MỤC LỤC

A. Một số kiến thức cần học

1.1 VLC (đề thi không thấy có, học cái này sau)

Bảng mã Huffman của DC

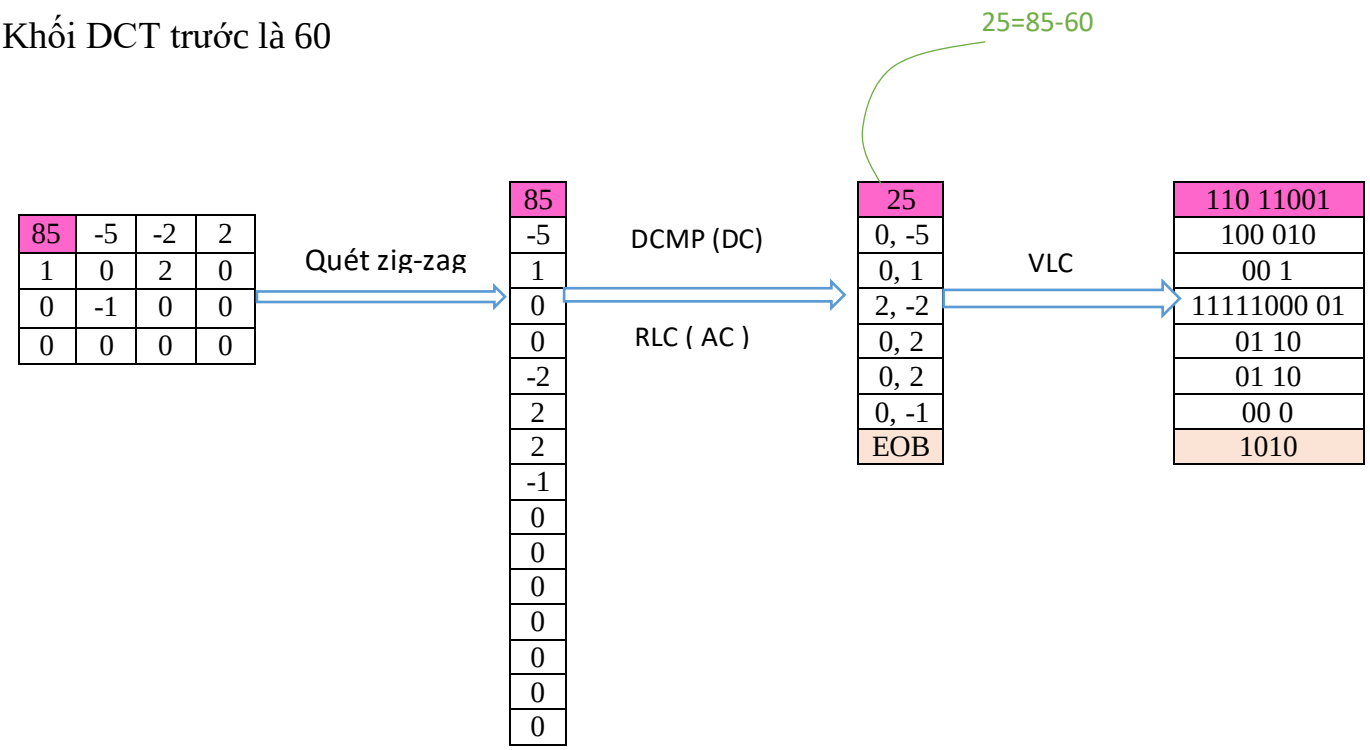
Phạm vi	Loại DC	Từ mã	Độ dài
0	0	010	3
−1, 1	1	011	4
−3, −2, 2, 3	2	100	5
−7, . . . , −4, 4, . . . , 7	3	00	5
−15, . . . , −8, 8, . . . , 15	4	101	7
−31, . . . , −16, 16, . . . , 31	5	110	8
−63, . . . , −32, 32, . . . , 63	6	1110	10
−127, . . . , −64, 64, . . . , 127	7	11110	12
−255, . . . , −128, 128, . . . , 255	8	111110	14
−511, . . . , −256, 256, . . . , 511	9	1111110	16
−1023, . . . , −512, 512, . . . , 1023	A	11111110	18
−2047, . . . , −1024, 1024, . . . , 2047	B	111111110	20

