# Nhập môn An toàn thông tin

PGS. Nguyễn Linh Giang

Bộ môn Truyền thông và Mạng máy tính

# Nội dung

I. Nhập môn An toàn thông tin
II. Đảm bảo tính mật
    I. Các hệ mật khóa đối xứng (mã hóa đối xứng)
    II. Các hệ mật khóa công khai ( mã hóa bất đối xứng )
III. Bài toán xác thực
    I. Cơ sở bài toán xác thực
    II. Xác thực thông điệp
    III. Chữ ký số và các giao thức xác thực
    IV. Các cơ chế xác thực trong các hệ phân tán
IV. An toàn an ninh hệ thống
    I. Phát hiện và ngăn chặn xâm nhập ( IDS, IPS )
    II. Lỗ hổng hệ thống

# Nội dung

- Tài liệu môn học:
  - W. Stallings “Networks and Internetwork security”
  - W. Stallings “Cryptography and network security”
  - Introduction to Cryptography – PGP
  - D. Stinson – Cryptography: Theory and Practice

# Chương II.
# Các phương pháp mật mã khóa đối xứng

1. Sơ đồ chung của phương pháp mật mã khóa đối xứng
2. Một số phương pháp mật mã khóa đối xứng kinh điển
3. Hệ mật hoàn hảo và không hoàn hảo
4. Phương pháp DES
5. Quản trị và phân phối khóa
6. Đảm bảo tính riêng tư sử dụng phương pháp mật mã khoá đối xứng

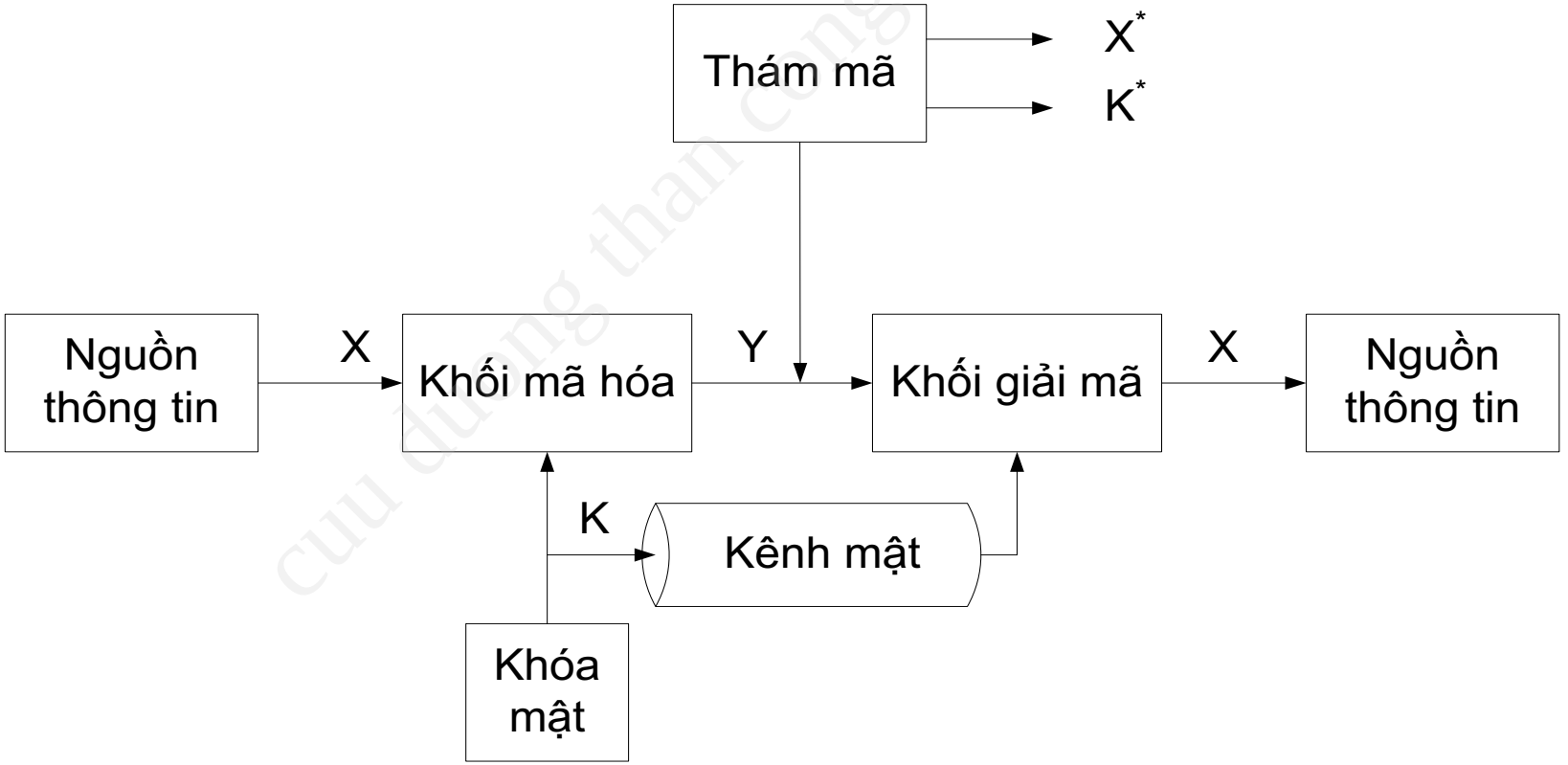
# Sơ đồ chung của phương pháp mã hóa đối xứng

- Sơ đồ mã hóa đối xứng
- Mật mã và thám mã

# Sơ đồ mật mã khóa đối xứng

- Một số thuộc tính của mô hình mật mã khóa đối xứng:
  - Thuật toán mã hóa phải đủ mạnh để không thể giải mã được thông điệp nếu chỉ dựa trên duy nhất nội dung của văn bản được mã hóa( ciphertext ).
  - Sự an toàn của phương pháp mã hóa đối xứng chỉ phụ thuộc vào độ bí mật của khóa mà không phụ thuộc vào độ bí mật của thuật toán.
- Phương pháp mật mã khóa đối xứng giả thiết rằng:
  - Thám mã không thực hiện được nếu chỉ biết thông điệp bị mã hóa và thuật toán mã hóa.
  - Không cần giữ bí mật thuật toán.
  - Chỉ cần giữ bí mật khóa.

# Sơ đồ mật mã khóa đối xứng




Mô hình hệ thống mã hóa đối xứng.

# Sơ đồ chung của phương pháp mật mã khóa đối xứng

- Nguồn thông tin:
  - Tập hợp thông điệp của nguồn:
    Các xâu ký tự $X = \{ X_1, X_2, ..., X_M \}$;
  - Thông điệp: xâu ký tự độ dài m:
    $X_i = [ x_{i1}, x_{i2}, ..., x_{im} ]$
    $x_{ik} \in A$; A – bảng ký tự nguồn; thông thường A= {0, 1}
  - Mỗi thông điệp $X_i$ có một xác suất xuất hiện $P( X = X_i )$ – thuộc tính thống kê của nguồn thông điệp:

# Sơ đồ chung của phương pháp mật mã khóa đối xứng

- Khóa mật mã
  - Tập hợp khoá K = { $K_1$, $K_2$, ... $K_L$},
  - Khóa độ dài l: $K_i=[k_{i1}, ..., k_{il}]$;

    $k_{ij} \in C$, C - bảng ký tự khóa; thông thường C = {0, 1}
  - Xác suất tạo khóa P{K=k} và phân bố xác suất tạo khóa.
  - Phân phối khóa giữa các bên trao đổi thông tin:
    - Phân phối khóa không tập trung: Nếu khóa K được tạo ra từ phía nguồn, khóa K cần được chuyển cho phía nhận tin thông qua một kênh bí mật .
    - Phân phối khóa tập trung: Khóa K do bên thứ ba được ủy quyền tạo ra và được phân phối cho cả hai phía gửi và nhận tin.

