Trong thuật toán này, khoá đơn giản chỉ là kích thước của khối , nếu biết kích thước của khối thì dễ dàng giải mã tin theo quy tắc sau :

Quá trình giải mã tin :
Sau khi nhận được ảnh đã giấu tin, quá trình giải mã tin sẽ được thực hiện theo các bước sau đây :

- Đọc header và bảng màu của ảnh để biết các thông tin về ảnh.
- Đưa phần dữ liệu ảnh vào mảng hai chiều .

Các bước này giống với quá trình giấu tin. Sau khi đã có được dữ liệu ảnh , ta chia ảnh thành các khối có kích thước giống kích thước khối khi thực hiện giấu, đây chính là khoá để giải mã. Chọn ra các khối đã giấu và giải tin theo quy tắc : đếm số bít 1 trong khối, nếu tổng số bít 1 là lẻ thì thì thu được bit 1, ngược lại thu được bit 0. Cứ tiếp tục cho đến khi hết các khối đã giấu tin.
Như vậy, sau khi hết các khối đã giấu tin, ta thu được một chuỗi bít đã đem giấu . Bước tiếp theo ta chuyển từ file nhị phân sang file văn bản .

Phân tích thuật toán
Đây là thuật toán rất đơn giản thực hiện một cách thức giấu tin trong ảnh, sau khi nghiên cứu thuật toán này chúng ta có thể đưa ra một số bình luận và đánh giá như sau :

- Việc chọn kích thước khối để giấu tin tuỳ thuộc vào kích thước ảnh và lượng thông tin cần giấu sao cho giấu dàn trải trên toàn ảnh. Ví dụ, nếu ta có một ảnh có kích thước 512*512 pixel và có một lượng thông tin cần giấu là 100 ký tự. Như vậy, file nhị phân thông tin cần giấu sẽ là 100*8 = 800 bít 0/1, vì mỗi kí tự mã ASCII biểu diễn bởi 1 byte. Ta có thể thấy rằng : để giấu được hết thông tin thì cần ít nhất 800 khối, vậy thì ta nên chia khối như thế nào để đủ khối giấu và dàn trải rộng trên ảnh. Lấy (512*512)/800 =327 dư 544. Với kết quả này, kích thước khối tối đa là 327 vậy thì ta có thể chọn các kích thước phù hợp với con số này (phù hợp theo nghĩa đủ lớn và không vượt quá 327), chẳng hạn như 20*15, 16*16.
- Sở dĩ ta nên chọn khối có kích thước lớn vì như vậy nếu như trong trường hợp các khối bị thay đổi thì khoảng cách bít bị biến đổi sẽ xa nhau (thưa) lám cho anh sau khi giấu khó bị nhận biết hơn.
- Độ an toàn của thuật toán này là không cao, vì ta chỉ cần biết được kích thước của các khối giấu tin là ta có thể giải mã được nhanh chóng.
- Thuật toán ở trên hoàn toàn có thể áp dụng được đối với ảnh màu hoặc ảnh đa mức xám. Các loại ảnh này có giá trị của mỗi điểm ảnh được biểu diễn bởi nhiều bít. Vậy làm thế nào để có được một ma trận điểm ảnh 0,1 để thực hiện giấu tin như thuật toán trên. Rất đơn giản , ta chỉ việc chọn từ một điểm ảnh đúng một bít và lưu vào trong ma trận hai chiều các bít 0,1 . Việc chọn này được thực hiện theo quy tắc chọn bít quan trọng nhất **LSB – Least Significiant Bit**

- Đối với ảnh màu và ảnh đa mức xám ta không cần quan tâm nhiều đến việc chọn điểm cần giấu vì ta đã dùng những bít ít quan trọng nhất để giấu rồi. Do vậy, tại mỗi bước giấu ta có thể chọn một bít bất kỳ để thay đổi.

