Volume 1, No. 2, July-August 2010

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Cryptography of a Binary Image using a Modified Hill Cipher

Dr. V. Umakanta Sastry*
Department of Computer Science and Engineering
SreeNidhi Institute of Science and Technology
Hyderabad – 501 301, Andhra Pradesh, India
vuksastry@rediffmail.com

D. S. R. Murthy
Department of Information Technology
SreeNidhi Institute of Science and Technology
Hyderabad – 501 301, Andhra Pradesh, India
dsrmurthy@sreenidhi.edu.in

Dr. S. Durga Bhavani
School of Information technology
Jawaharlal Nehru Technological University Hyderabad (JNTUH)
Hyderabad – 500 085, Andhra Pradesh, India
sdurga.bhavani@gmail.com

Abstract: In this paper, we have used a modified Hill cipher for encrypting a binary image. Here, we have illustrated the process of encryption by considering a couple of examples. The security of the image is totally achieved, as the encrypted version of the image, does not reveal any feature of the original image.

Keywords: Cryptography, Cipher, Binary image, Encrypted image, Modular arithmetic inverse.

I. INTRODUCTION

In a recent investigation [1, 2], we have modified the Hill cipher by developing an iterative procedure. In [1], we have multiplied the plain text matrix P with a key matrix K on both the sides of P, while in [2] we have used K on one side and K^{-1} on the other side of P as multiplicants. The process is strengthened by using a function called Mix (P), at any stage of the iterative process. It is further supplemented with the XOR operation between the plain text P and the key K. In this, the key is containing 32 decimal numbers, which are in the interval [0 – 255], and the modular arithmetic inverse of the key, represented in the form of a 16 x 16 matrix, is obtained by using mod 2.

In the present paper, our objective is to develop a block cipher, and to use it for the cryptography of a binary image. Here, we have taken a key containing 32 decimal numbers [1, 2], and generated a key matrix of size 32 x 32 in a special manner (discussed later), and applied it in the cryptography of a pair of binary images.

In Section 2, we have developed a procedure for the cryptography of a binary image. In Section 3, we have used a pair of examples and illustrate the process. Finally, in Section 4, we have drawn conclusions from the analysis.

II. DEVELOPMENT OF A PROCEDURE FOR THE CRYPTOGRAPHY OF A BINARY IMAGE

Consider a binary image whose gray level values can be represented in the form of a matrix given by

$\mathbf{P} = [\mathbf{P}_{ij}]$, i = 1 to n, j = 1 to n. (2.1)

Here, each P_{ij} is 0 or 1, where 1 corresponds to black and 0 corresponds to white.

Let us choose a key k. Let it be represented in the form of a matrix given by

$\mathbf{K} = [\mathbf{K}_{ij}]$, i = 1 to n, j = 1 to n, (2.2)

where each K_{ij} is either 0 or 1.

Let $\mathbf{C} = [\mathbf{C}_{ij}]$, i = 1 to n, j = 1 to n (2.3)

be a matrix, obtained on encryption.

The process of encryption and the process of decryption, which are quite suitable, for the problem on hand, are given in Fig. 1.

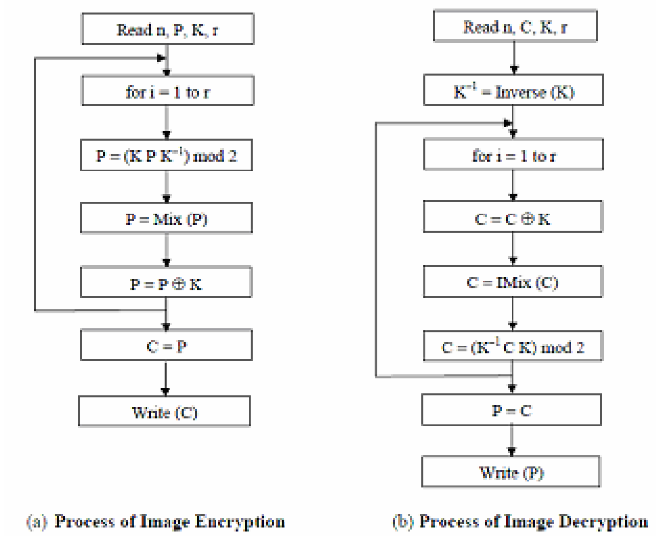

Figure 1: Schematic diagram for the cryptography of a image

Mix ( ) is a function used for mixing thoroughly the binary bits arising in the process of encryption at each stage of iteration. IMix ( ) is a function which represents the reverse process of Mix ( ). For a detailed discussion of these functions, and the algorithms involved in the processes of encryption and decryption, we refer to [1].

