Investigation of Matrix Encryption Algorithms

Mina Farmanbar

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the Degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
January 2012
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Elvan Yılmaz
Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Muhammed Salamah
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Alexander Chefranov
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Alexander Chefranov
2. Asst. Prof. Dr. Gürcü Öz
3. Asst. Prof. Dr. Önsen Toygar

ABSTRACT

There are many applications that have been developed in order to protect information from any public network attacks. One of the important aspects for secure transformation of information is cryptography, which is the focus of this thesis. We only focus on the Hill cipher algorithm. The Hill cipher can be broken by known plaintext-ciphertext attack due to its linearity. There are several Hill cipher modifications, which have been proposed in the literature of cryptography to make it stronger.

In this study, we deal with two different kinds of Hill cipher modifications which use interweaving, interlacing and 16 iterations. Both proposed ciphers have a substantial avalanche effect and are supposed to resist any cryptanalytic attack. However, it is not discussed why interweaving and interlacing with the iteration strengthen the Hill cipher and why the number of iterations is taken to be 16.

We show that strength of proposed ciphers is due to non-linear transformation used in bit-level permutations and investigate effect of number of iterations on the avalanche effect. We modify Hill cipher modifications [7, 8] by introducing columns swapping HC (CSHC) that uses swapping of columns of the binary bits of the plaintext characters instead of interlacing and interweaving as in [7, 8]. Another modification is arbitrary permutation Hill cipher (APHC) that uses an arbitrary permutation not known to an opponent instead of a fixed permutation (interweaving, interlacing and etc.). In these ciphers 1 or 2 iterations are used instead of 16 iterations as in [7, 8]. The test results obtained by comparison analysis of two new proposed ciphers indicate that any bit-level permutations can provide a considerable avalanche effect and two iterations can be used instead of 16.

Keyword: Hill cipher, linear transformation, Iteration, permutation, Avalanche effect

ÖZ

Kamuya açık ağlarda veri güvenliğini sağlamak için bir çok uygulama geliştirilmiştir. Veri dönüşüm güvenliğinin en önemli parçalarından biri şifrelemedir. Biz de bu tezde şifreleme konusuyla ilgili olarak, Hill Şifreleme Algoritması üzerinde durduk. Hill şifreleme yöntemi doğrusal olduğu için bilinen salt metin-şifreleme yöntemiyle kırılabilir. Literatürde, Hill şifrelemesini güçlendirmek için yapılmış çalışmalar mevcuttur.

Bu çalışmada, Hill şifrelemesini güçlendirmeyi amaçlayan iki yöntem [7 ve 8] üzerinde durduk. Bu yöntemler birlikte dokuma, birbirine dolama tekniklerini 16 kere tekrarlayarak kullanmaktadır. Ancak birlikte dokuma ve birbirine dolama tekniklerinin neden Hill şifrelemeyi güçlendirdiğinden veya tekrar sayısının neden 16 olduğundan bahsedilmemektedir.

Biz, öne sürülen şifrelemelerin gücünün, bit-seviyesi karıştırmalarındaki doğrusal olmayan dönüşümlerden kaynaklandığını gösterdik. Ek olarak da tekrar sayısının çığ etkisi üzerindeki etkisini araştırdık. Biz, Hill şifreleme üzerinde yapılan yeni değişiklikler olan [7 ve 8] in sahip olduğu birlikte dokuma ve birbirine dolama yöntemleri yerine, salt metindeki karakterlerin bitlerinin oluşturduğu sütunlarda yer değiştirme (CSHC) yöntemini öne sürdük. Diğer bir değişiklik ise sabit ve bilinen bir karıştırma yöntemi (birlikte dokuma, birbirine dolama, vs...) yerine, rastgele karıştırma yönteminin kullanımıdır. Bu yöntemlerde, 16 kere tekrarlama yerine 1 veya 2 defa tekrarlama kullanılmaktadır.

Öne sürülen yeni şifreleme yöntemlerinin karşılaştırma sonuçları, herhangi bir bit-seviyesi karıştırmanın önemli derecede çığ etkisi yarattığını ve 16 kere tekrarlama yerine 2 kere tekrarlamanın da kullanılabileceğini göstermiştir.

Anahtar Kelimeler: Hill şifreleme, doğrusal dönüşüm, tekrarlama, karıştırma, çığ etkisi

ACKNOWLEDGMENT

I would never have been able to complete this dissertation without the help of the people who have supported me with their best wishes.

I would like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Alexander Chefranov for his efforts and supports for doing this research. I sincerely thank to the committee members of my thesis defense jury for their helpful comments on this thesis. I gratefully acknowledge the chairman of Computer Engineering Department Assoc. Prof. Dr. Muhammad Salamah.

Last but not least I would also like to thank my dear parents, my elder brother, and younger sister for their continuous supports in my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Cryptography

In the age of universal computer and electronic connectivity, Internet provides communication between people on the world. Everyone expects to transmit information in a secure and trusted communication. So maintaining privacy of information becomes the most important issue in the network security. To protect information from any public network attacks particularly Internet, cryptography is necessary to come. Cryptography enables people to store information or transfer it over any untrusted networks. [1, 2]

According to [3], “cryptography is the science of encryption which uses mathematics to transform the information content in a secure form”. The word “Cryptography” comes from the Greek *kryptos*, that means secret or hidden and *graphos* that means writing. Cryptography is the study of transferring intelligible message to unintelligible message and retransferring back altered message to the original one. In all cryptosystems, the original message is called **Plaintext** and encrypted message is called the **Ciphertext**. The process of converting the plaintext to the ciphertext to make it difficult to be read is called as **Encryption,** and the process of converting the ciphertext back into its original form is called as **Decryption**. A **Cipher** is any method of encrypting information. A **Block Cipher** is one that breaks a message up into blocks and combines a key with each block. The study of methods for revealing the encrypted information without access to the secret information that is normally required is called the **Cryptanalysis".**

Cipher can be strong or weak. A strong cipher provides a ciphertext which is very difficult to read and resist to any cryptanalysis attacks but it is simply possible to read ciphertext which is the result of the weak cipher.

Nowadays, cryptography is divided into two categories, the symmetric key encryption and asymmetric key encryption. Symmetric key is also called conventional cryptography or secret key. In the secret-key encryption a single key is used for both encryption and decryption. In this, sender and receiver share a common key which is secret in both sides. So it cannot be read by anyone who is not allowed to read except the intended recipient. In contrast to symmetric key, asymmetric key encryption uses two different keys, one to encrypt the message and one to decrypt the message. One of them should be kept secret and the other one can be shared on the network. The secret key is called private key which is used to encrypt the message and the other key is called public key. Asymmetric key encryption is also called public key encryption. [3]

1.2 Hill Cipher

In classical cryptography, Hill Cipher is a symmetric key cipher which enciphers blocks of the plaintext. The Hill cipher was invented in 1929 by Lester S.Hill [4]. In the Hill cipher, ciphertext C is obtained by multiplication of a plaintext vector P by an $n \times n$ key matrix K, i.e. by a linear transformation. In this context, the basic steps of encryption and decryption are given by

$$C = KP \ (mod\ m), \tag{1.1}$$

and

$$P = K^{-1}C \ (mod\ m), \tag{1.2}$$

where K^{-1} is the modular arithmetic inverse of K, and $m > 1$ is an integer.

The Hill cipher is a polygraphic substitution cipher based on linear algebra and matrix transformation which uses matrices and matrix multiplication to transform blocks of plaintext letters into blocks of ciphertext. One of the basic components of classical ciphers is substitution cipher. [5]

A substitution cipher is a method of encryption that one letter is replaced by another letter. We have many types of substitution ciphers such as monoalphabetic ciphers, homophonic ciphers, polygraphic ciphers and polyalphabetic substitution ciphers.

In the monoalphabetic Substitution cipher which also called as Simple Substitution cipher, each character in the plaintext is replaced by a corresponding one from a cipher alphabet. In contrast to simple substitution, polygraphic substitution substitutes larger letter groups instead of individual plaintext letters.

The Hill cipher has several advantages in data encryption such as disguising letter frequencies of the plaintext, its simplicity because of using matrix multiplication and inversion for encryption and decryption, and its high speed [6]. However, it can be broken by known plaintext-ciphertext attack due to its linearity and this is the major drawback of the Hill cipher.

Symmetric encryption algorithms need to satisfy Diffusion and Confusion to be considered strong.

Diffusion means that several characters of the ciphertext should be changed by changing one character of the plaintext and similarly, if we change a character of the ciphertext, then several characters of the plaintext should change.