# Sơ đồ chung của phương pháp mật mã khóa đối xứng

- Mã mật:
  - Tập hợp thông điệp mã mật $Y = [ Y_1, Y_2, ..., Y_N ]$
  - Thông điệp mã mật: $Y_j = [y_{j1}, y_{j2}, ..., y_{jn}]$
  - $y_{jp} \in B$, B – bảng ký tự mã mật; thông thường B = {0, 1}

# Sơ đồ chung của phương pháp mật mã khóa đối xứng

- Quá trình mật mã và giải mã:
  - Quá trình mã hóa:

$$Y = E_K( X )$$

    - Để tăng thêm độ bất định của quá trình mã hóa, sử dụng số ngẫu nhiên R

$$Y = E_{K,R}( X )$$

  - Quá trình giải mã:
    - Bên nhận giải mã thông điệp bằng khóa được phân phối:

$$X = D_K( Y ) = D_K ( E_{K,R}( X ) )$$

# Sơ đồ chung của phương pháp mã hóa đối xứng

- Phía tấn công
  - Vấn đề đặt ra: đối phương nhận được thông điệp Y, nhưng không có được khóa K. Dựa vào thông điệp Y, đối phương phải khôi phục lại hoặc K, hoặc X hoặc cả hai.
    - Đối phương có thể chỉ cần khôi phục lại thông điệp X bằng thông điệp $X^*$.
    - Nếu đối phương muốn biết thêm các thông điệp trong tương lai: cần phải xác định được khóa K.

# Mật mã và thám mã

- Mật mã
  - Các tiêu chí phân loại hệ thống mật mã:
    - Dạng của phép toán tham gia vào mã hóa văn bản từ dạng thông thường sang dạng được mật mã hóa.
    - Phân loại các phương pháp mật mã theo số lượng khóa được dùng trong thuật toán;
    - Phân loại các phương pháp mật mã theo số lượng khóa được dùng trong thuật toán:

# Mật mã và thám mã

- Phân loại theo dạng của phép toán tham gia vào mã hóa văn bản từ dạng rõ sang dạng được mã.
  - Các phương pháp mã hóa thông thường này dựa vào các nguyên lý sau:
    - Phép thế: mỗi ký tự trong bản thông điệp sẽ được ánh xạ vào phần tử khác.
    - Phép hoán vị: các ký tự trong thông điệp ban đầu được phân bố lại.
    - Phép dịch;
    - Yêu cầu chính: không mất mát thông tin.

# Mật mã và thám mã

- Phân loại các phương pháp mật mã theo số lượng khóa được dùng trong thuật toán:
  - Nếu bên gửi và bên nhận cùng dùng chung một khóa: hệ thống mã hóa đối xứng.
  - Nếu hai khóa của bên gửi và bên nhận khác nhau: phương pháp mã hóa bất đối xứng.

# Mật mã và thám mã

- Phân loại các phương pháp mật mã theo số lượng khóa được dùng trong thuật toán:
  - Mã hóa khối ( block cipher ): bản rõ được xử lý theo từng khối thông tin và tạo đầu ra theo từng khối thông tin.
  - Mã hóa dòng ( stream cipher ): bản rõ được xử lý liên tục theo từng bit.

# Mật mã và thám mã

- Thám mã
  - Quá trình xác định nội dung bản rõ X hoặc khóa K hoặc cả hai từ bên thứ ba ( cryptanalyst ).
  - Chiến lược sử dụng phụ thuộc vào tính chất của sơ đồ mã hoá và những thông tin do thám mã nắm được.
  - Các dạng thám mã: Các dạng tấn công vào bản mã.
    - Chỉ biết bản mật ( ciphertext only attack).
    - Biết một số cặp bản rõ và bản mật tương ứng ( known plaintext attack ).
    - Lựa chọn trước bản rõ ( chosen plaintext attack ).
    - Bản mã cho trước ( chosen ciphertext attack ).
    - Văn bản tuỳ chọn ( chosen text attack ).

# Mật mã và thám mã

- Chỉ biết bản mật ( ciphertext only attack).
  - Dạng thám mã này là yếu nhất. Bên thám mã biết:
    - Thuật toán mật mã.
    - Bản mật.
  - Phương pháp tấn công: phương pháp vét cạn:
    - Thử tất cả các tổ hợp khóa có thể để tìm ra tổ hợp khóa thích hợp.
    - Khi không gian khóa lớn thì khó thực hiện.
  - Đối phương biết thuộc tính thống kê của nguồn tạo ra bản rõ, tìm bản mật qua phân tích thống kê.
    - Đối phương biết thêm: dạng ban đầu của bản rõ: ngôn ngữ, nguồn gốc, hoặc định dạng file.
  - Dễ đối phó: đối phương chỉ có lượng thông tin ít nhất để phá mã.

# Mật mã và thám mã

- Biết một số cặp bản rõ và bản mật tương ứng ( known plaintext attack ).
  - Thám mã biết:
    - Thuật toán mã hoá.
    - Mã mật.
    - Một hoặc một số cặp bản rõ – bản mật được dung cùng một khoá mật.
  - Thám mã tìm cách phát hiện khóa mật K.
  - Thám mã có thể dựa vào nguồn gốc của thông điệp và ước đoán được một số thông tin trong bản rõ. Từ đó dựa vào cặp thông điệp xác định khóa mật.

# Mật mã và thám mã

- Lựa chọn trước bản rõ ( chosen plaintext attack )
  - Khi bên thám mã thu được hệ thống nguồn
  - Sử dụng một bản rõ được lựa chọn trước để xác định bản mã và từ đó xác định cấu trúc khóa mật.
  - Thám mã biết:
    - Thuật toán mã hoá.
    - Văn bản mật mã.
    - Bản rõ được thám mã lựa chọn cùng với bản mật do khoá mật sinh ra.

# Mật mã và thám mã

- Bản mã chọn trước ( chosen ciphertext attack ). Thám mã biết:
  - Thuật toán mã hoá.
  - Bản mật.
  - Nội dung của một số bản mã và bản rõ đã được giải mã tương ứng sử dụng khoá mật.
  - Thám mã phải tìm bản rõ hoặc xác định được khóa mật.

# Mật mã và thám mã

- Văn bản tuỳ chọn ( chosen text attack ). Thám mã biết:
  - Thuật toán mã hoá.
  - Bản mật.
  - Bản rõ được thám mã lựa chọn cùng với bản mật sinh ra bởi khoá mật.
  - Nội dung của bản mật và bản rõ được đã giải mã tương ứng sử dụng mã mật.
  - Tìm khoá mật.

# Mật mã và thám mã

- Chỉ có các thuật toán mã hóa yếu sẽ bị phá đối với loại tấn công chỉ dùng văn bản mật.
- Các thuật toán mã hóa được thiết kế để chống dạng tấn công với bản rõ đã biết ( known plaintext attack ).

# Tính mật

- **An toàn vô điều kiện** ( unconditional secure ) – **Mật hoàn hảo**
  - Nếu bản mật không đủ thông tin để xác đinh duy nhất bản rõ tương ứng, không phụ thuộc vào đối phương có bao nhiêu bản mật.
  - Tính mật của bản mã được đảm bảo không phụ thuộc vào thời gian mà đối phương dùng để phá mã mật.
  - Chưa có sơ đồ mật mã đạt an toàn vô điều kiện trừ sơ đồ mã mật sử dụng một lần ( one-time pad )

# Tính mật

- **An toàn thực tiễn** hay **an toàn theo tính toán** ( computational secure ) :
  - Giá thành để phá hệ mật vượt quá giá trị của thông tin được mã.
  - Thời gian để phá hệ mật vượt quá thời hạn giữ mật của thông tin.

