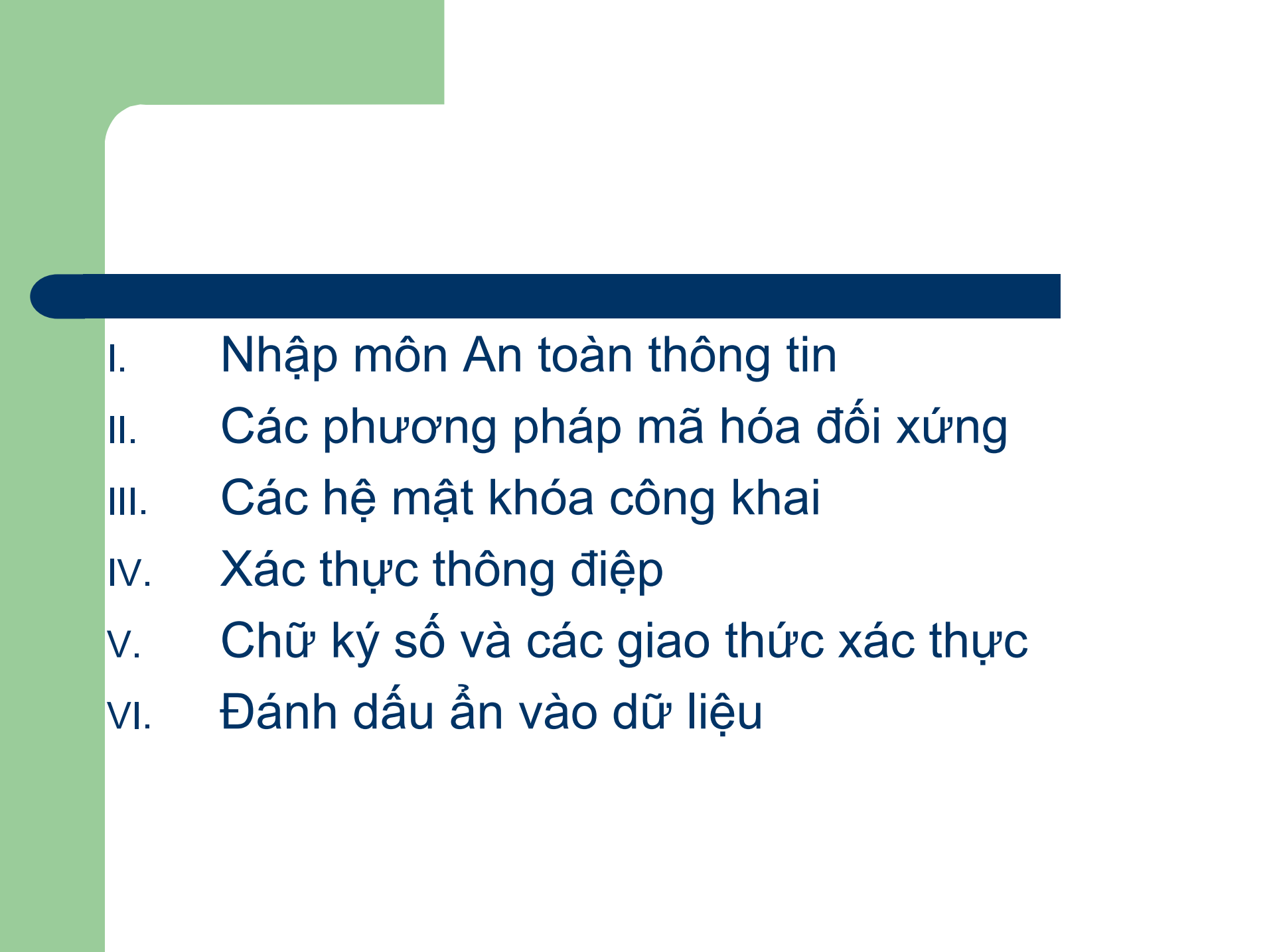


An toàn và An ninh thông tin

Nguyễn Linh Giang
Bộ môn Truyền thông
và Mạng máy tính



- 
- I. Nhập môn An toàn thông tin
 - II. Các phương pháp mã hóa đối xứng
 - III. Các hệ mật khóa công khai
 - IV. Xác thực thông điệp
 - V. Chữ ký số và các giao thức xác thực
 - VI. Đánh dấu ẩn vào dữ liệu

Chương II.

