



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# Nhập môn An toàn thông tin

## Chương 2. Mật mã khối và Mật mã khóa đối xứng

# Mã khối

# Nội dung

- Mã khối và tính an toàn
- Nguyên lý thiết kế
- Chuẩn mã hoá dữ liệu DES
- Phân tích hệ mật DES

# Khái niệm mã khối

- So sánh với mã đã học: stream cipher vs. block cipher

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

- TIN= 010100110111= (010)(100)(110)(111)
  - ➔ MÃ= 111 011 000 101 theo key=1
  - ➔ MÃ= 100 011 011 111 theo key=4
- Có 5 khóa,  $2^2 < 5 < 2^3$  nên cần 3 bit để biểu diễn ➔ kích thước khóa (và kích thước khối cùng) là 3.
- Nếu Eve tóm đc khối MÃ=001 sẽ suy ra TIN là 000 hoặc 101.

# Điều kiện cho an toàn mã khối

- Kích thước khối phải đủ lớn để chống lại các loại tấn công phá hoại bằng phương pháp thống kê.
  - Tuy nhiên cần lưu ý rằng kích thước khối lớn sẽ làm thời gian trễ lớn.
- Không gian khóa phải đủ lớn (tức là chiều dài khóa phải đủ lớn) để chống lại tìm kiếm vét cạn.
  - Tuy nhiên mặt khác, khóa cần phải đủ ngắn để việc làm khóa, phân phối và lưu trữ được hiệu quả.

# Nguyên tắc thiết kế

- Confusion (hỗn loạn) Sự phụ thuộc của Mã đối với TIN phải thật phức tạp để gây rắc rối hỗn loạn đối với kẻ thù có ý định tìm qui luật để phá mã.
  - Quan hệ hàm số của Mã với TIN nên là phi tuyến (non-linear).
- *Diffusion*. (khuếch tán) Làm khuếch tán những mẫu văn bản mang đặc tính thống kê (gây ra do dư thừa của ngôn ngữ) lẫn vào toàn bộ văn bản.
  - Nhờ đó tạo ra khó khăn cho kẻ thù trong việc dò phá mã trên cơ sở thống kê các mẫu lặp lại cao.
- Trong khi *confusion* được thực hiện bằng phép thay thế (substitution) thì *diffusion* được tạo ra bằng các phép chuyển đổi chỗ (transposition) hay hoán vị.

# Ví dụ: Phép hoán vị cột

- Để mã hóa TIN=“computer security”, viết lại thành nhiều hàng 5 cột

c o m p u  
t e r s e  
c u r i t  
y .

- Mã tạo ra bằng cách viết lại theo cột:

C T C Y O E U M R R P S I U E T

# Cài đặt

- Software: mềm dẻo, giá thành thấp.
- Hardware: nhanh.

- Study case:

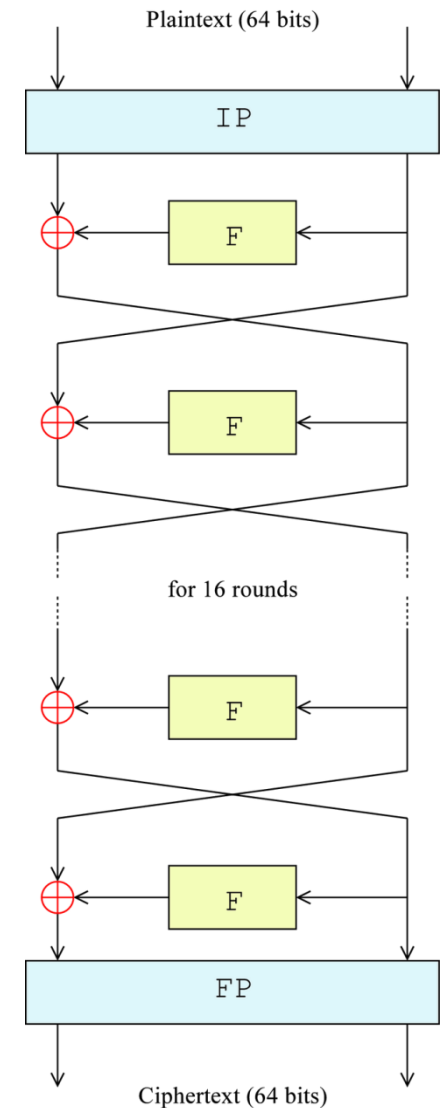
## Data Encryption Standard (DES) - 1977

- Hiện tại đã có chuẩn mới AES → topic bài tập lớn



# Khái niệm vòng lặp

- Các mã khối thường được xây dựng nhiều vòng lặp với mỗi vòng lặp cơ sở = việc thực hiện một hàm  $f$ .
  - đầu vào của một vòng lặp là đầu ra của vòng lặp trước và một khóa con phát sinh từ khóa đầy đủ dựa trên một thuật toán key-schedule.
- Giải mã sẽ là một quá trình ngược với các khóa con cho mỗi vòng sẽ được phát sinh theo thứ tự ngược.



The overall Feistel structure of DES

# Involution (đối hợp)

- Đặc biệt, hàm cơ sở vòng lặp  $f$  thông thường là một hàm có đặc tính đối hợp (involution), tức là nó bằng hàm ngược của nó:  $f = f^{-1}$  hay là  $f(f(x)) = x$ 
  - Ví dụ:  
 $x \in \{\text{tập các chuỗi nhị phân độ dài 3}\}$   
(bit thứ nhất và thứ hai đổi chỗ cho nhau, bit thứ ba giữ nguyên).
  - Như thế ta có  $f$  là một hàm xoay ốc, chẳng hạn cụ thể là  
 $f(101) = 011$   
 $f(f(101)) = 101$

# Lịch sử của DES

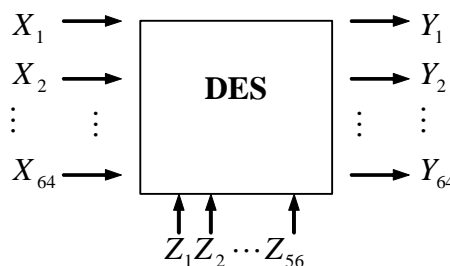
- Vào những năm đầu thập kỷ 70, nhu cầu có một chuẩn chung về thuật toán mã hóa đã trở nên rõ ràng:
  1. Sự phát triển của công nghệ thông tin và của nhu cầu an toàn & bảo mật thông tin.
  2. Các thuật toán ‘cây nhà lá vườn’ (ad hoc) không thể đảm bảo được tính tin cậy đòi hỏi.
  3. Các thiết bị khác nhau đòi hỏi sự trao đổi thông tin mã hóa.

# Tiêu chuẩn chung

1. Bảo mật ở mức cao
2. Thuật toán được đặc tả và công khai hoàn toàn, tức là tính bảo mật không được phép dựa trên những phần che giấu đặc biệt của thuật toán.
3. Việc cài đặt phải dễ dàng để đem lại tính kinh tế
4. Phải mềm dẻo để áp dụng được cho muôn vàn nhu cầu ứng dụng

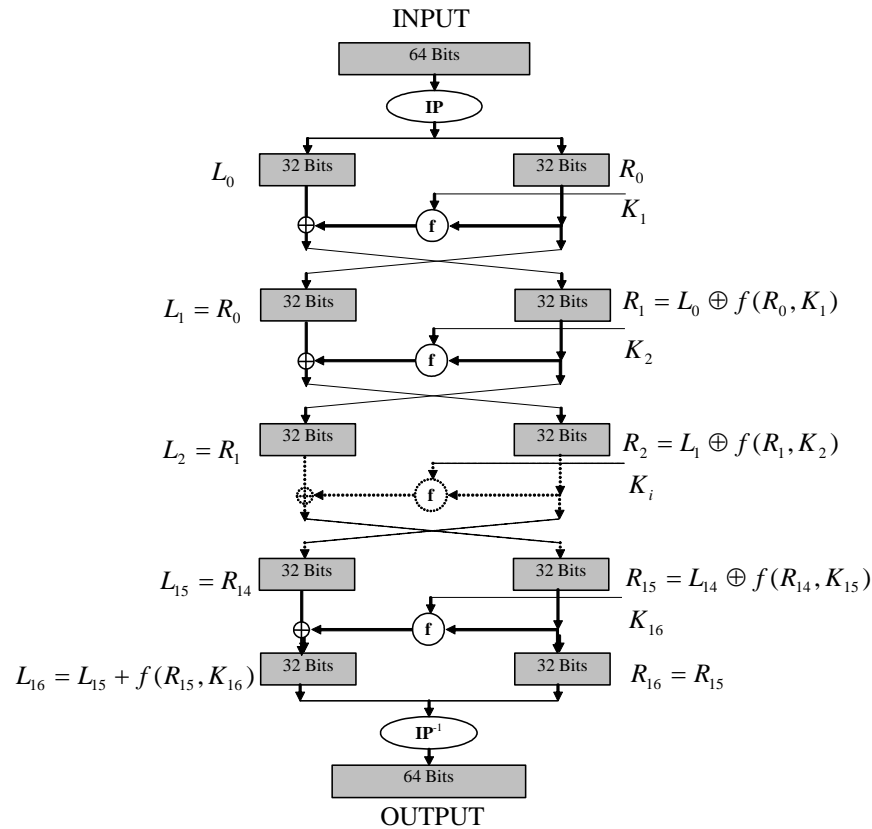
# DES

- Năm 1973, Cục quản lý các chuẩn quốc gia của Mỹ đã có văn bản cổ động cho các hệ thống mã hóa ở cơ quan đăng ký liên bang của Mỹ. Điều đó cuối cùng đã dẫn đến sự phát triển của Data Encryption Standard, viết tắt là DES.
  - DES, IBM, Lucifer.
  - dùng rộng rãi nhất, tranh cãi nhiều nhất
- Sơ đồ chung



- Đầu vào là khối độ dài 64 bits, đầu ra 64 bits và khóa là 56 bits.

