách áp dụng $GFN_{4,12}$, trong đó lấy 24 giá trị hằng số 32 bit $CON_i^{(128)}$ $(0 \le i < 24)$ làm các khóa vòng và $K = K_0 \| K_1 \| K_2 \| K_3$ làm đầu vào. Tiếp đó K và L được sử dụng để sinh $WK_i (0 \le i < 4)$ và $RK_j (0 \le j < 36)$ tại bước 2 và 3. 36 giá trị hằng số 32 bit $CON_i^{(128)}$ $(24 \le i < 60)$ sử dụng ở bước 3 được xác định trong 6.2.6.6. Hàm DoubleSwap Σ được xác định trong 6.2.6.5.

(Sinh L từ K)

1) $L \leftarrow GFN_{4,12}(CON_0^{(128)}, \ldots, CON_{23}^{(128)}, K_0, \ldots, K_3)$

Mở rộng K và L

2) $WK_0 \| WK_1 \| WK_2 \| WK_3 \leftarrow K$

3) Với i =0 đến 8

$T \leftarrow L \oplus (CON_{24+4i}^{(128)} \| CON_{24+4i+1}^{(128)} \| CON_{24+4i+2}^{(128)} \| CON_{24+4i+3}^{(128)})$

$L \leftarrow \Sigma(L)$

Nếu i lẻ: $T \leftarrow T \oplus K$

$RK_{4i} \| RK_{4i+1} \| RK_{4i+2} \| RK_{4i+3} \leftarrow T$

Bảng 8 chỉ ra mối quan hệ giữa các khóa vòng được sinh và dữ liệu có liên quan

Bảng 8 – Mở rộng K và L (Khóa 128 bit)

$WK_0\ WK_1\ WK_2\ WK_3$	K
$RK_0\ RK_1\ RK_2\ RK_3$	$L \oplus (CON_{24}^{(128)} \| CON_{25}^{(128)} \| CON_{26}^{(128)} \| CON_{27}^{(128)})$
$RK_4\ RK_5\ RK_6\ RK_7$	$\Sigma(L) \oplus K \oplus (CON_{28}^{(128)} \| CON_{29}^{(128)} \| CON_{30}^{(128)} \| CON_{31}^{(128)})$
$RK_8\ RK_9\ RK_{10}\ RK_{11}$	$\Sigma^2(L) \oplus (CON_{32}^{(128)} \| CON_{33}^{(128)} \| CON_{34}^{(128)} \| CON_{35}^{(128)})$
$RK_{12}\ RK_{13}\ RK_{14}\ RK_{15}$	$\Sigma^3(L) \oplus K \oplus (CON_{36}^{(128)} \| CON_{37}^{(128)} \| CON_{38}^{(128)} \| CON_{39}^{(128)})$
$RK_{16}\ RK_{17}\ RK_{18}\ RK_{19}$	$\Sigma^4(L) \oplus (CON_{40}^{(128)} \| CON_{41}^{(128)} \| CON_{42}^{(128)} \| CON_{43}^{(128)})$
$RK_{20}\ RK_{21}\ RK_{22}\ RK_{23}$	$\Sigma^5(L) \oplus K \oplus (CON_{44}^{(128)} \| CON_{45}^{(128)} \| CON_{46}^{(128)} \| CON_{47}^{(128)})$
$RK_{24}\ RK_{25}\ RK_{26}\ RK_{27}$	$\Sigma^6(L) \oplus (CON_{48}^{(128)} \| CON_{49}^{(128)} \| CON_{50}^{(128)} \| CON_{51}^{(128)})$
$RK_{28}\ RK_{29}\ RK_{30}\ RK_{31}$	$\Sigma^7(L) \oplus K \oplus (CON_{52}^{(128)} \| CON_{53}^{(128)} \| CON_{54}^{(128)} \| CON_{55}^{(128)})$
$RK_{32}\ RK_{33}\ RK_{34}\ RK_{35}$	$\Sigma^8(L) \oplus (CON_{56}^{(128)} \| CON_{57}^{(128)} \| CON_{58}^{(128)} \| CON_{59}^{(128)})$

6.2.6.3 Lược đồ khóa cho khóa 192 bit

Hai giá trị 128 bit K_L và K_R được sinh từ khóa 192 bit $K = K_0 \| K_1 \| K_2 \| K_3 \| K_4 \| K_5$ với $K_i \in \{0,1\}^{32}$. Sau đó hai giá trị L_L và L_R 128 bit được sinh bằng cách áp dụng $GFN_{8,10}$ mà lấy $CON_i^{(192)}$ $(0 \le i < 40)$ làm các khóa vòng và $K_L \| K_R$ ở trên làm đầu vào 256 bit. Hình 9 chỉ ra cấu trúc của $GFN_{8,10}$.

K_L, K_R và L_L, L_R được sử dụng để sinh $WK_i (0 \le i < 4)$ và $RK_j (0 \le j < 44)$ tại bước 4 và 5 dưới đây. Trong phần sau, 44 giá trị hằng số 32 bit $CON_i^{(192)}$ $(40 \le i < 84)$ được sử dụng.

Các bước dưới đây chỉ ra lược đồ khóa 192 bit/256 bit. Với lược đồ khóa 192 bit, giá trị của k được đặt là 192.

(Sinh L_L, L_R từ K_L, K_R với một khóa k bit)

1) Đặt $k = 192$ hoặc $k = 256$

2) Nếu k=192: $K_L \leftarrow K_0 \| K_1 \| K_2 \| K_3, K_R \leftarrow K_4 \| K_5 \| \tilde{K}_0 \| \tilde{K}_1$

hoặc nếu k=256 : $K_L \leftarrow K_0 \| K_1 \| K_2 \| K_3, K_R \leftarrow K_4 \| K_5 \| K_6 \| K_7$

2) Cho $K_L = K_{L0} \| K_{L1} \| K_{L2} \| K_{L3}, K_R = K_{R0} \| K_{R1} \| K_{R2} \| K_{R3}$

$$L_L \| L_R \leftarrow GFN_{8,10}(CON_0^{(k)}, \ldots, CON_{39}^{(k)}, K_{L0}, \ldots, K_{L3}, K_{R0}, \ldots, K_{R3})$$

(Mở rộng K_L, K_R và L_L, L_R với khóa k-bit)

4) $WK_0 \| WK_1 \| WK_2 \| WK_3 \leftarrow K_L \oplus K_R$

5) Với $i = 0$ đến 10 (nếu $k = 192$), hoặc 12 (nếu $k = 256$) thực hiện:

Nếu (i mod 4) = 0 hoặc 1:

$$T \leftarrow L_L \oplus (CON_{40-4i}^{(k)} \| CON_{40-4i-1}^{(k)} \| CON_{40-4i-2}^{(k)} \| CON_{40-4i-3}^{(k)})$$

$$L_L \leftarrow \Sigma(L_L)$$

nếu i lẻ: $T \leftarrow T \oplus K_R$

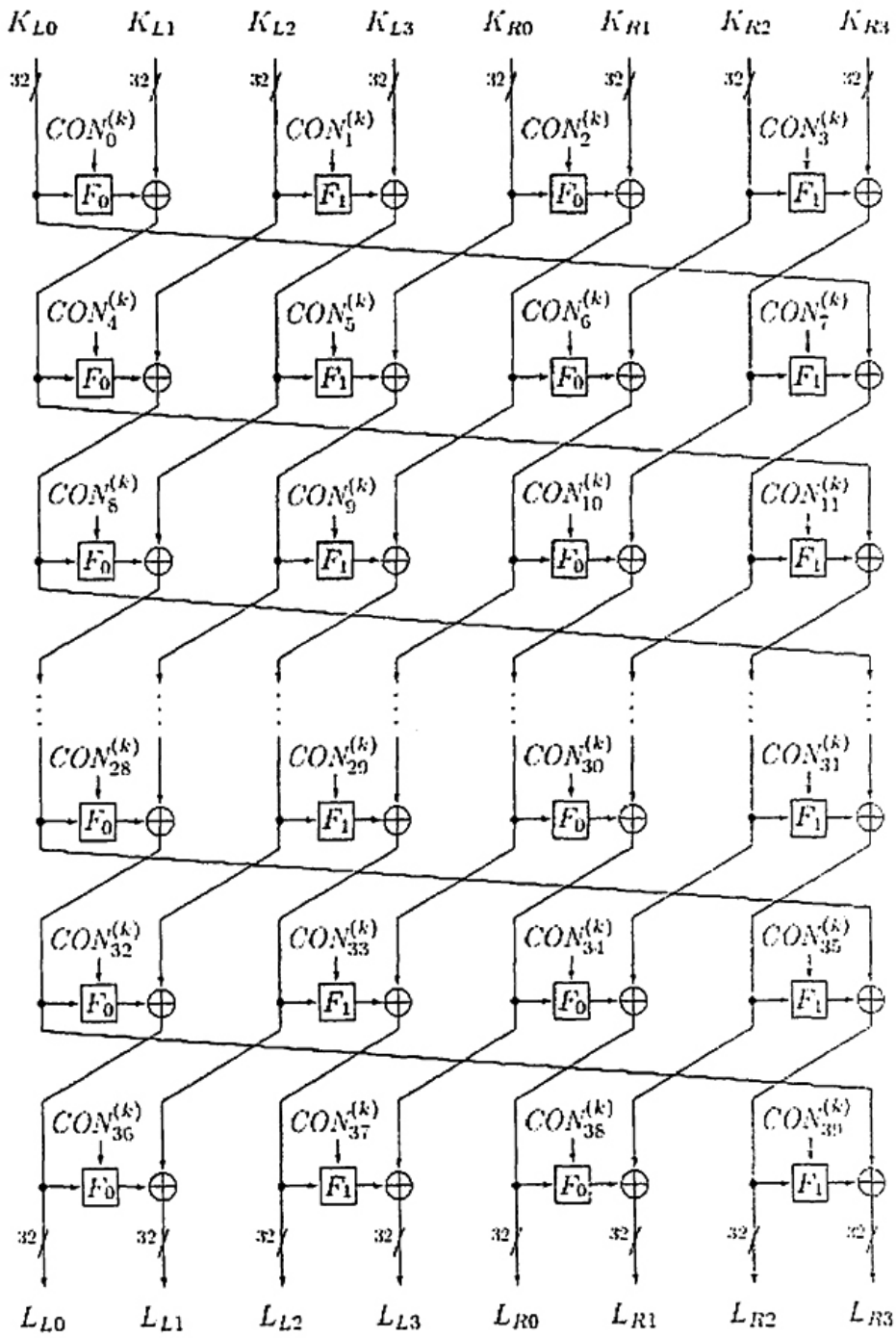
Hoặc:

$$T \leftarrow L_R \oplus (CON_{40-4i}^{(k)} \| CON_{40-4i+1}^{(k)} \| CON_{40-4i+2}^{(k)} \| CON_{40-4i-3}^{(k)})$$

$$L_R \leftarrow \Sigma(L_R)$$

nếu i lẻ: $T \leftarrow T \oplus K_L$

$$RK_{4i} \| RK_{4i+1} \| RK_{4i+2} \| RK_{4i+3} \leftarrow T$$



Hình 9 – Cấu trúc của $GFN_{8,10}$

Bảng 9 chỉ ra mối quan hệ giữa các khóa vòng được tạo và dữ liệu có liên quan

Bảng 9 – Mở rộng K_L, K_R, L_L, L_R (khóa 192 bit)

$WK_0\ WK_1\ WK_2\ WK_3$	$K_L \quad \oplus K_R$
$RK_0\ RK_1\ RK_2\ RK_3$	$L_L \quad \oplus (CON_{40}^{(192)} \| CON_{41}^{(192)} \| CON_{42}^{(192)} \| CON_{43}^{(192)})$
$RK_4\ RK_5\ RK_6\ RK_7$	$\Sigma(L_L) \oplus K_R \oplus (CON_{44}^{(192)} \| CON_{45}^{(192)} \| CON_{46}^{(192)} \| CON_{47}^{(192)})$
$RK_8\ RK_9\ RK_{10}\ RK_{11}$	$L_R \quad \oplus (CON_{48}^{(192)} \| CON_{49}^{(192)} \| CON_{50}^{(192)} \| CON_{51}^{(192)})$
$RK_{12}\ RK_{13}\ RK_{14}\ RK_{15}$	$\Sigma(L_R) \oplus K_L \oplus (CON_{52}^{(192)} \| CON_{53}^{(192)} \| CON_{54}^{(192)} \| CON_{55}^{(192)})$
$RK_{16}\ RK_{17}\ RK_{18}\ RK_{19}$	$\Sigma^2(L_L) \quad \oplus (CON_{56}^{(192)} \| CON_{57}^{(192)} \| CON_{58}^{(192)} \| CON_{59}^{(192)})$
$RK_{20}\ RK_{21}\ RK_{22}\ RK_{23}$	$\Sigma^3(L_L) \oplus K_R \oplus (CON_{60}^{(192)} \| CON_{61}^{(192)} \| CON_{62}^{(192)} \| CON_{63}^{(192)})$
$RK_{24}\ RK_{25}\ RK_{26}\ RK_{27}$	$\Sigma^2(L_R) \quad \oplus (CON_{64}^{(192)} \| CON_{65}^{(192)} \| CON_{66}^{(192)} \| CON_{67}^{(192)})$
$RK_{28}\ RK_{29}\ RK_{30}\ RK_{31}$	$\Sigma^3(L_R) \oplus K_L \oplus (CON_{68}^{(192)} \| CON_{69}^{(192)} \| CON_{70}^{(192)} \| CON_{71}^{(192)})$
$RK_{32}\ RK_{33}\ RK_{34}\ RK_{35}$	$\Sigma^4(L_L) \quad \oplus (CON_{72}^{(192)} \| CON_{73}^{(192)} \| CON_{74}^{(192)} \| CON_{75}^{(192)})$
$RK_{36}\ RK_{37}\ RK_{38}\ RK_{39}$	$\Sigma^5(L_L) \oplus K_R \oplus (CON_{76}^{(192)} \| CON_{77}^{(192)} \| CON_{78}^{(192)} \| CON_{79}^{(192)})$
$RK_{40}\ RK_{41}\ RK_{42}\ RK_{43}$	$\Sigma^4(L_R) \quad \oplus (CON_{80}^{(192)} \| CON_{81}^{(192)} \| CON_{82}^{(192)} \| CON_{83}^{(192)})$

6.2.6.4 Lược đồ khóa cho khóa 256 bit

Lược đồ khóa cho khóa 256 bit gần như giống hệt với lược đồ khóa 192 bit, ngoại trừ các giá trị hằng số, số RK_i được yêu cầu và khởi tạo của K_R.

Với khóa 256 bit, giá trị của K đặt là 256, các bước thực hiện giống như các bước của khóa 192 bit (xem mô tả trong 6.2.6.3). Sự khác biệt là chúng ta sử dụng $CON_i^{(256)}$ $(0 \le i < 40)$ làm các khóa vòng để sinh L_L, L_R và sau đó sinh $RK_j (0 \le j < 52)$, chúng ta sử dụng 52 giá trị hằng số 32 bit $CON_i^{(256)}$ $(40 \le i < 92)$. Bảng 10 chỉ ra mối quan hệ giữa các khóa vòng được sinh và mối dữ liệu có liên quan.

