BỘ QUỐC PHÒNG	CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
	Độc lập - Tự do - Hạnh phúc
Số: 161/2016/TT-BQP	*Hà Nội, ngày 21 tháng 10 năm 2016*

THÔNG TƯ
Ban hành Quy chuẩn kỹ thuật quốc gia về mật mã dân sự sử dụng trong lĩnh vực ngân hàng

Căn cứ Luật Tiêu chuẩn và Quy chuẩn kỹ thuật ngày 29 tháng 6 năm 2006;

Căn cứ Luật An toàn thông tin mạng ngày 19 tháng 11 năm 2015;

Căn cứ Luật Chất lượng sản phẩm, hàng hóa ngày 21 tháng 11 năm 2007;

Căn cứ Nghị định số 127/2007/NĐ-CP ngày 01 tháng 8 năm 2007 của Chính phủ quy định chi tiết thi hành một số điều của Luật Tiêu chuẩn và Quy chuẩn kỹ thuật;

Căn cứ Nghị định số 09/2014/NĐ-CP ngày 27 tháng 01 năm 2014 của Chính phủ quy định chức năng, nhiệm vụ, quyền hạn và cơ cấu tổ chức của Ban Cơ yếu Chính phủ;

Căn cứ Nghị định số 35/2013/NĐ-CP ngày 22 tháng 4 năm 2013 của Chính phủ quy định chức năng, nhiệm vụ, quyền hạn và cơ cấu tổ chức của Bộ Quốc phòng;

Theo đề nghị của Trưởng ban Ban Cơ yếu Chính phủ;

Bộ trưởng Bộ Quốc phòng ban hành Thông tư ban hành Quy chuẩn kỹ thuật quốc gia về mật mã dân sự sử dụng trong lĩnh vực ngân hàng.

Điều 1. Ban hành kèm theo Thông tư này Quy chuẩn kỹ thuật quốc gia về mật mã dân sự sử dụng trong lĩnh vực ngân hàng:

1. QCVN 4 : 2016/BQP, Mã hóa dữ liệu sử dụng trong lĩnh vực ngân hàng;

2. QCVN 5 : 2016/BQP, Chữ ký số sử dụng trong lĩnh vực ngân hàng;

3. QCVN 6 : 2016/BQP, Quản lý khóa sử dụng trong lĩnh vực ngân hàng.

Điều 2. Thông tư này có hiệu lực thi hành kể từ ngày 09 tháng 12 năm 2016.

Điều 3. Trưởng ban Ban Cơ yếu Chính phủ, Thủ trưởng các cơ quan, đơn vị, tổ chức và cá nhân có liên quan chịu trách nhiệm thi hành Thông tư này./.

Nơi nhận:
- Chính phủ (để báo cáo);
- Thủ tướng Chính phủ (để báo cáo);
- Các Bộ, cơ quan ngang Bộ, cơ quan thuộc Chính phủ;
- UBND các tỉnh, thành phố trực thuộc Trung ương;
- Thủ trưởng BQP, CN TCCT;
- Ban Cơ yếu Chính phủ;
- Cục Tiêu chuẩn - Đo lường - Chất lượng/BQP;
- Cục Kiểm tra văn bản QPPL Bộ Tư pháp;
- Công báo, Cổng TTĐTCP;
- Cổng TTĐTBQP;
- Vụ Pháp chế/BQP;
- Lưu: VT, BCY; HT110.

BỘ TRƯỞNG

Đại tướng Ngô Xuân Lịch

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

QCVN 4 : 2016/BQP

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ MÃ HÓA DỮ LIỆU SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on data encryption used in banking

HÀ NỘI - 2016

MỤC LỤC

Lời nói đầu

QCVN 4 : 2016/BQP do Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã – Ban Cơ yếu Chính phủ biên soạn, Ban Cơ yếu Chính phủ trình duyệt, Bộ Khoa học và Công nghệ thẩm định và được ban hành theo Thông tư số 161/2016/TT-BQP ngày 21 tháng 10 năm 2016 của Bộ trưởng Bộ Quốc phòng.

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ MÃ HÓA DỮ LIỆU SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on data encryption used in banking

1. QUY ĐỊNH CHUNG

1.1. Phạm vi điều chỉnh

Quy chuẩn kỹ thuật quốc gia này quy định mức giới hạn về đặc tính kỹ thuật mật mã của các thuật toán mã hóa dữ liệu dùng trong các sản phẩm mật mã dân sự sử dụng trong lĩnh vực ngân hàng.

1.2. Đối tượng áp dụng

Quy chuẩn này áp dụng đối với các doanh nghiệp kinh doanh sản phẩm, dịch vụ mật mã dân sự trong lĩnh vực ngân hàng; các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự.

1.3. Tài liệu viện dẫn

- *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST SP 800-90A Rev. 1, National Institute of Standards and Technology, June 2015. (*Khuyến cáo cho bộ sinh số ngẫu nhiên sử dụng bộ sinh bit ngẫu nhiên tất định*, NIST SP 800-90A Rev. 1, Viện tiêu chuẩn và công nghệ quốc gia (Mỹ), tháng 6 năm 2015).
- TCVN 7876:2007 Công nghệ thông tin – Kỹ thuật mật mã – Thuật toán mã dữ liệu AES.
- TCVN 11367-3:2016 (ISO/IEC 18033-3:2010) Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 3: Mã khối.
- ISO/IEC 10116:2006 Information technology -- Security techniques -- Modes of operation for an n-bit block cipher.
- ISO/IEC 9797-1:2011 Information technology -- Security techniques -- Message Authentication Codes (MACs) -- Part 1: Mechanisms using a block cipher.

1.4. Giải thích từ ngữ

Trong Quy chuẩn này, các từ ngữ dưới đây được hiểu như sau:

1.4.1.

Thông tin không thuộc phạm vi bí mật nhà nước

Là thông tin không thuộc nội dung tin "tuyệt mật", "tối mật" và "mật" được quy định tại Pháp lệnh Bảo vệ bí mật nhà nước ngày 28 tháng 12 năm 2000.

1.4.2.

Mật mã

Là những quy tắc, quy ước riêng dùng để thay đổi hình thức biểu hiện thông tin nhằm bảo đảm bí mật, xác thực, toàn vẹn của nội dung thông tin.

1.4.3.

Mật mã dân sự

Là kỹ thuật mật mã và sản phẩm mật mã được sử dụng để bảo mật hoặc xác thực đối với thông tin không thuộc phạm vi bí mật nhà nước.

1.4.4.

Sản phẩm mật mã dân sự

Là các tài liệu, trang thiết bị kỹ thuật và nghiệp vụ mật mã để bảo vệ thông tin không thuộc phạm vi bí mật nhà nước.

1.4.5.

Kỹ thuật mật mã

Là phương pháp, phương tiện có ứng dụng mật mã để bảo vệ thông tin.

1.4.6.

Mã hóa

Phép biến đổi (khả nghịch) dữ liệu bởi thuật toán mật mã để tạo ra bản mã, tức là giấu nội dung thông tin của dữ liệu.

1.4.7.

Giải mã

Phép toán ngược với phép mã hóa tương ứng.

1.4.8.

Mã khối

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

1.4.9.

Mã dòng

Hệ mật đối xứng với tính chất là thuật toán mã hóa bao gồm tổ hợp một dãy các ký tự của bản rõ với dãy các ký tự của khóa dòng, mỗi lần một ký tự, sử dụng một hàm khả nghịch.

1.4.10.

Khóa

Dãy các ký tự điều khiển sự vận hành của các thuật toán mật mã (ví dụ, phép mã hóa, giải mã).

1.4.11.

Khóa dòng

Dãy các ký tự giả ngẫu nhiên bí mật, được sử dụng bởi các thuật toán mã hóa và giải mã của mã dòng.

1.5. Các ký hiệu

n	độ dài tính bằng bit của bản rõ/bản mã đối với mã khối
E_k	hàm mã hóa với khóa K
D_k	hàm giải mã với khóa K
Nr	số vòng của thuật toán AES, bằng 10,12 hoặc 14 để chọn độ dài khóa tương ứng 128, 192, hoặc 256 bit .
Nb	Số các cột (các từ 32 bit) tạo nên Trạng thái.
$\oplus$	phép toán logic XOR trên xâu bit, nghĩa là nếu A và B là hai xâu cùng độ dài thì $A \oplus B$ là xâu bit bao gồm các bit là kết quả phép toán logic XOR của A và B
$\otimes$	phép nhân hai đa thức (mỗi đa thức có bậc bé hơn 4) theo mod $x^4 + 1$
$\wedge$	phép toán logic AND trên xâu bit, nghĩa là nếu A và B là các xâu bit cùng độ dài, thì $A \wedge B$ là xâu bit được tạo từ phép toán logic AND các bit tương ứng của A và B.
$\vee$	phép toán logic OR trên xâu bit, tức nếu A và B là hai xâu bit cùng độ dài, thì $A \vee B$ là xâu bit gồm các bit là kết quả của phép toán logic OR của A và B.
$\|\|$	phép ghép các xâu bit
$\bullet$	phép nhân trên trường hữu hạn
$\ggg_i$	phép dịch vòng sang trái i bit.
$\lll_i$	phép dịch vòng sang phải i bit
$\bar{x}$	phép bù bit của x
$a \bmod n$	với các số nguyên a và n, $a \bmod n$ ký hiệu số dư (không âm) trong phép chia a cho n. Một cách tương đương, $b = a \bmod n$ nếu b là số nguyên duy nhất thõa mãn các điều kiện sau: (i) $0 \le b < n$ (ii) $(b - a)$ là bội số nguyên của n
$\boxplus$	phép cộng trong số học mô-đun, nghĩa là nếu A và B là hai xâu t-bit thì $A \boxplus B$ được xác định bằng $(A + B) mod\ 2^t$
$\boxminus$	phép trừ trong số học mô-đun, nghĩa là nếu A và B là hai xâu t-bit thì $A \boxminus B$ được xác định bằng $(A - B) mod\ 2^t$

2. QUY ĐỊNH KỸ THUẬT

2.1. Quy định chung

- Quy định chi tiết về nguồn ngẫu nhiên:

Các số ngẫu nhiên được sử dụng cho các mục đích khác nhau như để sinh các tham số mật mã, các khóa mật mã, các giá trị ngẫu nhiên dùng một lần và các giá trị thách đố xác thực.

Một số bộ sinh bit ngẫu nhiên tất định DRBG được chấp thuận để sử dụng theo quy định chung bao gồm: HASH_DRBG, HMAC_DRBG và CTR_DRBG.

Các bộ sinh bit ngẫu nhiên RBG tuân theo SP800-90A phiên bản sửa đổi lại năm 2015 để sinh bit ngẫu nhiên cũng được chấp thuận để sử dụng tiếp.

2.2. Mã khối

Trong Quy chuẩn này quy định 3 loại mã khối áp dụng cho việc mã hóa dữ liệu để bảo vệ thông tin trong lĩnh vực ngân hàng.

Độ dài khối	Tên thuật toán	Độ dài khóa quy định áp dụng
64 bit	TDEA	Không nhỏ hơn 192 bit
128 bit	AES	Không nhỏ hơn 256 bit
	Camellia	Không nhỏ hơn 256 bit

- Quy định về ngưỡng thời gian và độ an toàn khóa cụ thể:

Độ an toàn theo bit	Thuật toán mã khối	Thời gian sử dụng quy định
112	3TDEA	Đến cuối năm 2030
256	AES-256	Từ năm 2030
	Camellia-256	

2.2.1. TDEA - Thuật toán mã dữ liệu bội ba

Trong Quy chuẩn này thuật toán mã dữ liệu bội ba (TDEA-Triple Data Encryption Algorithm) xử lý các khối dữ liệu 64 bit, sử dụng khóa mật mã có độ dài không nhỏ hơn 192 bit.

Phép mã hóa/giải mã TDEA là phép toán ghép các phép toán mã hoá và giải mã DES. Khóa của TDEA gồm ba khóa K_1, K_2 và K_3 là những khóa DES khác nhau ($K_1 \neq K_2, K_2 \neq K_3$ và $K_3 \neq K_1$).

2.2.1.1. Phép mã hóa/giải mã TDEA

2.2.1.1.1. Các định nghĩa mã hóa /giải mã

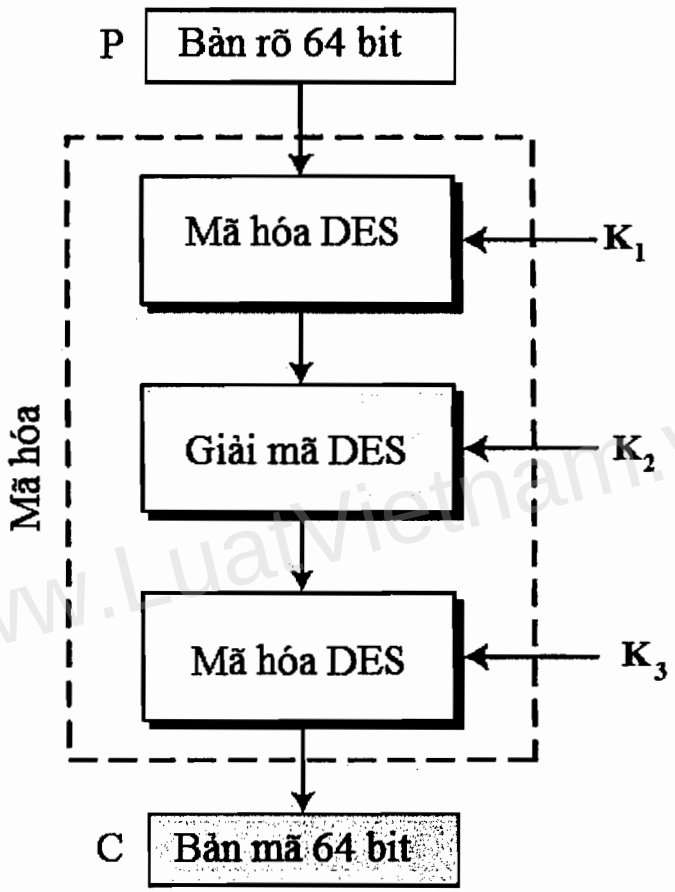
TDEA được xác định theo thuật ngữ của phép toán của DES, ở đấy E_k là phép toán mã hóa của DES với khóa K và D_k là phép giải mã của DES với khóa K.

2.2.1.1.2. Phép mã hóa

Phép biến đổi khối P có độ dài 64 bit thành khối C có độ dài 64 bit được xác định như sau:

$$C = E_{K_3}(D_{K_2}(E_{K_1}(P))).$$

Phép mã hóa/giải mã DES được mô tả tại Phụ lục A của Quy chuẩn này.



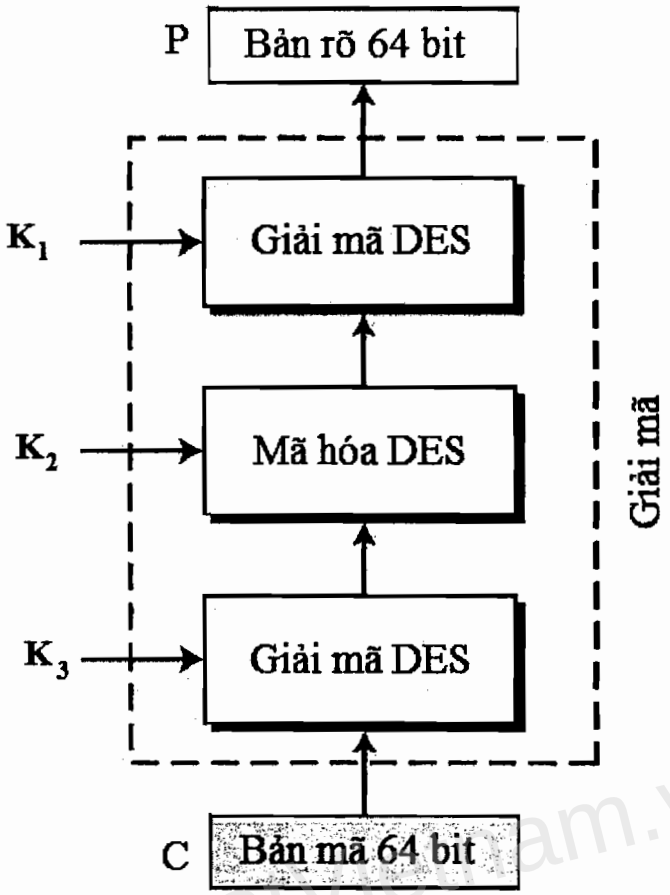
Hình 1: Sơ đồ phép mã hóa TDEA

2.2.1.1.3. Phép giải mã TDEA

Phép biến đổi khối C có độ dài 64 bit thành khối P có độ dài 64 bit được xác định như sau:

$$P = D_{K_1}(E_{K_2}(D_{K_3}(C))).$$

Phép mã hóa/giải mã DES được mô tả tại Phụ lục A của Quy chuẩn này.



Hình 2: Sơ đồ phép giải mã TDEA

2.2.1.2. Chu trình khóa

Phần lược đồ khóa DES được mô tả tại Điều A.5.

2.2.1.3. Chế độ hoạt động của TDEA

Các chế độ hoạt động của TDEA được quy định tại Mục 2.3 của Quy chuẩn này.

2.2.1.4. Khóa

Độ dài khóa sử dụng trong mã khối TDEA được quy định:

Độ dài khối	Tên thuật toán	Độ dài khóa quy định áp dụng
64 bit	TDEA	Không nhỏ hơn 192 bit

2.2.1.4.1. Các yêu cầu về khóa

Đối với các chế độ hoạt động của mã khối TDEA, 3 khóa mật mã (K_1, K_2 và K_3) xác định khóa cho hệ mật TDEA cần:

a. Phải được giữ bí mật.

b. Được sinh ra sử dụng bộ sinh bit ngẫu nhiên được quy định trong phần 2.1 của Quy chuẩn này.

c. Là các khóa mật mã khác nhau.

d. Đảm bảo toàn vẹn mà mỗi khóa không bị thay đổi trái phép kể từ khi được sinh ra, gửi đi hoặc được lưu trữ bởi một thực thể có thẩm quyền.

e. Được sử dụng theo thứ tự thích hợp theo quy định bởi chế độ hoạt động riêng biệt.

f. Không sử dụng để mã hóa nhiều hơn 2^{32} khối dữ liệu 64-bit.

g. Không sử dụng các khóa yếu và bán yếu *(Được quy định tại Điều 2.2.1.4.2)*.

2.2.1.4.2. Khóa yếu

Đối với thuật toán DES có một số khóa mật mã được coi là yếu khi sử dụng. Việc sử dụng các khóa yếu này ảnh hưởng đến độ an toàn của thuật toán TDEA. Các khóa yếu không được sử dụng trong thuật toán TDEA được liệt kê dưới đây (theo định dạng thập lục phân):

```
01010101 01010101
FEFEFEFE FEFEFEFE
E0E0E0E0 F1F1F1F1
1F1F1F1F 0E0E0E0E
```

Một số cặp khóa khi mã hóa thì bản rõ và bản mã là giống hệt nhau và cũng không được sử dụng. Các khóa bán-yếu đó là (theo định dạng thập lục phân):

```
011F011F010E010E và 1F011F010E010E01
01E001E001F101F1 và E001E001F101F101
01FE01FE01FE01FE và FE01FE01FE01FE01
1FE01FE00EF10EF1 và E01FE01FF10EF10E
1FFE1FFE0EFE0EFE và FE1FFE1FFE0EFE0E
E0FEE0FEF1FEF1FE và FEE0FEE0FEF1FEF1
```

Ngoài ra còn 48 khóa sau chỉ tạo ra 4 khóa con khác nhau (thay vì 16 khóa con khác nhau) và cũng không được sử dụng. Đó là các khóa sau (theo định dạng thập lục phân):

```
01011F1F01010E0E  1F1F01010E0E0101      E0E01F1FF1F10E0E
0101E0E00101F1F1  1F1FE0E00E0EF1F1      E0E0FEFEF1F1FEFE
0101FEFE0101FEFE  1F1FFEFE0E0EFEFE      E0FE011FF1FE010E
011F1F01010E0E01  1FE001FE0EF101FE      E0FE1F01F1FE0E01
011FE0FE010EF1FE  1FE0E01F0EF1F10E      E0FEFEE0F1FEFEF1
011FFEE0010EFEF1  1FE0FE010EF1FE01      FE0101FEFE0101FE
01E01FFE01F10EFE  1FFE01E00EFE01F1      FE011FE0FE010EF1
FE01E01FFE01F10E  1FFEE0010EFEF101      FE1F01E0FE0E01F1
01E0E00101F1F101  1FFEFE1F0EFEFE0E      FE1FE001FE0EF101
01E0FE1F01F1FE0E  E00101E0F10101F1      FE1F1FFEFE0E0EFE
```

01FE1FE001FE0EF1	E0011FFEF1010EFE	FEE0011FFEF1010E
01FEE01F01FEF10E	E001FE1FF101FE0E	FEE01F01FEF10E01
01FEFE0101FEFE01	E01F01FEF10E01FE	FEE0E0FEFEF1F1FE
1F01011F0E01010E	E01F1FE0F10E0EF1	FEFE0101FEFE0101
1F01E0FE0E01F1FE	E01FFE01F10EFE01	FEFE1F1FFEFE0E0E
1F01FEE00E01FEF1	E0E00101F1F10101	FEFEE0E0FEFEF1F1

CHÚ THÍCH: Các khóa yếu và khóa bán-yếu được liệt kê ở trên được biểu diễn với tính lẻ và được chỉ ra trong phần ngoài cùng bên phải của mỗi byte.

2.2.2. AES

Đối với thuật toán AES, độ dài của khối đầu vào, khối đầu ra và Trạng thái đều là 128 bit. Như vậy $Nb = 4$ là số lượng các từ 32 bit (số cột) của Trạng thái.

Trong Quy chuẩn này thuật toán AES, độ dài Khóa mã K quy định áp dụng là 256 bit. Độ dài khóa được biểu diễn bằng một số $Nk = 8$ thể hiện số lượng các từ 32 bit (số cột) của Khóa mã. Số vòng trong quá trình thực thi thuật toán AES (được ký hiệu là Nr) với khóa độ dài 256 bit là $Nr = 14$.

Quy chuẩn này quy định cụ thể các giá trị được phép dùng cho độ dài khóa (Nk), kích cỡ khối (Nb) và số lượng vòng lặp (Nr) như Bảng 1.

	Độ dài khóa *(Nk từ)*	Độ dài khối *(Nb từ)*	Số vòng *(Nr)*
AES-256	8	4	14

Bảng 1: Các tổ hợp Khóa - Khối - Vòng

2.2.2.1. Phép mã hóa/giải mã AES

Đối với cả hai Phép mã hóa và Phép giải mã, thuật toán AES sử dụng hàm vòng bao gồm bốn phép biến đổi khác nhau trên byte:

1) Phép thay thế byte sử dụng bảng thay thế (S-box).

2) Dịch hàng của mảng Trạng thái bằng các offset khác nhau.

3) Trộn dữ liệu trong mỗi cột của mảng Trạng thái.

4) Bổ sung khóa vòng vào Trạng thái.

2.2.2.1.1. Phép mã hóa AES

Thuật toán AES gồm một dãy các phép toán được thực hiện trên mảng hai chiều của các byte, được gọi là Trạng thái. Trạng thái gồm bốn dòng byte, mỗi dòng chứa 4 byte. Trong mảng Trạng thái được ký hiệu bằng s, mỗi byte riêng biệt có hai chỉ số với số dòng r với $0 \leq r < 4$ và số cột c, với $0 \leq c < 4$. Trạng thái được ký hiệu là $S = (s_{r,c})$.

Khi bắt đầu quá trình mã hóa, 16 byte của Trạng thái được khởi tạo với ρ_i byte bản rõ, tính từ trên xuống dưới và từ trái sang phải và được minh họa trên Hình 3.

ρ_0	ρ_4	ρ_8	ρ_{12}
ρ_1	ρ_5	ρ_9	ρ_{13}
ρ_2	ρ_6	ρ_{10}	ρ_{14}
ρ_3	ρ_7	ρ_{11}	ρ_{15}

→

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

Hình 3: Khởi đầu Trạng thái

Sau khi cộng khóa vòng ban đầu, Trạng thái được biến đổi bằng cách thực thi hàm vòng Nr lần, với vòng cuối khác một ít với $Nr - 1$ vòng đầu. Nội dung cuối cùng của Trạng thái chính là bản mã đầu ra .

Phép mã hóa đầy đủ có thể được mô tả như sau:

(1) $S = AddRoundKey(P, W_0)$

(2) $For\ i = 1\ to\ Nr - 1$

$S = SubBytes\ (S)$

$S = ShiftRows(S)$

$S = MixColumns(S)$

$S = AddRoundKey(S, W_i)$

(3) $S = SubBytes\ (S), S = ShiftRows(S)$

(4) $C = AddRoundKey\ (S, W_{Nr})$

Các phép biến đổi riêng biệt $SubBytes()$, $ShiftRows()$, $MixColumns()$, $AddRoundkey()$ xử lí Trạng thái và được mô tả tại Phụ lục B. Tất cả Nr vòng đều giống nhau, trừ vòng cuối cùng không chứa phép biến đổi $MixColumns()$. Trong phép toán ở trên, mảng W_i chứa các khóa vòng được mô tả tại Điều 2.2.2.2.

2.2.2.1.2. Phép giải mã AES

Tất cả các phép biến đổi sử dụng trong các phép mã hóa đều khả nghịch. Khi thực thi phép giải mã, dãy các phép biến đổi được sử dụng trong phép mã hóa vẫn được duy trì, nhưng thay bằng các phép biến đổi ngược như sau.

Phép giải mã đầy đủ có thể mô tả như sau:

(1) $S = AddRoundKey\ (C, W_{Nr})$

(2) $for\ i = Nr - 1\ down\ to\ 1$

$S = ShiftRows^{-1}(S)$

$S = SubBytes^{-1}(S)$

$S = AddRoundKey(S, W_i)$

$S = MixColumns^{-1}(S)$

(3) $S = ShiftRows^{-1}(S)$

$S = SubBytes^{-1}(S)$

(4) $P = AddRoundKey(S, W_0)$

Các phép biến đổi $SubBytes^{-1}(), ShiftRows^{-1}(), MixColumns^{-1}()$ thực hiện xử lí Trạng thái. Tất cả Nr vòng đều giống nhau, trừ vòng cuối không chứa phép biến đổi $MixColumns^{-1}()$.

Các phép biến đổi $SubBytes^{-1}(), ShiftRows^{-1}(), MixColumns^{-1}()$ thực hiện xử lí Trạng thái và được mô tả tại Phụ lục B. Tất cả Nr vòng đều giống nhau, trừ vòng cuối không chứa phép biến đổi $MixColumns^{-1}()$. Việc tính các khóa vòng W_i được mô tả tại Điều 2.2.2.2.

2.2.2.1.3. Quy định về S-Box và S-Box nghịch đảo trong AES

a) S-Box

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

Bảng 2: S-Box: các giá trị thay thế cho byte {xy} (theo dạng thập lục phân)

b) S-Box nghịch đảo

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

Bảng 3: S-Box nghịch: thay thế các giá trị theo byte {xy} (dạng thập lục phân)

2.2.2.2. Chu trình khóa

Thuật toán AES nhận khóa mật mã K và thực hiện thủ tục mở rộng khóa để tạo ra lược đồ khóa. Việc mở rộng khóa tạo ra tổng cộng $4(Nr + 1)$ từ: Thuật toán đòi hỏi tập khởi đầu gồm 4 từ, và mỗi vòng Nr vòng đòi hỏi 4 từ dữ liệu khóa. Lược đồ khóa nhận được là một mảng tuyến tính gồm các từ 4-byte, ký hiệu là w_j, với j nằm trong khoảng $0 \leq j < 4(Nr + 1)$.