# Cấu trúc DES: 16 vòng

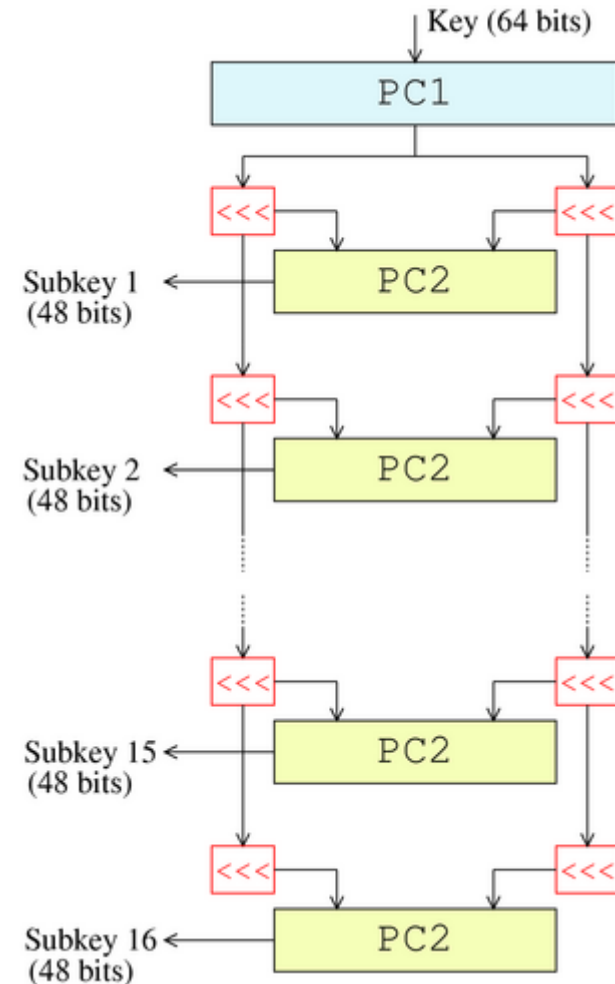


# Cấu trúc DES

- DES được cấu tạo bởi 16 bước lặp có cơ sở là hàm chuyển đổi phi tuyến  $f$ ;
- 16 bước lặp này được kẹp vào giữa hai tác tử giao hoán  $IP$  và  $IP^{-1}$ .
  - Hai tác tử này không có ý nghĩa gì về mặt mật mã mà hoàn toàn nhằm tạo điều kiện cho việc 'chip hóa' thuật toán DES.
- Hàm  $f$  là nguồn gốc của sức mạnh trong thuật toán DES này.
  - Sự lặp lại nhiều lần các bước lặp với tác dụng của  $f$  là nhằm tăng cường thêm mạnh lực của  $f$  về mặt lượng.

# Thuật toán sinh khóa con

- 16 vòng lặp của DES chạy cùng thuật toán như nhau nhưng với các khóa khác nhau, được gọi là các khóa con
  - sinh ra từ khóa chính của DES bằng một thuật toán sinh khóa con.
- Khóa chính K, 64 bit, qua 16 bước biến đổi, mỗi bước sinh 1 khóa con 48 bit.
- Thực sự chỉ có 56 bit của khóa chính được sử dụng
  - 8 parity bits, lọc ra qua PC1.
  - Các bộ biến đổi PC1 và PC2 là các bộ vừa chọn lọc vừa hoán vị.
  - R1 và R2 là các phép đẩy bit trái 1 và hai vị trí.





# Cấu trúc vòng lặp DES

- Mỗi vòng lặp của DES thực hiện trên cơ sở công thức sau:

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

- Ta cũng có thể viết lại

$$(L_i, R_i) = T \bullet F (L_{i-1}, R_{i-1})$$

- Trong đó  $F$  là phép thay thế  $L_{i-1}$  bằng  $L_{i-1} \oplus f(R_{i-1}, K_i)$
- $T$  là phép đổi chỗ hai thành phần  $L$  và  $R$ .
- Tức là mỗi biến đổi vòng lặp của DES có thể coi là một tích hàm số của  $F$  và  $T$  (trừ vòng cuối cùng không có  $T$ ).
- Viết lại toàn bộ **thuật toán sinh mã DES** dưới dạng công thức:

$$\text{DES} = (\text{IP})^{-1} \bullet F_{16} \bullet T \bullet F_{15} \bullet T \bullet \dots \bullet F_2 \bullet T \bullet F_1 \bullet (\text{IP})$$

# Thuật toán giải mã DES

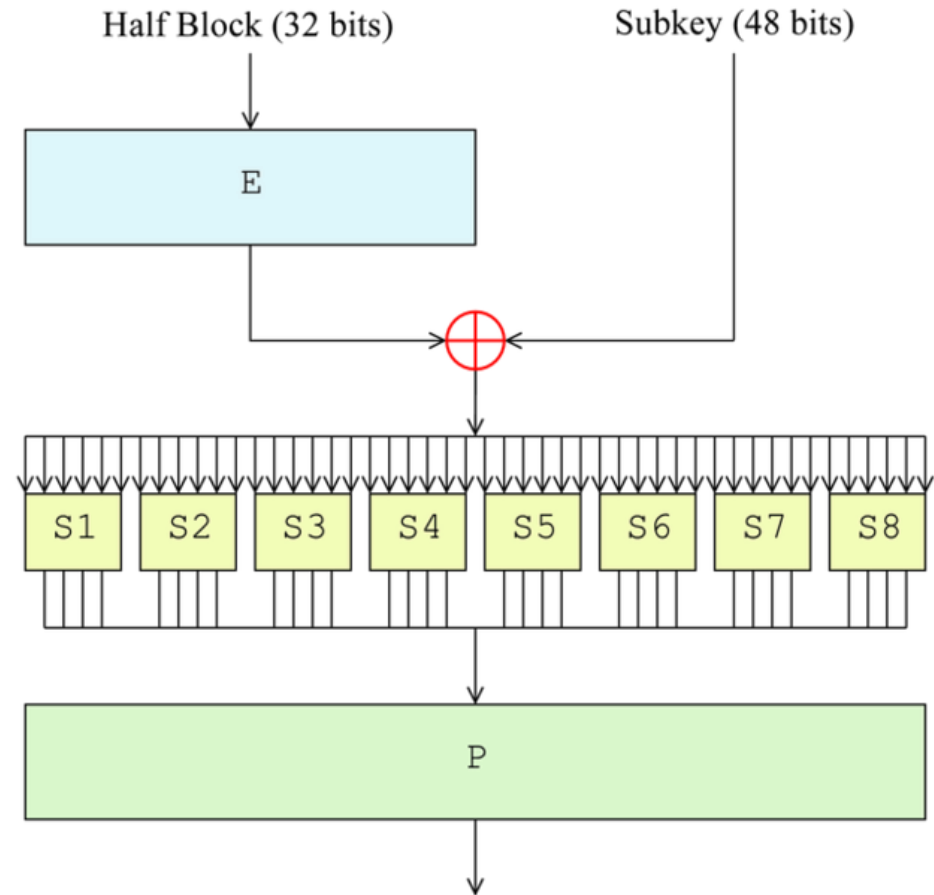
- Giống hệt như thuật toán sinh mã nhưng có các khóa con được sử dụng theo thứ tự ngược lại
  - Vì vậy, thuật toán giải mã có thể được viết lại dưới dạng công thức sau:

$$\text{DES}^{-1} = (\text{IP})^{-1} \bullet F_1 \bullet T \bullet F_2 \bullet T \bullet \dots \bullet F_{15} \bullet T \bullet F_{16} \bullet (\text{IP})$$

- Chú ý rằng mỗi hàm T hoặc F đều là các hàm có tính chất đối hợp ( $f=f^{-1}$ , hay  $f(f(x))=x$ )  $\rightarrow$  thực hiện  $\text{DES} \bullet \text{DES}^{-1}$  sẽ thu được phép đồng nhất.
  - Điều đó giải thích tại sao thuật toán giải mã lại giống hệt như sinh mã chỉ có khác thứ tự dùng khóa con.

# Cấu trúc hàm f

- 32 bit của  $R_{i-1}$  được mở rộng thành 48 bit thông qua E rồi đem XOR với 48 bit của  $K_i$ .
- 48 bit kết quả sẽ được phân thành 8 nhóm 6 bit; mỗi nhóm này sẽ qua một biến đổi đặc biệt, S-box, và biến thành 4 bit.
  - có 8 S-box khác nhau ứng với mỗi nhóm 6 bit
- 32 bit hợp thành từ 8 nhóm 4 bit (sau khi qua các S-box) sẽ được hoán vị lại theo P rồi đưa ra kết quả cuối cùng của hàm f ( $F_i$ ).



# Cấu trúc của các S-Box

- Mỗi S-box như một bộ biến đổi gồm 4 bảng biến đổi, mỗi bảng biến đổi 1 đầu vào 4 bit thành đầu ra cũng 4 bit (bảng 16 dòng).
  - Đầu vào 4 bit chính là lấy từ các bit 2-5 của nhóm 6 bit.
  - Các bit 1 và 6 sẽ dùng để xác định 1 trong 4 bảng biến đổi của S-box. Vì thế chúng được gọi là các bit điều khiển (CL và CR: left control và right control bit).

S <sub>5</sub>		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

# Các thuộc tính của S-Box

- Các nguyên tắc thiết kế của 8 S-boxes được đưa vào lớp ‘Classified information’ ở Mỹ.
- NSA đã tiết lộ 3 thuộc tính của S-boxes, những thuộc tính này bảo đảm tính confusion & diffusion của thuật toán.
  1. Các bit ra (output bit) luôn phụ thuộc không tuyến tính vào các bit vào (input bit).
  2. Sửa đổi ở một bit vào làm thay đổi ít nhất là hai bit ra.
  3. Khi một bit vào được giữ cố định và 5 bit còn lại cho thay đổi thì S-boxes thể hiện một tính chất được gọi là ‘phân bố đồng nhất’ (uniform distribution): so sánh số lượng bit số 0 và 1 ở các đầu ra luôn ở mức cân bằng.
    - Tính chất này khiến cho việc áp dụng phân tích theo lý thuyết thông kê để tìm cách phá S-boxes là vô ích.

