A Block Cipher Basing Upon a Revisit to the Feistel Approach and the Modular Arithmetic Inverse of a Key Matrix

S. Udaya Kumar V. U. K. Sastry A. Vinaya babu

Abstract

In this paper, we have developed a block cipher by using the modular arithmetic inverse of a key matrix. The key matrix is generated by taking the key in the form of twenty-eight numbers, wherein each number is represented in terms of seven binary bits. The encryption is carried out by using the key matrix containing binary bits. The decryption is performed by using the modular arithmetic inverse of the matrix.

I. INTRODUCTION

Majority of the block ciphers found in the literature are based upon Feistel Cipher [1]. The basic elements of this sort of cipher are diffusion and confusion, and they are achieved by mixing and permuting the elements of a plaintext that is to be encrypted. Initially, Feistel [2, 3] proposed that ciphertext can be obtained by operating with a matrix on a given plaintext. However, immediately he came to the conclusion that this type of cipher can be broken as this is essentially a variant of Hill cipher, which can be readily broken by known plaintext attack. Subsequently, he introduced a network, known as classical Feistel network, which involves a round function, wherein the number of rounds is sixteen. Then he has developed a cipher.

Manuscript was received on July 28, 2006.

S. Udaya Kumar is with SreeNidhi Institute of Science & Technology, Ghatkesar, Hyderabad - 501301, India. (Tel. ++91-9395533303, email: uksusarla@rediffmail.com)

V. U. K. Sastry is with SreeNidhi Institute of Science & Technology, Ghatkesar, Hyderabad -501301, India (Tel. ++91-9395533303, email: vuksastry@rediffmail.com).

A. Vinaya babu is with JNT University, Hyderabad, India. (email: dravinayababu@yahoo.com)

In a recent paper, Sastry and Janaki [4] have developed a systematic procedure for obtaining the modular arithmetic inverse of a matrix, and have shown that the Hill cipher can be modified appropriately so that no cryptanalytic attack is possible.

In the present paper our interest is to develop a block cipher, which makes use of a matrix in the process of encryption and the modular arithmetic inverse of the matrix in the process of decryption. In this analysis, we use a matrix consisting of binary elements and make use of the arithmetic with modulo-2. In this, we have developed the cipher in stages: firstly for a block of 14 binary bits and then for 28 bits, and ultimately, for 56 bits. We have also discussed the cryptanalysis, which indicates very clearly that no cryptanalytic attack can break the cipher in anyway.

II. DEVELOPMENT OF THE CIPHER

Consider a plaintext vector ‘P’, having ‘n’ components. Let $P = (p_1, p_2, \ldots, p_n)^T$, in which p_i, i = 1 to n are either ‘0’ or ‘1’, and the superscript T denotes the transpose of the vector. Let $K = [K_{ij}]$, i = 1 to n and j = 1 to n be an n x n matrix in which all K_{ij} are binary elements i.e. either ‘0’ or ‘1’. Let $C = (c_1, c_2, \ldots, c_n)^T$ be the corresponding cipher text.

We get the cipher text C by using the relation

$$C = KP \bmod 2. \qquad (2.1)$$

As the cipher given by (2.1) is similar to the Hill cipher, we know that it can be readily broken by known plaintext attack [1]. In order to overcome this

difficulty, we have adopted an iterative method indicated below.

Let us denote P by P^0 and C by C^0. Then (2.1) can be written as

$C^0 = KP^0 \bmod 2.$ (2.2)

Let $P^1 = C^0 \oplus K_1,$ (2.3)

where K_1 is the transpose of the first row of the matrix K, and $\oplus$ is the XOR operation.

Then the ciphertext corresponding to the first iteration can be written as

$C^1 = KP^1 \bmod 2.$ (2.4)

Similarly, we can obtain the cipher text in the second iteration by using the relations:

$P^2 = C^1 \oplus K_2$ (2.5)

and $C^2 = KP^2 \bmod 2,$ (2.6)

where K_2 is the transpose of the second row of the matrix K.

In general, we can write the iteration scheme as

$P^i = C^{i-1} \oplus K_i,$ (2.7)

and $C^i = KP^i \bmod 2$, i = 1 to n. (2.8)

Similarly, iterations are carried out by using the n columns and the two diagonals of the matrix K in the place of the rows in (2.7). Thus we get the ciphertext in its final form. Now, on performing the reverse operations, we carryout decryption and obtain the plaintext.

In what follows, we design algorithms for encryption and decryption, and also mention a procedure for obtaining the modular arithmetic inverse, K^{-1}, of a given matrix K.

