UPDATING...

I. Toán học mật mã

1. Một vài tính chất

Properties of Congruences

Congruences have the following properties:

1. $a \equiv b \pmod{n}$ if $n|(a - b)$.
2. $a \equiv b \pmod{n}$ implies $b \equiv a \pmod{n}$.
3. $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$ imply $a \equiv c \pmod{n}$.

To find $11^7 \bmod 13$, we can proceed as follows:

$11^2 = 121 \equiv 4 \pmod{13}$
$11^4 = (11^2)^2 \equiv 4^2 \equiv 3 \pmod{13}$
$11^7 = 11 \times 11^2 \times 11^4$
$11^7 \equiv 11 \times 4 \times 3 \equiv 132 \equiv 2 \pmod{13}$

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$
3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

An alternative form of Fermat's theorem is also useful: If p is prime and a is a positive integer, then

$$a^p \equiv a \pmod{p} \qquad \textbf{(2.11)}$$

Note that the first form of the theorem [Equation (2.10)] requires that a be relatively prime to p, but this form does not.

$p = 5, a = 3$	$a^p = 3^5 = 243 \equiv 3 \pmod{5} = a \pmod{p}$
$p = 5, a = 10$	$a^p = 10^5 = 100000 \equiv 10 \pmod{5} \equiv 0 \pmod{5} = a \pmod{p}$

1.1. Hàm Euler phi

- Nếu p là số nguyên tố: $\phi(p) = p - 1$
- Giả sử số $n = pq$, trong đó p#q là 2 số nguyên tố: $\phi(n) = \phi(p) * \phi(q) = (p - 1)(q - 1)$
- Một cách tính khác: $\phi(n) = n * \left(1 - \frac{1}{p}\right)\left(1 - \frac{1}{q}\right)$. *với p q* là 2 số nguyên tố khi thực hiện phân tích
- **Hàm euler PHI này cũng được dùng để tính tổng số cặp nghịch đảo nhân trong tập Z*n**

1.2. Euler's Theorem

Euler's theorem states that for every a and n that are relatively prime:

- $$a^{\phi(n)} \equiv 1 \pmod{n}$$
- **Now rewrite the Fermat theorem with this Euler form**

$$a^{\phi(n)+1} \equiv a \pmod{n}$$

1.3. Discrete logarithms

To see this last point, consider the powers of 7, modulo 19:

$$\begin{aligned}
7^1 &\equiv && 7 \pmod{19} \\
7^2 &= 49 = 2 \times 19 + 11 && \equiv 11 \pmod{19} \\
7^3 &= 343 = 18 \times 19 + 1 && \equiv 1 \pmod{19} \\
7^4 &= 2401 = 126 \times 19 + 7 && \equiv 7 \pmod{19} \\
7^5 &= 16807 = 884 \times 19 + 11 && \equiv 11 \pmod{19}
\end{aligned}$$

There is no point in continuing because the sequence is repeating. This can be proven by noting that $7^3 \equiv 1 \pmod{19}$, and therefore, $7^{3+j} \equiv 7^3 7^j \equiv 7^j \pmod{19}$, and hence, any two powers of 7 whose exponents differ by 3 (or a multiple of 3) are congruent to each other (mod 19). In other words, the sequence is periodic, and the length of the period is the smallest positive exponent m such that $7^m \equiv 1 \pmod{19}$.

-

2. Trường GF(2^n)

2.1. Đa thức tối giản

- **Chú ý:** Khi thực hiện nhân trên trường GF(2^n). Nếu bậc của kết quả **lớn hơn** n-1 thì phải thực hiện giảm bậc đa thức bằng cách **CHIA LẤY DƯ** cho đa thức tối giản

Bậc	Đa thức tối giản
1	$(x+1), (x)$
2	(x^2+x+1)
3	$(x^3+x^2+1), (x^3+x+1)$
4	$(x^4+x+1), (x^4+x^3+x^2+x+1), (x^4+x^3+1)$
5	$(x^5+x^2+1), (x^5+x^3+x^2+x+1), (x^3+x^2+1), (x^5+x^3+1),$ $(x^5+x^4+x^3+x+1), (x^5+x^4+x^3+x^2+x+1), (x^5+x^4+x^2+x+1)$
.....	
8	$(x^8+x^4+x^3+x+1)$

3. *Giải thuật Euclide mở rộng và ứng dụng*

3.1. Tìm UCLN

- Một vài tính chất:
  - $\gcd(a,b) = \gcd(|a|,|b|)$
  - $\gcd(a,b) = \gcd(b, a \bmod b)$
  - $$300 = 2^2 \times 3^1 \times 5^2$$ $$18 = 2^1 \times 3^2$$ $$\gcd(18,300) = 2^1 \times 3^1 \times 5^0 = 6$$
  -

Ý tưởng: luân phiên thực hiện a/b và a<->b đến khi nào không chia được nữa

Thực hiện:

Ví dụ: Tìm gcd(161,28)

q	r_1	r_2	r
5	161	28	21
1	28	21	7
3	21	7	0
	7	0	

=> GCD =7

- Tổng quát hóa lên ta mong muốn nhận được 2 số s và t sao cho $\gcd(a,b) = s*a + t*b$

$r_1 \leftarrow a; \quad r_2 \leftarrow b;$
$s_1 \leftarrow 1; \quad s_2 \leftarrow 0;$
$t_1 \leftarrow 0; \quad t_2 \leftarrow 1;$ (Initialization)

while $(r_2 > 0)$

{

$q \leftarrow r_1 / r_2;$

$r \leftarrow r_1 - q \times r_2;$
$r_1 \leftarrow r_2; \ r_2 \leftarrow r;$ (Updating r's)

$s \leftarrow s_1 - q \times s_2;$
$s_1 \leftarrow s_2; \ s_2 \leftarrow s;$ (Updating s's)

$t \leftarrow t_1 - q \times t_2;$
$t_1 \leftarrow t_2; \ t_2 \leftarrow t;$ (Updating t's)

