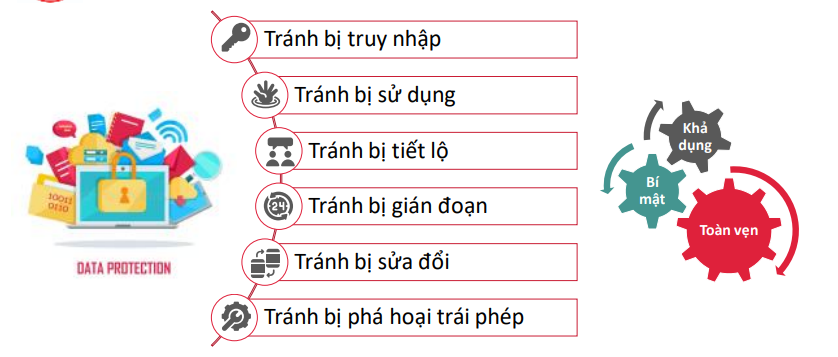
**CƠ SỞ LÝ THUYẾT MẬT MÃ HỌC**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# TỔNG QUAN MẬT MÃ HỌC

#### An toàn thông tin

###### Khái niệm : An toàn thông tin được hiểu là hành động phòng ngừa, ngăn cản sự truy cập, tiết lộ, chia sẻ, phát tán, ghi lại, sử dụng hoặc phá hủy những thông tin khi chưa được sự cho phép của chủ sở hữu.



###### Các mối đe doạ : bên trong, bên ngoài, kết hợp.

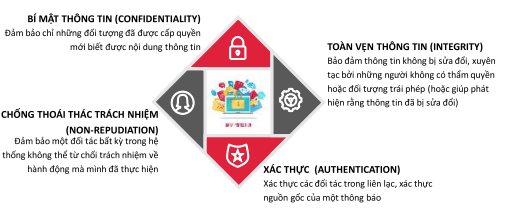
#### Một số vấn đề trong bảo vệ thông tin

###### Bí mật thông tin : đảm bảo đối tượng được cấp quyền mới biết được nội dung thông tin.

###### Toàn vẹn thông tin : đảm bảo thông tin không bị sửa đổi, xuyên tạc bởi người không có thẩm quyền hoặc đối tượng truy cập trái phép.

###### Chống thoái thác trách nhiệm : đảm bảo đối tác bất kỳ trong hệ thống không thể từ chối trách nhiệm về hành động của mình đã thực hiện.

###### Xác thực : xác thực các đối tượng trong liên lạc, xác thực nguồn gốc của một thông báo.



#### Thuật toán mật mã cơ bản

###### Mã hoá

###### Hàm băm, mã xác thực thông điệp

###### Chữ ký số

#### Quan hệ giữa dịch vụ ATTT & thuật toán



#### Sơ đồ hệ thống truyền tin số



#### Định nghĩa hình thức của toán học của hệ mật

##### Định nghĩa

###### **P** (plain text) : Dữ liệu gốc / Tập hữu hạn các bản rõ có thể.

###### **C** (cipher text) : Dữ liệu đã mã hoá / Tập hữu hạn các bản mã có thể.

###### **K** (key) :Chìa khoá mã-giải mã / Tập hữu hạn các khoá có thể

###### **E** (encrypt : mã hoá) : tập E­k : k **K,** với Ek là quy tắc mã hoá P C

###### **D** (decrypt : giải mã) : tập Dk : k **K**, với Dk là quy tắc giải mã Ek(P) P

|  |
| --- |
| Hệ mật : Một hệ mật gồm bộ 5 (**P, C, K, E, D**) thoả mãn các điều kiện: **D**k ( **E**k(x) ) = x (x là bản gốc, giải mã(mã khoá) = gốc |

###### Chú ý :

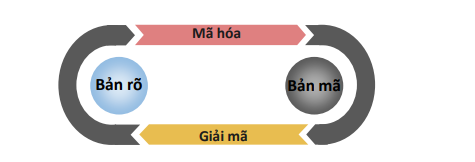
Với mỗi k sẽ được mô tả dưới dạng k = (ke, kd)

(mã dùng cho mã hoá, mã dùng cho giải mã)

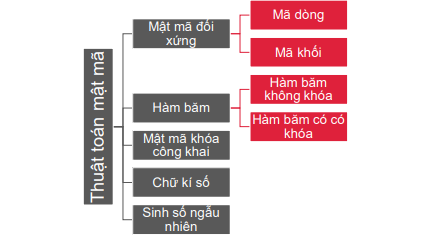
Ghi chú : - mã hoá : y = Ek e (x)

- giải mã : x = Dk d (y)

##### Quá trình chung của hệ mật



#### Thuật toán mã hoá



##### Cổ điển (ngày nay vẫn hay dùng trong trò chơi tìm mật thư)

###### Substitution (thay thế) :

Thay thế ký tự (nhóm ký tự) trong văn bản ban đầu bằng ký tự (nhóm ký tự) khác.

Ý tưởng trong các thuật toán hiện đại.

###### Transposition (hoán vị):

Các ký tự trong văn bản đầu chỉ thay đổi vị trí cho nhau.

