Implementation of Modified Playfair CBC Algorithm

Priyanka Goyal  
Dept. of Computer Science & Engineering  
ITM Group of Institutions  
Gwalior, India

Gaurav Sharma  
Dept. of Computer Science & Engineering  
ITM Group of Institutions  
Gwalior, India

Shivpratap Singh Kushwah  
Dept. of Computer Science & Engineering  
ITM Group of Institutions  
Gwalior, India

Abstract—These days, Information Security is the most summon aspects in the web and network application due to their swiftly origination. Therefore, secure exchanged data over the internet is vital. In this context, cryptography is used that converts information from its normal form into an unreadable form by using encryption techniques. This paper deals with a new solution approach to overcome the shortcomings of the Playfair algorithm. In this paper, the presented PlayfairCBC encryption mechanism makes the cryptanalysis complex. The encrypted text obtained is almost unreadable. The proposed PlayfairCBC algorithm is implemented and number of tests is performed to prove its efficiency. Finally, it has been analyzed on the basis of avalanche effect.

Keywords—Avalanche; Brute force; CBC; Cipher; Cryptanalysis; Encryption; Playfair

I. INTRODUCTION

Cryptography word comes from Greek which means “secret writing” [1]. It refers to the science and art of transforming messages in such a way that makes it secure and immune to attacks. Its fundamental objective is to allow two individuals to communicate in such a way that an adversary cannot understand what is being communicated. It is the study of mathematical techniques associated with information security. To achieve the security goals, cryptography involves three distinct mechanisms: symmetric key cryptosystem (sometimes called symmetric key cryptography, secret key cryptography or private key cryptography), asymmetric key cryptosystem (sometimes called asymmetric key cryptography or public key cryptography) and hash functions. Cryptology is the science of cryptography and cryptanalysis. Cryptanalysis, “code breaking”, is the science of attacking cryptographic systems and gain access to the entire data of encrypted messages, also when the cryptographic key is unknown.

II. LITERATURE SURVEY

In this section, we will discuss existing modified Playfair ciphers. Basu and Ray [2] proposed their modified 10 x 9 matrix playfair cipher which contains almost all the printable characters i.e. uppercase and lowercase alphabets, numbers, punctuation marks and special characters. In the variation projected by Packirisamy Murali and Gandhidoss Senthilkumar [3], the authors propose new rules which has numerous benefits as compare to traditional Playfair cipher. In the variation projected by Man et al. [4] and in the variation projected by Srivastava and Gupta [5], the 5×5 matrix has been replaced by 8×8 matrix. Srivastava et al. [6] thought of diagrams within the plaintext as single. Totally different variety of cipher attacks and non-vulnerability of recent cipher has been mentioned. A few contributions were also noticeable [7-11].

A Framework based on Probability analysis of Character occurrence [12] is anew approach which keeps track of the frequency of occurrences of each and every character in English language and substitutes the every next occurrence of the character with a character of least frequency of use. It the new word becomes a meaning word replace with the next character of least frequency. The modified algorithm is efficient than the original Playfair and can handle spaces, repetitive characters more efficiently but still lacks in the number of supported character set.

A DNA and Amino Acid-Based Implementation [13] modifies the Playfair cipher significantly by introducing DNA-based amino acid structures to the core of the ciphering process. The proposed work treats the plaintext as a binary stream. Each and every pair of bits of the binary stream is replaced with either A, C, G or T, which are the abbreviated forms of the four bases of DNA namely- Adenine, Cytosine, Guanine and Thymine respectively. The proposed algorithm is quite time consuming because of its lengthy procedure and requirement of multiple read / write operations. Additionally 8-bit ASCII is converted to codons or triplets of bit-pairs, so remaining offsets are to be taken care of. The proposed modification actually treats the plaintext file as binary data stream and thus enriches the character set and actually solves the problem of limited character support.