Bảng 10 – Mở rộng K_L, K_R, L_L, L_R (khóa 1256 bit)

$WK_0\ WK_1\ WK_2\ WK_3$	$K_L \quad \oplus K_R$
$RK_0\ RK_1\ RK_2\ RK_3$	$L_L \quad \oplus (CON_{40}^{(256)} \| CON_{41}^{(256)} \| CON_{42}^{(256)} \| CON_{43}^{(256)})$
$RK_4\ RK_5\ RK_6\ RK_7$	$\Sigma(L_L) \oplus K_R \oplus (CON_{44}^{(256)} \| CON_{45}^{(256)} \| CON_{46}^{(256)} \| CON_{47}^{(256)})$
$RK_8\ RK_9\ RK_{10}\ RK_{11}$	$L_R \quad \oplus (CON_{48}^{(256)} \| CON_{49}^{(256)} \| CON_{50}^{(256)} \| CON_{51}^{(256)})$
$RK_{12}\ RK_{13}\ RK_{14}\ RK_{15}$	$\Sigma(L_R) \oplus K_L \oplus (CON_{52}^{(256)} \| CON_{53}^{(256)} \| CON_{54}^{(256)} \| CON_{55}^{(256)})$
$RK_{16}\ RK_{17}\ RK_{18}\ RK_{19}$	$\Sigma^2(L_L) \quad \oplus (CON_{56}^{(256)} \| CON_{57}^{(256)} \| CON_{58}^{(256)} \| CON_{59}^{(256)})$
$RK_{20}\ RK_{21}\ RK_{22}\ RK_{23}$	$\Sigma^3(L_L) \oplus K_R \oplus (CON_{60}^{(256)} \| CON_{61}^{(256)} \| CON_{62}^{(256)} \| CON_{63}^{(256)})$
$RK_{24}\ RK_{25}\ RK_{26}\ RK_{27}$	$\Sigma^2(L_R) \quad \oplus (CON_{64}^{(256)} \| CON_{65}^{(256)} \| CON_{66}^{(256)} \| CON_{67}^{(256)})$
$RK_{28}\ RK_{29}\ RK_{30}\ RK_{31}$	$\Sigma^3(L_R) \oplus K_L \oplus (CON_{68}^{(256)} \| CON_{69}^{(256)} \| CON_{70}^{(256)} \| CON_{71}^{(256)})$
$RK_{32}\ RK_{33}\ RK_{34}\ RK_{35}$	$\Sigma^4(L_L) \quad \oplus (CON_{72}^{(256)} \| CON_{73}^{(256)} \| CON_{74}^{(256)} \| CON_{75}^{(256)})$
$RK_{36}\ RK_{37}\ RK_{38}\ RK_{39}$	$\Sigma^5(L_L) \oplus K_R \oplus (CON_{76}^{(256)} \| CON_{77}^{(256)} \| CON_{78}^{(256)} \| CON_{79}^{(256)})$
$RK_{40}\ RK_{41}\ RK_{42}\ RK_{43}$	$\Sigma^4(L_R) \quad \oplus (CON_{80}^{(256)} \| CON_{81}^{(256)} \| CON_{82}^{(256)} \| CON_{83}^{(256)})$
$RK_{44}\ RK_{45}\ RK_{46}\ RK_{47}$	$\Sigma^5(L_R) \oplus K_L \oplus (CON_{84}^{(256)} \| CON_{85}^{(256)} \| CON_{86}^{(256)} \| CON_{87}^{(256)})$
$RK_{48}\ RK_{49}\ RK_{50}\ RK_{51}$	$\Sigma^6(L_L) \quad \oplus (CON_{88}^{(256)} \| CON_{89}^{(256)} \| CON_{90}^{(256)} \| CON_{91}^{(256)})$

6.2.6.5 Hàm DoubleSwap

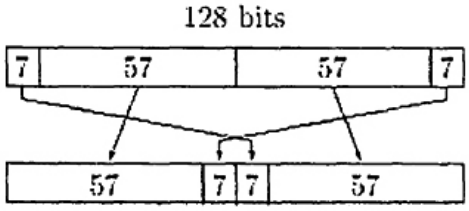
Hàm DoubleSwap $\Sigma: \{0,1\}^{128} \to \{0,1\}^{128}$ được định nghĩa như sau:

$X_{(128)} \mapsto Y_{(128)}$

$Y = X[7\text{-}63] \,||\, X[121\text{-}127] \,||\, X[0\text{-}6] \,||\, X[64\text{-}120],$

Với X[a-b] ký hiệu một xâu bit cắt từ bit thứ a tới bit thứ b của X. Bit 0 là bit có trọng số lớn nhất.

Hàm DoubleSwap được minh họa như Hình 10.



Hình 10 – Hàm DoubleSwap Σ

6.2.6.6 Các giá trị hằng số

Các giá trị hằng số 32 bit CON_i^k được sử dụng trong thuật toán lược đồ khóa. Chúng ta cần 60, 84, 92 giá trị hằng số cho khóa 128, 192, 256 bit tương ứng. Cho $P_{(16)}$= 0xb7e1 (=$(e-2)2^{16}$) và $Q_{(16)}$= 0x243f (=$(\pi-3)2^{16}$) với e là cơ số của logarit tự nhiên (2,71828....) và π (3,14159...). $CON_i^{(k)}$, với k = 128, 192, 256 được sinh theo cách dưới đây. (Xem Bảng 11 là các số lần lặp lại $l^{(k)}$ và giá trị khởi tạo $IV^{(k)}$.

1) $T_0^{(k)} \leftarrow IV^{(k)}$

2) Với $i = 0$ đến $l^{(k)} - 1$ thực hiện các bước sau:

2.1) $CON_{2i}^{(k)} \leftarrow (T_i^{(k)} \oplus P) \,||\, (T_i^{(k)} \lll 1)$

2.2) $CON_{2i+1}^{(k)} \leftarrow (T_i^{(k)} \oplus Q) \,||\, (T_i^{(k)} \lll 8)$

2.3) $T_{i+1}^{(k)} \leftarrow T_i^{(k)} \bullet 0x0002^{-1}$

Tại bước 2.3, phép nhân được thực hiện trên tường GF(2^{16}) xác định bởi đa thức nguyên thủy $z^{16} + z^{15} + z^{13} + z^{11} + z^5 + z^4 + 1$ (= 0x1a831). $0x0002^{-1}$ là hằng số ký hiệu phép nhân nghịch đảo của một phần tử trong trường hữu hạn z (=0x0002).

Bảng 11 – Số các giá trị hằng số được yêu cầu

k	# of $CON_i^{(k)}$	$l^{(k)}$	$IV^{(k)}$
128	60	30	0x428a $(=(\sqrt[3]{2}-1)\cdot 2^{16})$
192	84	42	0x7137 $(=(\sqrt[3]{3}-1)\cdot 2^{16})$
256	92	46	0xb5c0 $(=(\sqrt[3]{5}-1)\cdot 2^{16})$

Bảng 12 – 14 chỉ ra các giá trị của $T_i^{(k)}$ và Bảng 15-17 chỉ ra các giá trị của $CON_i^{(k)}$

Bảng 12— $T_i^{(128)}$

i	0	1	2	3	4	5	6	7
$T_i^{(128)}$	428a	2145	c4ba	625d	e536	729b	ed55	a2b2
i	8	9	10	11	12	13	14	15
$T_i^{(128)}$	5159	fcb4	7e5a	3f2d	cb8e	65c7	e6fb	a765
i	16	17	18	19	20	21	22	23
$T_i^{(128)}$	87aa	43d5	f5f2	7af9	e964	74b2	3a59	c934
i	24	25	26	27	28	29		
$T_i^{(192)}$	649a	324d	cd3e	669f	e757	a7b3		

Bảng 13 — $T_i^{(192)}$

i	0	1	2	3	4	5	6	7
$T_i^{(192)}$	7137	ec83	a259	8534	429a	214d	c4be	625f
i	8	9	10	11	12	13	14	15
$T_i^{(192)}$	e537	a683	8759	97b4	4bda	25ed	c6ee	6377
i	16	17	18	19	20	21	22	23
$T_i^{(192)}$	e5a3	a6c9	877c	43be	21df	c4f7	b663	8f29
i	24	25	26	27	28	29	30	31
$T_i^{(192)}$	938c	49c6	24e3	c669	b72c	5b96	2dcb	c2fd
i	32	33	34	35	36	37	38	39
$T_i^{(192)}$	b566	5ab3	f941	a8b8	545c	2a2e	1517	de93
i	40	41						
$T_i^{(192)}$	bb51	89b0						

Bảng 14 — $T_i^{(256)}$

i	0	1	2	3	4	5	6	7
$T_i^{(256)}$	b5c0	5ae0	2d70	16b8	0b5c	05ae	02d7	d573
i	8	9	10	11	12	13	14	15
$T_i^{(256)}$	bea1	8b48	45a4	22d2	1169	dcac	6e56	372b
i	16	17	18	19	20	21	22	23
$T_i^{(256)}$	cf8d	b3de	59ef	f8ef	a86f	802f	940f	9e1f
i	24	25	26	27	28	29	30	31
$T_i^{(256)}$	9b17	9993	98d1	9870	4c38	261c	130e	0987
i	32	33	34	35	36	37	38	39
$T_i^{(256)}$	d0db	bc75	8a22	4511	f690	7b48	3da4	1ed2
i	40	41	42	43	44	45		
$T_i^{(256)}$	0f69	d3ac	69d6	34eb	ce6d	b32e		

Bảng 15 — $CON_i^{(128)}$

i	0	1	2	3
$CON_i^{(128)}$	f56b7aeb	994a8a42	96a4bd75	fa854521
i	4	5	6	7
$CON_i^{(128)}$	735b768a	1f7abac4	d5bc3b45	b99d5d62
i	8	9	10	11
$CON_i^{(128)}$	52d73592	3ef636e5	c57a1ac9	a95b9b72
i	12	13	14	15
$CON_i^{(128)}$	5ab42554	369555ed	1553ba9a	7972b2a2
i	16	17	18	19
$CON_i^{(128)}$	e6b85d4d	8a995951	4b550696	2774b4fc
i	20	21	22	23
$CON_i^{(128)}$	c9bb034b	a59a5a7e	88cc81a5	e4ed2d3f
i	24	25	26	27
$CON_i^{(128)}$	7c6f68e2	104e8ecb	d2263471	be07c765
i	28	29	30	31
$CON_i^{(128)}$	511a3208	3d3bfbe6	1084b134	7ca565a7
i	32	33	34	35
$CON_i^{(128)}$	304bf0aa	5c6aaa87	f4347855	9815d543
i	36	37	38	39
$CON_i^{(128)}$	4213141a	2e32f2f5	cd180a0d	a139f97a
i	40	41	42	43
$CON_i^{(128)}$	5e852d36	32a464e9	c353169b	af72b274
i	44	45	46	47
$CON_i^{(128)}$	8db88b4d	e199593a	7ed56d96	12f434c9
i	48	49	50	51
$CON_i^{(128)}$	d37b36cb	bf5a9a64	85ac9b65	e98d4d32
i	52	53	54	55
$CON_i^{(128)}$	7adf6582	16fe3ecd	d17e32c1	bd5f9f66
i	56	57	58	59
$CON_i^{(128)}$	50b63150	3c9757e7	1052b098	7c73b3a7

Bảng 16 — $CON_i^{(192)}$

i	0	1	2	3
$CON_i^{(192)}$	c6d61d91	aaf73771	5b6226f8	374383ec
i	4	5	6	7
$CON_i^{(192)}$	15b8bb4c	799959a2	32d5f596	5ef43485
i	8	9	10	11
$CON_i^{(192)}$	f57b7acb	995a9a42	96acbd65	fa8d4d21
i	12	13	14	15
$CON_i^{(192)}$	735f7682	1f7ebec4	d5be3b41	b99f5f62
i	16	17	18	19
$CON_i^{(192)}$	52d63590	3ef737e5	1162b2f8	7d4383a6
i	20	21	22	23
$CON_i^{(192)}$	30b8f14c	5c995987	2055d096	4c74b497
i	24	25	26	27
$CON_i^{(192)}$	fc3b684b	901ada4b	920cb425	fe2ded25
i	28	29	30	31
$CON_i^{(192)}$	710f7222	1d2eeec6	d4963911	b8b77763
i	32	33	34	35
$CON_i^{(192)}$	524234b8	3e63a3e5	1128b26c	7d09c9a6
i	36	37	38	39
$CON_i^{(192)}$	309df106	5cbc7c87	f45f7883	987ebe43
i	40	41	42	43
$CON_i^{(192)}$	963ebc41	fa1fdf21	73167610	1f37f7c4
i	44	45	46	47
$CON_i^{(192)}$	01829338	6da363b6	38c8e1ac	54e9298f
i	48	49	50	51
$CON_i^{(192)}$	246dd8e6	484c8c93	fe276c73	9206c649
i	52	53	54	55
$CON_i^{(192)}$	9302b639	ff23e324	7188732c	1da969c6
i	56	57	58	59
$CON_i^{(192)}$	00cd91a6	6cec2cb7	ec7748d3	8056965b
i	60	61	62	63
$CON_i^{(192)}$	9a2aa469	f60bcb2d	751c7a04	193dfdc2
i	64	65	66	67
$CON_i^{(192)}$	02879532	6ea666b5	ed524a99	8173b35a
i	68	69	70	71
$CON_i^{(192)}$	4ea00d7c	228141f9	1f59ae8e	7378b8a8
i	72	73	74	75
$CON_i^{(192)}$	e3bd5747	8f9c5c54	9dcfaba3	f1ee2e2a
i	76	77	78	79
$CON_i^{(192)}$	a2f6d5d1	ced71715	697242d8	055393de
i	80	81	82	83
$CON_i^{(192)}$	0cb0895c	609151bb	3e51ec9e	5270b089

Bảng 17 — $CON_i^{(256)}$

i	0	1	2	3
$CON_i^{(256)}$	0221947e	6e00c0b5	ed014a3f	8120e05a
i	4	5	6	7
$CON_i^{(256)}$	9a91a51f	f6b0702d	a159d28f	cd78b816
i	8	9	10	11
$CON_i^{(256)}$	bcbde947	d09c5c0b	b24ff4a3	de6eae05
i	12	13	14	15
$CON_i^{(256)}$	b536fa51	d917d702	62925518	0eb373d5
i	16	17	18	19
$CON_i^{(256)}$	094082bc	6561a1be	3ca9e96e	5088488b
i	20	21	22	23
$CON_i^{(256)}$	f24574b7	9e64a445	9533ba5b	f912d222
i	24	25	26	27
$CON_i^{(256)}$	a688dd2d	caa96911	6b4d46a6	076cacdc
i	28	29	30	31
$CON_i^{(256)}$	d9b72353	b596566e	80ca91a9	eceb2b37
i	32	33	34	35
$CON_i^{(256)}$	786c60e4	144d8dcf	043f9842	681edeb3
i	36	37	38	39
$CON_i^{(256)}$	ee0e4c21	822fef59	4f0e0e20	232feff8
i	40	41	42	43
$CON_i^{(256)}$	1f8eaf20	73af6fa8	37ceffa0	5bef2f80
i	44	45	46	47
$CON_i^{(256)}$	23eed7e0	4fcf0f94	29fec3c0	45df1f9e
i	48	49	50	51
$CON_i^{(256)}$	2cf6c9d0	40d7179b	2e72ccd8	42539399
i	52	53	54	55
$CON_i^{(256)}$	2f30ce5c	4311d198	2f91cf1e	43b07098
i	56	57	58	59
$CON_i^{(256)}$	fbd9678f	97f8384c	91fdb3c7	fddc1c26
i	60	61	62	63
$CON_i^{(256)}$	a4efd9e3	c8ce0e13	be66ecf1	d2478709
i	64	65	66	67
$CON_i^{(256)}$	673a5e48	0b1bdbd0	0b948714	67b575bc
i	68	69	70	71
$CON_i^{(256)}$	3dc3ebba	51e2228a	f2f075dd	9ed11145
i	72	73	74	75
$CON_i^{(256)}$	417112de	2d5090f6	cca9096f	a088487b
i	76	77	78	79
$CON_i^{(256)}$	8a4584b7	e664a43d	a933c25b	c512d21e
i	80	81	82	83
$CON_i^{(256)}$	b888e12d	d4a9690f	644d58a6	086cacd3
i	84	85	86	87
$CON_i^{(256)}$	de372c53	b216d669	830a9629	ef2beb34
i	88	89	90	91
$CON_i^{(256)}$	798c6324	15ad6dce	04cf99a2	68ee2eb3

6.3 LEA

6.3.1 Thuật toán LEA

Thuật toán LEA là một mã khối đối xứng, có thể xử lý các khối dữ liệu 128 bit và có ba khóa có độ dài khác nhau: 128, 192, 256 bit [11]. LEA với các khóa 128, 192, 256 bit được gọi là "LEA-128", "LEA-192" và "LEA-256" tương ứng. Số vòng cho CLEFIA là 24 cho LEA-128, 28 cho LEA-192, và 32 cho LEA-256. Các hàm mã hóa và giải mã của LEA yêu cầu 25, 28 hoặc 32 khóa vòng cho LEA-128, LEA-192, LEA-256 tương ứng.