# Các thuộc tính của S-Box (2)

- 3 tính chất này đảm bảo tốt confusion & diffusion.
  - Sau 8 vòng lặp tất cả các bit ra của DES sẽ chịu ảnh hưởng của tất cả các bit vào và tất cả các bit của khóa.
- Tuy nhiên cấu tạo của S-box đã gây tranh luận mạnh mẽ từ hàng thập kỷ qua về khả năng cơ quan NSA (National Security Agency), Mỹ, vẫn còn che giấu các một số đặc tính của S-box hay cài bên trong những cửa bẫy (trapdoor) mà qua đó họ có thể dễ dàng phá giải mã hơn người bình thường.

# Điểm yếu của DES

- Ký hiệu  $\bar{k}$  là đảo bit của khoá  $k$ .
- Tính chất sau của DES:

$$\text{Nếu } y = DES_k(x) \text{ thì } \bar{y} = DES_{\bar{k}}(\bar{x})$$

giúp ta có thể tính  $\bar{y}$  từ  $x$  và  $k$ .

- Để tấn công vét cạn DES, ta chỉ cần thử một nửa số khoá  $k$  trong không gian khoá.

# Khoá yếu

- Khoá  $k$  gọi là yếu nếu các khoá con  $k_i$  sinh từ  $k$  giống hệt nhau:

$$k_1 = k_2 = \dots = k_{16}$$

- Các khoá này làm cho việc mã hoá và giải mã giống hệt nhau:

$$DES_k(\cdot) = DES_k^{-1}(\cdot)$$

- Ví dụ, bốn khoá sau đây là yếu:

0x0101010101010101, 0xFEFEFEFEFEFEFEFEFE

0xE0E0E0E0F1F1F1F1, 0x1F1F1F1F0E0E0E0E



# Các chế độ mã khối

# Nội dung

- Mã khối lý tưởng
- Chế độ ECB
- Mã hoá xác suất
- Chế độ CBC
- Một số chế độ mã khối dựa trên mã dòng

# Nội dung

- **Mã khối lý tưởng**
- Chế độ ECB
- Mã hoá xác suất
- Chế độ CBC
- Một số chế độ mã khối dựa trên mã dòng

# Mã khối lý tưởng

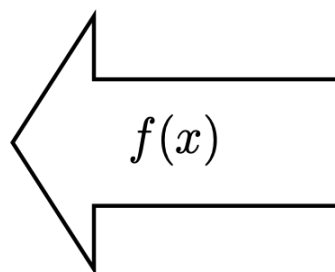
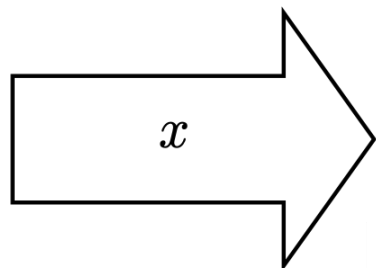
- Trên thực tế, người ta xem AES hoặc 3DES như hệ mã khối lý tưởng  $E(k, x)$ .

- Tức là, với mỗi khoá  $k$ , ánh xạ


$$F_k(x) = E(k, x)$$

là một hoán vị ngẫu nhiên độc lập.

# Hoán vị ngẫu nhiên



$x$	$f(x)$
00101	10101
11111	01110
10111	01011
00011	10001



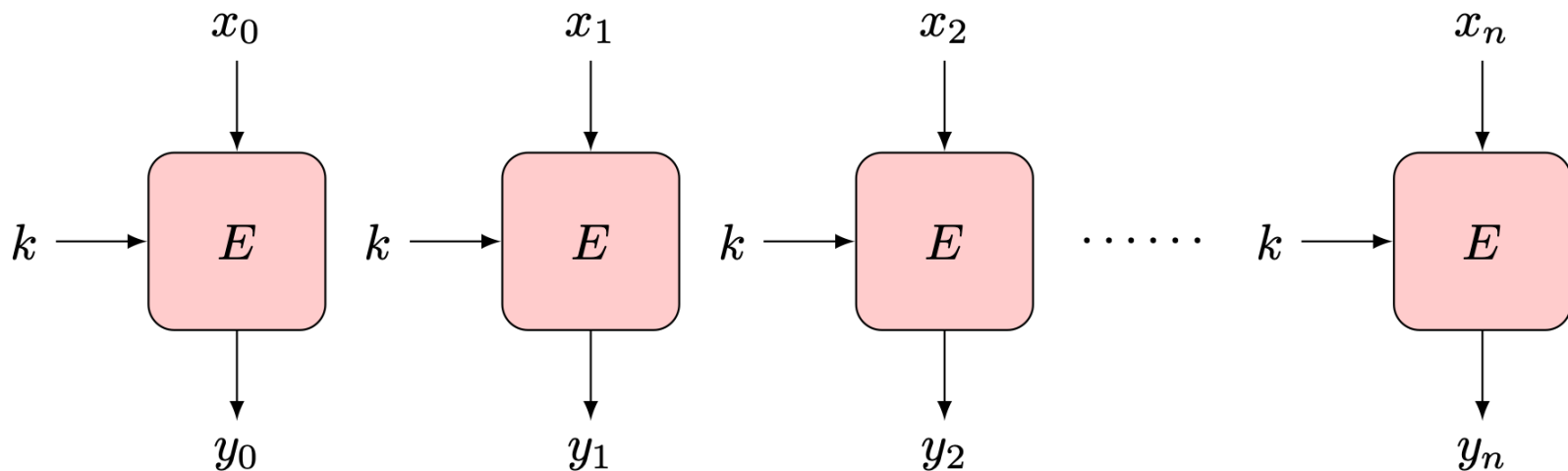
# Các chế độ sử dụng

- **Câu hỏi:** Làm thế nào để mã hoá thông điệp với độ dài bất kỳ? (dùng AES hoặc 3DES)
- **Trả lời:** Dùng một trong các chế độ sau:
  - “ECB” = “Electronic code book”
  - “CTR” = “Counter mode”
  - “CBC” = “Cipher Block Chaining”
  - “OFB” = “Output Feedback” • v.v.
  - v.v.

# Nội dung

- Mã khối lý tưởng
- **Chế độ ECB**
- Mã hoá xác suất
- Chế độ CBC
- Một số chế độ mã khối dựa trên mã dòng

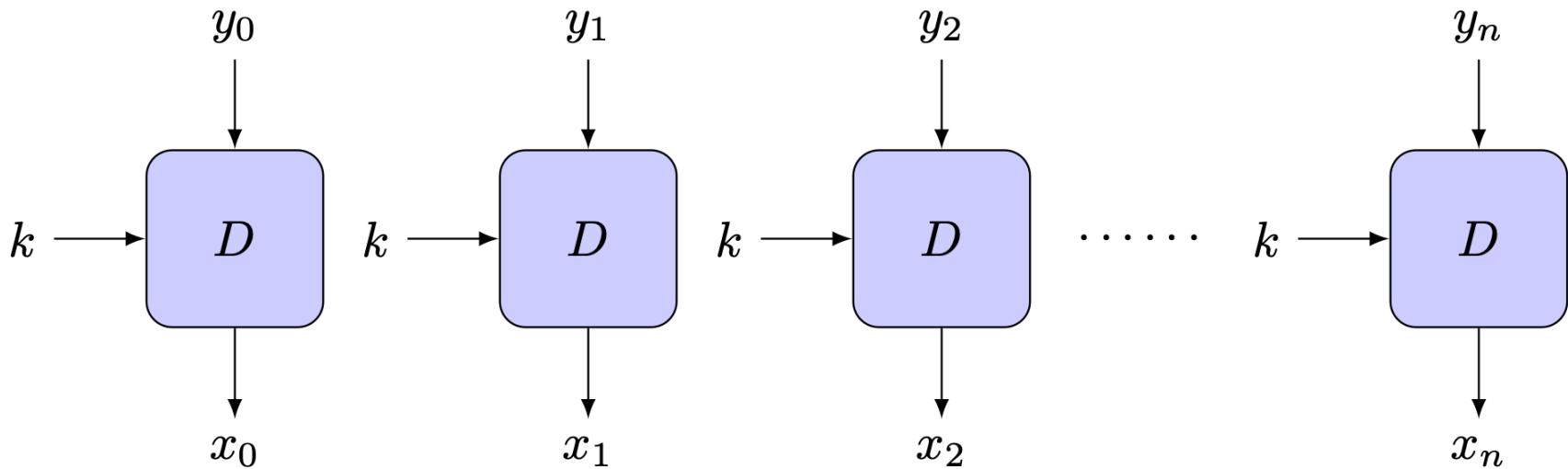
# Chế độ ECB (Electronic code book)



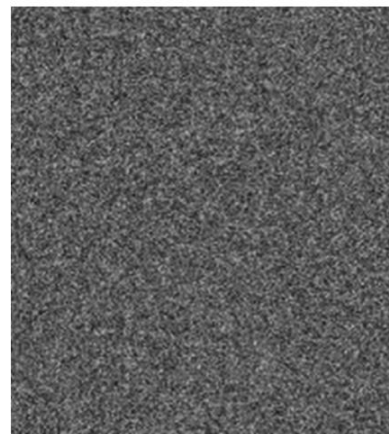
- Dữ liệu được chia thành các khối khối  $b$  bit, với  $b$  = kích thước khối.
- Với dữ liệu không chia hết cho  $b$  bit: Thêm dãy “10..0” để độ dài thông điệp chia hết cho  $b$ .
- Phép toán padding này cho có tính khả nghịch. Nó cho phép giải mã.