# Tính mật

- Ví dụ: thuật toán DES ( Data Encryption Standard ): Khoá nhị phân
  - Độ dài 32 bit $\Rightarrow$Số lượng khoá: $2^{32} \Rightarrow 35.8$ phút xử lý với tốc độ 1 phép mã hoá/$\mu$s $\Rightarrow 2.15$ ms với tốc độ $10^6$ phép mã hoá / $\mu$s.
  - Độ dài 56 bit $\Rightarrow$Số lượng khoá: $2^{56} \Rightarrow 1142$ năm xử lý với tốc độ 1 phép mã hoá/$\mu$s $\Rightarrow 10.01$ giờ với tốc độ $10^6$ phép mã hoá / $\mu$s.
  - Độ dài 128 bit $\Rightarrow$Số lượng khoá: $2^{128} \Rightarrow 5.4 \times 10^{24}$ năm xử lý với tốc độ 1 phép mã hoá/$\mu$s $\Rightarrow 5.4 \times 10^{18}$ năm với tốc độ $10^6$ phép mã hoá / $\mu$s.
- Ví dụ: Khoá sử dụng 26 ký tự bằng các phép hoán vị $\Rightarrow$Số lượng khoá: $26! \approx 4 \times 10^{26} \Rightarrow 6.4 \times 10^{12}$ năm xử lý với tốc độ 1 phép mã hoá/$\mu$s $\Rightarrow 6.4 \times 10^6$ năm với tốc độ $10^6$ phép mã hoá / $\mu$s.

# Một số phương pháp mật mã khóa đối xứng kinh điển

- Các phương pháp thay thế
  - Mã Caesar
    - Các ký tự chữ cái được gán giá trị ( a = 1, b = 2, ... )
    - Ký tự của bản rõ ( plaintext ) *p* được thay thế bằng ký tự của bản mật ( ciphertext ) *C* theo luật mã hoá sau:

      $$C = E(p) = (p + k) \bmod (26)$$

      Trong đó *k* nhận các giá trị từ 1 đến 25.
    - Trong phương pháp này, *k* chính là khoá mật mã.

# Một số phương pháp mã hóa đối xứng kinh điển

- Quá trình giải mã:

$$p = \mathrm{D}(C) = (C - k) \bmod (26)$$

- Phương pháp phá mã: một cách đơn giản: dùng các khoá *k* từ 1 đến 25 để giải mã cho đến khi nhận được thông điệp có ý nghĩa.
- Các vấn đề của mã Caesar:
  - Thuật toán mã hoá và giải mã đã biết trước.
  - Thám mã:
    - Không gian khóa nhỏ: chỉ có 25 khoá;
    - Khi thám mã bằng phương pháp vét cạn: chỉ cần thử với 25 khóa;
  - Ngôn ngữ trong bản rõ đã biết trước và dễ dàng nhận biết.

# Một số phương pháp mật mã khóa đối xứng kinh điển

- Các vấn đề cần nghiên cứu khi khảo sát các thuật toán mật mã:
  - Thuộc tính thống kê của nguồn tin bản rõ
  - Sơ đồ tạo khóa, không gian khóa, các sơ đồ quản lý và phân phối khóa
  - Thuật toán mã hóa và giải mã
  - Độ an toàn của hệ mật.
  - Vấn đề khẳng định giải mã được đúng bản rõ.

# Một số phương pháp mã hóa đối xứng kinh điển

- Mã mật Hill
  - Thuật toán mã hoá
    - Mỗi ký tự được gán giá trị số: a = 0, b = 1, ..., z = 25
    - Lựa chọn *m* ký tự liên tiếp của bản rõ;
    - Thay thế các ký tự đã lựa chọn bằng *m* ký tự mã mật.
    - Việc thay thế ký tự được thực hiện bằng *m* phương trình tuyến tính.
    - Hệ phương trình mã hóa:

      $C = KP \ (\text{mod } 26)$

      K- ma trận khóa
  - Thuật toán giải mã

    $P = K^{-1}C \ (\text{mod } 26)$

# Một số phương pháp mã hóa đối xứng kinh điển

- Ví dụ: với m = 3, hệ các phương trình tuyến tính có dạng sau:

$$C_1 = ( k_{11}p_1 + k_{12}p_2 + k_{13}p_3 ) \bmod 26$$
$$C_2 = ( k_{21}p_1 + k_{22}p_2 + k_{23}p_3 ) \bmod 26$$
$$C_3 = ( k_{31}p_1 + k_{32}p_2 + k_{33}p_3 ) \bmod 26$$

$$\begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}$$

$$\mathbf{C = KP}$$

# Một số phương pháp mã hóa đối xứng kinh điển

- Ma trận K là ma trận khoá mật mã
- Ví dụ: với ma trận **K** bằng:

$$K = \begin{pmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{pmatrix}$$

Xâu ký tự: "paymoremoney" sẽ được mã hoá thành "LNSHDLEWMTRW"

"pay" $\Leftrightarrow (15, 0, 24)$; $\mathbf{K}(15, 0, 24)^T \bmod 26 = (11, 13, 18) \Leftrightarrow$ "LNS"

# Một số phương pháp mã hóa đối xứng kinh điển

- Giải mã thông điệp bằng ma trận $\mathbf{K^{-1}}$.

$$K^{-1} = \begin{pmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{pmatrix}$$

- Hệ mã Hill:
- Các phép toán thực hiện theo modulo 26

$$\begin{cases} C = E_K(P) = KP \\ P = D_K(C) = K^{-1}C = K^{-1}KP = P \end{cases}$$

# Một số phương pháp mã hóa đối xứng kinh điển

- Độ an toàn của hệ mã Hill
  - Tính mật cao khi phía tấn công chỉ có bản mật.
  - Thám mã hệ mã Hill: dễ dàng bị phá nếu bên tấn công biết được bản rõ và bản mật tương ứng ( known plaintext attack )
    - Hệ mã mật Hill *m* x *m;*
    - Thám mã đã có m cặp bản rõ – bản mật, mỗi văn bản có độ dài *m*;
    - Tạo các cặp: $P_j = ( p_{1j}, p_{2j}, ..., p_{mj} )$ và $C_j = ( C_{1j}, C_{2j}, ..., C_{mj} )$ sao cho $C_j = KP_j$ với $1 \le j \le m$ đối với một khoá *K* chưa biết.
    - Xác định hai ma trận *m* x *m*, $\mathbf{X} = ( p_{ij} )$ và $\mathbf{Y} = ( C_{ij} )$

# Một số phương pháp mã hóa đối xứng kinh điển

- Ta có $\mathbf{Y} = \mathbf{XK} \Rightarrow \mathbf{K} = \mathbf{X}^{-1}\mathbf{Y}$.
- Ví dụ: bản rõ: "friday" được mã hoá bằng mật mã Hill 2 x 2 thành "PQCFKU".
  - Ta có: K( 5 17 ) = ( 15 16 ); K( 8 3 ) = ( 2 5 ); K( 0 24 ) = ( 10 20 )
  - Với hai cặp ban đầu ta có :

$$\begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix} \mathrm{K} \Rightarrow$$

$$\mathrm{K} = \begin{pmatrix} 5 & 17 \\ 8 & 3 \end{pmatrix}^{-1} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 9 & 1 \\ 2 & 15 \end{pmatrix} \begin{pmatrix} 15 & 16 \\ 2 & 5 \end{pmatrix} = \begin{pmatrix} 7 & 19 \\ 8 & 3 \end{pmatrix}$$

# Một số phương pháp mã hóa đối xứng kinh điển

- Hệ thống mã Vernam.
  - Hệ mã mật Vernam:
    - Dùng mã nhị phân
    - Mã hoá: $C_i = p_i \oplus k_i$
      - $p_i$: bit thứ *i* của bản rõ;
      - $k_i$: bit thứ *i* của khoá;
      - $C_i$: bit thứ i của bản mật;
      - $\oplus$: phép toán XOR.
    - Giải mã : $p_i = C_i \oplus k_i$
    - Tạo khoá:
      - Khoá có độ dài bằng bản rõ.
      - Khóa được chọn sao cho khoá và bản rõ độc lập thống kê.
      - Tạo vòng lặp với một khoá. Như vậy thực tế, hệ thống làm việc với một khóa rất dài nhưng lặp lại.