Confusion means that there is a relationship between the key matrix and the ciphertext. It means that every character in the key influences every character of the ciphertext.

1.3 Objectives

In the recent years, several significant modifications of the Hill cipher [7, 8, 9, 10], have been developed in order to modify it and achieve higher security. In [7, 8], the Hill cipher is modified by including interweaving, interlacing and iteration. Proposed in [7, 8] ciphers have avalanche effect and are supposed to resist any cryptanalytic attack. However, it is not discussed why interweaving and interlacing with the iteration strengthen the Hill cipher and why the number of iterations is taken to be 16.

In this thesis we show that strength of the ciphers [7, 8] is due to non-linear transformation used in it (bit-level permutations), investigate impact of number of iterations on the avalanche effect, and propose generalizations of the ciphers from [7, 8].

We compare the results of two experiments studying how changing number of iterations influences on the avalanche effect. Then we propose two new modifications by introducing columns swapping HC (CSHC) and arbitrary permutation HC (APHC) in order to provide a good avalanche effect. The examination of the proposed ciphers will be discussed.

1.4 Layout of the Thesis

The rest of the thesis is organized as follows. Chapter 2 gives a brief literature review. It also presents the recent modifications of the Hill cipher. Chapter 3 is dedicated to all the theorems that are used in this thesis such as demonstration of nonlinearity of the Hill cipher modifications in [7, 8] i.e. by proving non-linearity of bit-level permutation and some examples of the weaknesses of integer permutation due to its linearity. The investigation of the number of iterations, experimental analysis, and obtained results are discussed in Chapter 4. Chapter 5 contains the description of our proposed ciphers (CSHC) and (APHC) and comparison of test results. Chapter 6 presents the conclusion.

Chapter 2

LITERATURE REVIEW

Hill cipher is based on matrix multiplication. Multiplication of matrices is a linear transformation. Also any other compositions of matrices are linear. Therefore, the Hill cipher is seen to be a linear transformation. This is the major drawback of Hill cipher, so it can be broken by the known plaintext-ciphertext attack.

In the recent years, several significant modifications of Hill cipher [7, 8, 9, 10] have been developed in the literature of cryptography. In all these investigations, researchers have attempted to improve the security of the Hill cipher. They have tried to avoid linearity of the matrix transformation when constructing cryptosystems. Hence, The Hill cipher has been modified by including permutation on the bit-level matrix transformation and it is resistant to known plaintext-ciphertext attack.

Here, some new modifications of the Hill cipher are discussed. Modified Hill cipher [7] uses interlacing and iteration and [8] uses interweaving (transposition of the binary bits of the plaintext letters) and iteration in both encryption and decryption. In [9] a pair of key matrices is used. A key on one side of the plaintext matrix and its inverse on the other side are used in [10]. In [7, 8] each letter of the plaintext is represented into a binary form under consideration in terms of ASCII code and 128 is used as a fundamental operation. They are described as follows:

Input:

1. A plaintext P of 2n 7-bit ASCII characters is represented as:

$$P = [P_{ij}], i = 1\ to\ n, j = 1\ to\ 2 \tag{2.1}$$

2. A key matrix of 7-bit ASCII characters is also represented as:

$$K = [K_{ij}], i = 1\ to\ n, j = 1\ to\ n \tag{2.2}$$

and K^{-1} is shared between both communication parties.

Algorithm of encryption used in [7, 8]:

1. $P^0 = P$ (2.3)
2. For i $= 1, 2, \dots, 16$ do the following:
    2.1. Compute $P^i = KP^{i-1}\ mod\ 128$
    2.2. P^i = interlace (P^i) or P^i = interweave (P^i)
3. Compute $C = KP^N\ mod\ 128$

Algorithm of decryption used in [7, 8]:

1. Compute $P^N = K^{-1}C\ mod\ 128$ (2.4)
2. For $i = N, \dots, 1$ do the following:
    2.1. P^{i-1} = Iinterlace (P^i) as used in [7], or P^{i-1} = Iinterweave (P^i) as used in [8].

2.2. Compute $P^{i-1} = K^{-1}P^{i-1} \; mod \; 128$

3. $P = C^1$

Algorithm for interweave (P):

1. Convert P into a binary $n \times 14$ matrix

$$B = \begin{bmatrix} b_{1,1} & \dots & b_{1,14} \\ \vdots & & \vdots \\ b_{n,1} & \dots & b_{n,14} \end{bmatrix}$$

2. Rotate circular upward the jth column of B to get new column as $\begin{bmatrix} b_{2,j} \\ b_{3,j} \\ \vdots \\ b_{n,j} \\ b_{1,j} \end{bmatrix}$ where $j = 1, 3, 5, \dots$.
3. Similarly, rotate circular leftward the jth row of B where $j = 2, 4, 6, \dots$.
4. Construct P from B using first 7 bits of jth row for $P_{j,1}$, and last 7 bits for $P_{j,2}$, $j = 1, 2, \dots, n$

Algorithm for interlace (P):

1. Divide P into two binary $n \times 7$ B and D matrices, where $b_{k,j} = p_{k,j}$ and $d_{k,j} = p_{k,j} + 7$, $k = 1, 2, \dots, n$, $j = 1, 2, \dots, 7$
2. Mix $b_{k,j}$ and $d_{k,j}$ in terms of each $b_{k,j}$ lies adjacent to its corresponding $d_{k,j}$.

$$\mathrm{B}' = \begin{bmatrix} b_{1,1} & d_{1,1} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & b_{1,7} & d_{1,7} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & b_{\mathrm{n}/2,7} & d_{\mathrm{n}/2,7} \end{bmatrix}$$

$$\mathrm{D}' = \begin{bmatrix} b_{\mathrm{n}/2+1,1} & d_{\mathrm{n}/2+1,1} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & b_{\mathrm{n}/2+1,7} & d_{\mathrm{n}/2+1,7} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots & b_{\mathrm{n},7} & d_{\mathrm{n},7} \end{bmatrix}$$

3. Construct $P_{j,1}$ from $\mathrm{B}'_{j,1:7}$ and P_{j2} from $\mathrm{D}'_{j,1:7}$, $j = 1\ to\ n$

Algorithm of Inverse of interlace used in [7]:

```
1. $l = 1$
2. Convert $P$ into binary bits
3. For $i = 1, \dots, n$
   3.1. For $j = 1, \dots, 7$
   3.2. Temp $(l) = b_{ij}$
   3.3. Temp $(l + n \times 7)$= $d_{ij}$
   3.4. $l = l + 1$
4. $l = 1$
5. For $i = 1, \dots, n$
   5.1. For $j = 1, \dots, 7$
   5.2. $b_{ij}$= temp $(l)$
   5.3. $d_{ij}$=temp $(l + 1)$
```

5.4. $l = l + 2$

6. Convert binary bits to decimal numbers

Algorithm for Inverse of interweave (P) used in [8]:

1. Convert P into a binary $n \times 14$ matrix

$$B = \begin{bmatrix} b_{1,1} & \cdots & b_{1,14} \\ \vdots & & \vdots \\ b_{n,1} & \cdots & b_{n,14} \end{bmatrix}$$

2. Rotate circular rightward the jth row of B where $j = 2, 4, 6, \dots$.

3. Rotate circular downward the jth column of B to get new column as $\begin{bmatrix} b_{n,j} \\ b_{1,j} \\ \vdots \\ b_{n-1,j} \end{bmatrix}$ where $j = 1, 3, 5, \dots$.

4. Construct P from B using first 7 bits of jth row for $P_{j,1}$, and last 7 bits for $P_{j,2}$, $j = 1, 2, \dots, n$

In proposed algorithms, both interweaving and interlacing is a type of bit-level permutations and 16 iterations are used. The cryptanalysis and the avalanche effect of these ciphers have indicated that ciphers cannot be broken by any cryptanalytic attack.