III. ALGORITHMS FOR ENCRYPTION AND DECRYPTION AND PROCEDURE FOR K^{-1}

Algorithm for Encryption

{

1. Read P^0, n, K
2. $C^0 = KP^0 \bmod 2$
3. for i =1 to n do
   {
   $P^i = C^{i-1} \oplus K_i$ // K_i is the transpose of the i^{th} row of K
   $C^i = KP^i \bmod 2$
   }
4. $D^0 = C^n$
5. for i =1 to n do
   {
   $P^i = D^{i-1} \oplus L_i$ // L_i is the i^{th} column of K
   $D^i = KP^i \bmod 2$
   }
6. $E^0 = D^n$
7. for i =1 to 2 do
   {
   $P^i = E^{i-1} \oplus M_i$ //M_1 is the diagonal of K which goes from left to right, and M_2 is the diagonal of K that goes from right to left.
   $E^i = KP^i \bmod 2$
   }
8. $C = E^2$

}

Algorithm for decryption

{

1. Read C, n, K
2. Find K^{-1} // K^{-1} is the modular arithmetic inverse of K.
3. $E^2 = C$
4. for i = 2 to 1 do
   {
   $P^i = K^{-1} E^i \bmod 2$
   $E^{i-1} = P^i \oplus M_i$
   }
5. $D^n = E^0$
6. for i = n to 1 do
   {
   $P^i = K^{-1} D^i \bmod 2$
   $D^{i-1} = P^i \oplus L_i$
   }

7. $C^n = D^0$

8. for i = n to 1 do

{

$P^i = K^{-1}C^i \bmod 2$

$C^{i-1} = P^i \oplus Ki$

}

9. $P^0 = K^{-1}C^0 \bmod 2$

}

Modular Arithmetic Inverse of a Matrix

Let A be an n x n matrix, and B be its modular arithmetic inverse with mod N. Then we have

$AB \bmod N = I,$ (3.1)

where N is any positive integer.

In view of (3.1), we can write

$AB = I + NE,$ (3.2)

where E is an n x n matrix containing the quotients obtained on dividing the elements of AB by N.

Let us suppose that A is a non-singular matrix, and its inverse denoted by A^{-1} is obtained by using Gauss-Jordan elimination method with pivoting [5].

Operating on both the sides of (3.2) with A^{-1} we get

$B = A^{-1} + NA^{-1}E$ (3.3)

We know that $A^{-1} = \frac{[A_{ji}]}{\Delta}$, i =1 to n, j = 1 to n, (3.4)

where A_{ij} are the cofactors of a_{ij}, which are elements of A, and Δ is the determinant of A.

Now (3.3) can be written as

$\Delta B = [A_{ji}] + N [A_{ji}] E.$ (3.5)

Let D be the multiplicative inverse of Δ with respect to N.

Then we have $\Delta D \bmod N = 1.$ (3.6)

This multiplicative inverse D exists only when Δ and N are relatively prime to each other.

On multiplying (3.5) with D, we get

$D \Delta B = D [A_{ji}] + ND [A_{ji}] E.$ (3.7)

From (3.7) we have

$(D\Delta B) \bmod N = D [A_{ji}] \bmod N.$ (3.8)

Thus we have

$B = [DA_{ji}] \bmod N.$ (3.9)

Here it is to be noted that the modular arithmetic inverse of a matrix A exists only when

i) A is non-singular, and

ii) Δ is relatively prime to N.

In the present analysis, we take N = 2, and obtain the modular arithmetic inverse of K such that

$KK^{-1} \bmod 2 = K^{-1}K \bmod 2 = I.$ (3.10)

IV. Illustration of the cipher

Let us consider the plaintext “The Sun rises in the East”. Firstly let us focus our attention on the first two characters of the plaintext given by Th. By using the ASCII code, these two characters can be represented as a pair of numbers given by 84 and 104 respectively.

Let us suppose that the key comprises 28 numbers given by

{65, 71, 95, 121, 48, 31, 99, 81, 42, 19, 23, 41, 37, 19, 17, 67, 87, 105, 119, 13, 27, 31, 118, 117, 4, 110, 99, 35}, (4.1)

wherein each number is less than or equal to 127. We arrange all these numbers in the form of a 14 x 2 matrix such that each row contains only two adjacent numbers of the key. The first row contains 65 and 71. The second row contains 95 and 121, and so on. These numbers can be expressed in their binary form. Thus we have the key matrix K given by

