Modified Hill Cipher with Key Dependent Permutation and Circular Rotation

V. U. K. Sastry and V. Janaki,
1School of Computer Science and Informatics, Academic Affairs, Sreenidhi Institute of Science and Technology, Ghatkesar, Hyderabad
2Department of CSE, Kakatiya Institute of Technology and Science, Warangal, A.P. India

Abstract: In this study, we have modified the Hill cipher, by including a permutation and circular rotation into the cipher. Here both the permutation and the rotation depend upon the key. From the cryptanalysis and the avalanche effect, discussed in this study, we notice that the strength of the cipher is significant.

Key words: Modular arithmetic inverse, key dependent permutation, key dependent rotation

INTRODUCTION

The study of the block ciphers which depend upon a symmetric key attracted the attention of researchers, in the first part of last century. Hill developed a cipher, by taking a key in the form of a matrix and using the concept of modular arithmetic inverse. However, it was established that the cipher can be broken by the known plaintext attack.

In a recent research, we have examined the cryptology of the Hill cipher and introduced a systematic procedure for the development of the modular arithmetic inverse of a matrix. In this we have shown that the modular arithmetic inverse of a matrix exists only when the matrix is non-singular, and the value of the determinant is relatively prime to N, where N is a positive integer with which modulo operation is carried out. In this we have noticed that the Hill cipher is vulnerable as the matrix multiplication and the modular arithmetic lead to a direct mapping, without any transposition, connecting the plaintext and the ciphertext.

In the present research our objective is to develop a cipher which involves different types of transformations such as permutation and circular rotation in addition to matrix multiplication and modular arithmetic. In this investigation we have made use of iteration in order to enhance the strength of the cipher. This cipher is expected to be a very strong one, and it cannot be broken by any cryptanalytic attack.

DEVELOPMENT OF THE CIPHER

Consider a plaintext P. Let us suppose that it can be represented in the form of a matrix given by

\[ P = [P_{ij}], \quad i = 1 \text{ to } n, \quad j = 1 \text{ to } n \]  

Consider the key matrix K where

\[ K = [K_{ij}], \quad i = 1 \text{ to } n, \quad j = 1 \text{ to } n \]

Let us suppose that each element in the key matrix lies between 0 and 127.

Let

\[ C = [C_{ij}] \quad i = 1 \text{ to } n, \quad j = 1 \text{ to } n \]

be the ciphertext which can be obtained from the plaintext by applying a set of transformations.

In order to facilitate our analysis, let us represent the above matrices in terms of their corresponding vectors p, k, c given by

\[ p = [p_1, p_2, p_3, \ldots, p_n]. \]

\[ k = [k_1, k_2, k_3, \ldots, k_n]. \]

and

\[ c = [c_1, c_2, c_3, \ldots, c_n]. \]

In writing these vectors we have arranged the elements of the corresponding matrices in a row wise manner.

The procedures of encryption and decryption concerned to this cipher are shown in Fig. 1.

This cipher, depends upon the functions: (1) Convert( ), (2) Iconvert( ), (3) Permute( ), (4) Ipermute( ), (5) Lrotate( ), and (6) Rrotate( ).

The function Permute( ) permutes the binary bits of a plaintext vector in accordance with the numbers present in the key vector. This process of permutation can be described as follows:
Convert the components of \( p \), namely, \( p_1, p_2, p_3, \ldots, p_n \) into their corresponding binary bits. Thus we get the binary bits in the form

\[
p_{11}, p_{12}, \ldots, p_{17}, p_{21}, p_{22}, \ldots, p_{27}, \ldots, p_{n1}, p_{n2}, \ldots, p_{n27}
\]

(4)

where \( p_{11}, p_{12}, \ldots, p_{17} \) are the binary bits corresponding to \( p_1 \), \( p_{21}, p_{22}, \ldots, p_{27} \) are the binary bits corresponding to \( p_2 \), and \( p_{n1}, p_{n2}, \ldots, p_{n27} \) are the binary bits corresponding to \( p_n \).

Let the key vector (obtained from the key matrix) be

\[
k_1, k_2, \ldots, k_{n^2}
\]

(5)

As the numbers in the plaintext are lying between 0 and 127 we have only 7 bits for every number.

Here we have \( 7n^2 \) binary bits in the plaintext and \( n^2 \) numbers in the key. Thus we divide the string of \( 7n^2 \) binary bits into 7 substrings wherein each substring contains \( n^2 \) binary bits. Then we interchange the first bit of the first substring with the \( k_{i1} \)th bit of the entire string. Similarly we interchange the second bit of the first substring with the \( k_{i2} \)th bit of the entire string. We continue this process until all the bits in the first substring are over. In the same way, we carry out the process with the second substring and the third substring, and so on. However, if any one of the numbers in \( k_1 \) to \( k_{n^2} \) exceeds 7, the length of the string, then the interchange is ruled out, and it will not be done. For a detailed discussion of the permutation, let us refer to the example given in Appendix 1.