In [9] another modification of the Hill cipher is proposed by introducing a pair of key matrices. One key on the left side of plaintext matrix and one on the right side of plaintext are used as matrix multiplications. Each letter of the plaintext is represented into a binary form under consideration in terms of EBCDIC code and 256 is used as a fundamental operation. Also an iterative process which includes a bit-level permutation on the matrix transformation is used. The avalanche effect and cryptanalysis clearly shows that the cipher can be strong one. Their cipher process is described as follows:

Input:

1. A plaintext P of EBCDIC characters is represented in the form of square matrix given by:

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

2. Two key matrices are as:

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

$L = [L_{ij}], i = 1\ to\ n, j = 1\ to\ n$

Algorithm of encryption used in [9]:

```
1. $P^0 = P$
2. For $i = 1, 2, \dots, N$ do the following:
    2.1. Compute $P^i = (KP^{i-1}\, L)\ mod\ 256$
    2.2. $P^i =$ Permute $(P^i)$
3. $C = P^N$
```

Algorithm for Permute (P):

1. Convert P into a binary $n^2 \times 8$ matrix as:

$$B = \begin{bmatrix} b_{11} & \dots & b_{1,8} \\ \vdots & & \vdots \\ b_{n^2,1} & \dots & b_{n^2,8} \end{bmatrix}$$

2. Divide B into two halves. The upper half contains $n^2/2$ rows and eight columns, similarly the lower half.
3. Map upper half and lower half into D and E two matrices containing n^2 rows and four columns. Mix b_{ij} and placed it into D and E as follows:

$$D = \begin{bmatrix} b_{n^2/2\,,8} & \cdots & b_{(n^2/2)-6\,,1} \\ \vdots & \vdots & \vdots \\ b_{n^2/2\,,1} & \cdots & b_{(n^2/2)-6\,,1} \\ b_{(n^2/2)-1\,,8} & \cdots & b_{(n^2/2)-7\,,1} \\ \vdots & \vdots & \vdots \\ b_{(n^2/2)-1\,,1} & \cdots & b_{(n^2/2)-7\,,1} \end{bmatrix}$$

$$E = \begin{bmatrix} b_{(n^2/2)+1\,,1} & \cdots & b_{(n^2/2)+7\,,1} \\ \vdots & \vdots & \vdots \\ b_{(n^2/2)+1\,,8} & \cdots & b_{(n^2/2)+7\,,8} \\ b_{(n^2/2)+2\,,1} & \cdots & b_{n^2\,,1} \\ \vdots & \vdots & \vdots \\ b_{(n^2/2)+2\,,8} & \cdots & b_{n^2\,,8} \end{bmatrix}$$

4. Construct P_{j1} from D and P_{j2} from $j = 1\ to\ n^2$.

Algorithm of decryption used in [9]:

1. Find K^{-1} and L^{-1}
2. For $i = N, \dots, 1$ do the following:
   2.1. $C^{i-1} = \text{IPermute}\ (C^{i})$
   2.2. Compute $C^{i-1} = (K^{-1}C^{i-1}L^{-1})\ mod\ 256$
3. $P = C^{1}$

The pair of the key matrices, one on the left side of the plaintext and the other key on the right side of the plaintext and the process of permutation are totally responsible for the strength of the cipher.

Now we present a recently modified Hill cipher [10] by having a key on one side of the plain text matrix and its inverse on the other side. In this cipher bit-level permutation which named Mix process is applied. The Mix process mixes the plaintext at every stage of the iteration. The process of encryption is as follows:

Input:

1. A plaintext P is represented as a square matrix by:

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

where each P_{ij} is equal to 0 or 1.

2. A key matrix can be represented as:

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

where each K_{ij} is a binary number.

Algorithm of encryption used in [10]:

```
1. $P^0 = P$
2. For $i = 1, 2, \ldots, N$ do the following:
    2.1. Compute $P^i = (KP^{i-1}K^{-1}) \ mod \ 2$
    2.2. $P^i = \mathrm{Mix}(P^i)$
    2.3. $P^i = P^i \oplus K$
3. $C = P^N$
```

Algorithm for Mix (P):

1. Convert P into a binary $n \times 8n$ matrix as:

$$B = \begin{bmatrix} b_{111} & b_{112} \ldots & b_{118} & b_{121} & b_{122} \ldots & b_{128} & \ldots & b_{1n8} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ b_{n11} & b_{n12} & b_{n13} & b_{n21} & b_{n22} & b_{n28} & \ldots & b_{nn8} \end{bmatrix}$$

here, b_{111},b_{112}, …, b_{118} are binary bits corresponding to b_{11}. Similarly, b_{ij1}, b_{ij2},…, b_{ij8} are the binary bits representing b_{ij} where $i = 1$ to n, $j = 1$ to n.

2. Divide B into n^2 substrings. Matrix B can be considered as a single string in a row wise manner. As the length of the string is $8n^2$, it is divided into n^2 substrings, wherein the length of each substring is 8 bits.
3. If n^2 is divisible by 8, we focus our attention on the first 8 substrings. We place the first bits of these 8 binary substrings, in order, at one place and form a new binary substring. Similarly, we assemble the second 8 bits and form the second binary substring. Continuing in the same way, we construct all the binary substrings obtained from the plaintext. However, if n^2 is not divisible by 8, then we consider the rest of the string, and divide it into two halves. Then we mix these two halves by placing the first bit of the second half, just after the first bit of the first half, the second bit of the second half, next to the second bit of the first half, etc. Thus we get a new binary substring corresponding to the remaining string.
4. Construct P from B.

Algorithm of decryption used in [10]:

```
1. For $i = N, \dots, 1$ do the following:
    1.1. Compute $C^{i-1} = (KC^iK^{-1}) \, mod \, 2$
    1.2. $C^{i-1}$ = IMix $(C^{i-1})$
    1.3. $C^{i-1} = C^{i-1} \oplus K^{-1}$
2. $P = C^1$
```

Each specified ciphers which is described above, has its own avalanche effect and its own strong and weak points. All of them have used $N = 16$ iterations to provide a high performance but in the following chapters we discuss that number $N = 16$ of iterations is not a distinguished one and less number of iterations may be used instead.

Chapter 3

LINEAR AND NON-LINEAR MATRIX TRANSFORMATION

A transformation T is a linear transformation if it satisfies the following property:

$$\mathrm{T}(a_1\mathrm{X} + a_2\mathrm{Y}) = a_1\mathrm{T}(\mathrm{X}) + a_2\mathrm{T}(\mathrm{Y}) \tag{3.1}$$

where a_1, a_2 are scalars. Let's show some examples of linear and non linear transformation. Let us prove that how a transformation is linear or not. For example, consider the transformation:

$$T\binom{x}{y} = \binom{x}{y}$$

To prove that it is linear, let’s take the vector $V = \binom{2}{2}$ then $T(2V) = \binom{4}{4}$ that is equal to $2T(V) = \binom{4}{4}$.

To show that the transformation can be not linear, consider $T\binom{x}{y} = \binom{x^3}{y^2}$. To prove that it is not linear, $T(2V) = \binom{64}{16}$ which is not equal to $2T(V) = \binom{16}{8}$. So T is not linear because it doesn't keep scalar multiplication.

In cryptography, linearity is one of the important issues in the design of substitution ciphers. Linear systems are easy to exploit i.e. by one bit changing of the key, a fixed set of bits in the ciphertext will be changed simultaneously. Assume that plaintext $P = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ is encrypted with two different keys $K_1 = \begin{bmatrix} 2 & 1 \\ 3 & 5 \end{bmatrix}$ and $K_2 = \begin{bmatrix} 2 & 1 \\ 3 & 7 \end{bmatrix}$ which differ by one number. As it is shown below:

$$C_1 = K_1 P = \begin{bmatrix} 2 & 1 \\ 3 & 5 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 5 & 8 \\ 18 & 26 \end{bmatrix}$$

$$C_2 = K_2 P = \begin{bmatrix} 2 & 1 \\ 3 & 7 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 5 & 8 \\ 24 & 34 \end{bmatrix}$$

Some numbers will be same in the obtained ciphertexts C_1 and C_2, and some numbers will be different. So opponent can determine a good deal about the keys. Cryptographers try to design a perfectly non-linear system i.e. by changing one bit of the key or any key compositions, each bit of the ciphertext will be changed approximately 50 percent of the original one. Although it has been impossible to have a perfect non-linear substitution, but every well-designed encryption system tries to make its substitutions as non-linear as possible.

In this chapter we discuss about both integer-level and bit-level permutations on the matrix transformation which have been used in order to strengthen ciphers. We illustrate that bit-level permutation makes totally matrix transformation non linear. In addition, we show that any integer-level permutation provides overall linear matrix transformation.

In many ciphers such as Rail fence cipher and Route cipher, integer-level permutation is used to change the sequence of the numbers, while numbers may be visible or repeat in the ciphertext. Consequently, an opponent sets a linear system of equations and can find a relation between plaintext and corresponding ciphertext by re-arranging the ciphertext integer numbers and reveal the information. It is one of the known weaknesses in the linear- based ciphers.

In contrast, some other ciphers [7, 8, 9, 10] have used permutation on the binary forms of the message bit by bit. In these, the message characters are represented into their binary forms and then are permuted on the bit-level of the matrix transformation. So, strength of the ciphers is obtained due to non-linear matrix transformation. It can be significant because one changing in a bit can make a lot of changes in the whole matrix transformation.