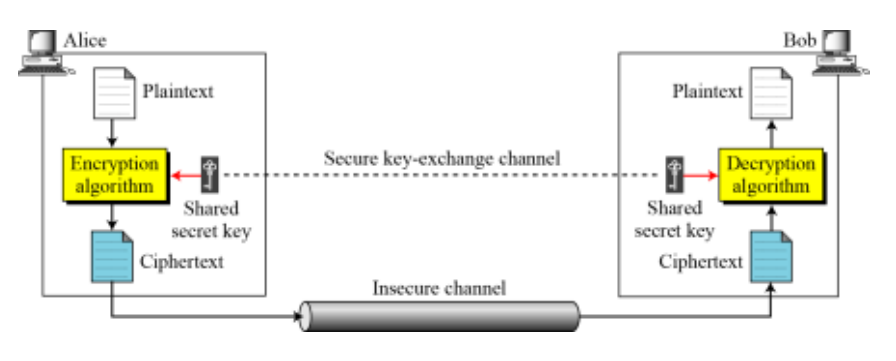
Bản thân ký tự không hề thay đổi.

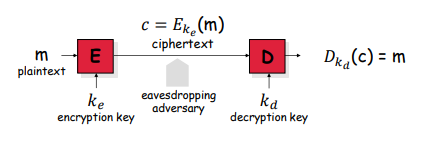
##### Hiện đại

###### **Symmetric cryptography** : mã hoá đối xứng

Từ ke dễ dàng tìm được kd (thông thường ke = kd)

Để đảm bảo an toàn, chìa khoá phải được giữ bí mật, nên có tên gọi khác là …





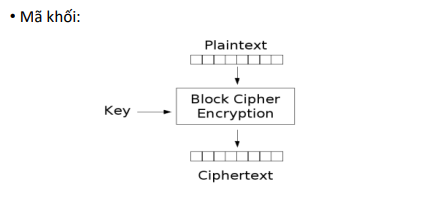
Mã hoá đối xứng có thể chia thành 2 nhóm phụ:

Block ciphers : Thuật toán khối

Từng khối dữ liệu trong văn bản ban đầu được thay thế bằng một khối dữ liệu khác có cùng độ dài.

Độ dài mỗi khối (block size) được tính bằng đơn vị bit, ví dụ thuật toán 3-Way có kích thước khối là 96 bit.

Thuật toán khối thông dụng : DES, 3DES, RC5, RC6, 3-Way, CAST, Camelia, Blowfish, MARS, Serpent, Towfish, GOST, …



Stream ciphers : Thuật toán dòng :

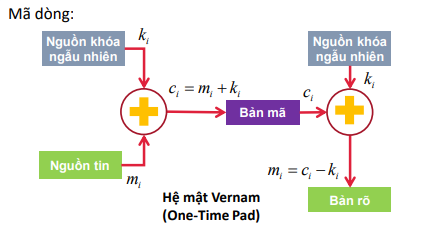
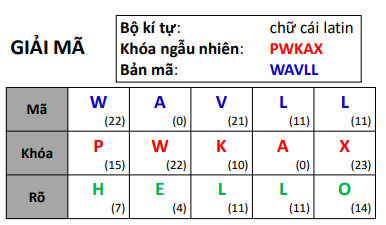
Dữ liệu đầu vào được mã hoá từng bit một.

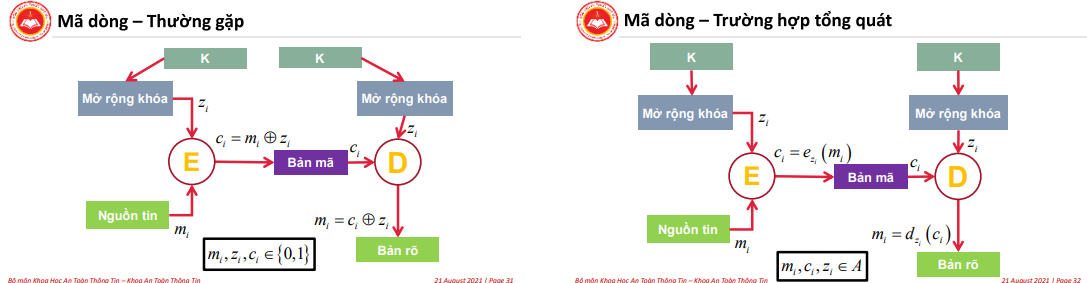
Có tốc độ nhanh hơn thuật toán khối, dùng khi khối lượng dữ liệu cần mã hoá chưa biết trước, ví dụ trong kết nối không dây.

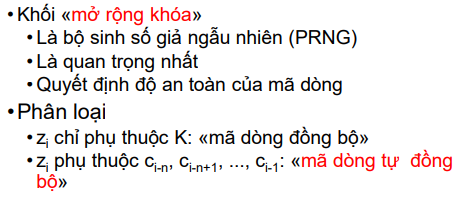
Có thể coi thuật toán dòng là thuật toán khối với kích thước mỗi khối là 1 bit.