Bảng mã huffman của AC

Run/ Category	Base Code	Length	Run/ Category	Base Code	Length
0/0	1010 (= EOB)	4			
0/1	00	3	8/1	11111010	9
0/2	01	4	8/2	1111111111000000	17
0/3	100	6	8/3	1111111110110111	19
0/4	1011	8	8/4	1111111110111000	20
0/5	11010	10	8/5	1111111110111001	21
0/6	111000	12	8/6	1111111110111010	22
0/7	1111000	14	8/7	1111111110111011	23
0/8	1111110110	18	8/8	1111111110111100	24
0/9	1111111110000010	25	8/9	1111111110111101	25
0/A	1111111110000011	26	8/A	1111111110111110	26
1/1	1100	5	9/1	111111000	10
1/2	111001	8	9/2	1111111110111111	18
1/3	1111001	10	9/3	1111111111000000	19
1/4	111110110	13	9/4	1111111111000001	20
1/5	11111110110	16	9/5	1111111111000010	21
1/6	1111111110000100	22	9/6	1111111111000011	22
1/7	1111111110000101	23	9/7	1111111111000100	23
1/8	1111111110000110	24	9/8	1111111111000101	24
1/9	1111111110000111	25	9/9	1111111111000110	25
1/A	1111111110001000	26	9/A	1111111111000111	26
2/1	11011	6	A/1	111111001	10
2/2	11111000	10	A/2	1111111111001000	18
2/3	1111110111	13	A/3	1111111111001001	19
2/4	1111111110001001	20	A/4	1111111111001010	20
2/5	1111111110001010	21	A/5	1111111111001011	21
2/6	1111111110001011	22	A/6	1111111111001100	22
2/7	1111111110001100	23	A/7	1111111111001101	23

VD:

Khối DCT trước là 60



Một số giải thích khi chuyển VLC (*theo cách hiểu vẹt của mình*)

		Đổi sang hệ Bin	Loại	Độ dài	Kết quả(Từ mã + Độ dài cái đổi ra hệ Bin)

	25	25 = 11001	5	8-3=5	110 11001
0,1	1	1=01	0/1	3-2=1	00 1
0,2	2	2=10	0/2	4-2=2	01 10
0, -5	-5	5=101 -> -5=010	0/3	6-3=3	100 010
2, -2	-2	2=10 -> -2=01	A/2	10-8=2	11111000 01
0, -1	-1	1=01 -> -1=10	0/1	3-2=1	00 0

EOB mặc định đổi ra là 1010

1.2 WU-LEE

□ Thuật toán: (..)

- Bước 1: Chia ảnh F thành các khối nhỏ, mỗi khối có kích thước là $m \times n$
- *Bước 2*: Với mỗi khối ảnh nhỏ F_i thu được từ bước 1, ta kiểm tra điều kiện sau:

$$0 < \boldsymbol{SUM(F_i \wedge K)} < \boldsymbol{SUM(K)}$$

  - Nếu đúng thì chuyển đến bước 3 để giấu thông tin vào trong khối F_i
  - Không đúng thì không giấu dữ liệu vào trong khối F_i (khối F_i được giữ nguyên)

- *Bước 3*: Gọi bít cần giấu vào trong khối F_i là b, thực hiện các bước sau để thay đổi F_i

```
if (SUM(Fi∧ K) mod 2 = b ) then giữ nguyên Fi
else if (SUM(Fi∧ K) =1 ) then
        Chọn ngẫu nhiên một bít (j,k) thỏa mãn đồng thời
        [Fi]jk = 0 và [K]jk = 1 sau đó chuyển giá trị của bít [Fi]jk thành 1
     else if (SUM(Fi∧K) =SUM(K) -1) then
             Chọn ngẫu nhiên một bít (j,k) thỏa mãn đồng thời
             [Fi]jk = 1 và [K]jk = 1 sau đó chuyển giá trị của bít [Fi]jk thành 0
          else Chọn ngẫu nhiên một bít mà [K]jk = 1 chuyển giá trị của bít [Fi]jk
               từ 0 trở thành 1, hoặc từ 1 trở thành 0
```

VD:

Thuật toán Wu-Lee (..)

□ VD:

- Cần nhúng thông tin B vào ảnh F sử dụng khóa K như sau

F_1 | F_2

1	1	0	1	1	1
1	1	1	1	1	0
0	1	0	0	0	0
0	0	1	0	0	0
1	1	0	1	1	1
0	1	1	0	1	0

F_3 | F_4

Thông tin giấu $B = 011$

1	1	0
1	1	1
0	1	0

K

GIẢI

Thuật toánWu-Lee (..)

□Bước 1:

- Chia ảnh F thành 4 khối nhỏ F_1, F_2, F_3, F_4 có kích thước là 3×3

□Bước 2:

- Với mỗi F_i, kiểm tra điều kiện$0 < SUM(F_i \wedge K) < SUM(K)$
  - Đúng thì giấu vào F_i (thực hiện bước 3)
  - Sai thì giữ nguyên F_i (không giấu)

□ Với F_1

- Vì $0 < SUM(F_1 \wedge K) = SUM(K) = 6$ nên không giấu được dữ liệu vào trong F_1

□ Với F_2

- Vì $0 < SUM(F_2 \wedge K) = 4 < SUM(K) = 6$, nên một bít sẽ được giấu vào khối F_2 (giấu bit 0)
- Thực hiện bước 3
  - Ta thấy $SUM(F_2 \wedge K)\ mod\ 2 = 4\ mod\ 2 = 0$ và cũng chính là bằng bít cần giấu $b = 0$ vì vậy khối F_2 được giữ nguyên

---

□ Với F_3

- Ta có $0 < SUM(F_3 \wedge K) = 3 < SUM(K) = 6$ nên có thể giấu bít thứ 2 là $b = 1$ vào khối này
- Thực hiện bước 3
  - Kiểm tra thấy $SUM(F_3 \wedge K)\ mod\ 2 = 3\ mod\ 2 = 1$ cũng chính bằng bít cần giấu nên ta vẫn giữ nguyên F_3

