CHAPTER 5

A LARGE BLOCK CIPHER INVOLVING KEY DEPENDENT PERMUTATION, INTERLACING AND ITERATION
5.1. INTRODUCTION

In the previous chapter, we have developed a block cipher, which involves key dependent permutation, interlacing and iteration. This analysis is, in a way, a modification of the classical Hill cipher. In this, the size of the key matrix is 384 bits and the size of the plaintext is 112 bits. Here we have seen that the strength of the cipher enhances significantly as the multiplication of the plaintext with the key matrix leads to thorough diffusion and the permutation together with interlacing offers a marked confusion.

In the present chapter, our objective is to generalize the preceding analysis by considering a plaintext whose size is as large as we please. In this, we have taken the key matrix to be of size $n \times n$ and the plaintext as a matrix of size $n \times m$, $m > 2$. Thus, in this cipher, all the plaintext can be taken as a single block and the processes of encryption and decryption can be carried out.

In section 5.2, we introduce the development of the cipher. In section 5.3, we present the algorithms for encryption and decryption. Then in section 5.4, we illustrate the cipher with a couple of examples. Subsequently, we discuss the avalanche effect and cryptanalysis in sections 5.5 and 5.6. Finally, we present the computations and conclusions in section 5.7.
5.2. DEVELOPMENT OF THE CIPHER

Consider a plaintext. On using the ASCII code, we write the plaintext in the form of a matrix $P=[P_{ij}]$, $i=1$ to $n$, $j=1$ to $m$ (pad if needed).

Let $K=[K_{ij}]$, $i=1$ to $n$, $j=1$ to $n$ be the key matrix.

Let $C=[C_{ij}]$ represents the ciphertext corresponding to the plaintext $P$.

Following Hill [105], the relations for encryption and decryption can be written as

$$C=KP \mod 128$$  \hspace{1cm} (5.2.1)

and

$$P=K^{-1}C \mod 128.$$  \hspace{1cm} (5.2.2)

where $K^{-1}$ is the modular arithmetic inverse of $K$.

Let us now introduce the process of permutation and interlacing. On writing each element of the matrix $[P_{ij}]$ in terms of binary bits, we have $[P_{ij}]=[b_{ij}^l]$, $i=1$ to $n$, $j=1$ to $m$, $l=1$ to 7.

Thus, each column of $[P_{ij}]$ is represented as a matrix of size $nx7$, and hence we have $m$ such matrices.

Let us now take $7n$ numbers (ranging from 1 to $7n$), in the order in which they appear in the key matrix and form a subkey.

We now focus our attention on the matrix corresponding to the first column of $[P_{ij}]$ (the size of this matrix is $nx7$). The elements of this are permuted by using the subkey (of size $7n$) mentioned above. Then, the
The aforementioned procedure is applied for the matrices corresponding to all the other columns of \([P_{ij}]\).

Thus we get a new matrix, which includes all the permuted matrices, of size \(n \times 7m\) and is denoted by \([e_{ij}]\). This \([e_{ij}]\) is divided into two equal halves, wherein each half contains \(7m/2\) columns, if \(m\) is an even number. Otherwise, it will be divided into two parts, wherein the left part contains \((7m+1)/2\) columns and the right one is having \((7m-1)/2\) columns.

Then, we place the first column of the right half next to the first column of the left half. The second column of the right half next to the second column of the left half, and so on till we exhaust all the columns of the right half. This completes the process of interlacing.

The reverse processes of interlacing and permutation are denoted as decomposition and inverse permutation respectively. These two are utilized in decryption.

In this cipher, we adopt an iterative procedure, which consists of 16 rounds. The procedures of encryption and decryption are depicted in the diagram shown in Fig. 5.1.
In this analysis, \( N \) denotes the number of iterations and it is taken as 16.
5.3. ALGORITHMS

The algorithms describing encryption, decryption, modular arithmetic inverse, permutation, interlace, inverse permutation and decompose are given below.