}

gcd $(a, b) \leftarrow r_1; \ s \leftarrow s_1; \ t \leftarrow t_1$

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
	7	0		-1	4		6	-23	

3.2. Tìm nghịch đảo nhân
- Điều kiện tồn tại nghịch đảo nhân gcd(a,b)=1
- Thực hiện tương tự bên trên nhưng chỉ dùng biến t, không dùng biến s

II. Hệ mật mã cổ điển

1. Hệ mật thay thế đơn ký tự:
   a. Additive cipher (shift cipher hoặc Caesar cipher): mã cộng
   b. Multiplicative cipher: mã nhân
   c. Affine ciphers: Mã Affine
   d. Monoalphabetic substitution cipher:
2. Hệ mật thay thế đa ký tự:
   a. Playfair Cipher: mã Playfair
   b. Vigenere cipher
   c. Hill Cipher
   d. One-time pad
   e. Rotor cipher
   f. Enigma machine
3. Mã chuyển vị
   a. Chuyển vị khong khoá
   b. Chuyển vị có khoá
4. Mật mã dòng và mật mã khối
   a. Stream cipher: mật mã cộng, mã vignere
   b. Block cipher: mã playfair, mã hill
   -

III. Hệ mật mã hiện đại KHÓA ĐỐI XỨNG

1. Lý thuyết cơ bản

- Một mật mã khối hiện đại có thể được thiết kế để hoạt động như là một mật mã thay thế (Substitution, S_Box) hoặc một mật mã chuyển vị (Permutation, P_Box).
- Chúng ta có thể có n! khóa, khi đó mỗi khóa sẽ có độ dài $\lceil \log_2 n! \rceil$ bits
- Transposition: the key is $\lceil \log_2 n! \rceil$ bits long.

- Substitution: the key is $\lceil \log_2(2^n)! \rceil$ bits long
- Mã khoá khối hiện đại thường là các mật mã thay thế khoá, trong đó khóa có thể chỉ cho phép ánh xạ một phần từ đầu vào tới đầu ra
- Hộp P-box có tính khả nghịch, tuy nhiên hộp P boxes nén và mở rộng thì không
- Hộp S-box có thể có hoặc không thể đảo ngược được. Trong hộp S-box đảo ngược, số bit đầu vào bằng với số bit đầu ra.
- Thuật toán hoán đổi là một trường hợp đặc biệt của phép dịch chuyển vòng tròn khi k = n / 2 (n chẵn)

1.1.1. Mật mã tích

- Là một mật mã phức tạp kết hợp sự thay thế, hoán vị,...
- Hai đặc điểm quan trọng là diffusion (phân tán) và confusion (làm rối)
- Khuếch tán/ Phân tán **(Diffusion)**: **Sử dụng P_Box**
    - Ý tưởng của **khuếch tán** là để ẩn giấu mối quan hệ giữa **bản mã (Cipher)** và **bản rõ (Plain)**.
    - Tức là nếu một bit trong bản tin plaintext thay đổi thì nhiều hoặc tất cả bit trong ciphertext bị thay đổi
- Làm rối (**Confusion**): **Sử dụng S_Box**
    - Ý tưởng của sự **làm rối/ nhầm lẫn** là để giấu mối quan hệ giữa **bản mã (Cipher)** và **khóa (Key)**
    - Tức là, nếu một bit đơn trong khóa bị thay đổi thì tất cả các bit trong ciphertext cũng sẽ bị thay đổi

MẬT MÃ KHỐI HIỆN ĐẠI LÀ TẤT CẢ CÁC MẬT MÃ TÍCH, CHÚNG ĐƯỢC CHIA THÀNH 2 LỚP

1.1.2. Phân loại hệ mật tích

- **Mã Feistel:**
    - Ví dụ: Hệ mật DES (sử dụng các thành phần có tính khả nghịch)
    - Có 3 loại thành phần: Tự nghịch đảo, nghịch đảo, và không nghịch đảo (hàm f(R,K))
- **Mã Non-Feistel:**
    - Ví dụ: Hệ mật AES
    - **CHỈ SỬ DỤNG** các thành phần có thể nghịch đảo
    - Một thành phần trong mật mã hoá có thành phần tương ứng trong giải mật mã.

1.1.3. Hệ mật dòng đồng bộ

- Key là độc lập với Plain text và Cipher text

1.1.4. Hệ mật dòng không đồng bộ

- Key phụ thuộc vào trạng thái trước của plain text hoặc cipher text (Hãy liên tưởng đến mạch Logic tuần tự)

2. Hệ mật DES (1973)

2.1. Tổng quan

- Là hệ mật mã khối
- Gồm 16 vòng Feistel
- Quy trình mã hóa được tạo thành bởi 2 phép hoán vị P_Box cho khởi tạo và kết thúc. 2 P_Box này nghịch đảo nhau
- **Hai thuộc tính** mong muốn của **mật mã khối: Hiệu ứng thác lũ** và **Tính đầy đủ**
- **Hiệu ứng hoàn thiện:** Mỗi bit trên Cipher phụ thuộc vào nhiều bit trên Plain
- **S_Box:**
    - Thiết kế cung cấp sự **nhầm lẫn và khuếch tán** các bit từ **mỗi vòng sang vòng tiếp theo**
    - Hộp S là phi tuyến
    - Các giá trị của mỗi hàng được hoán vị từ 0 đến 15
    - Nếu thay đổi 1 bit vào, 2 or more bit ra bị thay đổi
    - Nếu 2 bit đầu vào hộp S khác hai bit nào đó ở giữa, thì đầu ra của hộp S sẽ khác ít nhất 2 bit
    - Trong mỗi hộp S, nếu một bit đơn cố định (0, hoặc 1) còn các bit khác thay đổi ngẫu nhiên, sự khác nhau giữa các bit 0 và 1 là nhỏ nhất
- **P_Box:**
    - Cung cấp sự khuếch tán của các bit
    - DES có:
        - 1 hộp thẳng (Straight Permutation) (32->32)
        - 1 hộp mở rộng (32->48)
- **Điểm yếu:**
    - Weaknesses in S-boxes: Hộp S4, đầu ra các hộp S có thể giống nhau.