The function Convert( ) is developed for converting a vector into a two dimensional matrix. Iconvert( ) performs the reverse process to that of Convert( ).

Now, the process of left circular rotation denoted by Lrotate( ) and the process of right circular rotation denoted by Rrotate( ) are explained below. As it is mentioned above let us represent the numbers \( p_{11}, p_{12}, \ldots, p_{n27} \) (the components of \( p \)) in terms of their binary bits. Let the string of binary bits be written as

\[
p_{11}, p_{12}, \ldots, p_{17}, p_{21}, p_{22}, \ldots, p_{27}, p_{31}, p_{32}, \ldots, p_{n27}
\]

(6)

The key vector \( k \) has the components \( k_1, k_2, \ldots, k_{n^2} \), where each one of these numbers lies between 0 and 127. Let us now obtain a set of numbers \( d_i \), such that

\[
d_i = k_i \mod n_2, \quad i = 1 \text{ to } n_2
\]

(7)

Now we rotate the set of binary bits given by (6) through \( d_i \) positions to the left. The function Rrotate( ) is exactly opposite to that of Lrotate( ).

![Fig. 1: Schematic diagram of the cipher](image)

In Fig. 1 \( r \) denotes the number of iterations and it is taken as 16 in this analysis. \( K^{-1} \) is the modular arithmetic inverse of \( K \).

In what follows we design the algorithms for encryption and decryption of the cipher.

**ALGORITHMS**

**Algorithm for encryption:**

1. Read \( n, k, K, p, r \);
2. Permute(\( p \));
3. Lrotate(\( p \));
4. for \( i = 1 \) to \( r \) do
   - \( P = \text{Convert}(p) \);
   - \( P = KP \mod 128 \);
   - \( p = \text{Iconvert}(P) \);
   - Permute(\( p \));
   - Lrotate(\( p \));
5. \( P = K^{-1} \mod 128 \);
6. \( p = \text{Iconvert}(P) \);
7. Permute(\( p \));
8. \( P = \text{Convert}(p) \);
9. Lrotate(\( p \));
10. Permute(\( p \));

**Algorithm for decryption:**

1. Read \( c, k, K, n, r \);
2. Find \( K^{-1} \);
3. for \( i = 1 \) to \( r \) do
   - \( P = \text{Permuted}(p) \);
   - \( P = \text{Rotate}(p) \);
   - \( p = \text{Convert}(P) \);
   - \( P = K^{-1} \mod 128 \);
   - \( p = \text{Iconvert}(P) \);
   - Permute(\( p \));
   - Lrotate(\( p \));
5. \( P = K \mod 128 \);
6. \( p = \text{Iconvert}(P) \);
7. Permute(\( p \));
8. \( P = \text{Convert}(p) \);
9. Lrotate(\( p \));
10. Permute(\( p \));
Algorithm for decryption:
{
1. Read n,k,K,c,r;
2. Find K^{-1};
3. p=c;
4. Rrotate(p);
5. for i= 1 to r
   { 
6. Rrotate(p);
7. Ipermute(p);
8. P=Convert(p);
9. P= K^{-1} P mod 128;
10. p=Iconvert(P);
   }
12. Rrotate(p);
13. Ipermute(p);
14. write p;
}

Algorithm for modular arithmetic inverse: // A is an nxn matrix. N is a positive integer with which modular arithmetic inverse is carried out. Here N = 128.
{
1. Find the determinant of A. Let it be denoted by \( \Delta \), where \( \Delta \neq 0 \).
2. Find the inverse of A. The inverse is given by \( [A_{ji}] / \Delta \).
3. for i = 1 to N
   { 
    if ( (i\( \Delta \)) mod N = 1 ) d = i;
    //\( \Delta \) is relatively prime to N.
    break;
   }
B=( d[A_{ji}] ) mod N
   // B is the modular arithmetic inverse of A
}

ILLUSTRATION OF THE CIPHER

Consider the plaintext given below: In order to do a good business in any other country, we must have cordial relationship with all other localities of the country. We must supply only those commodities for which very good demand is there.