III. ILLUSTRATION OF THE CRYPTOGRAPHY OF AN IMAGE

Let us take a key k consisting of 32 decimal numbers. This is given by

© 2010, IJARCS All Rights Reserve

$$k = \left\{ \begin{matrix} 131 & 31 & 18 & 59 & 254 & 126 & 113 & 97 & 127 & 167 & 76 & 116 & 111 & 159 & 245 & 159 \\ 175 & 50 & 236 & 107 & 235 & 74 & 47 & 20 & 190 & 80 & 242 & 139 & 175 & 164 & 187 & 158 \end{matrix} \right\} \quad (3.1)$$

Let us write k in the form of a matrix by placing the first two numbers 131 and 31, in their binary form, in the first row, the next two numbers 18 and 59, in the second row, and so on.

Thus we get a matrix, denoted as Q, in the form

$$Q = \begin{bmatrix} 1&0&0&0&0&0&1&1&0&0&0&1&1&1&1&1 \\ 0&0&0&1&0&0&1&0&0&0&1&1&1&0&1&1 \\ 1&1&1&1&1&1&1&0&0&1&1&1&1&1&1&0 \\ 0&1&1&1&0&0&0&1&0&1&1&0&0&0&0&1 \\ 0&1&1&1&1&1&1&1&1&0&1&1&0&1&1&1 \\ 0&1&0&0&1&1&0&0&0&1&1&1&0&1&0&0 \\ 0&1&1&0&1&1&1&1&1&0&0&1&1&1&1&1 \\ 1&1&1&1&0&1&0&1&1&0&0&1&1&1&1&1 \\ 1&0&1&0&1&1&1&1&0&0&1&1&0&0&1&0 \\ 1&1&1&0&1&0&1&0&0&1&1&0&1&0&1&1 \\ 1&1&1&0&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&0&1&0&1&1&1&1&0&0&0&1&0&1&0&0 \\ 1&0&1&1&1&1&1&0&0&1&0&1&0&0&0&0 \\ 1&1&1&1&0&0&1&0&1&0&0&0&1&0&1&1 \\ 1&0&1&0&1&1&1&1&1&0&1&0&0&1&0&0 \\ 1&0&1&1&1&0&1&1&1&0&0&1&1&1&1&0 \end{bmatrix} \quad (3.2)$$

Now, in order to have a larger key, for convenience, let us take a key K in the form

$$K = \begin{bmatrix} Q & R \\ S & U \end{bmatrix} \quad (3.3)$$

where $U = Q^T$, in which T denotes the transpose, and matrices R and S are obtained from Q and U respectively, by adopting the following procedure.

To obtain R, we interchange the first row of Q with its 16^{th} row. Similarly, we interchange the second row of Q with its 15^{th} row. We continue this process till we exhaust all the remaining rows of Q. Thus we get R. In the same manner, we obtain S from U. Now a circular rotation through 4 rows is applied in the downward direction (i.e., the first row becomes the fifth row and the rest of the rows move in the same direction past 4 rows) on U, and hence we get the new U. Thus, we have