A. Lahiri [14] also proposed binary Playfair algorithm. It uses a reduced 4×4 key matrix to encrypt each bytes of the plaintext file. Moreover, Salman Khan [15] surveyed and analyzed numerous Playfair ciphers with different matrix sizes. These matrices are 9 x 9, 10 x 10, and 11 x 11. Kumar and Rana [16] propose 6X6 Polybius square which includes both the alphabets and numbers leads to secured communication. Further, Kumar et al. [17] proposes a 10X10 hybrid Polybius and Playfair cipher to hold the 95 printable characters of ASCII character set. Blum-Blum Shub generator is used to generate key matrix.
III. PROPOSED PLAYFAIRCBC ALGORITHM

In Playfair cipher, the alphabets are arranged in a key matrix of size 5*5 based on secret key. In our proposed PlayfairCBC algorithm, we can implement any matrix size. As its name suggests, our proposed algorithm uses output in a chain to construct the key matrix. Its overall architecture is as follows:

To fill in the key matrix table, the alphabets of the keyword (dropping any duplicate letters) are placed in serial order and the leftover spaces are filled with the remaining letters of the alphabet in a well ordered manner.

A. Encryption

To encrypt a message, the message is broken into digraphs (groups of 2 letters) and then mapped them out on the key table. Then following protocols are employed to each matching set of letters in the plaintext:

1. When both alphabets lie on the identical (and pair is left with one letter only), add an “X” after the first letter then encrypt the recently developed pair and proceed with.
2. When the alphabets repose on the same row of the table, they are to be interchanged with the letters immediate right of them respectively.
3. When the alphabets repose on the same column of the table, they are to be interchanged with the letters just below them respectively.
4. When the alphabets repose not on the same row or column, exchange them with the alphabets on the identical row respectively but of the column of the other keeping the order of the matching set unflawed.
5. After encryption of the first digraph, pick the output and reorganize the key matrix. It will be used to encrypt another digraph.

B. Decryption

To decrypt a message, the message is broken into digraphs (groups of 2 letters) and then mapped them out on the key table. Then following protocols are employed to each matching set of alphabet in the plaintext:

1. When the alphabets repose on the same row of the table, they are to be shuffled with the letters immediate left of them respectively.
2. When the alphabets repose on the same column of the table, they are to be interchanged with the letters immediately above them respectively.
3. When the alphabets does not repose on the same row or column, interchange them with the alphabets on the same row respectively but of the column of the other keeping the order of the matching set unflawed.
4. After decrypting first digraph, pick the first digraph and reorganize the key matrix. It will be used to decrypt another digraph.

IV. ILLUSTRATION OF THE PROPOSED PLAYFAIR CIPHER

In this section, our proposed Playfair cipher is demonstrates with the help of an example.

Enter the key (shared):

JAVAJ23$  

Key Matrix for 8*8 Playfair Cipher is

<table>
<thead>
<tr>
<th>J</th>
<th>A</th>
<th>V</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>S</th>
<th>!</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;</td>
<td>#</td>
<td>%</td>
<td>&amp;</td>
<td>'</td>
<td>(</td>
<td>)</td>
<td>*</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>.</td>
<td>/</td>
<td>0</td>
<td>4</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>:</td>
<td>&lt;</td>
<td>=</td>
<td></td>
</tr>
<tr>
<td>&gt;</td>
<td>?</td>
<td>@</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
<tr>
<td>G</td>
<td>H</td>
<td>I</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
</tr>
<tr>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>W</td>
<td>X</td>
</tr>
<tr>
<td>Y</td>
<td>Z</td>
<td>[</td>
<td>\</td>
<td>]</td>
<td>^</td>
<td>_</td>
<td>`</td>
</tr>
</tbody>
</table>

Consider the following plaintext.

MEET ME AT 05:00 PM TODAY

First, we have to separate it into di-grams.