5.3.1 ALGORITHM FOR ENCRYPTION
1. read n, N, K, P;
2. P⁰ = P;
3. P¹ = K⁰ mod 128;
4. for i=2 to N
   { 
      Permute();
      interlace();
      Pᵢ = Kᵢ₋¹ mod 128;
   }
5. C = Pᴺ;
6. write C;

5.3.2 ALGORITHM FOR DECRYPTION
1. read n, N, K, C;
2. find modinverse(K);
3. Pᴺ = C;
4. for i=N to 2
   { 
      Pᵢ₋¹ = K⁻¹ᵢ₋¹ mod 128;
      decompose();
      invpermute();
   }
5. P⁰ = K⁻¹₁ mod 128;
6. P = P⁰;
7. write P;

5.3.3 ALGORITHM FOR MODINVERSE
1. read n, K;
2. find Kᵢⱼ, Δ; /* Kᵢⱼ are the cofactors of the elements of K, and Δ is the determinant of K */
3. find d such that (dΔ) mod 128 = 1; /*d is the multiplicative inverse of Δ*/
4. K⁻¹ᵢⱼ = (Kᵢⱼd) mod 128;

5.3.4 ALGORITHM FOR PERMUTE
1. convert Pᵢ into binary bits;
2. construct [eᵢⱼ], i=1 to n, j=1 to 7m;
3. generate subkey;
4. for l=0 to (m-1)
   {
      k=1;
      for i=1 to n
      { 

for j=(7l+1) to (7l+7)
{
    temp[subkey[k]]=e_{ij}
    k++;
}
}
k=1;
for i=1 to n
{
    for j=(7l+1) to (7l+7)
    {
        e_{ij}=temp[k];
        k++;
    }
}
}

5.3.5 ALGORITHM FOR INVPERMUTE
1. convert $P_i$ into binary bits;
2. construct $[e_{ij}]$, $i=1$ to 8, $j=1$ to 14;
3. generate subkey;
4. for $l=0$ to $(m-1)$
   {
     k=1;
     for i=1 to n
     {
         for j=(7l+1) to (7l+7)
         {
             temp[k]=e_{ij}
             k++;
         }
     }
     k=1;
     for i=1 to n
     {
         for j=(7l+1) to (7l+7)
         {
             e_{ij}=temp[subkey[k]];
             k++;
         }
     }
   }

5.3.6 ALGORITHM FOR INTERLACE
1. $l=1$;
2. convert $P$ into binary bits;
3. for $i=1$ to $n$
   {
   }
for j=1 to 7
{
    temp(l) = b_\text{j};
    temp(l+1) = d_\text{j};
    l=l+2;
}

4. l=1;
5. for i=1 to n
{
    for j=1 to 7
    {
        b_\text{j}=temp(l);
        d_\text{j}=temp(l+n*7);
        l=l+1;
    }
}

5.3.7 ALGORITHM FOR DECOMPOSE

1. l=1;
2. convert P into binary bits;
3. for i = 1 to n
   {
        for j=1 to 7
        {
            temp(l)=b_\text{j};
            temp(l+n*7)=d_\text{j};
            l = 1 + 1;
        }
   }

4. l=1;
5. for i = 1 to n
   {
        for j= 1 to 7
        {
            b_\text{j} = temp(l);
            d_\text{j} = temp(l+1);
            l=1 + 2;
        }
   }

6. convert binary bits to decimal numbers;
5.4. ILLUSTRATION OF THE CIPHER

Let us consider the plaintext given below.

I am quite sure to assert that all the terrorists entered in to the jungle.
Let us burn the forest without any lapse of time. Peace cannot be
restored unless we do this immediately. Wish you best of luck.  \(5.4.1\)

Let us focus our attention on the first sixty four characters of the
above plaintext given by

I am quite sure to assert that all the terrorists entered in to\(\phi\) \(5.4.2\)

On using ASCII code, these characters can be represented as a
matrix of size 8x8 and it assumes the form

\[
P^0 = \begin{bmatrix}
73 & 32 & 97 & 109 & 32 & 113 & 117 & 105 \\
116 & 101 & 32 & 115 & 117 & 114 & 101 & 32 \\
116 & 111 & 32 & 97 & 115 & 115 & 101 & 114 \\
116 & 32 & 116 & 104 & 97 & 116 & 32 & 97 \\
101 & 114 & 114 & 111 & 114 & 105 & 115 & 116 \\
115 & 32 & 101 & 110 & 116 & 101 & 114 & 101 \\
100 & 32 & 105 & 110 & 32 & 116 & 111 & 32 \\
\end{bmatrix} (5.4.3)
\]