$$K = \begin{bmatrix} 1&0&0&0&0&0&1&1&0&0&0&1&1&1 \\ 1&0&1&1&1&1&1&1&1&1&1&0&0&1 \\ 0&1&1&0&0&0&0&0&0&1&1&1&1&1 \\ 1&1&0&0&0&1&1&1&0&1&0&0&1&1 \\ 0&1&0&1&0&1&0&0&0&1&0&0&0&1 \\ 0&0&1&0&1&1&1&0&1&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&1&0&0&1&1 \\ 0&0&1&0&0&0&1&1&0&0&0&0&1&1 \\ 1&0&1&0&1&1&1&1&1&0&1&0&0&1 \\ 1&1&1&0&1&1&1&0&0&0&1&1&0&1 \\ 0&0&1&1&0&1&1&0&0&1&1&1&1&1 \\ 1&1&1&0&1&1&0&1&1&1&0&1&0&1 \\ 0&0&0&0&1&0&0&1&1&0&1&1&1&0 \\ 1&1&0&0&0&1&1&0&1&0&0&0&1&1 \end{bmatrix} .$$ (4.2)

This is a 14 x 14 matrix consisting of binary bits 0 and 1. The plaintext – Th, represented by the

decimal numbers 84 and 104, is also converted into its binary form. Thus we have the vector

$$[1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0]^T \qquad (4.3)$$

On adopting the procedure given in the encryption algorithm, given in section 3.1, we perform 30 rounds (by including 14 rows, 14 columns, and 2 diagonals appropriately) and get the cipher text given by

$$[1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 1]^T \qquad (4.4)$$

Similarly, the ciphertext corresponding to the plaintext "The sun rises in the East" is obtained as

10100010100011110101101100001101111110011
01010101100110001011001110010011111001111
01100110111111100110100001110100111111111
00001111010111000110011000000111111. (4.5)

In order to perform decryption, the receiver who gets the key from the sender finds the key matrix, and obtains its modular arithmetic inverse (K^{-1}) by employing the relations

$KK^{-1} \bmod 2 = K^{-1}K \bmod 2 = I$..

The value of the determinant of K, denoted by Δ is found to be -85. As this is an odd number, it is relatively prime to 2. Hence, the modular arithmetic inverse (K^{-1}) exists, and is given by

$$K^{-1} = \begin{bmatrix} 0&0&1&0&0&1&0&1&0&0&0&1&0&0 \\ 1&1&1&1&0&0&1&1&1&0&1&1&0&1 \\ 0&0&1&0&0&0&0&1&1&1&0&1&1&1 \\ 0&1&1&0&1&0&0&1&1&1&1&0&1&1 \\ 1&1&0&1&1&0&0&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&0&1&1&0&0&0&1 \\ 0&0&1&1&1&0&1&0&1&0&1&1&1&1 \\ 0&0&1&0&0&0&0&1&1&0&0&1&0&0 \\ 0&1&0&1&0&1&1&1&0&0&1&0&1&0 \\ 0&0&1&0&1&0&0&1&0&1&1&0&1&1 \\ 0&1&1&0&0&1&0&1&0&0&1&1&1&0 \\ 1&0&0&0&0&1&0&1&1&1&0&0&1&1 \\ 0&1&0&0&1&1&1&1&1&1&0&1&0&0 \\ 0&1&1&1&0&1&0&0&0&0&1&0&0&0 \end{bmatrix} . \qquad (4.3)$$

We can readily verify that $K^{-1}\ K \bmod 2 = I$.

On using the decryption algorithm given in section 3.2, we obtain the plaintext corresponding to the ciphertext given by (4.5).

V. Cryptanalysis

Let us carryout cryptanalysis in the above case. As the key matrix is of size 14 x 14, the size of the key space is $2^{196} \approx (2^{10})^{20} \approx 10^{60}$. Thus one cannot break the cipher by applying brute force attack.

Now let us consider the case of the known plaintext attack. Here, we have as many plaintext - ciphertext pairs as we want. Thus we have the known plaintext vector P^0 and the ciphertext vector C obtained after 30 rounds by adopting the encryption algorithm. From the algorithm, we have

$C^0 = KP^0 \bmod 2$

$C^1 = K (C^0 \oplus K_1) \bmod 2$

$= K (KP^0 \bmod 2 \oplus K_1) \bmod 2.$

$C^2 = K ((K(KP^0 \bmod 2 \oplus K_1) \bmod 2) \oplus K_2) \bmod 2.$

This is the relation between the plaintext P^0 and the ciphertext C^2 after two rounds. After 30 rounds, the final ciphertext C will be connected to P^0 by means of a relation, which contains K and mod 2, each occurring 31 times, and the $\oplus$ operation involves the columns, the rows, and the diagonals of the key matrix, wherein the elements of the matrix are unknown. As this relation is obviously a highly non-linear one, the cipher cannot be broken by this approach too.