6.3.2 Các ký hiệu riêng cho LEA

Nr	số lượng vòng của thuật toán LEA; 24, 28 hoặc 32 cho độ dài khóa 128, 192 hoặc 256 bit tương ứng
K_i	khóa vòng 192 bit thứ *i* bao gồm 6 từ 32 bit $K_i[0] \parallel K_i[1] \parallel K_i[2] \parallel K_i[3] \parallel K_i[4] \parallel K_i[5]$ với $0 \le i \le Nr$
X_i	trạng thái thứ i bao gồm 4 từ 32 bit $X_i[0] \parallel X_i[1] \parallel X_i[2] \parallel X_i[3]$, $0 \le i \le Nr$
$\delta[i]$	các giá trị hằng số 32 bit được sử dụng để sinh các khóa vòng, $0 \le i \le 8$
$\{0,1\}^n$	tập các xâu nhị phân 32-bit
$\lll i$	phép dịch vòng trái *i*-bit
$\ggg i$	phép dịch vòng phải *i*-bit
$\boxplus$	phép cộng mô-đun 2^n
$\boxminus$	phép trừ mô-đun 2^n

6.3.3 Mã hóa LEA

Cho $P = P[0] \parallel P[1] \parallel P[2] \parallel P[3]$ là khối bản rõ 128 bit và $C = C[0] \parallel C[1] \parallel C[2] \parallel C[3]$ là khối bản mã, với $P[i], C[i] \in \{0,1\}^{32}$ $(0 \le i < 4)$. Cho $K_i = K_i[0] \parallel K_i[1] \parallel K_i[2] \parallel K_i[3] \parallel K_i[4] \parallel K_i[5]$ $(0 \le i \le Nr)$ là các khóa vòng 192 bit được cung cấp bởi lược đồ tại 6.3.5, với $K_i \in \{0,1\}^{32}$ $(0 \le i < 6)$.

Hoạt động mã hóa được mô tả như sau:

(1) $X_0[0] \parallel X_0[1] \parallel X_0[2] \parallel X_0[3] \leftarrow P[0] \parallel P[1] \parallel P[2] \parallel P[3]$

(2) Với *i = 0* đến *(Nr -1)*

$X_{i+1}[0] \leftarrow ((X_i[0] \oplus K_i[0]) \boxplus (X_i[1] \oplus K_i[1])) \lll 9$

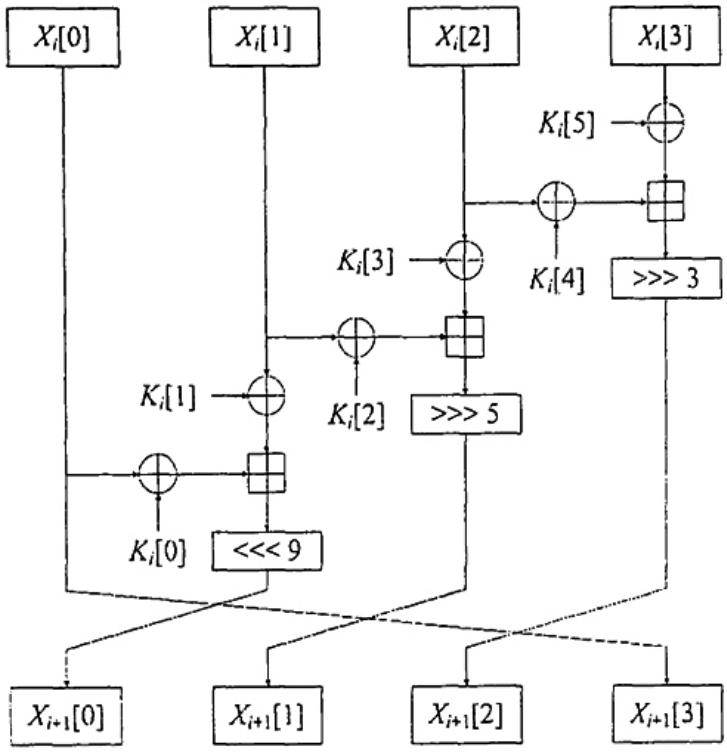
$X_{i+1}[1] \leftarrow ((X_i[1] \oplus K_i[2]) \boxplus (X_i[2] \oplus K_i[3])) \ggg 5$

$X_{i+1}[2] \leftarrow ((X_i[2] \oplus K_i[4]) \boxplus (X_i[3] \oplus K_i[5])) \ggg 3$

$X_{i+1}[3] \leftarrow X_i[0]$

(3) $C[0] \parallel C[1] \parallel C[2] \parallel C[3] \leftarrow X_{Nr}[0] \parallel X_{Nr}[1] \parallel X_{Nr}[2] \parallel X_{Nr}[3]$

Hình 11 đưa ra mã hóa hàm vòng LEA.



Hình 11: Mã hóa hàm vòng LEA

6.3.4 Giải mã LEA

Hoạt động giải mã thực hiện như sau:

(1) $X_{Nr}[0] \parallel X_{Nr}[1] \parallel X_{Nr}[2] \parallel X_{Nr}[3] \leftarrow C[0] \parallel C[1] \parallel C[2] \parallel C[3]$

(2) $i = (Nr - 1)$ đến 0:

$X_i[0] \leftarrow X_{i+1}[3]$

$X_i[1] \leftarrow ((X_{i+1}[0] \ggg 9) \boxminus (X_i[0] \oplus K_i[0])) \oplus K_i[1]$

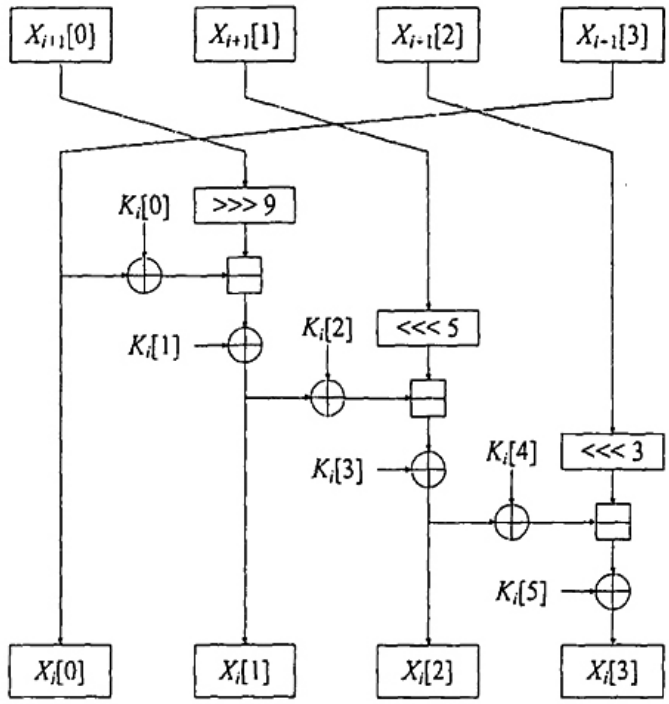
$X_i[2] \leftarrow ((X_{i+1}[1] \lll 5) \boxminus (X_i[1] \oplus K_i[2])) \oplus K_i[3]$

$X_i[3] \leftarrow ((X_{i+1}[2] \lll 3) \boxminus (X_i[2] \oplus K_i 4)) \oplus K_i[5]$

(3) $P[0] \parallel P[1] \parallel P[2] \parallel P[3] \leftarrow X_0[0] \parallel X_0[1] \parallel X_0[2] \parallel X_0[3]$

Hình 12 đưa ra quá trình giải mã hàm vòng LEA.

CHÚ THÍCH Hàm trừ trong Hình 12 có nghĩa là đối số bên trái bị khử từ đối số trên cùng.



Hình 12: giải mã hàm vòng LEA

6.3.5 Lược đồ khóa LEA

6.3.5.1 Cấu trúc chung

Lược đồ khóa LEA nhận các khóa 128, 192, 256 bit và đầu ra là các khóa vòng 192 bit K_i $(0 \leq i < Nr)$ cho các phần xử lý dữ liệu.

6.3.5.2 Lược đồ khóa cho LEA-128

Cho $K = K[0] \| K[1] \| K[2] \| K[3]$ là một khóa 128 bit, với $K[i] \in \{0,1\}^{32}$ $(0 \leq i < 4)$. Lược đồ khóa cho LEA-128 lấy K và 4 giá trị hằng số 32 bit $\delta[i](0 \leq i < 4)$ được mô tả tại 6.3.5.5 làm đầu vào và đầu ra là 24 khóa vòng 192 bit $K[i]$ $(0 \leq i < 24)$.

Lược đồ khóa LEA-128 được thực hiện như sau:

(1) $T[0] \| T[1] \| T[2] \| T[3] \leftarrow K[0] \| K[1] \| K[2] \| K[3]$

(2) Với *i = 0* đến *23*

$T[0] \leftarrow (T[0] \boxplus (\delta[i \bmod 4] \lll i)) \lll 1$

$T[1] \leftarrow (T[1] \boxplus (\delta[i \bmod 4] \lll (i+1))) \lll 3$

$T[2] \leftarrow (T[2] \boxplus (\delta[i \bmod 4] \lll (i+2))) \lll 6$

$T[3] \leftarrow (T[3] \boxplus (\delta[i \bmod 4] \lll (i+3))) \lll 11$

$K_i \leftarrow T[0] \| T[1] \| T[2] \| T[1] \| T[3] \| T[1]$

Thủ tục thực hiện lược đồ khóa LEA được minh họa trong Hình 13.

Phụ lục D
(tham khảo)
Về định danh hàm băm và sự lựa chọn độ dài có thể khôi phục được của thông điệp

Như được quy định trong Điều 6 (Yêu cầu), người dùng các lược đồ chữ ký được quy định trong tiêu chuẩn này phải lựa chọn một hàm băm kháng va chạm *h*. Điều quan trọng là bên xác thực có nhiều cách để xác định hàm băm nào đã được sử dụng khi tạo ra chữ ký, để quá trình xác thực có thể diễn ra một cách an toàn. Nếu có một tổ chức thứ ba nguy hiểm có thể thuyết phục bên xác thực rằng một hàm băm "yếu" đã được sử dụng để tạo ra chữ ký (ví dụ như một hàm băm thiếu tính chất một chiều) thì tổ chức thứ ba này có thể thuyết phục bên xác thực rằng một chữ ký có hiệu lực thực sự đã được áp dụng cho một thông điệp "sai".

Ba lược đồ chữ ký số được quy định trong tiêu chuẩn này cho phép một định danh hàm băm chứa trong giá trị đại diện của thông điệp *F* (xem 8.2.2). Nếu định danh hàm băm đó chứa trong *F* theo cách này thì kẻ tấn công không thể sử dụng lại một cách gian lận một chữ ký đã tồn tại với cùng M_1 và khác M_2, thậm chí ngay cả khi bên xác thực có thể bị thuyết phục chấp nhận các chữ ký đã được tạo ra bằng một hàm băm đủ yếu mà tiền ảnh có thể được tìm thấy. Điều này được cho là có thể giải quyết được vấn đề đã được đề cập đến trong đoạn trước.

Tuy nhiên, như đã thảo luận chi tiết trong [16], ngay cả khi một định danh hàm băm được bao gồm trong giá trị đại diện của thông điệp, kẻ tấn công khác đều có thể xảy ra nếu bên xác thực có thể bị thuyết phục bằng một hàm băm "yếu" đã được sử dụng. Từ yếu ở đây có nghĩa là hàm băm thiếu tính chất một chiều, có nghĩa là với một hàm băm cho trước có thể tính toán để tìm ra xâu đầu vào ánh xạ đến mã băm này bởi hàm băm. (CHÚ THÍCH rằng chính xác thì kiểu yếu này trước tiên phải được thúc đẩy bởi sự bao gồm của một định danh hàm băm trong giá trị đại diện của thông điệp.

Các tấn công được mô tả trong [16] hoạt động theo cách thức chung sau đây. Kẻ tấn công tạo ra "chữ ký" ngẫu nhiên và đối với mỗi "chữ ký" này áp dụng một hàm xác thực công khai của thực thể mà chữ ký của nó muốn giả mạo, và thu được "giá trị đại diện của thông điệp đã được khôi phục" (đây là bước "mở chữ ký"). Phần tiếp theo của tấn công sẽ rất khác nhau tùy thuộc vào định dạng của giá trị đại diện của thông điệp, nhưng bắt buộc kẻ tấn công phải xem xét xem giá trị đại diện của thông điệp đã được khôi phục có định dạng đúng tương ứng với chữ ký thật và rằng định danh hàm băm trong xâu này là định danh tương ứng với một hàm băm yếu hay không. Khả năng xảy ra việc này là rất khác nhau, nhưng có thể lớn đến 2^{-16} (và do đó kẻ tấn công không cần phải thử quá nhiều "chữ ký ngẫu nhiên" trước khi tìm được một chữ ký có các thuộc tính mong muốn).

Với "chữ ký" này, kẻ tấn công có thể nhúng mã băm vào trong giá trị đại diện của thông điệp đã được khôi phục, và lợi dụng thực tế là hàm băm yếu để phát hiện ra một phần thông điệp không thể khôi phục được, mà khi kết hợp với phần có thể khôi phục được có trong giá trị đại diện của thông điệp, băm để có được mã băm mong muốn. Như thế, kẻ tấn công có thể giả mạo một chữ ký mới với một M_1 "ngẫu nhiên". Do đó, kể cả khi có định danh hàm băm trong giá trị đại diện của thông điệp cũng không tránh được việc đòi hỏi người xác thức phải có một phương thức độc lập an toàn để biết được hàm băm nào được sử dụng để xác thực chữ ký.

Thảo luận này cũng liên quan đến việc lựa chọn độ dài có thể khôi phục được c^* cho lược đồ chữ ký 2 và 3. Như đã được mô tả trong 7.2.2, c^* sẽ được lựa chọn thỏa mãn điều kiện $c^* \leq c$, năng lực của lược đồ chữ ký. c^* thường được mong đợi là gần bằng c để tối đa độ dài phần có thể khôi phục được của thông điệp, và do đó tối thiểu độ dài phần không thể khôi phục được của thông điệp. c^* được khuyến nghị lựa chọn là một số bất kỳ nhỏ hơn c (ví dụ như c-16, c-24 hoặc c-80, tùy theo độ khó mong muốn), để làm cho các cuộc tấn công theo hình thức đã được mô tả ở trên trở nên khó khăn hơn.

Phụ lục E
(tham khảo)
Ví dụ

Phụ lục này bao gồm tổng cộng 12 ví dụ về tạo chữ ký và xác thực chữ ký làm việc theo ba lược đồ đã được quy định trong tiêu chuẩn này, cùng với hai ví dụ về tạo khóa.

Phụ lục E.1 bao gồm các ví dụ với số mũ công khai bằng 3.