Cải tiến thuật toán :
Với thuật toán này việc chọn khối khá đơn giản, ta bắt đầu từ khối đầu tiên và những khối liên tiếp phía sau một cách tuần tự. Tuy nhiên, ta có thể cải tiến thuật toán bằng cách chọn ngẫu nhiên một khối chưa giấu ở mỗi lần giấu. Khi đó, ta đã làm tăng được độ an toàn của thuật toán vì khoá bậy giờ còn có thêm cả chỉ số khối đã giấu tin cho từng bit. Hoặc ta có thể thay đổi kích thước khối ở mỗi lần giấu, chẳng hạn như lần 1 có kích thước khối là 8*8, lần 2 là 8*12, trong trường hợp này thì khoá sẽ là kích thước khối ở mỗi lần giấu.
Một cách cải tiến thuật toán khác nữa là ta sẽ tính hệ số phân bố bit của một ma trận điểm ảnh. Hệ số phân bố bit là đại lượng đặc trưng cho mức độ rời rạc của bit 0 và 1 của ma trận đó.Việc chọn bít nào để đảo giá trị sẽ tuỳ thuộc vào hệ số bít của ma trận đó lớn hay nhỏ.Với cách này, việc giấu bít vào trong ảnh đen trắng là rất hiệu quả.

Kỹ thuật 2 - Kỹ thuật giấu tin của WU_LEE :
1. Một số khái niệm cơ bản :

* Phép nhân bit (AND)

Gọi a và b là bít tuỳ ý, phép tính toán nhân bít AND, ký hiệu là ^ trên hai bít a và b cho ta giá trị 1 khi và chỉ khi a=b=1, trong các trường hợp còn lại a^b =0

* Phép cộng loại trừ (XOR)
Phép toán cộng trừ (còn gọi là phép toán so khác) XOR, ký hiệu là ⊕ trên hai bít a và b cho ta giá trị 1 nếu a ≠ b và giá trị 0 nếu a=b.

* Bảng giá trị chân lý của hai phép toán trên:

a	b	a^b	a⊕b
1	0	0	1
1	1	1	0
0	1	0	1
0	0	0	0

* Phát triển 2 phép toán trên đối với 2 ma trận

Cho A và B là hai ma trận bít cùng cấp. Ta thực hiện các phép toán như sau:

- Nếu A= (a_{ij}), B = (b_{ij}), C= (c_{ij}), D = (d_{ij})

  thì A ^ B = C với c_{ij}= a_{ij} ^ b_{ij}

  và A ⊕ B = D với d_{ij} = a_{ij} ⊕ b_{ij}

* Tổng giá trị các phân tử trong ma trận

Ta định nghĩa SUM(X) là tổng ác giá trị trên ma trận X. Chú ý rằng nếu X là một ma trận bít thì SUM(X) chính là tổng số bít 1 trong X.

2. Ý tưởng của thuật toán Wu_Lee

- Sử dụng ma trận khoá bí mật K là một ma trận nhị phân có kích thước m x n (bằng kích thước của khối ảnh giấu tin) nhằm làm tăng độ antoàn của thuật toán. Nếu trước đây chỉ biết kích thước khối là mxn thì đói phương rất dễ khai thác được bản tin mật, nay ngoài kích thước này còn phải biết giá trị cụ thể của khoá K.

- Sử dụng phép toán AND giữa ma trận điểm ảnh và ma trận khoá (Fi^K) nhằm quy định thuật toán chỉ được phép sửa các bít trong khối Fi ứng với bít 1 trong khoá K. Như vậy khoá K được xem nhu một mặt nạ, tạo ra khung hình cho thuật toán, tăng độ an toàn.

- Sử dụng phép SUM (Tính giá trị các bít 1 trong các ma trận nhị phân) để kiểm tra điều kiện anh toàn thì thông tin được đấu. Điều kiện an toàn là 0<SUM (Fi^K)<SUM(K) có nghĩa là quy định nếu khối Fi^K toàn 0 hoặc giống nhu khoá K thì không được giấu tin để tránh bị lộ.