ME ET ME AT 05:00 PM OD AY

Encrypt first di-gram
Now, the key matrix will change as per the ciphertext.

```
<table>
<thead>
<tr>
<th>N</th>
<th>D</th>
<th>J</th>
<th>A</th>
<th>V</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>*</td>
<td>!</td>
<td>&quot;</td>
<td>#</td>
<td>%</td>
<td>&amp;</td>
<td>'</td>
</tr>
<tr>
<td>@</td>
<td>+</td>
<td>,</td>
<td>-</td>
<td>/</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>:</td>
<td>;</td>
</tr>
<tr>
<td>&lt;</td>
<td>=</td>
<td>&gt;</td>
<td>?</td>
<td>@</td>
<td>B</td>
<td>C</td>
<td>E</td>
</tr>
<tr>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>O</td>
</tr>
<tr>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>W</td>
<td>X</td>
</tr>
<tr>
<td>Y</td>
<td>Z</td>
<td>[</td>
<td>\</td>
<td>]</td>
<td>^</td>
<td>_</td>
<td></td>
</tr>
</tbody>
</table>
```

Take another di-gram

```
<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>T</th>
<th>N</th>
<th>D</th>
<th>@</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>@</td>
<td>X</td>
<td>N</td>
<td>D</td>
<td>J</td>
</tr>
<tr>
<td>=</td>
<td>(</td>
<td>)</td>
<td>*</td>
<td>+</td>
<td>,</td>
<td>-</td>
</tr>
<tr>
<td>:</td>
<td>&lt;</td>
<td>=</td>
<td>&gt;</td>
<td>?</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>K</td>
<td>L</td>
<td>M</td>
</tr>
<tr>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>W</td>
</tr>
<tr>
<td>Y</td>
<td>Z</td>
<td>[</td>
<td>\</td>
<td>]</td>
<td>^</td>
<td>_</td>
</tr>
</tbody>
</table>
```

In this way, our algorithm will proceed.

**C. Benefits of Modified Playfair Cipher (PlayfairCBC)**

Unlike the traditional Playfair cipher the modified playfair cipher can handle all the printable characters. It includes uppercase and lowercase alphabets, numerical, punctuation marks and special characters. Both keyword and plaintext can contain one or more sentences.

**V. RESULT ANALYSIS AND DISCUSSION**

**A. Result Analysis based on Avalanche Effect**

This section presents output of the implementation of our proposed algorithm. The avalanche effect has been analyzed by altering only one character (1st character) in the plaintext.
Key Matrix for 8*8 Playfair Cipher is

```
J A V 1 2 3
# $ % 
+ . . . / 0 4
5 6 7 8 9 : ;
= > ? @ B C D E
F G H I K L M N
O P Q R S T U W
X Y Z \ \ \ \ _
```

KLFAIR CIPHER

After adding dummy character

KLFAIR CIPHERX

Encrypted text (CBC 8*8): LMM[

```
| M [ L A V | 1 2 |
| 3 ! " # $ % & ' ( ) + . - / 0 4 5 6 7 8 9 : ; |
| : < = > ? @ B C D E F G H I K N O P Q R S T U W |
| X Y Z \ \ \ \ _ |
```

Encrypted text (CBC 8*8): LMM[

```
| G J M [ L A V |
| 1 2 3 ! " # $ % & ' ( ) + . - / 0 4 5 6 7 8 9 : ; |
| : < = > ? @ B C D E F G H I K N O P Q R S T U W |
| X Y Z \ \ \ \ _ |
```

Encrypted text (CBC 8*8): LMM[

```
| F T G J M [ L A |
| 1 2 3 ! " # $ % & ' ( ) + . - / 0 4 5 6 7 8 9 : ; |
| : < = > ? @ B C D E F G H I K N O P Q R S T U W |
| X Y Z \ \ \ \ _ |
```

Encrypted text (CBC 8*8): LMM[

```
| C U 2 @ F T G J |
| M [ L A V 1 3 |
| ! " # $ % & ' ( ) + . - / 0 4 5 6 7 8 9 : ; |
| : < = > ? @ B C D E F G H I K N O P Q R S T U W |
| X Y Z \ \ \ \ _ |
```

Encrypted text (CBC 8*8): LMM[

```
| K ^ < H C U 2 @ F T |
| G J M [ L A V 1 3 |
| ! " # $ % & ' ( ) + . - / 0 4 5 6 7 8 9 : ; |
| : < = > ? @ B C D E F G H I K N O P Q R S T U W |
| X Y Z \ \ \ \ _ |
```