It is worth noticing that the non-linearity induced by the repetitive process, involving the key matrix, its elements, and the modulo-2 operation occurring several times, does not allow any cryptanalytic attack to break the cipher.

Here, it is to be noted that we have taken each block of the plaintext such that it consists of only two characters. When the process of encryption is repeated for a long plaintext, the statistical properties of the ciphertext may repeat and reveal the characters of the plaintext as a whole. This drawback can be overcome by taking blocks with more number of characters each time.

VI. Modified cipher for larger block size

Let us focus our attention on the plaintext, containing four characters, given by

"The $\not b$ ". (6.1)

Let us represent the characters in the form of numbers given by their ASCII codes and convert the same into 28 binary bits. Then we have the plaintext in the form

$$[1010100111010001100101010000 0]^T . \quad (6.2)$$

Let K be a key matrix of the size 28 x 28. It can be written in the form

$$K = \begin{bmatrix} Q & R \\ Y & Z \end{bmatrix}, \quad (6.3)$$

where Q is a 14x14 matrix, which is formed by taking the key in the form

[65, 71, 95, 121, 48, 31, 99, 81, 122, 119, 23, 41, 37, 11, 114, 67, 87, 105, 117, 115, 127, 31, 118, 116, 124, 113, 98, 35]. (6.4)

This key consists of 28 numbers. It may be noted here that $Z = Q^T$. R is obtained by interchanging the first and last rows of Q, the second and last but one rows of Q, and so on. The matrix Y is obtained by making the last column of Q the first row of Y, the last but one column of Q as the second row of Y, and so on.

Thus we have

$$K = \begin{bmatrix}
1&0&0&0&0&0&1&1&0&0&0&1&1&1&1&1&0&0&0&1&0&0&1&0&0&0&0&1&1\\
1&0&1&1&1&1&1&1&1&1&1&0&0&1&1&1&1&1&1&0&0&1&1&1&0&0&0&1\\
0&1&1&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&0&1&1&0&1&1&1&0&1&0&0\\
1&1&0&0&0&1&1&1&0&1&0&0&0&1&1&1&1&1&1&1&1&1&0&0&1&1&1&1&1\\
1&1&1&1&0&1&0&1&1&1&0&1&1&1&1&1&1&1&1&0&1&0&1&1&1&1&0&0&1\\
0&0&1&0&1&1&1&0&1&0&1&0&0&1&1&0&1&0&1&1&1&1&1&0&1&0&0&0&1\\
0&1&0&0&1&0&1&0&0&0&1&0&1&1&1&1&1&1&0&0&1&0&1&0&0&0&0&1&1\\
1&1&1&0&0&1&0&1&0&0&0&0&1&1&0&1&0&0&1&0&1&0&0&0&1&0&1&1\\
1&0&1&0&1&1&1&1&1&1&0&1&0&0&1&0&0&1&0&1&1&1&0&1&0&1&0&0&1\\
1&1&1&0&1&0&1&1&1&0&0&1&1&1&1&1&1&1&1&0&1&0&1&1&1&0&1&1&1\\
1&1&1&1&1&1&1&1&0&0&1&1&1&1&1&1&1&1&0&0&0&1&1&1&0&1&0&0&1\\
1&1&1&0&1&1&0&1&1&1&0&1&0&0&0&1&1&0&0&0&0&0&0&1&1&1&1&1\\
1&1&1&1&1&0&0&1&1&1&0&0&0&1&1&0&1&1&1&1&1&1&1&1&1&0&0&1\\
1&1&0&0&0&1&0&0&1&0&0&0&1&1&1&0&0&0&0&0&1&1&0&0&0&1&1&1\\
1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&0&1&1&0&0&1&1&1&1&1&1&1\\
1&0&1&0&1&0&1&1&0&1&1&0&0&1&0&0&1&1&1&0&1&1&0&1&1&1&1&1&1\\
1&0&1&0&1&0&0&0&0&0&1&1&0&0&0&1&1&0&1&1&0&1&1&1&1&1&1&1&0\\
0&1&1&0&0&1&1&0&1&0&1&0&0&0&0&1&0&0&1&0&0&0&0&0&1&0&1&0\\
0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&1&0&0&0&1&1&0&1&1&1&1&1&0\\
0&1&0&0&1&1&0&0&1&1&0&1&1&1&0&1&0&1&1&1&0&1&1&0&1&1&0&1\\
1&1&0&1&1&0&0&1&1&1&0&1&1&0&1&1&0&1&0&1&1&0&1&1&1&0&0&0\\
1&1&0&1&0&1&1&0&1&1&1&0&0&0&1&1&0&1&1&0&0&1&1&1&0&1&1&0\\
0&1&0&1&1&1&0&1&1&0&1&1&0&1&0&1&0&0&1&1&0&0&1&1&0&1&1&1\\
0&1&0&0&0&1&1&0&1&1&1&1&1&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0\\
0&1&0&0&1&0&0&0&0&0&1&0&1&0&0&1&1&0&0&1&1&0&1&0&1&0&0&0\\
0&1&1&0&1&1&0&1&1&1&1&1&1&0&1&0&1&0&1&0&0&0&0&0&1&1&0&0\\
0&0&1&1&1&0&1&1&0&1&1&1&1&1&1&1&0&1&0&1&0&1&1&0&1&1&0&0&1\\
1&1&0&1&1&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1
\end{bmatrix} . \quad (6.5)$$