- Weaknesses in P-boxes: Không rõ ràng việc sử dụng hộp P khởi tạo và kết thúc. Nó không đủ khả năng bảo mật.
- Weaknesses in Key: Có nhiều điểm yếu khi mỗi vòng dùng khóa 56 bit →kiểmtra số lượng khóa 2^56.

- **Mở rộng:** Hệ mật double DES, Triple DES với số lượng bit trong khóa gấp 2 gấp 3

2.2. Thuật toán và các bước mã hóa

2.2.1. Sinh khóa

B1: Đầu vào (64-bit)

B2: Parity drop (còn 56-bit) Bỏ 8 bit (8, 16, 24, 32, 40, 48, 56, 64)

B2: Split (28Left - 28Right)

B3: Shift left each part Left and Right. Then Combine them

B4: Compression D_Box (48-bit) -> This is Round Key

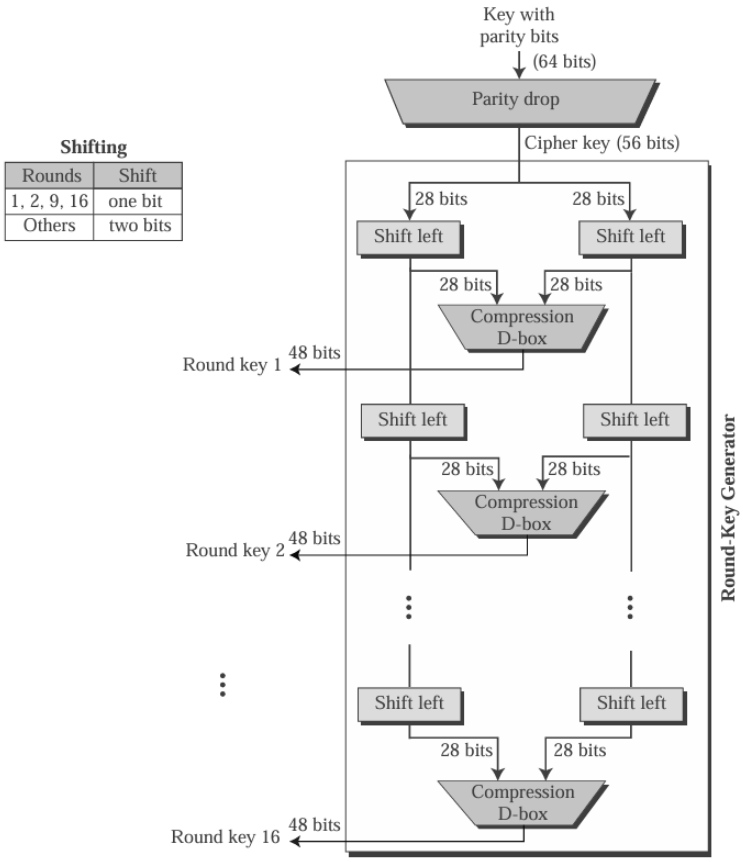
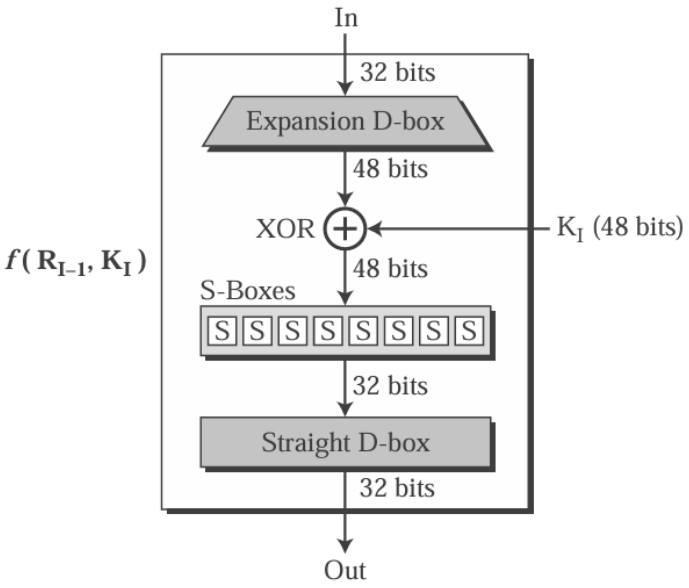


Fig. 6.10 *Key generation*

2.2.2. Mã hóa



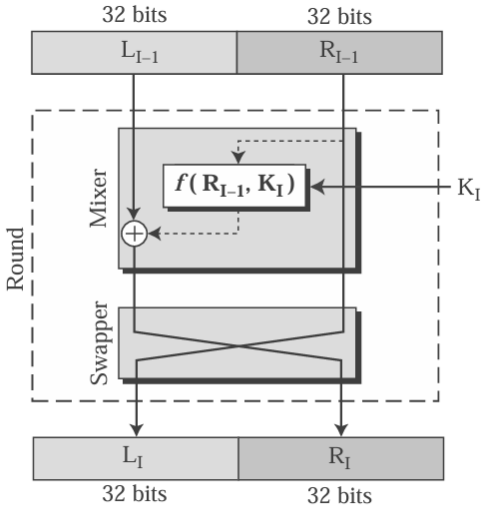


Fig. 6.4 *A round in DES (encryption site)*

- **B1:** Chia Left (32-bit), Right(32-bit) của Plain text - **B2:** Phần Right đưa qua hàm $f(R_{I-1}, K_I)$ - **B3:** Mix với phần Left của Plaintext - **B4:** Phần Right vòng trước là phần Left vòng sau	- **B1:** Chia Left (32-bit) và Right(32-bit) của Plain text - **B2:** Phần Right đưa qua hàm $f(R_{I-1}, K_I)$ - **B3:** Mix với phần Left của Plaintext - **B4:** Phần Right vòng trước là phần Left vòng sau