Thuật toán dòng thông dụng : RC4, A5/1, A5/2, Chameleon, …

Ví dụ Hệ mật Vermam





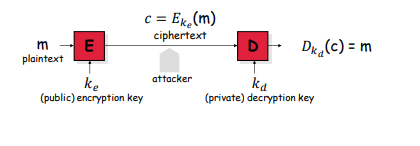
###### Asymmetric cryptography : mã hoá bất đối xứng

Mỗi bên có 2 khoá : khoá công khai (public key) và khoá bí mật (private key)

Bên gửi dùng khoá công khai của bên nhận để mã hoá

Bên nhận dùng khoá bí mật của mình để giải mã.

Hệ thống cho phép hai bên thực hiện giao dịch khi không chia sẻ trước một thông tin bí mật nào cả.



Hạn chế của thuật toán bất đối xứng là tốc độ chậm, do đó trong thực tế, người ta sử dụng thuật toán bất đối xứng để chia sẻ chìa khoá bí mật rồi sau đó dùng thuật toán đối xứng với chìa khoá bí mật trên để truyền thông tin.

#### Một số thuật toán nổi tiếng

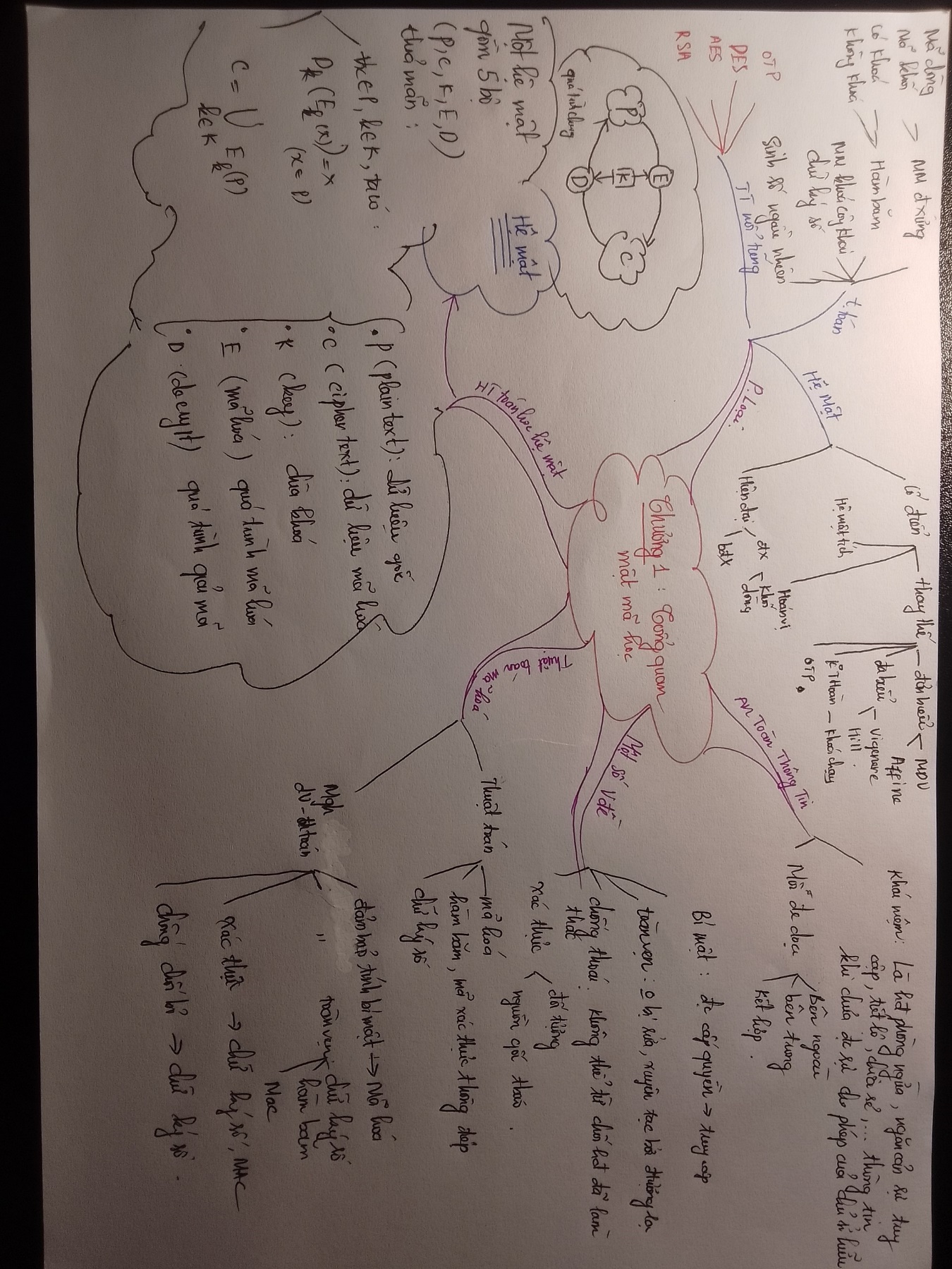
##### One-time Pad (OTP)