Phép mở rộng khóa đầy đủ cho AES-256 có thể được mô tả như sau:

(1) $[w_0, w_1, w_2, w_3] = K_0, [w_4, w_5, w_6, w_7] = K_1, \boldsymbol{Nk} = 8$

(2) for j = $\boldsymbol{Nk}$ to $4(\boldsymbol{Nr} + 1) - 1$:

if ($j \bmod \boldsymbol{Nk} = 0$) then

$$w_j = w_{j-Nk} \oplus SubBytes^{*}\left(\text{ShiftColumn}(w_{j-1})\right) \oplus R^{i/Nk}c$$

else if ($j \bmod \boldsymbol{Nk} = 4$) then

$$w_j = w_{j-Nk} \oplus SubBytes^{*}(w_{j-1})$$

else

$$w_j = w_{j-Nk} \oplus w_{j-1}$$

(3) $W_i = [w_{(4*i)}, w_{(4*i+1)}, w_{(4*i+2)}, w_{(4*i+3)}]$ $for\ 0 \leq i \leq \boldsymbol{Nr}$.

Trong phép toán trên K_0 và K_1 biểu thị hai nửa của khóa mật mã K 256-bit.

2.2.2.3. Chế độ hoạt động của AES

Các chế độ hoạt động của AES được quy định tại Mục 2.3 của Quy chuẩn này.

2.2.2.4. Khóa

Độ dài khóa sử dụng trong mã khối AES được quy định

Độ dài khối	Tên thuật toán	Độ dài khóa quy định áp dụng
128 bit	AES	Không nhỏ hơn 256 bit

2.2.3. Camellia

Trong Quy chuẩn này thuật toán mã đối xứng Camellia xử lí các khối 128 bit, sử dụng khóa mật mã độ dài không nhỏ hơn 256 bit.

2.2.3.1. Phép mã hóa/giải mã Camellia

2.2.3.1.1. Phép mã hóa Camellia

Quá trình mã hóa với khóa 256-bit làm việc trên 24 vòng, được chỉ ra trên Hình 4. Phép biến đổi khối 128 bit P vào khối 128-bit C được định nghĩa như sau (L và R là các biến độ dài 64 bit, kw, k và kl là các khóa vòng 64 bit):

(1) $L_0 \parallel R_0 = P \oplus (kw_1 \parallel kw_2)$

(2) for $i = 1$ to 24:

$$L_i = \mathrm{F}(L_{i-1}, k_i) \oplus R_{i-1}$$

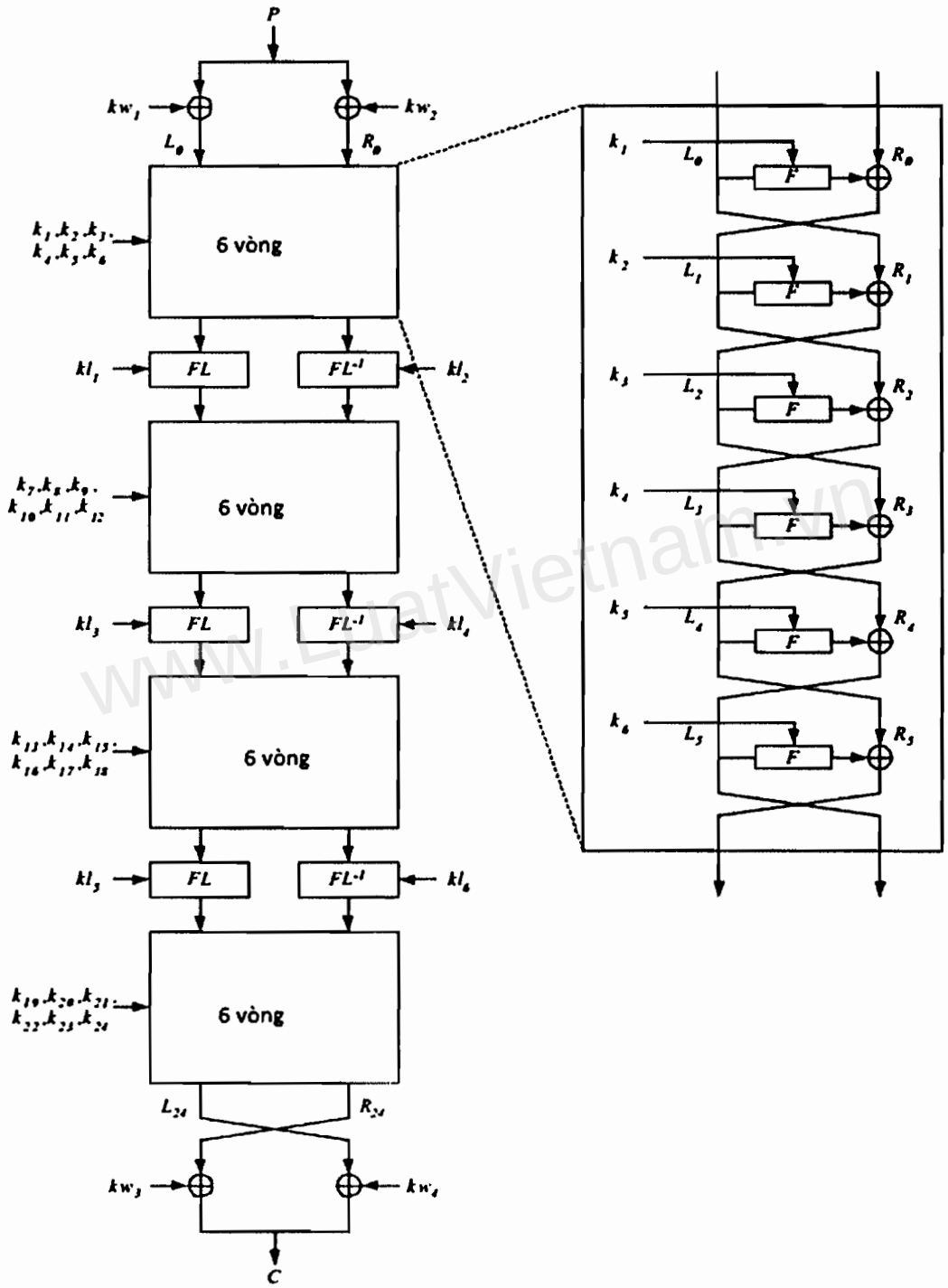
$$R_i = L_{i-1}$$

if ($i = 6$ or 12 or 18) then

$$L_i = FL(L_i, kl_{i/3-1})$$

$$R_i = FL^{-1}(R_i, kl_{i/3})$$

(3) $C = (R_{24} \oplus kw_3) \parallel (L_{24} \oplus kw_4)$



Hình 4: Thủ tục mã hóa Camellia cho khóa 256 bit

2.2.3.1.2. Phép giải mã Camellia

Quá trình giải mã cho khóa 256 bit được chỉ ra trên Hình 5, và là như nhau trong phép mã hóa, chỉ khác là vị trí và thứ tự các khóa vòng được đảo lại.

Phép giải mã được xác định như sau:

(1) $R_{24} \parallel L_{24} = C \oplus (kw_3 \parallel kw_4)$

(2) for $i = 24$ down to 1:

$R_{i-1} = \mathrm{F}(R_i, k_i) \oplus L_i$

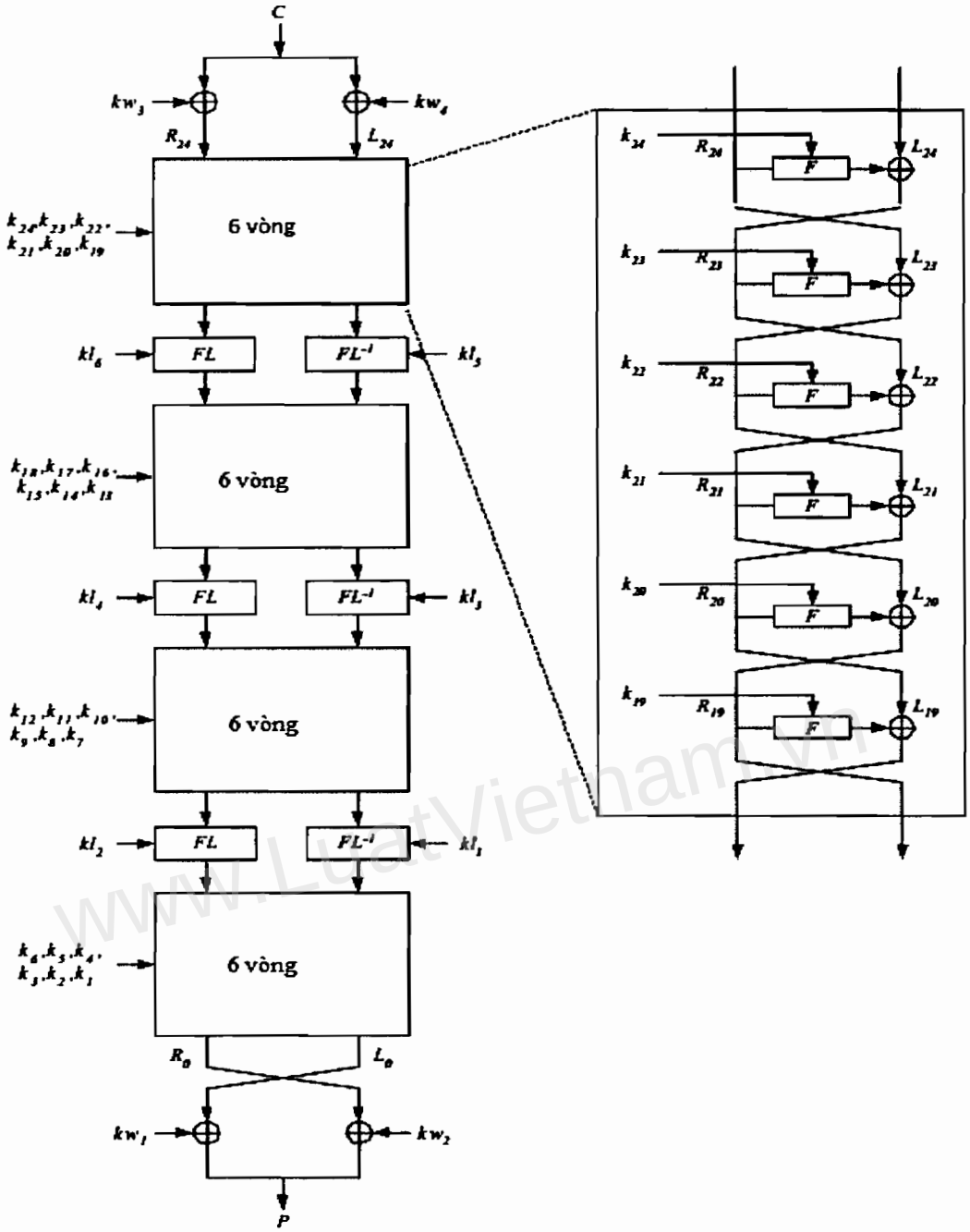
$L_{i-1} = R_i$

if (i = 19 or 13 or 7) then

$R_{i-1} = \mathrm{FL}(R_{i-1}, kl_{(i-1)/3})$

$L_{i-1} = FL^{-1}(L_{i-1}, kl_{(i-1)/3-1})$

(3) $\mathrm{P} = (\mathrm{L_0} \oplus \mathrm{kw_1}) \parallel (\mathrm{R_0} \oplus \mathrm{kw_2})$



Hình 5: Thủ tục giải mã Camellia cho 256 bit

2.2.3.1.3. F-hàm

F-hàm được chỉ ra trên Hình 8. F-hàm bao gồm phép toán cộng bit XOR, tiếp đó áp dụng tám S-box song song kích thước 8x8 bit, tiếp theo tầng khuếch tán (P-hàm). x_j, y_j, z_j, z'_j là các biến, mỗi biến 8 bit; các biến L_i, k_i, L'_i là các biến 64 bit. Đầu vào 64 bit L_i trước hết được cộng XOR với khóa vòng 64-bit k_i, sau đó được chia thành 8 đoạn 8-bit y_j, như sau:

$$y_1 \| y_2 \| y_3 \| y_4 \| y_5 \| y_6 \| y_7 \| y_8 = L_i \oplus k_i$$

ở đây,

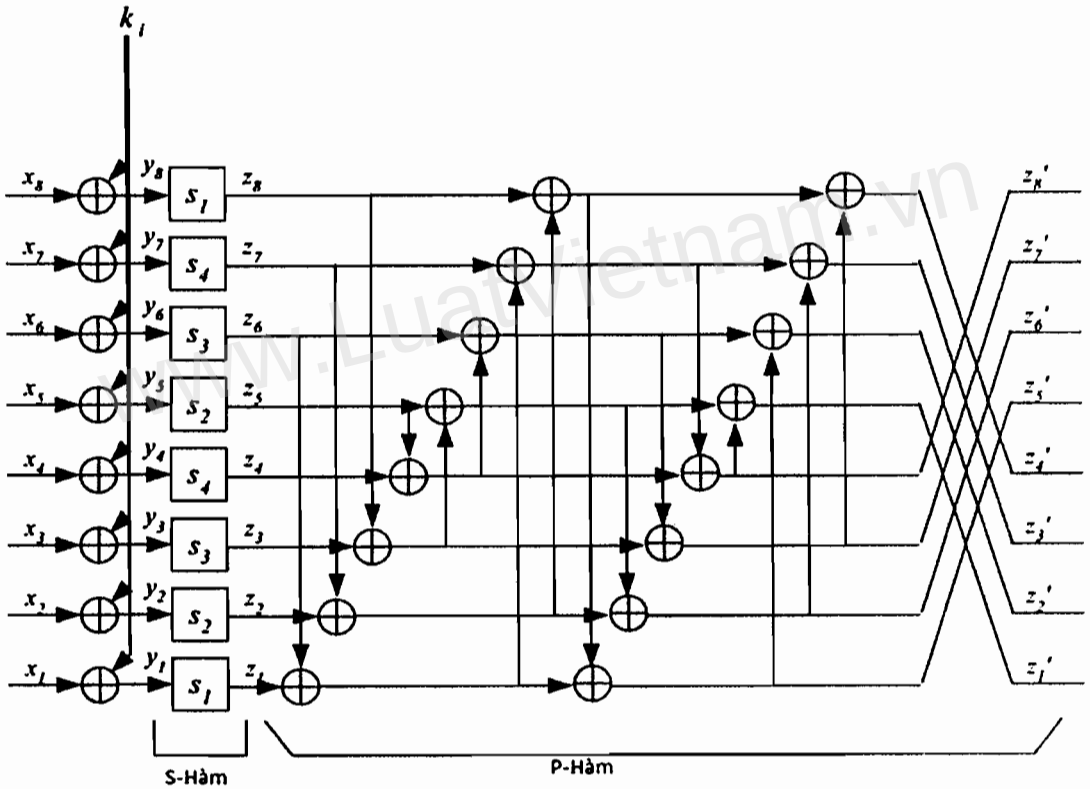
$$L_i = x_1 \| x_2 \| x_3 \| x_4 \| x_5 \| x_6 \| x_7 \| x_8.$$

Mỗi y_j sau đó được đi qua S-box kích thước 8x8 bit s_t để đưa ra 8 phân đoạn độ dài 8 bit z_j, ở đây

$z_1 = s_1[y_1], z_2 = s_2[y_2], z_3 = s_3[y_3], z_4 = s_4[y_4], z_5 = s_5[y_5], z_6 = s_6[y_6], z_7 = s_7[y_7], z_8 = s_8[y_8]$

8 phân đoạn 8-bit z_j được tác động bởi P-hàm, là lớp khuếch tán cho ra 8 phân đoạn 8-bit z'_j, trong đó

$$z'_1 = z_1 \oplus z_3 \oplus z_4 \oplus z_6 \oplus z_7 \oplus z_8, \qquad z'_5 = z_1 \oplus z_2 \oplus z_6 \oplus z_7 \oplus z_8,$$
$$z'_2 = z_1 \oplus z_2 \oplus z_4 \oplus z_5 \oplus z_7 \oplus z_8, \qquad z'_6 = z_2 \oplus z_3 \oplus z_5 \oplus z_7 \oplus z_8,$$
$$z'_3 = z_1 \oplus z_2 \oplus z_5 \oplus z_6 \oplus z_7 \oplus z_8, \qquad z'_7 = z_3 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_8,$$
$$z'_4 = z_2 \oplus z_3 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_7 \qquad z'_8 = z_1 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_7.$$



Hình 6: F-hàm

P-hàm có thể được biểu diễn cách khác, dưới dạng véc tơ ma trận như sau:

$$\begin{pmatrix} z_8 \\ z_7 \\ \vdots \\ z_1 \end{pmatrix} \mapsto \begin{pmatrix} z'_8 \\ z'_7 \\ \vdots \\ z'_1 \end{pmatrix} = P \begin{pmatrix} z_8 \\ z_7 \\ \vdots \\ z_1 \end{pmatrix},$$

ở đây:

$$P = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Sau đó đầu ra 64-bit của F-hàm L'_i được thiết lập bằng cách ghép các biến 8-bit z'_j:

$$L'_i = z'_1 \| z'_2 \| z'_3 \| z'_4 \| z'_5 \| z'_6 \| z'_7 \| z'_8.$$

S-box s1

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

S-box s2

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

S-box s3

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

S-box s4

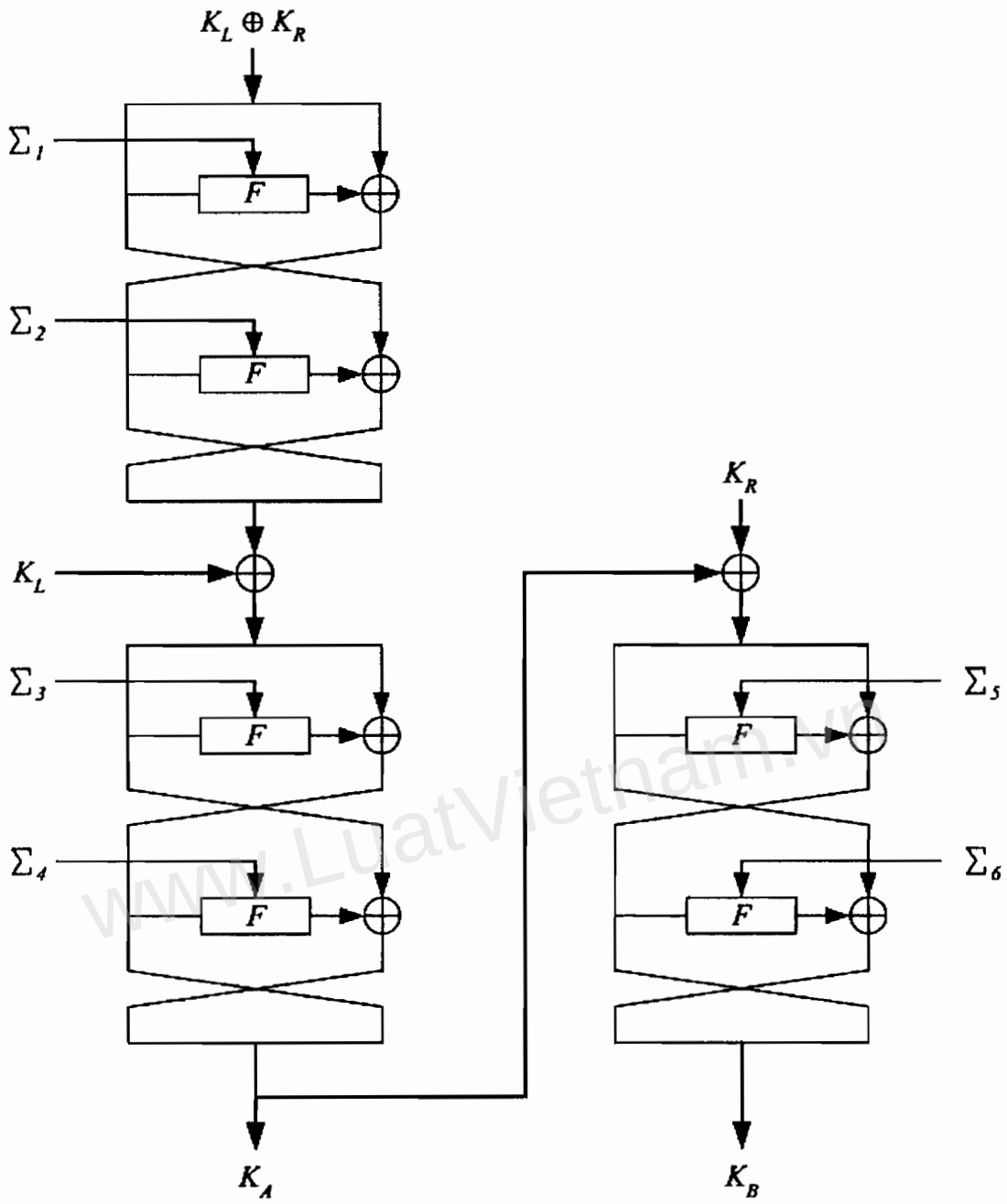
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	70	2c	b3	c0	e4	57	ea	ae	23	6b	45	a5	ed	4f	1d	92
1	86	af	7c	1f	3e	dc	5e	0b	a6	39	d5	5d	d9	5a	51	6c
2	8b	9a	fb	b0	74	2b	f0	84	df	cb	34	76	6d	a9	d1	04
3	14	3a	de	11	32	9c	53	f2	fe	cf	c3	7a	24	e8	60	69
4	aa	a0	a1	62	54	1e	e0	64	10	00	a3	75	8a	e6	09	dd
5	87	83	cd	90	73	f6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

2.2.3.2. Chu trình khóa

Với quy định khóa có độ dài 256 bit, khóa K là khóa 128-bit K_L và khóa 128-bit K_R. Như vậy,

$$K = K_L \parallel K_R$$

Lược đồ tạo khóa sử dụng F-hàm của mô-đun mã hóa, và là giống nhau cho cả phép mã hóa và giải mã. Khóa *K* được mã bằng các phương tiện của F-hàm, sử dụng các hằng của lược đồ tạo khóa, ở đây các hằng Σ_i, được xác định như những giá trị liên tục biểu diễn trong hệ Hexa của căn bậc hai số nguyên tố thứ *i*. Tiếp đó các khóa vòng được tạo, một phần từ những giá trị được dịch vòng của khóa K ($K = K_L \parallel K_R$) và phần còn lại từ các giá trị được dịch vòng của các khóa "được mã hóa" K_A và K_B (ở đây K_A, K_B có độ dài 128 bit).



Hình 7: Phần chính của lược đồ khóa

Đối với khóa 256-bit, đầu ra của phần chính của lược đồ tạo khóa là khóa con 128 bit K_A và khóa con 128-bit K_B. Lược đồ tạo khóa bao gồm ba phép toán 2-vòng. Mỗi phép toán 2 vòng được "khóa hóa" bằng một cặp hằng Σ_i.

Đầu vào 128-bit của phép toán 2-vòng thứ nhất nằm bên trái của Hình 9 là $K_L \oplus K_R$ và phép toán này được "khóa hóa" bởi hai hằng 64-bit Σ_1 và Σ_2. Tiếp đó đầu ra 128-bit từ phép toán 2-vòng thứ nhất được cộng bit XOR với K_L trước khi là đầu vào của phép toán 2-vòng thứ hai ở bên trái Hình 9. Phép toán 2-vòng thứ hai này được "khóa hóa" bởi các hằng 64 bit là Σ_3 và Σ_4. Đầu ra 128-bit của phép toán 2-vòng thứ hai là K_A, sau đó K_A lại được XOR với khóa con 128-bit K_R trước khi kết quả thu được làm đầu vào của phép toán 2-vòng thứ ba được chỉ ra ở phía phải của Hình 9. Phép toán 2-vòng thứ ba này được "khóa hóa" bởi hai hằng 64 bit là Σ_5 và Σ_6. Đầu ra 128 bit của phép toán 2-vòng thứ ba này là K_B.

Phép toán lập lược đồ tạo khóa đầy đủ được mô tả như sau (K_a, K_A và K_B có độ dài 128 bit):

(1) $K_a = 2RoundFeistel(K_L \oplus K_R, \Sigma_1, \Sigma_2)$

(2) $K_A = 2RoundFeistel(K_a \oplus K_L, \Sigma_3, \Sigma_4)$

(3) $K_B = 2RoundFeistel(K_A \oplus K_R, \Sigma_5, \Sigma_6)$ (chỉ dùng cho khóa 192/256 bit)

ở đây đầu vào 128-bit cho 2RoundFeistel được tách thành hai phần 64-bit $L_0 \parallel R_0$, đầu ra 128-bit từ 2RoundFeistel cũng được tách thành hai phần 64 bit $L_2 \parallel R_2$ và cả hai đầu vào "khóa vòng" 64-bit của 2RoundFeistel là Σ_i và Σ_{i+1}.

2RoundFeistel được mô tả như sau:

(1) với j = 0,1:

$$L_{j+1} = F(L_j, \Sigma_{i+j}) \oplus R_j$$

$$R_{j+1} = R_j$$

Các hằng của lược đồ khóa 64-bit được xác định trên Bảng 4

	Hằng
Σ_1	a09e667f3bcc908b
Σ_2	b67ae8584caa73b2
Σ_3	c6ef372fe94f82be
Σ_4	54ff53a5fld36f1c
Σ_5	10e527fade682d1d
Σ_6	b05688c2b3e6c1fd

Bảng 4: Các hằng trong lược đồ khóa

Cuối cùng, các khóa vòng 64-bit, $k, kw, và\ kl$ được dẫn xuất từ các khóa con 128-bit, K_L, K_R, K_A và K_B.

Hàm	Khóa vòng	Giá trị
	kw_1	$(K_L \lll_0)_L$
	kw_2	$(K_L \lll_0)_R$
F (vòng 1)	k_1	$(K_B \lll_0)_L$
F (vòng 2)	k_2	$(K_B \lll_0)_R$
F (vòng 3)	k_3	$(K_R \lll_{15})_L$
F (vòng 4)	k_4	$(K_R \lll_{15})_R$
F (vòng 5)	k_5	$(K_A \lll_{15})_L$
F (vòng 6)	k_6	$(K_A \lll_{15})_R$
FL	kl_1	$(K_R \lll_{30})_L$
FL^{-1}	kl_2	$(K_R \lll_{30})_R$

F (vòng 7)	k_7	$(K_B \lll_{30})_L$
F (vòng 8)	k_8	$(K_B \lll_{30})_R$
F (vòng 9)	k_9	$(K_L \lll_{45})_L$
F (vòng 10)	k_{10}	$(K_L \lll_{45})_R$
F (vòng 11)	k_{11}	$(K_A \lll_{45})_L$
F (vòng 12)	k_{12}	$(K_A \lll_{45})_R$
FL	kl_3	$(K_L \lll_{60})_L$
FL^{-1}	kl_4	$(K_L \lll_{60})_R$
F (vòng 13)	k_{13}	$(K_R \lll_{60})_L$
F (vòng 14)	k_{14}	$(K_R \lll_{60})_R$
F (vòng 15)	k_{15}	$(K_B \lll_{60})_L$
F (vòng 16)	k_{16}	$(K_B \lll_{60})_R$
F (vòng 17)	k_{17}	$(K_L \lll_{77})_L$
F (vòng 18)	k_{18}	$(K_L \lll_{77})_R$
FL	kl_5	$(K_A \lll_{77})_L$
FL^{-1}	kl_6	$(K_A \lll_{77})_R$
F (vòng 19)	k_{19}	$(K_R \lll_{94})_L$
F (vòng 20)	k_{20}	$(K_R \lll_{94})_R$
F (vòng 21)	k_{21}	$(K_A \lll_{94})_L$
F (vòng 22)	k_{22}	$(K_A \lll_{94})_R$
F (vòng 23)	k_{23}	$(K_L \lll_{111})_L$
F (vòng 24)	k_{24}	$(K_L \lll_{111})_R$
	kw_3	$(K_B \lll_{111})_L$
	kw_4	$(K_B \lll_{111})_R$

Bảng 5: Khóa vòng cho khóa bí mật 256 bit

2.2.3.3. Chế độ hoạt động của Camellia

Các chế độ hoạt động của Camellia được quy định tại Mục 2.3 của Quy chuẩn này.