# ECB: giải mã



# ECB không an toàn



Hình: Bên trái là Bản rõ. Ở giữa là chế độ ECB. Bên phải là Mã hoá an toàn

- **Vấn đề:** Nếu  $x_i = x_j$  thì  $y_i = y_j$
- ECB chỉ an toàn khi mã hoá dữ liệu ngẫu nhiên (Ví dụ, mã hoá các khoá).

# Ví dụ: Chuyển tiền giữa hai ngân hàng

Block #	1	2	3	4	5
	Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount \$

1. Giả sử: kích thước mỗi trường là  $n$ -bit (ví dụ 128 bit)
2. Giả sử: khoá  $k_{AB}$  để trao đổi thông tin giữa hai ngân hàng là cố định

# Oscar tấn công

Block #	1	2	3	4	5
	Sending Bank A	Sending Account #	Receiving Bank B	Receiving Account #	Amount \$

1. Oscar mở một tài khoản tại ngân hàng  $A$  và một tài khoản tại ngân hàng  $B$ ;
2. Oscar chuyển nhiều lần  $1\$$  từ tài khoản của anh ta ở ngân hàng  $A$  sang tài khoản ở ngân hàng  $B$ ;
3. Oscar bắt gói tin trên đường truyền và nhận được các bản mã giống nhau

$$B_1 \parallel B_2 \parallel B_3 \parallel B_4 \parallel B_5$$

và anh ta giữ lại bản mã  $B_4$

4. Trong tương lai, mỗi khi thấy lệnh chuyển tiền từ  $B_1$  tới  $B_3$ , thay block thứ 4 bởi  $B_4$

# Nội dung

- Mã khối lý tưởng
- Chế độ ECB
- **Mã hoá xác suất**
- Chế độ CBC
- Một số chế độ mã khối dựa trên mã dòng

# Mã hoá xác suất

- Mã hóa hai lần của cùng một thông điệp sẽ cho hai bản mã khác nhau
- Bản mã phải dài hơn bản rõ
- Nói một cách nôm na:

**Kích thước bản mã =  
Kích thước bản rõ + “dãy bit ngẫu nhiên”**

# Bài tập

- Hãy viết hàm giải mã cho hàm mã hoá Enc được định nghĩa bởi

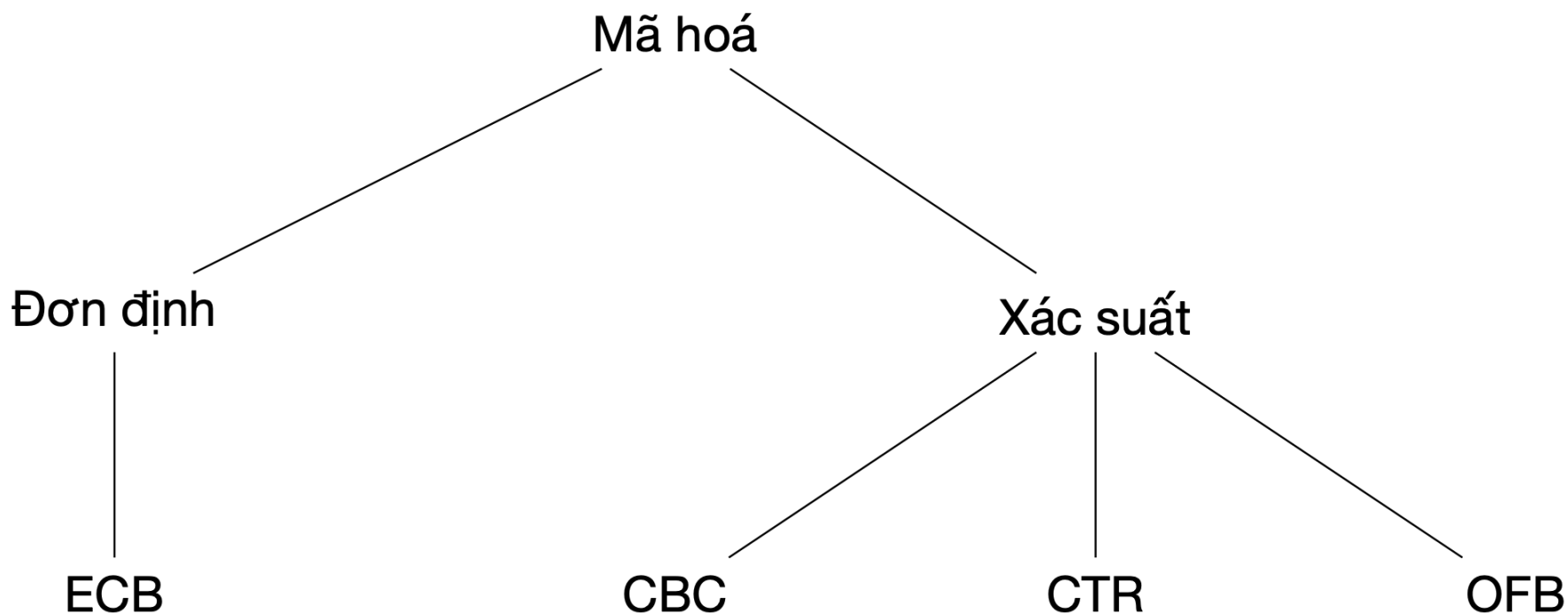
$\text{Enc}(k, m):$

$r = \text{random}()$

$c = \text{AES}(k, r) \oplus m$

return  $(r, c)$

# Dạng mã hoá

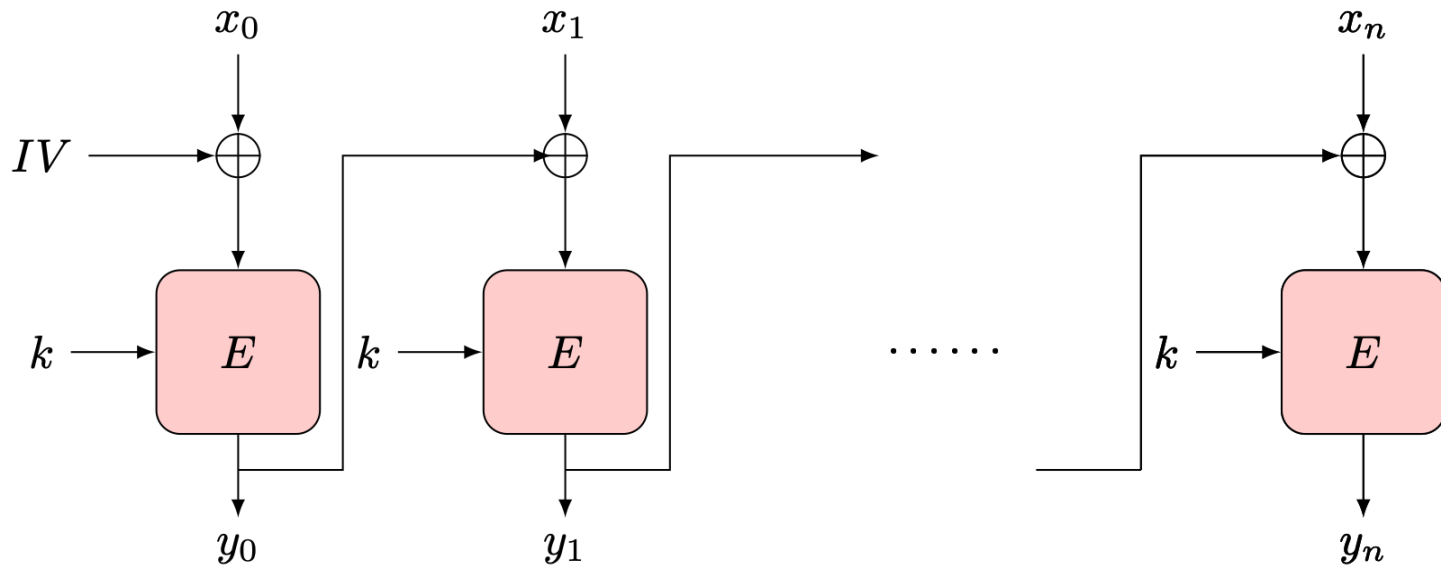




# Nội dung

- Mã khối lý tưởng
- Chế độ ECB
- Mã hoá xác suất
- **Chế độ CBC**
- Một số chế độ mã khối dựa trên mã dòng

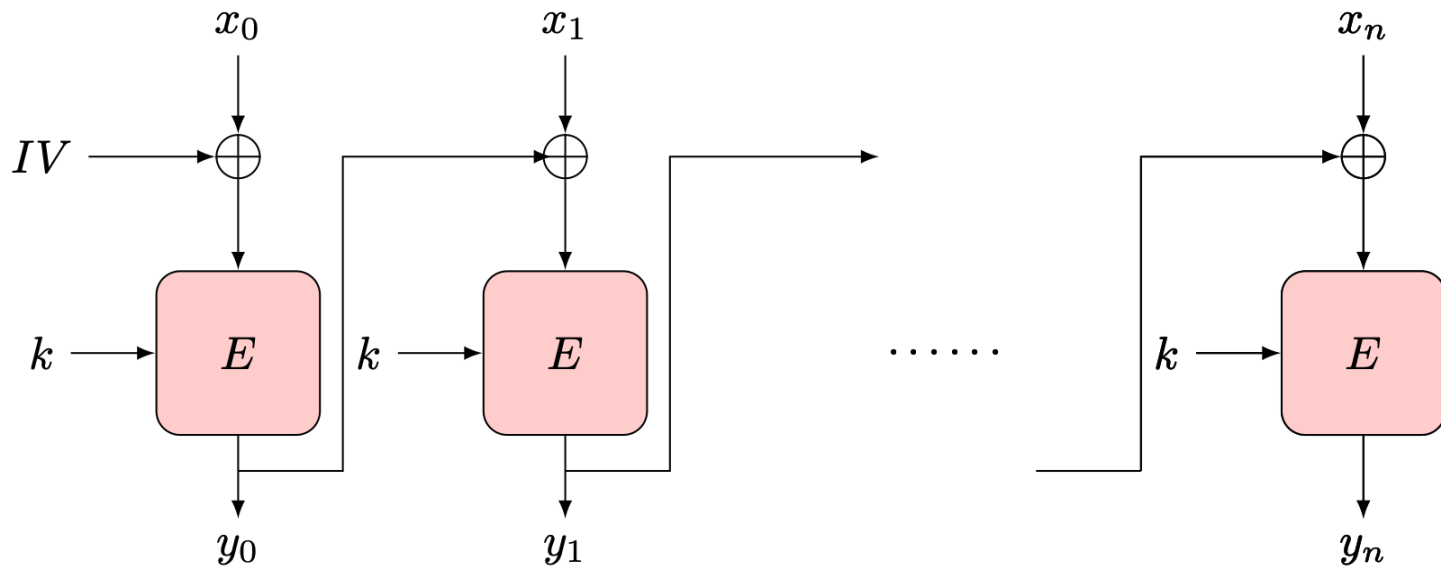
# Chế độ CBC



**Thuật toán.** Chọn IV (“initialization value”) một cách ngẫu nhiên, sau đó dùng  $y_i$  như “IV” cho  $x_{i+1}$ . Gửi IV cùng với bản mã