- E.1.1 bao gồm một ví dụ về tạo khóa.
- E.1.2 bao gồm ba ví dụ về tạo và xác thực chữ ký, tất cả đều là khôi phục toàn bộ thông điệp. Đối với mỗi lược đồ được quy định trong tiêu chuẩn này có một ví dụ tương ứng.
- E.1.3 bao gồm ba ví dụ về tạo và xác thực chữ ký, tất cả đều là khôi phục một phần thông điệp. Đối với mỗi lược đồ được quy định trong tiêu chuẩn này có một ví dụ tương ứng.

Phụ lục E.2 bao gồm các ví dụ với số mũ công khai bằng 2.

- E.2.1 bao gồm một ví dụ về tạo khóa.
- E.2.2 bao gồm ba ví dụ về tạo và xác thực chữ ký, tất cả đều là khôi phục toàn bộ thông điệp. Đối với mỗi lược đồ được quy định trong tiêu chuẩn này có một ví dụ tương ứng.
- E.2.3 bao gồm ba ví dụ về tạo và xác thực chữ ký, tất cả đều là khôi phục một phần thông điệp. Đối với mỗi lược đồ được quy định trong tiêu chuẩn này có một ví dụ tương ứng

E.1 Các ví dụ với số mũ công khai bằng 3

Phụ lục E.1 bao gồm các ví dụ với khóa công khai có số mũ bằng 3.

E.1.1 Ví dụ vê quá trình tạo khóa

Khóa trong ví dụ có mô-đun $k = 1024$ bit với số mũ công khai $v = 3$.

```
p =   FB961451 995C82F9 527CAAAF B3FB4254 6D00A01D 8B2BDE3D 2E7B8F7D 0C9E781E
      B7FABFC8 E86E9F6D ACE3435A 9D043A99 93F3E473 D93FA888 D3577906 77A94931
q =   FF0EAFCA 70585166 A8CD8E90 36E75290 2F32B863 068016B6 A89F2EA3 418882EF
      6F570122 F92D2E9B EFFF7329 1818F251 BF095D6E 208F93CD CEF4767A 568AB241
```

Số đồng dư công khai *n* là kết quả của các thừa số nguyên tố bí mật *p* và *q*. Độ dài của nó là 1024 bit.

```
n =   FAA8ED34 EEF1CE38 D29814B6 EEAA154D C060BB37 EB1A51E8 AB0396DD ADDFD334
      CB9BE20C 087B1DDF 1F78A397 62B5F20A 7A730086 30913CD2 EE60183D E249DD16
      9CA4EB3A E0420E51 13D73050 4A73A926 BEFBFF32 C89858DE 5E5B3899 FEC52521
      04933163 625F2963 5AB8FAA7 AA14C4F3 C0DD2470 DEFCEB39 2429110A C149A771
```

Số mũ của chữ ký bí mật s bằng nghịch đảo phép nhân của *v* mod $lcm(p-1, q-1)$.

```
s =   0A71B48C DF4A1342 5E1BAB87 9F471638 92AEB277 A9CBC369 B1CAD109 3C93FE22
      33267EC0 805A7693 F6A506D0 F9723F6B 1A6F755A ECB0B7DE 1F440102 94186936
      316AAC4B F39B37BF 6105DFA0 AEA60B82 C17306F2 179F2ED4 704D5A6F BCB141C0
      C9380F5A 5C0823CE 67E6ED81 7F8A5100 59E9541B 498C91F4 1ABE8C10 6220E72B
```

E.1.2 Các ví dụ về khôi phục toàn bộ

Ở đây trình bày ba ví dụ về tạo và xác thực chữ ký, mỗi ví dụ tương ứng với một trong ba lược đồ.

E.1.2.1 Ví dụ về lược đồ chữ ký 1

Ví dụ này sử dụng hàm băm chuyên dụng 3 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là SHA-1).

E.1.2.1.1 Quá trình ký

Thông điệp được ký là một xâu bao gồm 64 ký tự mã ASCII như sau.

abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopqopqrpqrs

Trong hệ thập lục phân, thông điệp *M* là một xâu có độ dài là 64 xâu bộ tám, nghĩa là 512 bit như sau.

```
M =   61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
      696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
```

160 bit mã băm được tính toán bằng cách áp dụng SHA-1 cho 512 bit của *M*.

```
H =   79EA0C76 F0056373 FFD6A5AA D389DD90 8B0C0E94
```

Định danh trong trường trailer xác định hàm băm được sử dụng; ISO/IEC 10118-3 thiết lập định danh cho hàm băm chuyên dụng 3 giá trị "33". Do đó, trường trailer *T* bao gồm 16 bit sau đây.

```
T = 33CC
```

Thông điệp là đủ ngắn để khôi phục toàn bộ. 1024 bit của xâu trung gian S_i kết quả của việc nối hai bit của tiêu đề bằng "01", bit dữ liệu thêm được thiết lập bằng '0', 332 (= 1024 – 512 – 160 – 16 – 4) bit đệm bằng '0', bit bao quanh bằng 1, 512 bit của M_1 (=*M*), 160 bit của *H* và 16 bit của trường trailer *T*. Xâu S_r có thể khôi phục kết quả của việc thay thế 82 xâu bộ bốn đệm bằng '0' bằng 82 xâu bộ bốn đệm bằng 'B' và tương tự đối với xâu bộ bốn bao quanh bằng '1' thay thế bằng 'A'.

```
S_r =  4BBBBBBB BBB3BBBB BBBBBBBB BBBBBBB3 BBBBBBB2 BBB3BB3B BBB3BBB5
       BBB6B3B2
       BB3BBBB3 BBB3BBBB BBBA6162 63646263 64656364 65666465 66676566 67686667
       68696768 696A6869 6A6B696A 6B6C6A6B 6C6D6B6C 6D6E6C6D 6E6F6D6E 6F706E6F
       70716F70 71727071 727379EA 0C76F005 6373FFD6 A5AAD389 DD908B0C 0E9433CC
```

Số nguyên có thể khôi phục f_r là số nguyên dương không dấu biểu diễn S_r. f_r tăng theo lũy thừa bậc *s* theo mô-đun *n*. Kết quả được biểu diễn bởi một số nguyên dương không dấu tạm thời *t*.

```
t =   D6369220 6E1FE0A5 7DF603C1 E5EE6025 B4EF2E69 3E8C3C9E BA00C57B 40860A35
      FCA66D68 33795AC1 91191515 FE852CAD C80F315C 86142051 ED322775 9F307934
      421D615F 39792C40 1319F233 CFFD18D0 12D17A02 442E5BBF B17DCFC5 654BEF59
      5F500A15 365CD5D0 BD27948E C938F7C3 BA775992 472E8921 7424A74B 868B63A8
```

Vì kết quả trên lớn hơn $n/2$, chữ ký $\Sigma = n - t$.

```
Σ =   24725B14 80D1ED93 54A210F5 08BBB528 0B718CCE AC8E1549 F1039362 6D59C8FE
      CEF57483 D501C31D 8E5F8E81 6430C55C B263CF29 AA7D1C81 012DF0C8 431963E2
      5A8789DB A6C8E211 00BD3E1C 7A769056 AC2A8530 8469FD1E ACDD68D4 997935C7
      A543274E 2C025392 9D916618 E0DBCD30 0665CABE 97CE6217 B0C469BE 7A2E43C9
```

Thông điệp đã ký có 128 xâu bộ tám chỉ bao gồm chữ ký vì M_2 là rỗng.

E.1.2.1.2 Quá trình xác thực

Phụ lục B
(tham khảo)
Các ví dụ số

Phụ lục này cung cấp các ví dụ số cho PRESENT, CLEFIA và LEA với mỗi độ dài khóa ký hiệu ở hệ thập lục phân.

B.1 Véc tơ kiểm tra PRESENT

B.1.1 PRESENT-80

	Bản rõ Khóa Bản mã	0123456789abcdef 0123456789abcdef f8dd50531d973bde	0123	
Vòng	Giá trị khóa vòng	Sau addRoundKey	Sau sLayer	Sau pLayer
1	0123456789abcdef	0000000000000000	cccccccccccccccc	ffff0000ffff0000
2	1024602468acf135	efdb9fdb68acf135	1278e278a3f425b0	19a22a346eeaa266
3	8f37a2048c048d14	96958830e2ee2f72	eae033bc161162d6	e302a14bee4d0eb2
4	e3c4d1e6f4409181	00c670ad1a0d9f33	cc4adcf75fc7e2bb	de6be5f8135f0bd3
5	62345c789a3cde8a	bc5fb3808963d559	84028b3c3eab700e	8d714149164f90698
6	92460c468b8f1345	1f374d0f9d7615dd	52bd97c2e7da5077	3ab096eb65d3bc6b
7	f37a3248c188d172	c9caa4a3a45b6d19	4e4ff9fbf908a75e	5fd9fa875b8d1fc6
8	0c4d1e6f46491832	5394e4e81dc407f4	0be919135749cd29	741d20ec61425fd5
9	0345c189a3cde8cd	7758e165c28fb718	dd0315a046328d53	c20cc4c61271dc27
10	f460c068b831347d	366c04aeaa40e85a	baa4c9f1ff9c130f	aef11ad1e2c587ed
11	f7a33e8c180d1703	1952245dfac890ee	5e0669072f43ec11	444cd36c59d88553
12	a4d1fef467d18304	e09d27983e090657	1ce76de3b1ceca0d	66bc7e393b9495c1
13	345c149a3fde8cfc	52e16aa3044a193d	0615affbc99f5eb7	0ff6569d4f17377b
14	360c068b829347fd	39fa5016cd847086	be2f0c5a4739dc3a	d51d56ccf163927a
15	fa33e6c180d17055	2f2eb00d71b2e22f	62618cc7d5861662	0ea0e7c6e11711c8
16	fd1fff467cd8301d	f3bf58909dcf21d5	2b8203ece7426570	638003eed6da4446
17	85c15fa3ffe8cf93	e6415c4d29328bd5	1a9504976eb63870	626415d241fab32a
18	a0c070b82bf47ff5	c2a4656a6a0eccdf	46f9a0afafc14472	3be0e16e6bc33152
19	833e54180e170577	b8deb57665d43425	837180daa079b960	8b9c222261aa723c
20	21ffd067ca8301cb	aa63f245ab2973f7	ffab2690f86edb2d	f2dcc4b9fcb6d28c
21	ac15c43ffa0cf95a	5ec8008606ba2bd7	0143cc3aca8f687d	0df52c9b135a5213
22	9c073582b887ff4b	91f21919abddad58	e5265e5ef877f703	85c8dfbcb5bd4abd
23	a3e57380e6b0571b	262dac3c530d1da6	6a67f4b40bc757fa	4a63bd3efa571a5e
24	dffd347cae701cdd	959e894254270683	e0e13e96096dca3b	a65da539ad271a53
25	815c7bffa68f95c2	2701dec70ba88f91	6dc5714dc8f332e5	61e2fba3883e5d39
26	9073702b8f7ff4dd	f1918b880741a9e4	25e53833cd95fe13	24ed70dcab0c5b7b
27	fe57120e6e0571e2	daba62d2c5092a99	7f8fa67640ce6fee	7837d7bfdf1fd204
28	0fd37fcae241cdcd	77e4a8753d5e1fc9	dd19f3d0b701524e	da81ca4b0cc5fed8
29	f5c781fa6ff95c46	2f464bb1633ca29e	629a9885abb4f6e1	3eea811ed0ee2969
30	47373eb8f03f4df1	79ddbfa620d16498	de7782fa6c75a9e3	cb4ef2f277abb235
31	b57108e6e7d71e08	7e3ffa14907cac3d	d1b22f59ecd4f4b7	a5ea86fd3c8be72b
32	5d37d6ae211cdcf5	f8dd50531d973bde		

B.1.2 PRESENT-128

	Bản rõ	0123456789abcdef		
	Khóa	0011223344556677	8899aabbccddeeff	
	Bản mã	88728500054418de		
Vòng	Giá trị khóa vòng	Sau addRoundKey	Sau sLayer	Sau pLayer
1	0011223344556677	01326754cdfeab98	c5b6ad094721f8e3	ad0ed4ca386b6559
2	25133557799bbddf	881de19d41f0d886	335715e7952c733a	02913758d32ffdce
3	29004488cd115599	2b9173d01e3ea857	68e5db7c51b1f30d	6d29bb89a62c1efd
4	63944cd55de66ef7	0ebdf75cfbca700a	c1872d04284fdccf	a45f953f18915419
5	29a4011223344557	8dfb942d3ba5114e	3728e967b8f05591	1ce24b2ceba0c5af
6	178e5133557799ba	0b6c1a1fbed75c15	c8a45f52817d0450	e4909e3625200e72
7	0ca69004488cd114	e8360e326dacdf66	13bac1b6a7f472aa	3aa3097873efe668
8	e35e3944cd55de67	d9fd303cbeba380f	7e27bcb4818fb3c2	4ebad512fa1d9a5c
9	19329a4011223346	57884f52eb3fa91a	0d33920618b2fe5f	486d410f353d78ab
10	488d78e51335577b	00e039ea26082fd0	cc1cbe1f6ac3627c	dd61d5ab0dde2b12
11	d364ca69004488cf	0e051fc20d9aa3dd	c1c05246c7effb77	a0bcabfb057f485f
12	252235e3944cd55f	859e9e1891339d00	30e1e153e5bbe7cc	28bb2acfa9bc9774
13	1d4d9329a4011220	35f6b9e60dbd8554	b02a8e1ac7873009	9da104d0b5588259
14	c99488d78e513356	54358c073b09b10f	09b034cdb8ce85c2	63fa073628916984
15	4975364ca690044b	2a8f317a8e016dcf	6f32b5df31c5a742	4b28c736f98d6fd4
16	ab2652235e3944ce	e00e9515a7b42b1a	1cc1e050fd89685f	68f56acb088992d3
17	7525d4d9329a4015	1dd0be123a13d2c6	577c8156bf5b764a	18d1f36e61dde6f8
18	faac99488d78e517	e27d6a26eca503ef	16d7af6a14f0cb12	2d2c76685f25b4a6
19	17d4975364ca6904	3af8e13b3befdda2	bf2315b8b81277f6	c3c2440ff29fdeae
20	e8eab2652235e390	2b28f66ad0aa3d3e	68632aaf7cffb7b1	477aa1f4bfbe11bf
21	5c5f525d4d9329a1	1b25f3a9f22d381e	58602bfe2667b351	4708a3722ffc861f
22	6ba3aac99488d78b	2cab09bbbb745194	64f8ce8888d905e9	3ff3ec26a4022035
23	a5717d4975364ca3	9a82916fd1346c96	ef36e5a275b9a4ea	ca3bdcc6fbab64f0
24	a5ae8eab2652235b	6f95526dddf947ab	a2e006a7772e9df8	a21f25d6e7f201ce
25	d195c5f525d4d934	738ae023c226d8fa	db3f1c6b466a732f	d51196e9737ff90d
26	e696ba3aac99488b	33872cd3dfe6b186	bb3d647b721a853a	d1191e84ebd3f3a6
27	ad465717d4975362	7c5f49933f44a0c4	d4029eebb299fc49	8fbdc60e17c889b9
28	989a5ae8eab26524	17279ce6fd7aec9d	5d6de41a27df14e7	5932fc7729d3d279
29	e1b5195c5f525d4a	b887e52b76818f33	833d1068da3532bb	91c31290626f78bb
30	0662696ba3aac993	97a17bfbc1c5b128	edf5d82845408563	ed08f8e6a2037845
31	d886d465717d4972	358e2c83d37e3137	b031643b7bd1b5bd	816b0ca5abcab3ff
32	091989a5ae8eab21	88728500054418de		