3. Hệ mật AES (Rijdael) 2001

3.1. Tổng quan

- Là mật mã khối
- Sử dụng 128-bit key và 128-bit plain text
- Là mã Non-Feistel

Kích thước khóa	Số vòng	Số từ mỗi vòng
128-bit	10	4
192-bit	12	6
256-bit	14	8

- Sử dụng **4 kiểu biến đổi**: Thay thế(Substitution), hoán vị (Permuation), trộn (Mixing) và thêm khóa (Key Adding)
- AES định nghĩa phép biến đổi đại số bằng cách sử dụng trường GF(2^8) với đa thức tối giản:

$$x^8 + x^4 + x^3 + x + 1$$

- **Mixing:**
  - Chúng ta cần một phép biến đổi interbyte làm thay đổi các bit bên trong một byte, dựa trên các bit bên trong các byte lân cận
  - Chúng ta cần trộn các byte để cung cấp sự khuếch tán các bit4
- **Add Round Key:**
  - AddRoundKey tiến hành từng cột một.
  - AddRoundKey thêm một từ khóa tròn với mỗi ma trận cột trạng thái;
  - Hoạt động trong AddRoundKey là phép cộng ma trận.
  - Phép tính cho mỗi phần tử là phép XOR
- **RotWord (SubWord):**
  - RotWord (rotate word) giống với quá trình dịch chuyển hàng ShiftRows, nhưng chỉ dịch chuyển một hàng.
  - SubWord (Substitute word) giống với quá trình chuyển đổi SubBytes, nhưng chỉ áp dụng cho 4 bytes. Mỗi byte trong word được thay bởi 1 byte khác
- Mỗi khóa vòng trong AES phụ thuộc vào khóa trước đó. Tuy nhiên, sự phụ thuộc là phi tuyến vì biến đổi SubWord. Việc bổ sung hằng số vòng (RCON) cũng đảm bảo mỗi khóa vòng sẽ khác với khóa trước đó
- **Round Constant (RCON):**
  - Là giá trị 4 bytes mà 3 bytes luôn luôn phải ở 0

3.2. Thuật toán và các bước thực hiện

3.2.1. Sinh các Round Key

- **B1:** Chuyển các ký tự ở Round Key sang hệ hexa (đây là Round Key 0) và ghi theo cột vào ma trận
    - Nhóm từng 4 bit để tạo thành 1 vector w

  Ví dụ: 54 68 61 74 **73 20 6D 79** 20 4B 75 6E **67 20 46 75**
    - W[0] = [54, 68, 61, 74]....

- **B2:** W[3] = [67, 20, 46, 75]
    - Dịch trái W[3] 1 bit =>>>>> W[3] = [20, 46, 75, 67]
    - Đưa qua S_box =>>>>>>>> W[3] = [B7, 5A, 9D, 85]
    - Adding RCON =>>>>>>>>> g(W[3]) = [B6, 5A, 9D, 85]

Table 7.4 *RCon constants*

Round	*Constant (RCon)*	*Round*	*Constant (RCon)*
1	$(\underline{\mathbf{01}}\ 00\ 00\ 00)_{16}$	6	$(\underline{\mathbf{20}}\ 00\ 00\ 00)_{16}$
2	$(\underline{\mathbf{02}}\ 00\ 00\ 00)_{16}$	7	$(\underline{\mathbf{40}}\ 00\ 00\ 00)_{16}$
3	$(\underline{\mathbf{04}}\ 00\ 00\ 00)_{16}$	8	$(\underline{\mathbf{80}}\ 00\ 00\ 00)_{16}$
4	$(\underline{\mathbf{08}}\ 00\ 00\ 00)_{16}$	9	$(\underline{\mathbf{1B}}\ 00\ 00\ 00)_{16}$
5	$(\underline{\mathbf{10}}\ 00\ 00\ 00)_{16}$	10	$(\underline{\mathbf{36}}\ 00\ 00\ 00)_{16}$

- **B3: Tính toán Round key 1**
    - $w[4] = w[0]\ \boldsymbol{xor}\ g(w[3])$
    - $w[5] = w[1]\ \boldsymbol{xor}\ w[4]$
    - $w[6] = w[2]\ \boldsymbol{xor}\ w[5]$
    - $w[7] = w[3]\ \boldsymbol{xor}\ w[7]$
- **B4: Thực hiện tương tự để tìm Round key tiếp theo**

3.2.2. Thực hiện mã hóa

- **ROUND 0:**
    - **Add Round key 0**
- **ROUND 1:**
    - **SubBytes:** Dùng S_Box

$$\begin{pmatrix} 00 & 3C & 6E & 47 \\ 1F & 4E & 22 & 74 \\ 0E & 08 & 1B & 31 \\ 54 & 59 & 0B & 1A \end{pmatrix} \begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix}$$

    - **Shift Row:** Dịch trái dòng n: n lần (n = 0,1,2,3)

$$\begin{pmatrix} 63 & EB & 9F & A0 \\ C0 & 2F & 93 & 92 \\ AB & 30 & AF & C7 \\ 20 & CB & 2B & A2 \end{pmatrix} \quad \begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix}$$

    - **Mix Column:**

• Mix Column multiplies fixed matrix against current State Matrix:

$$\begin{pmatrix} 02\,03\,01\,01 \\ 01\,02\,03\,01 \\ 01\,01\,02\,03 \\ 03\,01\,01\,02 \end{pmatrix} \begin{pmatrix} 63 & EB & 9F & A0 \\ 2F & 93 & 92 & C0 \\ AF & C7 & AB & 30 \\ A2 & 20 & CB & 2B \end{pmatrix} = \begin{pmatrix} BA & 84 & E8 & 1B \\ 75 & A4 & 8D & 40 \\ F4 & 8D & 06 & 7D \\ 7A & 32 & 0E & 5D \end{pmatrix}$$

• entry BA is result of $(02 \bullet 63) \oplus (03 \bullet 2F) \oplus (01 \bullet AF) \oplus (01 \bullet A2)$:

- **Chú ý nhận ma trận:** Hàng m * Cột n -> Phần tử thứ (m*n)
- Bấm máy tính nhân như bình thường
- Trong trường hợp 02h*AAh = 154h -> TRÀN ➔➔➔➔ $[(02h * AAh)\boldsymbol{xor}\ 100h]\boldsymbol{xor}\ 1Bh = 4Fh$
- Trong trường hợp 03 * XX thì tách thành (02*XX)+XX. Thực hiện tính 02*XX như bên trên