2.2.3.4. Khóa

Độ dài khóa sử dụng trong mã khối Camellia được quy định

Độ dài khối	Tên thuật toán	Độ dài khóa quy định áp dụng
128 bit	Camellia	Không nhỏ hơn 256 bit

2.3. Chế độ hoạt động của mã khối

Mục này của Quy chuẩn sử dụng một số ký hiệu sau:

C	Khối bản mã
CTR	Giá trị đếm
i	Biến lặp
j	Kích thước của biến bản rõ/bản mã
K	Khóa mật mã
n	Độ dài khối bản mã/bản rõ đối với một mã khối
m	Số khối bản mã được lưu trữ
P	Khối bản rõ
q	Số các biến bản rõ/bản mã
r	Kích thước bộ đệm phản hồi
SV	Biến khởi đầu
X	Khối đầu vào mã khối
Y	Khối đầu ra mã khối

2.3.1. Chế độ xích liên kết khối mã CBC (Cipher Block Chaining)

Chế độ CBC được xác định bởi một tham số xen kẽ $m > 0$, số khối bản mã phải được lưu trữ trong khi xử lý trong chế độ này. Giá trị của m cần nhỏ (thông thường $m = 1$) và lớn nhất là 1024.

a) Các biến được sử dụng trong chế độ CBC:

1) Dãy q khối bản rõ có độ dài n bit $P_1, P_2, \ldots, P_q$.

2) Khóa bí mật K.

3) Dãy các biến khởi đầu có độ dài n bit $SV_1, SV, \ldots, SV_m$.

b) Các biến đầu ra là dãy q biến bản mã có độ dài n bit $C_1, C_2, \ldots, C_q$

2.3.1.1. Phép mã hóa

Phép mã hóa của chế độ CBC như sau:

$$C_i = e_K(P_i \oplus SV_i), 1 \le i \le \min(m, q)$$

Nếu $q > m$, các khối bản rõ theo sau được mã hóa như sau:

$$C_i = e_K(P_i \oplus C_{i-m}), m + 1 \le i \le \mathrm{q}$$

CHÚ THÍCH: Tại bất kỳ thời điểm nào trong quá trình tính toán, giá trị của khối bản mã m gần nhất cần được lưu trữ.

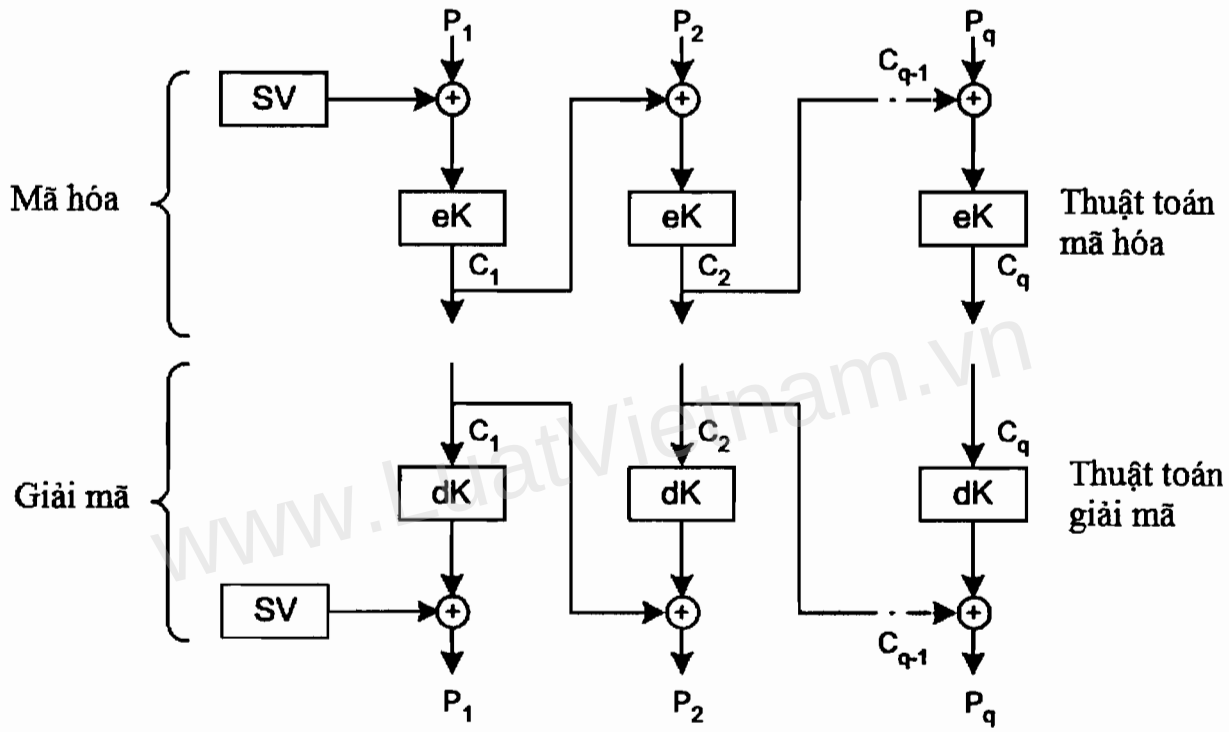
2.3.1.2. Phép giải mã

Phép giải mã của chế độ CBC như sau:

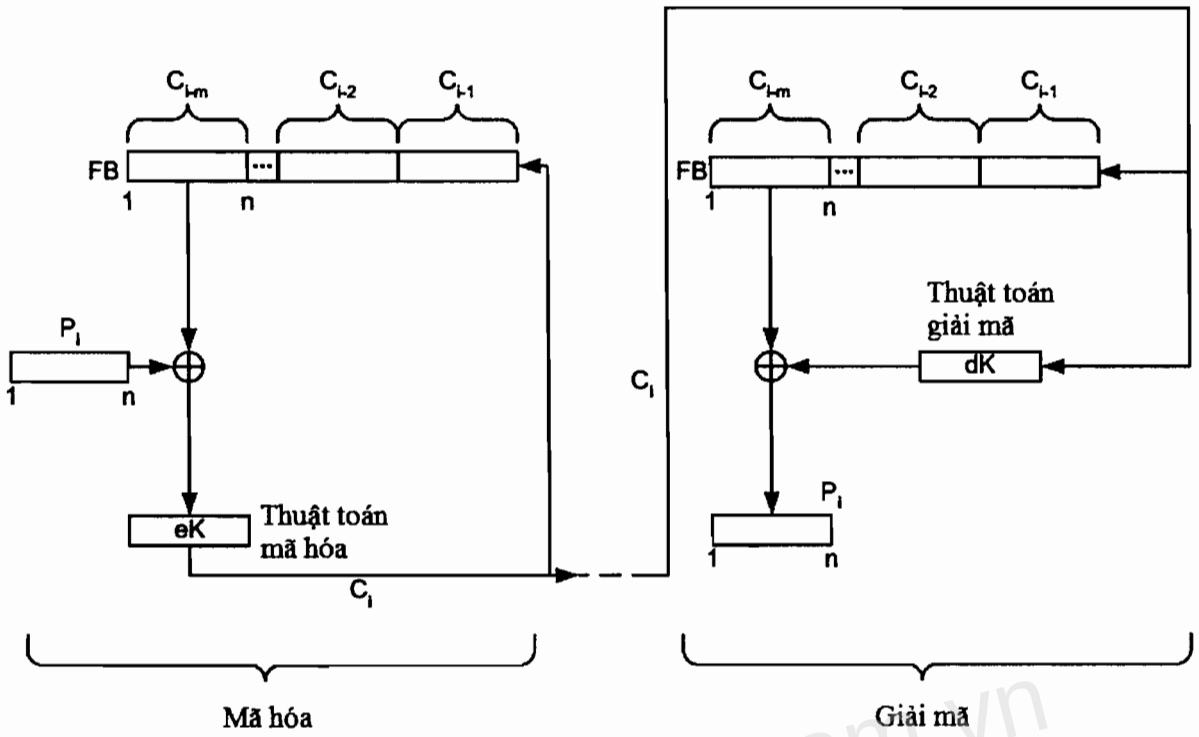
$$P_i = d_K(C_i) \oplus SV_i, 1 \leq i \leq \min(m, q)$$

Nếu $q > m$, các khối bản rõ theo sau được mã hóa như sau:

$$P_i = d_K(C_i) \oplus C_{i-m}, m + 1 \leq i \leq q$$



Hình 8: Chế độ xích liên kết khối mã với $m = 1$



Hình 9: Chế độ xích liên kết khối mã

- Biến khởi đầu SV được sinh ngẫu nhiên, giữ bí mật và thông báo cho nhau.

- Yêu cầu về đệm (padding)

+ Nếu độ dài bản rõ không phải là bội số của n thì yêu cầu sử dụng phương pháp đệm (padding) sau để bổ sung bản rõ sao cho độ dài của bản rõ là bội số của n:

Thêm một bit 1 vào cuối bản rõ, tiếp sau là các bit 0 (có thể không cần) để được bản rõ có độ dài là bội số của n. Đây chính là phương pháp đệm 2 trong tiêu chuẩn ISO/IEC 9797-1 hoặc ISO/IEC 10118-1.

2.3.2. Chế độ phản hồi mã CFB (Cipher FeedBack)

Chế độ CFB được xác định bởi 3 tham số:

- Kích thước của bộ đệm phản hồi, r, trong đó $n \leq r \leq 1024n$ và $r < qn$

- Kích thước của biến phản hồi, k, trong đó $1 \leq k \leq n$

- Kích thước của biến bản rõ, j, trong đó $1 \leq j \leq k$

Trong Quy chuẩn này thì giá trị của j và k bằng nhau.

Các biến được sử dụng trong chế độ CFB:

a) Các biến đầu vào

1) Dãy q biến bản rõ độ dài j bit $P_1, P_2, \ldots, P_q$.

2) Khóa bí mật K.

3) Biến khởi đầu độ dài r bit.

b) Các kết quả trung gian:

1) Dãy q khối đầu vào của mã khối độ dài n bit $X_1, X_2, \dots, X_q$.

2) Dãy q khối đầu ra của mã khối độ dài n bit $Y_1, Y_2, \dots, Y_q$.

3) Dãy q biến độ dài j bit $E_1, E_2, \dots, E_q$.

4) Dãy $q - 1$ biến phản hồi độ dài k bit $F_1, F_2, \dots, F_{q-1}$.

5) dãy q nội dung bộ đệm phản hồi độ dài r bit $FB_1, FB_2, \dots, FB_q$.

c) Các biến đầu ra là dãy q biến bản mã độ dài j bit $C_1, C_2, \dots, C_q$.

2.3.2.1. Phép mã hóa

Bộ đệm phản hồi FB được gán giá trị khởi đầu.

$$FB_1 = SV$$

Phép toán mã hóa mỗi biến bản rõ thực hiện trong 6 bước sau:

a) $X_i = n \sim FB_i$ (Lựa chọn n bit tận cùng bên trái của FB).

b) $Y_i = e_K(X_i)$ (Sử dụng mã khối).

c) $E_i = j \sim Y_i$ (Lựa chọn j bit tận cùng bên trái của Y_i).

d) $C_i = P_i \oplus E_i$ (Tạo biến bản mã).

e) $F_i = I(k - j) | C_i$ (Tạo biến phản hồi).

f) $FB_{i+1} = S_k(FB_i | F_i)$ (Hàm dịch chuyển trên FB).

Các bước trên lặp với biến đếm $i = 1, 2, \dots, q$, trong vòng lặp cuối dừng lại tại bước (d). j bit tận cùng bên trái của khối đầu ra Y của mã khối được sử dụng để mã hóa j-bit biến bản rõ theo phép cộng modulo 2. Các bit còn lại của Y được bỏ đi. Các bit biến bản rõ/bản mã có chỉ số từ 1 đến j.

Biến bản mã được tăng thêm bằng cách đặt $k - j$ bit trong vị trí bit tận cùng bên trái của mình để tạo biến phản hồi F độ dài k-bit. Sau đó các bit của bộ đệm phản hồi FB được dịch trái đi k vị trí và F được thêm vào vị trí k tận cùng bên phải để tạo ra giá trị mới của bộ đệm phản hồi FB. Trong phép toán dịch, k bit tận cùng bên trái của FB được bỏ đi. n bit tận cùng bên trái mới của FB được sử dụng như là đầu vào tiếp theo của X trong quá trình mã hóa.

2.3.2.2. Phép giải mã

Bộ đệm phản hồi FB được gán giá trị khởi đầu.

$$FB_1 = SV$$

Phép toán giải mã mỗi biến bản mã thực hiện trong 6 bước sau:

a) $X_i = n \sim FB_i$ (Lựa chọn n bit tận cùng bên trái của FB).

b) $Y_i = e_K(X_i)$ (Sử dụng mã khối).

c) $E_i = j \sim Y_i$ (Lựa chọn j bit tận cùng bên trái của Y_i).

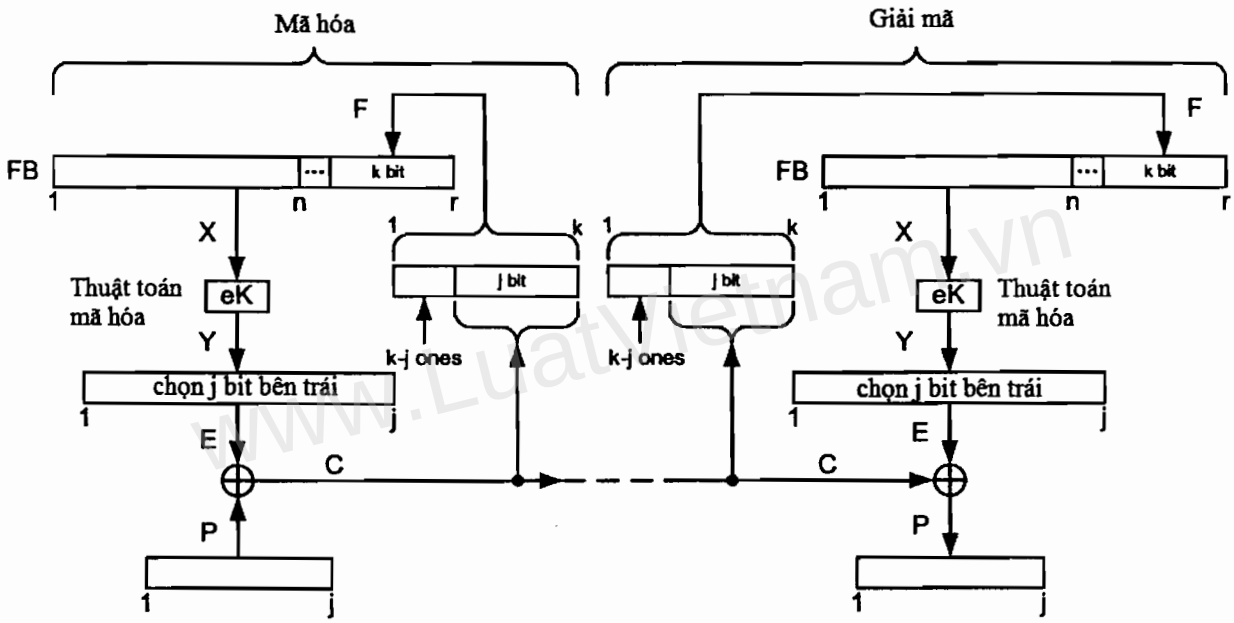
d) $P_i = C_i \oplus E_i$ (Tạo biến bản rõ).

e) $F_i = I(k - j) | C_i$ (Tạo biến phản hồi).

f) $FB_{i+1} = S_k(FB_i|F_i)$ (Hàm dịch chuyển trên FB).

Các bước trên lặp với biến đếm $i = 1, 2, \dots, q$, trong vòng lặp cuối dừng lại tại bước (d). j bit tận cùng bên trái của khối đầu ra Y của mã khối được sử dụng để giải mã j-bit biến bản mã theo phép cộng modulo 2. Các bit còn lại của Y được bỏ đi. Các bit biến bản rõ/bản mã có chỉ số từ *1* đến j.

Biến bản mã được tăng thêm bằng cách đặt $k - j$ bit trong vị trí bit tận cùng bên trái của mình để tạo biến phản hồi F độ dài k-bit. Sau đó các bit của bộ đệm phản hồi FB được dịch trái đi k vị trí và F được thêm vào vị trí k tận cùng bên phải để tạo ra giá trị mới của bộ đệm phản hồi FB. Trong phép toán dịch, k bit tận cùng bên trái của FB được bỏ đi. n bit tận cùng bên trái mới của FB được sử dụng như là đầu vào tiếp theo của X trong quá trình giải mã.



Hình 10: Chế độ phản hồi mã CFB

2.3.3. Chế độ phản hồi đầu ra OFB (Output Feedback):

Chế độ OFB được xác định bằng một tham số j là kích thước biến bản rõ với $1 \leq j \leq n$.

Các biến được sử dụng trong chế độ OFB là:

a) Các biến đầu vào:

1) Dãy q biến bản rõ độ dài j bit $P_1, P_2, \dots, P_q$.

2) Khóa bí mật K.

3) Biến khởi đầu độ dài n bit.

b) Các kết quả trung gian:

1) Dãy q khối đầu vào của mã khối độ dài n bit $X_1, X_2, \dots, X_q$.

2) Dãy q khối đầu ra của mã khối độ dài n bit $Y_1, Y_2, \dots, Y_q$.

3) Dãy q biến độ dài j bit $E_1, E_2, \dots, E_q$.

c) Các biến đầu ra, nghĩa là dãy q biến bản mã độ dài j bit $C_1, C_2, \dots, C_q$.

2.3.3.1. Phép mã hóa

Khối đầu vào X được gán giá trị khởi đầu

$$X_1 = SV$$

Phép toán mã hóa mỗi biến bản rõ thực hiện trong 4 bước sau:

a) $Y_i = e_k(X_i)$ (Sử dụng mã khối).

b) $E_i = j \sim Y_i$ (Chọn j bit tận cùng bên trái).

c) $C_i = P_i \oplus E_i$ (Tạo biến bản mã).

d) $X_{i+1} = Y_i$ (Phép toán phản hồi).

Các bước trên lặp với biến đếm $i = 1, 2, \ldots, q$, trong vòng lặp cuối dừng lại tại bước (c). Các biến bản mã và bản rõ có các bit với chỉ số từ *1* đến j.

Kết quả của mỗi lần sử dụng mã khối là Y_i và được đưa trở lại thành giá trị tiếp theo của X, đặt là X_{i+1}. j bit tận cùng bên trái của Y_i được sử dụng để mã hóa biến đầu vào.

2.3.3.2. Phép giải mã

Khối đầu vào X được gán giá trị khởi đầu

$$X_1 = SV$$

Phép toán giải mã mỗi biến bản mã thực hiện trong 4 bước sau:

a) $Y_i = e_k(X_i)$ (Sử dụng mã khối).

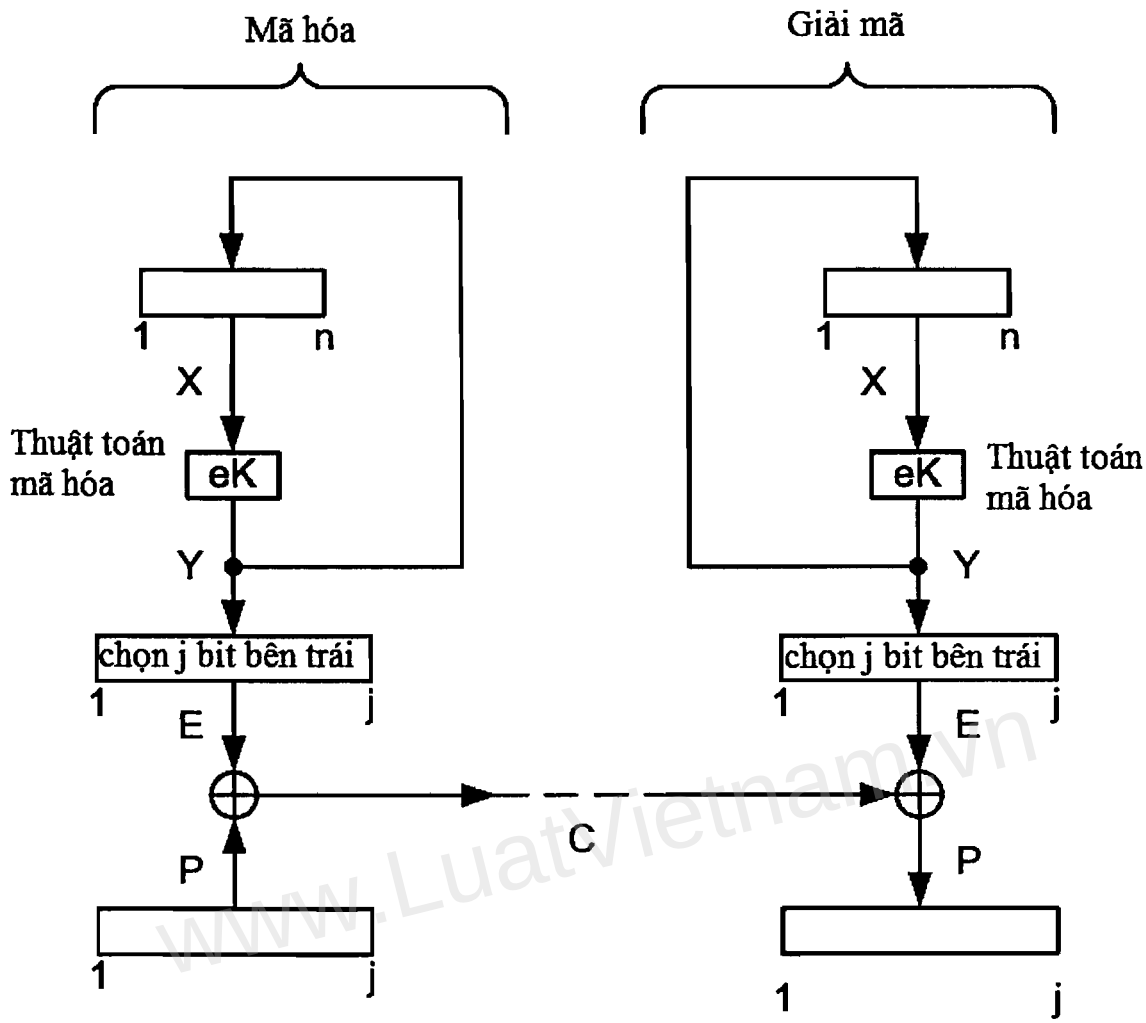
b) $E_i = j \sim Y_i$ (Chọn j bit tận cùng bên trái).

c) $P_i = C_i \oplus E_i$ (Tạo biến bản mã).

d) $X_{i+1} = Y_i$ (Phép toán phản hồi).

Các bước trên lặp với biến đếm $i = 1, 2, \ldots, q$, trong vòng lặp cuối dừng lại tại bước (c). Các biến bản mã và bản rõ có các bit với chỉ số từ *1* đến j.

Kết quả của mỗi lần sử dụng mã khối là Y_i và được đưa trở lại thành giá trị tiếp theo của X, đặt là X_{i+1}. j bit tận cùng bên trái của Y_i được sử dụng để giải mã biến đầu vào.



Hình 11: Chế độ phản hồi đầu ra OFB

2.3.4. Chế độ đếm CTR (Counter)

Chế độ CTR được xác định bằng một tham số j trong đó $1 \leq j \leq n$.

Các biến được sử dụng trong chế độ *CTR* là:

a) Các biến đầu vào:

1) Dãy q biến bản rõ $P_1, P_2, \ldots, P_q$, mỗi biến có độ dài j bit.

2) Khóa bí mật K.

3) Biến khởi đầu SV có độ dài n bit.

b) Các kết quả trung gian:

1) Dãy q khối đầu vào của mã khối có độ dài n bit $CTR_1, CTR_2, \ldots, CTR_q$.

2) Dãy q khối đầu ra của mã khối có độ dài n bit $Y_1, Y_2, \ldots, Y_q$.

3) Dãy q biến có độ dài j bit $E_1, E_2, \ldots, E_q$.

c) Các biến đầu ra, nghĩa là dãy q biến bản mã độ dài j bit $C_1, C_2, \ldots, C_q$.

2.3.4.1. Phép mã hóa

CTR được gán giá trị khởi đầu

$$CTR_1 = SV$$

Các phép toán để mã hóa mỗi biến bản rõ theo 4 bước sau:

a) $Y_i = e_K(CTR_i)$ (Sử dụng mã khối).

b) $E_i = j\sim Y_i$ (Chọn j bit tận cùng bên trái của Y_i).

c) $C_i = P_i \oplus E_i$ (Tạo ra biến bản mã).

d) $CTR_{i+1} = (CTR_i + 1) mod\ 2^n$ (Tạo ra giá trị đếm mới CTR).

Các bước trên lặp với biến đếm $i = 1, 2, \ldots, q$, trong vòng lặp cuối dừng lại tại bước (c). Các biến bản mã và bản rõ có các bit có chỉ số từ *1* đến j.

Giá trị đếm được mã hóa để đưa ra khối đầu ra Y_i và j bit tận cùng bên trái của khối đầu ra Y_i được sử dụng để mã hóa giá trị đầu vào. CTR sau đó được tăng 1 ($modulo\ 2^n$) để tạo ra giá trị đếm mới.

2.3.4.2. Phép giải mã

CTR được gán giá trị khởi đầu

$$CTR_1 = SV$$

Các phép toán để giải mã mỗi biến bản rõ theo 4 bước sau:

a) $Y_i = e_K(CTR_i)$ (Sử dụng mã khối).