3.1 Definition of Permutation

All possible arrangements of a sequence of n elements produces $n!$ permutations which is a huge number for large n (e.g., $n = 100$). A permutation can be identified by a matrix whose entries are all 0's and 1's, with exactly one 1 in each row and exactly one 1 in each column. For example permutation matrix $P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ is obtained from permutation (2, 1) which means that the position of second row and first row is swapped. According to [11], "pre-multiplying any $n \times n$ matrix A by an $n \times n$ permutation matrix P has the effect of re-arranging the rows of A. For example, if the permutation matrix P is obtained by swapping rows i and j of the $n \times n$ identity matrix, then rows i and j of A will be swapped in the product PA" which is shown below:

$$PA = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 5 & 11 \end{bmatrix} = \begin{bmatrix} 5 & 11 \\ 2 & 1 \end{bmatrix}$$

According to [11], “post-multiplying any $n \times n$ matrix A by an $n \times n$ permutation matrix P has the effect of re-arranging the columns of A. For example, if the matrix P is obtained by swapping rows i and j of the $n \times n$ identity matrix, then columns i and j of A will be swapped in the product AP” which is shown below:

$$AP = \begin{bmatrix} 2 & 1 \\ 5 & 11 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 11 & 5 \end{bmatrix}$$

3.2 Examples of Linear and Non-Linear Matrix Transformation

In this section, the strength of the ciphers [7, 8] which is due to non-linear transformation used in it (bit-level permutations) is discussed. In order to illustrate our aim let’s consider some examples, in which bit-level and integer-level permutations are used after matrix multiplication.

Examples below show that due to linearity, known plaintext-ciphertext attack is applicable in the case of integer-level permutation, and is already non-applicable in the case of trivial bit-level permutation that swaps just two bits.

We use in the both examples below the 2×2 key $K = \begin{bmatrix} 19 & 12 \\ 21 & 13 \end{bmatrix}$ and a pairs of plaintext-ciphertext matrices, $X_1 = \begin{bmatrix} 12 & 3 \\ 5 & 4 \end{bmatrix}$, $Y_1 = \begin{bmatrix} 2 & 1 \\ 5 & 11 \end{bmatrix}$, and $X_2 = \begin{bmatrix} 4 & 2 \\ 12 & 3 \end{bmatrix}$ which is considered as a new plaintext block. Let’s

$Y_i' = integer - level\, permutation\, (Y_i, Permutation\, P),$

and

$Y_i'' = bit - level\ permutation\ (Y_i, Permutation\ P)$,

where Y_i is some 2×2 integer ciphertext matrix where $i = 1, 2$, and P is an integer or binary level permutation.

Integer-level permutation returns 2×2 integer matrix with somehow permuted rows of Y_i and bit-level permutation returns 2×2 matrix with elements obtained from Y_i by some permutation of its bits.

Example 1: matrix multiplication followed by integer-level permutation.

After multiplying a permutation $P(2,1)$ by Y_1, we get $Y_1' = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 5 & 11 \end{bmatrix} = \begin{bmatrix} 5 & 11 \\ 2 & 1 \end{bmatrix}$. Using the known plaintext-ciphertext pairs, an opponent sets a linear system of equations as follows:

$$Y_1 = K_1 X_1 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 12 & 3 \\ 5 & 4 \end{bmatrix} = \begin{bmatrix} 5 & 11 \\ 2 & 1 \end{bmatrix} mod\ 26$$

So he sets $\begin{Bmatrix} 12a + 5b = 5 \\ 3a + 4b = 11 \end{Bmatrix}$ mod 26, and $\begin{Bmatrix} 12c + 5d = 2 \\ 3c + 4d = 1 \end{Bmatrix}$ mod 26 and then solves it and gets $K_1 = \begin{bmatrix} 21 & 13 \\ 19 & 12 \end{bmatrix}$ $mod\ 26$.

If a new plaintext is X_2, then the permuted ciphertext after matrix multiplication will be $Y_2' = \begin{bmatrix} 6 & 3 \\ 12 & 22 \end{bmatrix}$ and $K_1^{-1}Y_2' = X_2$ where the inverse of key is $K_1^{-1} = \begin{bmatrix} 18 & 13 \\ 17 & 25 \end{bmatrix}$ as it is shown below:

$$X_2 = {K_1}^{-1}Y_2' = \begin{bmatrix} 18 & 13 \\ 17 & 25 \end{bmatrix}\begin{bmatrix} 6 & 3 \\ 12 & 22 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 12 & 3 \end{bmatrix} mod\ 26$$

Hence, opponent has successfully broken our cipher and the original message is exposed.

Example 2: matrix multiplication followed by bit-level permutation. It shows that revealed plaintext is quite different from the original one.

Let $Y_1'' = \begin{bmatrix} 2 & 1 \\ 3 & 11 \end{bmatrix}$ be a result of bit permutation by swapping the two bits (b_2, b_1) of the $Y_{1\,i,j} = b_4b_3b_2b_1b_0\ ,\ where\ i = 2, j = 1$ i.e. applied permutation is $P(43120)$ out of five bits. So the key can be revealed by opponent after making a linear system and solving the related equations as $K_2 = \begin{bmatrix} 19 & 12 \\ 10 & 13 \end{bmatrix}\ mod\ 26$.

The permutation steps to get Y_1'' are shown as below:

$$Y_1'' = KX_1 = \begin{bmatrix} 19 & 12 \\ 21 & 13 \end{bmatrix}\begin{bmatrix} 12 & 3 \\ 5 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 5 & 11 \end{bmatrix} = \begin{bmatrix} 00010 & 00001 \\ 00101 & 01011 \end{bmatrix} = \begin{bmatrix} 00010 & 00001 \\ 00011 & 01011 \end{bmatrix}$$
$$= \begin{bmatrix} 2 & 1 \\ 3 & 11 \end{bmatrix}$$

If X_2 is a new plaintext, then $Y_2'' = \begin{bmatrix} 12 & 22 \\ 6 & 3 \end{bmatrix}$ will be permuted ciphertext. Here, $K_2^{-1}Y_2'' \neq X_2$, where $K_2^{-1} = \begin{bmatrix} 13 & 4 \\ 12 & 11 \end{bmatrix}$. It is shown below:

$$X_2 = K_2^{-1}Y_2'' = \begin{bmatrix} 13 & 4 \\ 12 & 11 \end{bmatrix} \begin{bmatrix} 12 & 22 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 24 & 12 \\ 2 & 11 \end{bmatrix} mod\ 26$$

From the example 1 and 2, it is shown that bit-level permutation provides non-linear transformation.

In the following section, some theorems which state remarkable definitions about the basic concepts of linear transformation are characterized, and theorems whose proof involve the use of both integer-level and bit-level permutations of matrix transformation. It is demonstrated that any integer-level permutation can be broken due to its linearity and bit-level permutation does not satisfy the linear transformation properties.

3.3 Linearity of Integer-Level Permutation

In order to prove the existence of linearity of the integer-level permutations P i.e. (3.1) is true, let's show some examples. Consider a vector A as a sequence of numbers or elements:

$$A = \{a1, a2, \dots, an\}$$

Permutation P on A is given by:

$$P(A)_k = A_i \qquad (3.2)$$

$i \rightarrow k$, where i maps to k

where A_i and A_k are each elements of A. Permutations re-arrange the elements of the vector A. Bijection of a permutation ensures that it has an inverse that returns each element to its original location. [11]

Let us apply (3.2) on the property of linear transformation (3.1).

$$[P(k_1A_1 + k_2A)]_k = [k_1.P(A_1) + k_2.P(A_2)]_k$$

Where, P indicates integer permutation and k_1 , A_1, k_2, A_2 are scalars. From (3.1), we get:

$$P(k_1A_1 + k_2A_2)_k = k_1.P(A_1)_k + k_2.P(A_2)_k$$

Now let's prove that right side and left side produce the same result.

$$(k_1A_1 + k_2A_2)_i = k_1.(A_1)_i + k_2.(A_2)_i$$

Then

$$k_1A_{1i} + k_2A_{2i} = k_1.(A_1)_i + k_2.(A_2)_i$$

Hence any compositions of matrix transformation which have integer-level permutations make linear transformation.

Example 1:

Consider a set $A = \{a, b, c, d\}$ with 4-elements. The matrix P given by

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

is a permutation matrix that maps the elements '$abcd$' to '$dcba$', i.e. swaps all elements as required. In vector notation, $P(a, b, c, d) = (dcba)$.

3.4 Non Linearity of Bit-Level Permutation

Let us prove that binary permutation P is non-linear transformation i.e. (3.1) is not true for bit-level permutation as a type of transformation.