$$K = \begin{bmatrix} 1&0&0&0&0&0&1&1&0&0&0&1&1&1&1&1&1&0&1&1&1&0&1&1&1&0&0&1&1&1&1&0 \\ 0&0&0&1&0&0&1&0&0&0&1&1&1&0&1&1&1&0&1&0&1&1&1&1&1&1&1&0&0&1&0&0 \\ 1&1&1&1&1&1&1&0&0&1&1&1&1&1&1&0&1&1&1&1&0&0&1&0&1&0&0&0&1&0&1&1 \\ 0&1&1&1&0&0&0&1&0&1&1&0&0&0&0&1&1&0&1&1&1&1&1&0&0&1&0&1&0&0&0&0 \\ 0&1&1&1&1&1&1&1&1&0&1&1&0&1&1&1&0&0&1&0&1&1&1&1&0&0&0&1&0&1&0&0 \\ 0&1&0&0&1&1&0&0&0&1&1&1&0&1&0&0&1&1&1&0&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&1&1&1&1&1&0&0&1&1&1&1&1&1&1&1&0&1&0&1&0&0&1&1&0&1&0&1&1 \\ 1&1&1&1&0&1&0&1&1&0&0&1&1&1&1&1&1&0&1&0&1&1&1&1&0&0&1&1&0&0&1&0 \\ 1&0&1&0&1&1&1&1&0&0&1&1&0&0&1&0&1&1&1&1&0&1&0&1&1&0&0&1&1&1&1&1 \\ 1&1&1&0&1&0&1&0&0&1&1&0&1&0&1&1&0&1&1&0&1&1&1&1&1&0&0&1&1&1&1&1 \\ 1&1&1&0&1&0&0&1&0&1&0&0&1&0&1&0&0&1&0&0&1&1&0&0&0&1&1&1&0&1&0&0 \\ 0&0&1&0&1&1&1&1&0&0&0&1&0&1&0&0&0&1&1&1&1&1&1&1&1&0&1&1&0&1&1&1 \\ 1&0&1&1&1&1&1&0&0&1&0&1&0&0&0&0&0&1&1&1&0&0&0&1&0&1&1&0&0&0&0&1 \\ 1&1&1&1&0&0&1&0&1&0&0&0&1&0&1&1&1&1&1&1&1&1&0&0&1&1&1&1&1&1&1&0 \\ 1&0&1&0&1&1&1&1&1&1&1&0&0&1&0&0&0&0&0&1&0&0&1&0&0&0&1&1&1&0&1&1 \\ 1&0&1&1&1&0&1&1&1&0&0&1&1&1&1&0&1&0&0&0&0&0&1&1&0&0&0&1&1&1&1&1 \\ 1&1&0&1&1&0&1&1&0&1&0&0&0&1&0&0&1&1&1&0&0&0&1&1&0&1&1&0&0&1&0&1 \\ 1&1&1&0&1&0&1&1&1&1&1&0&0&1&0&1&1&0&1&0&1&1&1&1&0&0&0&1&0&0&1&1 \\ 1&0&1&0&1&1&1&1&0&0&0&1&0&0&1&1&1&1&1&0&1&0&1&1&1&1&1&0&0&1&0&1 \\ 1&1&1&0&0&0&1&1&0&1&1&0&0&1&0&1&1&1&0&1&1&0&1&1&0&1&0&0&0&1&0&0 \\ 1&1&1&0&1&1&1&1&1&0&0&1&1&0&0&1&1&0&1&0&0&0&0&1&1&1&1&0&1&1&1&1 \\ 0&1&1&1&1&1&0&0&1&1&0&0&0&0&1&0&0&0&1&1&1&1&1&1&0&1&1&0&0&1&0&0 \\ 0&0&1&1&0&1&0&0&0&1&1&0&1&0&1&0&0&0&1&1&1&0&1&1&1&1&1&1&1&1&1&1 \\ 0&0&0&0&1&0&1&1&0&0&0&0&0&1&1&1&0&1&1&1&1&0&0&1&0&0&0&0&1&1&0&1 \\ 1&0&0&1&1&0&1&1&1&0&1&1&0&0&1&1&0&0&1&0&1&1&1&0&1&1&1&1&1&0&1&1 \\ 1&1&1&0&1&0&1&0&1&1&0&1&1&1&1&1&0&0&1&0&1&1&1&1&1&0&0&1&1&0&1&0 \\ 0&0&1&0&1&1&1&1&0&0&1&1&0&1&0&1&1&1&0&1&0&1&0&1&1&0&1&1&1&1&1&1 \\ 0&0&1&0&1&1&1&0&1&1&1&1&1&0&1&1&1&0&0&1&1&0&1&1&1&0&1&1&0&0&1&1 \\ 0&1&1&1&1&0&0&1&0&0&0&0&1&1&0&1&0&0&0&0&1&0&1&1&0&0&0&0&0&1&1&1 \\ 0&0&1&1&1&0&1&1&1&1&1&1&1&1&1&1&0&0&1&1&0&1&0&0&0&1&1&0&1&0&1&0 \\ 0&0&1&1&1&1&1&1&0&1&1&0&0&1&0&0&0&1&1&1&1&1&0&0&1&1&0&0&0&0&1&0 \\ 1&0&1&0&0&0&0&1&1&1&1&0&1&1&1&1&1&1&1&0&1&1&1&1&1&0&0&1&1&0&0&1 \end{bmatrix} \quad (3.4)$$

The afore mentioned operations are performed for

(1) enhancing the size of the original key matrix (16 x 16) to 32 x 32, and

(2) obtaining the modular arithmetic inverse of K, in a trial and error manner.