- Thông tin được giấu vào mỗi khối Fi các bít điểm ảnh chỉ là 1 bít. Khi thông tin được giấu, khối bìt F'i sau khi được giấu luôn đảm bảo tính bất biến: SUM(F'i) sau khi được giấu luôn đảm bảo tính bất biến: SUM (F'i^K) mod2=b (b chính là bít được giấu).

3. Thuật toán

Input:

- Một ảnh gốc nhị phân F.
- Một khoá bí mật K: là một ma trận nhị phân có kích thước m*n.
- Một file thông tin cần giấu P

Output:

- Một ảnh đã được giấu thông tin

Giấu tin:

Để cho đơn giản chúng ta coi kích cỡ của ảnh F là bội của m*n. Việc nhúng thông tin giấu vào trong ảnh sẽ được thực hiện bằng cách thay đổi một số bít của ảnh F theo quy tắc.

Bước 1: Chia ảnh F thành các khối nhỏ, mỗi khối có kích thước là m*n.

Bước 2: Với mỗi khối ảnh nhỏ Fi thu được từ bước S1, ta kiểm tra điều kiện an toàn khi giấu tin:

0 < SUM (Fi^K) < SUM(K)

Nếu đúng thì chuyển tới bước 3 để giấu thông tin vào trong khối Fi, còn nếu không thì không giấu dữ liệu vào trong khối Fi, khối Fi sẽ được giữ nguyên.

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

If(SUM(Fi^K)mod2=b) then

Giữ nguyên Fi

Else if (SUM(Fi^K)=1) then

Chọn ngẫu nhiên một bít (j,k) thoả mãn đồng thời $[Fi]_{ik} = 0$và $[K]_{ik} = 1$ sau đó, chuyển giá trị của bít $[Fi]_{ik}$ thành 0. else

Chọn ngẫu nhiên một bit mà $[K]_{jk} = 1$ chuyển giá trị của bít $[Fi]_{jk}$ từ 0 thành 1 hoặc từ 1 thành 0.

End if;

Việc chọn bít nào trong F để đảo cần tuân thủ theo nguyên tắc: Nếu Fi^K có nhiều bít 1 (SUM(Fi^K)= SUM(K)-1) thì chọn bít 1, ngược lại nếu FiAK có quá ít bít 1 (SUM (Fi^K)= 1) thì chọn 0 bít. Nguyên tắc này làm giảm khả năng bít đảo bị phát hiện.

Giải mã:

Nhờ bất biến có được khi giấu tin, ta dễ dàng giải mã để lấy lại thông tin đã giấu như sau. Duyệt lần lượt các khối Fi của ảnh đích F. Nếu Fi thoả điều kiện 0<SUM(Fi^K) <SUM(K) thì tính bít b đã được giấu vào trong khối bằng công thức b = SUM (F'i^K) mod 2.

4. Minh hoạ thuật toán.

F1 F2

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

F3 F4

Thông tin giấu B=011

0	1	0
1	1	1
0	1	0

K

F'1 F'2

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

Hình : Mô tả quá trình đảo bít để giấu tin của thuật toán trên 4 khối.

Giả sử một ảnh F có kích thước 6x6 và một ma trận khoá K có kích thước 3x3 như trong hình vẽ. Ta chia ảnh F thành 4 khối nhỏ mỗi khối sẽ có kích thước là 3x3 ta thu được F1, F2, F3, F4.