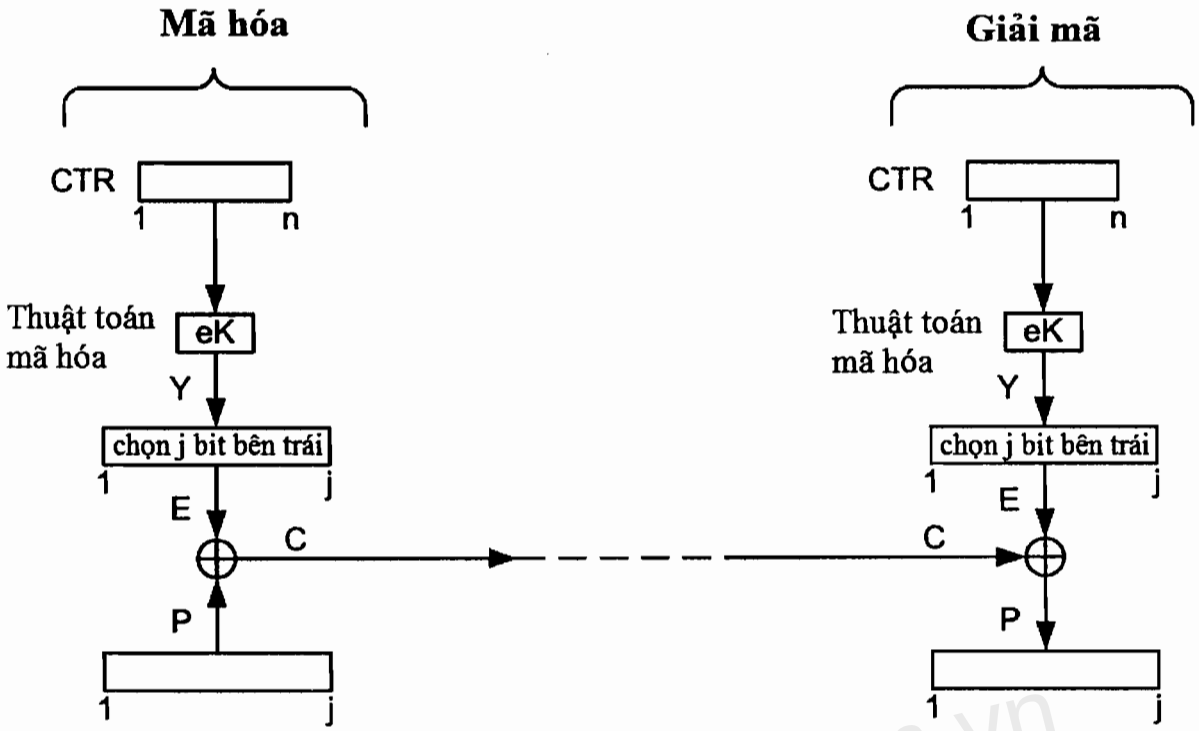
b) $E_i = j\sim Y_i$ (Chọn j bit tận cùng bên trái của Y_i).

c) $P_i = C_i \oplus E_i$ (Tạo ra biến bản rõ).

d) $CTR_{i+1} = (CTR_i + 1)\ mod\ 2^n$ (Tạo ra giá trị đếm mới CTR).

Các bước trên lặp với biến đếm $i = 1, 2, \ldots, q$, trong vòng lặp cuối dừng lại tại bước (c). Các biến bản mã và bản rõ có các bit có chỉ số từ *1* đến j.

Giá trị đếm được mã hóa để đưa ra khối đầu ra Y_i và j bit tận cùng bên trái của khối đầu ra Y_i được sử dụng để mã hóa giá trị đầu vào. CTR sau đó được tăng 1 (modulo 2^n) để tạo ra giá trị đếm mới.



Hình 12: Chế độ đếm *CTR*

2.4. Mã dòng

Mã dòng là mã thực hiện biến đổi từng bit dữ liệu của bản rõ sang bản mã sử dụng khóa độ dài k bit. Quy chuẩn này quy định sử dụng các mã khối được quy định tại Mục 2.2 sử dụng chế độ CFB, OFB hoặc CTR để thực hiện chức năng mã dòng.

3. QUY ĐỊNH VỀ QUẢN LÝ

3.1. Các mức giới hạn của đặc tính kỹ thuật mật mã và yêu cầu quản lý của các thuật toán mật mã để mã hóa dữ liệu nêu tại Quy chuẩn này là các chỉ tiêu chất lượng phục vụ được quản lý theo quy định về quản lý chất lượng sản phẩm, dịch vụ mật mã dân sự được quy định tại Luật An toàn thông tin mạng ngày 19 tháng 11 năm 2015.

3.2. Hoạt động kiểm tra chất lượng sản phẩm, dịch vụ mật mã được cơ quan quản lý nhà nước có thẩm quyền tiến hành định kỳ hàng năm hoặc đột xuất.

4. TRÁCH NHIỆM CỦA TỔ CHỨC, CÁ NHÂN

4.1. Các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự có trách nhiệm đảm bảo tuân thủ Quy chuẩn này và chịu sự kiểm tra của cơ quan quản lý nhà nước theo quy định.

4.2. Doanh nghiệp cung cấp sản phẩm, dịch vụ mật mã dân sự cho các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) có trách nhiệm thực hiện công bố hợp quy sản phẩm, dịch vụ mật mã dân sự phù hợp với Quy chuẩn này. Việc công bố hợp quy thực hiện theo Thông tư số 28/2012/TT-BKHCN ngày 12 tháng 12 năm 2012 của Bộ Khoa học và Công nghệ.

4.3. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã – Ban Cơ yếu Chính phủ có trách nhiệm tiếp nhận đăng ký công bố hợp quy, thực hiện quản lý, hướng dẫn và kiểm tra việc công bố hợp quy.

5. TỔ CHỨC THỰC HIỆN

5.1. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã – Ban Cơ yếu Chính phủ có trách nhiệm hướng dẫn, tổ chức triển khai quản lý kỹ thuật mật mã của thuật toán mã hóa dữ liệu theo Quy chuẩn này.

5.2. Trong trường hợp các quy định nêu tại Quy chuẩn kỹ thuật quốc gia này có sự thay đổi, bổ sung hoặc được thay thế thì thực hiện theo quy định tại văn bản mới./.

Phụ lục A

(Quy định)

Mô tả DES

A.1. Mở đầu

Thuật toán DES là mã khối đối xứng có thể xử lí các khối dữ liệu 64 bit, sử dụng khóa bí mật độ dài 64 bit. Mỗi bit thứ tám của khóa mật mã thường được sử dụng để kiểm tra tính chẵn lẻ và được bỏ qua.

A.2. Phép mã hóa DES

Phép mã hóa được chỉ ra trên Hình A.1

Bản rõ 64-bit trước hết được biến đổi qua hoán vị ban đầu ***IP***. Sau đó khối được chia thành hai nửa L_0 và R_0, mỗi nửa gồm 32 bit. Tiếp đó thực hiện 16 vòng biến đổi giống nhau được gọi là hàm f, trong đó dữ liệu được kết hợp với khóa. Trong mỗi vòng, nửa phải là đầu vào của hàm f được khóa hóa, hàm này nhận đầu vào 32 bit và khóa con 48 bit K_i và cho đầu ra 32-bit. Đầu ra này tiếp đó được XOR với nửa trái để tạo ra nửa trái mới đã được biến đổi. Tại phần cuối của mỗi vòng, trừ vòng cuối cùng, hai nửa trái và phải đổi chỗ cho nhau để tạo ra L_i và R_i tương ứng. Sau khi thực hiện vòng cuối cùng, hai nửa trái và phải được ghép lại với nhau và khối 64-bit nhận được lại được biến đổi qua phép hoán vị cuối IP^{-1} là hoán vị nghịch đảo của hoán vị ban đầu ***IP***. Đầu ra là bản mã 64-bit.

Phép mã hóa được xác định như sau (P và C là dữ liệu, K_i là khóa).

(1) $IP(P) = L_0 \| R_0$

(2) Với $i = 1, 2, \ldots, 16$:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

(3) $C = IP^{-1}(R_{16} \| L_{16})$

A.3. Phép giải mã DES

Phép giải mã cũng giống như phép mã hóa. Sự khác nhau chỉ ở chỗ, các khóa con K_i được sử dụng theo thứ tự ngược lại.

A.4. Các hàm DES

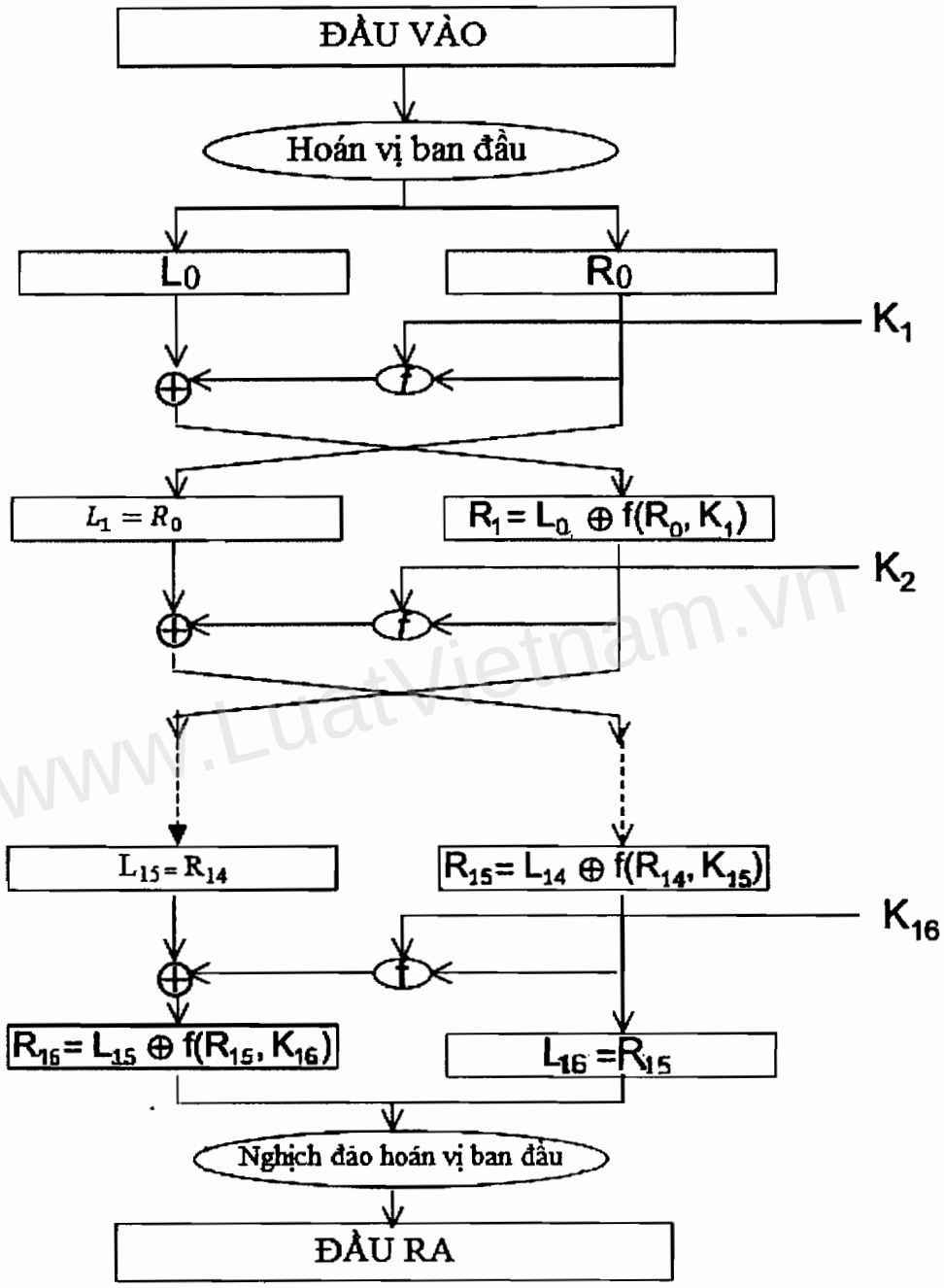
A.4.1. Phép hoán vị ban đầu IP

Phép hóa vị ban đầu IP được chỉ ra tại Bảng A.1. Hoán vị này nhận đầu vào 64-bit và cho đầu ra 64-bit, theo đó bit thứ nhất được hoán vị thành bit thứ 58, bit thứ hai thành bit thứ 50,v.v, bit cuối cùng thành bit thứ 7.

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3

61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Bảng A.1 – Hoán vị khởi tạo



Hình A.1 – Thủ tục mã hóa

A.4.2. Hoán vị nghịch đảo IP^{-1}

Hoán vị nghịch đảo IP^{-1} được chỉ ra trên Bảng A.2, nhận đầu vào 64-bit và cho đầu ra 64-bit. Đầu ra của Thuật toán nhận bit thứ 40 của khối đầu ra trước đó làm bit thứ nhất, bit thứ 8 làm bit thứ hai,v.v và bit thứ 25 của khối đầu ra trước đó làm bit cuối cùng.

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

Bảng A.2 – Hoán vị nghịch đảo IP^{-1}

A.4.3. Hàm *f*

Hàm ***f*** được chỉ ra trên Hình A.2

Hàm ***f*** nhận đầu vào 32 bit R và mở rộng thành R' có độ dài 48-bit bằng cách sử dụng phép hoán vị mở rộng $\boldsymbol{E}$. Sau đó 48-bit R' được XOR với khóa con 48-bit K, thu được dữ liệu 48-bit, dữ liệu này được viết thành 8 khối, mỗi khối 6-bit, bằng cách chọn các bit đầu vào của nó theo một thứ tự được qui định trên bảng. Các hàm được chọn duy nhất đó được gọi là các S – box, $\mathbf{S_1}, \mathbf{S_2}, \ldots, \mathbf{S_8}$, nhận đầu vào là các khối 6-bit r_i và cho đầu ra là các khối 4-bit $S_i(r_i)$. Hàm hoán vị $\boldsymbol{P}$ cho đầu ra 32-bit R'''' từ đầu vào 32-bit R''' bằng cách hoán vị các bit của khối đầu vào. R'''' là đầu ra của hàm ***f***.

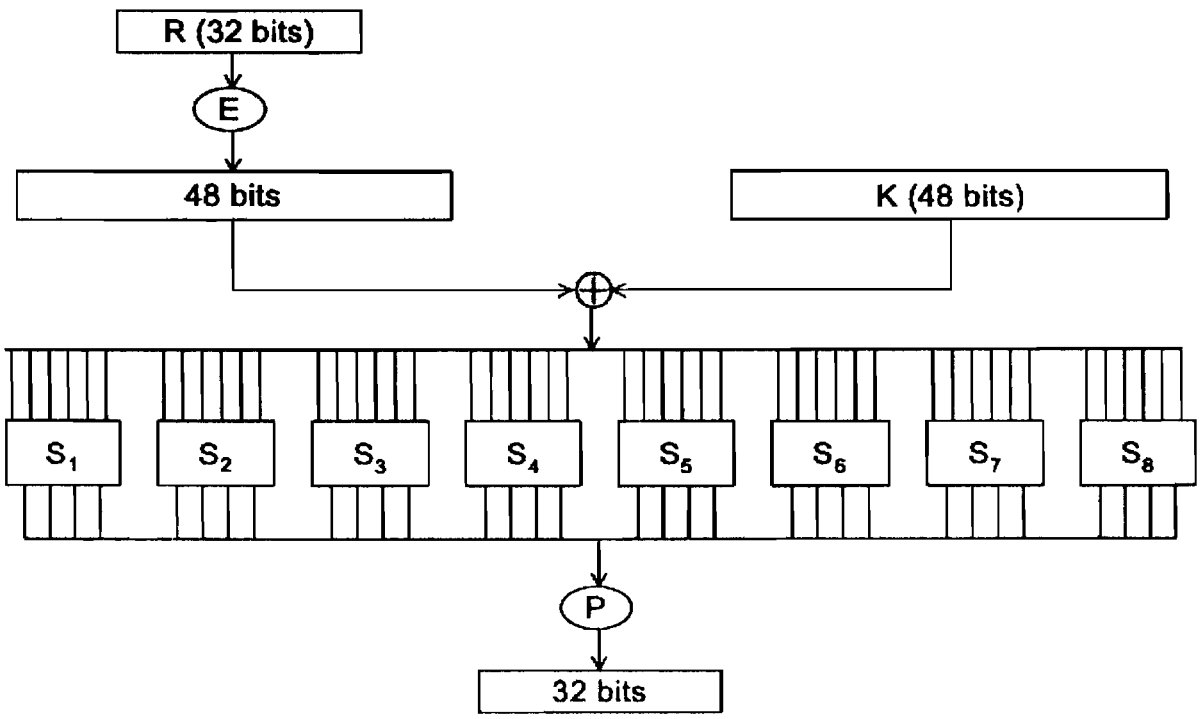
Hàm ***f*** do đó được xác định như sau (P và C là dữ liệu, K_i là khóa)

(1) $R' = E(R)$

(2) $R'' = R' \oplus K$

(3) $R'' = r1 \parallel r2 \parallel r3 \parallel r4 \parallel r5 \parallel r6 \parallel r7 \parallel r8$

(4) $R'''' = P(R''')$



Hình A.2 -- Tính $f(R, K)$

A.4.4. Hoán vị mở rộng E

Hoán vị mở rộng E được chỉ ra trên Bảng A.3. Hoán vị E nhận đầu vào 32- bit và cho đầu ra 48-bit. Ba bit đầu tiên của E là các bit ở các vị trí 32, 1 và 2, hai bit cuối ở các vị trí 32 và 1.

32	1	2	3	4	5	64	32
4	5	6	7	8	9	63	31
8	9	10	11	12	13	62	30
12	13	14	15	16	17	61	29
16	17	18	19	20	21	60	28
20	21	22	23	24	25	59	27
24	25	26	27	28	29	58	26
28	29	30	31	32	1	57	25

Bảng A.3 – Hoán vị mở rộng E

A.4.5. Hoán vị P

Hoán vị P được chỉ ra trên Bảng A.4. Hoán vị P nhận đầu vào 32-bit và cho đầu ra 32-bit. Đầu ra $P(L)$ của hàm P được xác định bởi Bảng A.4, thu được từ đầu vào L bằng cách lấy bit thứ 16 của L làm bit thứ nhất của $P(L)$, bit thứ bảy làm bit thứ hai của $P(L)$,v.v và bit thứ 25 của L làm bit thứ 32 của $P(L)$.

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Bảng A.4 – Hoán vị P

A.4.6. S-box

S-box được chỉ ra trên Bảng A.5. Mỗi S-box có 6- bit đầu vào và 4-bit đầu ra.

Nếu S_1 là hàm được xác định trên bảng và B là khối 6-bit, thì $S_1(B)$ được xác định như sau: các bit đầu tiên và cuối cùng của B được biểu diễn theo cơ số 2 là một số nằm trong khoảng từ 0 đến 3. Giả sử số đó là i. Bốn bit giữa của B được biểu diễn theo cơ số 2 là số nằm trong khoảng từ 0 đến 15. Giả sử số đó là j. Trên bảng đó là số nằm trên hàng thứ i và cột thứ j. Đó là số nằm trong khoảng từ 0 đến 15, và được biểu diễn duy nhất bởi khối 4-bit. Khối

này là đầu ra $S_1(B)$ của S_1 ứng với đầu vào ***B.*** Ví dụ, với đầu vào 011011, hàng được biểu diễn bởi 01 là hàng 1, cột được xác định bởi 1101 là cột 13. Nằm trên giao của hàng 1 và cột 13 là số 5, do đó đầu ra là 0101.

S1

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2

15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S3

10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4

7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S5

2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6

12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6

4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S7

4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8

13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Bảng A.5 – Các S-box

A.5. Lược đồ tạo khóa DES

Phần lược đồ khóa DES được chỉ ra trên Hình A.3. Lược đồ này nhận khóa 64-bit khóa **KEY** và cho ra 16 khóa con 48-bit $K_1, K_2, \ldots\ldots, K_{16}$.

Trong đó K_n, với $1 \leq n \leq 16$ là khối 48-bit trong bước (2) của thuật toán. Bởi vậy để mô tả ***KS*** chỉ cần mô tả việc tính toán của K_n từ ***KEY*** với n = 1, 2..., 16. Việc tính toán này được mô tả trên Hình A.3. Do đó để xác định đầy đủ ***KS***, chỉ cần mô tả hai lựa chọn hoán vị, cũng giống như lược đồ chuyển dịch sang trái. Một bit trong byte 8-bit của ***KEY*** có thể được khởi động để phát hiện lỗi trong tạo khóa, phân phối và lưu trữ khóa. Các bit 8, 16,..., 64 được sử dụng để bảo đảm là mỗi byte có tính chất lẻ. Lựa chọn hoán vị 1 được xác định bởi Bảng A.6

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

Bảng A.6 – Hoán vị khóa PC-1

Bảng trên được chia thành hai phần, phần thứ nhất xác định cách chọn các bit trong C_0, phần thứ hai xác định cách chọn các bit trong D_0. Các bit của ***KEY*** được đánh số từ 1 đến 64. Các bit trong C_0 tương ứng là 57, 49, 41,..., 44 và 36 của ***KEY***, còn các bit trong D_0 là các bit 63, 55, 47,..,12 và 4 của ***KEY***. Với C_0 và D_0 đã xác định, có thể xác định được C_n và D_n từ C_{n-1} và D_{n-1}, tương ứng với n = 1, 2,..., 16. Điều này đạt được bằng cách tuân thủ vào lược đồ phép dịch sang trái các khối riêng lẻ sau:

Số lần lặp	Số lần dịch trái
1	1

2	1
3	2
4	2
5	2
6	2
7	2
8	2
Số lần lặp	Số lần dịch trái
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

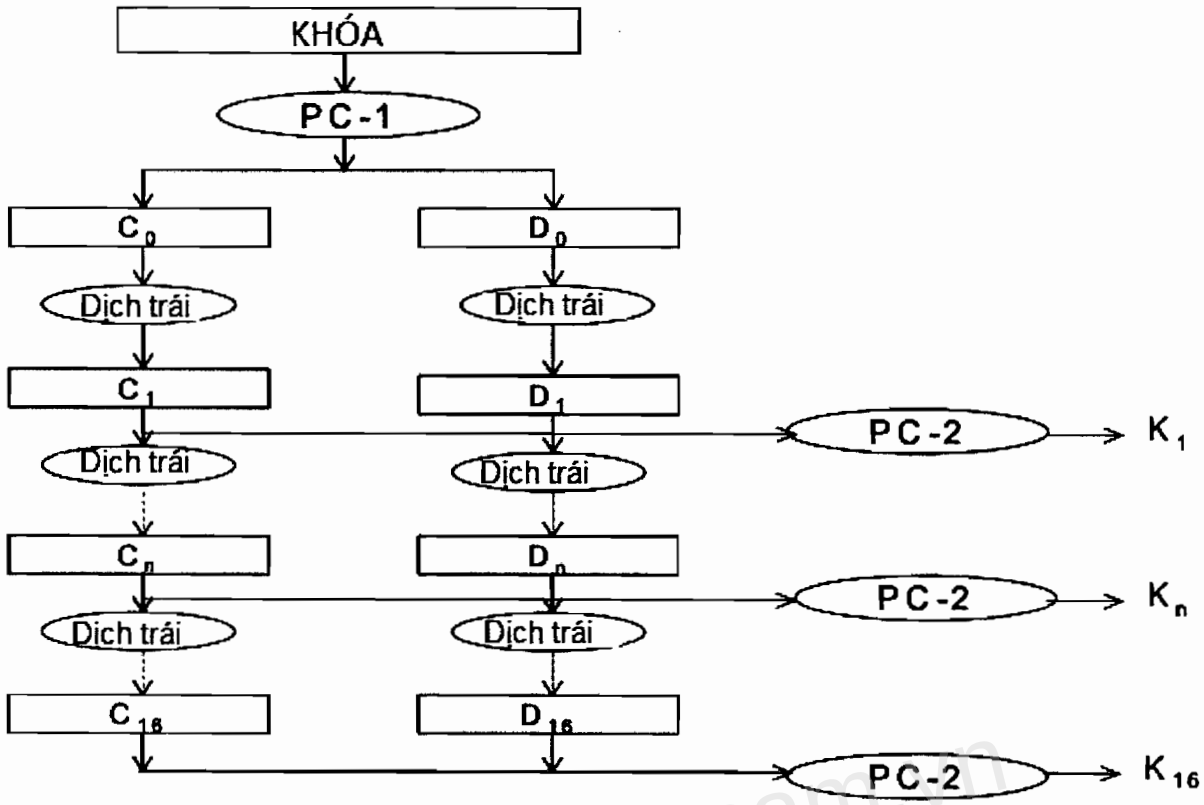
Bảng A.7 – Số các bit khóa được dịch chuyển của mỗi vòng

Ví dụ, C_3 và D_3 thu được từ C_2 và D_2 tương ứng bằng hai dịch chuyển sang trái, và C_{16} và D_{16} thu được từ C_{15} và D_{15} tương ứng bằng một lần dịch sang trái. Trong tất cả các trường hợp, việc dịch một lần sang trái được hiểu là dịch các bit sang trái một vị trí sao cho sau một lần dịch các bit ở 28 vị trí là các bit trước đó ở vị trí 2, 3,..., 28, 1. Việc chọn hoán vị 2 được xác định trên Bảng A.8

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	10	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Bảng A.8 – Hoán vị nén PC-2

Bởi vậy bit thứ nhất của K_n là bit thứ 14 của C_nD_n, bit thứ hai là bit thứ 17,v.v, bit thứ 47 là bit thứ 29, bit thứ 48 là bit thứ 32.



Hình A.3 – Tính toán lược đồ khóa

Phụ lục B

(Quy định)

Các phép biến đổi của AES

B.1. Các phép biến đổi AES

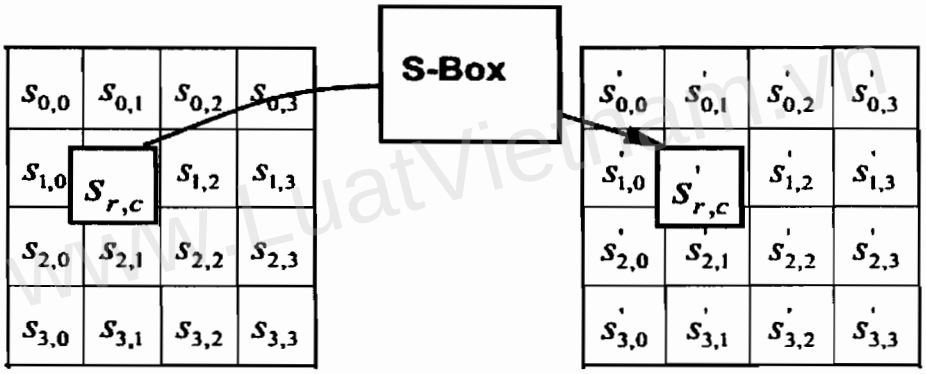
B.1.1. Các phép biến đổi xác định cho AES

Thuật toán AES sử dụng các phép biến đổi $SubBytes()$, $SubBytes^{-1}()$, $ShiftRows()$, $ShiftRows^{-1}()$, $MixColumns()$, $MixColumns^{-1}()$, $AddRoundKey()$, được mô tả dưới đây.

B.1.2. Phép biến đổi $SubBytes()$

Phép biến đổi $SubBytes()$ thực hiện thay thế mỗi byte Trạng thái $s_{i,j}$ bởi giá trị mới $s'_{i,j}$, bằng cách sử dụng bảng thay thế (S-box) khả nghịch.

Hình B.1 minh họa tác động của phép biến đổi SubBytes() lên bảng Trạng thái



Hình B.1: $SubBytes()$ áp dụng S-box cho từng byte của Trạng thái.

S-box được sử dụng trong phép biến đổi $SubBytes()$ và được trình bày theo hệ Hexa trên Bảng B.1

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

Bảng B.1: Các S-box của AES

Ví dụ, nếu $s_{1,1}$ = {53} thì giá trị thay thế là giá trị nằm trên giao của hàng có chỉ số '5' và cột có chỉ số '3' của Bảng 5, điều này cho kết quả là $s'_{1,1}$ có giá trị {ed}.

B.1.3. Phép biến đổi $SubBytes^{-1}()$

$SubBytes^{-1}()$ là phép biến đổi nghịch đảo của phép biến đổi $SubBytes()$, trong đó S-box nghịch đảo được áp dụng cho từng byte của Trạng thái. Điều này đạt được bằng cách áp dụng phép biến đổi nghịch đảo được mô tả tại Điều 5.2.4.2

S-box nghịch đảo được sử dụng trong phép biến đổi $SubBytes^{-1}()$ được mô tả trong Bảng B.2.