Given an expression as:

$$P(k_1 p_1 + k_2 p_2) = k_1 . P(p_1) + k_2 . P(p_2) \tag{3.2}$$

Where, P indicates permutation and k_1 , p_1, k_2, p_2 are 2-bit ASCII characters. From $X = (2x1 + x0)$ that is binary representation of a number, left side of (3.2) can be written as follows:

$$P((2k_{11} + k_{10})(2p_{11} + p_{10}) + (2k_{21} + k_{20})(2p_{21} + p_{20})\ mod\ 4)$$

$$= P(2(k_{11}p_{10} + k_{10}p_{11} + k_{21}p_{20} + k_{20}p_{21}) + k_{10}p_{10} + k_{20}p_{20})$$

$$= P(2c_1 + c_0)$$

As $(2c_1 + c_0) = 2c_0 + c_1$, so

$$c_0 = k_{11}p_{10} \oplus k_{10}p_{11} \oplus k_{21}p_{20} \oplus k_{20}p_{21} \oplus k_{10}p_{10}.k_{20}p_{20} \, ,$$

$$c_1 = k_{10}p_{10} \oplus k_{20}p_{20} \tag{3.3}$$

Right side can be written as:

$$\begin{aligned}(2k_{11} + k_{10})(2p_{10} + p_{11}) + (2k_{21} + k_{20})(2p_{20} + p_{21}) \ mod \ 4 \\ = 2(k_{11}p_{11} + k_{10}p_{10} + k_{21}p_{21} + k_{20}p_{20}) + k_{10}p_{11} + k_{20}p_{21} \\ = (2c_1 + c_0)\end{aligned}$$

So we have:

$$c_0 = k_{10}p_{11} \oplus k_{20}p_{21},$$

$$c_1 = k_{11}p_{11} \oplus k_{10}p_{10} \oplus k_{21}p_{21} \oplus k_{20}p_{20} \oplus k_{10}p_{11}.k_{20}p_{21} \tag{3.4}$$

From (3.3) and (3.4) we observe that left side and right side do not produce the same result. i.e. (3.2) does not hold linear property (3.1). Hence any compositions of matrix transformation which have bit-level permutations make non-linear transformation.

Many ciphers use a combination of two integer and bit level permutation to provide flexibility in the security. For instance, RC5 which is a symmetric block cipher uses both level permutations. It consists of one addition of elements and bit-wise explosive-OR and a left rotation of elements. [12]

Chapter 4

INVESTIGATION OF TWO RECENT HILL CIPHER MODIFICATIONS

In [7, 8], Hill cipher is modified by including interweaving, interlacing and iteration. Ciphers proposed in [7, 8] have avalanche effect and are supposed to resist any cryptanalytic attack.

In this chapter we investigate impact of number of iterations on the avalanche effect, and propose generalizations of the ciphers from [7, 8].

A desirable feature of any encipher algorithm is that a small change in either the plaintext or the key should produce a considerable change in the whole ciphertext i.e. changing one bit of the plaintext or one bit of the key should produce a change in a lots of bit of the ciphertext.

4.1 Investigation of the Number of Iterations in Two Recent Ciphers

In the proposed ciphers $N = 16$ iterations are used to ensure the security and provide a good avalanche effect. But it is not discussed why the number of iterations is taken to be 16. We examine avalanche effect of these ciphers using examples from [7, 8].

Plaintext is given by:

"The World Bank h", (4.1)

and key by (4.2) from [8],

$$K1 = \begin{bmatrix} 53 & 62 & 24 & 33 & 49 & 18 & 17 & 43 \\ 45 & 12 & 63 & 29 & 60 & 35 & 58 & 11 \\ 8 & 41 & 46 & 30 & 48 & 32 & 5 & 51 \\ 47 & 9 & 38 & 42 & 2 & 59 & 27 & 61 \\ 57 & 20 & 6 & 31 & 16 & 26 & 22 & 25 \\ 56 & 37 & 13 & 52 & 3 & 54 & 15 & 21 \\ 36 & 40 & 44 & 10 & 19 & 39 & 55 & 4 \\ 14 & 1 & 23 & 50 & 34 & 0 & 7 & 28 \end{bmatrix}, \quad (4.2)$$

and plaintext:

"The development", (4.3)

and key (4.4) from [8], for different number of iterations.

$$K2 = \begin{bmatrix} 53 & 62 & 124 & 33 & 49 & 118 & 107 & 43 \\ 45 & 112 & 63 & 29 & 60 & 35 & 58 & 11 \\ 88 & 41 & 46 & 30 & 48 & 32 & 105 & 51 \\ 47 & 99 & 38 & 42 & 112 & 59 & 27 & 61 \\ 57 & 20 & 6 & 31 & 106 & 126 & 22 & 125 \\ 56 & 37 & 113 & 52 & 3 & 54 & 105 & 21 \\ 36 & 40 & 43 & 100 & 119 & 39 & 55 & 94 \\ 14 & 81 & 23 & 50 & 34 & 70 & 7 & 28 \end{bmatrix}. \quad (4.4)$$

There are some mistakes in the example [8] illustrating the avalanche effect. The plaintext (4.3) in ASCII code shall have letter "l" represented by 108 that in [8, page 212, equation (1)] is shown as 109. Correct ASCII code representation for (4.3) is given in (4.5).

$$P = \begin{bmatrix} 84 & 108 \\ 104 & 111 \\ 101 & 112 \\ 32 & 109 \\ 100 & 101 \\ 101 & 110 \\ 118 & 116 \\ 101 & 32 \end{bmatrix}. \tag{4.5}$$

Result of multiplication of ASCII code representation [8, page 212, equation (1)] of plaintext (4.3) is not correct in row numbers 4 and 7 that is shown in [8, page 212,] as:

$$\begin{bmatrix} 27 & 112 \\ 17 & 83 \\ 83 & 113 \\ 108 & 41 \\ 37 & 25 \\ 38 & 86 \\ 59 & 61 \\ 127 & 11 \end{bmatrix}.$$

Correct result taking into account (4.5) is given by:

$$P^1 = \begin{bmatrix} 27 & 112 \\ 17 & 83 \\ 83 & 113 \\ 34 & 73 \\ 37 & 25 \\ 38 & 86 \\ 86 & 77 \\ 127 & 11 \end{bmatrix}. \tag{4.6}$$

Result b' of performing Interweave process on the binary matrix [8, page 213] b given by (4.7)

$$b = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \qquad (4.7)$$

is not correctly shown in [8, page 213] given by (4.8):

$$b' = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}. \qquad (4.8)$$

Let us show that at the left top corner of binary matrix (4.8) is not correct. Let's consider left top corner 4×4 submatrix of (4.7):

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

After applying 1st and 3rd columns upward circular shift, and then 2nd and 4th rows left shift we get:

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & b'_{2,5} \\ 1 & 0 & 0 & 0 \\ 1 & b'_{5,3} & 1 & b'_{5,5} \end{bmatrix}$$

that differs from left top corner of (4.8). We get the correct result of performing interweave on the correct plaintext (4.6) given by (4.9):

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \tag{4.9}$$

After converting these binary bits into decimal numbers we get:

$$P^1 = \begin{bmatrix} 27 & 82 \\ 35 & 99 \\ 2 & 89 \\ 79 & 18 \\ 36 & 19 \\ 109 & 57 \\ 87 & 79 \\ 118 & 67 \end{bmatrix}, \tag{4.10}$$

instead of constructed in [8, page 213] given by:

$$\begin{bmatrix} 35 & 98 \\ 19 & 83 \\ 114 & 67 \\ 100 & 57 \\ 79 & 56 \\ 46 & 23 \\ 118 & 82 \\ 95 & 90 \end{bmatrix}$$

After carrying out all 16 iterations, corrected ciphertext is obtained in the form:

$$C = \begin{bmatrix} 99 & 101 \\ 109 & 70 \\ 41 & 43 \\ 48 & 37 \\ 0 & 95 \\ 95 & 109 \\ 44 & 77 \\ 65 & 127 \end{bmatrix}$$

which is given not correctly in [8, page 213] by:

$$\begin{bmatrix} 114 & 8 \\ 100 & 65 \\ 56 & 81 \\ 71 & 24 \\ 8 & 81 \\ 37 & 4 \\ 0 & 73 \\ 117 & 99 \end{bmatrix}$$

In terms of binary bits, plaintext (4.5) can be writhen as:

1010100	1101100
1101000	1101111
1100101	1110000
0100000	1101101
1100100	1100101
1100101	1101110
1110110	1110100
1100101	0100000

(4.11)

On changing the 9th character of the plaintext *P* (4.5) from “l“ to “m”, we get:

1010100	1101101
1101000	1101111
1100101	1110000
0100000	1101101
1100100	1100101
1100101	1101110
1110110	1110100
1100101	0100000

(4.12)

Obtained ciphertexts by [8] corresponding to (4.11) and (4.12) are given respectively as:

1100011	1100101
1101101	1000110
0101001	0101011
0110000	0100101
0000000	1011111
1011111	1101101
0101100	1001101
1000001	1111111

(4.13)

and

0100000	0100011
1011110	1111010
0111000	0000110
0101010	1110110
0100110	0010110
0010111	1101101
0111100	1001011
0101011	1111110

(4.14)

Here, plaintexts differ exactly by one bit but ciphertexts differ by 43 bits after performing 16 iterations.