The key matrix \(K\) is given by

\[
K = \begin{bmatrix}
53 & 62 & 24 & 33 & 49 & 18 & 17 & 43 \\
45 & 12 & 63 & 29 & 60 & 35 & 58 & 11 \\
8 & 41 & 46 & 30 & 48 & 32 & 5 & 51 \\
47 & 9 & 38 & 42 & 2 & 59 & 27 & 61 \\
57 & 20 & 6 & 31 & 16 & 26 & 22 & 25 \\
56 & 37 & 13 & 52 & 3 & 54 & 15 & 21 \\
36 & 40 & 44 & 10 & 19 & 39 & 55 & 4 \\
14 & 1 & 23 & 50 & 34 & 0 & 7 & 28 \\
\end{bmatrix} (5.4.4)
\]
On using the key matrix \( K \) and the plaintext \( P \), we apply (5.2.1) and obtain the modified \( P \), denoted by \( P^1 \), as

\[
P^1 = \begin{bmatrix}
62 & 78 & 69 & 53 & 63 & 37 & 52 & 31 \\
110 & 83 & 12 & 55 & 4 & 11 & 26 & 78 \\
87 & 95 & 36 & 33 & 41 & 49 & 114 & 91 \\
79 & 69 & 105 & 81 & 107 & 35 & 0 & 40 \\
61 & 50 & 104 & 87 & 97 & 75 & 0 & 34 \\
51 & 12 & 124 & 66 & 36 & 93 & 61 & 117 \\
8 & 94 & 65 & 103 & 88 & 1 & 119 & 33 \\
115 & 22 & 117 & 98 & 120 & 122 & 32 & 57
\end{bmatrix}
\]  
(5.4.5)

By applying the process of permutation, described in section 5.2, we get the transformed \( P^1 \) as

\[
P^1 = \begin{bmatrix}
59 & 71 & 123 & 37 & 57 & 33 & 12 & 24 \\
119 & 112 & 20 & 127 & 21 & 97 & 40 & 44 \\
29 & 62 & 116 & 68 & 31 & 101 & 102 & 92 \\
63 & 67 & 112 & 30 & 88 & 120 & 88 & 80 \\
66 & 36 & 88 & 58 & 31 & 25 & 50 & 71 \\
102 & 21 & 72 & 49 & 12 & 91 & 66 & 90 \\
15 & 31 & 48 & 114 & 50 & 35 & 21 & 47 \\
114 & 117 & 46 & 9 & 96 & 42 & 19 & 122
\end{bmatrix}
\]  
(5.4.6)

On applying the interlacing process (see section 5.2) on \( P^1 \), we obtain

\[
P^1 = \begin{bmatrix}
31 & 75 & 72 & 43 & 66 & 93 & 18 & 97 \\
85 & 90 & 18 & 98 & 79 & 4 & 53 & 29 \\
86 & 59 & 124 & 1 & 80 & 120 & 38 & 103 \\
12 & 96 & 93 & 122 & 97 & 4 & 54 & 70 \\
7 & 119 & 61 & 57 & 11 & 46 & 13 & 47 \\
124 & 52 & 98 & 112 & 22 & 17 & 92 & 93 \\
55 & 106 & 106 & 74 & 124 & 8 & 92 & 102 \\
118 & 64 & 39 & 40 & 19 & 45 & 43 & 70
\end{bmatrix}
\]  
(5.4.7)

After carrying out all the sixteen rounds, we get the ciphertext in the form
The modular arithmetic inverse of $K$, denoted by $K^{-1}$, is given by

$$K^{-1} = \begin{bmatrix}
27 & 40 & 53 & 3 & 117 & 48 & 25 & 2 \\
41 & 60 & 17 & 92 & 5 & 21 & 106 & 81 \\
57 & 39 & 116 & 118 & 18 & 0 & 37 & 116 \\
94 & 97 & 52 & 27 & 94 & 102 & 104 & 19 \\
63 & 123 & 117 & 0 & 98 & 9 & 97 & 32 \\
61 & 50 & 54 & 60 & 101 & 12 & 69 & 56 \\
64 & 41 & 57 & 22 & 73 & 75 & 49 & 122 \\
71 & 61 & 17 & 32 & 42 & 88 & 81 & 113
\end{bmatrix} \quad (5.4.9)$$

By applying $K^{-1}$ on the ciphertext $C$, from (5.2.2), we get

$$P^N = \begin{bmatrix}
100 & 126 & 123 & 26 & 13 & 10 & 38 & 94 \\
24 & 22 & 54 & 51 & 116 & 93 & 102 & 44 \\
50 & 36 & 84 & 84 & 114 & 99 & 108 & 16 \\
113 & 70 & 2 & 99 & 106 & 90 & 90 & 58 \\
125 & 75 & 28 & 38 & 29 & 107 & 22 & 22 \\
29 & 25 & 106 & 55 & 47 & 101 & 31 & 12 \\
6 & 42 & 73 & 60 & 63 & 57 & 27 & 49 \\
100 & 8 & 81 & 19 & 27 & 1 & 16 & 65
\end{bmatrix} \quad (5.4.10)$$