B.2 Các ví dụ số CLEFIA

B.2.1 CLEFIA với khóa 128 bit

Khóa	ffeeddcc bbaa9988 77665544 33221100
Bản rõ	00010203 04050607 08090a0b 0c0d0e0f
Bản mã	de2bf2fd 9b74aacd f1298555 459494fd
L	8f89a61b 9db9d0f3 93e65627 da0d027e
$WK_{0,1,2,3}$	ffeeddcc bbaa9988 77665544 33221100
$RK_{0,1,2,3}$	f3e6cef9 8df75e38 41c06256 640ac51b
$RK_{4,5,6,7}$	6a27e20a 5a791b90 e8c528dc 00336ea3
$RK_{8,9,10,11}$	59cd17c4 28565583 312a37cc c08abd77
$RK_{12,13,14,15}$	7e8e7eec 8be7e949 d3f463d6 a0aad6aa
$RK_{16,17,18,19}$	e75eb039 0d657eb9 018002e2 9117d009
$RK_{20,21,22,23}$	9f98d11e babee8cf b0369efa d3aaef0d
$RK_{24,25,26,27}$	3438f93b f9cea4a0 68df9029 b869b4a7
$RK_{28,29,30,31}$	24d6406d e74bc550 41c28193 16de4795
$RK_{32,33,34,35}$	a34a20f5 33265d14 b19d0554 5142f434

Bản rõ		00010203 04050607 08090a0b 0c0d0e0f	
Khóa trắng khởi tạo		ffeeddcc bbaa9988	
Sau khi làm trắng		00010203 fbebdbcb 08090a0b b7a79787	
Vòng 1	Đầu vào	00010203 fbebdbcb 08090a0b b7a79787	
	Hàm F	F_0	F_1
	Đầu vào	00010203	08090a0b
	Khóa vòng	f3e6cef9	8df75e38
	Sau cộng khóa	f3e7ccfa	85fe5433
	Sau S	290246e1	777de8e8
	Sau M	547a3193	abf12070
Vòng 2	Đầu vào	af91ea58 08090a0b 1c56b7f7 00010203	
	Hàm F	F_0	F_1
	Đầu vào	af91ea58	1c56b7f7
	Khóa vòng	41c06256	640ac51b
	Sau cộng khóa	ee51880e	785c72ec
	Sau S	cb5d2b0c	63a5edd2
	Sau M	f51cebb3	82dfe347
Vòng 3	Đầu vào	fd15e1b8 1c56b7f7 82dee144 af91ea58	
	Hàm F	F_0	F_1
	Đầu vào	fd15e1b8	82dee144
	Khóa vòng	6a27e20a	5a791b90
	Sau cộng khóa	973203b2	d8a7fad4
	Sau S	c2c7c6c2	be59e10d
	Sau M	d8dfd8de	e15ea81c
Vòng 4	Đầu vào	c4896f29 82dee144 4ecf4244 fd15e1b8	
	Hàm F	F_0	F_1
	Đầu vào	c4896f29	4ecf4244
	Khóa vòng	e8c528dc	00336ea3
	Sau cộng khóa	2c4c47f5	4efc2ce7
	Sau S	9da4dafc	43bce633
	Sau M	b5b28e96	b65c519a

Vòng 5	Đầu vào	376c6fd2 4ecf4244	4b49b022 c4896f29
	Hàm F	F_0	F_1
	Đầu vào	376c6fd2	4b49b022
	Khóa vòng	59cd17c4	28565583
	Sau cộng khóa	6ea17816	631fe5a1
	Sau S	f26ad3e5	62af9f1b
	Sau M	29f08afd	be01d127
Vòng 6	Đầu vào	673fc8b9 4b49b022	7a88be0e 376c6fd2
	Hàm F	F_0	F_1
	Đầu vào	673fc8b9	7a88be0e
	Khóa vòng	312a37cc	c08abd77
	Sau cộng khóa	5615ff75	ba020379
	Sau S	b39c8e58	2dd1e9a2
	Sau M	5999a79e	0429b329
Vòng 7	Đầu vào	12d017bc 7a88be0e	3345dcfb 673fc8b9
	Hàm F	F_0	F_1
	Đầu vào	12d017bc	3345dcfb
	Khóa vòng	7e8e7eec	8be7e949
	Sau cộng khóa	6c5e6950	b8a235b2
	Sau S	8b737025	67a08eba
	Sau M	6ed11b09	dfd3cd32
Vòng 8	Đầu vào	1459a507 3345dcfb	b8ec058b 12d017bc
	Hàm F	F_0	F_1
	Đầu vào	1459a507	b8ec058b
	Khóa vòng	d3f463d6	a0aad6aa
	Sau cộng khóa	c7adc6d1	1846d321
	Sau S	e7ee5a5f	9e97f1a1
	Sau M	8c9d011c	93684eec
Vòng 9	Đầu vào	bfd8dde7 b8ec058b	81b85950 1459a507
	Hàm F	F_0	F_1
	Đầu vào	bfd8dde7	81b85950
	Khóa vòng	e75eb039	0d657eb9
	Sau cộng khóa	58866dde	8cdd27e9
	Sau S	4e821daf	59c56044
	Sau M	e6d6501e	6d5839b4
Vòng 10	Đầu vào	5e3a5595 81b85950	79019cb3 bfd8dde7
	Hàm F	F_0	F_1
	Đầu vào	5e3a5595	79019cb3
	Khóa vòng	018002e2	9117d009
	Sau cộng khóa	5fba5777	e8164cba
	Sau S	612d8f7b	0185a49c
	Sau M	3a1b0e97	b9b479c8
Vòng 11	Đầu vào	bba357c7 79019cb3	066ca42f 5e3a5595
	Hàm F	F_0	F_1
	Đầu vào	bba357c7	066ca42f
	Khóa vòng	9f98d11e	babee8cf
	Sau cộng khóa	243b86d9	bcd24ce0
	Sau S	f70f1144	cb72a481
	Sau M	28974052	4a6700b1
Vòng 12	Đầu vào	5196dce1 066ca42f	145d5524 bba357c7
	Hàm F	F_0	F_1
	Đầu vào	5196dce1	145d5524
	Khóa vòng	b0369efa	d3aaef0d
	Sau cộng khóa	e1a0421b	c7f7ba29
	Sau S	6f7efd4f	72642dce
	Sau M	ffb5db32	907d3820

Số nguyên có thể khôi phục f_r là số nguyên dương không dấu biểu diễn S_r. f_r tăng theo lũy thừa bậc s theo mô-đun n. Kết quả được biểu diễn bởi một số nguyên dương không dấu tạm thời t.

```
t =   F9DD9F72 FAB4AFFC ED3B0538 C5848B27 756AC50C B2890F4C BC268D96 C5E91EE8
      8E3B058F 2EF6585F EF5323CA 4E2C308C C6140CF5 F5357960 5B3BF0CC 621082EB
      77F4A42D 3567355E AA151FB4 652BAFFE 58A4B310 7A064669 FD4177C8 D79F5DE5
      E8C562FF A2D0F5D9 C409AEA0 D5B9F8DF 493AF2F1 8F91D828 C232C4CC 38C13113
```

Xâu nhị phân biểu diễn số nguyên t dưới dạng một số nguyên dương không dấu là chữ ký được tạo ra bởi hàm tạo chữ ký thay thế (xem Phụ lục A.6) $\Sigma' = t$.

Vì kết quả trên lớn hơn $n/2$, chữ ký $\Sigma = n - t$.

```
Σ =   00CB4DC1 F43D1E3B E55D0F7E 29258A26 4AF5F62B 3891429B E8DD0B46 E7F6B44C
      3D60DC7C D984C57F 30257FCD 1489C17D B45EF390 3B5BC372 93242771 B0395A2B
      24B0470D AADAD8F2 69C2109B E547F928 66574C22 4E921274 6119C0D1 2725C73B
      15CDCE63 BF8E3389 96AF4C06 D45ACC14 77A2317F 4F6B1310 55F64C3D CB88765E
```

Thông điệp đã ký có 128 xâu bộ tám chỉ bao gồm chữ ký vì M_2 là rỗng.

E.1.2.3.2 Quá trình xác thực

Chữ ký Σ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 3 mô-đun n, do đó thu được số nguyên f_s.

```
f_s =   7DDD9912 CE7805EC 9E5D0A05 8BA2EE12 8A2B290E 2DDC53FC 01051889 C0C0E01B
        872657A4 CCFEAD1C 24AD33AE 504CE328 0C7D7710 D836C410 7C210BCC 91F68096
        CD9C8244 15AD1AE4 18CF9094 B94D80B9 600F5B68 1BF7334B 5B212B36 EB5AAAAE
        2E8E2958 F5689DF5 80AE1404 4CFE3C4D B616849B A0B8A8AD 26F10275 25B830B5
```

Vì f_s là đồng dư với $(n - 12)$ mô-đun 16, số nguyên được khôi phục là $f'_r = n - f_s$.

```
f'_r =  7CCB5422 2079C84C 343B0AB1 6307273B 36359229 BD3DFDEC A9FE8054 AD1EF319
        44758A67 3B7C70C2 FACB6FE9 12690EE2 6DF58975 585A78C2 723F0C71 50535C80
        8F0868F6 CA94F36C FB079FBB 91262860 5BECA3CA ACA12593 033A0D64 136A7A72
        D6050B0A 6CF68B6D DA0AE6A3 5D1688A6 0AC69FD5 3E44428B FD380E94 DB9176BC
```

f'_r được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục S'_r. Hàm tạo mặt nạ *MGF1* áp dụng cho 856 (= 1024 – 160 – 8) bit bên trái nhất của S'_r, từ đó thu được xâu được khôi phục trung gian S'_i.

```
S'_i =  00000000  00000000  00000000  00000000  00000000  00000000  00000000
        00000000  00000000  00000000  00000000  00000000  00000000  00000000
        00000000  00000000  00000000  00000000  00000000  00000000  00000000
        00000000  00000000  00000000  00000000  00000000  000001A3  5D1688A6
        0AC69FD5 3E44428B FD380E94 DB9176BC
```

S'_i biểu diễn xâu được khôi phục trung gian như sau.

– Bit bên trái nhất của S'_i được thiết lập bằng '0' vì $\delta = 1$ ($\delta = (1 - 1024) mod 8$). 106 xâu bộ tám bên trái nhất của xâu nhị phân còn lại bằng '0'; nó được theo sau bởi xâu bộ tám bao quanh "01"; 107 xâu bộ tám đó được chuyển sang bên trái của S'_i.

– Xâu bộ tám bên phải nhất của S'_i bằng "BC"; xâu bộ tám đó cũng được chuyển sang bên phải của S'_i.

Vì trailer bằng "BC", hàm băm được sử dụng đã được biết đến hoàn toàn; hàm băm chuyên dụng 3 trong ví dụ này.

Xâu còn lại 160 bit được giả định là mã băm *H'* vì không còn thừa dữ liệu.

H' = A35D1688 A60AC69F D53E4442 8BFD380E 94DB9176

Thông điệp đã được khôi phục *M'* được giả định là rỗng và do đó, khôi phục là toàn bộ. Mã băm khác *H''* được tính bằng cách áp dụng SHA-1 cho xâu nhị phân có độ dài 224 (=64+160), kết quả của việc ghép thêm 64 bit của độ dài thông điệp đã được khôi phục *C'* và 160 bit của mã băm của phần thông điệp không thể khôi phục được (phần này bằng rỗng) $h(M_2')$. $H'' = h(C'||h(M_2))$.

H'' = A35D1688 A60AC69F D53E4442 8BFD380E 94DB9176

Vì hai mã băm *H'* và *H''* là giống nhau, chữ ký *Σ* được chấp nhận.

E.1.3 Các ví dụ về khôi phục một phần

Ở đây trình bày ba ví dụ về tạo và xác thực chữ ký, mỗi ví dụ tương ứng với một trong ba lược đồ.

E.1.3.1 Ví dụ về lược đồ chữ ký 1

Ví dụ này sử dụng hàm băm chuyên dụng 1 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là RIPEMD-160).

E.1.3.1.1 Quá trình ký

Ví dụ này mô tả chữ ký của một thông điệp 132 xâu bộ tám, có nghĩa là 1056 bit.

M = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98

160 bit mã băm được tính toán bằng cách áp dụng hàm băm chuyên dụng 1 cho 1056 bit của *M*.

H = F0EA911A F528FA38 777D4B9A 58B6FDA4 2D7E1999

Hàm băm được sử dụng đã được biết đến hoàn toàn. Do đó, trường trailer *T* bao gồm 8 bit sau đây.

T = BC

Thông điệp là quá dài để có thể được khôi phục hoàn toàn bởi quá trình xác thực. Do đó, nó được chia làm hai phần.

– M_1 bao gồm 848 bit bên trái nhất.
– M_2 bao gồm 208 bit còn lại, có nghĩa là 26 xâu bộ tám.

M_1 = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
FEDCBA98 76543210 FEDC

M_2 = BA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98

1024 bit của xâu trung gian S_i kết quả của việc ghép thêm hai bit tiêu đề bằng "01", bit dữ liệu thêm bằng '1', bốn (= 1024 – 848 – 160 – 8 – 4) bit đệm bằng '0', bit bao quanh bằng 1, 848 bit của M_1, 160 bit của H và 8 bit của trường trailer T. Xâu có thể khôi phục S_r kết quả của việc xâu bộ bốn bao quanh bằng 1 được thay thế bằng 'A'.

```
Sr =  6AFEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
      10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
      10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
      10FEDCBA 98765432 10FEDCF0 EA911AF5 28FA3877 7D4B9A58 B6FDA42D 7E1999BC
```

Số nguyên có thể khôi phục f_r là số nguyên dương không dấu biểu diễn S_r. f_r tăng theo lũy thừa bậc s theo mô-đun n. Kết quả được biểu diễn bởi một số nguyên dương không dấu tạm thời t.

```
t =   C9DE5B79 67CFD8BE 506749A2 F2E5035C 9C2C5E94 3DD46838 AEF7144E A01283F0
      95C35FE5 53A87553 AEADBBCE 2B9876EC 14EA5C31 EA11BCC1 F33E5161 7B4B73C2
      38EB6D4C AA3DF32D 1434E846 B2E74146 E24C7171 D2A0FBED 77E37371 1444360B
      962A9C27 D9CC2E15 4FE30BEC A3E20B4C 0CCF472F 70E64A9C 9FFAA56A 98BC1079
```

Vì kết quả trên lớn hơn $n/2$, chữ ký $\Sigma = n - t$.

```
Σ =   30CA91BB 8721F57A 8230CB13 FBC511F1 24345CA3 AD45E9AF FC0C848F 0DCD4F44
      35D68226 B4D2A88B 70CAE7C9 371D7B1E 6568A454 467F8010 FB21C6DC 66FE6954
      63B97DEE 36041B23 FFA24809 678C67DF DCAF8DC0 F5F75CF0 E677C528 EA80EF15
      6E68953B 8892FB4E 0AD5EE3B 0632B9A7 B40DDD41 6E16A09C 842E6B9F 688D96F9
```

Thông điệp đã ký có 128 xâu bộ tám chữ ký Σ cùng với 26 xâu bộ tám của thông điệp không thể khôi phục được M_2, có nghĩa là chỉ nhiều hơn 22 xâu bộ tám so với thông điệp M.