□ Với F_4

- Ta có $0 < SUM(F_4 \wedge K) = 4 < SUM(K)$ nên có thể giấu bít thứ 3 là $b = 1$ vào khối này
- Thực hiện bước 3
  - Kiểm tra thấy $SUM(F_4 \wedge K)\ mod\ 2 = 4\ mod\ 2 = 0 \neq b$
  - Kiểm tra $SUM(F_4 \wedge K) = 4 \neq 1$ và $\neq SUM(K) - 1 = 5$ vì vậy chọn ngẫu nhiên một bít $[K]_{jk} = 1$ rồi đảo bít $[F]_{jk}$, cụ thể ở đây ta chọn bít $[K]_{21} = 1$ và đảo bít $[F_4]_{21}$ từ 1 thành 0

□ Kết quả

F_1			F_2		
1	1	0	1	1	1
1	1	1	1	1	0
0	1	0	0	0	0
0	0	1	0	0	0
1	1	0	1	1	1
0	1	1	0	1	0
F_3			F_4		

Thông tin giấu $B = 011$ →

K		
1	1	0
1	1	1
0	1	0

F_1'			F_2'		
1	1	0	1	1	1
1	1	1	1	1	0
0	1	0	0	0	0
0	0	1	0	0	0
1	1	0	0	1	1
0	1	1	0	1	0
F_3'			F_4'		

1. 3 Jsteg

■ **Bài tập áp dụng:**

□ Cho ma trận các hệ số DCT 8 × 8 như sau:

1480	49	-61	0	0	0	1	0
10	0	1	-22	11	8	0	0
1	1	0	0	0	0	0	0
-19	0	1	27	1	1	1	1
1	1	0	0	0	0	0	0
-30	0	-19	1	0	1	0	1
0	0	0	0	1	-4	1	0
1	1	0	0	1	0	1	1

□ Áp dụng thuật toán Jsteg để ẩn đoạn mã 8 bit m = 00101101 và thực hiện trích xuất m sau khi nhúng.

GIẢI

■ Giải: Thực hiện nhúng tin m = 00101101

1480	49	-61	0	0	0	1	0
10	0	1	-22	11	8	0	0
1	1	0	0	0	0	0	0
-19	0	1	27	1	1	1	1
1	1	0	0	0	0	0	0
-30	0	-19	1	0	1	0	1
0	0	0	0	1	-4	1	0
1	1	0	0	1	0	1	1

→

1480	48	-60	0	0	0	1	0
11	0	1	-22	11	9	0	0
1	1	0	0	0	0	0	0
-18	0	1	27	1	1	1	1
1	1	0	0	0	0	0	0
-30	0	-19	1	0	1	0	1
0	0	0	0	1	-4	1	0
1	1	0	0	1	0	1	1

■ Các kĩ thuật trên ảnh số

■ Trích xuất:

1480	48	-60	0	0	0	1	0
11	0	1	-22	11	9	0	0
1	1	0	0	0	0	0	0
-18	0	1	27	1	1	1	1
1	1	0	0	0	0	0	0
-30	0	-19	1	0	1	0	1
0	0	0	0	1	-4	1	0
1	1	0	0	1	0	1	1

→ 00101101010

Mẹo nhỏ: Lẻ giảm / chẵn tăng

1.4 OUTGESS

VD1: Ẩn tin

■ Bài tập áp dụng:

□ Cho ma trận các hệ số DCT 8 × 8 như sau:

1480	49	-61	0	0	0	1	0
10	0	1	-22	11	8	0	0
1	1	0	0	0	0	0	0
-19	0	1	27	1	1	1	1
1	1	0	0	0	0	0	0
-30	0	-19	1	0	1	0	1
0	0	0	0	1	-4	1	0
1	1	0	0	1	0	1	1

□ Dựa trên thuật toán Outguess, yêu cầu:

- Tạo chuỗi giả ngẫu nhiên DCT bằng cách trích rút các hệ số DCT trên theo sắp xếp zig – zag, sau đó dịch phải k = 3 vị trí
- Thực hiện ẩn tin m = 01101011 trong khối ma trận trên

Giải

■ Giải:

□ Sắp xếp Zig – zag:

1480	49	-61	0	0	0	1	0
10	0	1	-22	11	8	0	0
1	1	0	0	0	0	0	0
-19	0	1	27	1	1	1	1
1	1	0	0	0	0	0	0
-30	0	-19	1	0	1	0	1
0	0	0	0	1	-4	1	0
1	1	0	0	1	0	1	1

Sắp xếp zig-zag

1480	49	10	1	0	-61	0	1
1	-19	1	0	0	-22	0	0
11	0	1	1	-30	0	0	0
27	0	8	1	0	0	0	1
0	-19	0	1	1	0	1	0
1	0	0	0	1	0	0	0
0	0	1	1	0	1	0	0
-4	1	0	1	1	0	1	1

dịch phải 3 bit

■ Thực hiện ẩn tin m = 0110 1011 trong khối ma trận trên

0	1	1	1480	49	10	1	0
-61	0	1	1	-19	1	0	0
-22	0	0	11	0	1	1	-30
0	0	0	27	0	8	1	0
0	0	1	0	-19	0	1	1
0	1	0	1	0	0	0	1
0	0	0	0	0	1	1	0
1	0	0	-4	1	0	1	1