The modular arithmetic inverse is obtained as

$$K^{-1} = \begin{bmatrix} 0&0&0&0&0&0&1&0&0&0&1&1&0&1&1&1&1&1&1&0&1&1&0&1&1&0&0&1&1&1&1&0 \\ 0&1&0&0&1&0&0&1&0&1&0&0&0&1&0&0&0&0&1&1&0&1&0&1&0&1&0&1&0&0&1&0 \\ 0&1&0&1&0&1&0&0&1&1&0&1&0&0&0&1&1&0&1&0&1&1&1&0&1&1&1&1&1&1&1&0 \\ 0&1&0&0&1&0&1&1&0&0&1&0&0&0&1&1&0&1&1&0&1&0&0&0&1&0&0&1&0&1&1&0 \\ 0&1&0&0&0&0&0&0&1&1&1&0&0&1&1&0&0&0&1&1&1&1&1&0&0&0&0&0&0&0&0&0 \\ 0&1&1&0&0&1&1&1&1&0&0&1&1&0&1&1&1&1&1&0&0&0&0&0&0&0&0&1&0&1&1&1 \\ 1&0&0&0&0&0&0&0&1&0&0&1&1&1&1&1&1&1&1&0&0&0&1&0&1&0&1&0&1&0&1&0 \\ 0&0&0&0&1&1&1&0&0&0&1&1&0&0&1&0&1&0&1&1&0&1&1&0&1&1&0&0&0&1&0&1 \\ 1&0&1&1&0&1&1&1&1&0&0&1&0&1&1&1&1&1&1&1&1&1&0&0&1&0&1&0&1&0&1&1 \\ 0&0&0&1&1&0&1&0&1&1&1&1&1&0&1&1&1&0&1&1&0&0&1&0&1&1&0&0&0&1&0&1 \\ 0&0&0&1&1&0&1&0&0&0&1&0&0&0&1&1&0&1&1&1&0&1&0&0&0&0&1&1&1&0&1&0 \\ 1&0&1&1&0&0&0&0&0&0&0&0&1&1&1&1&1&1&0&1&0&0&0&0&1&1&0&1&1&0&0&1 \\ 1&0&1&1&1&1&1&1&0&0&0&1&1&1&0&0&1&1&0&0&1&0&0&0&0&1&1&0&1&1&1&1 \\ 1&0&0&1&1&0&1&0&1&1&1&1&0&0&0&1&1&1&0&0&1&0&1&1&0&0&1&1&0&1&0&0 \\ 0&0&0&0&1&1&0&0&1&0&0&1&0&0&1&1&0&1&0&0&1&0&1&0&1&1&1&1&1&0&1&1 \\ 1&0&0&0&0&0&1&1&0&1&0&1&0&1&0&0&0&1&1&0&1&1&0&1&1&0&1&0&1&0&1&0 \\ 1&1&1&0&1&1&0&0&0&1&0&0&0&0&0&0&0&1&1&1&0&1&1&1&1&0&0&1&1&0&1&1 \\ 0&0&1&0&0&0&1&0&1&0&0&1&0&0&1&0&1&1&0&0&0&1&0&0&1&0&1&0&1&0&1&0 \\ 1&0&0&0&1&0&1&1&0&0&1&0&1&0&1&0&0&1&0&1&0&1&1&1&1&1&1&1&0&1&1&1 \\ 1&1&0&0&0&1&0&0&1&1&0&1&0&0&1&0&0&1&1&0&0&1&1&0&1&0&0&1&0&0&0&1 \\ 0&1&1&0&0&1&1&1&0&0&0&0&0&0&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1&1&1 \\ 1&1&0&1&1&0&0&1&1&1&1&0&0&1&1&0&0&1&1&1&1&1&1&0&1&0&0&0&0&0&0&0 \\ 1&1&1&1&1&0&0&1&0&0&0&0&0&0&0&1&0&1&1&1&0&1&0&1&0&1&0&1&0&1&0&0 \\ 0&1&0&0&1&1&0&0&0&1&1&1&0&0&0&0&1&1&0&1&1&0&1&0&0&0&1&1&0&1&1&0 \\ 1&1&1&0&1&0&0&1&1&1&1&0&1&1&0&1&1&1&1&1&1&1&0&1&0&1&0&1&0&0&1&1 \\ 1&1&0&1&1&1&1&1&0&1&0&1&1&0&0&0&1&1&0&1&1&0&1&0&0&0&1&1&0&1&0&0 \\ 1&1&0&0&0&1&0&0&0&1&0&1&1&0&0&0&1&1&1&0&0&1&0&1&1&1&0&0&0&0&1&0 \\ 1&1&1&1&0&0&0&0&0&0&0&0&1&1&0&1&1&0&1&1&1&0&0&1&1&0&1&1&0&0&0&0 \\ 0&0&1&1&1&0&0&0&1&1&1&1&1&1&0&1&0&0&1&1&1&1&1&1&0&1&1&0&0&0&0&1 \\ 1&0&0&0&1&1&1&1&0&1&0&1&1&0&0&1&0&0&1&1&0&0&1&0&1&1&0&0&1&1&0&1 \\ 1&1&0&0&1&0&0&1&0&0&1&1&0&0&0&0&0&0&1&0&1&1&0&1&1&1&1&1&0&1&0&1 \\ 0&0&1&0&1&0&1&0&1&1&0&0&0&0&0&1&0&1&1&0&0&1&0&1&0&1&0&1&1&0&1&1 \end{bmatrix} \quad (3.5)$$

From (3.4) and (3.5), we can readily find that

$$\mathbf{K\,K^{-1} \bmod 2 = K^{-1}\,K \bmod 2 = I}. \quad (3.6)$$

Let us consider the image of a hand, which is given below.

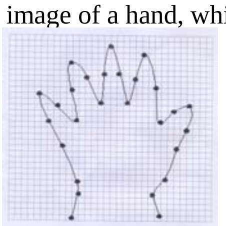

Fig. 2. Image of a Hand

This image can be represented in the form of a binary matrix P given by

$$P = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&1&0&1&0&0&0&1&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&1&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&1&0&0&1&1&1&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&0&0&0&0&1&0&0&1&0&1&0&0&1&1&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&1&0&0&0&1&0&0&1&0&1&0&0&1&1&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&1&0&0&1&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&0&0 \\ 0&0&0&0&0&1&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \end{bmatrix} \quad (3.7)$$

where 1 denotes black and 0 denotes white.