# Một số phương pháp mã hóa đối xứng kinh điển

- Độ an toàn:
  - Hệ thống Vernam có thể bị phá nếu đối phương biết một bản mã có độ dài đủ lớn, sử dụng một số bản rõ đã biết.
  - Với khoá được sinh ngẫu nhiên, có độ dài bằng độ dài bản rõ, không lặp lại: sơ đồ mã sử dụng một lần ( one-time pad ): không thể phá khoá. Đầu ra độc lập thống kê với bản rõ.
  - Vấn đề nảy sinh: tính mật cho quá trình trao đổi khoá

# Một số phương pháp mã hóa đối xứng kinh điển

- Đảm bảo an toàn dữ liệu: nhiều thuật toán mật mã.
- Đánh giá một thuật toán mật mã theo:
  - Tính mật, độ phức tạp, tốc độ thực hiện thuật toán, sơ đồ phân phối khoá.
- Các phương pháp mật mã kinh điển như phương pháp mã hoá thay thế, hoán vị còn đơn giản.
  - Nhược điểm:
    - Độ an toàn thấp vì không đạt độ phức tạp cần thiết;
    - Dễ bị lộ khoá do cả người gửi và người nhận đều sử dụng cùng một khoá => cần pha phân phối khoá dễ bị tấn công

# Phương pháp mật mã DES

- Bản rõ X, bản mã mật Y là các chuỗi nhị phân độ dài 64 bit.
- Khóa K có độ dài 56 bit.
- Từng khối 64 bit được mã hóa độc lập sử dụng chung một khóa.

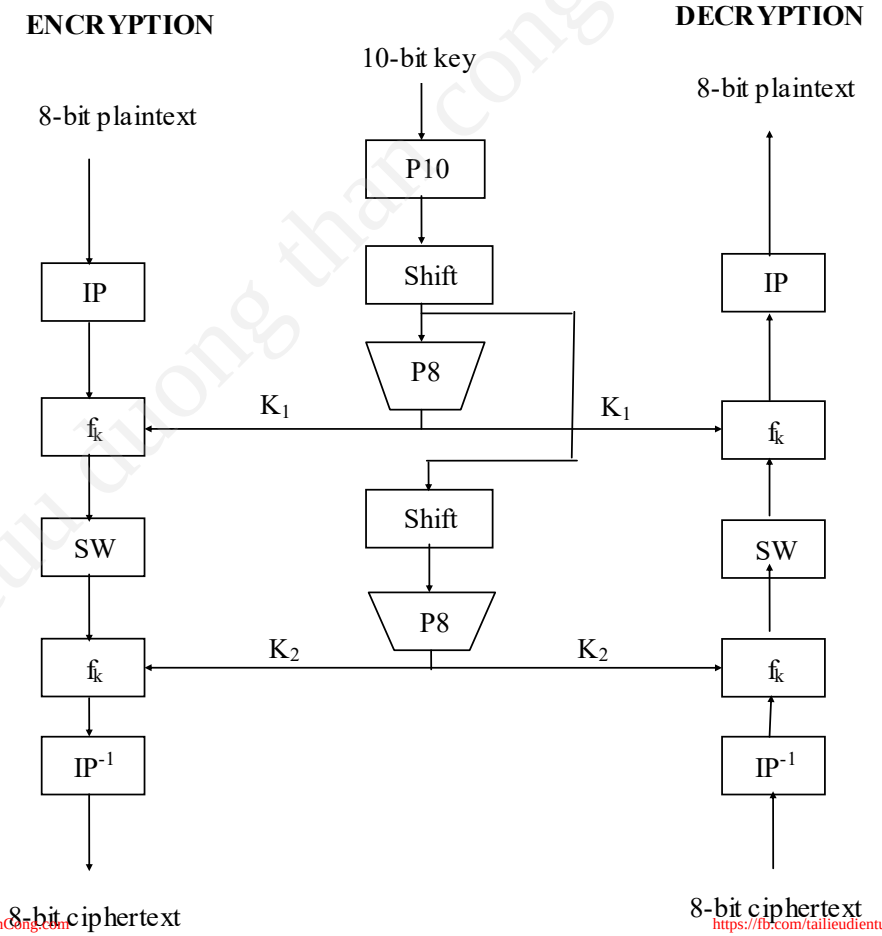
# Phương pháp mật mã DES

- Phương pháp S-DES( DES giản lược )
- Phương pháp mật mã DES

# S- DES
# (Simplified data encryption standard)

- Cấu trúc của DES là rất phức tạp
  - S-DES - phiên bản đơn giản của DES;
  - Cho phép:
    - Mã hoá và giải mã bằng tay;
    - Hiểu biết sâu về hoạt động chi tiết của giải thuật DES.
- S-DES đơn giản hơn nhiều so với DES
  - Các tham số của S-DES nhỏ hơn trong DES;
    - Do giáo sư Edward Schaefer thuộc trường đại học Santa Clara phát triển

# Thuật toán S-DES (Simplified DES)



# Thuật toán S-DES (Simplified DES)

- Giải thuật mã hoá S-DES sử dụng phương pháp mã hoá theo khối
- Đầu vào:
  - 8-bit block của bản rõ
  - 10-bit khoá
- Đầu ra:
  - 8-bit của bản mã

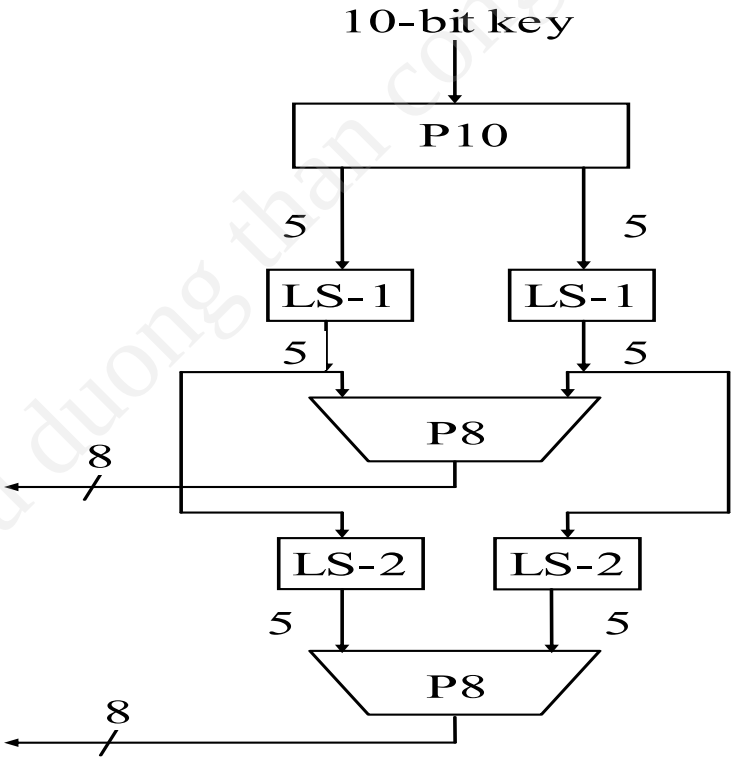
# Thuật toán S-DES

- Thuật toán mã hoá bao gồm 4 hàm:
  - Hàm IP(Initial Permutation)
  - Hàm $f_k$
  - Hàm SW (Switch)
  - Hàm $IP^{-1}$
- Thuật toán mã hoá::

  $$ciphertext=IP^{-1}(f(SW(f(IP(plaintext)))))$$
- Thuật toán giải mã :

  $$plaintext = IP(f(SW(f(IP^{-1}(ciphertext)))))$$

# Thuật toán S-DES (Simplified DES)
## Sinh khoá trong S-DES



# Thuật toán S-DES (Simplified DES)
# Các hàm sinh khoá:

- P10:Đây là hàm hoán vị tuân theo luật như trong bảng

| P10 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 5 | 2 | 7 | 4 | 10 | 1 | 9 | 8 | 6 |