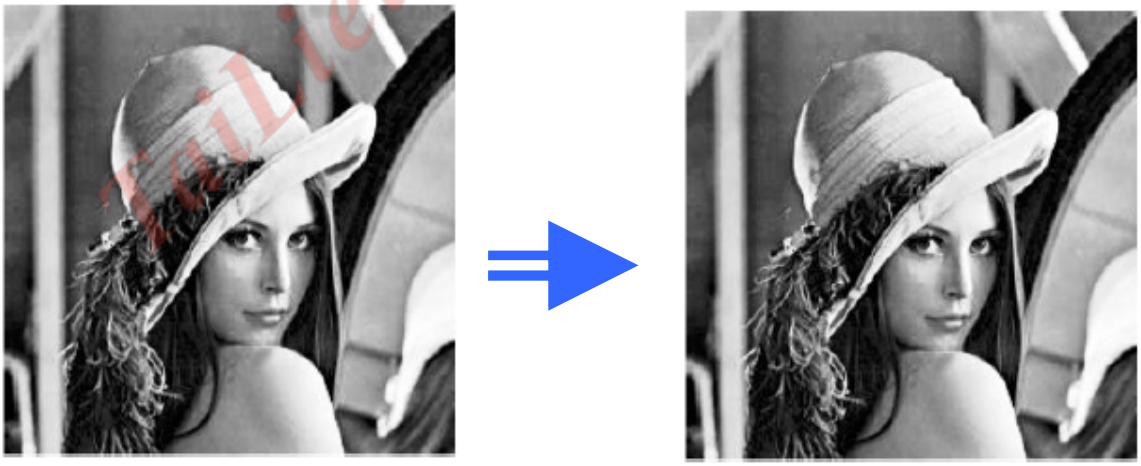
- Vì SUM(F1^K) =SUM(K) nên không giấu dữ liệu vào trong F1.

- Vì SUM(F2^K) =3 nên một bít có thể được giấy vào khối 2. Theo ví dụ trên bít đầu tiên được giấy là bít 0. Nên theo S3 ta sẽ chọn một bít có [F2]ij = 0 và [K]ij = 1 và đổi giá trị [F2]ij thành 1, F2 chuyển thành F2 như trên hình vẽ (bít đổi được đánh giấu xám).

- Với F3 SUM(F3^K) = 3 nhưng bít cần giấu là bít 1 nên theo Sl ta giữ nguyên F3 nhưng thực tế F3 vẫn được dấu một bít 1.

- Tương tự đối với F4, SUM (F1^K) = 4, và bít cần giấu là bít1 nên theo S3 ta chọn một bít ở [F4]ij = 1 và [K]ij= 1 rồi chuyển [F4]ij = 0. Bít thay đổi được đánh giấu xám.

Ví dụ minh hoạ:



Hình : Ảnh trước và sau khi giấu các bít thông tin

5. Phân tích đánh giá thuật toán

- Vì khoá K bí mật nên thông tin đã được nhúng là bí mật. Thuật toán này làm thay đổi nhiều nhất một bít của khối Fi khi giấu một bít thông tin vào trong khối nên với một khối có kích thức m*n đủ lớn thì sự thay đổi của Fi là nhỏ để đảm bảo được tính an toàn của thuật toán.

- Vì phép toán AND được sử dụng để tính Fi^K, nên giá trị lớn nhất của SUM(FI^K) không thể vượt quá SUM(K) và do tính chất của phép toán AND, nếu có một khối nào thay đổi thì vị trí thay đổi chỉ xảy ra ở phần tử có giá trị l trong khoá K. Vì thế, nếu một ảnh F hoàn toàn trắng nào đó được truyền đi thi kẻ thù khi bắt được thông tin sẽ dễ dàng tìm ra được vị trí l của khoá K, đó là lí do

mà ta không dùng trường hợp SUM (F1^K) - 0. Đây là một kẽ hở của thuật toán đối với khoá.

- Với trường hợp SUM (F1^K) = SUM (K) cũng tương tự nếu F hoàn toàn đen thì vị trí của bít thay đổi thì cũng là vị trí mà bít tương ứng ở khoá là 1.

