Thực hành Bài 4: Mã khối hiện đại DES – AES (Tuỳ chọn ko bắt buộc đối với các nhóm)

I. Chuẩn mã hóa dữ liệu (DES)

Mã hóa:

1. Viết hàm y = IP(x) thực hiện hoán vị IP
   Input: x - chuỗi số 64 bit
   Output: y – chuỗi số 64 bit là hoán vị của x theo ma trận IP
2. Viết hàm SPLIT(x, L, R) tách chuỗi số 64 bit (x) thành 2 nửa 32 bit trái (L) và phải (R);
3. Viết hàm R1 = E(R) mở rộng chuỗi số 32 bit (R) thành chuỗi số 48 bit (R1) theo ma trận mở rộng E.
   Input: R - chuỗi số 32 bit
   Output: R1 – chuỗi số 48 bit
4. Viết hàm XR1K = XOR(R1, Ks) thực hiện phép XOR bit hai chuỗi số 48 bit R1 và Ks.
   Input: R1, Ks - chuỗi số 48 bit
   Output: XR1K – chuỗi số 48 bit
5. Viết hàm SXR1K = SUB(XR1K) thực hiện phép thế byte bằng bảng S-box cho chuỗi số 48 bit XR1K.
   Input: XR1K - chuỗi số 48 bit
   Output: SXR1K – chuỗi số 32 bit
6. Viết hàm F = P(SXR1K) thực hiện hoán vị P
   Input: SXR1K - chuỗi số 32 bit
   Output: F – chuỗi số 32 bit là hoán vị của x theo ma trận P

Sinh khóa

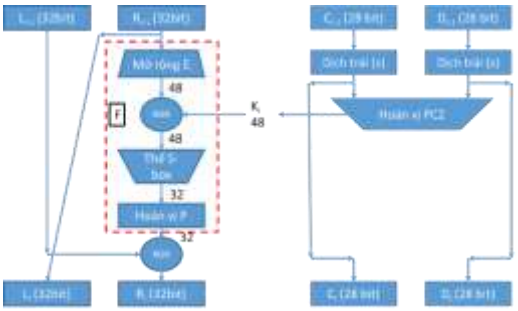
1. Viết hàm K1 = PC1(K) thực hiện hoán vị PC1
   Input: K - chuỗi số 64 bit
   Output: K1 – chuỗi số 56 bit là hoán vị của K theo ma trận PC1
2. Viết hàm SPLIT_KEY(K1, C, D) tách chuỗi số 56 bit (K1) thành 2 nửa 28 bit trái (C) và phải (D);
   Input: K1 – chuỗi số 56 bit
   Output: C, D – chuỗi số 28 bit
3. Viết hàm ShiftLeft(x, s) dịch vòng trái s bit đối với chuỗi số 28 bit (x)
   Input: x – chuỗi số 28 bit, s – số nguyên dương <28
   Output: x – chuỗi số 28 bit đã dịch vòng trái s bit
4. Viết hàm Ks = PC2(C, D, s) thực hiện hoán vị PC2
   Input: C, D - chuỗi số 28 bit, s – số nguyên dương < 28
   Output: Ks – chuỗi số 48 bit là hoán vị của C, D theo ma trận PC2.

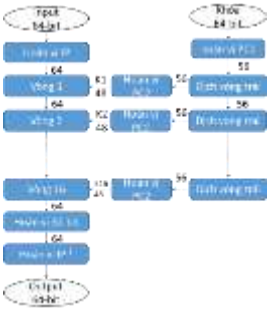
Mã hóa DES – xây dựng hàm y = DES (x, k) thực hiện mã hóa theo thuật toán DES

input: x, k – chuỗi số 64 bit
Output: y – chuỗi số 64 bít được mã hóa từ x theo thuật toán DES với khóa k

Cấu trúc DES

Chi tiết một vòng lặp DES





II. Chuẩn mã hóa nâng cao (AES)

Mã hóa

1. Viết hàm y = SUBBYTE(state) thực hiện việc thế byte.
   Input: state – ma trận 4x4 = 16 byte
   Output: y – ma trận 4x4 = 16 byte byte là kết quả thay thế byte x theo bảng S-box
2. Viết hàm y = SHIFTROW(state) thực hiện việc dịch hàng.
   Input: state – ma trận 4x4 = 16 byte
   Output: y – ma trận 4x4 = 16 byte byte là kết quả dịch hàng.
3. Viết hàm y = MIXCOLUMN(state) thực hiện việc nhân ma trận.
   Input: state – ma trận 4x4 = 16 byte
   Output: y – ma trận 4x4 = 16 byte byte là kết quả mixcolumn của state
4. Viết hàm y = ADDROUNDKEY(state, K) thực hiện việc nhân ma trận.
   Input: state, K – ma trận 4x4 = 16 byte
   Output: y – ma trận 4x4 = 16 byte byte là kết quả AddRoundKey của state và khóa K.

Sinh khóa

Viết hàm KeyExpansion mở rộng khóa từ khóa K (16 byte) thành 176 byte (11 x 16 byte)

Input: K mảng 16 byte
Output: K+ mảng 176 byte.

Mã hóa AES – xây dựng hàm mã hóa AES(x, k) thực hiện mã hóa theo thuật toán AES – 128 bit khóa

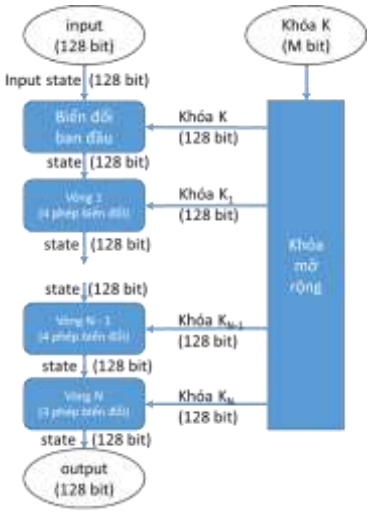
input: x, k – chuỗi số 128 bit
Output: y – chuỗi số 128 bít được mã hóa từ x theo thuật toán AES với khóa k

Cấu trúc thuật toán AES

Chi tiết một vòng lặp (từ vòng 1 đến N – 1)

1. **Substitute bytes**
2. **ShiftRows**
3. **MixColumns**
4. **AddRoundKey**

Riêng vòng thứ N không có phép MixColumns.



Khóa (bit)	**128**
Input (bit)	**128**
Số vòng lặp	**10**
Khóa vòng lặp (bit)	**128**
Khóa mở rộng (bytes)	**176**

Một ví dụ AES-128

Input **0123456789abcdeffedcba9876543210**

Key	**0f1571c947d9e8590cb7add6af7f6798**
output	**ff0b844a0853bf7c6934ab4364148fb9**