##### DES (Data Encryotion Standard)

##### AES

##### RSA

#### Hàm Hash

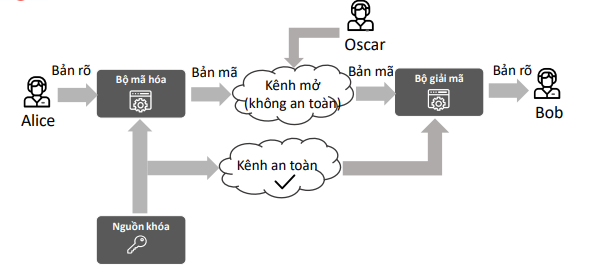


\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# HỆ MẬT KHOÁ BÍ MẬT

## CÁC HỆ MẬT CỔ ĐIỂN

#### Sơ đồ khối của một hệ truyền tin mật.



#### Phân Loại

#### Hệ mật thay thế đơn biểu.

##### Lý thuyết

###### Khi khoá được chọn, mỗi ký tự của bản rõ được ánh xạ đến một ký tự duy nhất của bản mã.

###### Độ dài khoá 26 -> số khoá có thể là 26!.

###### Các đặc trưng về ngôn ngữ, tần suất suất hiện của các chữ trong bản rõ và chữ tương đương trong bản mã là như nhau => Phương pháp thám mã bằng phương pháp thống kê tần suất.

###### Gồm:

Mã dịch vòng (MDV – shift cipher)

Mã Affine

##### Mã dịch vòng (Shift Cipher)

#### 

###### Ví dụ

P: ANH

k = 3

|  |  |  |  |
| --- | --- | --- | --- |
| **P** | A | N | H |
| **Mã bản rõ** | 0 | 13 | 7 |
| **Khoá** | 3 | 3 | 3 |
| **Mã bản mã** | 3 | 16 | 10 |
| **C** | D | Q | K |

y = E­k(x) = (x + k) mod 26

KL: ANH ---(mdv, k=3)---> DQK

##### Mã Affine

###### 

###### Chú ý :

Nghịch đảo modulo + định lý ơ-clit mở rộng

a\*i 1 mod m (i là nghịch đảo modulo của a theo mod m)

Ví dụ:

P = ANH

k = (7, 3)

Giải mã:

y = Ek(x) = 7x + 3 mod 26

|  |  |  |  |
| --- | --- | --- | --- |
| **P** | A | N | H |
| **Mã bản rõ** | 0 | 13 | 7 |
| **Mã bản mã** | 3 | 16 | 0 |
| **C** | D | Q | A |

Mã hoá:

x = Dk(x) = a-1(y – b) mod 26 = 15(y – 3) mod 26

|  |  |  |  |
| --- | --- | --- | --- |
| **C** | D | Q | A |
| **Mã bản rõ** | 3 | 16 | 0 |
| **Mã bản mã** | 0 | 13 | 7 |
| **P** | A | N | H |

Thám mã:

Gcd(a, 26) = 1 => Oa = 12 => K = 12x26 = 312 => vét cạn

Mở rộng bảng ký tự, chữ số, chữ cái => K = 75x95 = 7125 => vét cạn

#### Hệ mật thay thế đa biểu.

##### Lý thuyết

###### Sử dụng nhiều bảng chữ cái để mã, khắc phục điểm yếu của hệ mật thay thế đơn biểu.

###### Một chữ có thể mã bằng bất kỳ chữ nào => mất bớt cấu trúc của bản rõ => khó thám mã hơn.

###### Gồm:

Hệ mật Vigenere

Hệ mật Hill

##### Hệ mật Vigenere

#### 

###### Sơ đồ:

#### 

###### Mô tả:

#### 

###### Ví dụ:

m = 3, k = ANH

P = LEPHUONG

|  |  |  |
| --- | --- | --- |
| A | N | H |
| 0 | 13 | 7 |

Mã hoá:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P | L | E | P | H | U | O | N | G |
| Pr | 11 | 4 | 15 | 7 | 20 | 14 | 13 | 6 |
| K | 0 | 13 | 7 | 0 | 13 | 7 | 0 | 13 |
| Cr | 11 | 17 | 22 | 7 | 7 | 21 | 13 | 19 |
| C | L | R | U | H | H | V | N | T |

###### => C = LRUHHVNT

Giải mã:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| C | L | R | U | H | H | V | N | T |
| Cr | 11 | 17 | 22 | 7 | 7 | 21 | 13 | 19 |
| K | 0 | 13 | 7 | 0 | 13 | 7 | 0 | 13 |
| Pr | 11 | 4 | 15 | 7 | 20 | 14 | 13 | 6 |
| P | L | E | P | H | U | O | N | G |

=> P = LEPHUONG

###### Nhận xét:

Số khoá = 26m => khó thám mã

Chữ mã khác nhau cho cùng 1 chữ bản rõ => tần suất được làm phẳng => loại bỏ nhược điểm HM đơn biểu.

Độ dài khoá có hạn, tạo vòng lặp chưa hoàn toàn => số khoá thực tế nhỏ hơn, nhưng có thể coi là bản đơn giản của hệ mật OTP.

Sử dụng bảng Saint Cyr để tra.

##### Hệ mật Hill

#### 

###### Chú ý:

Phép toán thực hiện trên Z26

TH m = 2,

Định thức: det(K) = ad – bc

Đk K khả nghịch ( K-1): det(K) 0

Đk bản mã(C) duy nhất: gcd(det(K), 26) = 1

Ma trận nghịch đảo : K-1 = (det(K))-1

m = 3, ….