Để tránh những trường hợp trên thuật toán đã phải đưa ra phụ thuộc 0 < SUM(F1^K) < SUM (K). Nhưng cho dù như thế đi chăng nữa thì vị trí tương ứng với bít bị thay đổi vũng tương ứng với bít ở vị trí đó trong khoá K có giá trị l, và bít không bao giờ bị thay đổi tương ứng sẽ là bít 0 ở vị trí đó trong khoá K. Và như thế việc chọn khoá K như thế nào là một công việc hết sức quan trọng.

- Nếu ảnh F được lựa chọn để giấu thông tin có quá nhiều điểm trắng hoặc quá nhiều điểm đen thì tỉ lệ bít giấu được sẽ thấp.

-Nói chung đối với ảnh đen trắng thuật toán này vẫn chưa đạt được những yêu cầu cần thiết về khả năng giấu. Độ an toàn thông tin đã phần nào được cải thiện hơn so với kỹ thuật thứ nhất. Vì vậy, chất lượng ảnh hưởng cũng tốt hơn. Tuy nhiên về vấn đề số lượng thông tin giấu lại giảm đi vì có những khối sẽ không được giấu tin .

6. Cải tiến thuật toán

Ta có thể cải tiến thuật toán này khi áp dụng đối với ảnh màu bằng cách khi ma trận bít không thoả mã điều kiện an toàn 0 < sum (Fi^K) < sum(K) ta vẫn thực hiện giấu tin theo ý tưởng của thuật toán, vì đối với ảnh màu ta đã áp dụng kỹ thuật tách các bit LSB, nên không ảnh hưởng đến chất lượng ảnh mà số lượng tin giấu sẽ nhiều hơn.

Kỹ thuật 3 - Kỹ thuật giấu tin CHAN_PAN_TSENG

Trong mục này đề cập đến một kỹ thuật đơn giản và đáng tin cậy để giấu những thông tin quan trọng vào trong một ảnh đen trắng (ảnh nhị phân) bằng cách sử dụng một khoá bí mật K (Private Key) và một ma trận trọng số do Yu-Yuan Chen, Hsiang – Kuang Pan và Yu – Chee Lseng thuộc khoa Công nghệ thông tin và Khoa học máy tính, Đại học Quốc gia Đài Loan nghiên cứu. Phương pháp này được chứng minh là có độ an toàn dữ liệu cao, bảo đảm chất lượng ảnh gốc và có tỷ lệ giữa kích thước thông tin giấu được với kích thước ảnh môi trường tương đối lớn so với các phương pháp khác và cho phép giấu được tới $[\log_2 (m*n +1)]$ bít dữ liệu vào trong mỗi khối ảnh có kích thước m*n mà chỉ cần thay đổi nhiều nhất 2 bít trong khối ảnh đó.

1.Một số khái niệm dùng trong thuật toán

- ***Khoá bí mật*** : Khoá là một ma trận nhị phân có cùng kích thước m*n với kích thước của khối ảnh. Khoá được dùng một cách bí mật giữa người gửi và người nhận.

- ***Ma trận trọng số cấp r :*** Ma trận trọng số W cấp r là một ma trận số nguyên có kích thước bằng kích thước của khối ảnh m*n và thoả mãn các điều kiện sau :
  - W có các phần tử nằm trong khoảng giá trị $(0..2^{r-1})$ với r cho trước thoả mãn điều kiện sau : $2^r < m*n$
  - Mỗi một phần tử có giá trị từ $1..2^{r-1}$ xuất hiện ít nhất một lần.

Ví dụ : Xây dựng một ma trận trọng số kích thước 4*4, ma trận này chứa các giá trị nằm trong khoảng 0..15 và mỗi giá trị từ 1,2,3..7 xuất hiện ít nhất một lần.

Một ví dụ ma trận trọng số W :

1	2	3	4
5	6	7	1
2	3	4	5
4	5	7	1

Các phép toán trên ma trận dùng trong thuật toán :

Phép toán XOR hai ma trận : $A \oplus B$.

Phép toán nhân hai ma trận $A \otimes B$.

Phép toán tính tổng giá trị trong ma trận SUM(X).