Now, on using the key matrix K given by (6.5), and the plaintext given in (6.2), we apply the encryption algorithm presented in section 4.1, and obtain the ciphertext given by

$$[0101101100 1110011100 11111100]^T . \quad (6.6)$$

As n is 28, here we have to perform 58 rounds (28 with rows, 28 with columns, and 02 with diagonals). As the determinant of K, denoted by Δ = 71576.967967 $\cong$ 71577, it is relatively prime to 2. Thus the modular arithmetic inverse of K can be obtained by applying the procedure given in section 3.3. Hence,

$$K^{-1}=\begin{bmatrix}
0&0&1&0&0&0&1&0&0&0&1&1&1&1&1&0&1&1&0&0&0&1&0&1&1&1&0&0\\
1&0&1&1&0&1&1&1&0&1&1&1&0&0&1&1&0&1&1&1&0&0&1&1&0&0&0&0\\
0&1&0&1&0&1&0&0&1&0&1&0&0&1&1&1&1&1&1&1&1&0&1&1&0&1&1&1\\
1&0&0&1&0&1&0&1&1&0&1&0&1&0&0&1&1&0&0&1&0&0&1&0&1&1&0&0\\
1&0&0&0&1&1&0&1&1&0&0&0&0&1&0&0&1&0&1&1&0&1&1&1&0&1&1&1\\
1&0&1&0&0&0&1&1&0&1&1&1&1&1&1&0&0&0&1&0&0&1&1&1&1&1&1&1\\
1&1&0&0&0&0&1&0&0&1&0&0&0&1&0&0&1&1&0&1&1&0&0&0&0&0&0&0\\
0&1&0&0&1&1&1&0&0&1&1&0&0&1&1&1&0&1&1&1&0&1&1&0&1&0&1&0\\
0&1&1&0&1&1&1&0&1&0&1&1&1&0&0&1&0&1&0&1&0&0&1&1&1&0&0&1\\
0&1&0&0&1&1&1&1&1&0&0&0&1&0&1&0&0&1&0&1&1&0&0&0&1&0&0&1\\
0&1&1&1&0&1&0&1&0&1&0&1&0&1&0&1&1&0&0&0&0&1&0&0&1&1&0&1\\
1&1&0&1&1&0&0&1&0&1&0&0&0&0&1&1&1&0&0&0&1&0&0&0&1&1&0&0\\
0&1&0&0&1&1&0&1&1&1&0&0&0&1&1&1&1&0&0&0&1&1&0&1&0&1&0&1\\
1&1&1&0&1&1&1&1&1&1&1&0&0&1&0&1&0&0&1&1&0&0&0&1&0&1&1&1\\
1&1&1&1&0&0&0&1&0&0&0&1&0&0&0&0&1&1&1&0&1&0&0&0&1&1&0&1\\
0&0&1&1&1&0&1&1&1&0&1&1&0&1&0&0&0&0&1&1&0&0&1&1&1&0&1&1\\
1&0&0&1&0&1&0&0&1&0&1&0&1&0&1&1&1&0&1&1&0&1&1&1&1&1&1&1\\
0&1&0&1&0&1&1&0&1&0&1&0&0&1&0&0&1&1&0&1&0&0&1&0&0&1&1&0\\
1&0&0&0&0&1&1&0&1&1&0&0&0&1&1&1&1&0&1&1&1&0&1&1&0&1&0&0\\
1&1&1&1&1&0&1&1&0&0&0&1&0&1&1&1&1&1&1&1&1&0&0&1&0&0&0&1\\
1&0&0&0&1&0&0&1&0&0&0&0&1&1&0&0&0&0&0&0&0&1&1&0&1&1&0&0\\
1&0&0&1&1&0&0&1&1&1&0&0&1&0&0&1&0&1&0&1&1&0&1&1&1&0&1&1\\
0&1&1&1&0&1&0&1&1&1&0&1&1&0&1&0&0&1&1&1&0&0&1&0&1&0&1&0\\
0&1&0&0&0&1&1&1&1&1&0&0&1&0&1&0&0&1&0&0&0&1&1&0&1&0&0&1\\
1&0&1&0&1&0&1&0&1&1&1&0&1&1&0&0&1&0&0&0&0&1&1&0&1&0&1&0\\
0&0&0&0&1&0&1&0&0&1&1&0&1&1&0&0&1&1&0&0&0&1&0&0&0&1&1&1\\
1&0&0&0&1&1&1&0&1&1&0&0&1&0&1&0&1&0&1&0&1&1&0&0&0&1&1&1\\
1&0&0&1&1&1&1&1&1&1&1&0&1&1&1&1&1&1&0&1&0&0&0&1&1&0&0&1&0
\end{bmatrix}. \quad (6.6)$$