On adopting the iterative procedure given in Fig. 1, we get the encrypted image C in the form

© 2010, IJARCS All Rights Reserve

$$C = \begin{pmatrix}
1&0&0&1&0&0&1&1&1&1&1&0&0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&1&0&1&1\\
0&1&0&1&1&0&0&1&0&0&1&0&0&1&0&0&0&1&0&1&0&0&0&0&1&1&0&1&1&0&0&0\\
0&0&1&1&1&1&0&0&0&1&0&0&0&1&1&1&0&1&1&1&1&1&1&0&0&1&0&0&1&1&0&0\\
1&0&0&1&0&0&0&0&1&1&0&1&0&1&1&1&1&0&1&1&1&1&1&1&0&1&0&0&1&1&0&1\\
1&0&1&1&0&0&0&0&0&1&0&1&0&0&0&0&1&0&0&0&1&1&1&0&1&0&0&0&1&0&0&0\\
1&1&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0&0&1&0&1&1&1&0&0&1&1\\
0&1&0&0&1&0&0&0&1&1&1&0&0&1&1&1&0&1&0&1&1&0&1&0&0&1&1&1&1&0&1&1\\
1&1&1&1&1&1&1&0&1&0&0&0&1&0&1&1&0&1&1&1&1&1&1&0&0&0&0&0&0&0&1&1\\
0&0&1&0&1&1&1&1&0&0&1&0&0&0&0&1&0&0&0&1&0&1&1&1&1&1&0&1&1&0&0&1\\
1&0&0&0&1&0&0&1&0&0&1&0&0&0&0&1&1&0&1&1&1&1&1&0&0&1&0&1&0&1&0&1\\
0&1&1&1&0&0&1&0&0&1&1&0&1&1&1&0&1&1&1&0&0&1&1&1&1&1&1&1&1&1&1&0\\
1&0&0&1&1&0&0&0&1&0&0&0&1&1&0&0&1&0&0&1&0&0&1&0&1&0&0&0&0&0&0&1\\
0&1&1&1&0&1&0&0&1&1&1&0&0&1&1&0&1&1&1&1&0&0&0&1&1&1&0&0&1&0&1&1\\
1&1&0&0&1&1&1&1&0&0&1&1&0&1&1&1&0&0&1&0&1&0&0&0&1&1&1&0&1&0&0&0\\
0&0&1&0&1&0&0&1&1&0&0&1&0&0&0&1&1&0&0&0&1&1&0&1&1&1&1&0&0&1&0&1\\
1&1&1&0&1&0&1&1&0&0&1&0&1&1&0&1&1&1&1&0&0&1&1&0&1&0&1&1&1&0&0&1\\
1&1&1&1&1&0&1&0&0&0&0&1&1&0&1&0&1&0&0&1&1&0&1&1&0&1&1&0&0&1&0&0\\
0&1&1&0&0&1&0&0&0&0&0&1&1&1&1&0&1&0&1&0&1&1&0&0&0&0&1&1&0&0&1&0\\
0&0&1&0&0&0&1&1&0&1&1&0&1&0&0&1&1&0&1&1&1&0&1&0&1&0&0&0&1&1&0&0\\
1&1&1&0&1&1&0&1&1&1&1&1&1&1&0&1&1&1&1&1&0&1&1&0&1&1&0&0&0&0&0&0\\
1&0&1&1&1&0&0&1&0&0&1&1&0&1&0&1&0&1&1&0&0&0&1&0&0&1&0&1&1&0&1&0\\
0&1&1&1&0&0&0&1&1&1&0&0&1&0&1&0&0&1&0&0&0&1&1&1&0&1&1&1&0&0&1&1\\
0&1&1&0&0&1&1&0&0&1&1&0&1&0&1&0&1&0&0&0&0&1&1&0&0&0&0&0&0&0&0&1\\
0&1&0&1&0&1&1&1&1&0&0&1&1&0&1&0&0&1&1&1&1&0&1&0&0&1&0&0&0&1&0&1\\
0&0&0&0&0&1&1&0&1&0&1&1&1&1&1&1&0&0&0&1&1&1&0&1&0&0&1&0&0&1&0&1\\
0&1&0&0&1&0&0&1&0&1&0&1&1&0&1&1&1&1&1&1&1&0&1&0&0&0&1&1&1&1&0&1\\
1&1&1&0&0&0&0&1&1&0&1&1&1&1&0&1&0&0&1&1&0&1&0&0&1&0&0&0&1&0&0&0\\
1&0&1&0&1&1&1&1&1&0&1&0&1&0&0&1&1&1&1&1&1&1&0&1&0&0&0&1&1&0&1&0\\
0&1&0&0&1&0&1&1&0&1&0&0&1&1&1&1&1&0&0&0&1&0&0&0&0&1&0&0&0&1&0&1\\
0&0&0&0&1&1&1&1&0&0&0&1&1&1&0&1&1&0&0&1&1&0&1&1&1&1&1&0&1&1&1&1\\
0&0&1&1&0&0&1&1&0&1&0&0&0&1&0&1&0&1&0&1&1&0&1&1&1&1&1&0&1&0&1&0\\
0&0&0&1&1&1&0&1&1&0&0&1&1&1&1&0&0&1&0&1&1&1&1&1&0&0&1&1&1&1&0&0
\end{pmatrix} \tag{3.8}$$