- LS-1: Là hàm dịch vòng 1 bit
- LS-2: Là hàm dịch vòng 2 bit
- P8:Là hàm hoán vị tuân theo luật như trong bảng

| P8 | | | | | | | |
|---|---|---|---|---|---|---|---|
| 6 | 3 | 7 | 4 | 8 | 5 | 10 | 9 |

# Thuật toán S-DES Mã hoá S-DES:

## Hàm IP và hàm $IP^{-1}$:

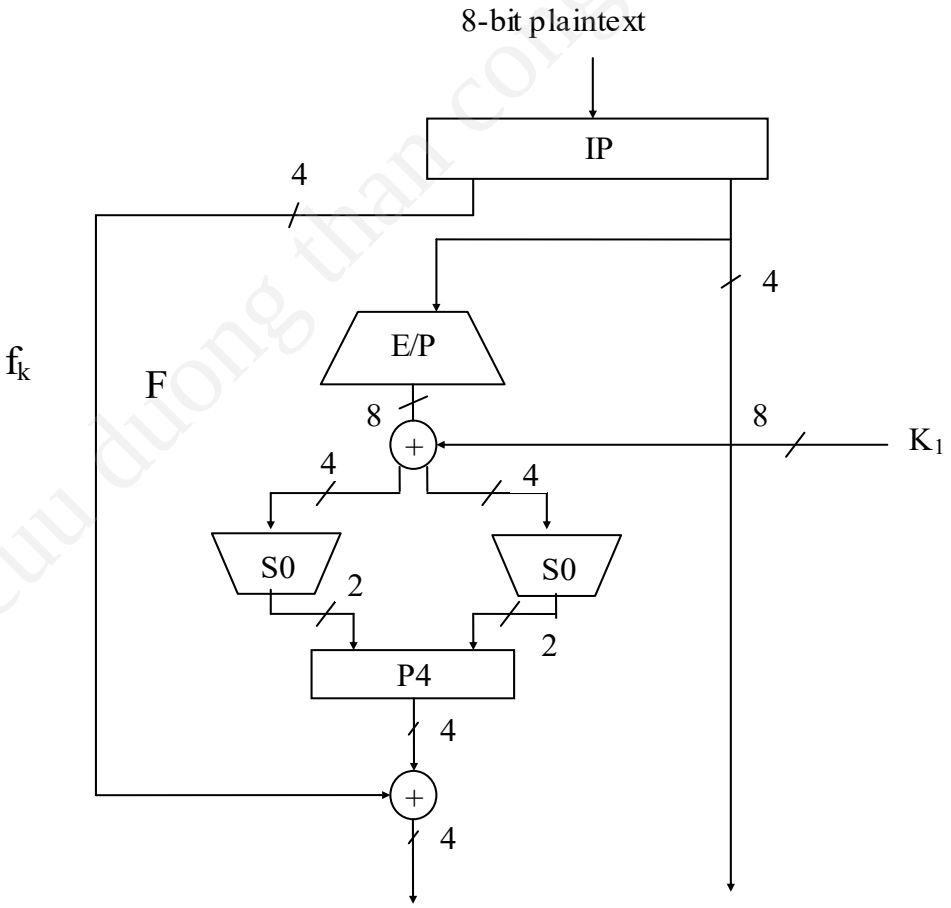
+ Hàm IP tuân theo luật sau:

| IP | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 6 | 3 | 1 | 4 | 8 | 5 | 7 |

+ Hàm $IP^{-1}$ tuân heo luật sau:

| $IP^{-1}$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 1 | 3 | 5 | 7 | 2 | 8 | 6 |

# Thuật toán S-DES Hàm $f_k$:

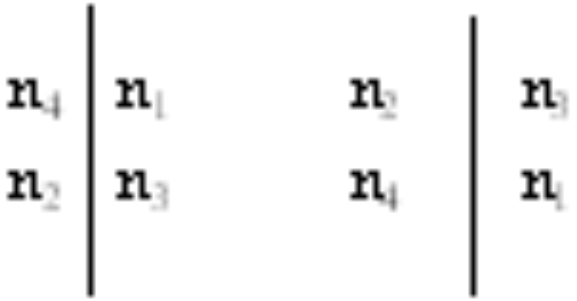


# Thuật toán S-DES
# E/P(expension/permutation):

- Hàm E/P tuân theo luật sau:

| E/P | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 1 | 2 | 3 | 2 | 3 | 4 | 1 |

- Nếu gọi 4 bit đầu vào là ($n_1$,$n_2$,$n_3$,$n_4$) thì E/P được biểu diễn chi tiết như sau



# Thuật toán S-DES
## Khối thay thế S-box

- Đầu vào S-box: khối 8 bit được chia thành hai khối 4 bit;
- Mỗi khối 4 bit được đưa vào $S_0$ và $S_1$
- Thay thế mỗi khối 4 bit bằng khối 2 bit;
- Các khối $S_0$ và $S_1$ được định nghĩa như sau:

$S_0$:

| | | | |
|---|---|---|---|
| 1 | 0 | 3 | 2 |
| 3 | 2 | 1 | 0 |
| 0 | 2 | 1 | 3 |
| 3 | 1 | 3 | 2 |

$S_1$:

| | | | |
|---|---|---|---|
| 0 | 1 | 2 | 3 |
| 2 | 0 | 1 | 3 |
| 3 | 0 | 1 | 2 |
| 2 | 1 | 0 | 3 |

# Thuật toán S-DES
## Khối thay thế S-box

- Phần tử trong khối S-box có độ dài 2 bit;
- Quá trình thay thế trong S-box:
  - Với 4 bit đầu vào là (b1,b2,b3,b4);
    - b1 và b4 kết hợp thành một số chỉ hàng của S box,
    - b2 và b3 tạo thành số chỉ cột trong S box;
    - Phần tử nằm trên hàng và cột đã xác định thay thế cho 4 bit đầu vào của S-box đó.

# Thuật toán S-DES
## Hoán vị P4

- Hoán vị P4 tuân theo luật sau:

| P4 | | | |
|---|---|---|---|
| 2 | 4 | 3 | 1 |

# Thuật toán S-DES
## Hàm SW

- Hàm $f_k$ chỉ thực hiện trên 4 bit trái của đầu vào;
- Hàm SW hoán đổi 4 bit phải và 4 bit trái để lần áp dụng hàm $f_k$ thứ 2 sẽ thực hiện trên 4 bit phải.
- Áp dụng hàm $f_k$ lần 2 thực hiện các hàm E/P ,$S_0$,$S_1$,P4 như trên.

# Mật mã DES (Data Encryption Standard)

- Phương pháp mật mã DES được Ủy ban tiêu chuẩn Mỹ (U.S National Bureau for Standards) công bố năm 1971 để sử dụng trong các cơ quan chính phủ liên bang.
- Thuật tóan được IBM phát triển.
- DES có một số đặc điểm sau:
  - Sử dụng khoá 56 bít.
  - Xử lý khối vào 64 bít, biến đổi khối vào thành khối ra 64 bít.
  - Mã hoá và giải mã được sử dụng cùng một khoá.
  - DES được thiết kế để thực hiện hiệu quả bằng phần cứng.
  - DES thường được sử dụng để mã hoá các dòng dữ liệu mạng và mã hoá dữ liệu được lưu trữ trên đĩa.