ẩn tin m

0	1	1	1480	48	11	1	0
-61	0	1	1	-18	1	0	0
-23	0	0	10	0	1	1	-31
0	0	0	27	0	8	1	0
0	0	1	0	-19	0	1	1
0	1	0	1	0	0	0	1
0	0	0	0	0	1	1	0
1	0	0	-4	1	0	1	1

■ Sắp xếp các xáo trộn về vị trí ban đầu:

□ Bước 1: Dịch trái 3 bit

0	1	1	1480	48	11	1	0
-61	0	1	1	-18	1	0	0
-23	0	0	10	0	1	1	-31
0	0	0	27	0	8	1	0
0	0	1	0	-19	0	1	1
0	1	0	1	0	0	0	1
0	0	0	0	0	1	1	0
1	0	0	-4	1	0	1	1

dịch trái 3 bit

1480	48	11	1	0	-61	0	1
1	-18	1	0	0	-23	0	0
10	0	1	1	-31	0	0	0
27	0	8	1	0	0	0	1
0	-19	0	1	1	0	1	0
1	0	0	0	1	0	0	0
0	0	1	1	0	1	0	0
-4	1	0	1	1	0	1	1

■ Bước 2: Đọc theo từng hàng của MT rồi viết theo kiểu zig zag vào MT kết quả:

1480	48	11	1	0	-61	0	1
1	-18	1	0	0	-23	0	0
10	0	1	1	-31	0	0	0
27	0	8	1	0	0	0	1
0	-19	0	1	1	0	1	0
1	0	0	0	1	0	0	0
0	0	1	1	0	1	0	0
-4	1	0	1	1	0	1	1

Sắp xếp zig-zag

1480	48	-61	0	0	0	1	0
11	0	1	-23	10	8	0	0
1	1	0	0	0	0	0	0
-18	0	1	27	1	1	1	1
1	1	0	0	0	0	0	0
-31	0	-19	1	0	1	0	1
0	0	0	0	1	-4	1	0
1	1	0	0	1	0	1	1

VD2: Trích xuất

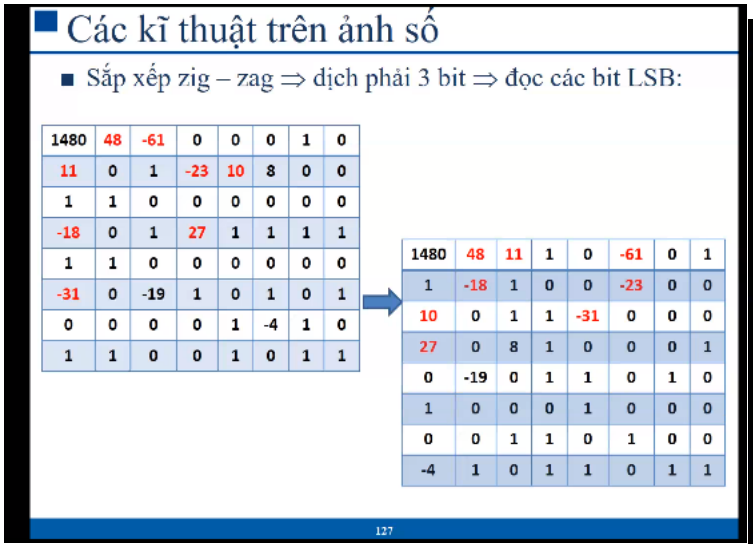
Các kĩ thuật trên ảnh số

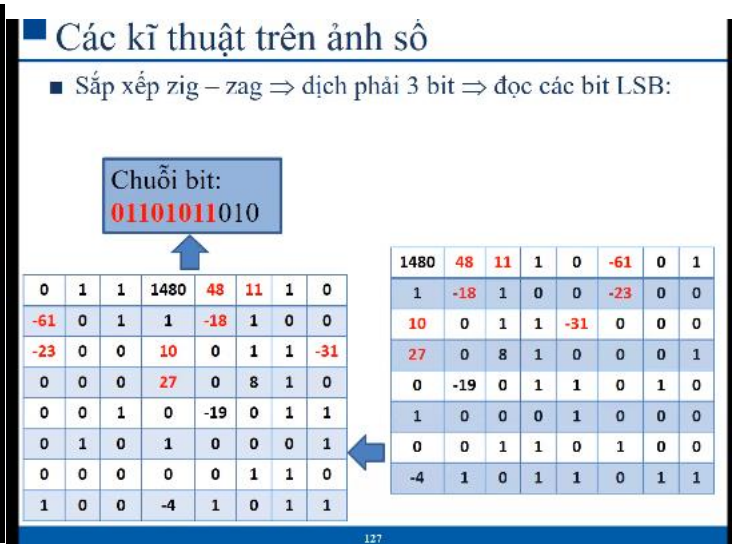
- MT khối ảnh sau nhúng:

1480	48	-61	0	0	0	1	0
11	0	1	-23	10	8	0	0
1	1	0	0	0	0	0	0
-18	0	1	27	1	1	1	1
1	1	0	0	0	0	0	0
-31	0	-19	1	0	1	0	1
0	0	0	0	1	-4	1	0
1	1	0	0	1	0	1	1

- BTVN: Trích xuất m từ khối ảnh kết quả

GIẢI





B. LÀM ĐỀ:

Cơ quan an ninh bắt được một điệp viên, qua quá trình khai thác điệp viên này đã khai ra cách thức liên lạc trao đổi thông tin với tổ chức của hắn ở nước ngoài qua hình thức thực hiện giấu tin trong ảnh. Dựa vào cách thức liên lạc, thực hiện quá trình trao đổi thông tin mô tả ở bên dưới, giả sử **K = (SBD + 43) mod 255**

Sinh viên giúp cơ quan an ninh tiến hành thực hiện:

a) Giải mã thông tin **M** có độ dài 8 bit từ tổ chức bên ngoài gửi cho điệp viên với $\mathbf{T_{extract}}$ và $\mathbf{X_{extract}}$

b) Giấu thông điệp **P** với độ dài 8 bít trong đó P = (SBD+19) mod 255 vào $\mathbf{T_{embed}}$ và vào $\mathbf{X_{embed}}$

Trong đó, quá trình trao đổi thông tin diễn ra như sau:

- **Bước 1**: Tổ chức bên ngoài và điệp viên thống nhất một khóa chung quy ước **K** (là một số tự nhiên trong khoảng từ 0-255). Khóa K này sẽ được chuyển thành 8 bít thông tin K = $k_7k_6k_5k_4k_3k_2k_1k_0$
- **Bước 2**: Khi điệp viên nhận được ảnh giấu tin, hắn sẽ thực hiện trích xuất tin như sau:
  - ***Bước 2a***: Lấy ma trận điểm ảnh $T_{extract}$ và tiến hành giải mã theo thuật toán **Wulee** với khóa K_{exract} để thu được Q_i từ các khối T_i tương ứng, ta có giá trị $Q=Q_7Q_6Q_5Q_4Q_3Q_2Q_1Q_0$. Trường hợp T_i không được giấu tin thì loại bỏ Q_i và thêm giá trị 0 vào bít có trọng số lớn nhất ví dụ Q_3 và Q_5 không giấu tin thì $Q= \mathbf{00}Q_7Q_6Q_4Q_2Q_1Q_0$

	Ma trận $T_{extract}$								
T_0	1	0	1	0	0	0	0	1	T_4
	1	0	0	1	1	0	1	1	
T_1	0	1	0	1	1	0	1	0	T_5
	1	0	0	1	0	1	1	0	
T_2	0	0	0	1	0	1	1	0	T_6
	1	1	0	1	0	0	0	1	
T_3	0	0	1	1	1	0	1	0	T_7
	0	1	0	1	1	1	0	1	

$K_{extract}$			
k_7	k_6	k_5	k_4
k_3	k_2	k_1	k_0

  - ***Bước 2b*:** Trường hợp Q lẻ thì thuật toán giấu tin trong ảnh là thuật toán **Outguess** và thực hiện trích rút các hệ số của ma trận DCT $\mathbf{X_{extract}}$ = X , biết rằng ma trận DCT ảnh gốc đã bị xáo trộn bằng cách sắp xếp zig-zag sau đó dịch vòng từ trên xuống 3 hàng trước khi nhúng.
  - Trường hợp Q chẵn thì thuật toán giấu tin trong ảnh sử dụng là thuật toán **Jsteg** và thực hiện trích rútcác hệ số của ma trận DCT $\mathbf{X_{extract}}$ = **X** lần lượt theo chiều dọc từ trên xuống dưới và từ trái sang phải.

X							
435	-37	-64	0	45	0	1	0
3	0	1	13	-21	1	0	0
0	1	19	0	0	0	0	0
12	0	23	13	1	-47	1	1
1	15	0	0	0	17	0	0
23	0	-19	1	0	1	0	1
0	-11	0	0	1	29	1	0
1	1	0	0	1	0	1	1

- **Bước 3**: Khi giấu tin gửi đi P, điệp viên thực hiện như sau
  - ***Bước 3a:*** Lấy ma trận điểm ảnh T_{embed} và tiến hành mã hóa theo thuật toán Wulee với khóa K_{embed}

T_{embed}							
T_1		T_2		T_3		T_4	
1	0	1	0	0	0	0	1
1	0	0	1	1	0	1	1
0	1	0	1	1	0	1	0
1	0	0	1	0	1	1	0
0	0	0	1	0	1	1	0
1	1	0	1	0	0	0	1
0	0	1	1	1	0	1	0
0	1	0	1	1	1	0	1
T_5		T_6		T_7		T_8	