E.1.3.1.2 Quá trình xác thực

Chữ ký Σ là một xâu nhị phân đại diện một số nguyên dương không dấu, nhỏ hơn $n/2$. Số nguyên này tăng theo lũy thừa bậc 3 mô-đun n, do đó thu được số nguyên f_s.

```
fs =  8FAA107A 567B7A06 C19937FC 5633C11B AF61DE7D 52A3FDB6 9A04BC23 15697F02
      BA9D0551 7004C9AD 0E79C6DC CA3F9DD8 697423CB 981AE8A0 DD613B83 49D388E4
      5BA60E80 47CBBA1F 02D85395 B1FD54F4 ADFD2276 302204AC 4D5C5BDF 664ED0EE
      F39454A8 C9E8D531 49BA1DB6 BF83A9FE 97E2EBF9 61B150E0 6D2B6CDC 63300DB5
```

Vì f_s là đồng dư với $(n - 12)$ mô-đun 16, nó được thay thế bởi phần dư của nó với n, có nghĩa là số nguyên được khôi phục là $f'_r = n - f_s$.

```
f'r =  6AFEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
       10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
       10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432 10FEDCBA 98765432
       10FEDCBA 98765432 10FEDCF0 EA911AF5 28FA3877 7D4B9A58 B6FDA42D 7E1999BC
```

f'_r được biểu diễn dưới dạng một số nguyên dương không dấu bởi xâu đã được khôi phục S'_r.

– Xâu bộ tám bên trái nhất của S'_r bằng "6A"; nó bao gồm tiêu đề bằng "01", bit dữ liệu thêm bằng '1' (khôi phục một phần), một bit đệm bằng '0' và một xâu bộ bốn đệm bằng 'A'; xâu bộ tám này được chuyển sang bên trái của S'_r.

– Xâu bộ tám bên phải nhất của S'_T bằng "BC"; xâu bộ tám đó cũng được chuyển sang bên phải của S'_T.

Vì trailer bằng "BC"; hàm băm được sử dụng đã được biết đến hoàn toàn, hàm băm chuyên dụng 1 trong ví dụ này.

Xâu còn lại 1008 bit được chia làm hai phần.

– M_1^* bao gồm 848 bit bên trái nhất.
– H' bao gồm 160 bit bên phải nhất.

```
M1* = FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
      FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
      FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
      FEDCBA98 76543210 FEDC

H' =  F0EA911A F528FA38 777D4B9A 58B6FDA4 2D7E1999
```

Vì khôi phục là một phần, thông điệp đã được khôi phục M^* bao gồm M_1^* và M_2^*, phần có thể và không thể khôi phục được.

```
M* =  FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
      FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
      FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
      FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210 FEDCBA98 76543210
      FEDCBA98
```

Mã băm còn lại H'' được tính bằng các áp dụng hàm băm chuyên dụng 1 cho M^*.

```
H'' = F0EA911A F528FA38 777D4B9A 58B6FDA4 2D7E1999
```

Vì hai mã băm H' và H'' là giống nhau, chữ ký Σ được chấp nhận.

E.1.3.2 Ví dụ về lược đồ chữ ký 2

Ví dụ này sử dụng hàm băm chuyên dụng 3 trong ISO/IEC 10118-3 (còn được biết đến với tên gọi là SHA-1).

E.1.3.2.1 Quá trình ký

Thông điệp để ký là xâu 112 ký tự mã ASCII sau đây.

```
abcdbcdecdefdefgefghfghighijhijkijkljklmklmnlmnomnopnopq
opqrpqrsqrstrstustuvtuvwuvwxvwxywxyzxyzayzabzabcabcdbcde
```

Trong hệ thập lục phân, thông điệp M là xâu bộ tám có độ dài 112 xâu bộ tám, có nghĩa là 896 bit sau đây.

```
M =  61626364 62636465 63646566 64656667 65666768 66676869 6768696A 68696A6B
     696A6B6C 6A6B6C6D 6B6C6D6E 6C6D6E6F 6D6E6F70 6E6F7071 6F707172 70717273
     71727374 72737475 73747576 74757677 75767778 76777879 7778797A 78797A61
     797A6162 7A616263 61626364 62636465
```

160 bit của salt S được tạo ra.

```
S =  4C95C1B8 7A1DE8AC C193C14C F3147FE9 C6636078
```

Vòng 13	Đầu vào	9d4e3a7e 1e0f2d96	7e9359f3 e0de260a
	Hàm F	F_0	F_1
	Đầu vào	9d4e3a7e	7e9359f3
	Khóa vòng	6e3ff82a	74ac3ffd
	Sau cộng khóa	f371c254	0a3f660e
	Sau S	29ea68e8	b4f530a8
	Sau M	17524741	4b8c607e
Vòng 14	Đầu vào	095d6ad7 7e9359f3	ab524674 9d4e3a7e
	Hàm F	F_0	F_1
	Đầu vào	095d6ad7	ab524674
	Khóa vòng	b9696e2e	cc0b3a38
	Sau cộng khóa	b03404f9	67597c4c
	Sau S	152a2f03	52161e39
	Sau M	f7ee818b	7902f3eb
Vòng 15	Đầu vào	897dd878 ab524674	e44cc995 095d6ad7
	Hàm F	F_0	F_1
	Đầu vào	897dd878	e44cc995
	Khóa vòng	ed785cbd	9c077c13
	Sau cộng khóa	640584c5	784bb586
	Sau S	459d9e10	636b5a11
	Sau M	4034defc	0228bdd4
Vòng 16	Đầu vào	eb669888 e44cc995	0b75d703 897dd878
	Hàm F	F_0	F_1
	Đầu vào	eb669888	0b75d703
	Khóa vòng	04978d83	2ec058ba
	Sau cộng khóa	eff1150b	25b58fb9
	Sau S	90e4ee38	e7691f3b
	Sau M	4a679609	05b2b4a9
Vòng 17	Đầu vào	ae2b4f9c 0b75d703	8ccf6cd1 eb669888
	Hàm F	F_0	F_1
	Đầu vào	ae2b4f9c	8ccf6cd1
	Khóa vòng	4bbd5f6a	31fe8de8
	Sau cộng khóa	e59610f6	bd31e139
	Sau S	f6a5286d	b15d7589
	Sau M	7z0df49d	bad65e22
Vòng 18	Đầu vào	7978239e 8ccf6cd1	51b0c6aa ae2b4f9c
	Hàm F	F_0	F_1
	Đầu vào	7978239e	51b0c6aa
	Khóa vòng	b76da574	3a6fa8e7
	Sau cộng khóa	ce1586ea	6bdf6e4d
	Sau S	919c117f	283aaa43
	Sau M	ef24fe56	08916103
Vòng 19	Đầu vào	63eb9287 51b0c6aa	a6ba2e9f 7978239e
	Hàm F	F_0	F_1
	Đầu vào	63eb9287	a6ba2e9f
	Khóa vòng	521213ce	4f1f59d8
	Sau cộng khóa	31f98149	e9a57747
	Sau S	5d03e265	3c8d7bda
	Sau M	b7464b63	e1d086a7

Vòng 20	Đầu vào	e6f68dc9	a6ba2e9f	98a8a539	63eb9287
	Hàm F	F_0		F_1	
	Đầu vào	e6f68dc9		98a8a539	
	Khóa vòng	c13624f6		ee91f6a4	
	Sau cộng khóa	27c0a93f		7639539d	
	Sau S	20b5938b		09893194	
	Sau M	3cae819e		b603c454	
Vòng 21	Đầu vào	9a14af01	98a8a539	d5e856d3	e6f68dc9
	Hàm F	F_0		F_1	
	Đầu vào	9a14af01		d5e856d3	
	Khóa vòng	17f68fde		f6c360a9	
	Sau cộng khóa	8de220df		232b367a	
	Sau S	6666bff2		b383a1bd	
	Sau M	7ae08a5d		662b2c4d	
Vòng 22	Đầu vào	e2482f64	d5e856d3	80dda184	9a14af01
	Hàm F	F_0		F_1	
	Đầu vào	e2482f64		80dda184	
	Khóa vòng	6288bc72		c0ad856b	
	Sau cộng khóa	80c09316		407024ef	
	Sau S	cdb5f1e5		fbe99290	
	Sau M	3d9dac60		108259db	
Đầu ra		e2482f64	e875fab3	80dda184	8a96f6da
Khóa trắng cuối cùng			77777777		77777777
Sau khóa trắng		e2482f64	9f028dc4	80dda184	fde181ad
Bản mã		e2482f64	9f028dc4	80dda184	fde181ad

B.2.3 CLEFIA với khóa 256 bit

Khóa	ffeeddcc	bbaa9988	77665544	33221100
	f0e0d0c0	b0a09080	70605040	30201000
Bản rõ	00010203	04050607	08090a0b	0c0d0e0f
Bản mã	a1397814	289de80c	10da46d1	fa48b38a
L_L	477e8f09	66ee5378	2cc2be04	bf55e28f
L_R	d6c10b89	4eeab575	84bd5663	cc933940
$WK_{0,1,2,3}$	0f0e0d0c	0b0a0908	07060504	03020100
$RK_{0,1,2,3}$	58f02029	15413cd0	1b0c41a4	e4bacd0f
$RK_{4,5,6,7}$	6c498393	9846231b	1fc716fc	7c81a45b
$RK_{8,9,10,11}$	fa37c259	0e3da2ee	aacf9abb	8ec0aad9
$RK_{12,13,14,15}$	b05bd737	8de1f2d0	8ffee0f6	b70b47ea
$RK_{16,17,18,19}$	581b3e34	03263f89	2f7100cd	05cee171
$RK_{20,21,22,23}$	b523d4e9	176d7c44	6d7ba5d7	f797b2f3
$RK_{24,25,26,27}$	25d80df2	a646bba2	6a3a95e1	3e3a47f0
$RK_{28,29,30,31}$	b304eb20	44f8824e	c7557cbc	47401e21
$RK_{32,33,34,35}$	d71ff7e9	aca1fb0c	2deff35d	6ca3a830
$RK_{36,37,38,39}$	4dd7cfb7	ae71c9f6	4e911fef	90aa95de
$RK_{40,41,42,43}$	2c664a7a	8cb5cf6b	14c8de1e	43b9caef
$RK_{44,45,46,47}$	568c5a33	07ef7ddd	609dc860	ac9e50f8
$RK_{48,49,50,51}$	c0c18358	4f53c80e	33e01cb9	80251e1c

Bản rõ		00010203	04050607	08090a0b	0c0d0e0f
Khóa trắng khởi tạo			0f0e0d0c		0b0a0908
Sau khi làm trắng		00010203	0b0b0b0b	08090a0b	07070707
Vòng 1	Đầu vào	00010203	0b0b0b0b	08090a0b	07070707
	Hàm F	F_0		F_1	
	Đầu vào	00010203		08090a0b	
	Khóa vòng	58f02029		15413cd0	
	Sau cộng khóa	58f1222a		1d4836db	
	Sau S	4ee41927		2c78a1ac	
	Sau M	2db2101b		d87ee718	
Vòng 2	Đầu vào	26b91b10	08090a0b	df79e01f	00010203
	Hàm F	F_0		F_1	
	Đầu vào	26b91b10		df79e01f	
	Khóa vòng	1b0c41a4		e4bacd0f	
	Sau cộng khóa	3db55ab4		3bc32d10	
	Sau S	aa5afadb		0f1e1928	
	Sau M	317e029c		c0cc96ba	
Vòng 3	Đầu vào	39770897	df79e01f	c0cd94b9	26b91b10
	Hàm F	F_0		F_1	
	Đầu vào	39770897		c0cd94b9	
	Khóa vòng	6c498393		8846231b	
	Sau cộng khóa	553e8b04		488bb7a2	
	Sau S	5487484e		d84876a0	
	Sau M	c3a7ac1d		7ae05884	
Vòng 4	Đầu vào	1cde4c02	c0cd94b9	5c594394	39770897
	Hàm F	F_0		F_1	
	Đầu vào	1cde4c02		5c594394	
	Khóa vòng	1fc716fc		7c81a45b	
	Sau cộng khóa	03195afe		20d8e7cf	
	Sau S	c607fa95		12f002c9	
	Sau M	5edee0ce		4cfb0e90	
Vòng 5	Đầu vào	9e137477	5c594394	758c0607	1cde4c02
	Hàm F	F_0		F_1	
	Đầu vào	9e137477		758c0607	
	Khóa vòng	fa37c259		0e3da2ee	
	Sau cộng khóa	6424b62e		7bb1a4e9	
	Sau S	4592c8d2		46f3a044	
	Sau M	adfd33ae		42450650	
Vòng 6	Đầu vào	f1a4703a	758c0607	5e9b4a52	9e137477
	Hàm F	F_0		F_1	
	Đầu vào	f1a4703a		5e9b4a52	
	Khóa vòng	aacf9abb		8ec0aad9	
	Sau cộng khóa	5b6bea81		d05be08b	
	Sau S	22285e04		f822d448	
	Sau M	0fa52ed4		aa7a0a9c	
Vòng 7	Đầu vào	7a2928d3	5e9b4a52	34697eeb	f1a4703a
	Hàm F	F_0		F_1	
	Đầu vào	7a2928d3		34697eeb	
	Khóa vòng	b05bd737		8de1f2d0	
	Sau cộng khóa	ca72ffe4		b9888c3b	
	Sau S	23ed8e68		172b59c0	
	Sau M	8b158630		334e2af2	

Vòng 8	Đầu vào	d58ecc62 34697eeb	c2ea5ac8 7a2928d3
	Hàm F	F_0	F_1
	Đầu vào	d58ecc62	c2ea5ac8
	Khóa vòng	8ffee0f6	b70b47ea
	Sau cộng khóa	5a702c94	75e11d22
	Sau S	facf9d64	586f2c19
	Sau M	72c2027e	a582d5f0
Vòng 9	Đầu vào	46ab7c95 c2ea5ac8	dfabfd23 d58ecc62
	Hàm F	F_0	F_1
	Đầu vào	46ab7c95	dfabfd23
	Khóa vòng	581b3e34	03263f89
	Sau cộng khóa	1eb042a1	dc8dc2aa
	Sau S	177afd6a	57664735
	Sau M	51d5740a	110287d7
Vòng 10	Đầu vào	933f2ec2 dfabfd23	c48c4bb5 46ab7c95
	Hàm F	F_0	F_1
	Đầu vào	933f2ec2	c48c4bb5
	Khóa vòng	2f7100cd	05cee171
	Sau cộng khóa	bc4e2e0f	c142aac4
	Sau S	e0434cd9	22fd2380
	Sau M	a768d32a	b6ae4f2b
Vòng 11	Đầu vào	78c32e09 c48c4bb5	f00533be 933f2ec2
	Hàm F	F_0	F_1
	Đầu vào	78c32e09	f00533be
	Khóa vòng	b523d4e9	176d7c44
	Sau cộng khóa	cde0fae0	e7684ffa
	Sau S	3fd410d4	02ef5310
	Sau M	08bd9b01	2fdb3f65
Vòng 12	Đầu vào	cc31d0b4 f00533be	bce411a7 78c32e09
	Hàm F	F_0	F_1
	Đầu vào	cc31d0b4	bce411a7
	Khóa vòng	6d7ba5d7	f797b2f3
	Sau cộng khóa	a14a7563	4b73a354
	Sau S	1b512562	c94a71eb
	Sau M	7c2c762b	81ca0b59
Vòng 13	Đầu vào	8c294595 bce411a7	f9092550 cc31d0b4
	Hàm F	F_0	F_1
	Đầu vào	8c294595	f9092550
	Khóa vòng	25d80df2	a646bba2
	Sau cộng khóa	a9f14867	5f4f9ef2
	Sau S	93e47852	5c26cae5
	Sau M	4a87c858	54bc68d5
Vòng 14	Đầu vào	f663d9ff f9092550	988db861 8c294595
	Hàm F	F_0	F_1
	Đầu vào	f663d9ff	988db861
	Khóa vòng	6a3a95e1	3e3a47f0
	Sau cộng khóa	9c594c1e	a6b7ff91
	Sau S	58ff39b0	054d1d75
	Sau M	d82301d4	085d5025