Table 1 shows comparison results that are obtained for the ciphers [7, 8] by changing first character of the plaintext (4.1) from "T" to "U", and the 9th character of the plaintext (4.4) from "l" to "m" for different number of iterations ranging from 1 to 100. We also change the key (4.2) element $K1_{3,3}$ from 46 to 47 and the key (4.3) element $K2_{3,6}$ from 32 to 33.

Table 1: Avalanche effect investigation for recent ciphers

Iteration number	Change in plain text		Change in key	
	Number of bits that differ		Number of bits that differ	
	[7]	[8]	[7]	[8]
1	56	64	30	51
2	52	59	55	61
3	53	54	57	59
4	56	53	58	55
5	53	40	56	56
6	62	61	58	56
7	57	59	59	48
8	61	54	62	61
9	44	63	61	62
10	62	62	47	60
11	53	64	51	54

12	56	60	60	56
13	57	50	49	66
14	52	54	57	64
15	60	62	61	57
16	65	43	55	57
17	51	60	66	56
18	51	60	53	62
19	68	53	62	50
20	59	59	57	53
50	58	63	56	49
100	59	53	58	61

From Table 1 we can see that for all numbers of iteration avalanche effect is approximately the same. Hence, number 16 of iteration is not a distinguished one and less number of iteration may be used instead. In the next chapter we propose two modifications by including bit-level permutation and $N = 2$ iterations.

Chapter 5

PROPOSED CIPHERS AND THEIR INVESTIGATION

In this chapter, we modify ciphers [7, 8] by introducing a new cipher, columns swapping HC (CSHC). It uses swapping of columns of the binary bits of the plaintext characters instead of interlacing and interweaving as in [7, 8]. Another modification, arbitrary permutation hill cipher (APHC), uses an arbitrary permutation not known to an opponent instead of a fixed permutation (interweaving, interlacing and etc.).

Permutation is shared between sender and receiver as the key is shared. Permutation is a vector of the same length as plaintext matrix (i.e. $L = n \times 14$) with integer components from $\{1, \dots, L\}$. All values from $1, \dots, L$ are represented in Permutation in some order. For example, if $L = 4$, and Permutation = (4132) then applying Permutation to B = $(b_0 b_1 b_2 b_3)$, we get $(b_3 b_0 b_2 b_1)$. In the proposed ciphers $N = 1, 2$ iterations are used instead of $N = 16$ iterations for [7, 8].

We now introduce the concept of proposed ciphers. Inputs are as follows for both ciphers:

1. Let a plaintext Pconsist of 2n 7-bit ASCII characters and is represented in the form of a matrix:

$$P = [P_{ij}], i = 1\ to\ n, j = 1\ to\ 2$$

2. A key K and its inverse K^{-1} are represented as:

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

$$K^{-1} = [K'_{ij}], i = 1\ to\ n, j = 1\ to\ n$$

where $K.K^{-1} = I$ and shared by both communications parties. Let $C = [C_{ij}], i = 1\ to\ n,\ j = 1\ to\ 2$ be the corresponding ciphertext matrix.

Algorithm for encryption by CSHC and APHC:

1. $P^0 = P$
2. For $i = 1, \dots, N$ where $N \in \{1, 2\}$. do the following:

2.1. Compute $P^i = KP^{i-1}\ mod\ 128$

2.2. $P^i =$ columns swapping (P^i) for CSHC, or $P^i =$ Permute (Permutation, P^i) for APHC.

3. If $AD = True$ then $C = KP^N\ mod\ 128$, otherwise $C = P^N$.

Algorithm for decryption by CSHC and APHC:

1. If $AD = True$ then $P = K^{-1}C \; mod \; 128$, otherwise $P = C$.
2. For $i = N, \ldots, 1$ where $N \in \{1, 2\}$. do the following:

2.1. P^i = columns swapping (P^i) as use in CSHC, or $P^i =$ IPermute (Permutation, P^i) as use in APHC.

2.2. $P^{i-1} = K^{-1}P^i \; mod \; 128$

3. $P = P^0$

Algorithm for columns swapping (P):

1. Convert P into two binary matrices as:

$$\mathrm{E} = \begin{bmatrix} e_{1,1} & \cdots & \mathrm{e}_{1,7} \\ \vdots & & \vdots \\ e_{n,1} & \cdots & \mathrm{e}_{n,7} \end{bmatrix}, F = \begin{bmatrix} f_{1,1} & \cdots & f_{1,7} \\ \vdots & & \vdots \\ f_{n,1} & \cdots & f_{n,7} \end{bmatrix}$$

2. Swap the jth column of E by jth column of F where $j = 2, 4, 6$ as:

$$E' = \begin{bmatrix} e_{11} & f_{12} & e_{13} & f_{14} & e_{15} & f_{16} & e_{17} \\ e_{21} & f_{22} & e_{23} & f_{24} & e_{25} & f_{26} & e_{27} \\ e_{31} & f_{32} & e_{33} & f_{34} & e_{35} & f_{36} & e_{37} \\ e_{41} & f_{42} & e_{43} & f_{44} & e_{45} & f_{46} & e_{47} \\ e_{51} & f_{52} & e_{53} & f_{54} & e_{55} & f_{56} & e_{57} \\ e_{61} & f_{62} & e_{63} & f_{64} & e_{65} & f_{66} & e_{67} \\ e_{71} & f_{72} & e_{73} & f_{74} & e_{75} & f_{76} & e_{77} \\ e_{81} & f_{82} & e_{83} & f_{84} & e_{85} & f_{86} & e_{87} \end{bmatrix}$$

$$F' = \begin{bmatrix} f_{11} & e_{12} & f_{13} & e_{14} & f_{15} & e_{16} & f_{17} \\ f_{21} & e_{22} & f_{2,3} & e_{24} & f_{25} & e_{26} & f_{27} \\ f_{31} & e_{32} & f_{33} & e_{34} & f_{35} & e_{36} & f_{37} \\ f_{41} & e_{42} & f_{43} & e_{44} & f_{45} & e_{46} & f_{47} \\ f_{51} & e_{52} & f_{53} & e_{54} & f_{55} & e_{56} & f_{57} \\ f_{61} & e_{62} & f_{63} & e_{64} & f_{65} & e_{66} & f_{67} \\ f_{71} & e_{72} & f_{73} & e_{74} & f_{75} & e_{76} & f_{77} \\ f_{81} & e_{82} & f_{83} & e_{84} & f_{85} & e_{86} & f_{87} \end{bmatrix}$$

3. Construct $P_{j,1}$ from $\mathrm{E}'_{j,1:7}$ and $P_{j,2}$ from $\mathrm{F}'_{j,1:7}$, $j = 1\ to\ n$

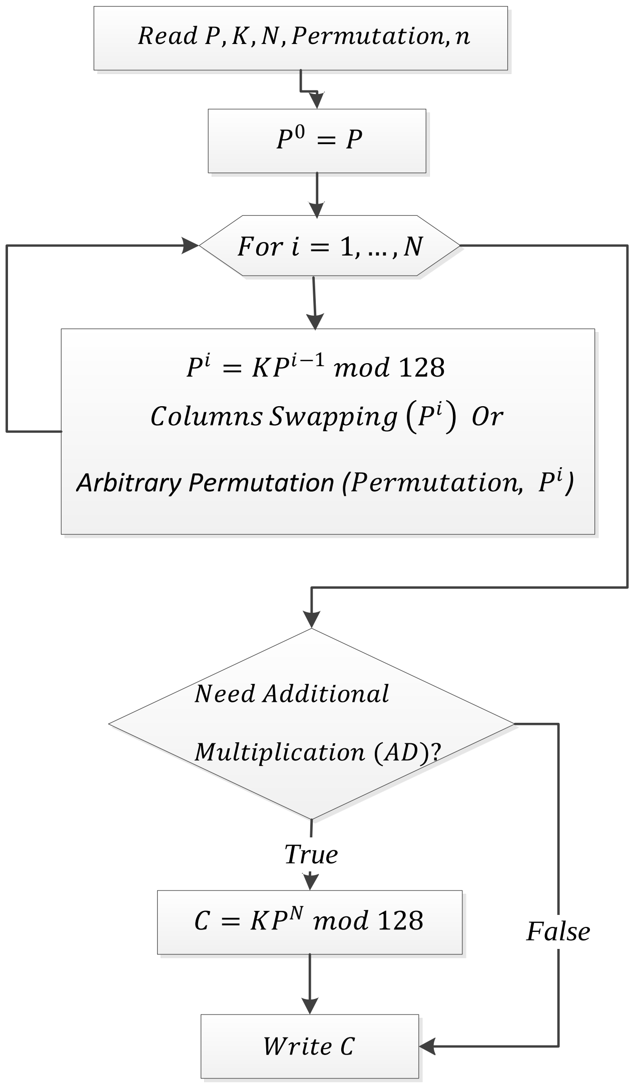
Algorithm for APHC (Permutation, P):