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

Bảng B.2: S-box nghịch đảo AES

B.1.4. Phép biến đổi $ShiftRows()$

Trong phép biến đổi $ShiftRows()$, các byte ở ba dòng cuối của Trạng thái được dịch vòng lên một số lượng bytes khác nhau (offsets). Hàng thứ nhất, hàng 0 được giữ nguyên (không dịch chuyển).

Cụ thể, phép biến đổi $ShiftRows()$ được thực hiện như sau:

$$S'_{r,c} = S_{r,(c+r)mod4} \quad \text{với } 0 < r < 4, \text{ và } 0 \leq c < 4, \text{ ở đây } r \text{ là số thứ tự của hàng.}$$

Theo đó các byte dịch sẽ chuyển sang trái (nghĩa là các giá trị thấp hơn của c trong một hàng cho trước), trong khi các byte phía trái ngoài cùng dịch vòng sang các vị trí phía phải ngoài cùng của hàng (tức là những giá trị cao hơn của c trong hàng cho trước).

Hình B.2 mô tả phép biến đổi $ShiftRows()$, trong đó các byte được dịch vòng sang trái.

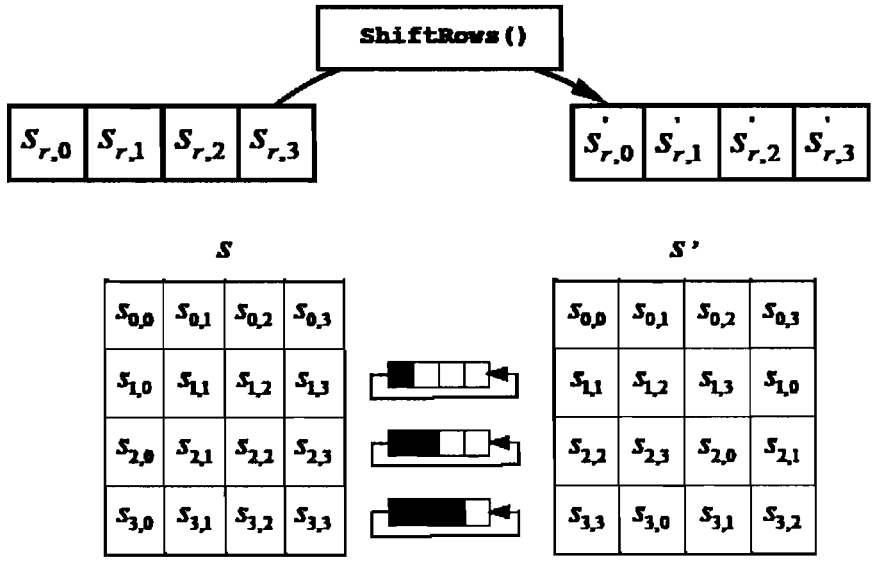
B.1.5. Phép biến đổi $ShiftRows^{-1}()$

$ShiftRows^{-1}()$ là phép nghịch đảo của phép biến đổi $ShiftRows()$. Các bytes trong ba dòng cuối của Trạng thái được dịch vòng lên một số lượng bytes khác nhau. Dòng thứ nhất, dòng 0 không dịch chuyển. Ba dòng dưới cùng được dịch vòng lên $4 - r$ bytes, ở đây r là số thứ tự vòng.

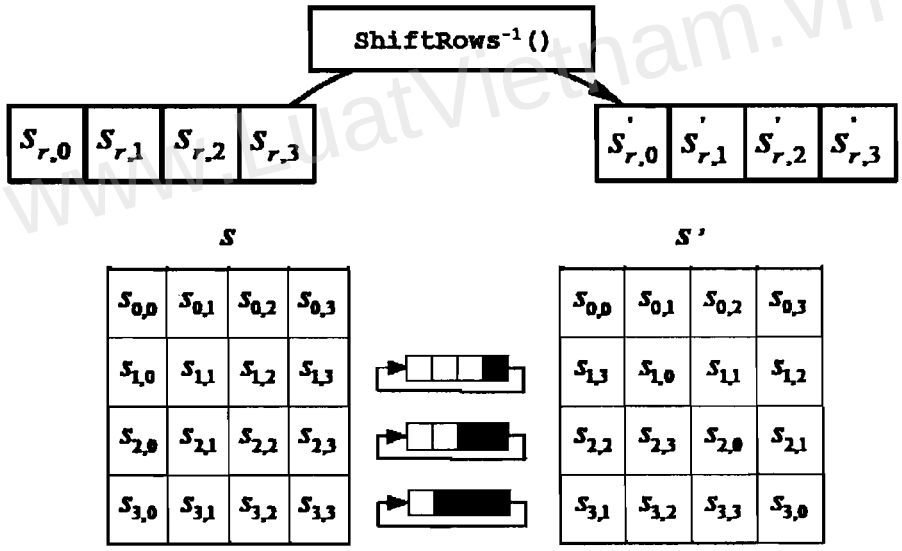
Cụ thể, phép biến đổi $ShiftRows^{-1}()$ được thực hiện như sau:

$S_{r,(c+r)mod4} = S'_{r,c}$ với $0 < r < 4$, và $0 \leq c < 4$.

Hình B.3. mô tả phép biến đổi $ShiftRows^{-1}()$



Hình B.2: $ShiftRows()$ dịch vòng ba dòng cuối của Trạng thái



Hình B.3: $ShiftRows^{-1}()$ dịch vòng ba dòng cuối của Trạng thái.

B.1.6. Phép biến đổi $MixColumns()$

Phép biến đổi $MixColumns()$ thao tác trên Trạng thái, thay mỗi cột bằng cột khác. Các cột của Trạng thái được xem như những đa thức trên trường $GF(2^8)$ và được nhân modulo $x^4 + 1$ với đa thức cố định $a(x)$ cho trước, $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$. Phép nhân này có thể viết dưới dạng phép nhân ma trận:

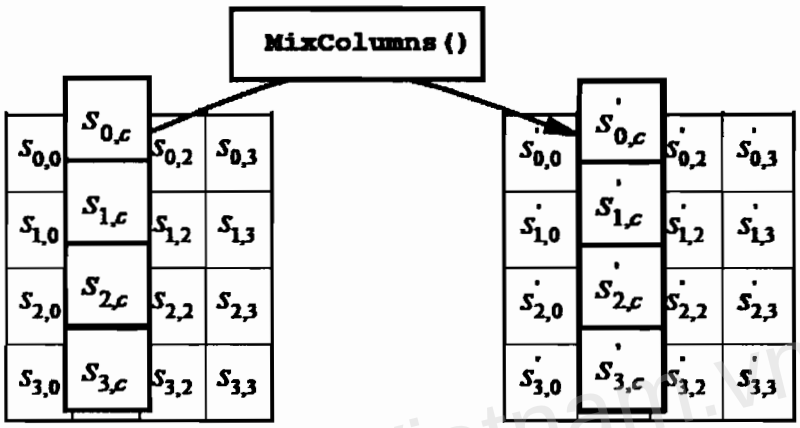
$$s'(x) = a(x) \otimes s(x): \quad \begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{với } 0 \leq c < 4.$$

Kết quả của phép nhân trên là bốn byte trong cột được thay thế như sau:

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$
$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$
$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})$$

Toán tử $\oplus$ trong các biểu thức trên biểu thị phép cộng trong trường $GF(2^8)$, với phép cộng bit XOR. Phép nhân được thực hiện theo modulo của đa thức bất khả qui của trường. Trong trường hợp thuật toán AES đó là đa thức $x^8 + x^4 + x^3 + x + 1$.

Hình B.4 mô tả phép biến đổi $MixColumns()$.



Hình B.4 – $MixColumns$ () thao tác trên Trạng thái, thay cột bằng cột khác

B.1.7. Phép biến đổi $MixColumns^{-1}()$

$MixColumns^{-1}()$ là phép biến đổi nghịch đảo của phép biến đổi $MixColumns()$. $MixColumns^{-1}()$ thao tác trên Trạng thái, thay mỗi cột bằng cột khác. Phép biến đổi này có thể biểu diễn dưới dạng phép nhân ma trận, ở đấy mỗi byte được coi như một phần tử của trường hữu hạn $GF(2^8)$:

$$s'(x) = a^{-1}(x) \otimes s(x) \qquad \begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \qquad \text{với } 0 \le c < 4$$

Kết quả của phép nhân là bốn byte trong mỗi cột được thay thế như sau:

$$s'_{0,c} = (\{0e\} \bullet s_{0,c}) \oplus (\{0b\} \bullet s_{1,c}) \oplus (\{0d\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$
$$s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{0b\} \bullet s_{2,c}) \oplus (\{0d\} \bullet s_{3,c})$$
$$s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{0e\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c})$$

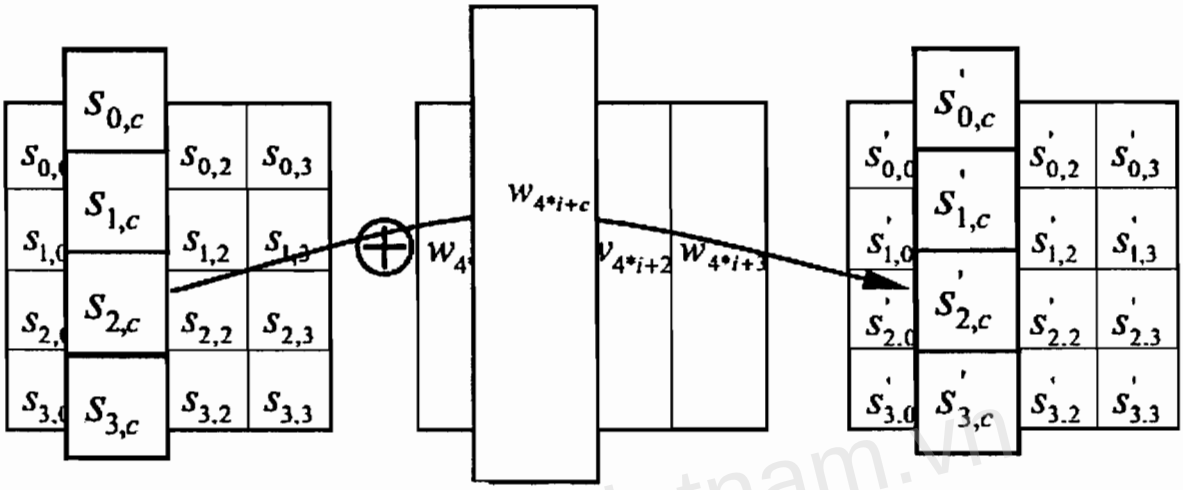
B.1.8. Phép biến đổi $AddRoundKey()$

Trong phép biến đổi $AddRoundKey()$, khóa vòng được cộng vào Trạng thái bằng phép cộng bit đơn giản XOR. Mỗi khóa vòng gồm bốn từ (128 bit) lấy từ lược đồ khóa (được mô tả tại 5.2.5). Bốn từ này được cộng vào cột Trạng thái như sau:

$$[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{(4*i+c)}] \quad \text{với } 0 \le c < 4$$

Ở đây, $0 \le c < 4$ và $w_{(4*i+c)}$ là các từ của lược đồ khóa thứ c của khóa vòng thứ i $W_i = [w_{(4*i)}, w_{(4*i+1)}, w_{(4*i+2)}, w_{(4*i+3)}]$ và i là giá trị thuộc khoảng $0 \le i \le Nr$. Trong phép mã hóa, phép cộng khóa vòng ban đầu xảy ra khi $i = 0$, trước ứng dụng thứ nhất của hàm vòng. Việc áp dụng phép biến đổi $AddRoundKey()$ cho **Nr** vòng của phép mã hóa xảy ra khi $1 \le i \le Nr$.

Hoạt động của phép biến đổi $AddRoundKey()$, được minh họa trên Hình B.5. Địa chỉ byte trong các từ của lược đồ khóa được mô tả trong Điều 2.2.2.2.



Hình B.5: $AddRoundKey()$ cộng bit XOR từng cột của Trạng thái với mỗi từ lấy từ lược đồ khóa

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

QCVN 5 : 2016/BQP

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ CHỮ KÝ SỐ SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on digital signature used in banking

HÀ NỘI - 2016

Mục lục

Lời nói đầu

QCVN 5 : 2016/BQP do Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã – Ban Cơ yếu Chính phủ biên soạn, Ban Cơ yếu Chính phủ trình duyệt, Bộ Khoa học và Công nghệ thẩm định và được ban hành theo Thông tư số 161/2016/TT-BQP ngày 21 tháng 10 năm 2016 của Bộ trưởng Bộ Quốc phòng.

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ CHỮ KÝ SỐ SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on digital signature used in banking

1. QUY ĐỊNH CHUNG

1.1. Phạm vi điều chỉnh

Quy chuẩn kỹ thuật quốc gia này quy định mức giới hạn của các đặc tính kỹ thuật mật mã của chữ ký số sử dụng trong lĩnh vực ngân hàng.

1.2. Đối tượng áp dụng

Quy chuẩn này áp dụng đối với các doanh nghiệp kinh doanh sản phẩm, dịch vụ mật mã dân sự trong lĩnh vực ngân hàng; các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự.

1.3. Tài liệu viện dẫn

- *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST SP 800-90A Rev. 1, National Institute of Standards and Technology, June 2015. (*Khuyến cáo cho bộ sinh số ngẫu nhiên sử dụng bộ sinh bit ngẫu nhiên tất định*, NIST SP 800-90A Rev. 1, Viện tiêu chuẩn và công nghệ quốc gia (Mỹ), tháng 6 năm 2015).
- RSA Laboratories. *PKCS#1 v2.1: RSA Cryptography Standard*. June 2002. (Phòng thí nghiệm RSA. *PKCS#1 v2.1: Tiêu chuẩn mật mã RSA*. Tháng 6 năm 2002).
- TCVN 7635:2007 Kỹ thuật mật mã – Chữ ký số.
- NIST, Federal Information Processing Standards Publication 186-4, *Digital Signature Standard (DSS)*, July 2013. (NIST, Tiêu chuẩn xử lý thông tin liên bang, *Chuẩn chữ ký số (DSS)*, tháng 7 năm 2013).
- NIST, Federal Information Processing Standards Publication 180-4, *Secure Hash Standard (SHS)*, August 2015. (NIST, Tiêu chuẩn xử lý thông tin liên bang, Chuẩn *hàm băm an toàn (SHS)*, tháng 8 năm 2015).

1.4. Giải thích từ ngữ

Trong Quy chuẩn này, các từ ngữ dưới đây được hiểu như sau:

1.4.1.

Thông tin không thuộc phạm vi bí mật nhà nước

Là thông tin không thuộc nội dung tin "tuyệt mật", "tối mật" và "mật" được quy định tại Pháp lệnh Bảo vệ bí mật nhà nước ngày 28 tháng 12 năm 2000.

1.4.2.

Mật mã

Là những quy tắc, quy ước riêng dùng để thay đổi hình thức biểu hiện thông tin nhằm bảo đảm bí mật, xác thực, toàn vẹn của nội dung thông tin.

1.4.3.

Mật mã dân sự

Là kỹ thuật mật mã và sản phẩm mật mã được sử dụng để bảo mật hoặc xác thực đối với thông tin không thuộc phạm vi bí mật nhà nước.

1.4.4.

Sản phẩm mật mã dân sự

Là các tài liệu, trang thiết bị kỹ thuật và nghiệp vụ mật mã để bảo vệ thông tin không thuộc phạm vi bí mật nhà nước.

1.4.5.

Kỹ thuật mật mã

Là phương pháp, phương tiện có ứng dụng mật mã để bảo vệ thông tin.

1.4.6.

Mã hóa

Phép biến đổi (khả nghịch) dữ liệu bởi thuật toán mật mã để tạo ra bản mã, tức là che giấu nội dung thông tin của dữ liệu.

1.4.7.

Giải mã

Phép toán ngược với phép mã hóa tương ứng.

1.4.8.

Mật mã phi đối xứng

Hệ thống dựa trên kỹ thuật mật mã phi đối xứng, trong đó phép biến đổi công khai được sử dụng để mã hóa, phép biến đổi bí mật được sử dụng để giải mã.

1.4.9.

Kỹ thuật mật mã phi đối xứng

Kỹ thuật mật mã phi đối xứng sử dụng hai phép biến đổi liên quan đến nhau, phép biến đổi công khai (được xác định bởi khóa công khai) và phép biến đổi bí mật (được xác định bởi khóa riêng). Cả hai phép biến đổi này có tính chất là cho biết phép biến đổi công khai, về mặt tính toán không thể có khả năng xác định được phép biến đổi bí mật.

1.4.10.

Chữ ký số

Một chuỗi số, kết quả của phép biến đổi mật mã trên thông điệp dữ liệu nhằm cung cấp một phương tiện để kiểm tra tính xác thực của nguồn gốc thông điệp dữ liệu, tính toàn vẹn của dữ liệu và tính không thể chối bỏ của người đã ký.

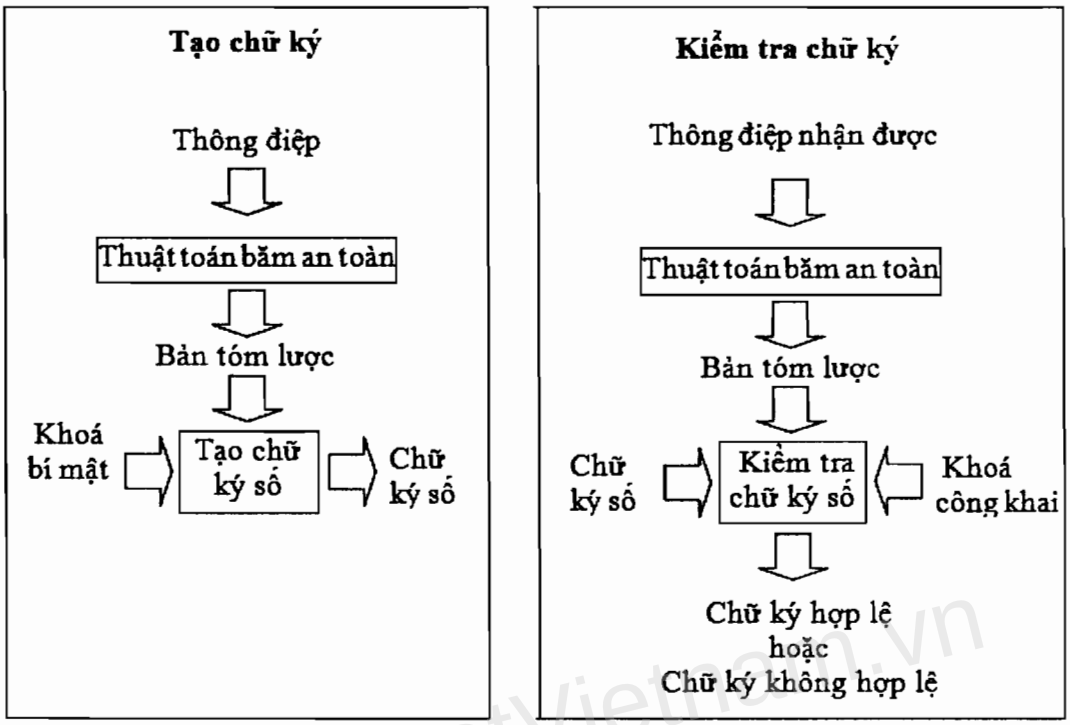
1.5. Các ký hiệu

AES	Tiêu chuẩn mã hóa tiên tiến (*Advanced Encryption Standard*)
$Octet$	Chuỗi bit có độ dài bằng 8
n	Modulo RSA
d	Số mũ bí mật RSA
e	Số mũ công khai RSA
(n, e)	Khóa công khai RSA của người ký
c	Một biểu diễn của bản mã, là số nguyên thuộc $(0, n-1)$
C	Bản mã được biểu diễn ở dạng chuỗi octet
EM	Chuỗi Octet biểu diễn thông điệp đã được ghi mã
$emLen$	Độ dài theo Octet của EM
k	Độ dài modulo n tính theo octet
m	Một biểu diễn của thông điệp (văn bản), là số nguyên thuộc $(0, n-1)$
M	Thông điệp (văn bản), chuỗi octet
$I2OSP$	Hàm cơ sở chuyển đổi từ dạng số nguyên sang chuỗi octet (*Integer-to-Octet-String Primitive*)
$OS2IP$	Hàm cơ sở chuyển đổi từ chuỗi octet sang số nguyên (*Octet-String-to-Integer-Primitive*)
LCM	Bội chung nhỏ nhất (*Least Common Multiplier*)
$nlen$	Độ dài modulo n theo bit
$security\_strength$	độ mạnh về an toàn (*security_strength*) là một số nguyên biểu thị lượng tính toán cần thiết để phá hệ mã
N	là ký hiệu độ dài theo bit của số nguyên tố q
L	là ký hiệu độ dài theo bit của số nguyên tố p
$\|\|$	Toán tử nối hai chuỗi
$PKCS$	Tiêu chuẩn mật mã khoá công khai (*Puplic Key Cryptography Standard*) do Phòng thí nghiệm RSA (Mỹ) ban hành.

PSS	Lược đồ ký xác suất (*Probabilistic Signature Scheme*)
RSA	Tên của hệ mã do ba nhà toán học Rivest, Shamir và Adleman sáng tạo ra
$RSAVP$	Phép toán cơ sở phục vụ cho kiểm tra chữ ký RSA
$RSASP$	Phép toán ký RSA cơ sở
$RSASSA$	Lược đồ ký RSA kèm phụ lục (*RSA Signature Scheme with Appendix*)
SHA	Thuật toán băm an toàn (*Secure Hash Algorithm*).
$Word$	Từ (32 bit)
DSA	Thuật toán chữ ký số
EC	Đường cong Elliptic
$ECDSA$	Thuật toán chữ ký số dựa trên đường cong Elliptic

2. QUY ĐỊNH KỸ THUẬT

2.1. Chữ ký số



Hình 1 - Mô tả quá trình tạo và kiểm tra chữ ký số

Chữ ký số là một dạng chữ ký điện tử được tạo ra bằng sự biến đổi một thông điệp dữ liệu sử dụng hệ thống mật mã phi đối xứng theo đó người có được thông điệp dữ liệu ban đầu và khoá công khai của người ký có thể xác định được chính xác:

- Việc biến đổi nêu trên được tạo ra bằng đúng khoá bí mật tương ứng với khoá công khai trong cùng một cặp khóa;
- Sự toàn vẹn nội dung của thông điệp dữ liệu kể từ khi thực hiện việc biến đổi nêu trên.

Quá trình sinh chữ ký số trên một thông điệp dữ liệu yêu cầu sử dụng: 1) hàm băm mật mã thực hiện tính toán trên dữ liệu sẽ được ký, 2) sử dụng khóa mật mã và thuật toán ký để tạo chữ ký số trên đầu ra của hàm băm. Quy chuẩn này quy định sử dụng khóa mật mã trong thuật toán ký số, hàm băm mật mã được sử dụng trong quá trình sinh chữ ký số.

2.1.1. Quy định kỹ thuật

2.1.1.1. Quy định chung:

Quá trình chữ ký số	Sử dụng	Ghi chú
Tạo chữ ký số	Độ an toàn $\geq$ 112 bit: DSA: $\|p\| \geq 2048$ và $\|q\| \geq 224$ RSA: $\|n\| \geq 2048$ EC: $\|n\| \geq 224$	

Xác thực chữ ký số	Độ an toàn ≥ 112 bit: DSA: $\|p\| \geq 2048$ và $\|q\| \geq 224$ RSA: $\|n\| \geq 2048$ EC: $\|n\| \geq 224$	

2.1.1.2. Quy định chi tiết về nguồn ngẫu nhiên:

Các số ngẫu nhiên được sử dụng cho các mục đích khác nhau như để sinh các tham số mật mã, các khóa mật mã, các giá trị ngẫu nhiên dùng một lần và các giá trị thách đố xác thực.

Một số bộ sinh bit ngẫu nhiên tất định DRBG được chấp thuận để sử dụng theo quy định chung bao gồm: HASH_DRBG, HMAC_DRBG và CTR_DRBG.

Các bộ sinh bit ngẫu nhiên RBG tuân theo SP800-90A phiên bản sửa đổi lại năm 2015 để sinh bit ngẫu nhiên cũng được chấp thuận để sử dụng tiếp.

2.1.2. Chữ ký số RSA-PSS

2.1.2.1. Các yêu cầu chung

1. Cặp khoá RSA dùng để ký thì không được dùng cho mục đích khác (chẳng hạn dùng lại để mã thông điệp);
2. Hai số nguyên tố p, q và số mũ bí mật d cần phải được giữ bí mật tránh việc bị truy cập bất hợp pháp, làm lộ hoặc sửa đổi. Modulo n và số mũ công khai e phải được công bố công khai;
3. Mỗi người sử dụng cần có Modulo n riêng;
4. Độ dài của Modulo n $(nlen)$ không được nhỏ hơn 2048 bit và nên được thay đổi theo thời gian như sau.

Năm (y)	độ an toàn	*nlen* tối thiểu
y<2020	112	2048
	128	3072

Trong đó, độ mạnh về an toàn $(security\_strength)$ là một số nguyên biểu thị lượng tính toán cần thiết để phá hệ mã.

Vì các phương pháp phá hệ mã thường xuyên được hoàn thiện nên cần phải định kỳ 3 đến 5 năm một lần xem xét lại $nlen$ tối thiểu (có thể tham khảo chi tiết yêu cầu này trong tài liệu *NIST Special Publication 800-57: Recommendation for Key Management – Part1: General, January 2016*).

- Phiên bản áp dụng: Áp dụng phiên bản 2.1 của tiêu chuẩn RSA Cryptography Standard PKCS #1 v2.1.

- Áp dụng lược đồ RSAES-OAEP để mã hoá và RSASSA-PSS để ký.

2.1.2.2. Quy định chi tiết về các khóa RSA

1. Số mũ công khai e cần phải được chọn với các ràng buộc sau:

a) Số mũ công khai e cần được chọn trước khi tạo số mũ bí mật d;

b) Số mũ công khai e cần phải là số nguyên dương lẻ sao cho

$$65{,}537 \leq e < 2^{nlen-2security\_strength}$$

Với $nlen$ là độ dài của modulo n theo bit.

Chú ý rằng e có thể là giá trị bất kỳ mà thoả mãn ràng buộc 1(b); p và q sẽ được chọn (trong mục 2) sao cho e là nguyên tố cùng nhau với cả $(p-1)$ và $(q-1)$.

2. Hai số nguyên tố p và q được tạo ngẫu nhiên và giữ bí mật cần phải được chọn với các ràng buộc sau:

a) $(p-1)$ và $(q-1)$ cần phải là nguyên tố cùng nhau với số mũ công khai e;

b) Mỗi một trong bốn số $(p+1)$, $(p-1)$ và $(q+1)$, $(q-1)$ cần phải có các nhân tử nguyên tố lớn hơn $2^{security\_strength+20}$;

c) Nhân tử nguyên tố bí mật p, q cần phải được chọn ngẫu nhiên từ các số nguyên tố thoả mãn $(\sqrt{2})(2^{(nlen/2)-1}) \leq q < p \leq (2^{(nlen/2)}-1)$;

d) $|p-q| > 2^{(nlen/2-100)}$.

3. Số mũ bí mật d cần phải được lựa chọn sau khi tạo p và q với các ràng buộc:

a) Số mũ d cần phải lớn hơn $2^{(nlen/2)}$, và

b) $d = e^{-1} mod(LCM((p-1),\ (q-1)))$

(Chi tiết về hàm tạo các tham số RSA có thể tham khảo trong tài liệu *FIPS 186-4: Digital Signature Standard*).