On using (3.4), (3.5), (3.8), and the procedure for decryption (See Fig. 1.(b)), we get back the original binary image P, given by (3.7).

From the matrix C, on connecting each 1 with its neighbouring 1, we get an image which is in a zigzag manner (See Fig. 3).

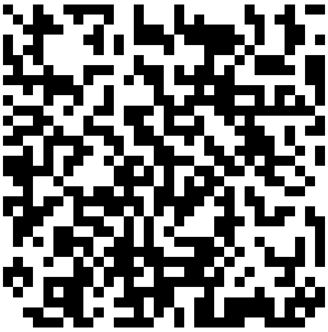

Fig. 3. Encrypted image of the hand

It is interesting to note that, the original image and the encrypted image differ totally, and the former one, exhibits all the features very clearly, while the later one does not reveal anything.

Now let us consider another image which is consisting of the upper half of a person. This is shown Fig. 4.

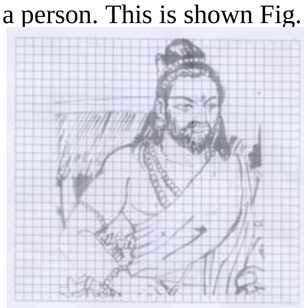

Fig. 4. Image of a Person

This image can be represented in the form of a matrix P, containing binary bits, as shown below.

$$P = \begin{pmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&1&1&0&0&0&1&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&1&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&1&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&1&1&1&1&1&0&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0\\
0&0&0&0&1&1&1&1&0&1&1&1&1&1&1&1&1&0&1&1&1&1&1&1&1&1&1&1&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&1&1&1&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&0&0&0&1&1&1&0&0&1&1&1&1&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&0&0&0&0&1&1&1&0&0&1&1&1&0&0&1&1&1&0&0&0&0&0\\
0&0&0&0&1&0&1&1&0&0&0&1&1&0&1&1&1&1&0&1&0&0&0&0&1&1&1&0&0&0&0&0\\
0&0&0&0&1&0&1&1&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1&0&0&0&0&0\\
0&0&0&0&1&0&1&0&0&0&0&1&0&0&0&1&1&1&0&0&0&0&0&0&1&1&1&0&0&0&0&0\\
0&0&0&0&1&1&1&1&1&0&0&1&1&1&0&1&1&0&0&0&0&0&1&0&1&1&1&0&0&0&0&0\\
0&0&0&0&1&1&0&0&1&1&1&1&0&1&1&1&0&0&0&1&0&1&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&1&1&0&1&1&1&1&0&0&1&0&0&0&0&0&1&1&1&1&0&0&0&0\\
0&0&0&0&1&0&1&1&1&1&1&1&1&1&1&1&0&1&0&1&1&0&0&1&1&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1&1&0&0&1&0&0&1&0&1&1&1&1&0&0&0&0\\
0&0&0&0&1&0&0&0&0&1&1&1&1&1&1&1&0&1&1&0&0&1&1&1&1&1&1&1&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&1&1&1&1&0&1&1&0&0&0&1&1&0&1&1&1&1&1&0&0&0&0\\
0&0&0&0&0&1&1&0&0&0&0&1&1&1&0&0&0&0&1&1&1&1&1&1&0&1&0&1&0&0&0&0\\
0&0&0&0&0&0&1&1&0&1&1&1&1&1&0&0&1&1&1&0&1&1&1&1&1&1&1&1&0&0&0&0\\
0&0&0&0&0&1&1&1&1&1&0&0&0&0&1&1&1&1&1&0&1&1&0&1&1&1&1&0&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&0&0&1&1&1&1&1&0&0&1&1&0&1&1&1&1&1&0&0&0&0\\
0&0&0&0&1&1&1&1&1&1&1&0&0&0&1&1&1&0&0&0&1&1&0&1&1&1&1&0&0&0&0&0
\end{pmatrix} \tag{3.9}$$