1. Convert P into a binary $n \times 14$ matrix:

$$B = \begin{bmatrix} b_{1,1} & \dots & b_{1,14} \\ \vdots & & \vdots \\ b_{n,1} & \dots & b_{n,14} \end{bmatrix}$$

2. Apply Permutation vector (i.e. size $L = n \times 14$) to the bits of B of size $n \times 14$ that is considered as a row-vector $v = (v_1, v_2, \dots v_{n\times 14})$.
3. Construct P from B using first 7 bits of jth row for $P_{j,1}$, and last 7 bits for $P_{j,2}$, $j = 1, 2, \dots, n$.

The process of the CSHC and APHC are given in Fig 1.



$Read\ P, K, N, Permutation, n$
$P^0 = P$
$For\ i = 1, \dots, N$
$P^i = KP^{i-1}\ mod\ 128$
$Columns\ Swapping\ (P^i)\ Or$
Arbitrary Permutation $(Permutation,\ P^i)$
$Need\ Additional\ Multiplication\ (AD)?$
$True$
$False$
$C = KP^N\ mod\ 128$
$Write\ C$

Figure 1: Schematic diagram of the CSHC and APHC. (N denotes the number of iterations, recommended values: N=1, 2)

To illustrate the CSHC, let's consider to the plaintext given in (4.3) and the key given in (4.4) where $n = 8$. Here, the key elements are less than 128 and each number can be represented in terms of 6-bit binary. So, the size of the key matrix is $6 \times 64 = 384$ binary bits. The ASCII code matrix corresponding to plaintext is given in (4.5). On multiplying plaintext matrix by the key we get (4.6). After dividing it into two binary matrices, we get:

$$e = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$f = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Now, we illustrate the process of our permutation.

$$e' = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\mathrm{f}' = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Transformed plaintext, after the first iteration is as follows:

$$\mathrm{P}^1 = \begin{bmatrix} 49 & 90 \\ 19 & 81 \\ 113 & 83 \\ 8 & 99 \\ 13 & 49 \\ 6 & 118 \\ 92 & 71 \\ 95 & 43 \end{bmatrix}.$$

To illustrate APHC let's $b_6b_5b_4b_3b_0b_2b_1$ be a result of 3-bits permutation by swapping three bits b_2, b_1 and b_0 out of the 7-bits ASCII code binary representation by $b_6b_5b_4b_3b_2b_1b_0$. On multiplying plaintext in (4.3) by the key in (4.4) we get (4.6). After converting it into a binary matrix, we get:

$$\mathrm{e} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

The process of APHC after performing (6543021) permutation on the e_{ij} where $i = 0$, $j = 0$ is:

$$e' = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Plaintext matrix, after the first iteration is as follows:

$$P^1 = \begin{bmatrix} 29 & 112 \\ 17 & 83 \\ 83 & 113 \\ 108 & 41 \\ 37 & 25 \\ 38 & 86 \\ 59 & 61 \\ 127 & 11 \end{bmatrix}.$$

5.1 Statistical Analysis of the Proposed CSHC and APHC

To test the strength of the CSHC and APHC we examine both changing in the plaintext and key. Table 2 shows the avalanche effect of CSHC when changing first character of the plaintext (4.3) from “T” to “U” which differs by one bit, then changing second character from “h” to “i” and so on where $N = 1, 2$ and additional multiplication $AD = True$. We also change the key (4.4) element $K2_{3,6}$ from 32 to 33.

Table 2: Avalanche effect of CSHC where AD= True and N=1, 2

Plaintext Characters	**Original key**		**Changed key**	
	$\mathbf{N=1,}$ $\mathbf{AD=True}$	$\mathbf{N=2,}$ $\mathbf{AD=True}$	$\mathbf{N=1,}$ $\mathbf{AD=True}$	$\mathbf{N=2,}$ $\mathbf{AD=True}$
“T” to “U”	44	44	46	64
“h” to “i”	42	55	60	61
“e” to “f”	40	55	59	49
“d” to “e”	60	60	61	66
“e” to “f”	56	45	51	61
“v” to “w”	55	50	55	52
“e” to “f”	56	45	49	49
“l” to “m”	51	51	56	62
“o” to “p”	48	51	56	56
“p” to “q”	49	47	64	65
“m” to “n”	51	58	53	57
“e” to “f”	47	54	60	57
“n” to “o”	53	44	65	55
“t” to “u”	45	42	63	50

From Table 2, we can see that after iterations where $N = 1, 2$ avalanche effect is more or less the same. Hence, one iteration can be sufficient i.e. $N = 1$.

More results are produced here to show differentiate between effects of number of swapped bits in APHC where the number of swapped bits starts from $2, \ldots, 7$. Since, it is unknown to opponent 1 or 2 iterations are sufficient. The results are obtained to compare with proposed ciphers in [7, 8]. Table 3 shows the avalanche effect of different arbitrary permutations that swap only m bits in the plaintext (4.3) where $m = 2, . . , 7$. Results are obtained from performing 17 APHCs by only 1 iteration without no additional multiplications AD, i.e. $N = 1, AD = False$. The average of results is included in the table.

Table 3: Avalanche effect of APHC where N=1 and AD=False

2-bits	**3-bits**	**4-bits**	**5-bits**	**6-bits**	**7-bits**
35	35	33	35	41	39
37	35	39	39	41	39
37	35	37	39	41	39
39	37	33	37	39	39
35	37	33	35	35	39
37	35	35	37	35	35
35	35	37	37	35	39
37	33	35	37	35	39
35	33	35	37	35	35
39	37	37	37	35	37
35	37	37	37	35	37
39	35	37	37	33	35
37	37	37	37	33	31
37	37	37	37	33	35
37	37	37	37	39	35
37	37	37	37	41	35
35	37	37	39	41	35
36.65	35.82	36.06	37.12	36.88	36.65

We have tested APHC by different number of iteration and conditions. Table 4 included avalanche effect of 1 iteration with additional multiplication and Table 5 reports the avalanche effect of 2 iterations with additional multiplication respectively.

Table 4: Avalanche effect of APHC where N=1 and AD=True

2-bits	**3-bits**	**4-bits**	**5-bits**	**6-bits**	**7-bits**
51	64	28	47	49	28
45	58	48	28	28	56
45	55	45	47	54	28
44	28	42	34	28	59
28	58	45	30	55	28
42	28	51	28	27	51
44	58	38	50	61	28
49	32	43	28	28	35
41	31	28	44	62	26
46	28	40	49	47	46
56	28	47	47	51	59
32	41	49	44	28	54
56	41	43	27	28	41
42	56	28	29	28	52
44	55	47	30	55	50
52	55	30	50	54	55
51	58	26	50	48	56
45.18	45.53	39.88	38.94	43.00	44.24

Table 5: Avalanche effect of APHC where N=2 and AD=True

2-bits	**3-bits**	**4-bits**	**5-bits**	**6-bits**	**7-bits**
50	50	48	51	47	49
53	49	48	48	48	48
48	56	52	54	48	51
45	54	47	46	45	49
50	51	49	49	50	48
53	50	52	50	50	48
50	45	49	51	52	48
48	52	47	48	48	45
49	48	47	48	47	54
55	55	51	61	48	48
50	47	52	54	51	51
47	49	49	49	48	54
54	54	51	50	48	48
47	57	54	52	48	51
50	47	51	54	47	47
49	55	54	48	54	48
54	45	47	56	42	50
50.125	50.875	49.88	51.125	48.375	49.25

We have performed our experimental results by changing the key (4.4) element $K2_{3,6}$ from 32 to 33 and 1 or 2 iterations. Table 6 shows the avalanche effect of using 1 iteration without additional multiplication Ad.