K_{embed}	
k_7	k_3
k_6	k_2
k_5	k_1
k_4	k_0

  - ***Bước 3b***: Trường hợp P lẻ sử dụng thuật toán giấu **Jsteg**, sau đó thực hiện nhúng tin $P=P_7P_6P_5P_4P_3P_2P_1P_0$ vào ma trận hệ số DCT $\mathbf{X_{embed}} = \mathbf{X}$.
  - Trường hợp P chẵn thì sử dụng thuật toán giấu tin **Outgess**, thực hiện nhúng tin $P=P_7P_6P_5P_4P_3P_2P_1P_0$ vào ma trận hệ số DCT $\mathbf{X_{embed}} = \mathbf{X}$ trong đó điệp viên tạo chuỗi giả ngẫu nhiên bằng sắp xếp zig-zag, sau đó dịch vòng theo chiều từ trên xuống (P mod 3) + 1 hàng.

GIẢI

T0	1	0	1	0	0	0	0	1	T4
	1	0	0	1	1	0	1	1	
T1	0	1	0	1	1	0	1	0	T5
	1	0	0	1	0	1	1	0	
T2	0	0	0	1	0	1	1	0	T6
	1	1	0	1	0	0	0	1	
T3	0	0	1	1	1	0	1	0	T7
	0	1	0	1	1	1	0	1	

➢ Bước 1:

Giả sử SBD = 91

K = (91 + 43) mod 255 =134

$134_{10} = 10000110_2$

$K_{extract}$			
1	0	0	0
0	1	1	0

➢ Bước 2a:

❖ Tính: T_i ^ $K_{extract}$

1	0	0	0
0	0	0	0

T_0^$K_{extract}$

0	0	0	0
0	0	0	0

T_1^$K_{extract}$

0	0	0	0
0	1	0	0

T_2^$K_{extract}$

0	0	0	0
0	1	0	0

T_3^$K_{extract}$

0	0	0	0
0	0	1	0

T_4^$K_{extract}$

1	0	0	0
0	1	1	0

T_5^$K_{extract}$

0	0	0	0
0	0	0	0

T_6^$K_{extract}$

1	0	0	0
0	1	0	0

T_7^$K_{extract}$

❖

SUM ($K_{extract}$)=3

✓ SUM (T_0^$K_{extract}$) = 1 < SUM ($K_{extract}$)=3
SUM (T_0^$K_{extract}$) mod 2 = 1 mod 2 = 1
➔ Q_0 = 1
➔ Q = $Q_7Q_6Q_5Q_4Q_3Q_2Q_1$**1**

✓ SUM (T_1^$K_{extract}$) = 0 => T_1 không nhúng tin
➔ Q_1 = 0
➔ Q = **0** $Q_7Q_6Q_5Q_4Q_3Q_2$1

✓ SUM (T_2^$K_{extract}$) = 1 < SUM ($K_{extract}$)=3

SUM (T_2^$K_{extract}$) mod 2 = 1 mod 2 = 1

➔ Q_2 = 1
➔ Q = 0 $Q_7Q_6Q_5Q_4Q_3$**1** 1

✓ SUM (T_3^$K_{extract}$) = 1 < SUM ($K_{extract}$)=3

SUM (T_3 ^ $K_{extract}$) mod 2 = 1 mod 2 = 1

➔ Q_3 = 1
➔ Q = 0 $Q_7Q_6Q_5Q_4$ **1** 1 1

✓ SUM (T_4 ^ $K_{extract}$) = 1 < SUM ($K_{extract}$)=3

SUM (T_4 ^ $K_{extract}$) mod 2 = 1 mod 2 = 1

➔ Q_4 = 1
➔ Q = 0 $Q_7Q_6Q_5$ **1** 1 1 1

✓ SUM (T_5 ^ $K_{extract}$) = 3 = SUM ($K_{extract}$)=3=> T_5 không nhúng tin
➔ Q = **0** 0 Q_7Q_6 1 1 1 1

✓ SUM (T_6 ^ $K_{extract}$) = 0 => T_6 không nhúng tin
➔ Q = **0** 0 0 Q_7 1 1 1 1

✓ SUM (T_7 ^ $K_{extract}$) = 2 < SUM ($K_{extract}$)=3

SUM (T_7 ^ $K_{extract}$) mod 2 = 2 mod 2 = 0

➔ Q_7 = 0
➔ Q = 0 0 0 **0** 1 1 1 1

➔Vậy Q = 0 0 0 0 1 1 1 1 $_2$ = 15 $_{10}$ (Nếu *Q lẻ* làm *Outguess*, *Q chẵn* làm *jsteg*. Ở trong trường hợp này, vì Q =15 là lẻ nên chỉ dùng thuật toán Outguess ở bước 2b nhé)

➢ Bước 2b:

b1, Giả sử nếu Q lẻ. Ta làm *Outguess*

435	-37	-64	0	45	0	1	0
3	0	1	13	-21	1	0	0
0	1	19	0	0	0	0	0
12	0	23	13	1	-47	1	1
1	15	0	0	0	17	0	0
23	0	-19	1	0	1	0	1
0	-11	0	0	1	29	1	0
1	1	0	0	1	0	1	1