Quy trình mã hoá

Chia sâu bản rõ thành các sâu con độ lớn (1x2)

Ta có Pi = (1x2) => Ci = (1x2)(2x2) = (1x2) = (X, Y)

…..

Kết luận, đưa ra bản mã

Quy trình giải mã

Chia sâu mã

Tìm định thức

Kiểm tra điều kiện

Tìm K-1

Tìm Pi

Kết luận, đưa ra bản rõ tìm được

#### Hệ mật thay thế không tuần hoàn

##### Lý thuyết

###### Phép thế lý tưởng là sử dụng nhiều bảng chữ cái để không nhận diện được sự phân bố tần suất.

###### Vigenere đề xuất hệ mật khoá tự sinh (Hệ mật khoá chạy)

##### Hệ mật khoá chạy

###### Tư tưởng:

Key = khoá đã cho + nối tiếp bản rõ

Sử dụng mã Vigenere để mã tiếp

Gây mất khái niệm chu kỳ

###### Mô tả:

Mã hoá

Key = khoá đã cho + nối tiếp bản rõ

Sử dụng mã vigenere để mã hoá

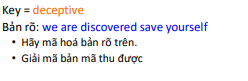
Giải mã

B1: dùng khoá, tìm đoạn bản rõ đầu tiên

B2: Dùng đoạn bản rõ vừa thu được làm khoá của đoạn tiếp theo

B3: Tiếp tục thao tác cho đến hết khoá

###### Ví dụ:



Mã hoá:



Giải mã:



###### Nhận xét:

K = 26m

Thực tế, độ an toàn chưa đạt mong đợi, độ khó phụ thuộc vào khoá ban đầu, nếu khoá không đủ lớn có thể vét cạn để thám mã.

#### Hệ mật OTP

##### Lý thuyết:

Do Gillbert Vernam đưa ra 1971

Hệ mật hoàn thiện, có độ an toàn tuyệt đối

##### Tư tưởng:

Khoá có độ dài bằng bản rõ

Sử dụng phép xor cộng mudulo cho 2 để mã hoá

##### Mô tả:

|  |
| --- |
| P = C = K = (Z2)n, n >= 1, n Z.x = (x1, x2, …, xn), k = (k1, k2, …, kn) Mã hoá:  y = ek(x) = (x1 k1, x2 k2, …, xn kn)  Giải mã:  x = dk(y) = (y1 k1, y2 k2, …, yn kn) |

##### Chú ý:

###### Phép xor:

Ký kiệu :

Cộng modulo cho 2

|  |  |  |
| --- | --- | --- |
| x | k | x k |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Bảng chân lý:

Tính chất:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P |  | K | = | C |
| Ko ngẫu nhiên |  | Ngẫu nhiên |  | Ngẫu nhiên |

Chứng minh:

y = ek(x) = x k

x = dk(y) = y k

=> x = (x k) k = x (k k) = x 0 = x

##### Ưu / nhược

###### Ưu:

An toàn tuyệt đối

###### Nhược:

Khó khăn khi truyền tin => sử dụng trong quân đội

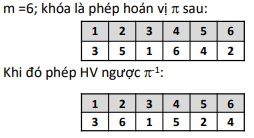
#### Hệ mật hoán vị

###### Tư tưởng: Thay đổi vị trí các chữ trong bản rõ

###### Mô tả:

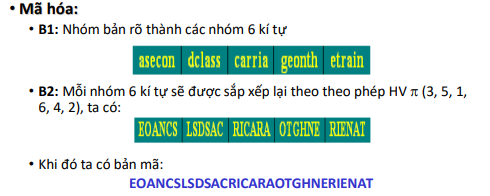
|  |
| --- |
| \* Cho m là số nguyên dương xác định . P = C = , k là tất cả các hoán vị có thể có của :  \* Với khóa π, ta xác định k:  ( là hoán vị ngược của π) |

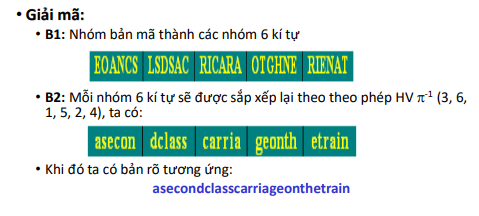
###### Chú ý:

(lật ngược, sắp xếp lại)

Khi độ dài bản rõ không khớp với khoá (không là bội chung) thì ta thêm các ký tự vô nghĩa (zzz…) vào cuối bản rõ để mã hoá tiếp.