# Giải mã DES

- Với DES, có thể sử dụng cùng chức năng để giải mã hoặc mã hoá một khối.
- Điểm khác biệt: khi giải mã, các khoá phải được sử dụng theo thứ tự ngược lại.

# Độ an toàn của DES

- Độ an tòan hệ mật khoá đối xứng phụ thuộc hai tham số: độ phức tạp của thuật toán và độ dài của khoá.
- Khoá có độ dài 56 bít, không gian khóa sẽ có $2^{56}$ khoá có thể sử dụng.
- Nếu tính mật của phương pháp chỉ phụ thuộc vào độ phức tạp của thuật toán.
  - Có nghĩa rằng sẽ không có phương pháp nào khác để phá hệ thống mật mã ngoài cách thử mọi tổ hợp khoá có thể: phương pháp tấn công vét cạn (brute-force attack).
  - Nếu một siêu máy tính có thể thử một triệu khoá trong một giây, thì thời gian tổng cộng để để tìm ra khoá đúng là khoảng 2000 năm.

# Kết luận

- DES đã được phân tích kỹ lưỡng và công nhận là vững chắc. Các hạn chế của nó đã được hiểu rõ và có thể xem xét trong quá trình thiết kế.
  - Để tăng độ an toàn của DES, sử dụng các hệ thống mật mã DES mở rộng
  - DES mở rộng khoá có thể là 128 bít, 192 bít,... độ lớn khối có thể là 128 bít. Do vậy, độ an toàn của DES mở rộng cao hơn rất nhiều.

# Mật mã khối ( block cipher )

- Định nghĩa
  - Mã khối là mật mã khóa đối xứng thực hiện trên nhóm bit có độ dài cố định. Nhóm bit này được gọi là một khối. Quá trình chuyển đổi không thay đổi.
  - Khi mã hóa, mã khối có thể thực hiện trên từng khối độ dài 128 bit của bản rõ tại đầu vào thứ nhất và cho ra khối 128 bit của mã mật.
    - Quá trình biến đổi được kiểm soát bằng đầu vào thứ hai: khóa mật
  - Quá trình giải mã thực hiện tương tự: nhận tại đầu vào thứ nhất khối 128 bit của mật mã, khóa mật và tại đầu ra ta nhận được khối 128 bit của bản rõ

# Mật mã khối ( block cipher )

- Để mã hóa bản tin có độ dài lớnhơn kích thước khối, ( ví dụ 128 bit ), các chế độ xử lý ( mode of operation ) được sử dụng.
- Mã hóa khối tương phản với mã hóa dòng ( stream cipher ), trong đó mỗi ký tự được thao tác một lần và quá trình chuyển đổi thay đổi trong suốt quá trình mã hóa.
- Ví dụ mã hóa khối:
  - Thuật toán DES do công ty IBM xây dựng và công bố năm 1977.
  - Hậu duệ của DES, Advanced Encryption Standard (AES), ra đời năm 2001.

# Mật mã khối

- Mã khối và mã dòng: mã theo từng khối bit và mã từng bit riêng rẽ

Plaintext= 010100110111=
(010)(100)(110)(111)
Ciphertext = 111 011 000 101
theo key=1
Ciphertext = 100 011 011 111
theo key=4

| key | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| 0 | 001 | 111 | 110 | 000 | 100 | 010 | 101 | 011 |
| 1 | 001 | 110 | 111 | 100 | 011 | 010 | 000 | 101 |
| 2 | 001 | 000 | 100 | 101 | 110 | 111 | 010 | 011 |
| 3 | 100 | 101 | 110 | 111 | 000 | 001 | 010 | 011 |
| 4 | 101 | 110 | 100 | 010 | 011 | 001 | 011 | 111 |

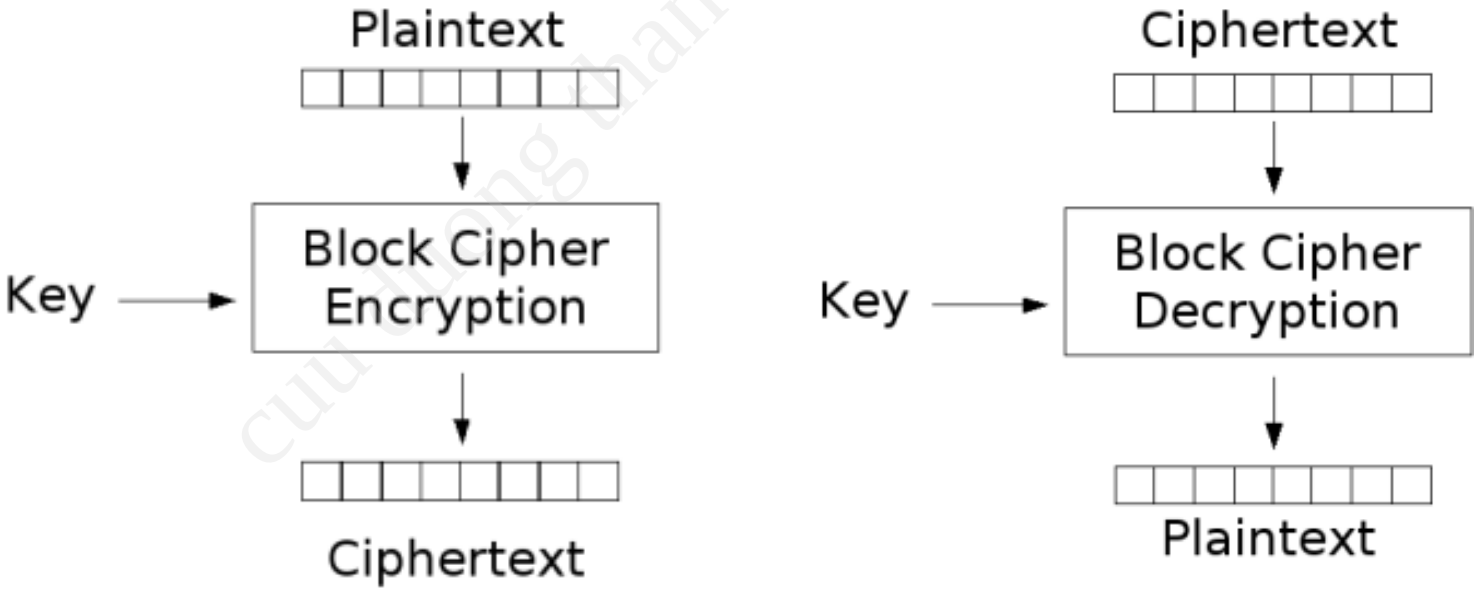
- Số lượng key: $2^2 < 5 < 2^3$ key cần có độ dài 3 bit để cho kích thước khoá bang kích thước khối: key size= block size= 3.
- Kích thước khoá nhỏ: không an toàn.
- Nếu bên tấn công Eve bắt được thông điệp mật C=001, sẽ khó lựa chọn P= 000 hay 101.

# Mật mã khối ( block cipher )

- Mật mã khối gồm một cặp thuật toán:
  - Thuật toán mã hóa, $E$, và
  - Thuật toán giải mã, $E^{-1}$.
  - Cả hai thuật toán đều có hai đầu vào:
    - Khối dữ liệu đầu vào kích thước $n$ bit và
    - Khóa độ dài $k$ bit,
  - Đầu ra là khối dữ liệu kích thước $n$-bit.

# Mật mã khối ( block cipher )

- Sơ đồ mã hóa và giải mã khối



# Mật mã khối ( block cipher )

- Đối với một khóa xác định, hàm giải mã là hàm ngược của hàm mã hóa:

$$E_K^{-1}( E_K ( M ) ) = M$$

  Đối với mỗi khối *M* và khóa *K*.

- Với mỗi khóa *K*, $E_K$ là hoán vị ( song ánh ) trên tập hợp các khối đầu vào. Mỗi khóa sẽ lựa chọn một hoán vị từ tập hợp $2^n!$ phần tử.