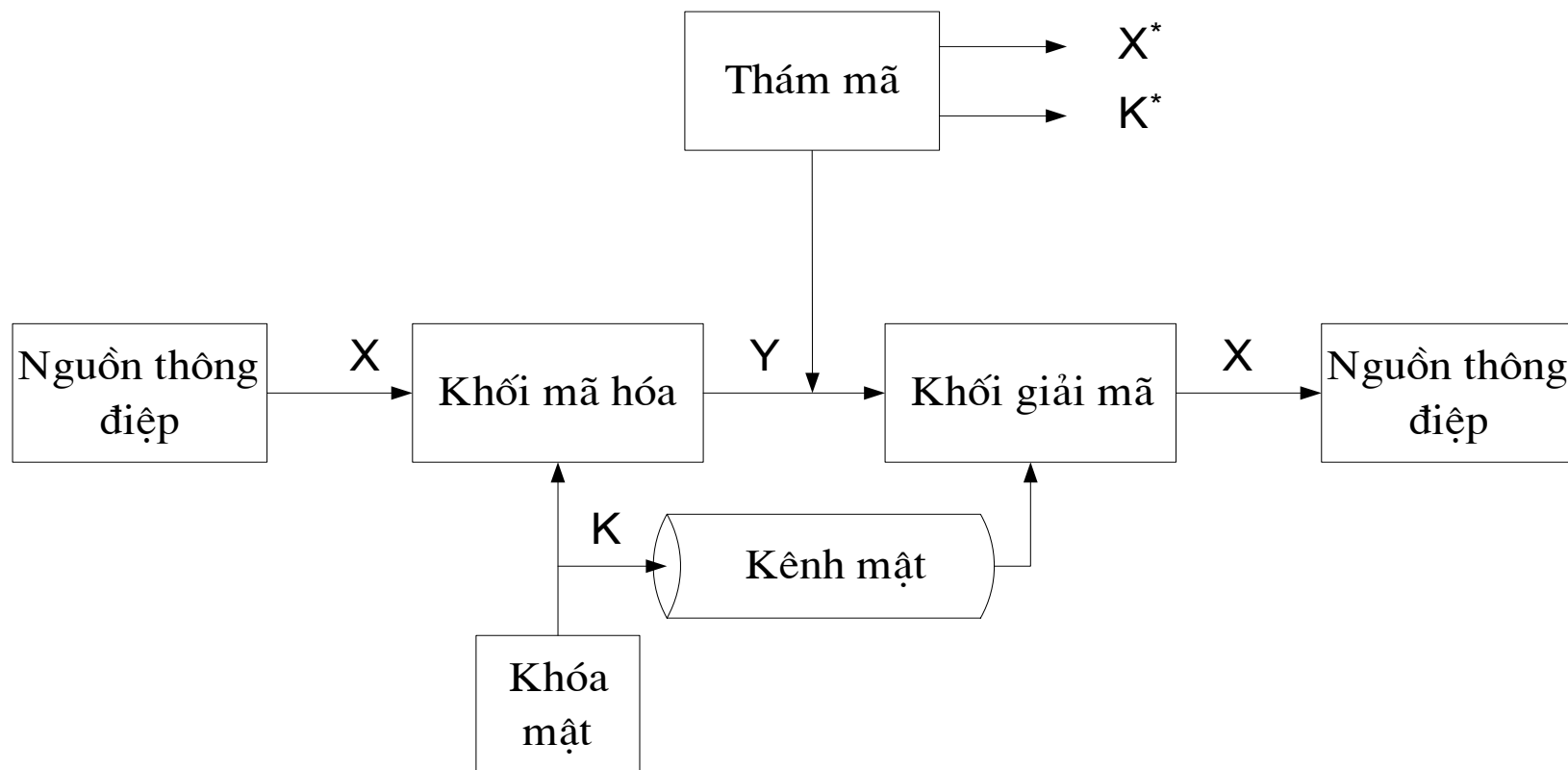
Các phương pháp mã hóa đối xứng

1. Sơ đồ chung của phương pháp mã hóa đối xứng
2. Một số phương pháp mã hóa đối xứng kinh điển
3. Phương pháp DES
4. Quản trị và phân phối khóa

Sơ đồ mã hóa đối xứng

- Giả thiết
 - 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.