$$IV \parallel y_0 \parallel y_1 \parallel \dots \parallel y_n$$

# CBC: công thức đại số



- $y_{-1} = IV$

// Khởi tạo

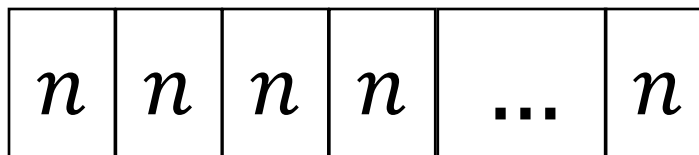
- $y_i = E_k(y_{i-1} \oplus x_i)$  với  $i = 0, 1, \dots$

# Sử dụng IV thế nào?

- IV không cần giữ bí mật
- Nhưng phải là “nonce” = “number used only once”
- **Ví dụ:** IV có thể là
  - ✓ ngẫu nhiên “thật”
  - ✓ bộ đếm “counter” (phải được lưu trữ bởi Alice)
  - ✓  $ID_A \parallel ID_B \parallel time$

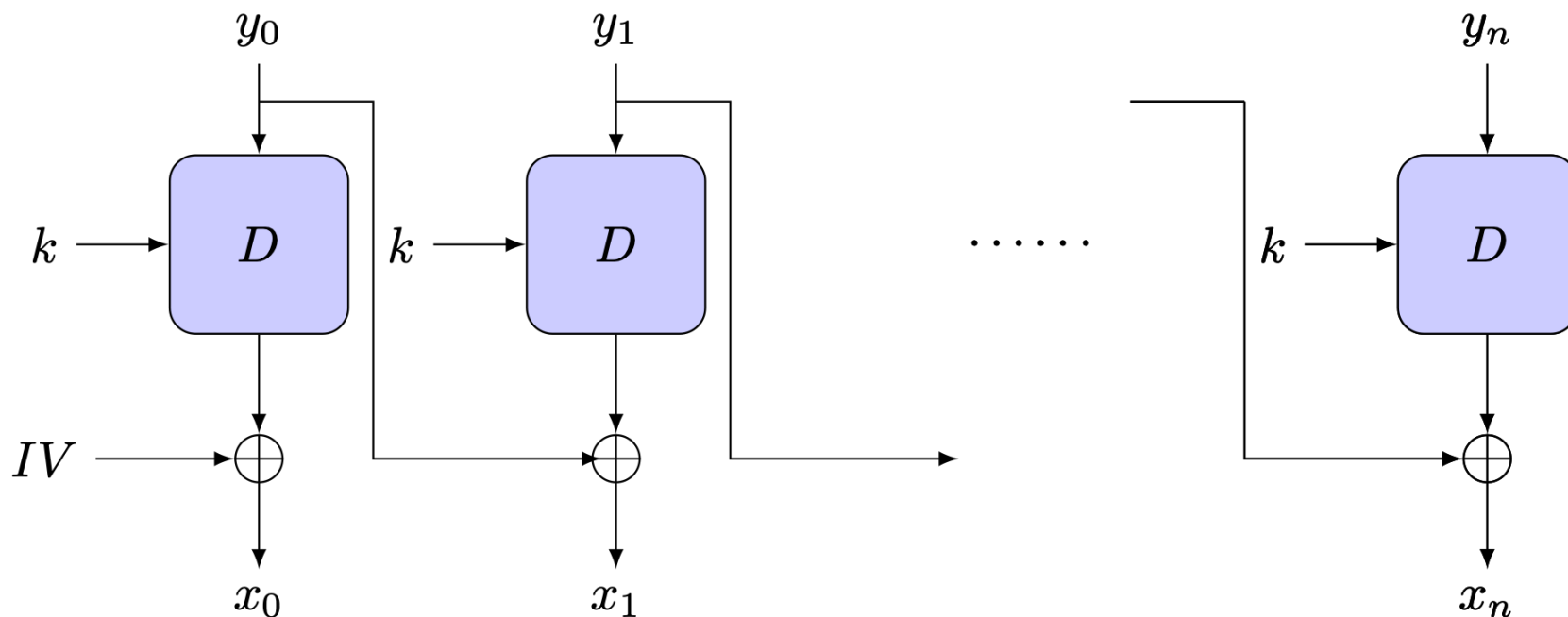
# Padding cho CBC

- Padding  $n$  byte, với  $n > 0$ ,

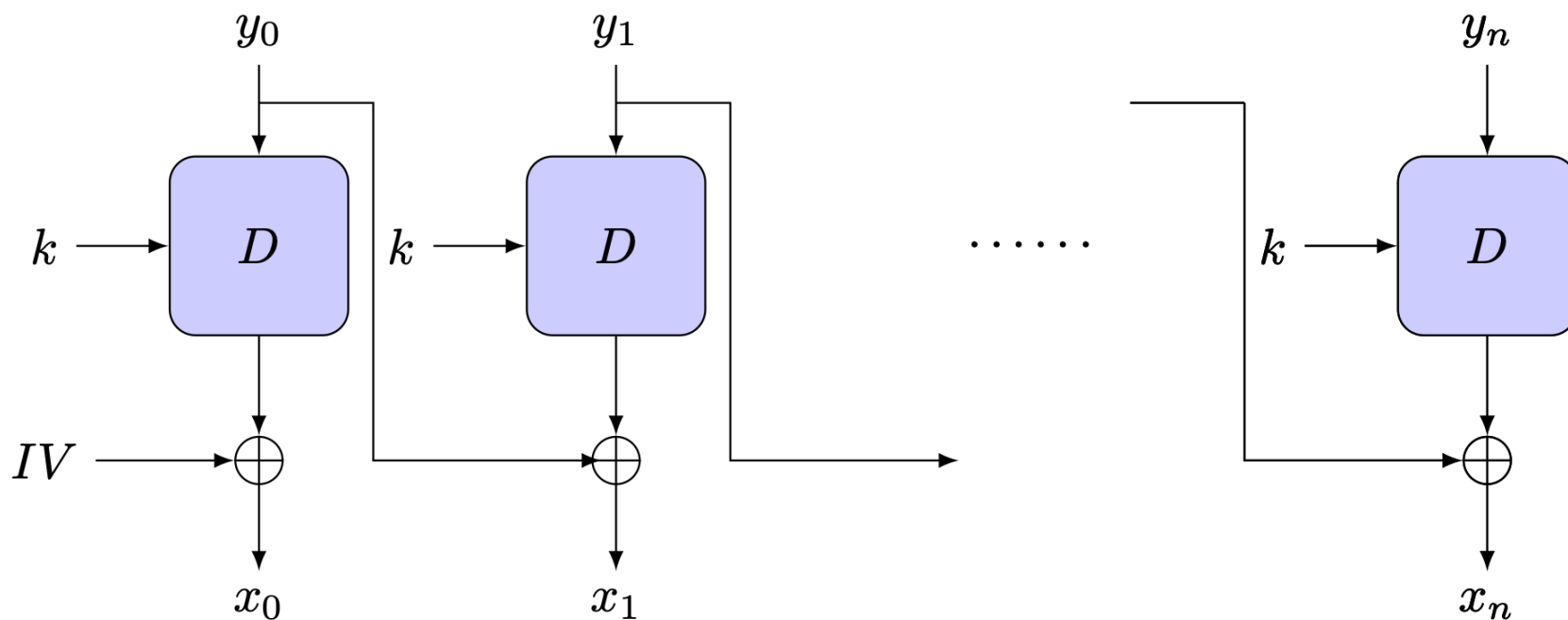


- Nếu không cần pad, thêm một khối giả
- Khi giải mã, loại bỏ pad.

# CBC: giải mã



# Bài tập



- Hãy viết công thức đại số cho mạch giải mã của chế độ CBC.