###### Ví dụ:





#### Hệ mật tích

## 

#### Thám mã

#### 

#### 

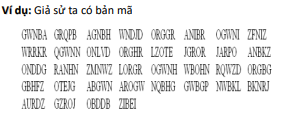
###### Các bước:

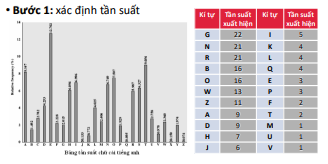
Bước 1: xây dựng bảng tần suất

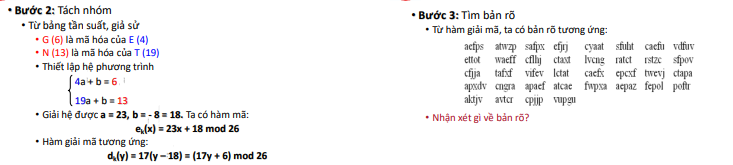
Bước 2: tách nhóm

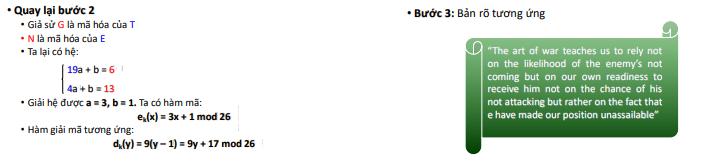
Bước 3: Tìm bảng mã

###### Ví dụ: thám mã Affine:











## LÀM BÀI TẬP

## CÁC HỆ MẬT HIỆN ĐẠI

### Phân loại

###### Mã đối xứng (hệ mật khoá bí mật):

Mã khối

DES

AES

Mã dòng

RC4

###### Mã bất đối xứng (kệ mật khoá công khai):

…

### Mã khối

##### Nguyên lý thiết kế mã khối

###### Theo Shannon, có 2 nguyên tắc cơ sở để có độ bảo mật cao:

Tính xáo trộn : (confussion, phép thế) : sự phụ thuộc vào bản mã đối với bản rõ phải thực sự phức tạp để gây rắc rối cho kẻ thám mã không tìm ra quy luật được . Quan hệ của mã – tin là phi tuyến.

Tính khuếch tán (diffusion, phép hoán vị ) : làm khuếch tán những mẩu văn bản mang đặc tính thống kê (gây ra do dư thừa ngôn ngữ) lẫn vào toàn bộ văn bản. Sự thay đổi 1 bit trong 1 khối bản rõ phải dẫn tới sự thay đổi hoàn toàn trong khối mã tạo ra.

###### Độ an toàn của thuật toán mã hoá

Không phụ thuộc vào thuật toán mã hoá

Phụ thuộc vào khoá

=> Hệ mã hoá được cho là an toàn khi công khai tt mã hoá, cũng không thể giải mã được bản mã (phụ thuộc vào khoá)

### Hệ mật DES (Data Encryption Standard) : Tiêu chuẩn mã hoá dữ liệu

###### https://lilthawg29.files.wordpress.com/2021/08/image-32.png?w=1024

#### Giới thiệu

##### Tồng quan

DES (Data Encryption Standard) là chuẩn mã hóa dữ liệu đầu tiên trên thế giới, do Cơ quan an ninh Quốc gia Hoa Kỳ (NSA) đề xuất trên cơ sở cải tiến thuật toán Lucifer do hãng IBM công bố năm 1964. DES đã được sử dụng rộng rãi ở Hoa Kỳ và nhiều quốc gia khác trong các thập kỷ 70, 80, 90 cho đến khi được thay thế bởi Tiêu chuẩn mã hóa dữ liệu tiên tiến AES (Advanced Encryption Standard) vào năm 2002.

##### Lịch sử hình thành và phát triển:

###### 1972:

###### 1974:

###### ­1976:

##### Sử dụng cấu trúc mạng Feistel Network để mã hóa và giải mã dữ liệu (phân chia dữ liệu thành hai nửa và thực hiện một loạt các hoạt động, bao gồm hoán vị (permutation), thay thế (substitution), và phép XOR, trước khi kết hợp nửa dữ liệu lại với nhau.)

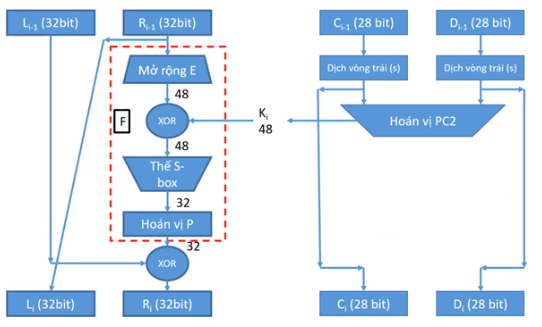
#### Thuật toán DES

##### Sơ đồ hoàn chỉnh:

###### 

##### Chi tiết thuật toán: Gồm 16 vòng

###### Chi tiết 1 vòng:



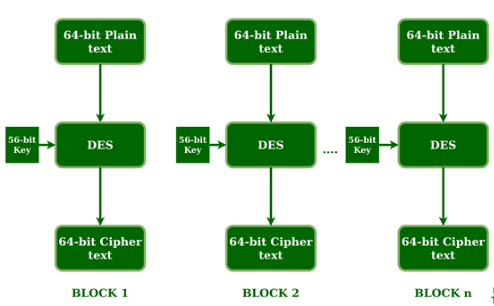
###### Thuật toán mã:

Chuẩn bị đầu vào (input)