Sơ đồ mã hóa đối xứng



Mô hình hệ thống mã hóa đối xứng.

Sơ đồ chung của phương pháp mã hóa đối xứng

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

Sơ đồ chung của phương pháp mã hóa đối xứng

- Khóa mật mã
 - Tập hợp khoá $K = \{ K_1, K_2, \dots K_L \}$,
 - Khóa độ dài l : $K_i = [k_{i1}, \dots, k_{il}]$;
 $k_{ij} \in C$, C - bảng ký tự khóa; thông thường $C = \{0, 1\}$

Sơ đồ chung của phương pháp mã hóa đối xứng

- Mã mật:

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

Sơ đồ chung của phương pháp mã hóa đối xứng

- Quá trình mã hóa và giải mã:

- Quá trình mã hóa:

$$Y = E_K(X)$$

- 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.

Sơ đồ chung của phương pháp mã hóa đối xứng

- Mật mã
 - Phân loại các 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;
 - Số lượng khóa được dùng trong thuật toán.
 - Hệ thống mã hóa đối xứng.
 - Hệ thống mã hóa không đối xứng.
 - Phương thức mà văn bản ban đầu được xử lý:
 - Mã hóa khối;
 - Mã hóa dòng.

Sơ đồ chung của phương pháp mã hóa đối xứng

- Thám mã
 - Chỉ biết văn bản được mã hoá;
 - Biết một số văn bản gốc và mật mã tương ứng;
 - Tấn công bằng văn bản rõ được lựa chọn trước;
 - Tấn công bằng mật mã cho trước;
 - Tấn công bằng bản rõ tùy chọn.

Sơ đồ chung của phương pháp mã hóa đối xứng

- Sơ đồ mã hóa được coi là **an toàn vô điều kiện**
 - Văn bản mã mật không chứa đủ thông tin để xác định duy nhất văn bản gốc tương ứng;
- Sơ đồ mã mật được coi là **an toàn theo tính toán**
 - Giá thành tấn công vượt quá giá trị của thông tin mật;
 - Thời gian giải mật vượt quá thời hạn giữ mật của thông tin.

Sơ đồ chung của phương pháp mã hóa đối xứng

- 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/ μ s $\Rightarrow 2.15$ ms với tốc độ 10^6 phép mã hoá / μ s.
 - Độ dài 56 bit \Rightarrow Số lượng khoá: $2^{56} \Rightarrow 1142$ năm xử lý với tốc độ 1 phép mã hoá/ μ s $\Rightarrow 10.01$ giờ với tốc độ 10^6 phép mã hoá / μ 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/ μ s $\Rightarrow 5.4 \times 10^{18}$ năm với tốc độ 10^6 phép mã hoá / μ s.

Một số phương pháp mã hóa đối xứng kinh điển

- Các phương pháp thay thế
 - Mã Caesar
 - Các ký tự chữ cái được gán giá trị ($a = 1, b = 2, \dots$)
$$C = E(p) = (p + k) \bmod (26)$$
Trong đó $k = 1 \dots 25$.
 - k là khoá mật mã.
 - Quá trình giải mã:
$$p = D(C) = (C - k) \bmod (26)$$

Một số phương pháp mã hóa đối xứng kinh điển

- 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 khóa;
 - Khi thám mã bằng phương pháp vét cạn: chỉ cần thử với 25 khóa;
 - Ngôn ngữ trong bản gốc đã biết trước và dễ dàng nhận biết.