- Sắp xếp zig-zag

435	-37	3	0	0	-64	0	1
1	12	1	0	19	13	45	0
-21	0	23	15	23	0	0	0
13	0	1	1	0	0	0	1
0	-19	-11	1	1	0	1	0
-47	0	0	0	1	17	0	0
0	0	1	1	0	1	0	0
29	1	0	1	1	0	1	1

- Dịch vòng từ trên xuống 3 hàng

-47	0	0	0	1	17	0	0
0	0	1	1	0	1	0	0
29	1	0	1	1	0	1	1
435	-37	3	0	0	-64	0	1
1	12	1	0	19	13	45	0
-21	0	23	15	23	0	0	0
13	0	1	1	0	0	0	1
0	-19	-11	1	1	0	1	0

- Kết quả

-47	0	0	0	1	17	0	0
0	0	1	1	0	1	0	0
29	1	0	1	1	0	1	1
435	-37	3	0	0	-64	0	1
1	12	1	0	19	13	45	0
-21	0	23	15	23	0	0	0
13	0	1	1	0	0	0	1

0	-19	-11	1	1	0	1	0

→ m =1 1 1 1 1 0 0 1

b2, Giả sử nếu Q chẵn. Ta làm *Jsteg*

435	-37	-64	0	45	0	1	0
3	0	1	13	-21	1	0	0
0	1	19	0	0	0	0	0
12	0	23	13	1	-47	1	1
1	15	0	0	0	17	0	0
23	0	-19	1	0	1	0	1
0	-11	0	0	1	29	1	0
1	1	0	0	1	0	1	1

435	3	0	12	1	23	0	1
-37	0	1	0	15	0	-11	1
-64	1	19	23	0	-19	0	0
0	13	0	13	0	1	0	0
45	-21	0	1	0	0	1	11
0	1	0	-47	17	1	29	0
1	0	0	1	0	0	1	1
0	0	0	1	0	1	0	1

➔ m =1 0 1 1 1 1 0 1

➢ Bước 3a:

P = (91+19) mod 255 =110_{10} = 01101110_2 (P chỉ để nhúng ở bước sau cùng thôi)

K_{embed}	
1	0
0	1
0	1
0	0

1	0
0	0
0	1
0	0

$T_1 \wedge K_{embed}$

1	0
0	1
0	1
0	0

$T_2 \wedge K_{embed}$

0	0
0	0
0	0
0	0

$T_3 \wedge K_{embed}$

0	0
0	1
0	0
0	0

$T_4 \wedge K_{embed}$

0	0

0	0

0	0

1	0

0	1
0	0
0	0

$T_5 \wedge K_{embed}$

0	1
0	1
0	0

$T_6 \wedge K_{embed}$

0	0
0	0
0	0

$T_7 \wedge K_{embed}$

0	1
0	0
0	0

$T_8 \wedge K_{embed}$

SUM (K_{embed})=3

✓ **Với T_1**

- 0 < SUM ($T_1 \wedge K_{embed}$) = 2 < SUM (K_{embed})= 3

  -> Có thể giấu bit thứ 1 là p=0 vào khối này (**0**1101110)

- SUM ($T_1 \wedge K_{embed}$) mod 2 = 2 mod 2 = 0; p= 0. Vậy SUM ($T_1 \wedge K_{embed}$) mod 2 = p
  --> Khối T_1 được giữ nguyên.

✓ **Với T_2**

- 0 < SUM ($T_2 \wedge K_{embed}$) = 3; SUM (K_{embed})= 3. Vì SUM ($T_2 \wedge K_{embed}$) = SUM (K_{embed})

-> Nên không giấu được dữ liệu vào trong T_2

✓ **Với T_3**

- SUM ($T_3 \wedge K_{embed}$) = 0

-> Nên không giấu được dữ liệu vào trong T_3

✓ **Với T_4**

- 0 < SUM ($T_4 \wedge K_{embed}$) = 1 < SUM (K_{embed})= 3

-> Có thể giấu bit thứ 2 là p= 1 vào khối này (0**1**101110)

- SUM ($T_4 \wedge K_{embed}$) mod 2 = 1 mod 2 = 1 ; p= 1. Vậy SUM ($T_4 \wedge K_{embed}$) mod 2 = p
  --> Khối T_4 được giữ nguyên.

✓ **Với T_5**

- 0 < SUM ($T_5 \wedge K_{embed}$) = 1 < SUM (K_{embed})= 3

-> Có thể giấu bit thứ 3 là p=1 vào khối này (01**1**01110)

- SUM ($T_5 \wedge K_{embed}$) mod 2 = 1 mod 2 = 1 ; p= 1. Vậy SUM ($T_5 \wedge K_{embed}$) mod 2 = p

--> Khối T_5 được giữ nguyên

✓ **Với T_6**

- 0 < SUM ($T_6 \wedge K_{embed}$) = 2 < SUM (K_{embed})= 3

-> Có thể giấu bit thứ 4 là p=0 vào khối này (011**0**1110)

- SUM ($T_6 \wedge K_{embed}$) mod 2 = 2 mod 2 = 0 ; p= 0. Vậy SUM ($T_6 \wedge K_{embed}$) mod 2 = p

--> Khối T_6 được giữ nguyên.