2. Ý tưởng của thuật toán CHEN_PANG_TSENG

Thuật toán sử dụng một ma trận khoá và một ma trận trọng sô để giấu thông tin. Việc sử dụng thêm một ma trận trọng sô W và phép toán XOR giữa ma trận điểm ảnh và ma trận khoá sẽ làm cho thuật toán đảm bảo được tốt an toàn thông tin và cũng giấu được nhiều thông tin hơn trong mỗi khối ảnh bằng cách thay đổi nhiều nhất 2 bít mỗi khối ảnh .

3. Thuật toán giấu tin trong ảnh CHEN_PANG_TSENG

Input:

- F : Một ma trận ảnh gốc dùng để giấu thông tin . F được chia thành các khối F_i , mỗi ma trận điểm ảnh F_i có kích thước là m*n, để cho đơn giản ta giả sử rằng F là bội của các F_i.

- K : Một ma trận khoá cấp m*n

- W : Ma trận trọng số cấp m*n

- r : Số lượng bít sẽ nhúng trong mỗi khối ảnh m*n

- B : Là lượng thông tin cần giấu gồm k*r bít, k sẽ là số khối ảnh được giấu.

- d : Độ chênh lệch trọng số.

Output :

Ảnh đã nhúng tin F' chứa B, F' được tạo từ các khối đã nhúng tin F'_i . Mỗi khối F'_i thu được từ khối F_i tương ứng sau khi đã giấu r bít thông tin từ B.

Giấu tin :

Thuật toán sẽ thực hiện việc giấu tin bằng cách biến đổi mỗi khối bít F_i thành F'_i sao cho luôn thoả mãn điều kiện sau :

$$SUM(F_i \oplus K) \otimes W) \equiv b_1b_2\ldots b_r \pmod{2^r} \quad (*)$$

Trong đó, $b_1b_2\ldots b_r$ là dạng biểu diễn nhị phân tạo từ dãy r bít liên tiếp trong B.

Mỗi khối bít F_i bị biến đổi nhiều nhất là 2 bít thông tin (Vì cần tạo ra điều kiện có trường hợp sẽ phải thay đổi trong khối ảnh F_i 2 bit thông tin).

Quá trình biến đổi được thực hiện gồm 4 bước sau đây :

Bước 1:

Tính ma trận $T = F_i \oplus K$

Tính ma trận $P = T \otimes W$

Bước 2:

Tính tổng giá trị trong ma trận P, Sum = SUM(P).

Bước 3:

Với ma trận T và với mọi $w = 1,2,\ldots,2^r - 1$, ta xác định tập hợp S_w như sau :

$$S_w = \{(j,k) \mid (W[j,k] = w \wedge T[j,k] = 0) \vee (W[j,k] = 2^r - w \wedge T[j,k] = 1)\}$$

Dễ nhận thấy rằng : S_w là tập hợp các toạ độ (j,k) của ma trận F_i sao cho khi đảo bít F_i [j,k] thì Sum ở bước 2 tăng thêm w đơn vị. Thực vậy, ta có :

- Trường hợp 1 : Nếu W[j,k] = w và T[j,k] = 0 , khi đó đảo bít F_i [j,k] sẽ làm cho bít T[j,k] = 1, do đó Sum tăng lên w.
- Trường hợp 2 : Nếu $W[j,k] = 2^r - w$ và T[j,k] = 1, khi đó đảo bít F_i [j,k] sẽ làm cho T[j,k] = 0, do đó Sum sẽ giảm đi $2^r - w$, tức là tăng lên w theo mod 2^r . Từ định nghĩa của tập S_w , ta có : $S_{w'} = S_w$

Bước 4 :

Ký hiệu $d = (b_1b_2\ldots b_r) - SUM(P) \pmod{2^r}$.

Ta cần thực hiện việc đảo bít F_i để được F'_i , sao cho tổng Sum tính được ở bước 2 khi thay F_i bởi F'_i sẽ tăng lên d.