Vòng 15	Đầu vào	212a2484 988db861 847415b0 f663d9ff	
	Hàm F	F_0	F_1
	Đầu vào	212a2484	847415b0
	Khóa vòng	b304eb20	44f8824e
	Sau cộng khóa	922ecfa4	c08c97fe
	Sau S	86d2c9a0	b5ff567d
	Sau M	dbf56073	87e2a6a2
Vòng 16	Đầu vào	4378d812 847415b0 71817f5d 212a2484	
	Hàm F	F_0	F_1
	Đầu vào	4378d812	71817f5d
	Khóa vòng	c7557cbc	47401e21
	Sau cộng khóa	842da4ae	36c1617c
	Sau S	9e19b889	a10c5414
	Sau M	6791a3e3	e177d3a8
Vòng 17	Đầu vào	e3e5b653 71817f5d c05df72c 4378d812	
	Hàm F	F_0	F_1
	Đầu vào	e3e5b653	c05df72c
	Khóa vòng	d71ff7e9	aca1fb0c
	Sau cộng khóa	34fa41ba	6cfc0c20
	Sau S	d4e1be2d	32bc13bf
	Sau M	2743ef2d	6fec0aab
Vòng 18	Đầu vào	56c29070 c05df72c 2c94d2b9 e3e5b653	
	Hàm F	F_0	F_1
	Đầu vào	56c29070	2c94d2b9
	Khóa vòng	2deff35d	6ca3a830
	Sau cộng khóa	7b2d632d	40377a89
	Sau S	56193719	fb13c1b7
	Sau M	ee6316fa	5e3245b7
Vòng 19	Đầu vào	2e3ee1d6 2c94d2b9 bdd7f3e4 56c29070	
	Hàm F	F_0	F_1
	Đầu vào	2e3ee1d6	bdd7f3e4
	Khóa vòng	4dd7cfb7	ae71c9f6
	Sau cộng khóa	63e92e61	13a63a12
	Sau S	373c4c54	8fe6c54b
	Sau M	87aab08e	8f8d16f3
Vòng 20	Đầu vào	ab3e6237 bdd7f3e4 d94f8683 2e3ee1d6	
	Hàm F	F_0	F_1
	Đầu vào	ab3e6237	d94f8683
	Khóa vòng	4e911fef	90aa95de
	Sau cộng khóa	e5af7dd8	49e5135d
	Sau S	f6ad88be	65f68f77
	Sau M	0889df33	f418c84f
Vòng 21	Đầu vào	b55e2cd7 d94f8683 da262999 ab3e6237	
	Hàm F	F_0	F_1
	Đầu vào	b55e2cd7	da262999
	Khóa vòng	2c664a7a	8cb5cf6b
	Sau cộng khóa	993866ad	5693e6f2
	Sau S	2c2b6cee	0df150e5
	Sau M	8999e772	da5415d2

Vòng 22	Đầu vào	50d661f1 da262999	716a77e5 b55e2cd7
	Hàm F	F_0	F_1
	Đầu vào	50d661f1	716a77e5
	Khóa vòng	14c8de1e	43b9caef
	Sau cộng khóa	441ebfef	32d3bd0a
	Sau S	12b052ac	c7bbb182
	Sau M	f5efd89e	744a9ced
Vòng 23	Đầu vào	2fc9f107 716a77e5	c114b03a 50d661f1
	Hàm F	F_0	F_1
	Đầu vào	2fc9f107	c114b03a
	Khóa vòng	568c5a33	07ef7ddd
	Sau cộng khóa	7945ab34	c6fbcde7
	Sau S	a2a77e2a	4cd7e238
	Sau M	e84f6d9b	ce67e20a
Vòng 24	Đầu vào	99251a7e c114b03a	9eb183fb 2fc9f107
	Hàm F	F_0	F_1
	Đầu vào	99251a7e	9eb183fb
	Khóa vòng	608dc860	ac9e50f8
	Sau cộng khóa	f9a8d21e	322fd303
	Sau S	f84572b0	c7d8f1c6
	Sau M	20634b77	591b3f55
Vòng 25	Đầu vào	e177fb4d 9eb183fb	76d2ce52 99251a7e
	Hàm F	F_0	F_1
	Đầu vào	e177fb4d	76d2ce52
	Khóa vòng	c0c18358	4f53c80e
	Sau cộng khóa	21b67815	3981065c
	Sau S	a14dd39c	c8e20aa5
	Sau M	3f88fbef	89ff5caf
Vòng 26	Đầu vào	a1397814 76d2ce52	10da46d1 e177fb4d
	Hàm F	F_0	F_1
	Đầu vào	a1397814	10da46d1
	Khóa vòng	33e01cb9	80251e1c
	Sau cộng khóa	92d964ad	90ff58cd
	Sau S	864445ee	9a8e803f
	Sau M	5949235a	183d49c7
Đầu ra		a1397814 2f9bed08	10da46d1 f94ab28a
Khóa trắng cuối cùng		07060504	03020100
Sau khi làm trắng		a1397814 289de80c	10da46d1 fa48b38a
Bản mã		a1397814 289de80c	10da46d1 fa48b38a

B.3 Các ví dụ số LEA

B.3.1 LEA-128

Khóa K	3c2d1e0f 78695a4b b4a59687 f0e1d2c3
K_0	003a0fd4 02497010 194f7db1 02497010 090d0883 02497010
K_1	11fdcbb1 9e98e0c8 18b570cf 9e98e0c8 9dc53a79 9e98e0c8
K_2	f30f7bb5 6d6628db b74e5dad 6d6628db a65e46d0 6d6628db
K_3	74120631 dac9bd17 cd1ecf34 dac9bd17 540f76f1 dac9bd17
K_4	662147db c637c47a 46518932 c637c47a 23269260 c637c47a
K_5	e4dd5047 f694285e e1c2951d f694285e 8ca5242c f694285e
K_6	baf8e5ca 3e936cd7 0fc7e5b1 3e936cd7 f1c8fa8c 3e936cd7
K_7	5522b80c ee22ca78 8a6fa8b3 ee22ca78 65637b74 ee22ca78
K_8	8a19279e 6fb40ffe 85c5f092 6fb40ffe 92cc9f25 6fb40ffe
K_9	9dde584c cb00c87f 4780ad66 cb00c87f e61b5dcb cb00c87f
K_{10}	4fa10466 f728e276 d255411b f728e276 656839ad f728e276
K_{11}	9250d058 51bd501f 1cb40dae 51bd501f 1abf218d 51bd501f
K_{12}	21dd192d 77c644e2 cabfaa45 77c644e2 681c207d 77c644e2
K_{13}	de7ac372 9436afd0 10331d80 9436afd0 f326fe99 9436afd0
K_{14}	fb3ac3d4 93df660e 2f65d8a3 93df660e df92e761 93df660e
K_{15}	27620087 265ef76e 4fb29864 265ef76e 2656ed1a 265ef76e
K_{16}	227b88ec d0b3fa6f c86a08fd d0b3fa6f a864cba9 d0b3fa6f
K_{17}	f1002361 e5e85fc3 1f0b0408 e5e85fc3 488e7ac4 e5e85fc3
K_{18}	c65415d5 51e176b6 eca88bf9 51e176b6 edb89ece 51e176b6
K_{19}	9b6fb99c 0548254b 8de9f7c2 0548254b b6b4d146 0548254b

Khóa K	3c2d1e0f 78695a4b b4a59687 f0e1d2c3
K_{20}	7257f134 06051a42 36bcef01 06051a42 b649d524 06051a42
K_{21}	a540fb03 34b196e6 f7c80dad 34b196e6 71bc7dc4 34b196e6
K_{22}	8fbee745 cf744123 907c0a60 cf744123 8215ec35 cf744123
K_{23}	0bf6adba df69029d 5b72305a df69029d cb47c19f df69029d

Khóa K	3c2d1e0f 78695a4b b4a59687 f0e1d2c3
Bản rõ P	13121110 17161514 1b1a1918 1f1e1d1c
Bản mã C	354ec89f 18c6c628 a7c73255 fd8b6404
X_0	13121110 17161514 1b1a1918 1f1e1d1c
X_1	0f079051 693d668d e5edcfd4 13121110
X_2	3fc44a2d f767ea2a a0b67cf0 0f079051
X_3	99e912cd 906fd05d 4d293e55 3fc44a2d
X_4	43048c71 5faa8d15 dfc687fb 99e912cd
X_5	862a337d 419f623d 4b97dd8a 43048c71
X_6	055b3a34 a2eb0f67 af9873ba 862a337d
X_7	38875cb8 19f1c052 02e13d1c 055b3a34
X_8	f1ddbcca 2c031302 8a5f86d6 38875cb8
X_9	f770a17e c47d9365 2df8cda7 f1ddbcca
X_{10}	58a898f4 db57aa1e 20d820a4 f770a17e
X_{11}	11c9f487 bf079d6e 28c10b82 58a898f4
X_{12}	a7e4a0e4 e8e97f62 47727e5f 11c9f487
X_{13}	d1ea924a 2298587f f2afc1d0 a7e4a0e4
X_{14}	7e91cf8c fcca259f 86ab69cf d1ea924a
X_{15}	809fd3e9 ef492067 536df05e 7e91cf8c
X_{16}	2b54eee2 98b175f9 d9c14ac4 809fd3e9
X_{17}	63eb48a2 7ad2716d 783a355e 2b54eee2
X_{18}	4b34e264 101d5f00 7fee2017 63eb48a2
X_{19}	ba42cf9e d156295c b88c1f9d 4b34e264
X_{20}	970433ea a0d420cb 4b96b2c1 ba42cf9e
X_{21}	49facf18 6f1fe3c2 3744e7b8 970433ea
X_{22}	d1527e90 8ce66afe 1d55c7f1 49facf18
X_{23}	fd8b6404 8675df3b e4b9d73f d1527e90
X_{24}	354ec89f 18c6c628 a7c73255 fd8b6404

B.3.2 LEA-192

Khóa K	3c2d1e0f 78695a4b b4a59687 f0e1d2c3 c3d2e1f0 8796a5b4
K_0	003a0fd4 02497010 194f7db1 090d0883 2ff5805a c2580b27
K_1	11fdcbb1 9e98e0c8 18b570cf 9dc53a79 5c145788 9771b5e5
K_2	f30f7bb5 6d6628db b74e5dad a65e46d0 6f44da96 f643115f
K_3	74120631 dac9bd17 cd1ecf34 540f76f1 aa1a5bdb fbafaae7
K_4	13f8a031 34f28728 31fdb409 0e31481b df498117 cf9371f1
K_5	0967c312 b3484ec8 3aae5b3d 5a9714a0 b2d4dd5f 3a1fcdf7
K_6	0ac47404 59e9e54d a60dc00a 566139d3 898dce4f 582d72dd

Khóa K	3c2d1e0f 78695a4b b4a59687 f0e1d2c3 c3d2e1f0 8796a5b4
K_7	77f3ea4c e2a73c6d b6f1249a 6a172700 bc0e539c 2e46fdbb
K_8	b4e0e98a 3d028c05 b8d3a050 dbd67bef df675c7a 99eefbb0
K_9	e68584f6 ce31ef45 96c105ac 2a1be677 9d72b8b0 33cecc54
K_{10}	c22ffd76 1ab7167e 42bb3060 7da517f5 4aa0e8d3 0a070c3c
K_{11}	e200a765 c2be17b3 7f22543f 3e4eb7a1 c992a6f4 a783c823
K_{12}	c13cc747 ffcc8185 66514e9e e4ccc199 cd5c766d a004f676
K_{13}	1d3a1fa6 d46894ec f49c33e6 782fda7e 1fe6346c 0ffe981c
K_{14}	78b97c3d 956e8ee8 49ab721c 2672138a 037ea242 ce5fe8a4
K_{15}	225f7158 32d83e3e e118f6aa 1fb83751 4d27715c ed2fba4e
K_{16}	8dfbc56d e0a907db e4af091c 5e123225 d0e8d2e1 cc4501fb
K_{17}	8422a8f0 46a12f92 415152ad f55417f5 38736248 c6e29ded
K_{18}	5723715e abfa788c c3646af7 64af9186 8fc855ec 2bc36989
K_{19}	5e6b28e3 e0f5f592 eb3dd108 0551012a 50e4221d 97e85c0f
K_{20}	4e258e14 92298f0b 771269c3 6f934254 c0933b6b 421159b8
K_{21}	d76953f4 6a3e36be 53b656fb 610c22e0 9f399330 acf7e7e9
K_{22}	fe0b573b cbb73085 89ed67fc 77014cef e1b8431f ba1b4105
K_{23}	06de3450 b3f5b2fe df1cec27 fb22bd10 8e3de6fe 3d4acd27
K_{24}	c5444873 5bec968b 8b2af393 11e2f6ca 9cb3694f 94c56b91
K_{25}	939a1a93 27f101bb 5381bae7 49ebd1b1 f6d5fca7 0ca24bbc
K_{26}	7b03490b de00acfb c7f8abfe 410a14c1 d37932a9 14029327
K_{27}	bd948525 2c75004d c52486d5 0f07e2fa 1963e1fd 882719c3

Khóa K	3c2d1e0f 78695a4b b4a59687 f0e1d2c3 c3d2e1f0 8796a5b4
Bản rõ P	23222120 27262524 2b2a2928 2f2e2d2c
Bản mã C	325eb96f 871bad5a 35f5dc8c f2c67476
X_0	23222120 27262524 2b2a2928 2f2e2d2c
X_1	0f085091 030483d2 be4ab9ef 23222120
X_2	23fc7579 99fa0bb5 92d65065 0f085091
X_3	1e64758b 6b19e3e6 3edbb998 23fc7579
X_4	8da45638 d886dfdd 2da2b83c 1e64758b
X_5	a29dfd15 d368 7adf b89c47b4 8da45638
X_6	34e43c2e b6268ba7 584086d7 a29dfd15
X_7	df6e285b 83f26855 198fbb0c 34e43c2e
X_8	e62dde25 dd1cdf46 b8049544 df6e285b
X_9	d715e465 0e4d136e 35bc93a5 e62dde25
X_{10}	1ab97de4 a5c19c64 cfd627b0 d715e465
X_{11}	1a155930 4ccf6ee2 8c5136f7 1ab97de4
X_{12}	0eef4d0d 9f3065e1 405fc9b9 1a155930
X_{13}	a0dd5c61 fcefa1a4 48e2adc3 0eef4d0d
X_{14}	dcf21fcc f9ca084f 0b02cdd8 a0dd5c61
X_{15}	dfd53021 2eee92c5 eedfe48b dcf21fcc
X_{16}	81dce833 4e0af1c2 3abac76b dfd53021
X_{17}	9646ef75 607a7771 9fbc49ec 81dce833
X_{18}	7f40d072 ac609c27 5de1c810 9646ef75

Khóa K	3c2d1e0f 78695a4b b4a59687 f0e1d2c3 c3d2e1f0 8796a5b4
Bản rõ P	23222120 27262524 2b2a2928 2f2e2d2c
Bản mã C	325eb96f 871bad5a 35f5dc8c f2c67476
X_{19}	fd0bae5f 35429a83 11f5e49f 7f40d072
X_{20}	2feb9af2 0799218a e5374a5f fd0bae5f
X_{21}	fd86cfee a7d97a82 7c97ed23 2feb9af2
X_{22}	add0adf1 e09057e1 ccd95f65 fd86cfee
X_{23}	06c45cfe 392aaa1d ae9fd56c add0adf1
X_{24}	f3032315 b1df9d75 1627928d 06c45cfe
X_{25}	f4eec840 6a15d699 2392c666 f3032315
X_{26}	b353eb6a ad286c22 5a9d146d f4eec840
X_{27}	f2c67476 44333e44 6d5a1045 b353eb6a
X_{28}	325eb96f 871bad5a 35f5dc8c f2c67476