  - **Add Round Key 1**

- **Thực hiện tương tự với các Round tiếp theo**
- **Round cuối (Round 10)** (không phải Round 9 nhé)
  - **CHÚ Ý:** Round 10 không có Mix Columns

IV. Hệ mật dòng (KHÓA ĐỐI XỨNG)

In stream cipher, one byte is encrypted at a time while in block cipher N bits are encrypted at a time

1. Phân loại hệ mật dòng

- **Hệ mật dòng đồng bộ (Synchronous):** Sinh key độc lập với các Cipher text và Plain text trước (KAK)
- **Self-synchronizing stream ciphers:** rely on previous ciphertext bits to generate keystreams (CTAK)

2. Hệ mật RC4

2.1. Tổng quan

- For RC4, stream combinations are done on byte-length strings of plaintext. 256 bytes of memory are required for the state array.

V. Hệ mật mã KHÓA BẤT ĐỐI XỨNG

1. Hệ mật RSA

- RSA sử dụng lũy thừa mô-đun để mã hóa/ giải mã hóa; Để tấn công được, Eve cần tính $^{e}\sqrt{C} \bmod n$
- Cặp khóa (n,e) public key dùng để mật mã hóa
- Cặp khóa (n,d) private key dùng để giải mật mã hóa

Key Generation:

- RSA relies on the difficulty of **factoring large integers**.
- The public key consists of two numbers:
  - A product of two large prime numbers (usually denoted as `n`).
  - A small exponent (usually denoted as `e`).
- The private key is derived from the same two prime numbers.

- RSA dựa trên việc khó khăn khi phân tích số nguyên lớn
- n là tích của 2 số nguyên tố lớn, e là số nguyên tố, d là nghịch đảo nhân của e trong tập $\phi(n)$

2. Hệ mật KnapSack

- **Private key:** Là chuỗi siêu tăng
- **Số n:** Lớn hơn tổng của chuỗi siêu tăng
- **Số r:** Đồng dư với n
  - **Kinh nghiệm:**
    - Chọn r và n là 2 số nguyên tố
    - Hoặc chọn r là số nguyên tố và n không phải là bội của n
- **Public key:** Public Key = Private Key * r (mod n)
- **Chú ý:** Nếu đề cho thêm Permutation, thì ta phải đổi vị trí các phần tử trong Public key thì mới ra được Public key
- **Mã hóa** (Plain text <P>, Cipher text <C>):
  - Chia dòng bit ra thành từng khối. Độ dài mỗi khối bằng tổng số phần tử của tập Private key

- Nhân từng phần tử trong khối với từng phần tử trong **Public key**

```
100100  {31, 62, 14, 90, 70, 30}

1x31+0x62+0x14+1x90+0x70+0x30 = 121
```

- **Giải mật:**
  - Tìm nghịch đảo nhân của **r** trong **Zn**
  - $\boldsymbol{P = C * r^{-1}\ (mod\ n)}$
  - Ví dụ: $r^{-1} = 71, C = 121 => P = 11$
  - Sau khi tìm được **P** thực hiện tách **P** ra thành tổng các phần tử trong **Private key**
  - Ví dụ: **Private key** = {1, 2, 4, 10, 20, 40} => P = 11 = 1*1 + 0*2 + 0*4 + 1*10 + 0*20 + 0*40 ➔ C = 100100

3. Hệ mật Rabin

- **Ý tưởng:** Recovering the entire plaintext from the ciphertext could be proven to be as hard as factoring

3.1. Sinh khóa

- Chọn p và q là 2 số nguyên tố lớn, thường chọn $p \equiv q \equiv 3\ (mod\ 4)$ để dễ dàng tính toán <đi thi đề cho>
- **Private key:** p và q
- **Public key:** n = p.q

3.2. Mã hóa

- $\boldsymbol{C = m^2\ (mod\ n)}$
- **m :** là một phần tử của tập Plain text
- **Tập Plain text: P** = {0, 1, ..., n-1}

3.3. Giải mật mã

- **B1: Tính:** $\begin{cases} m_p = \sqrt{c}\ (mod\ p) \\ m_q = \sqrt{c}\ (mod\ q) \end{cases}$. Theo phương pháp sau:
  - **Nếu:** chọn $p \equiv q \equiv 3\ (mod\ 4)$
  - **Thì:** $\begin{cases} m_p = c^{\frac{p+1}{4}}\ (mod\ p) \\ m_q = c^{\frac{p+1}{4}}\ (mod\ q) \end{cases}$
- **B2: Tính**: $y_p\ và\ y_q$. Sao cho $y_p * p + x_q * q = 1$.
  - Tức là dùng giải thuật Euclide để tìm $y_p\ và\ y_q$. Thỏa mãn: $y_p * p + x_q * q = \gcd(p, q) = 1$
  - Tính toán dùng giải thuật Euclide mở rộng đã trình bày bên trên
- **B3: Tính:** Tập các phần tử là plain text

$$\begin{aligned} r &= (y_p \cdot p \cdot m_q + y_q \cdot q \cdot m_p) \bmod n \\ -r &= n - r \\ s &= (y_p \cdot p \cdot m_q - y_q \cdot q \cdot m_p) \bmod n \\ -s &= n - s \end{aligned}$$

  - **Để đơn giản hóa:** (-a) mod n **~** a mod n. Gtri còn lại tính theo: n – (a mod n)
  - Tập các phần tử trên là Plain text

4. Hệ mật Elgmal

4.1. Nhắc lại kiến thức về Primitive root

Let's consider the set $\mathbb{Z}_n^*$ of all numbers that have multiplicative inverses mod n. As we mentioned previously, these numbers in the range from 0 to $n-1$ are relatively prime to n. Therefore, as we already know, $\mathbb{Z}_{10}^* = \{1, 3, 5, 7\}$. Moreover, if n is a prime p, then $\mathbb{Z}_p^* = \{1, 2, \ldots, p-1\}$ since all numbers from 1 to $p-1$ are relatively prime to p.