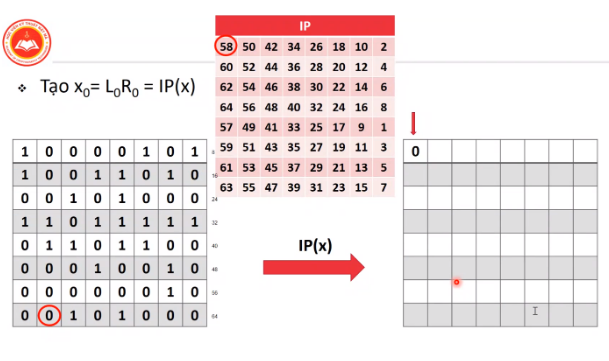
Chọn thông điệp

Chuyển sang nhị phân

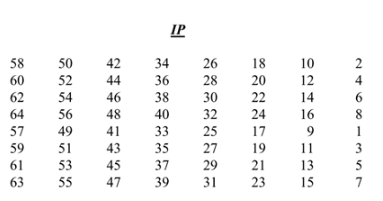
Chia thành các Block 64 bit (padding(đệm) nếu cần )



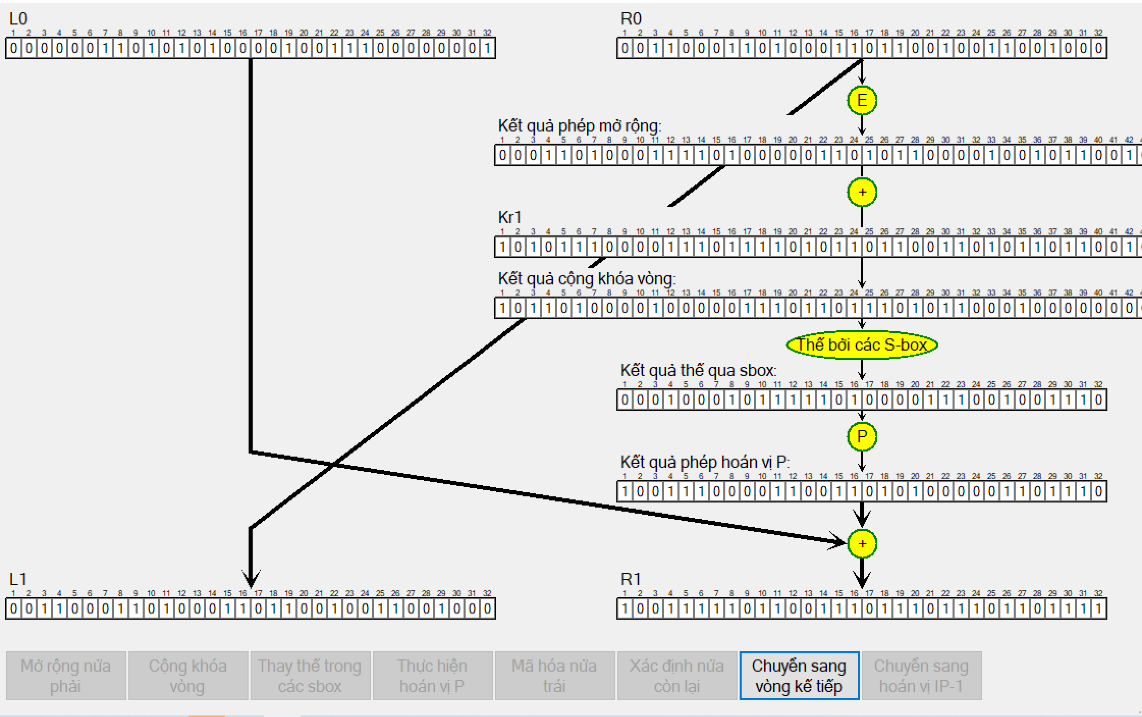
Hoán vị IP



Ghi lại x ----(viết lại từ dưới lên thứ tự cột 2, 4, 6, 8, 1, 3, 5, 7)---> IP(x)

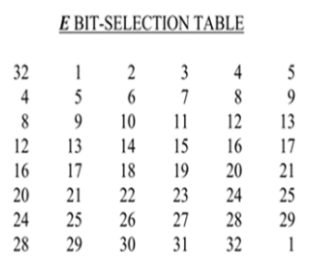


Hàm F (Feistel):

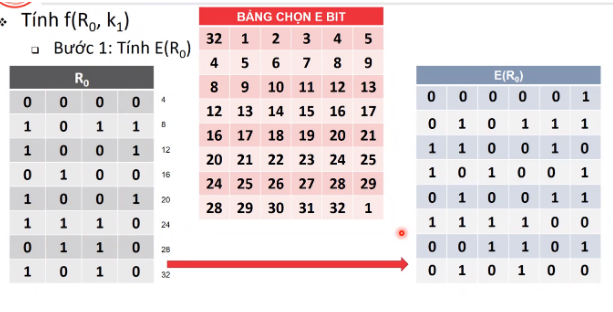


**E (Mở rộng R)**

Đưa phần R0 qua hoán vị mở rộng E ( mục đích của hoán vị mở rộng này là mở rộng R0 lên 48 bits để tý XOR với K1 cũng 48 bits).

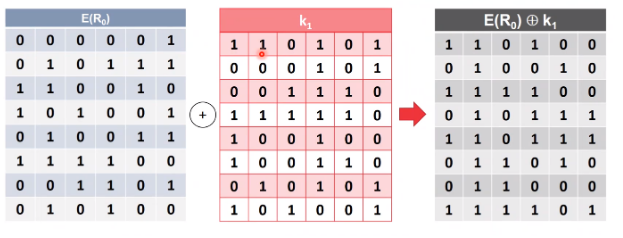


Hoán vị mở rộng E nhìn chung là việc lặp lại 2 bit cuối của hàng trước và viết tiếp bit của hàng sau.