B.3.3 LEA-256

Khóa K	3c2d1e0f 78695a4b b4a59687 f0e1d2c3 c3d2e1f0 8796a5b4 4b5a6978 0f1e2d3c
K_0	003a0fd4 02497010 194f7db1 090d0883 2ff5805a c2580b27
K_1	a83e7ef9 053eca29 d359f988 8101a243 9bbf34b3 9228434f
K_2	2efee506 8b5f7bd4 9991e811 72dbc20c 2384c97f cefee47f
K_3	c571782c 00da90b1 b940a552 5db79619 4bc9a125 5d08a419
K_4	72de26cc d69bc26f 46a7f207 66ff4d81 a87862fc a5f63601
K_5	7909c4fa f3f93651 72cb0bcd ae69b2e3 80f2ca4b f13efcce
K_6	7669dbb9 6b7a5b8e fefbf6b1 ec608c8e 76e9d5d2 13ca4bf6
K_7	c5eeec7a aa42a59d 1f22cd00 fdd92bdc d6bbe3e8 15d459ec
K_8	cda7632a 9cf01bef 6596e261 8c1de14c 1127c3b8 48b3f629
K_9	3723d0e1 fc0317ec 3fdd5378 0201ae1d e55db65e e4c84dbc
K_{10}	3633db3f e4c24fc2 bb1e1fd7 a339425c fe3e1bdf d61c808d
K_{11}	bdca3449 beb8aa4e 145a9697 eb6fcd87 8b88ca72 7677a84b
K_{12}	d11005e9 558275c5 bc742819 3f17e888 20fcb71f 60886959
K_{13}	8d9446c4 67d2d167 855a6aef 69ea517c 36e48e11 0d3f4e86
K_{14}	bb0ede65 cceecc06 efc9c49f 44902261 bd8549c0 a7e7f682
K_{15}	772101e6 b4b9a250 6faa7b73 7318b792 1e57e751 fd43b41c
K_{16}	4ec21b5f dcfbf30b a4046947 be0e781c d74e21ac 6b1f5d22
K_{17}	e8b8e02b 4a662d2d b50f9ca9 01c98c69 9eb28089 216cfd3f
K_{18}	92f0126b 7b9961aa 581f94ac ab4be6dd c2a91af5 fb4e8e0c
K_{19}	4c2c8f04 81a45991 1fcb946c bccbb5b5 8088e9cb 8c1b2f89
K_{20}	192061be 78e5cf04 f239ab5c e8471e86 9e6217c7 e5fdf35c
K_{21}	83c3150d 766387f8 a1092ac7 6aa6f41d 16e200f9 6bdc26ca
K_{22}	52345706 db70d4af a8d8ffeb 492ee661 4cd1e991 d75d8352
K_{23}	85a9c5fb 1e0f569e 7ff7c600 3f36a1d8 e406ad00 4ded8f16
K_{24}	512bb2f4 772b192c 2e6168bd 76af67e1 d893a786 3e276f69
K_{25}	d11ee3ad b7f3c612 d3b19318 89fee4db b6c3aedd 05420f90
K_{26}	04f662f0 8fb41a6c 2f42dd5e a8ad1839 46474e45 46418de0
K_{27}	351550c6 6e8014f6 04924365 5f353d6f 4eba8d76 924a4318
K_{28}	5aba711c a36b1398 5b3e7bf4 7b3a2cf9 1d006ebe 0d5683e5

Khóa K	3c2d1e0f 78695a4b b4a59687 f0e1d2c3 c3d2e1f0 8796a5b4 4b5a6978 0f1e2d3c
K_{29}	4f56916f 215dccd2 9f57886f 876d1357 46013d49 2a4932a3
K_{30}	aa285691 ebefe7d3 e960e64b dd893f0f 6a234412 495d13c9
K_{31}	71c683e8 8069dfd0 6c1a501d 00699418 262142f0 a91a7393

Khóa K	3c2d1e0f 78695a4b b4a59687 f0e1d2c3 c3d2e1f0 8796a5b4 4b5a6978 0f1e2d3c
Bản rõ P	33323130 37363534 3b3a3938 3f3e3d3c
Bản mã C	f6af51d6 c189b147 ca00893a 97e1f927
X_0	33323130 37363534 3b3a3938 3f3e3d3c
X_1	0f0810d1 030583d2 a246bdef 33323130
X_2	e370475a 379d1cd0 7b627f7b 0f0810d1
X_3	a212c114 c5be3591 435bb556 e370475a
X_4	90bcb059 94df55a0 d8e15ef6 a212c114
X_5	4e5cc849 f484b5d8 ef0fc663 90bcb059
X_6	a520787d ae3db194 fa2feb17 4e5cc849
X_7	231a5d45 f338ad75 9d4b9850 a520787d
X_8	dd744e80 0a6569a0 3f9c93a9 231a5d45
X_9	d141f34e 311ba7ed b34c9f6f dd744e80
X_{10}	f5a76166 3e00a130 b1f9a58d d141f34e
X_{11}	af52973c c4befd35 aae4a642 f5a76166
X_{12}	3df5e119 b8937ebb b4a7a6ab af52973c
X_{13}	ede0ddb3 2c84bd26 2c86c203 3df5e119
X_{14}	960f7157 477a5b5a 29659f76 ede0ddb3
X_{15}	2c8d1d71 e0b54ae6 fbdd003c 960f7157
X_{16}	720a9b5f 18bf274a 0a1af597 2c8d1d71
X_{17}	1aa88202 c3867edc c49ce291 720a9b5f
X_{18}	e16c34f7 69defa5b 15b2990f 1aa88202
X_{19}	c7837b0b cf85d76f 17203201 e16c34f7
X_{20}	a3061bb3 bbe1ce55 00a3f8e9 c7837b0b
X_{21}	54f6bcfa c195ea5b b8280ef0 a3061bb3
X_{22}	662f351e 49995ddc 4ef48970 54f6bcfa
X_{23}	09db178e 4748e08a 30ba1411 662f351e
X_{24}	751113cb 9a425ee2 200fee63 09db178e
X_{25}	47d21a23 08561dff 86131859 751113cb
X_{26}	f7aaf6ac 4f5eac5b f4247a5b 47d21a23
X_{27}	8e952768 3de52e9b 367ed97c f7aaf6ac
X_{28}	cb641a2d 8d161a90 dbd4a137 8e952768
X_{29}	b6e87380 93b8b779 c9530e82 cb641a2d
X_{30}	48bd3559 5ad96ae7 2e0feb8b b6e87380
X_{31}	97e1f927 853a0309 437c41fc 48bd3559
X_{32}	f6af51d6 c189b147 ca00893a 97e1f927

Phụ lục C
(tham khảo)

Bảng đặc tính

Phụ lục này tổng hợp các thuộc tính hạng nhẹ của các mã khối mô tả trong tiêu chuẩn này. Trong phụ lục C.1 TCVN 12854-1 đưa ra các chỉ số phần cứng cho mã khối hạng nhẹ. Sử dụng các chỉ số, các thuộc tính hạng nhẹ của PRESENT và CLEFIA được tổng hợp trong bảng C.1.

Bảng C.1 – Bảng đặc tính

	Tên thuật toán								
	PRESENT [10][11][12]				CLEFIA [7][14]				
Kích thước khóa [bit]	80	128	80	128	128	128	128	192	265
Kích thước khối [bit]	64	64	64	64	128	128	128	128	128
Tiết diện chip[a] [GE]	1075	1391	1570	1884	2488	4950	5979	8536	8482
Chu kỳ[b] [CLK]	547	559	32	32	328	36	18	22	6
Số bit trên chu kỳ [bit/CLK]	0,12	0,11	2	2	0,39	3,56	7,11	5,82	4,92
Năng lượng[c] [GE]	1075	1391	1570	1884	2488	4950	5979	8536	8482
Năng lượng[d] [GE*CLK]	588025	777569	50240	60288	816064	178200	107622	187792	220532
Năng lượng trên mỗi bit[e] [GE*CLK/bit]	9188	12150	785	942	6367	1392	841	1467	1732
Công nghệ [μm]	0,18		0,18		0,13	0,09	0,09	0,09	0,09
Hỗ trợ giải mã	KHÔNG		KHÔNG		KHÔNG	CÓ	CÓ	CÓ	CÓ
Đặc tính	Diện tích nhỏ nhất		Diện tích nhỏ và năng lượng thấp (dựa vào vòng)		Diện tích nhỏ nhất	Diện tích nhỏ (dựa vào vòng)	Năng lượng thấp và hiệu quả cao (dựa vào vòng)		

[a] Tiết diện chip được đo bởi các cổng tương đương (GE). Các con số được đưa ra đã thu được bằng cách sử dụng một công cụ tổng hợp tự động thiết kế điện tử tự động

[b] Số chu kỳ xung nhịp [CLK] để một lần thực thi thuật toán tạo ra đầu ra. Ta thu được bởi kiến trúc cài đặt phần cứng.

[c] Với giả thiết các ứng dụng phần cứng hạng nhẹ có xung nhịp ở tần suất thấp vài trăm kHz, năng lượng tiêu thụ có thể được ước lượng bằng cách sử dụng phép đo như với diệc tích chip [6].

[d] Năng lượng tiêu thụ ký hiệu năng lượng tiêu thụ trên một khoảng thời gian cụ thể. Ta có thể được ước tính bằng việc nhân năng lượng tiêu thụ với số chu kỳ đếm được.

[e] Ta có thể thu được bằng cách chia năng lượng cho kích thước khối.

Bảng C.2 đến C.4 tóm tắt hiệu năng dữ liệu đối với LEA và chỉ ra hiệu năng của LEA trên vi điều khiển Mtmega128 8 bit, vi điều khiển MSP430 16 bit, và vi điều khiển Cortex-M3 32 bit. Tốc độ được đo trên chu kỳ mỗi byte, vì vậy các số nhỏ hơn tương ứng với tốc độ cao hơn. Chú ý rằng khối/độ dài khóa theo bit của LEA-128, LEA-192, và LEA-256 là 128/128, 128/192, và 128/256 bit tương ứng.

Bảng C.2: Triển khai tốc độ cao của LEA trên Atmega128

Tên thuật toán	LEA-128	LEA-192	LEA-256
Kích cỡ mã [byte]	862	934	934
Kích cỡ RAM [byte]	433	761	865
Chu kỳ mã hóa [CLK]	2689	3589	4081
Tốc độ [CLK/byte]	168,06	224,31	255,06

Bảng C.3: Triển khai tốc độ cao của LEA trên MSP430

Tên thuật toán	LEA-128	LEA-192	LEA-256
Kích cỡ mã [byte]	650	666	664
Kích cỡ RAM [byte]	440	768	872
Chu kỳ mã hóa [CLK]	2056	2435	2765
Tốc độ [CLK/byte]	128,50	152,19	172,81

Bảng C.4: Triển khai tốc độ cao của LEA trên Cortex-M3

Tên thuật toán	LEA-128	LEA-192	LEA-256
Kích cỡ mã [byte]	808	930	1058
Kích cỡ RAM [byte]	472	776	880
Chu kỳ mã hóa [CLK]	424	565	644
Tốc độ [CLK/byte]	26,50	35,31	40,25

Phụ lục D
(tham khảo)

Giới hạn của mã khối với một khóa đơn

Phụ lục này đưa ra cận trên về các phép mã hóa mã khối nên thực hiện với một khóa đơn. Độ an toàn của hầu hết các chế động mã khối giảm xuống khi số lần mã hóa tiến đến $2^{n/2}$ [8][9][12], với n là kích thước khối theo bit. Thông tin đủ để có thể phân biệt được đầu ra của mã khối với một hàm ngẫu nhiên (sau $2^{n/2}$ lần mã hóa. Ví dụ, với n = 64, mã hóa $2^{64/2} = 2^{32}$ khối, là đủ để phơi bày mã khối trước các tấn công có thể khi sử dụng trong một vài chế độ hoạt động của mã khối (như CBC hoặc chế độ bộ đếm). Vì vậy, với các mã khối n-bit, việc mã hóa lớn hơn $2^{n/2}$ khối sử dụng một khóa đơn với các chế độ hoạt động cụ thể phải được ngăn chặn.

Cảnh báo này không chỉ áp dụng cho các mã khối hạng nhẹ được quy đinh trong tiêu chuẩn này mà còn cho tất cả các loại mã khối bao gồm cả các mã khối được quy định trong TCVN 11367-3 (ISO/IEC 18033-3). để biết thêm thông tin, xem tài liệu chuẩn 12 (SD 12) của ISO/IEC JTC 1/SC 27.

Thư mục tài liệu tham khảo

[1] TCVN 11495-1 (ISO/IEC 9797-1), Công nghệ thông tin – Các kỹ thuật an toàn – Mã xác thực thông điệp (MAC) – Phần 1: Cơ chế sử dụng mã khối

[2] TCVN 11817-1, Công nghệ thông tin – Các kỹ thuật an toàn - Xác thực thực thể - Phần 1: Tổng quan

[3] TCVN 11816 (ISO/IEC 10116), Công nghệ thông tin – Các kỹ thuật an toàn – Chế độ hoạt động cho mã khối n-bit.

[4] TCVN 7817-1 (ISO/IEC 11770)-1, Công nghệ thông tin – Các kỹ thuật an toàn – Quản lý khóa – Phần 1: Khung tổng quát

[5] TCVN 11367-1 (ISO/IEC 18033-1Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 1: Tổng quan

[6] TCVN 12854-1 (ISO/IEC 29192-1), Công nghệ thông tin – Các kỹ thuật an toàn – Mật mã hạng nhẹ - Phần 1: Tổng quan

[7] Akishita T., Hiwatari H., Compact Hardware Implementations of the 128-bit Blockcipher CLEFIA. Available at <http:// www .sony .net/ clefia>, 2011

[8] Bellare M., Desai A., Jokipii E., Rogaway P., Á Concrete Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation. In Proceedings of the 38th Symposium on Foundations of Computer Science, pp. 394-405, IEEE, 1997

[9] Bellare M., Kilian J., Rogaway P., The Security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences* (JCSS), volume **61**, no 3, pp. 362-399, 2000

[10] Bogdanov A., Knudsen L., Leander G., Paar C., Poschmann A., Robshaw M. et al. , PRESENT — An Ultra-Lightweight Block Cipher. In Proceedings of Workshop on Cryptographic Hardware and Embedded Systems — CHES 2007, Lecture Notes in Computer Science volume 4727, pp. 450-466, Springer-Verlag, 2007

[11] Hong D., Lee J.K., Kim D.C., Kwon D., Ryu G.H., Lee D., LEA: A 128-bit Block Cipher for Fast Encryption on Common Processors. In Proceedings of World Conference on Information Security Applications — WISA 2013, Lecture Notes in Computer Science volume 8267, pp. 3-27, Springer-Verlag, 2014

[12] Poschmann A., Lightweight Cryptography — Cryptographic Engineering for a Pervasive World. Europäischer Universitätsverlag, Ph.D. thesis, 2009

[13] Rogaway P., Efficient Instantiations of Tweakable Block Ciphers and Refinements of Modes OCB and PMAC. Available at <http:// seclab .cs .ucdavis .edu/ papers/ offsets .pdf>

[14] Rolfes C., Poschmann A., Leander G., Paar C., Ultra-Lightweight Implementations for Smart Devices – Security for 1000 Gate Equivalents. 8th Smart Card Research and Advanced Application Conference, CARDIS 2008, Lecture Notes in Computer Science volume 5189, pp. 89-103, Springer-Verlag, 2008

[15] Shirai T., Shibutani K., Akishita T., Moriai S., Iwata T., The 128-bit Blockcipher CLEFIA. In Proceedings of Workshop on Fast Software Encryption 2007 — FSE 2007, Lecture Notes in Computer Science volume 4593, pp. 181-195, Springer-Verlag, 2007

[16] Standing Document 12 (SD12) of ISO/IEC JTC 1/SC 27. Available at http:// www .jtc1sc27 .din .de/ sbe/ SD12