Encrypted text (CBC 8*8): LMM[

```
| S W X Y Z \ \ \ \ _ |
```

Encrypted text (CBC 8*8): LMM[

```
| F T G J M [ L A |
| 1 2 3 ! " # $ % & ' ( ) + . - / 0 4 5 6 7 8 9 : ; |
| : < = > ? @ B C D E F G H I K N O P Q R S T U W |
| X Y Z \ \ \ \ _ |
```

Encrypted text (CBC 8*8): LMM[

```
| S W X Y Z \ \ \ \ _ |
```

KLFAIR CIPHERX

Enter the key:

```
MEANS SECRET WRI
```

Actual Plaintext

CRYPTOGRAPHY A WORD WITH GREXEK ORIGIN MEANS SECRET WRITING.

Enter the key:

```
JAVA 123
```

Avalanche Effect in 8*8 PlayfairCBC is: 12

Binary value:

```
010111011110
```

Difference in bits = 29

Similarly, if we take longer plaintext then the result will be more fascinating. Consider another example.

Actual Plaintext

CRYPTOGRAPHY A WORD WITH GREXEK ORIGIN MEANS SECRET WRITING.

Enter the key:

```
JAVA 123
```
Key Matrix for 8*8 Playfair Cipher is

<table>
<thead>
<tr>
<th>J</th>
<th>A</th>
<th>V</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>!</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;</td>
<td>#</td>
<td>$</td>
<td>%</td>
<td>&amp;</td>
<td>’</td>
<td>( )</td>
</tr>
<tr>
<td>*</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>:</td>
<td>;</td>
<td>&lt;</td>
</tr>
<tr>
<td>=</td>
<td>&gt;</td>
<td>?</td>
<td>@</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>K</td>
<td>L</td>
<td>M</td>
</tr>
<tr>
<td>O</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
</tr>
<tr>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>[</td>
<td>\</td>
<td>^</td>
<td>_</td>
</tr>
</tbody>
</table>

For Avalanche Effect, Enter Text to be Encrypted (8*8 PlayfairCBC):

KRYPTOGRAPHY A WORD WITH GREXK ORIGIN MEANS SECRET WRITING.

Encrypted text (CBC 8*8): ISZOMVFT2KLX1 V AHGHIKAFH[@2DG1ZUM23BLB]

Avalanche Effect in 8*8 PlayfairCBC is: 56

Binary value:

```
01010010010011111010110101101011011011001
01100111000101001010000101110110101011001
1001111000011101001010000101000101001001
1111011000101101101110110110110111011001
101010110111001
```

Difference in bits = 201

It can be clearly seen that by the change of only 1 character in the input, the avalanche effect is 56 or 201 (49.507 %) in number of bits. If more number of characters is changed in the input then the output will be more different.

B. Result Analysis based on Ciphertext Only Attack

Our Proposed PlayfairCBC can also be implemented using 16 x 16 matrix also. To launch a ciphertext only attack, the number of di-grams the attacker has to search would be 256 x 256 i.e. 65536 in the modified cipher instead of 26 x 26 i.e. 676 di-grams in the traditional cipher.

C. Result Analysis based on Brute Force Attack

The worst case may involve traversing the complete search area. The size of key domain in the modified cipher would be 256! (Factorial 256) instead of 25! (Factorial 25) for the traditional Playfair cipher.

VI. CONCLUSION

The original Playfair cipher uses a digraph substitution technique to encrypt/decrypt alphabets based on a reference 5x5 key matrix, which is formed from the given key. The algorithm is strictly restricted to “English Alphabet”, that to either in uppercase or in lowercase character. No numbers, punctuations are supported. Several modification attempts focuses on elimination of several limitations. Some increases the character-set of the key matrix, some uses the ASCII values and others incorporated randomness. But these modifications stand strong in their own purposes. The overall limitations of the cipher were not eliminated by these individual modifications.

The proposed PlayfairCBC uses the encrypted digraph in order to re-construct the key matrix. It can be clearly seen that by the change of only 1 character in the input, the avalanche effect is 56 or 201 (49.507 %) in number of bits. If more number of characters is changed in the input then the output will be more different.
REFERENCES