Let us focus our attention on the first 16 characters given by
In order to do a.
This plaintext can be written in the form
p = [73 110 32 111 114 100 101 114 32 116 111 32 100 111 32 97] (8)

Let us choose the key vector in the form
k = [18 4 7 3 4 6 5 42 40 31 9 22 35 17 23 71] (9)

On using (4.1), (4.2) and the encryption algorithm 3.1, we get the ciphertext c in the form
c = [114 114 102 123 67 61 85 34
93 106 11 56 122 92 71 9] (10)

The inverse of the key matrix K can be obtained as
\[
\begin{bmatrix}
75 & 85 & 127 & 37 \\
120 & 115 & 69 & 84 \\
110 & 101 & 22 & 64 \\
115 & 125 & 76 & 114
\end{bmatrix}
\] (11)

On using the decryption algorithm given in 3.2, we get back the plaintext in the form p = [73 110 32 111 114 100 101 114 32 116 111 32 100 111 32 97] This is same as (8).

CRYPTANALYSIS

Let us now consider the cryptanalysis of the cipher. In this the length of the key is \( 7n^2 \) binary bits. Thus the size of the key space is \( 2^{7n^2} \). Hence the cipher cannot be broken by ciphertext only attack, when \( n^2 = 4 \).

In the case of the known plaintext attack, though we know as many pairs of plaintext and ciphertext as we like, here it is not possible to form a direct equation connecting the plaintext and the ciphertext matrices as there are several transformations, namely, permutation and rotation involved in the process of encryption. Thus the cipher cannot be broken by the known plaintext attack also. Further it may be mentioned that no special choice of the plaintext/ciphertext will enable the cryptanalyst to break the cipher.

In the light of the above discussion we find that this cipher is a very strong one and it cannot be broken by any cryptanalytic attack.

AVALANCHE EFFECT

Let us consider the avalanche effect. On changing the first character ‘I’ in the plaintext (8) to ‘H’, the plaintext in its binary form changes by one bit. On
using the encryption algorithm, on the modified plaintext, we get the ciphertext in the form,

\[
\begin{align*}
000000000101100111101100100 & \quad (12) \\
1101110111010000001111001110 & \\
00010111110100101001001111001100 & \\
\end{align*}
\]

On converting (10) into its binary form, we get

\[
\begin{align*}
1110010111001101100111101 & \\
1000011011110110101010110011 & \quad (13) \\
1011011110100001011011000 & \\
1111010111001100011001001 & \\
\end{align*}
\]

On comparing (12) and (13) we find that the two ciphertexts differ in 52 bits.

Now let us change the key by one bit. To this end, we replace \( k_2 = 4 \) by \( k_2 = 5 \). Then on using the plaintext (8), the modified key, and the algorithm, we get the ciphertext in the form

\[
\begin{align*}
01111000010100000010111011 & \\
00111011111101001110110011 & \quad (14) \\
100111010110001110100010 & \\
000100111110001100111001 & \\
\end{align*}
\]

On comparing the ciphertexts (12) and (14) we notice that these two ciphertexts differ in 55 bits.

From the above discussion, it is clearly seen that the cipher exhibits a strong avalanche effect.

**COMPUTATIONS AND CONCLUSIONS**

In this paper we have developed a block cipher by introducing a key dependent permutation and a key dependent circular rotation. The algorithms developed in this analysis are implemented in C language.

The ciphertext corresponding to the entire plaintext is obtained in hexadecimal notation as follows.

72 72 66 7B 43 3D 55 22 5D 6A 0B 38 7A 5C 47 09 3A 60 77 3A 7B 22 52 7A 56 OC 3F 3B 50 71 53 23 15 4A 21 12 7C75 1A 53 33 1D 13 0C 54 3B OE 20 0D 75 65 34 2C 09 7B 3E 17 30 76 3F 59 23 45 65 3E 61 75 4F 4F 51 06 00 38 28 45 6B 18 15 74 1E 06 52 57 03 7E 31 52 7F 16 70 6C 3A 6D 67 61 6B 66 20 05 2E 7A 70 75 08 3D 15 45 07 6C 21 72 15 1C 09 71 05 13 4E 19 50 2F 04 71 6C 5E 30 5D 4A 52 41 62 69 62 5E 61 6D 48 4A 48 4A 2E 0C 7E 55 35 OD 06 3C 4E 38 7B 33 4A 69 48 36 0B 32 26 5E 2F 12 6E 39 0A 5D 0F 30 2C 31 7A 7A 1D 03 40 63 14 49 74 68 54 31 03 75 4F 07 2D 24 22 62 2A 27 17 30 30 13

From the above analysis we conclude that the strength of the cipher increases enormously as the permutation and the rotation are key dependent, and they include a lot of confusion into the structure of the plaintext at various stages of the iteration.

**REFERENCES**

1. William Stallings, Cryptography and Network Security, 3rd Edn., Pearson Education.
2. Sastry, V.U.K. and V. Janaki. On the Modular Arithmetic Inverse in the Cryptology of Hill cipher, Proceedings of North American Technology and Business Conference, September 2005, Canada.