On using (3.4), (3.5), (3.9), and encryption algorithm given in Fig. 1.(a), we get the encrypted image C, given by

$$C = \begin{pmatrix}
1&0&1&0&0&0&0&1&1&1&0&0&0&0&0&1&1&1&0&1&1&1&0&0&0&0&0&1&0&0&0&0\\
1&0&0&0&1&0&1&0&0&0&1&1&0&0&0&0&0&1&0&1&0&0&0&1&0&0&0&1&1&0&1&0\\
0&0&1&0&1&0&1&0&0&0&0&1&0&1&0&0&1&0&1&1&0&0&1&1&0&0&0&1&0&0&1&0\\
0&0&0&1&1&1&1&0&0&0&1&0&1&1&1&1&0&1&1&1&1&0&1&0&1&0&1&0&1&1&0&1\\
0&1&0&1&0&0&1&1&1&1&0&0&1&0&1&1&0&0&1&0&0&1&0&0&1&1&1&1&0&0&1&0\\
0&0&0&1&1&1&0&1&1&0&0&0&0&0&0&1&1&0&1&0&1&1&1&1&0&0&0&1&0&0&0&0\\
0&0&1&1&0&0&1&0&1&1&0&1&1&1&0&0&1&1&0&1&0&0&1&1&1&0&1&0&0&1&0&0\\
1&1&1&0&0&0&0&0&1&1&0&0&1&1&1&0&0&1&0&1&1&1&1&1&0&1&1&0&1&1&0&1\\
1&0&1&1&1&1&0&1&1&0&0&1&0&0&0&1&0&0&1&0&0&1&1&0&0&0&0&1&0&1&0&1\\
0&1&0&1&1&0&0&1&1&1&1&0&0&1&0&1&1&0&0&1&1&1&0&0&0&0&0&1&1&1&1&0\\
1&1&1&1&0&0&1&1&0&0&0&1&1&0&1&1&0&0&0&0&1&0&1&0&1&0&1&1&1&0&1&0\\
0&0&1&0&1&0&1&1&1&0&1&1&0&0&0&0&1&1&1&0&1&1&1&0&0&1&0&1&0&1&0&1\\
0&0&1&0&1&1&0&0&0&1&1&1&0&0&1&1&1&0&1&1&1&1&1&1&0&0&0&1&1&0&1&1\\
0&1&1&0&0&1&1&1&1&1&1&0&1&0&0&1&1&0&1&1&1&0&1&0&0&0&1&1&0&1&1&1\\
1&0&1&0&1&1&0&1&0&0&1&1&0&0&0&1&1&0&1&0&1&1&0&1&0&0&1&0&0&1&0&1\\
0&0&0&1&1&1&1&0&0&1&0&0&0&0&0&1&0&1&0&0&1&0&0&0&1&1&0&1&1&0&1&0\\
1&0&0&0&1&1&1&1&0&0&0&1&1&0&1&0&1&1&0&1&0&1&1&0&0&1&0&0&0&0&0&1\\
0&1&1&1&1&1&1&1&1&0&0&0&1&0&1&1&0&1&0&1&0&0&1&0&1&1&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&1&1&1&0&1&0&1&0&0&1&0&1&1&0&0&1&1&1&0&0&0&0&0&0\\
0&0&0&1&0&0&1&0&0&1&1&1&0&1&0&1&0&1&0&0&0&1&0&1&1&1&1&1&1&1&1&0\\
1&0&0&0&1&0&1&0&1&1&0&1&0&0&0&1&0&1&0&1&1&0&0&0&0&1&0&1&1&1&0&0\\
1&0&1&1&1&1&0&1&1&0&1&0&0&1&1&1&1&0&0&1&0&0&0&0&0&0&1&1&1&1&0&0\\
1&0&1&1&1&1&1&1&1&1&0&0&1&0&1&0&1&1&0&0&0&0&1&0&0&0&1&1&1&1&1&0\\
1&0&0&1&1&0&1&1&1&0&1&0&0&1&0&1&0&0&0&0&1&0&1&1&0&0&1&1&0&1&1&0\\
0&0&0&1&0&0&1&1&0&0&0&0&0&1&1&0&1&1&0&0&0&1&0&1&0&0&1&1&0&1&1&0\\
1&0&1&1&0&1&0&1&1&0&1&0&1&0&0&0&0&0&0&0&0&1&0&1&1&1&0&0&0&1&1&0\\
1&0&1&0&0&0&1&1&1&1&1&1&0&0&1&1&0&1&1&0&0&0&0&0&1&0&1&1&0&0&0&0\\
1&1&1&0&0&0&1&0&1&0&1&0&0&0&1&0&1&1&1&0&1&1&1&1&1&1&0&1&1&0&1&1\\
1&0&0&0&0&1&1&1&0&0&1&1&1&0&1&0&1&0&0&1&1&1&0&0&1&0&0&0&1&1&1&0\\
1&1&1&1&1&1&0&0&1&0&1&0&1&1&0&1&0&1&0&1&1&1&1&0&0&0&1&0&0&1&0&0\\
1&1&0&1&0&0&1&1&0&0&0&1&0&0&1&1&1&0&1&1&1&1&0&0&0&0&0&0&1&0&0&1\\
1&0&0&1&0&0&0&0&0&0&1&1&1&1&1&0&1&0&1&0&1&0&1&0&0&1&0&1&1&0&0&1
\end{pmatrix} \tag{3.10}$$

On applying the process of decryption given in Fig. 1.(b), we get the original binary image given in (3.9).