# Mật mã khối

- Điều kiện để tạo mã khối an toàn:
  - Kích thước khối phải đủ lớn để ngăn chặn tấn công bang phân tích thống kê.
    - Kích thước khối lớn sẽ làm chậm quá trình xử lý
  - Không gian khoá (độ dài khoá) cần phải đủ lớn để ngăn chặn tấn công vét cạn.
    - Khoá không được có độ dài quá lớn – khó phân phối và quản trị

# Mật mã khối ( block cipher )

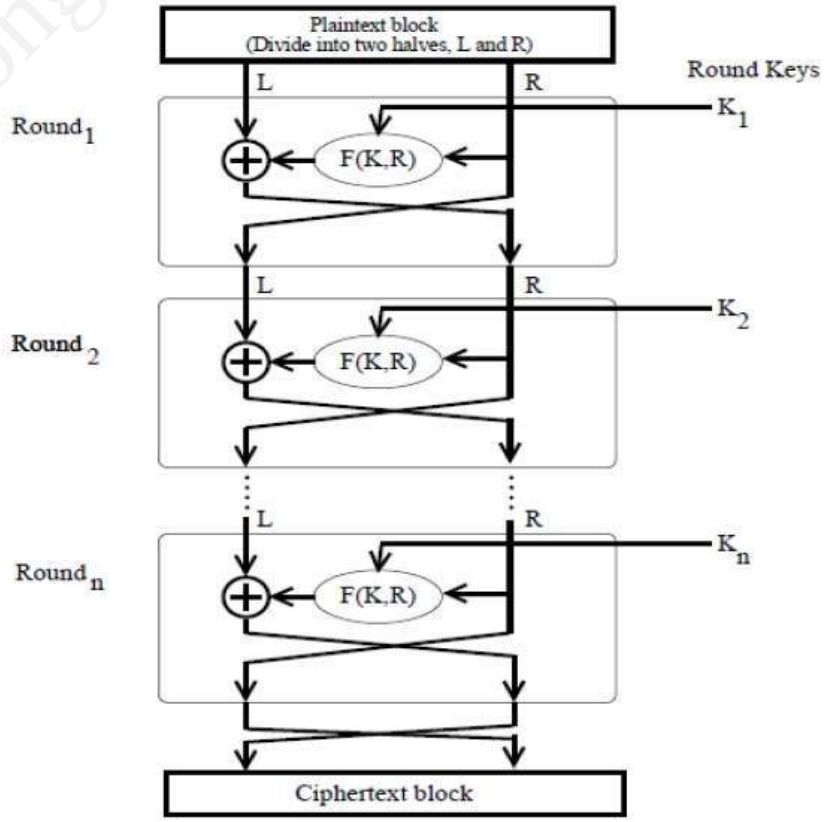
- Kích thước khối *n*, thông thường bằng 64 hoặc 128 bit,
  - Một số mật mã khối có kích thước khối thay đổi.
- Kỹ thuật cho phép bản rõ với độ dài tùy ý được mã hóa: sơ đồ padding.
- Mỗi chế độ làm việc có một đặc tính riêng biệt phụ thuộc vào sai số truyền lan, tính dễ truy cập ngẫu nhiên và khả năng bị tổn thương đối với từng loại tấn công cụ thể.
- Kích thước khóa *k* thường bằng 40, 56, 64, 80, 128, 192 và 256 bits.
  - Đến năm 2006, độ dài khóa 80 bits thường được chọn là kích thước khóa tối thiểu để ngăn chặn tấn công vét cạn ( brute force attacks ).

# Mật mã khối

- Điều kiện để thiết kế các mã khối an toàn
  - Hàm nhập nhằng (Confusion):
    - Là hàm làm cho mối quan hệ giữa bản mật vào bản rõ đủ phức tạp để đối phương không phát hiện được quy luật;
    - Hàm này là hàm phi tuyến
    - Thường dùng phép thế
  - Hàm khuếch tán (Diffusion):
    - Làm phân tán thông tin từ bản rõ vào toàn bộ bản mật;
    - Sự thay đổi trong bản rõ sẽ ảnh hưởng tới nhiều phần của bản mật
    - Ngăn chặn tấn công bằng phân tích thống kê
    - Thường dùng hoán vị, đổi chỗ

# Mật mã khối ( block cipher )

- **Mã khối lặp**
  - Phần lớn các mật mã khối được xây dựng trên cơ sở áp dụng lặp đi lặp lại một hàm f đơn giản gồm thay thế và hoán vị.
    - Mỗi vòng lặp được gọi là một chu kỳ ( round ) và hàm lặp được gọi là hàm quay vòng. Thông thường số lần quay vòng từ 4 đến 32 lần.
    - Nhiều mật mã khối có thể được phân loại thành các cấu trúc Feistel - các mạng thay thế-hoán vị.
    - Các phép toán số học, lôgic ( đặc biệt là XOR ), các S-box và các phép hoán vị thường được sử dụng



# Mật mã khối

- Các thuộc tính của mật mã khối:
  - Kích thước khối Block size: kích thước càng lớn sẽ càng an toàn
  - Kích thước khoá Key size: kích thước càng lớn sẽ càng an toàn (không gian khoá lớn)
  - Số vòng (number of round): càng nhiều vòng lặp, độ phức tạp và tính an toàn càng cao.
  - Mode mã hoá (Encryption modes): xác định phương thức phân chia thông điệp lớn thành  các khối – quan trọng đối với sự an toàn của thông điệp mật

# Mật mã khối ( block cipher )

- **Mật mã khối và các mật mã cơ sở khác**
  - Mật mã khối có thể được sử dụng để xây dựng các mật mã cơ sở khác (cryptographic primitives).
    - Để cho các phương pháp mã hóa này trở thành mật mã an toàn, những thuật toán này cần được xây dựng theo phương pháp đúng đắn.
  - Các mật mã dòng có thể được xây dựng dựa trên mật mã khối.
    - Các chế độ OFB-mode và CTR mode là những chế độ theo khối.
    - Cho phép chuyển một mật mã khối thành mật mã dòng.
  - Mật mã khối có thể sử dụng để xây dựng các hàm băm, và ngược lại, những hàm băm có thể là cơ sở để xây dựng những mật mã khối.
    - Ví dụ: những mật mã khối dựa trênhàm băm:SHACAL, BEAR và LION.