Also, if there is a number $\alpha \in \mathbb{Z}_p^*$ such that the set $\{\alpha \bmod p, \alpha^2 \bmod p, \ldots \alpha^{p-1} \bmod p\} = \mathbb{Z}_p^*$, then α is called a primitive root of p. Let's assume $p = 7$, the table below shows that only 3 and 5 are its primitive roots.

a	a^1	a^2	a^3	a^4	a^5	a^6	
1	1	1	1	1	1	1	
2	2	4	1	2	4	1	
3	3	2	6	4	5	1	All distinct values from 1 to 6
4	4	2	1	4	2	1	
5	5	4	6	2	3	1	All distinct values from 1 to 6
6	6	1	6	1	6	1	

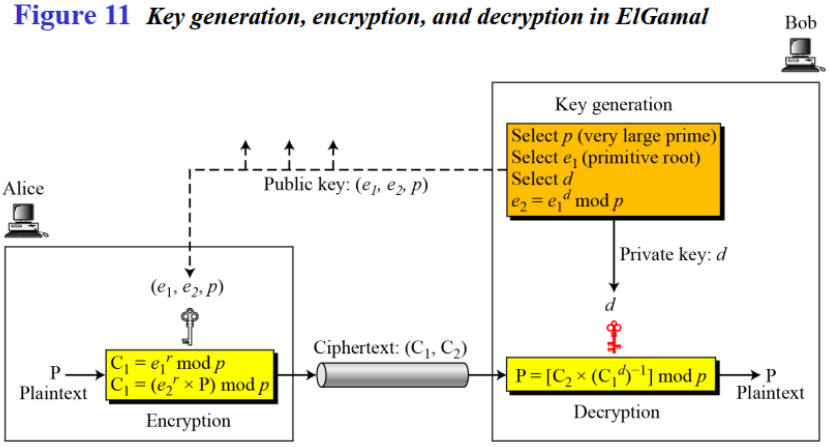
If α is a primitive root of p, then, for any $1 \leq b \leq n-1$, $\log_\alpha b \bmod p$ is a distinct value. This shows in the table above: $\log_5(2) \bmod 7 = 4$ (a unique value). However, $\log_4(2) \bmod 7 = 2$ or 5. This follows from the definition of primitive roots.

-

4.2. Bảng Primitive root

4.3. Thuật toán mã hóa

Figure 11 *Key generation, encryption, and decryption in ElGamal*



VI. Hàm băm mật mã

1. Tổng quan

- Hashing là quá trình biến một dữ liệu đầu vào có độ dài bất kỳ thành một chuỗi đầu ra đặc trưng có độ dài cố định. Hashing được thực hiện thông qua hàm băm (hash function).
- **Ứng dụng:** đảm bảo tính toàn vẹn dữ liệu
- **INPUT:** độ dài tùy ý
- **OUTPUT:** độ dài cố định
- **Hai nhóm hàm nén:**
  - Chức năng nén được thực hiện từ đầu: Tóm lược thông điệp Message Digest (MD)
    - MD5: chia bản tin thành khối 512-bit để tạo thành 1 digest 128-bit
    - SHA (secure hash function): dựa trên cấu trúc MD5

      Table 12.1 *Characteristics of Secure Hash Algorithms (SHAs)*

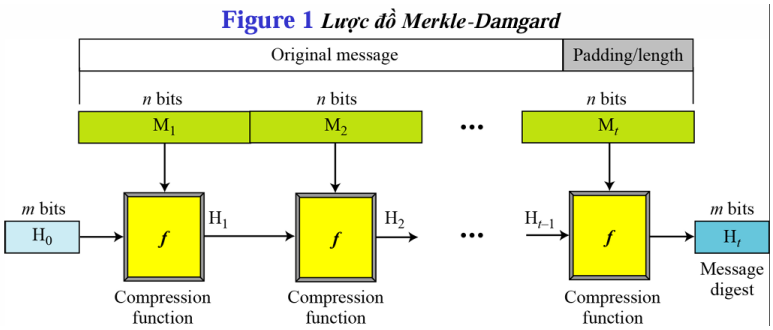
Characteristics	*SHA-1*	*SHA-224*	*SHA-256*	*SHA-384*	*SHA-512*
Maximum Message size	$2^{64} - 1$	$2^{64} - 1$	$2^{64} - 1$	$2^{128} - 1$	$2^{128} - 1$
Block size	512	512	512	1024	1024
Message digest size	160	224	256	384	512
Number of rounds	80	64	64	80	80
Word size	32	32	32	64	64

    -
  - Mật mã khóa đối xứng đóng vai trò như một hàm nén: Xoáy nước Whirlpool

2. Các lược đồ

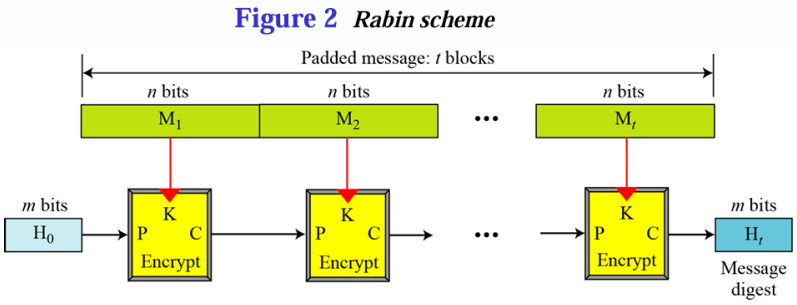
2.1. Lược đồ Merkel-Damgard

Figure 1 *Lược đồ Merkle-Damgard*



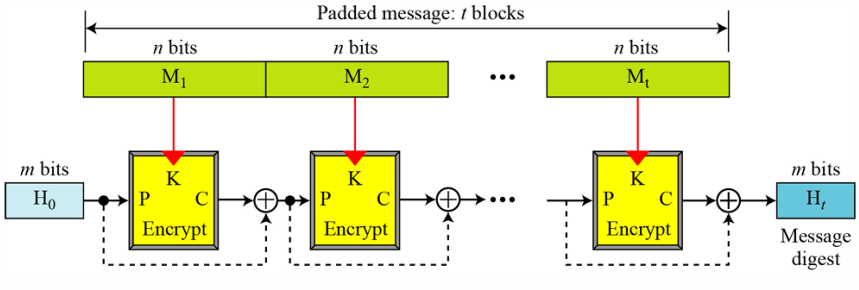
2.2. Rabin Scheme: Lược đồ Rabin

Figure 2 *Rabin scheme*



- Dựa trên lược đồ Merkle-Damgard
- Hàm nén được thay thế bằng một hệ mật nào đó. Trong đó:
  - Các khối bản tin ⬅➡ Khóa K
  - Input ⬅➡ Plain text
  - Output / Digest ⬅➡ Cipher text
  - Kích thước của digest bằng kích thước của mã khối dữ liệu. Ví dụ: trong hệ DES thì digest = 64bit