On using the K^{-1} given in (6.7), and the ciphertext given in (6.6), and applying the decryption algorithm, we obtain the original plaintext. In this case, the plaintext is of length 28 binary bits. This is also a short one. Thus we further generalize this procedure by considering a plaintext of length 56 bits and obtain the corresponding ciphertext as shown in Fig. 1.

In the process of encryption, we use the procedure for encryption (PE) described earlier in this section (Section 6). In the process of interlacing (IL), we mix the string W on the left side with the string W on the right side such that the first bit of the right side W is next to the first bit of the left side W, and the second bit of the right side W is next to the second bit of the left side W, and so on. And this process is continued till we exhaust both the Ws. On the whole, the process of encryption with interlacing is carried out for sixteen rounds.

The process of decryption, which is a reverse process of encryption, can be schematically represented as shown in Fig. 2. Here we adopt the decryption procedure, which is already descried in this section. In the process of decomposition, we keep the first bit of 2W in the left side W as the first bit, and the second bit of 2W as the first bit of the right side W, and this process is continued till all the bits of 2W are exhausted. The process of decryption, which involves decomposition, is also carried out for sixteen rounds.

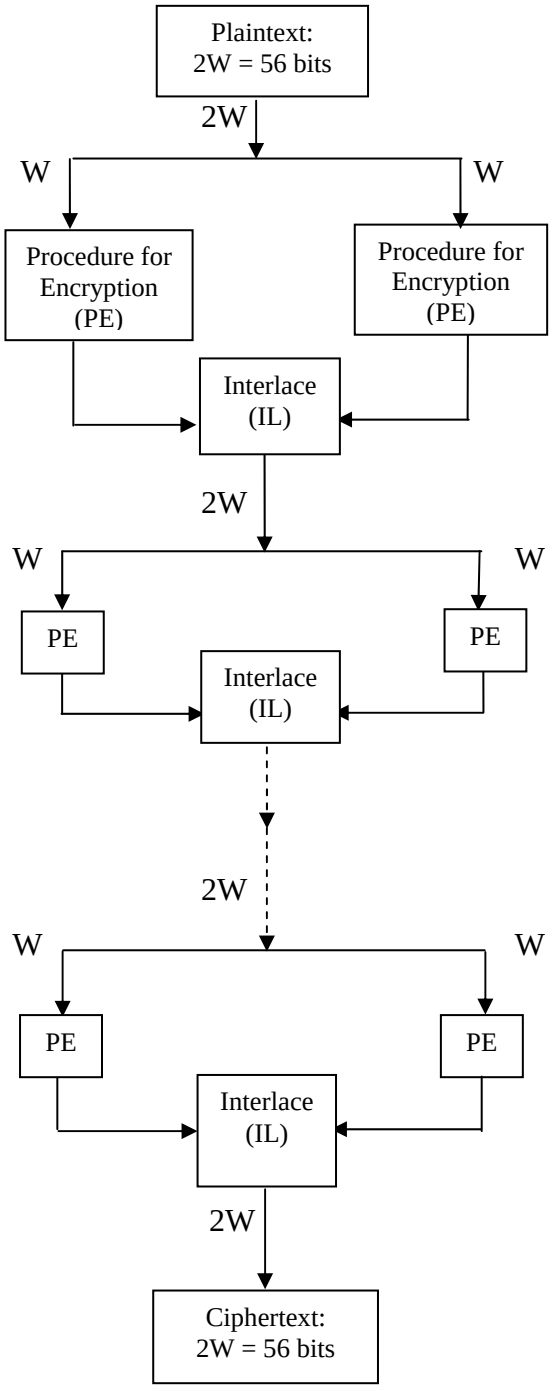

Fig. 1. Process of Encryption for 56 bits Plaintext

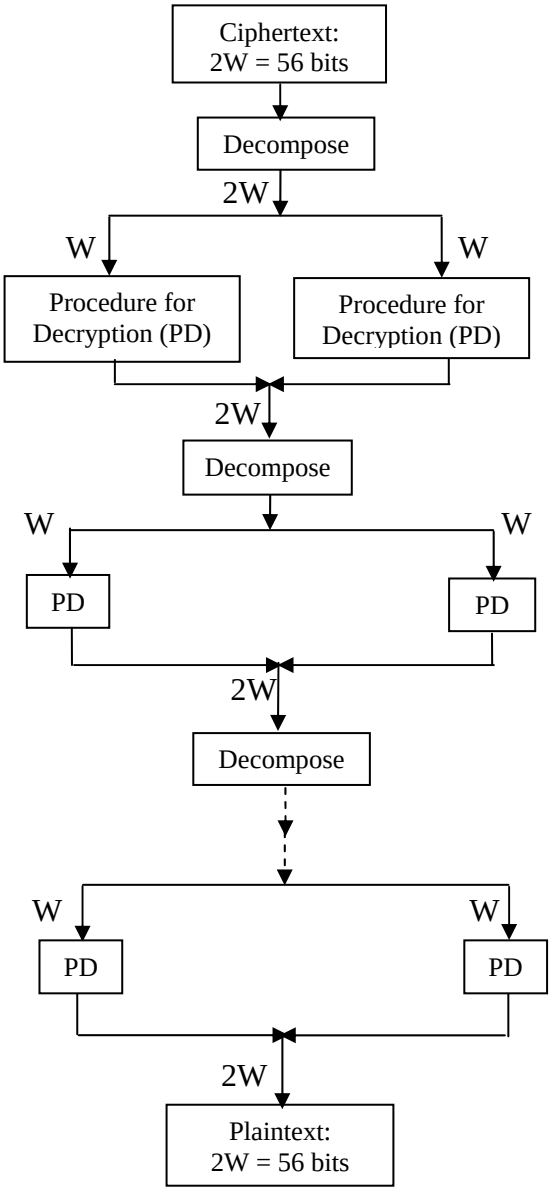

Fig. 2. Process of Decryption for 56 bits Ciphertext

VII. Avalanche Effect

In order to check up the robustness of the algorithm, firstly we have focused our attention on the plaintext "The␣Sun␣" and obtained the corresponding ciphertext using the key K given by (6.5). The ciphertext is

11010010110001000011011101011111001010011001100111111111. (7.1)