2.1.2.3. Tạo chữ ký số

$RSASSA-PSS-SIGN(K, M)$

Đầu vào:	K	khoá bí mật RSA của người ký
	M	thông điệp sẽ được ký, là một chuỗi octet
Đầu ra:	S	chữ ký, chuỗi octet có độ dài k, với k là độ dài của modulo RSA theo octet
Thông báo lỗi:		"văn bản quá dài", "lỗi định dạng"

Các bước:

1. Mã hoá $EMSA-PSS$: Áp dụng thao tác $EMSA-PSS-ENCODE$ vào văn bản M để tạo ra thông điệp được định dạng EM có độ dài $\lceil (modBits-1)/8 \rceil$ octet sao cho độ dài bit của số nguyên $OS2IP(EM)$ nhiều nhất là $modBits-1$, với $modBits$ là độ dài theo bit của số n (modun RSA):

$$EM = EMSA-PSS-ENCODE\ (M, modBits-1).$$

Chú ý rằng độ dài octet của EM sẽ bằng k - 1 nếu $modBits-1$ chia hết cho 8 và bằng k nếu $modBits-1$ không chia hết cho 8. Nếu hàm $EMSA-PSS-ENCODE$ cho ra

thông báo lỗi "văn bản quá dài" thì $RSASSA - PSS - SIGN$ cũng cho ra thông báo lỗi "văn bản quá dài" và dừng lại. Nếu $EMSA - PSS - ENCODE$ cho ra thông báo "lỗi định dạng" thì $RSASSA - PSS - SIGN$ cũng cho ra thông báo "lỗi định dạng" và dừng lại.

2. Chữ ký *RSA*:

a. Chuyển thông điệp đã được định dạng (chuỗi octet) EM thành biểu diễn thông điệp ở dạng số nguyên m.

$$m = OS2IP(EM).$$

b. Áp dụng phép toán cơ sở $RSASP$ với K là khoá bí mật RSA và biểu diễn thông điệp m để tạo ra biểu diễn chữ ký là số nguyên s:

$$s = RSASP(K, m).$$

c. Chuyển chữ ký s dạng số nguyên thành chữ ký S dạng chuỗi octet có độ dài k:

$$S = I2OSP(s, k).$$

Đưa ra chữ ký S.

2.1.2.4. Xác thực chữ ký số

$RSASSA - PSS - VERIFY((n, e), M, S)$

Đầu vào:	(n, e)	khoá công khai RSA của người ký
	M	thông điệp mà chữ ký của nó cần được kiểm tra, là chuỗi octet
	S	chữ ký được kiểm tra, chuỗi octet có độ dài k, với k là độ dài theo octet của số n, modulo RSA
Đầu ra:		"chữ ký hợp lệ" hoặc "chữ ký không hợp lệ"

Các bước:

1. Kiểm tra độ dài: Nếu độ dài của chữ ký S không là k octet, cho ra thông báo lỗi "chữ ký không hợp lệ" và dừng lại;
2. Kiểm tra chữ ký RSA;

a. Chuyển chữ ký S thành biểu diễn chữ ký ở dạng số nguyên s;

$$s = OS2IP(S)$$

b. Áp dụng phép toán cơ sở RSAVP với khoá công khai RSA là (n, e) và biểu diễn chữ ký s để tạo ra m là số nguyên biểu diễn thông điệp;

$$m = RSAVP((n, e), s)$$

c. Chuyển biểu diễn thông điệp m thành thông điệp đã được định dạng EM có độ dài $emLen = \lceil (modBits - 1)/8 \rceil$ octet, với $modBits$ là độ dài theo bit của số n (Modulo RSA):

$$EM = I2OSP\ (m, emLen)$$

Chú ý rằng $emLen$ sẽ bằng $k-1$ nếu $modBits-1$ chia hết cho 8 và bằng k nếu $modBits-1$ không chia hết cho 8. Nếu $I2OSP$ cho ra thông báo lỗi "số nguyên quá lớn" thì $RSASSA-PSS-VERIFY$ cho ra thông báo lỗi "chữ ký không hợp lệ" và dừng lại.

3. *Kiểm tra* $EMSA-PSS$: Áp dụng thao tác kiểm tra $EMSA-PSS-VERIFY$ vào thông điệp M và thông điệp đã được định dạng EM để xác định xem chúng có tương ứng với nhau hay không;

$$Result = EMSA-PSS-VERIFY\ (M, EM, modBits-1).$$

4. Nếu kết quả (Result) là "phù hợp" thì cho ra "chữ ký hợp lệ". Ngược lại sẽ cho ra "chữ ký không hợp lệ".

2.1.3. Chữ ký số ECDSA

2.1.3.1. Quy định chi tiết về các khóa ECDSA:

Kiểm tra tính hợp lệ của các tham số miền $(p, SEED, a, b, G, n, h)$ như sau:

Xâu $SEED$ dùng để sinh ngẫu nhiên đường cong Elliptic xác định trên trường F_p với p là số nguyên tố lẻ.

Trước khi sử dụng một bộ tham số miền, tính hợp lệ của nó phải được kiểm tra theo thuật toán dưới đây:

1. Kiểm tra p là một số nguyên tố lẻ.
2. Kiểm tra a, b, x_G, y_G là các phần tử của trường F_p.
3. Kiểm tra rằng a và b được dẫn xuất tương ứng từ $SEED$.
4. Kiểm tra $(4a^3 + 27b^2)$ khác 0 và $j(E) \neq 0; 1728$ trong F_p
5. Kiểm tra $y_G^2 = x_G^3 + ax_G + b$ trong F_p.
6. Kiểm tra n là nguyên tố và $n > 4\sqrt{p}$.
7. Kiểm tra $nG = O_E$.
8. Kiểm tra đường cong có thuộc danh sách các đường cong yếu:
   a. Thoả mãn điều kiện MOV, (chú ý rằng một đường cong thoả mãn điều kiện MOV sẽ không phải là đường cong siêu biến).
   b. Kiểm tra đường cong không kì dị, nghĩa là $\#E \neq p$.

Nếu bất kỳ sự kiểm tra nào ở trên thất bại thì tham số miền phải được xem là không hợp lệ.

Điều kiện MOV được hiểu là không có giá trị k nguyên dương nào $0 < k < B$ để cho $p^k - 1$ chia hết cho n. Trên thực hành hiện nay $|p| = 224$ bit thì người ta xét với $B = 15$ là đủ vì khi đó $|p^k| = 3360 > 2048$.

Các hệ số a, b của đường cong được sinh ngẫu nhiên trên F_p từ đầu vào $SEED$ và có thể kiểm tra được.

Khóa bí mật d phải được sinh ngẫu nhiên trong khoảng $[1, n-1]$.

Đường cong Elliptic xác định trên trường hữu hạn F_p với tối thiểu $|p| = 224$ bit và được xác định cụ thể như sau:

Độ dài bit của n	Độ dài bit của p
224 - 255	|p| = 224
256 - 383	|p| = 256
384 - 511	|p| = 384
≥ 512	|p| = 521

Đại lượng Cofactor được định nghĩa và ký hiệu là $h = \#E(F_p)/n$ tuân theo bảng dưới đây:

Độ dài bit của n	Giá trị h cực đại cho phép
224 - 255	2^{14}
256 - 383	2^{16}
384 - 511	2^{24}
≥ 512	2^{32}

2.1.3.2. Tạo chữ ký số

Thuật toán Chữ ký số Đường cong Elliptic (ECDSA) thực hiện việc sinh chữ ký của thông báo m, làm việc như sau:

Sinh chữ ký $ECDSA$

Đầu vào: Các tham số miền (E, P), khoá bí mật d, thông báo m.

Đầu ra : Chữ ký (r, s).

1. Lấy $0 < k < q$ một cách ngẫu nhiên
2. $(x_R, y_R) \leftarrow kP$
3. $r \leftarrow x_R \bmod q$
4. Nếu $r = 0$ thì chuyển về bước 1
5. $k \leftarrow k^{-1} \bmod q$
6. $e \leftarrow H(m)$
7. $s \leftarrow k(e + rd) \bmod q$
8. Nếu $s = 0$ thì quay về bước 1

9. Trả về (r, s)

2.1.3.3. Xác thực chữ ký số

Để kiểm tra chữ ký số (r, s) của thông báo m, người kiểm tra tính các bước sau ($Q = dP$ - ký hiệu khoá công khai):

Kiểm tra chữ ký $ECDSA$

Đầu vào: Các tham số miền (E, P), khoá công khai Q, thông báo m, chữ ký (r, s)

Đầu ra : Chấp nhận hoặc bác bỏ chữ ký.

1. Kiểm tra rằng $0 < r, s < q$
2. $s' \leftarrow s^{-1} \bmod q$
3. $e \leftarrow H(m)$
4. $h_1 \leftarrow s'e \bmod q$
5. $h_2 \leftarrow s'r \bmod q$
6. $R = (x_R, y_R) \leftarrow h_1 P + h_2 Q$
7. Nếu $R = 0$ thì bác bỏ chữ ký.
8. Nếu $x_R \bmod q = r$ thì chấp nhận, ngược lại thì bác bỏ.

2.1.4. Chữ ký số DSA

2.1.4.1. Quy định chi tiết về các khóa DSA:

Các tham số riêng bí mật k phải là số được sinh ngẫu nhiên $0 < k < q$ với độ dài $|q|$ không nhỏ hơn 224 bit.

2.1.4.2. Tạo chữ ký số

Chọn N là độ dài bit của $q, min(N, outlen)$ biểu thị số nguyên dương nhỏ nhất N và $outlen$ là độ dài bit của khối đầu ra hàm băm.

Chữ ký của một thông điệp M bao gồm cặp số r và s được tính toán như sau:

$r = (g^k \bmod p) \bmod q.$

$z =$ các bit tận cùng bên trái $min(N, outlen)$ của $Hash(M)$.

$s = (k^{-1}(z + xr)) \bmod q.$

Khi tính s, xâu z thu được từ hàm $Hash(M)$ sẽ được biến đổi sang một số nguyên.

Chú ý r có thể được tính bất cứ khi nào nếu biết k, p, q và g. Ví dụ: bất cứ khi nào các tham số miền p, q và g được biết, và k được tính toán trước thì r cũng có thể được tính trước vì thông tin của thông báo được ký không yêu cầu các tính toán của r. Việc tính trước giá trị k, k^{-1} và r sẽ được bảo vệ một cách tương tự như khóa riêng x tới khi tính toán xong s.

Giá trị của r và s sẽ được kiểm tra để xác định xem $r = 0$ hay $s = 0$. Nếu một trong hai $r = 0$ hoặc $s = 0$ thì một giá trị mới của k sẽ được sinh ra và chữ ký sẽ được tính toán lại. Nếu chữ ký số được sinh ra đúng đắn, thì rất hiếm khi xảy ra $r = 0$ hoặc $s = 0$.

Chữ ký (r, s) có thể được truyền đi cùng thông điệp để xác thực.

2.1.4.3. Xác thực chữ ký số

Gọi M', r', và s' là các phiên bản tương ứng đã nhận được của M, r và s; gọi y là khóa công khai của người ký; gọi N là độ dài bit của q, và $\boldsymbol{min}(N, outlen)$ biểu thị số nguyên dương nhỏ nhất N và $outlen$, ở đây $outlen$ là độ dài bit của khối đầu ra hàm băm.

Tiến trình xác thực chữ ký được thực hiện như sau:

1. Kiểm tra nếu một trong hai điều kiện $0 < r' < q$ và $0 < s' < q$ bị vi phạm thì chữ ký bị từ chối và được coi là không hợp lệ.
2. Nếu cả hai điều kiện ở bước 1 được thỏa mãn thì:

   $w = (s')^{-1} mod\ q$

   $z =$ Các bit tận cùng bên trái $min(N, outlen)$ của $Hash(M')$.

   $u1 = (zw)\ mod\ q.$

   $u2 = ((r')w)\ mod\ q.$

   $v = (((g)^{u1}\ (y)^{u2})\ mod\ p)\ mod\ q.$
3. Nếu $v = r'$, khi đó chữ ký được xác thực.
4. Nếu v khác r', khi đó thông báo hoặc chữ ký có thể đã bị thay đổi, có thể do lỗi trong tiến trình sinh chữ ký hoặc chữ ký có thể bị giả mạo.

2.1.5. Chữ ký số RSASSA-PKCS1-v1_5

2.1.5.1. Các yêu cầu chung

1. Cặp khoá RSA dùng để ký thì không được dùng cho mục đích khác (chẳng hạn dùng lại để mã thông điệp);
2. Hai số nguyên tố p, q và số mũ bí mật d cần phải được giữ bí mật tránh việc bị truy cập bất hợp pháp, làm lộ hoặc sửa đổi. Modulo n và số mũ công khai e phải được công bố công khai;
3. Mỗi người sử dụng cần có Modulo n riêng;
4. Độ dài của Modulo n $(nlen)$ không được nhỏ hơn 2048 bit và nên được thay đổi theo thời gian như sau.

Năm (y)	độ an toàn	*nlen* tối thiểu
y<2020	112	2048
	128	3072

Trong đó, độ mạnh về an toàn $(security\_strength)$ là một số nguyên biểu thị lượng tính toán cần thiết để phá hệ mã.

Vì các phương pháp phá hệ mã thường xuyên được hoàn thiện nên cần phải định kỳ 3 đến 5 năm một lần xem xét lại $nlen$ tối thiểu (có thể tham khảo chi tiết yêu cầu này trong tài liệu *NIST Special Publication 800-57: Recommendation for Key Management – Part1: General, January 2016*).

2.1.5.2. Quy định chi tiết về các khóa RSA

1. Số mũ công khai e cần phải được chọn với các ràng buộc sau:

a) Số mũ công khai e cần được chọn trước khi tạo số mũ bí mật d;

b) Số mũ công khai e cần phải là số nguyên dương lẻ sao cho

$$65{,}537 \leq e < 2^{nlen-2security\_strength}$$

Với $nlen$ là độ dài của modulo n theo bit.

Chú ý rằng e có thể là giá trị bất kỳ mà thoả mãn ràng buộc 1(b); p và q sẽ được chọn (trong mục 2) sao cho e là nguyên tố cùng nhau với cả $(p-1)$ và $(q-1)$.

2. Hai số nguyên tố p và q được tạo ngẫu nhiên và giữ bí mật cần phải được chọn với các ràng buộc sau:

a) $(p-1)$ và $(q-1)$ cần phải nguyên tố cùng nhau với số mũ công khai e;

b) Mỗi một trong bốn số $(p+1)$, $(p-1)$ và $(q+1)$, $(q-1)$ cần phải có các nhân tử nguyên tố lớn hơn $2^{security\_strength+20}$;

c) Nhân tử nguyên tố bí mật p, q cần phải được chọn ngẫu nhiên từ các số nguyên tố thoả mãn $(\sqrt{2})(2^{(nlen/2)-1}) \leq q < p \leq 2^{(nlen/2)} - 1$;

d) $|p-q| > 2^{(nlen/2-100)}$.

3. Số mũ bí mật d cần phải được lựa chọn sau khi tạo p và q với các ràng buộc:

a) Số mũ d cần phải lớn hơn $2^{(nlen/2)}$, và

b) $d = e^{-1} mod\ (LCM((p-1),(q-1)))$

(Chi tiết về hàm tạo các tham số RSA có thể tham khảo trong tài liệu *FIPS 186-4: Digital Signature Standard*).

2.1.5.3. Tạo chữ ký số

$$RSASSA - PKCS1 - V1\_5 - SIGN\ (K, M)$$

Đầu vào:	K	khóa bí mật RSA của người ký
	M	thông báo được ký theo xâu bộ 8
Đầu ra:	S	Chữ ký, một xâu bộ 8 của độ dài k trong đó k là độ dài theo bộ 8 của RSA modulo n
Các lỗi		"Thông báo quá dài"; "Modulo RSA quá ngắn"

Các bước thực hiện:

1. Mã hóa *EMSA-PKCS1-v1_5*:

Áp dụng mã hóa *EMSA-PKCS1-v1_5* đối với thông báo M để tạo ra bản mã EM có độ dài là k bộ 8.

$$EM = EMSA - PKCS1 - V1\_5 - ENCODE\ (M, k).$$

2. Chữ ký *RSA*:

a. Biến đổi bản mã EM thành số nguyên m

$$m = OS2IP(EM)\ .$$

b. Áp dụng chữ ký số nguyên thủy $RSASP1$ cho khóa riêng K và thông điệp m để tạo ra số nguyên s

$$s = RSASP1\ (K, m).$$

c. Biến đổi s tương ứng thành chữ ký S có độ dài k bộ 8

$$S = I2OSP\ (s, k).$$

3. Đưa ra chữ ký S.

2.1.5.4. Xác thực chữ ký số

$RSASSA - PKCS1 - V1\_5 - VERIFY\ ((n, e), M, S)$

Đầu vào:	(n, e)	khóa công khai RSA của người ký
	M	thông báo (chữ ký số) cần xác thực theo xâu bộ 8
	S	Chữ ký cần xác thực là xâu bộ 8 độ dài k trong đó k là độ dài theo bộ 8 của RSA modulo n
Đầu ra:		"Chữ ký hợp lệ" hoặc "chữ ký không hợp lệ"
Các lỗi:		"Thông điệp quá dài"; "Modulo RSA quá ngắn"

Các bước thực hiện:

1. Kiểm tra độ dài: Nếu độ dài của chữ ký S không phải là k các bộ 8 thì xuất đầu ra "chữ ký không hợp lệ" và dừng lại.

2. Xác thực *RSA*:

a. Biến đổi S thành số nguyên s

$$s = OS2IP\ (S).$$

b. Sử dụng $RSAVP1$ nguyên thủy cho khóa RSA công khai (n, e) và s để tạo ra số nguyên m

$$m = RSAVP1\ ((n, e), s).$$

Nếu đầu ra $RSAVP$ "chữ ký số đại diện nằm ngoài dải hợp lệ" thì đưa ra "chữ ký số không hợp lệ" và dừng lại.

c. Biến đổi m thành EM có độ dài k các bộ 8:

$$EM = I2OSP\ (m, k).$$

Nếu đầu ra $I2OSP$ "số nguyên quá lớn" thì đưa ra "chữ ký số không hợp lệ" và dừng lại.

3. *EMSA-PKCS1-v1_5 encoding*:

Áp dụng mã hóa *EMSA-PKCS1-v1_5* đối với thông điệp M để tạo ra bản mã thứ 2 EM' có độ dài k các bộ 8:

$$EM' = EMSA - PKCS1 - V1\_5 - ENCODE\ (M, k)\ .$$

4. So sánh EM *và* EM' nếu giống nhau thì chữ ký hợp lệ, ngược lại chữ ký là không hợp lệ.

2.2. Hàm băm

Hàm băm: Một thuật toán chuyển đổi mỗi thông báo biểu diễn dưới dạng bit có độ dài bất kỳ thành một chuỗi bit có độ dài cố định. Chuỗi bit có độ dài cố định đó được gọi là "giá trị băm", "mã băm", hai đơn giản là "tóm lược" của thông báo đầu vào. Các thuật toán băm mật mã được thiết kế sao cho thỏa mãn các tính chất sau:

- Tính một chiều hay tính kháng tiền ảnh: Không thể tìm được trong thời gian cho phép một thông báo có giá trị băm cho trước;
- Tính kháng tiền ảnh thứ hai: Cho trước thông điệp M1, không thể tìm được trong thời gian cho phép một thông điệp M2 khác M1 sao cho giá trị băm của M1 và M2 là như nhau;
- Tính kháng xung đột: Không thể tìm được hai chuỗi bit khác nhau có cùng một giá trị băm.

Trong Quy chuẩn này các hàm băm sau đây được phép sử dụng.

Độ an toàn	Hàm băm áp dụng
128	SHA-256, SHA-512/256, SHA3-256
192	SHA-384, SHA3-384
≥ 256	SHA-512, SHA3-512

3. QUY ĐỊNH VỀ QUẢN LÝ

3.1. Các mức giới hạn của đặc tính kỹ thuật mật mã và yêu cầu quản lý của chữ ký số nêu tại Quy chuẩn này là các chỉ tiêu chất lượng phục vụ được quản lý theo quy định về quản lý chất lượng sản phẩm, dịch vụ mật mã dân sự được quy định tại Luật An toàn thông tin mạng ngày 19 tháng 11 năm 2015.

3.2. Hoạt động kiểm tra chất lượng sản phẩm, dịch vụ mật mã được cơ quan quản lý nhà nước có thẩm quyền tiến hành định kỳ hàng năm hoặc đột xuất.

3.3. Một số quy định về ngưỡng thời gian và độ an toàn khóa cụ thể:

- Quy định nghiệp vụ chung đối với độ an toàn khóa mật mã:

Độ an toàn theo bit	Thời hạn sử dụng quy định
96	Đến cuối năm 2020
112	Đến cuối năm 2030
≥ 128	Từ năm 2030

- Thời hạn sử dụng được quy định chi tiết đối với độ an toàn khóa mật mã khóa công khai tính theo bit:

Thời hạn Quy định	RSA	DSA	ECDSA (Độ dài p)
2020	$\|p\| = 1536$	$\|p\| = 1536, \|q\| = 192$	192 - 224
2030	$\|p\| = 2048$	$\|p\| = 2048, \|q\| = 224$	224 - 255
Sau 2030	$\|p\| = 3072$	$\|p\| = 3072, \|q\| = 256$	256

- Các hàm băm tương ứng theo Độ an toàn bit:

Hàm băm	Độ an toàn
SHA-224	112
SHA-256	128
SHA-512	256
SHA-384	192
WHIRLPOOL	256

- Độ an toàn theo bit quy đổi giữa RSA, DSA và ECDSA như sau:

Độ an toàn	ECDSA	RSA	DSA
112	224	2048	2048
128	256	3072	3072
192	384	7680	7680
256	512	15360	15360

3.4. Quy định về an toàn cài đặt và sử dụng:

Các thuật toán chữ ký số khi cài đặt phần mềm và phần cứng còn cần có đủ khả năng chống lại các tấn công kênh kề nhất là chống lại việc tính ra được các bit khóa trong quá trình thực hiện thuật toán.

4. TRÁCH NHIỆM CỦA TỔ CHỨC, CÁ NHÂN

4.1. Các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự có trách nhiệm đảm bảo tuân thủ Quy chuẩn này và chịu sự kiểm tra của cơ quan quản lý nhà nước theo quy định.

4.2. Doanh nghiệp cung cấp sản phẩm, dịch vụ mật mã dân sự cho các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) có trách nhiệm thực hiện công bố hợp quy sản phẩm, dịch vụ mật mã dân sự phù hợp với Quy chuẩn này. Việc công bố hợp quy thực hiện theo Thông tư số 28/2012/TT-BKHCN ngày 12 tháng 12 năm 2012 của Bộ Khoa học và Công nghệ.

4.3. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã – Ban Cơ yếu Chính phủ có trách nhiệm tiếp nhận đăng ký công bố hợp quy, thực hiện quản lý, hướng dẫn và kiểm tra việc công bố hợp quy.

5. TỔ CHỨC THỰC HIỆN

5.1. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã – Ban Cơ yếu Chính phủ có trách nhiệm hướng dẫn, tổ chức triển khai quản lý kỹ thuật mật mã của Chữ ký số theo Quy chuẩn này.

5.2. Trong trường hợp các quy định nêu tại Quy chuẩn kỹ thuật quốc gia này có sự thay đổi, bổ sung hoặc được thay thế thì thực hiện theo quy định tại văn bản mới./.

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

QCVN 6 : 2016/BQP

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ QUẢN LÝ KHÓA SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on key management used in banking

HÀ NỘI – 2016

MỤC LỤC

Lời nói đầu

QCVN 6 : 2016/BQP do Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã – Ban Cơ yếu Chính phủ biên soạn, Ban Cơ yếu Chính phủ trình duyệt, Bộ Khoa học và Công nghệ thẩm định và được ban hành theo Thông tư số 161/2016/TT-BQP ngày 21 tháng 10 năm 2016 của Bộ trưởng Bộ Quốc phòng.

QUY CHUẨN KỸ THUẬT QUỐC GIA VỀ QUẢN LÝ KHÓA SỬ DỤNG TRONG LĨNH VỰC NGÂN HÀNG

National technical regulation on key management used in banking

1. QUY ĐỊNH CHUNG

1.1. Phạm vi điều chỉnh

Quy chuẩn kỹ thuật quốc gia này quy định các yêu cầu về quản lý khoá mật mã sử dụng kỹ thuật mật mã phi đối xứng để bảo mật dữ liệu trong lĩnh vực ngân hàng bao gồm: Sinh khoá bí mật dùng để liên lạc giữa hai thực thể bằng cơ chế thoả thuận khoá và dẫn xuất khoá, sử dụng kỹ thuật mật mã phi đối xứng; sinh khoá bí mật cho một thực thể bởi thực thể khác bằng cơ chế truyền khoá sử dụng mật mã phi đối xứng; vận chuyển khoá công khai của một thực thể đến một thực thể khác bằng đường truyền có bảo vệ; quy định về quản lý sử dụng khoá mật mã an toàn.

Quy định về tạo khoá, đăng ký khoá, thu hồi, cài đặt, khôi phục và các vấn đề thuộc quản lý khoá khác không thuộc phạm vị điều chỉnh của quy chuẩn này.

1.2. Đối tượng áp dụng

Quy chuẩn này áp dụng đối với các doanh nghiệp kinh doanh sản phẩm, dịch vụ mật mã dân sự trong lĩnh vực ngân hàng; các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự.

1.3. Tài liệu viện dẫn

- TCVN 7817-3:2007 (ISO/IEC 11770-3:1999) Công nghệ thông tin – Kỹ thuật mật mã – Quản lý khóa – Phần 3: Các cơ chế sử dụng kỹ thuật phi đối xứng.
- ISO/IEC 11770-3:2015 Information technology – Security techniques – Key management – Part 3: Mechanisms using asymmetric techniques.
- Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, SP 800-56A Revision 2, National Institute of Standards and Technology, May 2013.
- *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, NIST SP 800-90A Rev. 1, National Institute of Standards and Technology, June 2015. (*Khuyến cáo cho bộ sinh số ngẫu nhiên sử dụng bộ sinh bit ngẫu nhiên tất định*, NIST SP 800-90A Rev. 1, Viện tiêu chuẩn và công nghệ quốc gia (Mỹ), tháng 6 năm 2015).

1.4. Giải thích từ ngữ

Trong Quy chuẩn này, các từ ngữ dưới đây được hiểu như sau:

1.4.1.

Thông tin không thuộc phạm vi bí mật nhà nước

Là thông tin không thuộc nội dung tin "tuyệt mật", "tối mật" và "mật" được quy định tại Pháp lệnh Bảo vệ bí mật nhà nước ngày 28 tháng 12 năm 2000.

1.4.2.

Mật mã

Là những quy tắc, quy ước riêng dùng để thay đổi hình thức biểu hiện thông tin nhằm bảo đảm bí mật, xác thực, toàn vẹn của nội dung thông tin.

1.4.3.

Mật mã dân sự

Là kỹ thuật mật mã và sản phẩm mật mã được sử dụng để bảo mật hoặc xác thực đối với thông tin không thuộc phạm vi bí mật nhà nước.

1.4.4.

Sản phẩm mật mã dân sự

Là các tài liệu, trang thiết bị kỹ thuật và nghiệp vụ mật mã để bảo vệ thông tin không thuộc phạm vi bí mật nhà nước.

1.4.5.

Kỹ thuật mật mã

Là phương pháp, phương tiện có ứng dụng mật mã để bảo vệ thông tin.

1.4.6.

Mã hóa

Phép biến đổi (khả nghịch) dữ liệu bởi thuật toán mật mã để tạo ra bản mã, tức là giấu nội dung thông tin của dữ liệu.

1.4.7.