# Nội dung

- Mã khối lý tưởng
- Chế độ ECB
- Mã hoá xác suất
- Chế độ CBC
- **Một số chế độ mã khối dựa trên mã dòng**



# Mã dòng

- Sử dụng một hàm sinh số giả ngẫu nhiên

$$G: \mathcal{K} \rightarrow \{0,1\}^n,$$

là hàm đơn định từ không gian khoá đến dãy bit độ dài  $n$

- Mã hoá  $y = E_k(x) = G(k) \oplus x$
- Giải mã  $x = D_k(y) = G(k) \oplus y$

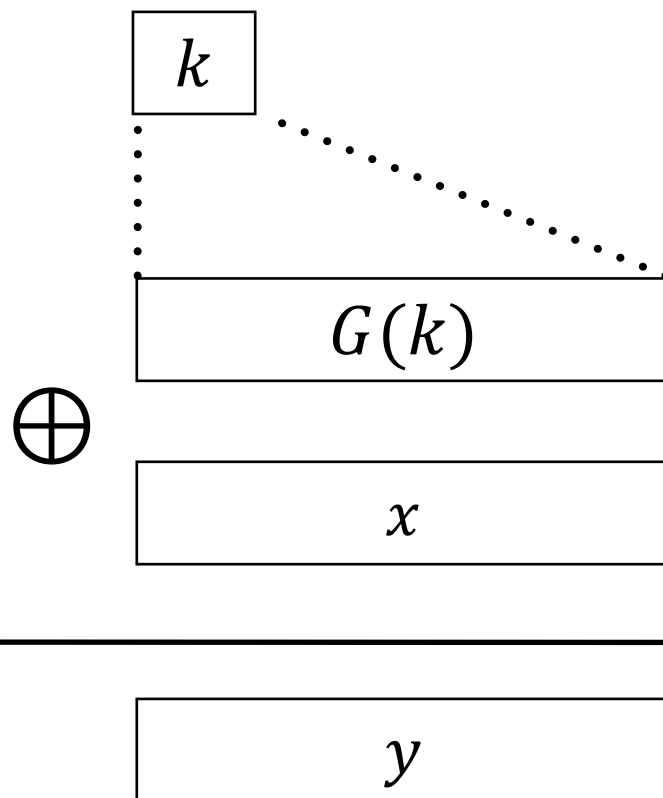
# Mã dòng

- Mã hoá

$$y = E_k(x) = G(k) \oplus x$$

- Giải mã

$$x = D_k(y) = G(k) \oplus y$$



# Mã dòng và mã khối

- Các chế độ mã khối trong mục này đều dựa trên nguyên lý của hệ mã dòng: *mã khối an toàn được dùng xây dựng các hàm sinh số giả ngẫu nhiên*

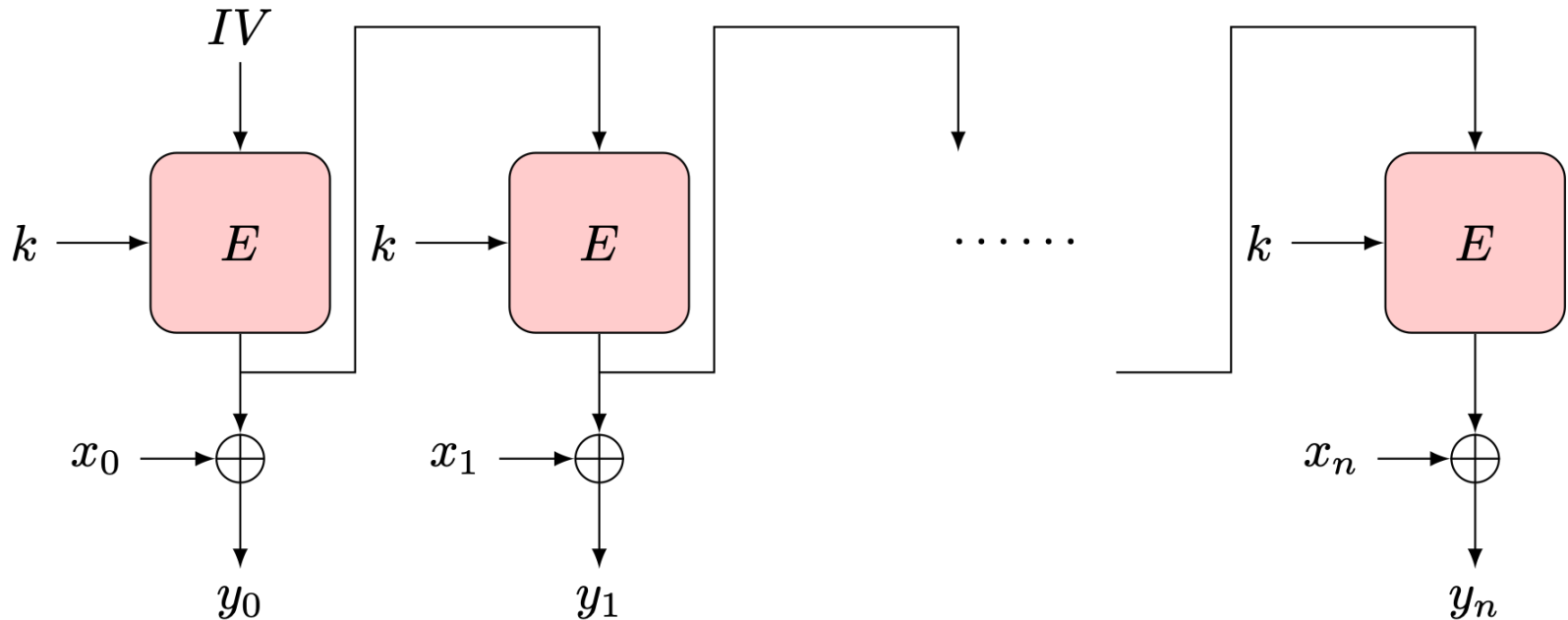
- **Ví dụ:**

$$G(k) = E_k(0) \parallel E_k(1) \parallel \cdots \parallel E_k(n)$$

- Hàm mã hoá và giải mã của mã dòng đều giống nhau

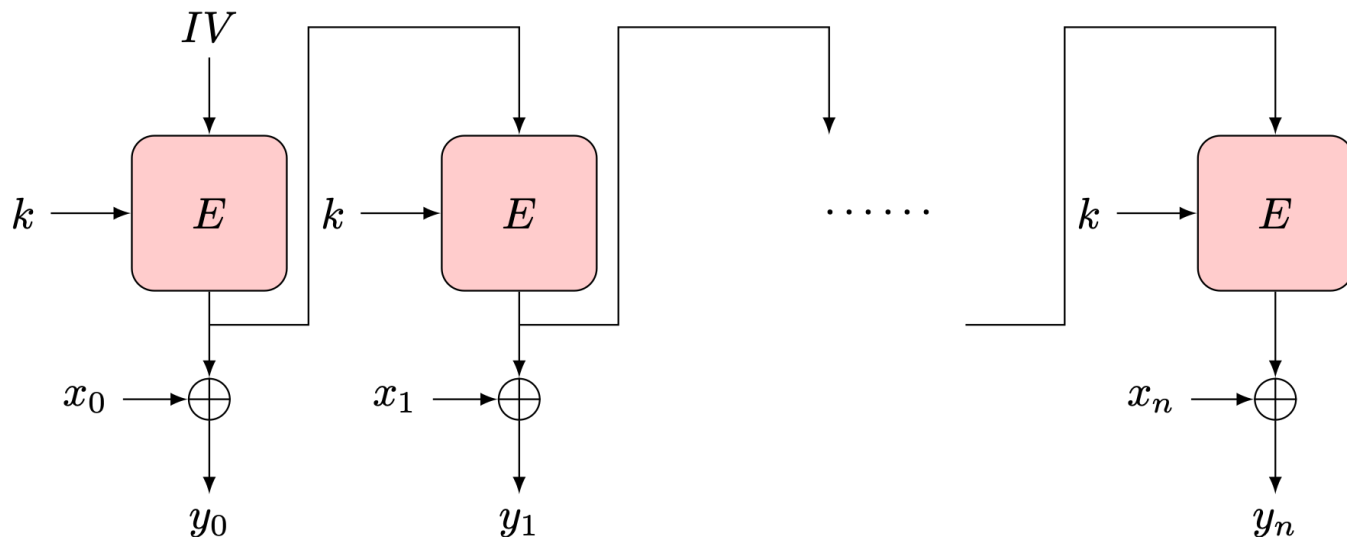
$$D_k(z) = E_k(z) = G(k) \oplus z$$

# Chế độ Output Feedback (OFB)



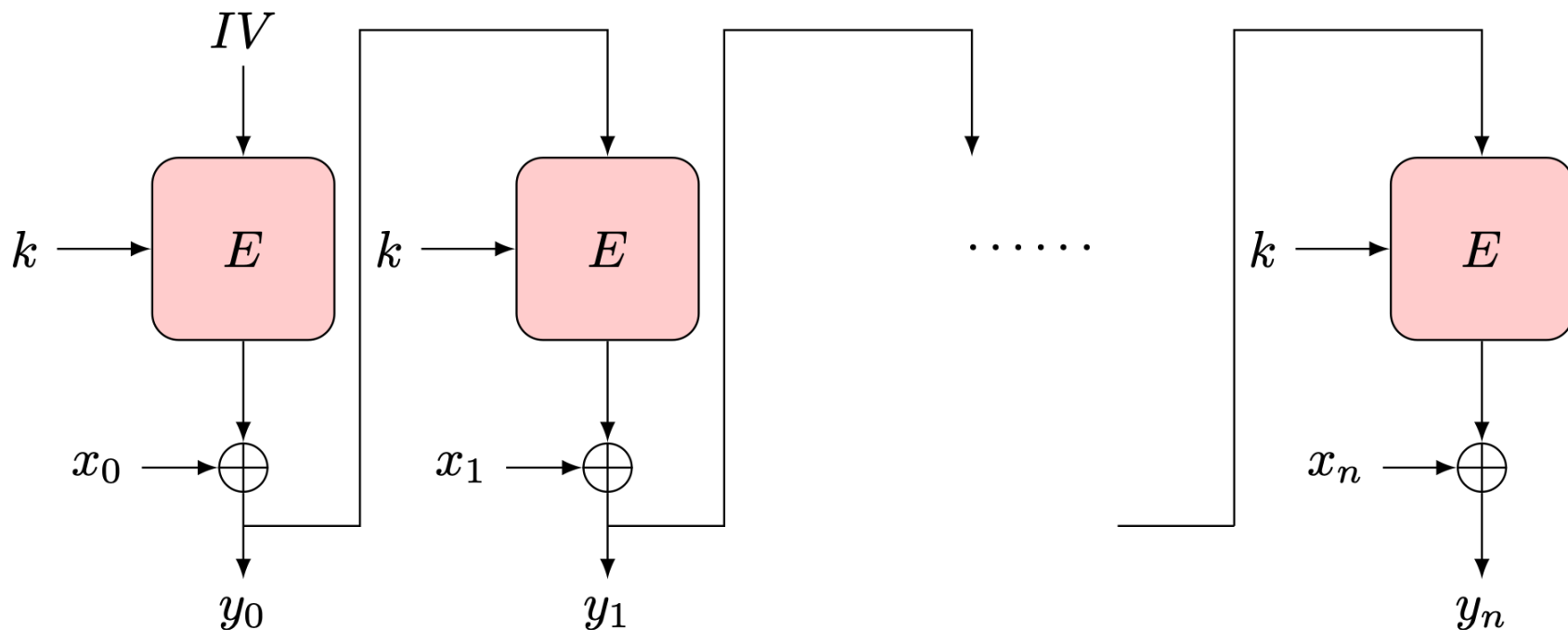
- Sử dụng  $IV$  ngẫu nhiên truyền cùng bản mã
- Không cần padding