Then on changing the plaintext in only one bit position, i.e. by taking the plaintext as "She␣Sun␣", we have found the corresponding ciphertext as

01001011011001010100000110111111110101011000011110111001. (7.2)

On comparing (7.1) and (7.2), we find that the ciphertexts under consideration differ in large number of bits (28 bits out of 56 bits). This clearly indicates that the algorithm exhibits a very strong avalanche effect. Now let us consider the case wherein the key is changed in one bit position, i.e. by replacing 48 by 50. Then the key assumes the form

[65, 71, 95, 121, 50, 31, 99, 81, 122, 119, 23, 41, 37, 11, 114, 67, 87, 105, 117, 115, 127, 31, 118, 116, 124, 113, 98, 35]. (7.3)

In this case the ciphertetxt for the plaintext "The␣Sun␣" is given by

11110111001111001011000110100101000110010111101101100101. (7.4)

From (7.1) and (7.4), it is readily noticed that the algorithm once again shows a very strong avalanche effect.

VIII. Computational Experiments and Conclusions

In this paper, by using the basic ideas of Feistel's approach and the modular arithmetic inverse of a matrix, we have developed a block cipher for block size of 56 bits. The algorithms developed for encryption and decryption are implemented in C language. Computational experiments are carried out with plaintexts containing ASCII characters.

The results obtained in this analysis indicate that the encryption and the decryption are fully in agreement with each other. The ciphertext obtained for the plaintext "The Sun rises in the East" is shown in fig. 3.

Plaintext: The Sun rises in the East

Ciphertext:

1101001011000100001101110101101010011010001100101010000011111000110000110111001101101110010110100111100111100101101101101110100010110110000101000001110100110100011001011101011101110001000000000111110100010000000100000100000.

Fig. 3. Plaintext and Ciphertext.

A plaintext in the form of a program and its corresponding ciphertext are given in Appendix A.