Giải mã

Phép toán ngược với phép mã hóa tương ứng.

1.4.8.

Mã phi đối xứng

Hệ thống dựa trên kỹ thuật mật mã phi đối xứng, trong đó phép biến đổi công khai được sử dụng để mã hóa, phép biến đổi bí mật được sử dụng để giải mã.

1.4.9.

Kỹ thuật mật mã phi đối xứng

Kỹ thuật mật mã phi đối xứng sử dụng hai phép biến đổi liên quan đến nhau, phép biến đổi công khai (được xác định bởi khóa công khai) và phép biến đổi bí mật (được xác định bởi khóa riêng). Cả hai phép biến đổi này có tính chất là cho biết phép biến đổi công khai, về mặt tính toán không thể có khả năng xác định được phép biến đổi bí mật.

1.4.10.

Thẻ khoá

Thông điệp quản lý khoá được gửi từ một thực thể tới một thực thể khác trong quá trình thực hiện một cơ chế quản lý khoá.

1.4.11.

Vận chuyển khoá

Tiến trình truyền một khoá từ một thực thể đến một thực thể khác với bảo vệ thích hợp.

1.4.12.

Xác thực thực thể lẫn nhau

Sự xác thực giữa hai thực thể đảm bảo về định danh của mỗi thực thể.

1.4.13.

Xác thực khoá từ thực thể *A* đến thực thể *B*

Đảm bảo cho B rằng chỉ có *A* là thực thể sở hữu khoá đúng.

1.4.14.

Xác thực khoá hai chiều

Đảm bảo xác thực khoá từ *A* đến *B* và từ *B* đến *A*.

1.4.15.

Xác nhận khoá từ *A* đến *B*

Đảm bảo cho thực thể *B* là thực thể *A* sở hữu khoá đúng.

1.4.16.

Xác nhận khoá hai chiều

Đảm bảo xác nhận khoá từ *A* đến *B* và từ *B* đến *A*.

1.4.17.

Thiết lập khoá

Quá trình đảm bảo sự khả dụng một khoá bí mật dùng chung cho một hoặc nhiều thực thể. Thiết lập khoá bao gồm thoả thuận khoá và vận chuyển khoá.

1.4.18.

Thoả thuận khoá

Tiến trình kiến tạo một khoá bí mật dùng chung giữa hai thực thể theo cách mà không có bên nào có thể định trước giá trị cho khoá

1.5. Các ký hiệu

ID_A, ID_B	Định danh của các thực thể A và B
$Cert_X$	Chứng chỉ khoá công khai của thực thể X
$F(h, g)$	Hàm thỏa thuận khóa
$HASH$	Hàm băm
DH	Diffie-Hellman

MQV	Menezes-Qu-vanstone
K	Khóa bí mật cho hệ mật đối xứng
K_{AB}	Khóa bí mật chia sẻ giữa hai thực thể A và B
KT_{Ai}	Thông báo thỏa thuận khóa được gửi bởi thực thể A sau giai đoạn xử lý i
r	Số ngẫu nhiên được sinh trong quá trình thực hiện của một lược đồ.
H	Tập các phần tử có thể của r
S_X	Hàm tạo chữ ký sử dụng khóa riêng của thực thể X
V_X	Hàm kiểm tra chữ ký của thực thể X
$\|\|$	Phép nối hai phần tử dữ liệu với nhau
$\lceil x \rceil$	Số nguyên nhỏ nhất lớn hơn hoặc bằng số thực x
$len(x)$	Độ dài số nguyên x tính bằng bit
$nlen$	Độ dài modulo n tính theo bit
(n, e)	Khoá công khai theo RSA
(n, d)	Khoá riêng theo RSA
L	Độ dài số nguyên tố p trong bài toán logarit rời rạc
N	Độ dài số nguyên tố q trong bài toán logarit rời rạc
q	Ước nguyên tố của $p - 1$
$\#(E)$	Cấp hay lực lượng của đường cong elliptic E
m	Bậc của điểm sinh G
$X(P)$	Hoành độ x của điểm P trên đường cong elliptic E
h	Các đồng thừa số được tính theo công thức $h = \#E/m$
l	Thừa số phụ trong phép nhân đồng thừa số $l = h^{-1} mod\, m$
$\pi(P)$ –	Phép biến đổi điểm P trên đường cong E thành số nguyên $\pi(P) = \left(X(P) mod 2^{\lceil \rho/2 \rceil}\right) + 2^{\lceil \rho/2 \rceil}, \rho = \lceil log_2 n \rceil$

2. QUY ĐỊNH KỸ THUẬT

2.1. Các yêu cầu đối với giao thức thoả thuận và vận chuyển khoá

Điều này quy định các giao thức thiết lập khoá bí mật dùng để mã dữ liệu được trao đổi giữa hai thực thể A và B, vận chuyển khoá bí mật từ thực thể A sang thực thể B, vận chuyển khoá công khai của thực thể A sang thực thể B.

Để thực hiện giao thức, mỗi thực thể X đảm bảo các điều kiện sau:

- Sở hữu cặp khoá để ký và kiểm tra chữ ký (S_X, V_X) được cơ quan thẩm quyền cấp dưới dạng chứng thư số $Cert_X$.
- Sử dụng một cặp khoá công khai để mã hoá (E_X, D_X) được quy định tại Điều 2.7 của Quy chuẩn này.
- Sử dụng chung với thực thể thứ hai hàm thoả thuận khoá F là một trong hai hàm được xác định tại Điều 2.2.2, một hàm dẫn xuất khoá KDF được quy định tại Điều 2.4 của Quy chuẩn này và một hàm kiểm tra mật mã $MAC_{K_{AB}}$ dưới dạng hàm băm được quy định tại Điều 2.2 của QCVN 5 : 2016/BQP Quy chuẩn kỹ thuật quốc gia về Chữ ký số sử dụng trong lĩnh vực ngân hàng.
- Mỗi thực thể được tiếp cận các khoá công khai của thực thể kia theo cơ chế vận chuyển khoá công khai tại Điều 2.6.

2.2. Giao thức thỏa thuận khóa sử dụng kỹ thuật mật mã phi đối xứng trên trường hữu hạn

Thoả thuận khoá bí mật giữa hai thực thể A và B được thực hiện qua 5 bước:

Bước 1-4. Hai bên thoả thuận bí mật chia sẻ K_{AB}.

Bước 5. Hai bên sử dụng hàm dẫn xuất khoá KDF được quy định tại Điều 2.4 để thiết lập khoá bí mật chung *K*.

Lược đồ thoả thuận khoá bí mật chia sẻ K_{AB} được thể hiện trên Hình 1.

2.2.1 Giao thức

Bước 1. Kiến thiết thẻ khoá (A1): Thực thể A sinh một giá trị ngẫu nhiên và bí mật r_A thuộc H, tính $F(r_A, g)$, kiến thiết thẻ khoá KT_{A1} và gửi tới thực thể B:

$$KT_{A1} = F(r_A, g) \parallel Text1$$

Bước 2. Xử lý thẻ khoá và Kiến thiết khoá (B1): Thực thể B sinh một giá trị ngẫu nhiên và bí mật r_B thuộc H, tiếp đó tính $F(r_B, g)$, rồi tính khóa bí mật dùng chung:

$$K_{AB} = F(r_B, F(r_A, g))$$

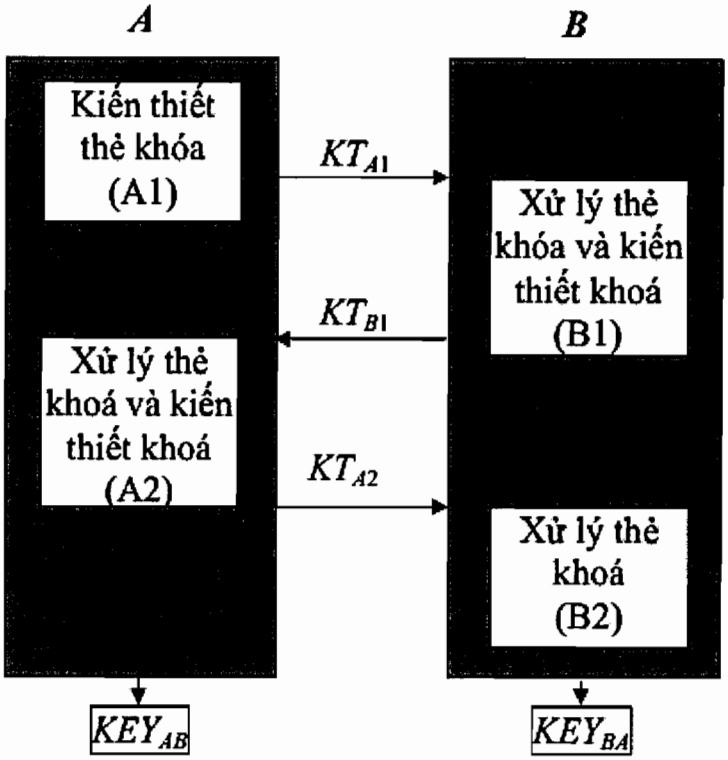
Tiếp theo, thực thể *B* tạo thẻ khoá KT_{B1}được ký như sau:

$$KT_{B1} = S_B(DB_1) \parallel MAC_{K_{AB}}(DB_1) \parallel Text3$$

trong đó

$$DB_1 = F(r_B, g) \parallel F(r_A, g) \parallel ID_A \parallel Text2$$

và gửi ngược trở lại cho thực thể A.



Hình 1: Giao thức thoả thuận khoá bí mật chia sẻ K_{AB}

Bước 3. Xử lý thẻ khoá (A2): Thực thể A kiểm tra chữ ký của thực thể B trên thẻ khoá KT_{B1} bằng cách sử dụng khóa kiểm tra công khai của B, kiểm tra định danh phân biệt của A và giá trị $F(r_A, g)$ đã được gửi ở Bước 1 **(A1).** Nếu quá trình kiểm tra thành công thì thực thể A sẽ tính khóa bí mật dùng chung là:

$$K_{AB} = F(r_A, F(r_B, g))$$

Thực thể A lại sử dụng khóa bí mật dùng chung K_{AB} để kiểm tra giá trị kiểm tra mật mã $MAC_{K_{AB}}(DB_1)$. Sau đó thực thể A tạo thẻ khoá KT_{A2} được ký như sau:

$$KT_{A2} = S_A(DB_2) \parallel MAC_{K_{AB}}(DB_2) \parallel Text5$$

trong đó

$$DB_2 = F(r_A, g) \parallel F(r_B, g) \parallel ID_B \parallel Text4$$

và gửi thẻ này tới thực thể B.

Bước 4. Xử lý thẻ khoá (B2): Thực thể B kiểm tra chữ ký của thực thể A trên thẻ khoá KT_{A2} sử dụng khóa kiểm tra công khai của A, sau đó kiểm tra định danh phân biệt của B và kiểm tra các giá trị $F(r_A, g)$,và $F(r_B, g)$, xem có phù hợp với các giá trị được trao đổi ở các bước trước hay không. Nếu quá trình kiểm tra thành công thì thực thể B sẽ kiểm tra giá trị kiểm tra mật mã $MAC_{K_{AB}}(DB_2)$ bằng cách tính:

$$K_{AB} = F(r_A, F(r_B, g))$$

Bước 5. Thực thể A và thực thể B tính khoá bí mật chung K

Các thực thể A và B sử dụng hàm dẫn xuất khoá tại Điều 2.4 để tính khoá

$$K = KDF(K_{AB}, OtherInput)$$

Trong đó $OtherInput$ chứa các định danh ID_A, ID_B của A,B và các thông tin khác do A và B thoả thuận.

CHÚ THÍCH: Các trường $Text1$, $Text2$, $Text3$, $Text4$, $Text5$ là những trường dữ liệu chứa chứng thư số của A và B, và có thể một số thông tin khác như tem thời gian, định danh phiên liên lạc, v.v..)

2.2.2. Hàm thoả thuận khoá

2.2.2.1. Hàm thoả thuận khoá DH

Cho trường hữu hạn nguyên tố F_p, ký hiệu tập $H = \{1, \dots, p-2\}$; g là phần tử sinh thuộc hàm thoả thuận khoá Diffie-Hellman (DH) được xác định theo công thức:

$$F(h, g) = g^h, h \in H$$

CHÚ THÍCH: Với hàm thoả thuận DH thì thẻ khoá KT_{A1} có dạng :

$$KT_{A1} = g^{r_A} \parallel Text_A$$

2.2.2.2. Hàm thoả thuận khoá MQV

- $(a, A); (b, B)$ là cặp khoá phi đối xứng tĩnh của A và B, $A = g^a, B = g^b$
- $(x, X), (y, Y)$ là cặp khoá phi đối xứng tức thời của A và B, $X = g^x, Y = g^y$, $d = 2^l + (X \bmod 2^l), e = 2^l + (Y \bmod 2^l), l = |p|/2$, l độ dài p tính bằng bit, $F(x, g) = (YB^e)^{x+da} = (XA^d)^{y+eb}$

2.3. Giao thức thoả thuận khoá sử dụng mật mã trên đường cong elliptic

2.3.1. Giao thức DH

Bước 1. Kiến thiết thẻ khoá (A1)

Thực thể A chọn ngẫu nhiên và bí mật số r_A thuộc khoảng $\{2, \dots, m-2\}$, tính $r_A G$, kiến thiết thẻ khoá $KT_{A1} = r_A G$ và gửi cho B.

Bước 2. Xử lí thẻ khoá và kiến thiết thẻ khoá (B1)

Thực thể B kiểm tra thẻ khoá KT_{A1} có phải là điểm nằm trên đường cong Elliptic hay không (kiểm tra theo tiêu chuẩn ISO/IEC 15946-1). Thực thể B chọn ngẫu nhiên và bí mật r_B thuộc khoảng $\{2, \dots, m-2\}$, tính $r_B G$, tính bí mật chia sẻ $K_{AB} = (r_B l)(hKT_{A1})$, kiến thiết thẻ được ký hiệu KT_{B1}:

$KT_{B1} = S_B(DB_1) \parallel MAC_{K_{AB}}(DB_1)$ với $DB_1 = r_B G \parallel KT_{A1} \parallel ID_A \parallel Text1$

và gửi cho A.

Bước 3. Xử lí thẻ khoá (A2)

Thực thể A kiểm tra chữ ký của B trên thẻ khoá KT_{B1} sử dụng khoá kiểm tra công khai của B. Nếu sử dụng lược đồ chữ ký có khôi phục bản rõ thì việc kiểm tra bao gồm cả việc khôi phục khối dữ liệu DB_1 từ chữ ký và kiểm tra liệu định danh phân biệt của A và giá trị $r_A G$ có chứa trong DB_1 không. Nếu sử dụng chữ ký có đính kèm bản rõ thì việc kiểm tra bao gồm cả việc thiết kế lại khối dữ liệu DB_1 sử dụng giá trị trong KT_{A1}, định danh phân biệt của A, giá trị nhận được $r_B G$ và kiểm tra chữ ký trên khối dữ liệu này.

Tiếp đó thực thể A kiểm tra, liệu giá trị $r_B G$ nhận được từ KT_{B1} có phải là điểm trên đường cong elliptic hay không (kiểm tra theo tiêu chuẩn ISO/IEC 15946-1). Nếu đúng thì A tính khoá chia sẻ $K_{AB} = (r_A.l)(h.r_B G)$.

Sử dụng K_{AB}, thực thể A kiểm tra $MAC_{K_{AB}}(DB_1)$. Tiếp đó A thiết kế thẻ khoá có ký hiệu $KT_{A2} = S_A(DB_2) \| MAC_{K_{AB}}(DB_2)$ ở đây $DB_2 = r_A G \| r_B G \| B \| Text2$ ($DB_2 = r_A G \| r_B G \| ID_B \| Text2$) và gửi cho B.

Bước 4. Xử lí thẻ khoá (B2)

Thực thể B kiểm tra chữ ký của A trên thẻ khoá KT_{A2} bằng cách sử dụng khoá kiểm tra công khai của A. Nếu sử dụng sơ đồ chữ ký có khôi phục bản rõ thì điều này bao gồm cả việc khôi phục khối dữ liệu DB_2 từ chữ ký và kiểm tra định danh phân biệt của B, các giá trị $r_A G$ và $r_B G$ có chứa trong khối này không. Nếu sử dụng chữ ký có đính kèm bản rõ thì việc kiểm tra bao gồm cả việc thiết kế lại khối dữ liệu DB_2 sử dụng các giá trị trong KT_{A1}và KT_{B1}, định danh phân biệt của B và kiểm tra chữ ký trên khối dữ liệu này.

Nếu việc kiểm tra thành công thì thực thể B kiểm tra $MAC_{K_{AB}}(DB_2)$ sử dụng khoá chia sẻ $K_{AB} = (r_B l)(hKT_{A1})$.

Bước 5. Thiết lập khoá bí mật K

Thực thể A và B sử dụng hàm dẫn xuất khoá KDF tại Điều 2.4 để thiết lập khoá bí mật.

$$K = KDF(K_{AB}, OtherInput).$$

2.3.2. Thoả thuận khoá MQV

Bước 1. Thiết kế thẻ khoá (A1)

Thực thể A chọn ngẫu nhiên và bí mật số r_A thuộc khoảng $\{2,...,m-2\}$, tính $r_A G$ và kiến thiết thẻ khoá $KT_{A1} = r_A G$ và gửi cho thực thể B.

Bước 2. Kiến thiết thẻ khoá (B1)

Thực thể B kiểm tra thẻ KT_{A1} có phải là điểm trên đường cong elliptic (kiểm tra theo tiêu chuẩn ISO/IEC 15946-1:2016). Thực thể B chọn ngẫu nhiên và bí mật số r_B thuộc khoảng $\{2,...,m-2\}$, tính $r_B G$, thiết kế thẻ khoá $KT_{B1} = r_B G$, tiếp đó tính khoá bí mật chia sẻ K_{AB}:

$$K_{AB} = ((r_B + \pi(KT_{B1})d_B).l)(h.(KT_{A1} + \pi(KT_{A1})P_A))$$

Tiếp đó B tính $K = KDF(K_{AB})$ và gửi $f_{K_{AB}}(2, KT_{A1}, KT_{B1})$ gửi $MAC_K(2, KT_{A1}, KT_{B1})$ cho A cùng với thẻ khoá KT_{B1}.

Bước 3. Kiến thiết thẻ khoá (A2)

Thực thể A tính khoá bí mật chia sẻ :

$$K_{AB} = ((r_A + \pi(KT_{A1})d_A).l)(h.(KT_{B1} + \pi(KT_{B1})P_B))$$

và kiểm tra $MAC_K(2, KT_{A1}, KT_{B1})$

Tiếp đó A tính $MAC_K(3, KT_{A1}, KT_{B1})$và gửi cho B.

Bước 4. Kiểm tra (B2)

Thực thể B tính $MAC_K(3, KT_{A1}, KT_{B1})$ và kiểm tra thực thể A.

2.4. Hàm dẫn xuất khóa KDF

2.4.1. Hàm dẫn khoá 1

Dạng thức của KDF

$KDF(Z, OtherInput)$ trong đó $OtherInput$ là $keydatalen$ và $OtherInfo$ (các đại lượng này được giải thích về sau).

Các giá trị cố định:

$hashlen$: số nguyên chỉ độ dài đầu ra (theo bit) của hàm băm được sử dụng để dẫn xuất ra các khối của dữ liệu khóa bí mật.

$max\_hash\_inputlen$: số nguyên có giá trị là độ dài lớn nhất (theo bit) của (các) chuỗi bit đầu vào của hàm băm.

Các hàm bổ trợ:

H: là hàm băm được chấp thuận là hàm băm được quy định tại QCVN 5 : 2016/BQP Quy chuẩn kỹ thuật quốc gia về Chữ ký số sử dụng trong lĩnh vực ngân hàng.

Đầu vào:

Z: Chuỗi byte bí mật chia sẻ trước.

$keydatalen$: số nguyên chỉ độ dài (theo bit) của dữ liệu khóa bí mật được sinh ra; $keydatalen$ cần nhỏ hơn hoặc bằng $Hashlen \times (2^{32} - 1)$

$OtherInfo$: Chuỗi bit sau:

$$AlgorithmID||PartyAInfo||PartyBInfo\{||SuppPubInfo\}\{||SuppPrivInfo\}$$

trong đó các trường con được định nghĩa như sau:

- $AlgorithmID$: Chuỗi bit chỉ ra cách thức phân tách dữ liệu khóa đã được dẫn xuất ra và dữ liệu khóa được dẫn xuất sẽ được sử dụng cho những thuật toán nào. Ví dụ, $AlgorithID$ có thể chỉ ra rằng các bit 1-80 được dùng là 80-bit khóa cho $HMAC$ và các bit 81-208 được dùng là 128-bit khóa cho AES.

- $PartyAInfo$: Một chuỗi bit chứa các thông tin công khai được yêu cầu bởi ứng dụng sử dụng hàm KDF được đóng góp bởi bên A trong quá trình dẫn xuất khóa. Ở mức tối thiểu, $PartyAInfo$ chứa ID_A là định danh bên A. Xem chú ý phần dưới.

- PartyBInfo: Một chuỗi bit chứa các thông tin công khai được yêu cầu bởi ứng dụng sử dụng hàm KDF được đóng góp bởi bên B trong quá trình dẫn xuất khóa. Ở mức tối thiểu, $PartyVInfo$ chứa ID_B là định danh bên B. Xem chú ý phần dưới.

- (Tùy chọn) $SuppPubInfo$: Một chuỗi bit chứa các thông tin công khai bổ sung cả hai bên cùng biết (mutual-known)

- (Tùy chọn) $SuppPrivInfo$: Một chuỗi bit chứa thông tin bí mật bổ sung cả hai bên cùng biết (mutual-known)

(Ví dụ, một khóa bí mật đối xứng chia sẻ trước được truyền thông qua một kênh riêng biệt)

Thuật toán:

1. Tính $reps = [keydatalen/hashlen]$.

2. Nếu $(reps > 2^{32} - 1)$ thì ABORT: chỉ thị lỗi và dừng.

3. Khởi tạo bộ đếm chuỗi bit 32-bit big-endian *counter* bằng 00000001_{16}.

4. Nếu $counter||Z||OtherInfo$ có độ dài lớn hơn $max\_hash\_inputlen$ thì chỉ thị lỗi và dừng lại.

5. Vòng lặp với $i = 1$ đến $reps$, thực hiện :

5.1. Tính $Hash_i = H(counter||Z||OtherInfo)$.

5.2. Tăng $counter$ lên 1 ($modulo\ 2^{32}$), xử lý nó dưới dạng số nguyên không âm 32-bit.

6. Lấy $Hash$ là $Hash_{reps}$ nếu $(keydatalen/hashlen)$ là số nguyên, ngược lại lấy số lượng $(keydatalen\ mod\ hashlen)$ bit bên trái của đoạn dữ liệu $Hash_{reps}$.

7. Lấy $DerivedKeyingMaterial = Hash_1||\ Hash_2||\ldots.||Hash_{reps-1}||Hhash$

***Đầu ra*:**

Chuỗi bit $DerivedKeyingMaterial$ có độ dài $keydatalen$ bit (hoặc thông báo lỗi). Thuật toán KDF tạo ra dữ liệu khóa có độ dài lớn nhất là $hashlen \times (2^{32} - 1)$. Bất kỳ lời gọi hàm KDF nào trong trường hợp sử dụng giá trị $keydatalen$ lớn hơn $hashlen \times (2^{32} - 1)$ sẽ dẫn tới chỉ thị lỗi và dừng mà không cho ra $DerivedKeyingMaterial$. Bất kỳ lời gọi hàm KDF nào dùng để băm một chuỗi bit có độ dài lớn hơn $max\_hash\_inputlen$ cũng sẽ dẫn tới thông báo lỗi và dừng mà không xuất ra $DerivedKeyingMaterial$.

Chú ý:

a) ID_A và ID_B sẽ được biểu diễn trong $OtherInfo$ là hai đơn vị thông tin riêng rẽ.

b) Bên A sẽ là bên khởi tạo và bên B có thể là bên trả lời của giao thức sử dụng lược đồ thỏa thuận khóa được dùng để xác định khóa bí mật chia sẻ trước Z.

2.4.2. Hàm dẫn xuất khoá 2

Ký hiệu $hashlen$ chỉ độ dài đầu ra của hàm hash được chọn và $maxhashlen$ *là* độ dài đầu vào cực đại của hàm hash.

Đầu vào

Đầu vào của hàm dẫn xuất khoá là

- Z Xâu bit là bí mật chia sẻ.

CHÚ THÍCH: Giao thức thoả thuận khoá sử dụng mật mã trên đường cong elliptic dẫn xuất ra khoá bí mật chia sẻ K_{AB} hoặc dưới dạng một điểm trên đường cong elliptic hoặc dưới dạng ghép hai điểm. Trong trường hợp thứ nhất, để có được khoá mật chia sẻ Z làm đầu vào cho hàm dẫn xuất khoá thì phải áp dụng hàm π để chuyển điểm trên đường cong elliptic thành số nguyên và từ đó chuyển sang xâu bit. Trong trường hợp thứ hai áp dụng hàm π cho cả hai điểm để được hai số nguyên z_1, z_2; hai số nguyên này sau đó được biến đổi thành các xâu bít và được ghép lại với nhau.

- *Keydatalen* Số nguyên biểu thị độ dài tính bằng bit của dữ liệu khoá được tạo ra, nhỏ hơn đại lượng $hashlen \times (2^{32} - 1)$

- (Tuỳ chọn) Xâu bit $SharedInfor$ gồm một dữ liệu nào đó được hai thực thể dùng chung nhằm chia sẻ bí mật Z.

Thuật toán

Hàm dẫn xuất khoá được tính như sau:

1. Khởi động bộ đếm 32-bit 00000001 (Hệ thập lục)

2. Vòng lặp với $i = 1$ đến $j = \lceil keydatalen/hashlen \rceil$ thực hiện:

- Tính $Hash_i = H(Z \parallel counter \parallel SharedInfo)$

3. Tăng giá trị bộ đếm

4. Tăng i

5. Giả sử $HHash_j$ biểu thị $HHash_j$ khi $Keydatalen/hashlen$ là số nguyên và biểu thị $(Keydatalen(hashlen \times (j-1)))$ bit bên trái nhất của $Hash_i$ trong trường hợp ngược lại

6. Đặt $Keydata = Hash_1 \parallel Hash_2 \parallel \cdots \parallel Hash_{j-1} \parallel HHash_j$

Đầu ra

Là dữ liệu khoá ở dạng xâu bit có độ dài bằng $keydatalen$.

CHÚ THÍCH: Lưu ý là hàm dẫn xuất khoá tạo ra dữ liệu khoá có độ dài nhỏ hơn $hashlen \times (2^{32} - 1)$ bit. Bất kì lược đồ nào gọi hàm dẫn xuất khoá cho xâu bít lớn hơn hoặc bằng $hashlen \times (2^{32} - 1)$ bit sẽ cho ra thông báo "lỗi" và dừng lại. Tương tự, tất cả hàm dẫn xuất khoá được gọi ra không băm các xâu bít có độ dài lớn hơn $maxhashlen$. Bất kì lược đồ nào gọi hàm dẫn xuất khoá băm các xâu bít có độ dài lớn hơn $maxhashlen$ đều cho ra thông báo "lỗi" và dừng lại.

2.5. Giao thức vận chuyển khóa bí mật

Điều này trình bày giao thức vận chuyển khoá bí mật, một khoá được truyền từ thực thể A sang thực thể B và một khoá được truyền từ thực thể B sang thực thể A.