# OFB: công thức đại số

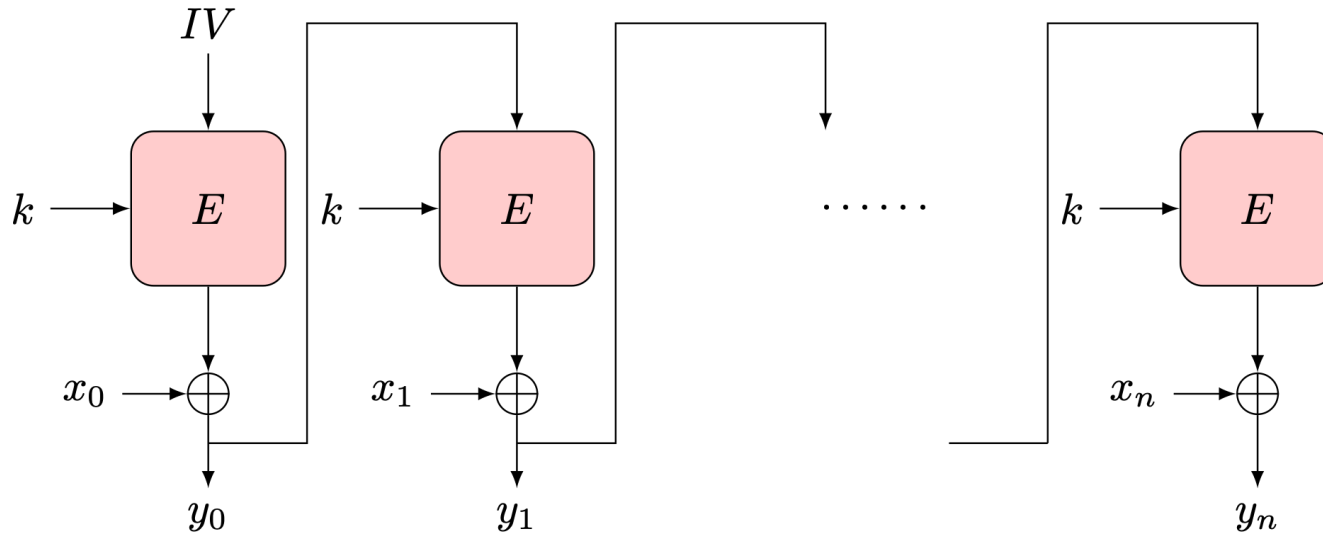


- $s_{-1} := IV$  // Khởi tạo
- $s_i := E_k(s_{i-1})$  // Khối bit giả ngẫu nhiên
- $y_i := s_i \oplus x_i$  với  $i = 0, 1, 2, \dots$

# Chế độ Cipher Feedback (CFB)

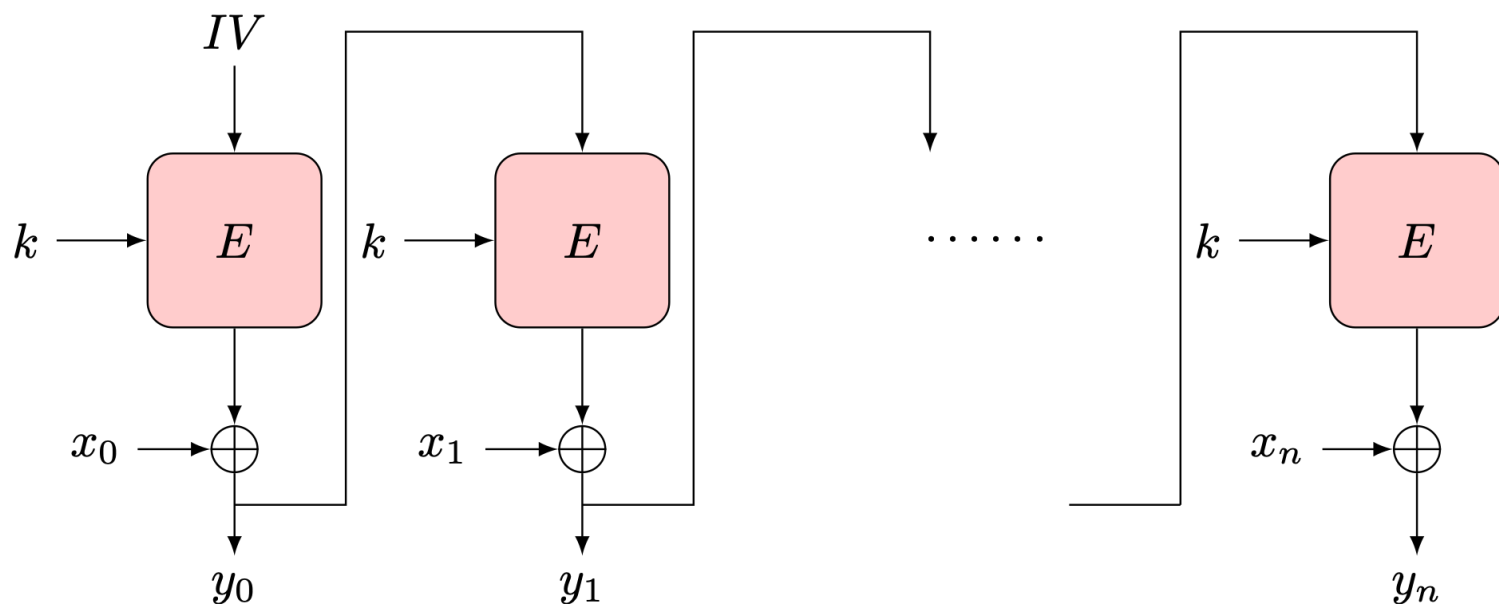


# CFB: công thức đại số



- $y_{-1} := IV$  // Khởi tạo
- $s_i := E_k(y_{i-1})$  // Khối bit giả ngẫu nhiên
- $y_i := s_i \oplus x_i$  với  $i = 0, 1, 2, \dots$

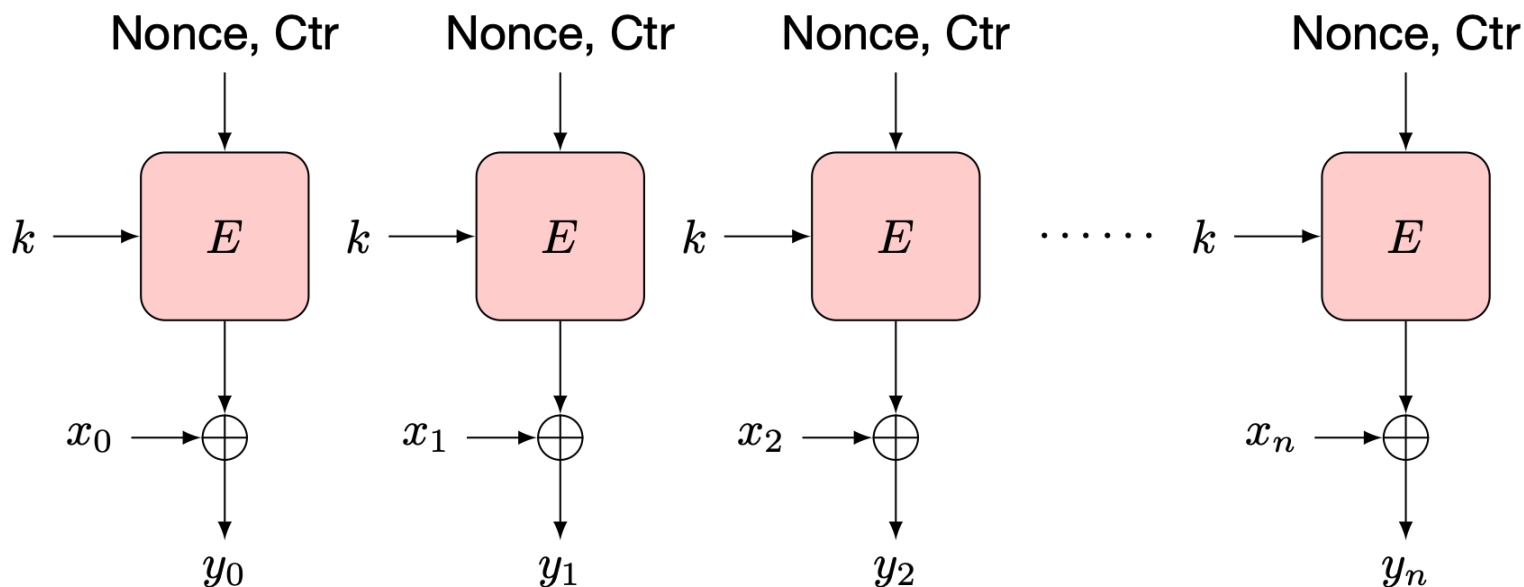
# Bài tập



- Hãy mô tả mạch giải mã ở dạng công thức đại số cho chế độ CFB.