# Mật mã khối ( block cipher )

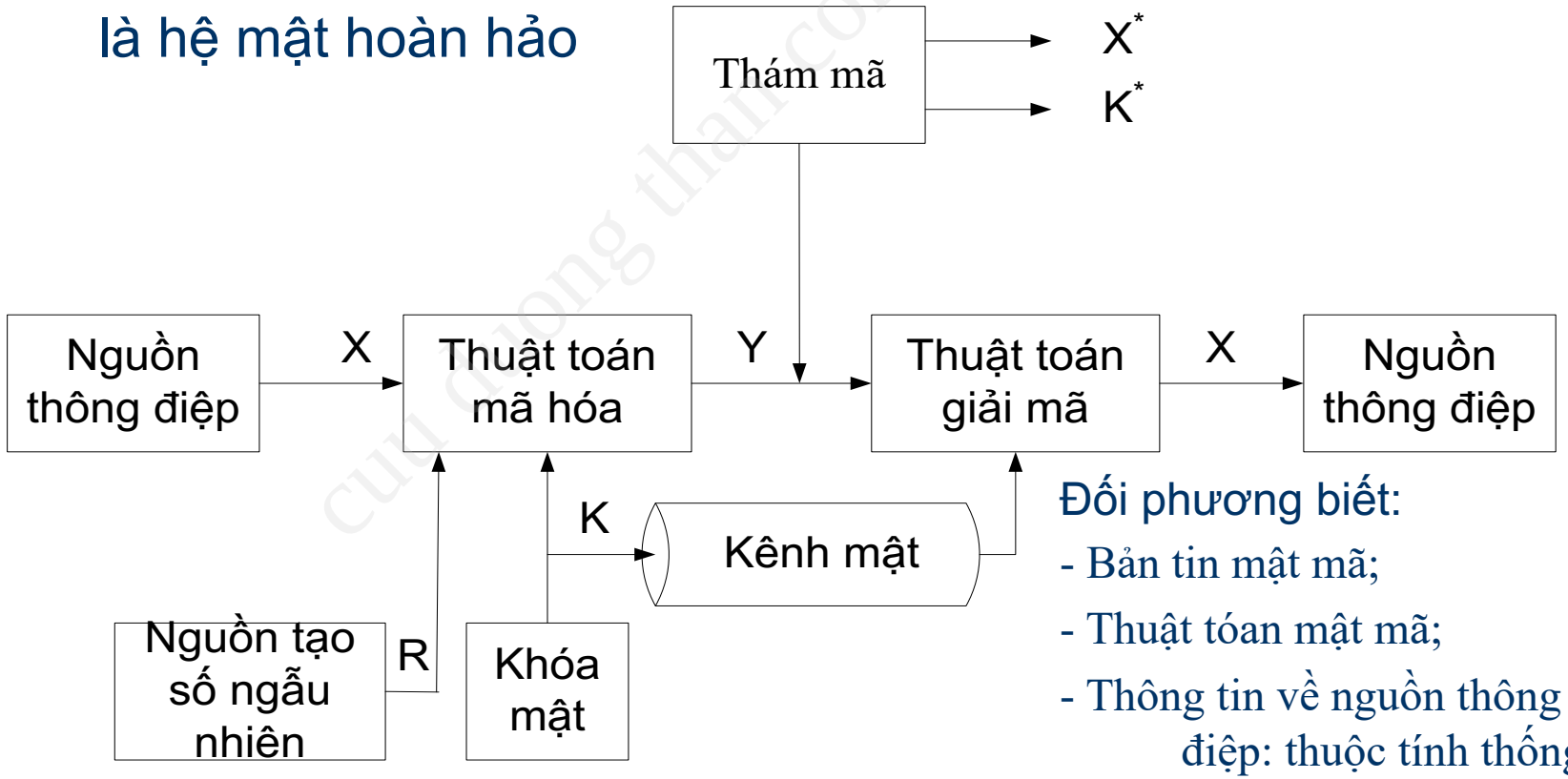
- Những bộ sinh số giả ngẫu nhiên an toàn theo phương diện mật mã có thể được tạo nên từ các mật mã khối.
- Mã xác thực thông điệp (MAC) cũng thường được xây dựng từ các mật mã khối. Ví dụ: CBC-MAC, OMAC, PMAC.
- Các mã xác thực cũng được xây dựng từ mật mã khối.
  - Cho phép thực hiện mã hóa mật và mã hóa xác thực đồng thời.
  - Điều này làm cho phương pháp cung cấp được cả tính riêng tư cũng như tính xác thực đồng thời. Ví dụ CCM, EAX, GCM, OCB.

# Mật mã khối ( block cipher )

- **Thám mã khối**
  - Bên cạnh các phương pháp thám mã như thám mã tuyến tính, thám mã vi phân, còn có một số phương pháp khác:
    - Thám mã vi phân ngắn gọn ( Truncated differential cryptanalysis );
    - Thám mã vi phân từng phần ( Partial differential cryptanalysis );
    - Tấn công trượt ( Slide attacks );
    - Tấn công Bu-mê-răng ( Boomerang attacks ),
    - Tấn công toàn phương và tích phân,
    - Tấn công XSL,
    - Tấn công vi sai bất khả và
    - Tấn công đại số
  - Đối với mỗi mật mã khối mới được xây dựng, để có tính hiệu quả, mật mã đó phải thể hiện độ an toàn đối với các tấn công đã biết.

# Hệ mật hoàn hảo và không hoàn hảo

- Điều kiện cần để hệ mật là hệ mật hoàn hảo




Đối phương biết:

- Bản tin mật mã;
- Thuật tóan mật mã;
- Thông tin về nguồn thông điệp: thuộc tính thống kê của bản rõ

# Hệ mật hoàn hảo và không hoàn hảo

- Nguồn thông tin $X = [ X_1, X_2, ..., X_M ]$, $X_i \in A$; A – bảng ký tự bản rõ cùng các thuộc tính thống kê ( latin, nhị phân, ...).
- Khoá $K = [ K_1, K_2, ... K_L ]$, khóa K được tạo ra.
  - Các ký tự của khoá K nằm trong một bảng ký tự: bảng ký tự nhị phân $\{ 0, 1 \}$
- Bộ tạo số ngẫu nhiên: $R = [ R_1, R_2, ..., R_J ]$;
- Thông điệp được mã hóa là hàm của X, R và K : $Y = [ Y_1, Y_2, ..., Y_N ]$

$$Y = E_{KR}( X )$$

- Bên nhận giải mã thông điệp bằng khóa đã phân phối:

$$X = D_K( Y )$$

# Hệ mật hoàn hảo và không hoàn hảo

- Giả thiết sử dụng khóa và tạo số ngẫu nhiên:
  - Khóa mật chỉ được sử dụng một lần.
  - M bit của văn bản rõ sẽ được mã hoá trước khi khoá mật **K** và chuỗi ngẫu nhiên **R** thay đổi.
- Đối phương chỉ biết được văn bản mã mật **Y** và thuật tóan mã hóa, giải mã.
- Sơ đồ hệ mật hoàn hảo: Văn bản rõ **X** độc lập thống kê với văn bản mã mật **Y**.

$$\mathbf{P(X = x \mid Y = y) = P(X = x)}$$

đối với mọi bản tin rõ: $\mathbf{X} = [\, x_1, x_2, ..., x_M \,]$ và bản tin mật **Y**.

# Hệ mật hoàn hảo và không hoàn hảo

- Ví dụ: hệ mã Vernam
  - Bảng chữ cái: $A = \{ 0, 1, ..., |A|-1 \}$
  - Độ dài của văn bản gốc, khoá và văn bản mã bằng nhau: $M = L = N$.
  - Khoá được chọn ngẫu nhiên: $P( \mathbf{K} = k ) = |A|^{-M}$ đối với $|A|^M$ tổ hợp khoá.
  - Quá trình mã hoá: $Y_i = X_i \oplus K_i$, $i = 1, 2, ..., M$.
  - Do với mỗi ký tự $x_j$ thuộc $X_i$ và $y_i$ thuộc $Y_j$ ta có duy nhất ki thuộc Kj, do đó: $P( Y = y \mid X = x ) = P( Z = z ) = |A|^{-M}$ không phụ thuộc vào X.

# Hệ mật hoàn hảo và không hoàn hảo

- Định lý: đối với hệ mật hoàn hảo

$$H(X) = H(X \mid Y) \leq H(K)$$

- Nếu bản rõ và bản mã cùng bảng ký tự, : $L_X = L_K$
  - Khi dấu bằng xảy ra: trong trường hợp one time pad.

- Các hệ mật không hoàn hảo
  - Đặt vấn đề: khi nào thám mã có thể phá được các hệ mật không hoàn hảo ?!

# Quản trị và phân phối khóa trong mã hóa đối xứng

- Đặt vấn đề:
  - Trong kỹ thuật mật mã truyền thống, hai phía tham gia vào truyền tin phải chia sẻ khoá mật $\Rightarrow$ khoá phải được đảm bảo bí mật : phải duy trì được kênh mật phân phối khóa.
  - Khóa phải được sử dụng một lần: Khoá phải được thường xuyên thay đổi.
  - Mức độ an toàn của bất kỳ hệ mật sẽ phụ thuộc vào kỹ thuật phân phối khoá.