✓ **Với T_7**

- SUM(T_7 ^ K_{embed}) = 0

-> Nên không giấu được dữ liệu vào trong T_7

✓ **Với T_8**

- 0 < SUM (T_8 ^ K_{embed}) = 2 < SUM (K_{embed})= 3

-> Có thể giấu bit thứ 5 là p=1 vào khối này (01101110)

-
  - SUM(T_8 ^ K_{embed}) mod 2 = 2 mod 2 = 0 ; p= 1.Vậy SUM(T_8 ^ K_{embed}) mod 2 # p
  - SUM(T_8 ^ K_{embed}) = 2 # 1
  - SUM(T_8 ^ K_{embed}) = 2 ; SUM(K)-1=3-1=2. Vậy SUM(T_8 ^ K_{embed}) = SUM(K_{embed})-1

  Chọn ngẫu nhiên một bit thỏa mãn đồng thời $[K]_{11}$ =1 và $[T_8]_{11}$ =1 chuyển giá trị của bit $[T_8]_{11}$ từ 1 trở thành 0.

1	0
0	1
0	0
0	0

⟹

0	0
0	1
0	0
0	0

Vậy: ta thu được T’ sau khi nhúng P:

T'_1		T'_2		T'_3		T'_4	
1	0	1	0	0	0	0	0
0	0	0	1	0	0	0	1
0	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0
T'_5		T'_6		T'_7		T'_8	

Giấu tin bằng thuật toán Outguess (vì P chẵn)

➢ Bước 3b:

b1, Giả sử nếu Q lẻ. Ta làm *Jsteg*

P = 0 1 1 0 1 1 1 0

435	-37	-64	0	45	0	1	0
3	0	1	13	-21	1	0	0
0	1	19	0	0	0	0	0
12	0	23	13	1	-47	1	1
1	15	0	0	0	17	0	0
23	0	-19	1	0	1	0	1
0	-11	0	0	1	29	1	0
1	1	0	0	1	0	1	1

⇒

435	-36	-65	0	45	0	1	0
2	0	1	13	-21	1	0	0
0	1	19	0	0	0	0	0
12	0	23	13	1	-47	1	1
1	15	0	0	0	17	0	0
23	0	-19	1	0	1	0	1
0	-11	0	0	1	29	1	0
1	1	0	0	1	0	1	1

b2, Giả sử nếu Q chẵn. Ta làm *Outguess*

435	-37	-64	0	45	0	1	0
3	0	1	13	-21	1	0	0
0	1	19	0	0	0	0	0
12	0	23	13	1	-47	1	1
1	15	0	0	0	17	0	0
23	0	-19	1	0	1	0	1
0	-11	0	0	1	29	1	0
1	1	0	0	1	0	1	1

- Sắp xếp zig-zag

435	-37	3	0	0	-64	0	1
1	12	1	0	19	13	45	0
-21	0	23	15	23	0	0	0
13	0	1	1	0	0	0	1
0	-19	-11	1	1	0	1	0
-47	0	0	0	1	17	0	0
0	0	1	1	0	1	0	0
29	1	0	1	1	0	1	1

- Dịch vòng theo chiều trên xuống (p mod 3) + 1 = (110 mod 3) +1 =3 hàng.

-47	0	0	0	1	17	0	0
0	0	1	1	0	1	0	0
29	1	0	1	1	0	1	1
435	-37	3	0	0	-64	0	1
1	12	1	0	19	13	45	0
-21	0	23	15	23	0	0	0
13	0	1	1	0	0	0	1
0	-19	-11	1	1	0	1	0

- Thực hiện ẩn tin: P = 0 1 1 0 1 1 1 0

-47	0	0	0	1	17	0	0
0	0	1	1	0	1	0	0
29	1	0	1	1	0	1	1
435	-37	3	0	0	-64	0	1
1	12	1	0	19	13	45	0
-21	0	23	15	23	0	0	0
13	0	1	1	0	0	0	1
0	-19	-11	1	1	0	1	0

-46	0	0	0	1	17	0	0
0	0	1	1	0	1	0	0
29	1	0	1	1	0	1	1
435	-36	3	0	0	-65	0	1
1	13	1	0	18	13	45	0
-21	0	23	15	23	0	0	0
13	0	1	1	0	0	0	1
0	-19	-11	1	1	0	1	0

Note tí:

- Xanh : DC
  Hồng : AC

- Ở ví dụ trên: Q với P người ta cho đều là chẵn, nhưng mình làm thêm TH lẻ để mọi người hình dung cách làm thôii nha. Chứ trong bài làm, chỉ làm 1 TH là:bước 2b làm Q chẵn , bước 3b làm P chẵn thôi nhé.

- Link file word:
https://drive.google.com/file/d/1SkIPLE_kHVev_o2_KaoeTOP-x7mhuiv2/view?usp=sharing

- Không hiểu chỗ nào cứ mò slide xem qua 1 chút.