2.3. Lược đồ Davies – Meyer



-
- Thêm C = P xor C sau mỗi hàm băm nhằm tăng độ bảo mật, cải thiện việc bị tấn công vào quá trình trung gian

2.4. Lược đồ Matyas-Meyer-Oseas

- Hình xem trong slide (Khác biệt)
- Cout = C xor K
- Là phiên bản kép của Davies-Meyer
- các khối bản tin sử dụng cho khóa mật. Lược đồ này sử dụng tốt nhất khi khóa mật và khối bản tin cùng kích thước.
- **Khi đó hệ AES được sử dụng là tốt nhất**

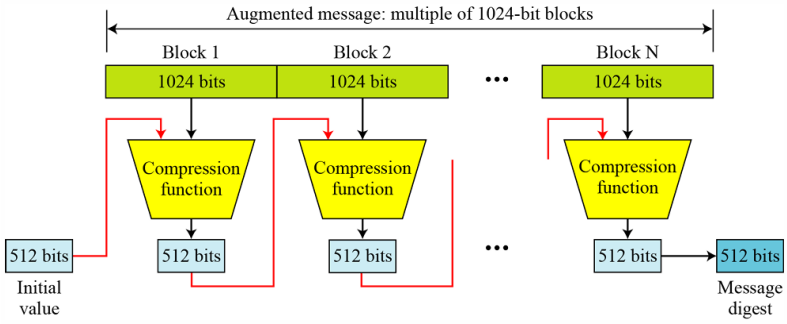
2.5. Lược đồ Miyaguchi-Preneel

- Mở rộng của Matyas-Meyer-Oseas
- Cout = C xor P xor K

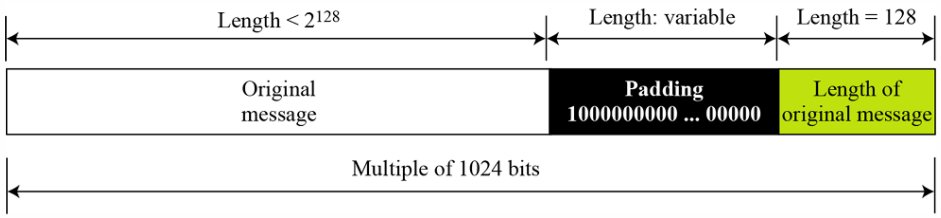
3. SHA-512

- **Đầu ra**: 512-bit
- **Dựa trên lược đồ**: Merkle-Damgard

Figure 6 *Message digest creation SHA-512*



- **SHA-512** tạo bản tóm tắt tin nhắn **512 bit** từ một bản tin **nhỏ hơn 2^128 bit**



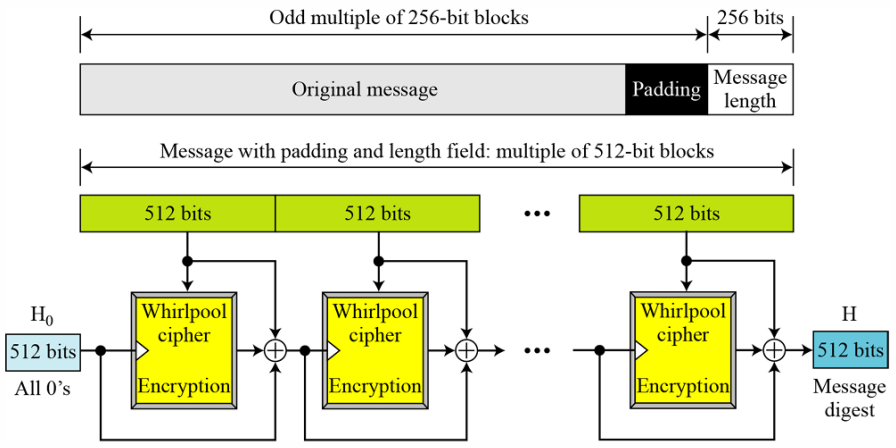
- **Phần Original message (M):** độ dài nhỏ hơn 2^128 bit
- **Phần Length of Original message**: độ dài cố định 128 bit
- **Phần Padding (P):** được thêm vào để đủ độ dài là bội của 1024
- **Độ dài phần Padding** tính bởi: $\boldsymbol{P = (-M - 128)\ mod\ 1024}$
- **Độ dài tối thiểu của Padding:** 0 xảy ra khi |M| = 896 mod 1024 = -128 mod 1024
- **Số bit đệm tối thiểu phải thêm:** 1 xảy ra khi |M| = 897 mode 1024

3.1. Biểu diễn dưới dạng Words

- 1 word = 64 bits
- **Input:** bội của 16 words
- **Output:** 8 words

4. WHIRLPOOL

- Whirlpool là một hàm băm mật mã được lặp đi lặplại, dựa trên lược đồ Miyaguchi-Preneel
- Sử dụng mật mã khối khóa đối xứng thay cho hàm nén
- Mật mã khối là một mật mã AES đã được sửa đổi đã được điều chỉnh cho mục đích này