On applying the decomposition algorithm (see sections 5.2 and 5.3),

the transformed $P^N$ assumes the form
We now apply the inverse permutation algorithm described in section 3 on the $P_N$ obtained above and get the new $P_N$ as

$$
P_N = \begin{bmatrix} 83 & 116 & 18 & 52 & 75 & 81 & 21 & 21 \\ 79 & 74 & 91 & 77 & 57 & 44 & 104 & 0 \\ 54 & 52 & 96 & 41 & 71 & 107 & 31 & 33 \\ 15 & 83 & 1 & 15 & 98 & 24 & 50 & 79 \\ 116 & 54 & 49 & 9 & 2 & 92 & 66 & 39 \\ 90 & 108 & 77 & 7 & 40 & 10 & 14 & 74 \\ 91 & 0 & 112 & 75 & 7 & 126 & 110 & 63 \\ 60 & 94 & 115 & 81 & 54 & 85 & 91 & 5 \end{bmatrix} \quad (5.4.12)$$

After carrying out all the sixteen rounds, we get the deciphered text in the form

$$

This is same as the plaintext given in (5.4.3).
Let us now consider another example, wherein we have taken the complete plaintext given by (5.4.1). This plaintext is containing 207 characters. To represent this in the form of a matrix consisting of \( n \) rows and \( m \) columns, where \( n = 8 \) and \( m \) is having an appropriate value, depending upon the number of characters, we add one more character ($ is added here) to the plaintext. With this padding, the plaintext can be represented in terms of ASCII codes is given in (5.4.14).

\[
\begin{array}{cccccccc}
73 & 32 & 97 & 109 & 32 & 113 & 117 & 105 \\
116 & 101 & 114 & 101 & 100 & 32 & 105 & 110 \\
101 & 116 & 32 & 117 & 115 & 32 & 98 & 117 \\
111 & 117 & 116 & 32 & 97 & 110 & 121 & 32 \\
97 & 99 & 101 & 32 & 99 & 97 & 110 & 110 \\
101 & 115 & 115 & 32 & 119 & 101 & 32 & 100 \\
46 & 32 & 32 & 87 & 105 & 115 & 104 & 32 \\
\end{array}
\]

(5.4.14)

Then, on adopting the process of encryption, we get the ciphertext, in hexadecimal notation, as shown below:

F45CE2BBB263629C83A3DF35B015BB4574DD1A8C45A5CFD0C93D6 107DE4C2025E6D342505CD0206BC8FC8E55134C2F48DD61EC68739 A4F0C60CA5886728398191B858BB5E47B241D3A4E76D4FC0CFEBCA A749F72B67D8EE12922F3276FD80FAFB80ADA008D154E92C5942 BAB7989A4C19CF0DFB37F761A6B9EB5DB2B4E89034162B3CF4A841 0DD3A00435. (5.4.15)

On using the process of decryption, we readily find that this ciphertext can be brought into the form of the original plaintext.

In what follows, we discuss the avalanche effect which exhibits the effects confusion and diffusion caused by the basic elements (key, Iteration, Permutation and Interlacing) of the cipher.
5.5. AVALANCHE EFFECT

By applying the encryption algorithm on the plaintext given in (5.4.2), using the key matrix $K$, the corresponding ciphertext can be obtained as

\[
\begin{align*}
110111000011111100010110000011110101101001100100000010111110 \\
011000111111001100010100100100111101111011101100101110001 \\
001011011000001000110100111110010001111111100000111101111 \\
000110101101101100011010100111111011110110011001101011100 \\
11110101100110110110011111101001100110011010000001 \\
10111000010110001101010011010011101010010010001101010101 \\
1110000011110110110011111011001110110011000001011101011000000 \\
1010101001111010100011111001111011011110101101110011. \quad (5.5.1)
\end{align*}
\]

We now change the third character of the plaintext given in (5.4.2) from a to c. As the ASCII codes of a and c are 97 and 99. Then the plaintext will have a change of one binary bit.

The ciphertext corresponding to the modified plaintext is

\[
\begin{align*}
10011101100010101011111011011100010100010011001101010010010 \\
001111010000101010111110110110001100110111110101011011111011110 \\
110001001111101100000010100000000100000010100100000110011001000000 \\
00001000101100001010110001000001111111001001111101011110110110111 \\
00101110001010011001001100100111100111001100110011110111011110111111 \\
001110000000001110111011111001011110110111110110110010001 \\
010001101101000111110101100111111111110111111110111101111011111011110110011 \\
100001110001101100110110010100101000110001001001. \quad (5.5.2)
\end{align*}
\]

We readily notice that the ciphertexts given in (5.5.1) and (5.5.2) differ by 224 bits (out of 448 bits), which is substantial.