Vd: 

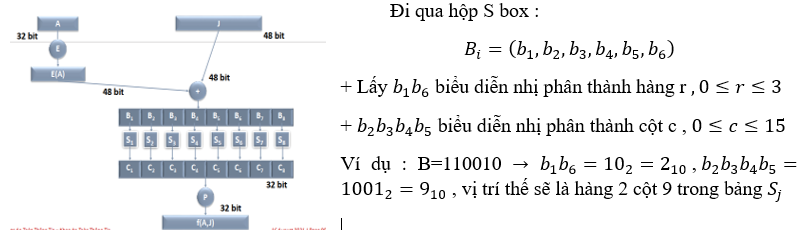
**XOR**

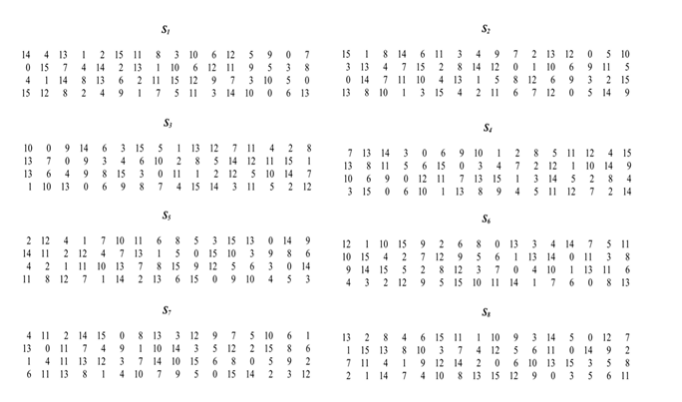
Ri(48) XOR Ki(48)

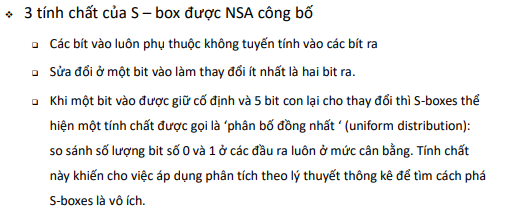
Vd: 

**S (S-boxes)**

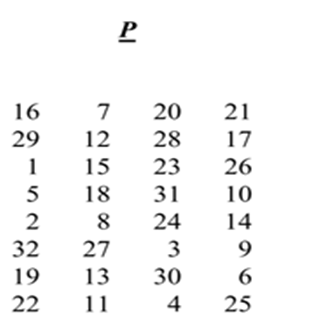
Chia 48 bits thành 8 khối, mỗi khối 6 bits (mỗi hàng 1 khối)







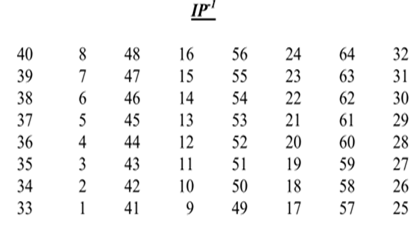
**P (Hoán vị P)**



Sau đó lấy L0 XOR với kết quả vừa tìm được ta được R1,L1 được gán = R0 ban đầu.

Hoán vị IP-1

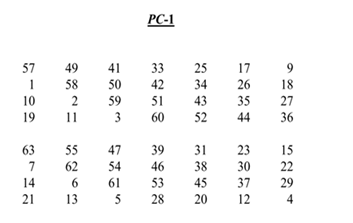
Khi hết 16 vòng thì cho qua hoán vị IP-1



###### Thuật toán sinh khoá key

Hoán vị PC-1

Khoá 64 bits từ ban đầu qua hoán vị PC-1(Permuted Choice – 1) loại bỏ 8 bits chia hết cho 8 : 8,16,24,32,40,48,56,64.



C0= 28 bits đầu, D0 = 28 bits cuối.

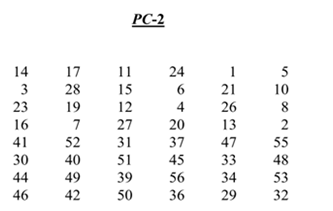
Thuật toán dịnh trái bit (…)

dịch vòng trái C0và D0

Dịch vòng trái ở các vòng khác nhau, chỉ có vòng 1,2,9,16 là dịch 1 trái 1 bit, còn lại là 2 bit.

Hoán vị PC-2

PC-2 hoạt động tương tự PC-1 chỉ khác ở sự hoán vị, các bit 9,18,22,25,35,38,43,54 bị lược bỏ.



Đầu ra lúc này là K1 = 48 bits.

Lưu lại kết quả của C0và D0 sau khi dịch vòng trái và gán vào C1 và D1 để chuẩn bị cho vòng 2.

#### Tính chất DES

###### 

###### 

###### 

#### Các biến thể của DES

##### DES bội hai

###### 

##### DES bội ba

###### 

### Thuật toán AES

###### https://lilthawg29.files.wordpress.com/2021/09/image-34.png?w=520