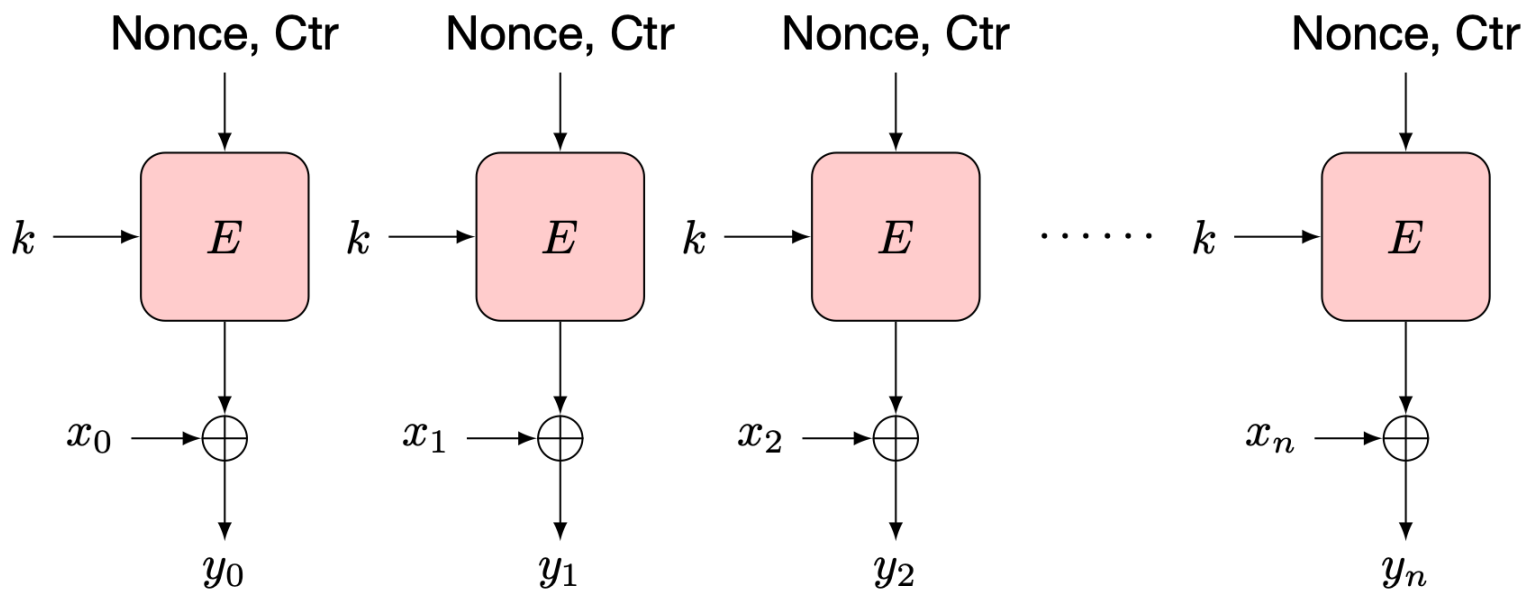


# Chế độ Counter (CTR)



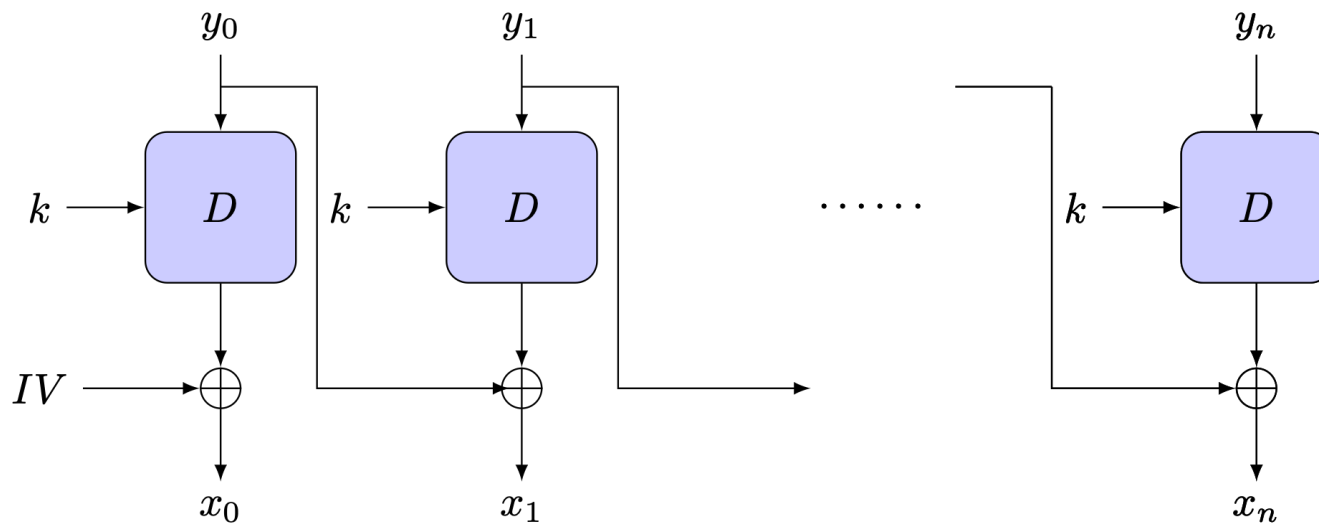
- Đảm bảo cặp **Nonce, Ctr** cặp không bao giờ lặp lại.
- Ctr được bắt đầu từ 0 cho mỗi thông điệp; và tăng ( **$\text{Ctr} = \text{Ctr} + 1$** ) sau mỗi khối của thông điệp.

# Bài tập



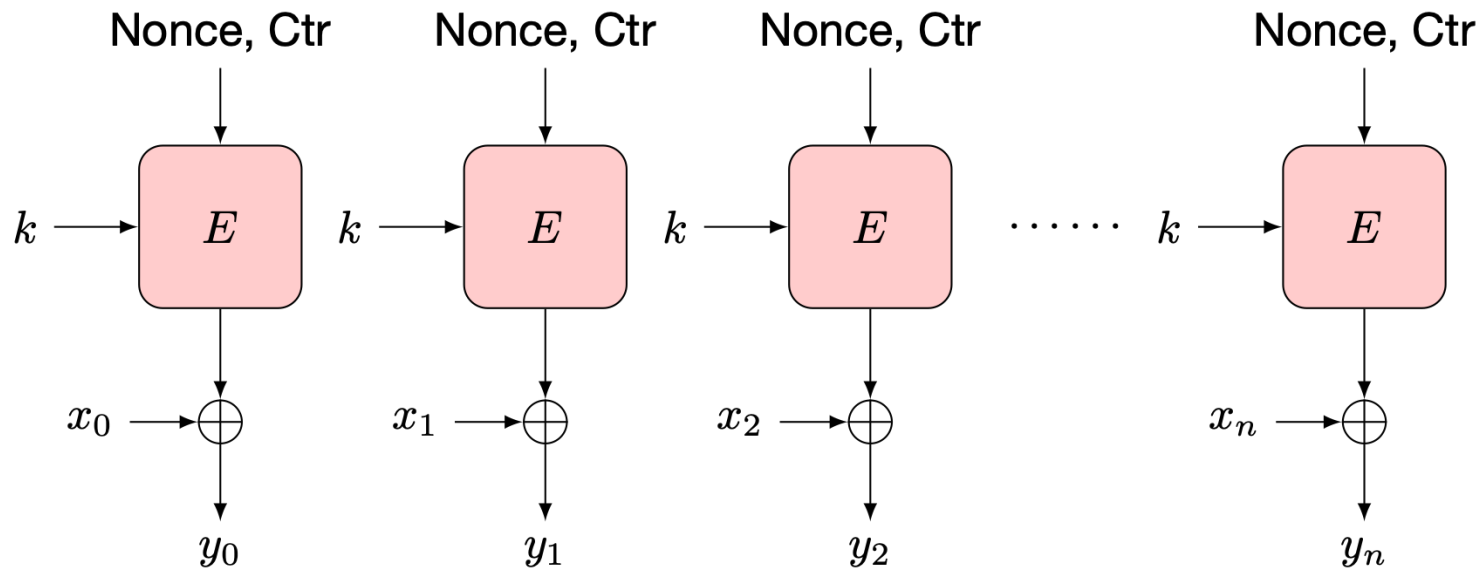
- Hãy mô tả mạch giải mã cho chế độ CTR.

# Bài tập



- Xét thông điệp  $x$  gồm  $\ell$  khối AES (ví dụ  $\ell = 100$ ). Alice mã hóa  $x$  dùng chế độ **CBC** và truyền bản mã kết quả tới Bob.
- Do mạng lỗi, khối bản mã số  $\ell/2$  bị mất trong khi truyền. Mọi bản mã khác được truyền và nhận đúng.
- Khi Bob giải mã bản mã nhận được, bao nhiêu khối bản rõ sẽ bị mất?

# Bài tập



- Xét thông điệp  $x$  gồm  $\ell$  khối AES (ví dụ  $\ell = 100$ ). Alice mã hóa  $x$  dùng chế độ **CTR (với Nonce ngẫu nhiên)** và truyền bản mã kết quả tới Bob.
- Do mạng lỗi, khối bản mã số  $\ell/2$  bị mất trong khi truyền. Mọi bản mã khác được truyền và nhận đúng.
- Khi Bob giải mã bản mã nhận được, bao nhiêu khối bản rõ sẽ bị mất?



25 YEARS ANNIVERSARY  
**SOICT**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you for  
your attentions!**



[soict.hust.edu.vn/](http://soict.hust.edu.vn/)



[fb.com/groups/soict](https://fb.com/groups/soict)