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: $a = 0, b = 1, \dots, z = 25$
- Lựa chọn m ký tự liên tiếp của văn bản gốc;
- Thay thế các ký tự đã lựa chọn bằng m ký tự mã mật, được tính bằng m phương trình tuyến tính.
- Hệ phương trình mã hóa:

$$C = KP \pmod{26}$$

K- ma trận khóa

- Thuật toán giải mã

$$P = K^{-1}C \pmod{26}$$

Một số phương pháp mã hóa đối xứng kinh điển

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

$$C_1 = (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26$$

$$C_2 = (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26$$

$$C_3 = (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26$$

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

$$\mathbf{C} = \mathbf{K}\mathbf{P}$$

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” $\Leftrightarrow (15, 0, 24)$; $K(15, 0, 24)^T \bmod 26 = (11, 13, 18) \Leftrightarrow \text{“LNS”}$

Một số phương pháp mã hóa đối xứng kinh điển

- Giải mã thông điệp bằng ma trận 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

- Mức độ an toàn của hệ mã Hill
 - Mã mật Hill có tính mật cao khi phía tấn công chỉ có văn bản mật.
 - Thám mã hệ mã Hill: dễ dàng bị bẻ khóa nếu bên tấn công biết được văn bản rõ và văn bản mật tương ứng (known plaintext attack)
 - Hệ mã mật Hill $m \times m$;
 - Thám mã đã có m cặp văn bản gốc – văn bản mật, mỗi văn bản có độ dài m ;
 - Tạo các cặp: $P_j = (p_{1j}, p_{2j}, \dots, p_{mj})$ và $C_j = (C_{1j}, C_{2j}, \dots, C_{mj})$ sao cho $C_j = KP_j$ với $1 \leq j \leq m$ đối với một khoá K chưa biết.
 - Xác định hai ma trận $m \times m$, $\mathbf{X} = (p_{ij})$ 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: văn bản gốc: “friday” được mã hoá bằng mã mật 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 :

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

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

Một số phương pháp mã hóa đối xứng kinh điển

- Hệ thống Vernam.
 - Để chống lại quá trình thám mã, cần lựa chọn khoá thoả mãn:
 - Khoá có độ dài bằng văn bản rõ.
 - Khoá được chọn sao cho khoá và văn bản gốc độc lập thống kê.
 - Hệ mã mật Vernam:
 - Dùng cho mã nhị phân
 - $C_i = p_i \oplus k_i$
 - p_i : bit thứ i của văn bản gốc;
 - k_i : bit thứ i của khoá;
 - C_i : bit thứ i của văn bản được mã hoá;
 - \oplus : phép toán XOR.

Một số phương pháp mã hóa đối xứng kinh điển

- Giải mã bằng phép toán ngược: $p_i = C_i \oplus k_i$
- Tạo khoá: tạo vòng lặp với một khoá. Như vậy thực tế, hệ thống làm việc với một khoá rất dài nhưng lặp lại.
- Hệ thống Vernam có thể bị phá nếu đối phương biết một văn bản mã có độ dài đủ lớn, sử dụng một số văn bản gốc đã biết.
- Với khoá được sinh ngẫu nhiên, có độ dài bằng độ dài văn bản gốc, không lặp lại: sơ đồ mã sử dụng một lần (one-time pad): không thể phá khoá. Đầu ra độc lập thống kê với văn bản gốc.
- Vấn đề nảy sinh: đảm bảo mật cho quá trình gửi và nhận khoá ngẫu nhiên.

Mã hóa 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

Mã hóa khối (block cipher)

- Để mã hóa bản tin có độ dài lớn hơn 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.

Mã hóa 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 dòng (Stream Cipher)

- Mật mã dòng là mật mã khóa đối xứng, trong đó các ký tự của bản rõ được mã hóa lần lượt và quá trình biến đổi các ký tự tiếp theo thay đổi trong quá trình mã hóa. Một tên khác của mật mã dòng là mật mã trạng thái vì quá trình mã hóa từng ký tự phụ thuộc vào trạng thái hiện thời. Trong thực tế, ký tự có thể là từng bit hoặc byte.
- Stream ciphers represent a different approach to symmetric encryption from block ciphers. Block ciphers operate on large blocks of digits with a fixed, unvarying transformation. This distinction is not always clear-cut: in some modes of operation, a block cipher primitive is used in such a way that it acts effectively as a stream cipher. Stream ciphers typically execute at a higher speed than block ciphers and have lower hardware complexity. However, stream ciphers can be susceptible to serious security problems if used incorrectly: see stream cipher attacks — in particular, the same starting state must never be used twice.