On comparing the matrices of P and C, we find that, the 1s in P are scattered when P is transformed into C. On connecting 1s in C, we get an image as shown in Fig. 5.

Fig. 5. Encrypted image of a Person

© 2010, IJARCS All Rights Reserve

Here, we also notice that, the features of the original image are totally lost, when the original image is transformed into the encrypted one.

IV. CONCLUSIONS

In this analysis, we have made use of a modified Hill cipher for encrypting binary images. In this, we have illustrated the procedure by considering a pair of examples: (1) the image of a hand, and (2) the image of the upper half of a person.

Here, we have noticed that, the encrypted images are totally different from the original images, and the security of the images is completely enhanced, as no feature of the original images can be traced out in any way from the encrypted images.

This analysis can be extended for the images of signatures and thumb impressions.

V. REFERENCES

[1] U. K. Sastry, D. S. R. Murthy, S. Durga Bhavani, “A Block Cipher Involving a Key Applied on Both the Sides of the Plain Text”, International Journal of Computer and Network Security (IJCNS), Vol. 1, No. 1, pp. 27 – 30, Oct. 2009.

[2] V. U. K. Sastry, D. S. R. Murthy, S. Durga Bhavani, “A Block Cipher Having a Key on One Side of the Plain Text Matrix and its Inverse on the Other Side”, Accepted for publication in International Journal of Computer Theory and Engineering (IJCTE), Vol.2, No. 5, Oct. 2010.

Authors:

Dr. V. U. K. Sastry is presently working as Professor in the Dept. of Computer Science and Engineering (CSE), Director (SCSI), Dean (R & D), SreeNidhi Institute of Science and Technology (SNIST), Hyderabad, India. He was Formerly Professor in IIT, Kharagpur, India and worked in IIT, Kharagpur during 1963 – 1998. He guided 12 PhDs, and published more than 40 research papers in various international journals. He is a Member, Editorial Board and Reviewer of International Journal of Computational Intelligence and Information Security (IJCIIS), Senior Member of International Association of Computer Science and Information Technology (IACSIT) and Reviewer of International Journal of Computer and Network Security (IJCNS). His research interests are Network Security & Cryptography, Image Processing, Data Mining and Genetic Algorithms and published research papers in International Journal of Computer and Network Security (IJCNS), International Journal of Computer Theory and Engineering (IJCTE) and International Journal of Computational Intelligence and Information Security (IJCIIS).

Dr. S. Durga Bhavani is presently working as Professor in School of Information Technology (SIT), JNTUH, Hyderabad, India. She has more than 18 years of teaching experience. Her research area includes Evidential Reasoning, Cryptography and Image Processing. She has no. of research publications to her credit.

Prof. D. S. R. Murthy obtained B. E. (Electronics) from Bangalore University in 1982, M. Tech. (CSE) from Osmania University in 1985 and presently pursuing Ph.D. from JNTUH, Hyderabad since 2007. He is presently working as Professor in the Dept. of Information Technology (IT), SNIST since Oct. 2004. He earlier worked as Lecturer in CSE, NIT (formerly REC), Warangal, India during Sep. 1985 – Feb. 1993, as Assistant Professor in CSE, JNTUCE, Anantapur, India during Feb. 1993 – May 1998, as Academic Coordinator, ISM, Icfaian Foundation, Hyderabad, India during May 1998 – May 2001 and as Associate Professor in CSE, SNIST during May 2001 - Sept. 2004. He worked as Head of the Dept. of CSE, JNTUCE, Anantapur during Jan. 1996 – Jan 1998, Dept. of IT, SNIST during Apr. 2005 – May 2006, and Oct. 2007 – Feb. 2009. He is a Fellow of IE(I), Fellow of IETE, Senior Life Member of CSI, Life Member of ISTE, Life Member of SSI, DOEACC Expert member, and Chartered Engineer (IE(I) & IETE). He is a Reviewer of International Journal of Advanced Research in Computer Science (IJARCS), International Journal of Computational Intelligence and Information Security (IJCIIS) and International Journal of Computational Intelligence and Information Security (IJCIIS). He is a member of International Association of Computer Science and Information Technology (IACSIT). He published a text book on C Programming & Data Structures. His research interests are Image Processing and Image Cryptography and published research papers in International Journal of Computer and Network Security (IJCNS), International Journal of Computer Theory and Engineering (IJCTE), International Journal of Computational Intelligence and Information Security (IJCIIS) and in International Journal of Advanced Research in Computer Science (IJARCS).

© 2010, IJARCS All Rights Reserve