- Nếu d = 0, không thay đổi F_i
- Nếu $d \neq 0$, ta thực hiện các công việc sau :

  1 .Chọn h bất kỳ thuộc tập $\{1,2,\ldots.2^r - 1\}$ sao cho $S_{hd} \neq \varnothing$ và $S-(h-1)d \neq \varnothing$

2. Chọn phần tử (j,k) bất kỳ thuộc S_{hd} và đảo bít F_i [j,k] (nếu là bit 0 thì đổi thành 1 và ngược lại)

3. Chọn phần tử (j,k) bất kỳ thuộc $S_{-(h-1)d}$ và đảo bít F_i [j,k]

Rõ ràng, để tăng Sum lên d, ta có thể chon hai tập khác rỗng là S_{hd} và $S_{-(h-1)d}$ .Thật vậy, hai tập này chứa các vị trí bít trong khối F_i mà ta có thể đảo để tăng Sum lên hd và –(h-1)d một cách tương ứng , kết quả cuối cùng là Sum sẽ tăng lên hd + (-(h-1)d) = d.

Tương tự như các tập S_w khác, ta có thể coi tập S_0 là chứa các vị trí mà khi đảo những bít có vị trí này trên F_i thì sẽ tăng Sum lên 0. Kết quả này cũng đạt được nếu ta không đảo bất kỳ một bít nào trên F_i. Vì vậy, ta có thể coi S_0 là tập trống và khi nói đảo 1 bít có vị trí thuộc tập S_0 có nghĩa là không cần làm gì cả.

Ví dụ minh hoạ :

Giả sử ta có một ma trận ảnh F 8*8 được chia thành 4 ma trận khối ảnh F1, F2, F3, F4 có cùng cỡ 4*4, một ma trận khoá K 4*4 và một ma trận trọng số có cùng cỡ như sau:

Ma trận F 8*8 :

F1 F2

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

F3 F4

K =

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

W =

1	2	3	4
5	6	7	1
2	3	4	5
6	7	1	2

Trong ví dụ này, ta chọn m = n = 4, chọn r = 3, ta giấu bít sau : B = 001010000001 vào trong ảnh F. Như vậy, đoạn bít 001 sẽ được giấu vào khối F1, 010 vào trong F2, 000 vào trong F3, 001 vào trong F4.

Bước 1:

Ta thực hiện phép toán XOR của Fi và K. Kết quả như sau :

F1 ⊗ K

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

F3 ⊗ K

F2 ⊗ K

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

F4 ⊗ K

Bước 2:

- Ta thực hiện phépnhân từng khối bít ma trận kết quả ở trên với ma trận trọng số :

$(F1 \oplus K) \otimes W$ $(F2 \oplus K) \otimes W$

1	0	0	4	1	0	3	0
5	6	0	0	5	0	7	1
0	3	4	0	2	3	4	0
0	0	1	0	0	0	0	2
0	0	3	0	0	2	0	0
0	6	7	1	5	6	7	0
0	0	4	5	2	0	0	5
6	0	0	2	0	7	0	2

$(F3 \oplus K) \otimes W$ $(F4 \oplus K) \otimes W$

Với F1 :

Chú ý rằng : Ta có $2^3 = 8$

Tính SUM $((F1 \oplus K) \otimes W) = 0 \pmod 8$. Vì chuỗi 3 bít cần giấu đầu tiên là 001 nên ta phải thay đổi để tăng trọng số lên 1(Vì $d = 1 - 0 \pmod{2^r} = 1$)

Ta xây dựng tập S_1 , với h =1 :

Ta nhận thấy , tại ô (2,4) thì W[2,4] = 0 và T[2,4] = 0, thoả mãn điều kiện theo thuật toán . Vậy $S_1 = \{(2,4)\} \neq \varnothing$, ta chọn luôn ô này để đảo bít . Khi đó ma trận khối ảnh F1 là :

F'_1

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