In the development of the cipher, the secret key contains only 28 numbers (see (4.1) and (6.4)). Here, it is to be noted that whatever may be the size of the key matrix K (see (4.2) and (6.6)) and the size of the plaintext, the size of the secret key remains the same, and of course this is to be sent to the receiver by the sender in a secured manner.

From the above analysis, it is worth noticing that this cipher is a very strong cipher as it cannot be broken by any cryptanalytic attack.

IX. Acknowledgements

The authors are highly grateful to the Management of SreeNidhi Institute of Science & Technology, for their kind encouragement and providing the necessary facilities to carryout this research work.

References

[1] William Stallings,"Cryptography and Network Security: Principles and Practices", Third edition, Chapter 3, pp.63.

[2] Feistel, H. "Cryptography and Computer Privacy", Scientific American, May 1973.

[3] Feistel, H., Notz. W., and Smith, J. "Some Cryptographic Techniques for Machine-to-Machine Data Communications", Proceedings of the IEEE, November 1975.

[4] V.U.K.Sastry, V.Janaki, "On the Modular Arithmetic Inverse in the Cryptology of Hill Cipher", Proceedings of North American Technology and Business Conference, September 2005.

[5] William H. Press, Brian P. Flannery, Saul A. Teukolsky, William T. Vetterling, Numerical Recipes in C: The Art of Scientific Computing, Second Edition, 1992, Cambridge University Press, pp. 36 – 39.

Appendix A

Plaintext:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<ctype.h>
main()
{
    FILE *fp1,*fp2;
    char c,d;
    clrscr();
fp1=fopen("ptext.txt","r");
fp2=fopen("dctext.txt","r");
while(((c=getc(fp1))!=EOF)&&(d=getc(fp2))!=EOF)
    {
        if(c!=d)
        {
            printf("both files are not same\n");
            getch();
            exit(1);
        }
    }
printf("Both files are same\n");
fclose(fp1);
fclose(fp2);
getch();
return 1;
}
```

Ciphertext:

011010010111111011111000101010000011100111
001110011010101000100001011110011111101100
100111111110101010110010010111011011001000
100011111011110011111000011000010100011110
110011111111100111101111110001100110010111
110011110100001001011000010011111001111000
000101011010010111111011111000101010000011
100111001110011010101000100001011110011111
101100101100011111111111110001000111010111
011100101110101101001010000111001111010111
001010101101001010010110001101101000010111
100000100000100110010100111001011011001100
100001111010100111110000010101010100101000
101010001100100111011011010000010110000010
011000100110011111101010111001011111011010
101010010001011110011010101010110111111111
010110100101110001110110010000101000001100
111011010101000110110110111010011111010010
000111010110110010000011001101010001110110
011011100010111101010001001001001101001000
101100111000100011011110011000010010110100
011010011010010101001111010010001001000110
010100001001100000110101110001011000011100
110001110001100100011101101111000000100010
111101110000100010001000001000000101100100
101111010111110011001010101110110111000010
001010111101011000010010010001111101000110
001011111010000111111100111010110111011101
110000010011000000000110010000011001010010
010111001101001110011010110100000100011101
100001100001000010101000110100100000101000
100111100111001100111001101111100001111100
010001111000101101011101111100110101100100
000011001000100101110000101101000000110001
101111011011001010011010010001110000111000
110110001000010000100010001111011001001000
011000010100111000010010111001100100000011
001111111011001001111110000111110010010101
001100111001111111110100000010000001001111
100001000110001100011000000110101001101010
011111001110000110011000000110111111111001
110110111011000001000000110101101011000100
010101001101110111100011000110110010010100
111111010110010001110001010010110000100100
001100011010000011000101110010110111110011
011001001110000100010000011000010010011000
100010011000010001110001010100001101110100
001101100111111000101010000111101010010111
111001001000111010001001001010111111010011
100001110000010101111011001101011001100001
100001001010111010001001001010111111010010
100011001110011000000111110101000100011100
001111110110111010001001001010111111010000
001001011011110000101011111010100111101110
001010000001110010110111001001010000111000
001010100001000101101101101101000010010011
000000101101010010111111010000010010110111
100001001011001001010010111011000111100000
011000101111000011110110011101010100111000
110000100001010100011010001010000111011010
010011110010111001010111101001011100110100
001011010000011000110111111100110010101011
101101110001110011101000111010110110011010
011001010000010110101111011010000111100111
001011001011001000010111011111001100010100
110110101101011010001111101010001110100010
011101000011111011110011001100011001100100
000101001000100011001000100011010011010011
01001011110.