- Whirlpool dựa trên hệ mật kiểu non Feistel giống như AES để thực hiện phép mã hóa băm. Whirlpool gồm 10 round, mỗi khối có kích thước khối và khóa là 512 bits được đánh số từ K0 đến K10
- Giống như AES, Whirlpool sử dụng các block và state
- Một block xem như là 1 ma trận hàng 64 bytes;
- Một state là ma trận vuông kích thước 8×8 bytes
- Tuy nhiên, không giống như AES, việc viết block vào state là theo hang
- **Mỗi vòng sử dụng bốn phép biến đổi.** ➔ SubBytes cung cấp phép biến đổi phi tuyến

Table 12.5 *Main characteristics of the Whirlpool cipher*

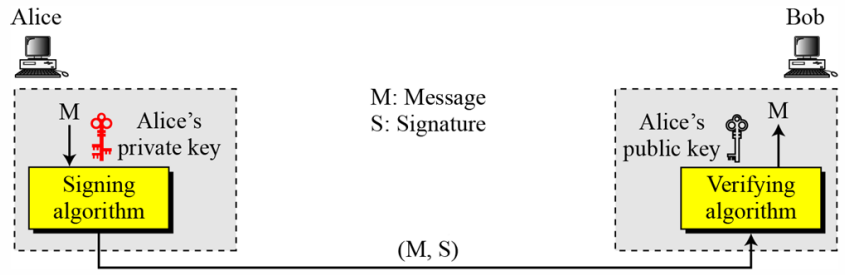
Block size: 512 bits
Cipher key size: 512 bits
Number of rounds: 10
Key expansion: using the cipher itself with round constants as round keys
Substitution: SubBytes transformation
Permutation: ShiftColumns transformation
Mixing: MixRows transformation
Round Constant: cubic roots of the first eighty prime numbers

VII. Chữ ký số

1. Tổng quan

- Đối với chữ ký điện tử, người nhận sẽ nhận được thông điệp và chữ ký. Người nhận cần áp dụng kỹ thuật xác minh kết hợp giữa bản tin và chữ ký để xác minh tính xác thực
- Đối với **chữ ký thông thường**, thường có mối **quan hệ một-nhiều** giữa chữ ký và các tài liệu.
- Đối với **chữ ký điện tử**, có mối **quan hệ 1-1** giữa chữ ký và bản tin

Figure 2 *Adding key to the digital signature process*



- Một chữ ký điện tử cần một hệ thống khóa công khai
- Người **ký bằng khóa riêng** của cô ấy; người xác minh **xác minh bằng khóa công khai của người ký**
- Phân biệt **KHÓA BÍ MẬT** và **KHÓA CÔNG KHAI** trong **chữ ký số** và trong **hệ mật mã khóa bất đối xứng**
    - **Hệ mật mã khóa bất đối xứng:** khóa bí mật và công khai **sinh bởi bên người nhận**. Khóa **công khai** được **sử dụng cho phép mã hóa**, **khóa bí mật** được sử dụng để **giải mã hóa**.
    - **Chữ ký số:** khóa bí mật và công khai do **bên người gửi sử dụng**, bên **gửi sử dụng khóa bí mật**, bên **nhận sử dụng khóa công khai** của bên gửi
- **ỨNG DỤNG:**
    - Xác thực bản tin
    - Tính toàn vẹn bản tin
    - Không bác bỏ
    - Bí mật
- Chữ ký điện tử không cung cấp quyền riêng tư. Nếu có nhu cầu về quyền riêng tư, một lớp mã hóa / giải mã khác phải được áp dụng
- Có **03** loại tấn công:
    - Tấn công chỉ vào khóa
    - Tấn công bản tin đã biết
    - Tấn công bản tin được chọn
- Có **02** loại giả mạo:
    - Giả mạo tồn tại
    - Giả mạo có chọn lọc

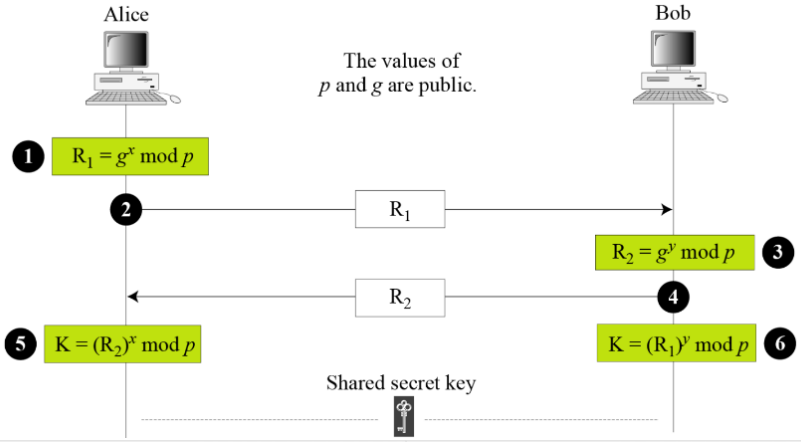
2. Các lược đồ chữ ký điện tử

- **Dựa trên RSA:** Khi Digest được ký thay vì chính thông điệp, tính nhạy cảm của lược đồ chữ ký số RSA phụ thuộc vào độ mạnh của thuật toán băm
- **Dựa trên Elgamal:**
- **Dựa trên Schnorr:** Dựa trên lược đồ Elgamal
- **DSS (Digital Signature Standard)**
- **Elliptic Curve**

2.1. So sánh

- Tính toán chữ ký DSS nhanh hơn tính toán chữ ký RSA khi sử dụng cùng một p
- Chữ ký DSS nhỏhơn chữkýElGamal vì q nhỏ hơn p.

VIII. Trao đổi khóa bằng thuật toán DHKE (Diffie- Hellman Key Exchange)



Alice	Bob
Public Keys available = P, G	Public Keys available = P, G
Private Key Selected = a	Private Key Selected = b
Key generated = $x = G^a modP$	Key generated = $y = G^b modP$
Exchange of generated keys takes place	
Key received = y	key received = x
Generated Secret Key = $k_a = y^a modP$	Generated Secret Key = $k_b = x^b modP$
Algebraically, it can be shown that $k_a = k_b$	
Users now have a symmetric secret key to encrypt	

- P là số nguyên tố được chọn, G là primitive root of P