Table 6: Avalanche effect of APHC by changing key element where N=1 and AD=False

2-bits	**3-bits**	**4-bits**	**5-bits**	**6-bits**	**7-bits**
9	9	17	10	9	9
2	11	9	14	17	13
11	9	13	10	9	9
11	11	9	16	17	19
11	9	13	10	9	9
9	11	9	16	9	21
9	11	15	10	9	13
9	11	9	16	7	9
9	9	13	9	9	7
9	11	9	13	13	13
9	11	9	10	9	19
11	11	9	8	13	19
9	13	11	9	9	15
11	9	9	7	11	19
11	13	17	10	9	21
9	9	9	10	15	17
11	13	13	10	9	21
9.41	10.65	11.35	11.06	10.76	14.88

Table 7 and Table 8 respectively show the avalanche effect of APHC by changing the key element where 1 and 2 iterations are used with additional multiplication.

Table 7: Avalanche effect of APHC by changing key element where N=1 and AD=True

2-bits	**3-bits**	**4-bits**	**5-bits**	**6-bits**	**7-bits**
58	58	58	53	68	49
58	57	58	55	52	51
58	56	58	56	62	48
50	54	54	53	58	49
54	57	59	56	68	59
60	52	54	61	66	52
62	58	58	57	68	59
62	58	58	58	60	60
59	58	53	53	65	49
59	56	59	58	55	70
59	56	50	58	57	56
53	58	53	51	62	50
56	58	53	51	52	63
59	58	49	53	55	61
58	58	52	55	56	64
58	55	58	56	65	49
56	58	55	53	60	51
57.59	56.76	55.24	55.12	60.53	55.29

Table 8: Avalanche effect of APHC by changing key element where N=2 and AD=True

2-bits	**3-bits**	**4-bits**	**5-bits**	**6-bits**	**7-bits**
54	59	59	65	58	58
56	53	59	67	60	59
58	60	59	68	58	58
58	60	57	62	53	55
56	63	57	62	58	51
57	66	57	56	53	61
53	52	59	65	53	53
62	59	68	67	57	58
62	59	64	67	55	62
59	69	66	71	58	57
60	64	72	50	58	60
60	62	66	65	55	56
61	59	67	60	55	53
61	52	72	63	56	61
63	64	67	63	47	53
61	62	59	63	57	44
60	59	66	60	53	62
58.88	60.12	63.18	63.18	55.53	56.53

Table 10 shows the average of the avalanche effect investigations for APHC that swaps selected m bits of the plaintext (4.3) to determine how changing bits provide avalanche effect where $m = 2, \dots ,7$. Table 3 displays the number of swapped bits with its applied permutation order to the plaintext characters $P_{j,1}$ or $P_{j,2}$ that is represented as 7-bit binary $b_6b_5b_4b_3b_2b_1b_0$ where $j = 1, 2, \dots ,8$.

Table 9: Permutation and bit locations

m	**Permutation**	**Element indices**
2	6453210	p_{11} , p_{31}
3	6543021	p_{32} , p_{71}
4	6234510	p_{72} , p_{12}
5	2345160	p_{72} , p_{52}
6	1234560	p_{62} , p_{41}
7	0123456	p_{32} , p_{12}

We have carried out APHC process for both plaintext (4.3) by changing “T” to ”U” and key (4.4) by changing element $K2_{3,6}$ from 32 to 33 by performing iteration and additional multiplications AD i.e. $N = 1, 2$ and $AD = True \,||\, False$.

Table 10: Avalanche effect average of APHC where additional multiplication AD= True || False and N=1, 2

m	**Change in plaintext**			**Change in key**		
	$\mathbf{N = 1, AD = False}$	$\mathbf{N = 1, AD = True}$	$\mathbf{N = 2, AD = True}$	$\mathbf{N = 1, AD = False}$	$\mathbf{N = 1, AD = True}$	$\mathbf{N = 2, AD = True}$
2	36.6	45.1	50.1	9.41	57.5	58.8
3	35.8	45.5	50.8	10.6	56.7	60.1
4	36.0	39.8	49.8	11.3	55.2	63.1
5	37.1	38.0	51.1	11.0	55.1	63.1
6	36.8	43.0	48.3	10.7	60.5	55.5
7	36.6	44.2	49.2	14.8	55.2	56.5

We have seen that a small change of the plaintext or key bits results in changing approximately half of the output bits. From Table 10, we found that even any simple bit-level permutations can provide a substantial avalanche effect and it can give the same performance in terms of avalanche effect than other complicated permutations which have been used in the literature survey. Thus, if permutation is unknown applying $N = 2$ iterations are recommended due to non-linearity of the bit-level permutation. In the case of known permutation more iteration are expected to resist known plaintext-ciphertext attack.

Chapter 6

CONCLUSION

The focus of this study was to modify the Hill cipher which is susceptible to known plaintext-ciphertext attack due to its linearity. In this case using a bit-level permutation which made it overall non-linear matrix transformation seems to be necessary as well as matrix multiplication.

In this thesis we have generalized two Hill cipher modifications [7, 8] which use bit-level permutation and 16 iterations. In both cases, the Hill cipher has been made secure against the cryptanalytic attack.

The primary goal of this study was to show that the strength of the newly proposed Hill cipher modifications [7, 8] is due to bit-level permutation used in matrix transformation.

Secondary goal of this thesis was to decrease time consuming of the encryption while it provides a substantial avalanche effect. This is successfully provided using the one or two iterations. We found that for numbers of iterations from 1 to 100 avalanches effect are approximately the same. Hence, use of 16 iterations which is used in [7, 8] is not reasonable and less number of iteration may be used instead.

The main focus of this thesis was to modify the Hill cipher to provide a good avalanche effect and make it more secure as other modifications in [7, 8]. Therefore, we proposed two ciphers columns swapping cipher CSHC and arbitrary permutation cipher APHC by using a bit-level permutation and iteration where $N = 1, 2$. Some statistical tests for examining the CSHC and APHC are performed. The results obtained in this analysis indicate that any bit-level permutation can provide a substantial avalanche effect, i.e. a slight change in the plaintext or key effects on approximately half of the output bits.

REFERENCES

[1] C. GaryKessler, “An Overview of Cryptography”, 2011. *http://www.garykessler.net/library/crypto.html*

[2] R. Martinelli, H. Laboratories, “Encryption Algorithms and Permutation Matrices”, 2003. *http://www.haikulabs.com/encrmat4.htm*

[3] B. Forouzan, "Cryptography and Network Security", First Edition, McGraw Hill, 2006.

[4] W. Stallings, “Cryptography and Network Security Principles and Practices”, Prentice Hall, 2005.

[5] M. Eisenberg, “Hill Ciphers and Modular Linear Algebra”, mimeographed notes, University of Massachusetts, 1998.

[6] M. Amin, H. Diab and I.A. Ismail, "How to Repair the Hill Cipher", Journal of Zhejiang University Science, vol. 7, no. 12, pp. 2022-2030, 2006.

[7] V.U.K. Sastry and N. Ravi Shankar, "Modified Hill Cipher with Interlacing and Iteration", Journal of Computer Science, vol. 3, no. 11, pp. 854-859, 2007.

[8] N. Ravi Shankar, S. Durga Bhavani and V. Umakanta Sastry, "A Modified Hill Cipher Involving Interweaving and Iteration", International Journal of Network Security, vol. 10, no. 3, pp. 210-215, 2010.

[9] A. Varanasi, S.U daya Kumar and V.U.K. Sastry, "A Modified Hill Cipher Involving a Pair of Keys and a Permutation", International Journal of Computer and Network Security, vol. 2, no. 9, pp. 150-108, 2010.

[10] D. S. R. Murthy, S. Durga Bhavani and V. U. K. Sastry, "A Block Cipher Having a Key on One Side of the Plaintext Matrix and its Inverse on the Other Side", International Journal of Computer Theory and Engineering, vol. 2, no. 5, pp. 805-810, 2010.

[11] S. Lang, “Linear Algebra”, Chapter 2, Third Edition, Addison Wesley Publishing Company, 1987.

[12] E. Biham, A. Shamir, “A Differential Cryptanalysis of the Data Encryption Standard”, Springer, 1993.

[13] Wu. Tzong-chen, C. Yi Shiung Yeh, "A New Cryptosystem Using Matrix Transformation, IEEE International Carnahan Conference on Security Technology, 1991.