Let us now change the key matrix element $K_{25}$ from 60 to 62. With this change, the original key and the modified key differ by one bit. On using the modified key and the original plaintext, and applying the encryption algorithm, we get the ciphertext in the form

\[
\begin{align*}
0000110111111101110100000011111000111110101100000111101110101110001 \\
010010001101100111111100111100110001111110100100100011111110111000110110101010101
\end{align*}
\]
The ciphertexts given in (5.5.1) and (5.5.3) differ by 234 bits (out of 448 bits), which is also very significant.

From the above discussion, we conclude that the avalanche effect is highly pronounced.

5.6. CRYPTANALYSIS

The different types of attacks which are found in the literature of cryptology are: (1) Ciphertext only (Brute Force) attack, (2) Known plaintext attack, (3) Chosen plaintext and (4) Chosen ciphertext attack.

However, we restrict our attention to the first two cases (See William Stallings [21]), and discuss them in detail.

Let us now consider the brute force attack.

As the key matrix is chosen to contain the numbers 0 to 63, the size of the key space is $2^{6n^2}$.

If the time required for the execution of the cipher with one value of the key in the key space is $10^{-7}$ seconds, then the time required for the entire computation (i.e., for all the keys in the key space) is

$$\frac{2^{6n^2} \times 10^{-7}}{365 \times 24 \times 60 \times 60} \approx 2^{6n^2} \times 3.12 \times 10^{-15} \approx 10^{1.8n^2} \times 3.12 \times 10^{-15} = 3.12 \times 10^{1.8n^2-15} \text{ years.}$$
This is very large, when n≥4. When n = 4, the time required is $3.12 \times 10^{13.8}$ years. Hence, this cipher cannot be broken by the ciphertext only (brute force) attack.

Let us now discuss the known plaintext attack. In this, we know as many pairs of plaintext and ciphertext as we require for our analysis. The basic equations governing the cipher are

1. $P^0 = P,$ \hspace{1cm} (5.6.1)
2. $P^i = (KP^{i-1}) \mod 128,$ \hspace{1cm} (5.6.2)
3. $P^i = \text{permute}(P^i),$ i = 2 to 16, \hspace{1cm} (5.6.3)
4. $P^i = \text{interlace}(P^i),$ i = 2 to 16. \hspace{1cm} (5.6.4)

For the sake of convenience, let us represent the function permute as $\Phi,$ and the function interlace as $\langle \rangle.$

Thus, we can write (5.6.3) and (5.6.4) as

1. $P^i = \Phi(P^i),$
2. $P^i = \langle P^i \rangle$

respectively.

Then, the relation connecting C and P, at the end of the sixteenth iteration, can be written in the form

$$C = (K(\langle \Phi((K...\langle \Phi((K \Phi(KP) \mod 128)>\mod 128)>...)\mod 128)>)) \mod 128.$$
As P is involved under the operations permutation and interlacing at various stages of the iteration process, it cannot be taken out and transferred to the left side by obtaining the modular arithmetic inverse of P as it could be done in the case of the classical Hill cipher. Thus this cipher cannot be broken by adopting the known plaintext attack.

In the light of the above discussion, we conclude that it is difficult to break this cipher by employing the conventional cryptanalytic techniques.

5.7. COMPUTATIONS AND CONCLUSIONS

In this chapter, we have extended the analysis of the previous chapter by considering a plaintext of any size. In this analysis, we have illustrated the cipher by considering two cases. In the first one, the plaintext is an 8x8 matrix, and in the second one it is of size 8x26.

The algorithms designed in this analysis are implemented in C language.

As the key size and the plaintext size are significantly large, and as the iteration together with the permutation and the interlacing are effectively leading to diffusion and confusion, the cipher is found to be a very strong one. From the cryptanalysis we conclude that it is difficult to break the cipher.

In the case of the complete plaintext, which is taken in the form of a single block, the time required for encryption is $8.5 \times 10^{-3}$ seconds, and the time required for decryption is $13 \times 10^{-3}$ seconds. These results
indicate that the algorithm is quite efficient and it can be applied in any context for transmission of information.

The entire plaintext given by (5.4.1) is encrypted by taking 64 characters at a time. The ciphertext is shown below.

This analysis can be extended to the case, wherein we take multiple key matrices so that the process is strengthened further.