Bước 1. Kiến thiết thẻ khóa (A1): Thực thể A tạo ra một thẻ khóa KT_{A1} bao gồm một số ngẫu nhiên r_A và một trường dữ liệu tùy chọn $Text1$ rồi gửi nó cho thực thể B:

$$KT_{A1} = r_A \parallel Text1$$

Bước 2. Mã khóa khối khóa (B1.1): Thực thể B có một khóa K_B và muốn gửi một cách an toàn cho thực thể A. Trước hết B tạo ra khối dữ liệu khóa bao gồm định danh riêng biệt của bên gửi B, khóa K_B và trường dữ liệu tùy chọn $Text2$. Thực thể B mã hóa khối dữ liệu khóa này bằng phép mã công khai E_A của A, thu được khối mã:

$$BE_1 = E_A(ID_B \parallel K_B \parallel Text2)$$

Bước 3. Kiến thiết thẻ khóa (B1.2): Thực thể B tạo ra một khối thẻ khóa bao gồm định danh riêng biệt của bên nhận A, một số ngẫu nhiên r_A nhận được ở bước 1, một số ngẫu nhiên mới r_B (tùy chọn) do B tạo ra, khối đã mã BE_1 và một trường dữ liệu tùy chọn $Text3$. Tiếp đó, B tiến hành ký khối dữ liệu thẻ bằng phép ký bí mật của mình và gửi kết quả cho A:

$$KT_{B1} = S_B(r_B \parallel r_A \parallel ID_A \parallel BE_1 \parallel Text3) \parallel Text4$$

Bước 4. Kiểm tra thẻ khóa (A2.1):Thực thể A sử dụng phép kiểm tra công khai của bên gửi V_B để kiểm tra chữ ký số của thẻ khóa nhận được KT_{B1}. Tiếp đó A kiểm tra định danh riêng biệt A và kiểm tra giá trị nhận được r_A xem có khớp với số ngẫu nhiên nhận được ở Bước 1 (A1) hay không.

Bước 5. Giải mã khối khóa (A2.2): Thực thể A tiến hành giải mã khối BE_1 bằng phép giải mã bí mật D_A của mình. Tiếp đó, *A* kiểm tra định danh riêng biệt của bên gửi B. Nếu tất cả lần kiểm tra đều thành công thì *A* chấp nhận khóa K_B.

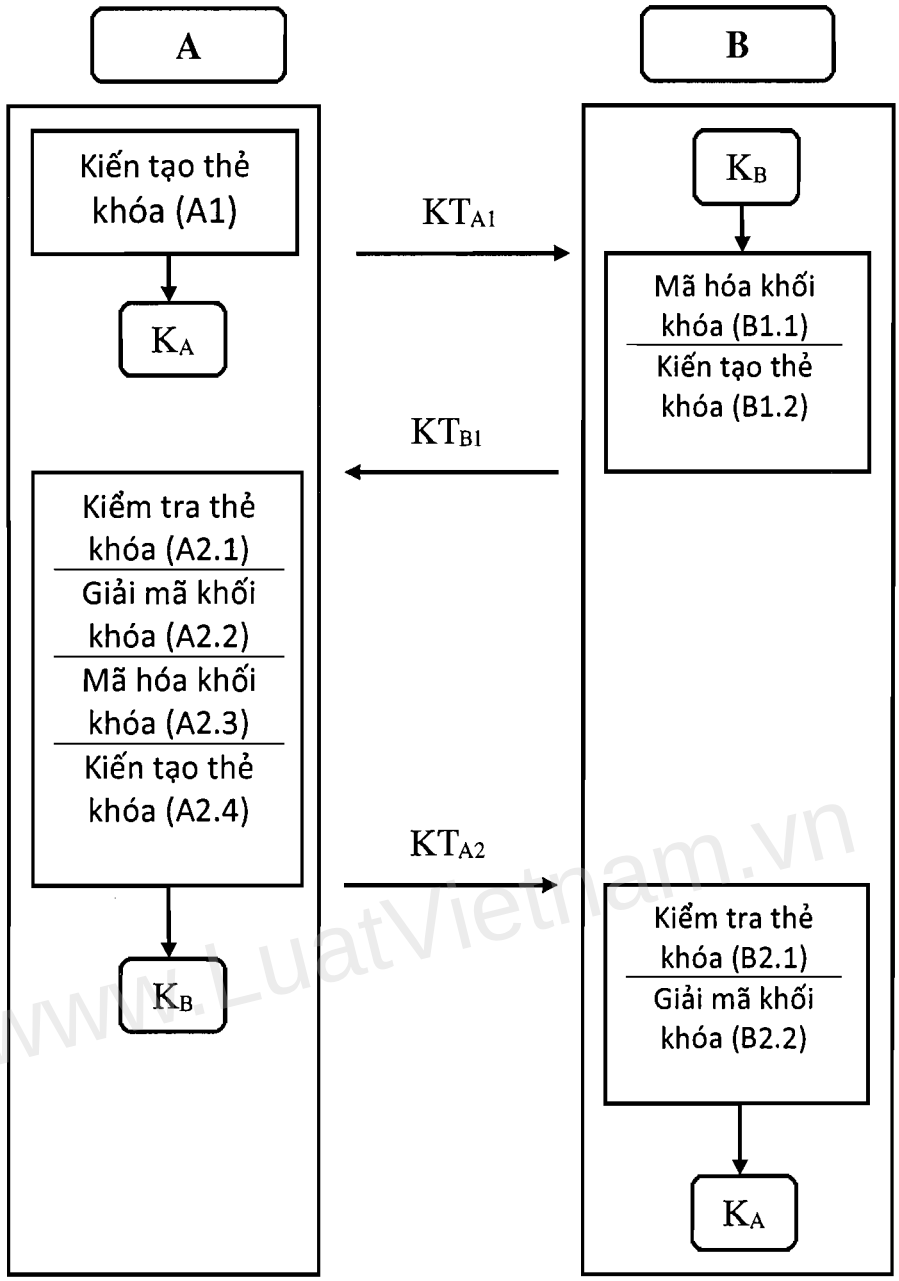
Bước 6. Mã khóa khối khóa (A2.3):Thực thể A có một khóa K_A muốn gửi cho B một cách an toàn. Trước hết, A tạo ra khối dữ liệu khóa bao gồm định danh riêng biệt của bên gửi A, khóa K_A và trường dữ liệu tùy chọn $Text5$. Tiếp đến, A mã hóa khối dữ liệu khóa này bằng phép mã công khai E_B của B để được khối mã:

$$BE_2 = E_B(ID_A \parallel K_A \parallel Text5)$$

Bước 7. Kiến thiết thẻ khóa (A2.4): Thực thể A tạo ra một khối thẻ khóa bao gồm định danh riêng biệt của bên nhận *B*, một số ngẫu nhiên r_A do A tạo ra bước 1 (A1), một số ngẫu nhiên mới r_B do B tạo ra ở (B1.2), khối đã mã BE_2 và một trường dữ liệu tùy chọn $Text6$. Tiếp đó, A tiến hành ký khối dữ liệu thẻ bằng phép ký bí mật của mình và gửi kết quả cho B:

$$KT_{A2} = S_A(r_A \parallel r_B \parallel ID_B \parallel BE_2 \parallel Text6) \parallel Text7$$

Bước 8. Kiểm tra thẻ khóa (B2.1) Thực thể B sử dụng phép kiểm tra công khai của bên gửi V_A để kiểm tra chữ ký số của thẻ khóa nhận được KT_{A2}. Tiếp đó B kiểm tra định danh riêng biệt B của mình và kiểm tra giá trị nhận được r_B xem có khớp với số ngẫu nhiên ở Bước 3 (B1.2) hay không. Ngoài ra, B cũng kiểm tra giá trị ngẫu nhiên nhận được r_A xem có khớp với số ngẫu nhiên ở Bước 1 (A1) hay không.



Hình 2 – Cơ chế vận chuyển khóa bí mật

Bước 9. Giải mã khối khóa (B2.2): Thực thể B tiến hành giải mã khối BE_2 bằng phép giải mã bí mật D_B của mình. Tiếp đó, B kiểm tra định danh riêng biệt của bên gửi A. Nếu tất cả phép kiểm tra đều thành công thì B chấp nhận khóa K_A.

CHÚ THÍCH: Trong tình huống chỉ có một bên gửi khoá bí mật cho bên kia, chẳng hạn thực thể B muốn chuyển khoá bí mật K_B cho A, thì chỉ cần thực hiện giao thức trên từ bước 1 đến bước 5.

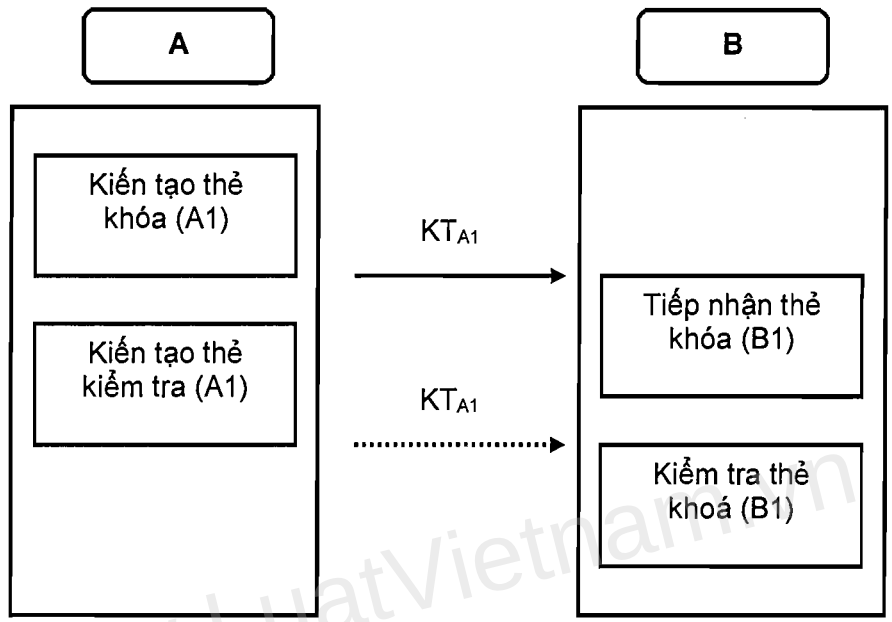
2.6. Giao thức vận chuyển khoá công khai

Điều này trình bày hai giao thức truyền thông tin khóa công khai từ một thực thể A đến một thực thể B. Giao thức thứ nhất (giao thứ 2.6.1) không sử dụng bên thứ ba tin cậy, theo giao thức này để kiểm tra tính toàn vẹn và nguồn gốc của thông tin khóa công khai, hai thực thể sử dụng một hàm băm được quy định tại QCVN 5 : 2016/BQP. Giao thức thứ hai (giao thức

2.6.2) giả thiết là chứng thư khoá công khai hợp lệ $Cert_A$ của A được cấp bởi bên thứ ba tin cậy là Tổ chức chứng thực và cấp chứng chỉ khoá công khai CA, theo giao thức này thực thể B có thể truy cập vào bản sao có xác thực của phép kiểm tra khoá công khai của tổ chức này.

2.6.1 Giao thức vận chuyển khoá công khai không sử dụng bên thứ ba tin cậy

Giao thức gồm bốn bước và được minh hoạ trên Hình 3.



Hình 3: Cơ chế vận chuyển khóa công khai

Bước 1. Kiến thiết thẻ khóa (A1): *A* tạo ra một thẻ khóa KT_{A1} bao gồm thông tin khóa công khai của A và gửi đến cho B:

$$KT_{A1} = PKI_A \parallel Text1$$

Bước 2. Tiếp nhận thẻ khóa (B1): B tiếp nhận được thẻ khóa, trích lấy thông tin khóa công khai PKI_A. Hoặc B sẽ thực hiện kiểm tra khóa kiểm tra của A hoặc sẽ lưu trữ nó ở nơi tránh được giả mạo phục vụ để cho lần kiểm tra sau hoặc sẽ sử dụng.

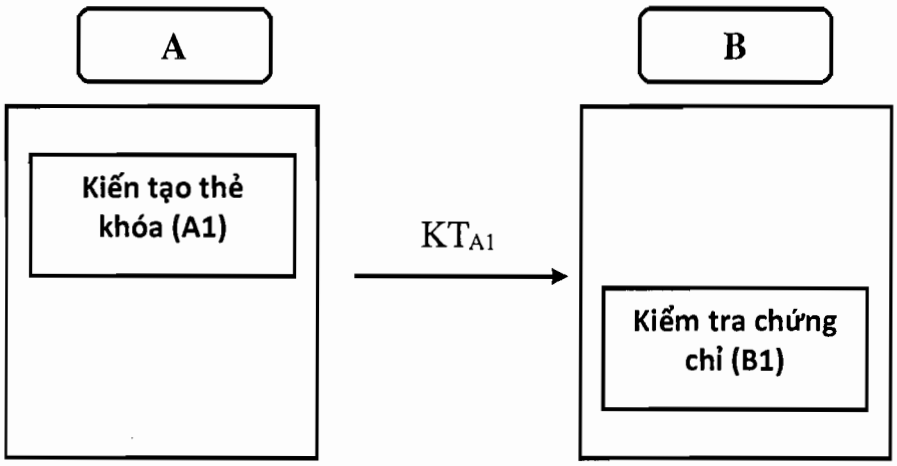
Bước 3. Kiến tạo thẻ kiểm tra (A2): A tính toán giá trị kiểm tra $hash(PKI_A)$ đối với thông tin khóa công khai của mình và gửi giá trị kiểm tra này cùng với các định danh tùy chọn riêng biệt của A và B rồi gửi đến thực thể B sử dụng một kênh truyền có xác thực và độc lập thứ hai (ví dụ kênh truyền điện báo hoặc đường truyền thư đăng ký trước).

$$KT_{A2} = A \parallel B \parallel hash(PKI_A) \parallel Text2$$

Bước 4. Kiểm tra thẻ khóa (B2): Dựa vào thông tin thẻ khóa nhận được KT_{A2}, B có thể tùy chọn kiểm tra định danh riêng biệt của A và B, tính toán ra giá trị kiểm tra trên thông tin khóa công khai của A nhận được từ thẻ khóa KT_{A1} và so sánh với giá trị kiểm tra nhận được từ thẻ khóa KT_{A2}. Nếu kết quả kiểm tra thành công thì B lấy khóa công khai của A đưa lên danh sách các khóa đang hoạt động (danh sách này được bảo vệ chống lại sự giả mạo).

2.6.2 Giao thức vận chuyển khoá công khai sử dụng bên thứ ba tin cậy

Giao thức gồm hai bước và được minh hoạ trên Hình 4.



Hình 4 – Cơ chế vận chuyển khóa công khai 2

Bước 1. Kiến thiết thẻ khóa (A1): Thực thể A tạo ra một thẻ khóa KT_A bao gồm chứng chỉ khóa công khai của A và gửi nó cho B:

$$KT_A = Cert_A \parallel Text$$

Bước 2. Kiểm tra chứng chỉ (B1): Dựa trên thông tin nhận được về chứng chỉ khóa công khai, B sử dụng phép kiểm tra công khai V_{CA} của CA để kiểm tra tính xác thực của thông tin khóa công khai và kiểm tra cả tính hợp lệ đối với khóa công khai của A.

2.7. Quy định kỹ thuật cho các tham số

2.7.1. Quy định về nguồn ngẫu nhiên

Các số ngẫu nhiên được sử dụng cho các mục đích khác nhau như để sinh các tham số mật mã, các khóa mật mã, các giá trị ngẫu nhiên dùng một lần và các giá trị thách đố xác thực.

Một số bộ sinh bit ngẫu nhiên tất định DRBG được chấp thuận để sử dụng theo quy định chung bao gồm: HASH_DRBG, HMAC_DRBG và CTR_DRBG.

Các bộ sinh bit ngẫu nhiên RBG tuân theo SP800-90A phiên bản sửa đổi lại năm 2015 để sinh bit ngẫu nhiên cũng được chấp thuận để sử dụng tiếp.

2.7.2. Quy định đối với tham số RSA

2.7.2.1. Các yêu cầu chung

1. Số mũ công khai e cần phải được chọn với các ràng buộc sau:

a) Số mũ công khai e cần được chọn trước khi tạo số mũ bí mật d;

b) Số mũ công khai *e* cần phải là số nguyên dương lẻ sao cho

$$65{,}537 \leq e < 2^{nlen-2security\_strength}$$

Với $nlen$ là độ dài của modulo *n* theo bit.

Chú ý rằng e có thể là giá trị bất kỳ thoả mãn ràng buộc 1(b); p và q sẽ được chọn (trong mục 2) sao cho *e* là nguyên tố cùng nhau với cả $(p-1)$ và $(q-1)$.

2. Hai số nguyên tố p và q được tạo ngẫu nhiên và giữ bí mật cần phải được chọn với các ràng buộc sau:

a) $(p-1)$ và $(q-1)$ cần phải nguyên tố cùng nhau với số mũ công khai e;

b) Mỗi một trong bốn số $(p+1)$, $(p-1)$ và $(q+1)$, $(q-1)$ cần phải có các nhân tử nguyên tố lớn hơn $2^{security\_strength+20}$;

c) Nhân tử nguyên tố bí mật p, q cần phải được chọn ngẫu nhiên từ các số nguyên tố thoả mãn $(\sqrt{2})(2^{(nlen/2)-1}) \leq q < p \leq 2^{(nlen/2)} - 1$;

d) $|p-q| > 2^{(nlen/2-100)}$.

3. Số mũ bí mật d cần phải được lựa chọn sau khi tạo p và q với các ràng buộc:

a) Số mũ d cần phải lớn hơn $2^{(nlen/2)}$, và

b) $d = e^{-1} mod\ (LCM((p-1),(q-1)))$

(Chi tiết về hàm tạo các tham số RSA có thể tham khảo trong tài liệu *FIPS 186-4: Digital Signature Standard*)

2.7.2.2. Quy định ngưỡng giá trị cho tham số theo thời hạn sử dụng

Thời hạn Quy định	RSA
2020	$\|p\| = 1536$
2030	$\|p\| = 2048$
Sau 2030	$\|p\| = 3072$

2.7.3. Hệ mật dựa trên Logarit rời rạc DL

Tham số của hệ mật dựa trên bài toán Logarit rời rạc trên trường hữu hạn F_p là bộ :

(p, q, g), trong đó p là đặc số của trường F_p, q là bậc của nhóm $F^*(p)$ và là ước của $p-1$ thoả mãn các điều kiện sau:

- p, q là số nguyên tố

- Độ dài của p, q được cho dưới bảng sau:

Năm	Độ dài p	Độ dài q
Đến 2020	$\|p\| = 1536$	$\|q\| = 192$
Đến 2030	$\|p\| = 2048$	$\|q\| = 224$
Sau năm 2030	$\|p\| = 3072$	$\|q\| = 256$

2.7.4. Hệ mật ECC

2.7.4.1. Quy định về các khóa

Kiểm tra tính hợp lệ của các tham số miền $(p, SEED, a, b, G, n, h)$ như sau:

Xâu $SEED$ dùng để sinh ngẫu nhiên đường cong Elliptic xác định trên trường F_p với p là số nguyên tố lẻ.

Trước khi sử dụng một bộ tham số miền, tính hợp lệ của nó phải được kiểm tra theo thuật toán dưới đây:

1. Kiểm tra p là một số nguyên tố lẻ.
2. Kiểm tra a, b, x_G, y_G là các phần tử của trường F_p.
3. Kiểm tra rằng a và b được dẫn xuất tương ứng từ $SEED$.
4. Kiểm tra $(4a^3 + 27b^2)$ khác 0 và $j(E) \neq 0; 1728$ trong F_p
5. Kiểm tra $y_G^2 = x_G^3 + ax_G + b$ trong F_p.
6. Kiểm tra n là nguyên tố và $n > 4\sqrt{p}$.
7. Kiểm tra $nG = O_E$.
8. Kiểm tra đường cong có thuộc danh sách các đường cong yếu:
   a. Thoả mãn điều kiện MOV, (chú ý rằng một đường cong thoả mãn điều kiện MOV sẽ không phải là đường cong siêu biến)
   b. Kiểm tra đường cong không kì dị, nghĩa là $\#E \neq p$.

Nếu bất kỳ sự kiểm tra nào ở trên thất bại thì tham số miền phải được xem là không hợp lệ.

Điều kiện MOV được hiểu là không có giá trị k nguyên dương nào $0 < k < B$ để cho $p^k - 1$ chia hết cho n. Trên thực hành hiện nay $|p| = 224$ bit thì người ta xét với $B = 15$ là đủ vì khi đó $|p^k| = 3360 > 2048$.

Các hệ số a, b của đường cong được sinh ngẫu nhiên trên F_p từ đầu vào *SEED* và có thể kiểm tra được.

Khóa bí mật d phải được sinh ngẫu nhiên trong khoảng $[1, n - 1]$.

Đường cong Elliptic xác định trên trường hữu hạn F_p với tối thiểu $|p| = 224$ bit và được xác định cụ thể như sau:

Độ dài bit của n	Độ dài bit của p
224 - 255	|p| = 224
256 - 383	|p| = 256
384 - 511	|p| = 384
≥ 512	|p| = 521

Đại lượng Cofactor được định nghĩa và ký hiệu là $h = \#E(F_p)/n$ tuân theo bảng dưới đây:

Độ dài bit của n	Giá trị h cực đại cho phép
224 - 255	2^{14}

256 - 383	2^{16}
384 - 511	2^{24}
≥ 512	2^{32}

2.7.5. Độ an toàn theo bit quy đổi giữa RSA, DL và ECC

Độ an toàn	ECC	RSA	DL
112	224	2048	2048
128	256	3072	3072
192	384	7680	7680
256	512	15360	15360

3. QUY ĐỊNH VỀ QUẢN LÝ

3.1. Các mức giới hạn của đặc tính kỹ thuật mật mã và yêu cầu quản lý khóa mật mã nêu tại Quy chuẩn này là các chỉ tiêu chất lượng phục vụ được quản lý theo quy định về quản lý chất lượng sản phẩm, dịch vụ mật mã dân sự được quy định tại Luật An toàn thông tin mạng ngày 19 tháng 11 năm 2015.

3.2. Hoạt động kiểm tra chất lượng sản phẩm mật mã được cơ quan quản lý nhà nước có thẩm quyền tiến hành định kỳ hàng năm hoặc đột xuất.

3.3. Quản lý sử dụng khóa mật mã:

- Các loại khoá mật mã phải được lập thành danh mục với việc mô tả chi tiết mục đích sử dụng, thời hạn sử dụng.
- Đảm bảo sử dụng khoá đúng mục đích được quy định (chẳng hạn khoá dùng để mã các khoá khác không được sử dụng để mã hoá dữ liệu), không sử dụng khoá đã hết hạn sử dụng hoặc huỷ khoá trước thời hạn đảm bảo khả năng giải mã khi dữ liệu hết thời hạn bảo mật. Không cung cấp khóa cho người nhận không có thẩm quyền.
- Khoá mật mã phải được bảo vệ chống lại các mối nguy cơ như bị lộ (ngoại trừ các khóa công khai), sửa đổi, phá hủy và tái sử dụng. Việc bảo vệ phải thực hiện suốt cả vòng đời của khoá. Mục tiêu bảo vệ phải phù hợp với từng loại khoá. Đối với khoá công khai cần đảm bảo tính toàn vẹn và tính khả dụng, đối với khoá bí mật phải đảm bảo đầy đủ cả ba tính chất: bí mật, toàn vẹn và khả dụng. Bảo vệ khoá phải kết hợp đồng bộ các giải pháp bằng kỹ thuật mật mã, phương tiện vật lý và phương tiện tổ chức để tạo ra một vùng an toàn để cất giữ khoá, sử dụng khoá và thực thi thuật toán mật mã:
  + Khoá phải được lưu trữ trong một thiết bị lưu trữ an toàn tách biệt (được gọi là vật mang khoá như HSM, thẻ Token hay thẻ thông minh) để đảm bảo khoá được lưu trữ và tương tác với các thuật toán mật mã an toàn. Trong trường hợp vật mang khoá là mô-đun an toàn phần cứng HSM, yêu cầu an toàn tối thiểu mức 3, trường hợp vật mang khóa là thẻ Token hay thẻ thông minh, yêu cầu an toàn tối thiểu mức 2 (mức an toàn được quy định tại TCVN 11295:2016 (ISO/IEC 19790:2012) Công nghệ thông tin – Các kỹ thuật an toàn – Yêu cầu an toàn cho mô-đun mật mã).
  + Trường hợp khoá lưu trữ ở dạng mềm cần sử dụng kỹ thuật mật mã để bảo vệ (Tuân thủ theo các quy định tại QCVN 4 : 2016/BQP và QCVN 5 : 2016/BQP).
  + Trường hợp hệ thống khoá được tổ chức theo cấu trúc phân cấp, các khoá cùng tầng chỉ được dùng để bảo vệ khoá ở tầng kế tiếp. Khoá chủ, tức khoá ở tầng cao nhất, phải được bảo vệ bằng cách chia ra nhiều thành phần và mỗi thành phần được bảo vệ riêng. Biện pháp này nhằm chống nguy cơ khoá bị lộ từ bên trong nội bộ.
- Việc huỷ khoá phải đảm bảo huỷ bỏ được tất cả các bản ghi khoá, bao gồm cả các bản sao dự phòng, sao cho không thể khôi phục được bất kì thông tin khoá nào dù bằng bất kì phương tiện nào. Đồng thời không để bất kì khoá nào bị huỷ bỏ trước thời hạn nhằm đảm bảo khả năng giải mã khi dữ liệu hết thời hạn bảo mật.
- Nếu khoá bị tổn thương hoặc bị nghi ngờ tổn thương hoặc có sự thay đổi chủ sở hữu thì phải thực hiện thu hồi khoá để vô hiệu hoá nguy cơ bị lộ hoặc mất an toàn.
- Thiết bị lưu trữ khoá phải được quản lý bởi người sử dụng. Người sử dụng có trách nhiệm bảo quản khoá ở vị trí an toàn, chống tiếp cận trái phép, chống sao chép hoặc bị đánh cắp. Không cung cấp khóa cho người không có thẩm quyền.

4. TRÁCH NHIỆM CỦA TỔ CHỨC, CÁ NHÂN

4.1. Các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) sử dụng sản phẩm, dịch vụ mật mã dân sự có trách nhiệm đảm bảo tuân thủ Quy chuẩn này và chịu sự kiểm tra của cơ quan quản lý nhà nước theo quy định.

4.2. Doanh nghiệp cung cấp sản phẩm, dịch vụ mật mã dân sự cho các tổ chức tín dụng (trừ quỹ tín dụng nhân dân cơ sở có tài sản dưới 10 tỷ, tổ chức tài chính vi mô) có trách nhiệm thực hiện công bố hợp quy sản phẩm, dịch vụ mật mã dân sự phù hợp với Quy chuẩn này. Việc công bố hợp quy thực hiện theo Thông tư số 28/2012/TT-BKHCN ngày 12 tháng 12 năm 2012 của Bộ Khoa học và Công nghệ.

4.3. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã – Ban Cơ yếu Chính phủ có trách nhiệm tiếp nhận đăng ký công bố hợp quy, thực hiện quản lý, hướng dẫn và kiểm tra việc công bố hợp quy.

5. TỔ CHỨC THỰC HIỆN

5.1. Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã – Ban Cơ yếu Chính phủ có trách nhiệm hướng dẫn, tổ chức triển khai quản lý kỹ thuật mật mã của Quản lý khóa theo Quy chuẩn này.

5.2. Trong trường hợp các quy định nêu tại Quy chuẩn kỹ thuật quốc gia này có sự thay đổi, bổ sung hoặc được thay thế thì thực hiện theo quy định tại văn bản mới./.