AN APPROACH OF HYBRID TRANSPOSITION METHOD BASED ON ASCII VALUE IN CRYPTOGRAPHY

VISHAL THAKOR, PAPRI GHOSH, PRAVIN BHATHAWALA

1Assistant Professor, Auro University, Gujarat, India
2Research Scholar, Pacific University, Rajasthan, India
3UGC Visiting Professor, HNG University, Gujarat, India
E-mail: 1vishalthakor@yahoo.co.in, 2papist_g@rediffmail.com, 3pcb1010@yahoo.com

Abstract – Every day millions of people interact electronically through different computer applications such as e-mail, e-commerce websites, ATM machines, or different mobile device applications and uses Internet either directly or indirectly to transfer the data over the networks. With perpetual growth of Internet usage, the demand for data security while transferring and storing the data has increased. This has further led to increase the importance of cryptography and its various techniques for security reasons.

In this paper, a hybrid cryptography method is proposed which uses both substitution and transposition techniques to secure the plain text. The proposed method first applies substitution with the help of three different keys followed by transposition using forth key consist of matrix size of equivalent ASCII values of plaintext. The forte of the method is its all four keys. Within the same manner decryption will be done at receiver's facet but in reverse way of encryption.

Keywords - Cryptography, Encryption, Decryption, Cryptanalysis, Symmetric key, Asymmetric key, Hybrid, Transposition

I. HISTORY OF CRYPTOGRAPHY

From ancient time, Human being had two natural needs: I) communication and sharing of information II) maintaining confidentiality of what they communicate[1]. These two fundamental needs gave birth to the art of coding the messages in such a way that only the targeted people could have access to the information and if message have been stolen by any unauthorized person, he/she could not extract any information. i.e., “Cryptography is the art and science of protecting information from undesirable individuals by converting it into a form not recognizable by its attackers while stored and transmitted”.[2] The word “Cryptography” comes from Greek word “Kryptos” which means hidden writing[3].

II. INTRODUCTION

Cryptography consists of two processes. 1) Encryption: is a process of converting plain text into unreadable text usually called cipher. 2) Decryption: is a process of converting cipher text back into original plaintext. In other words, it is the science of using mathematics to encrypt and decrypt information.

Cryptography in today’s digital world offers three core areas that protect your data from an unauthorized access of your data and fraud. They are Confidentiality, Integrity, and Availability[4].

Confidentiality: It assures that private or confidential information is not made available or disclosed to unauthorized individuals.

Integrity: It assures that information and programs are changed only in a specified and authorized manner.

Availability: It assures that systems work on time and service is not denied to authorized users.

Cryptography systems are characterized along three independent dimensions as follows [4]:

1) The type of operations used for transforming plaintext to cipher text.
   • All encryption algorithms are based on two general principles: Substitution, in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element, and Transposition, in which elements in the plaintext are rearranged.
   • The fundamental requirement is that no information be lost (i.e., that all operations are reversible).
2) The number of keys used.
   • If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption.
   • If the sender and receiver use different keys, the system is referred to as asymmetric, two-key, or public-key encryption.
3) The way in which the plaintext is processed.
   • A block cipher processes the input one block of elements at a time, producing an output block for each input block.
   • A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.

III. PROPOSED TECHNIQUE

There are many different basic methods for cryptography such as caesar cipher, monoalphabetic, polyalphabetic, playfair, rail fence, column transpose which us either substitution or transposition technics to convert the plain text into cipher. Our proposed
method is based on these fundamental methods and combines both the techniques, substitution and transposition, which is named as Hybrid Transposition technique. The traditional substitution method adds or subtracts some digits depending on the key from the original text character and generates the cipher accordingly[4]. Similarly, the classical transposition method comprise of a matrix and a key to interchange the columns. In order to add more security to the cipher, either the same key or another key can be applied for further interchange of columns which is called double transposition[5]. The proposed method first applies substitution with the help of three different keys followed by transposition using forth key consist of matrix size on equivalent ASCII values of plaintext. The forte of the method is its all four keys. Same process can be followed in reverse way to get the original text from the cipher text.

**Encryption:**

The idea behind this technique is to develop an algorithm with higher security which is difficult to break because of its keys strength. It uses four different keys to encrypt the text and same keys will be used by the receiver to get the original text. The process starts by reading the number of characters in provided plaintext. Once the characters are read and counted, a matrix size is decided to form a square matrix to accommodate the plaintext. Before storing characters into the matrix, each character is converted to its equivalent ASCII value and then placed row wise into the matrix. To understand these steps, consider the following example:

**Plaintext:** Hello Singapore

**Number of characters N (including space):** 15

**Matrix size (MxM):** 4x4, where M * M or (M²) ≥ N (i.e. 16 > 15)

**Equivalent ASCII:** 72 101 108 108 111 32 83 105 110 103 97 112 111 114 101

**Matrix:**

<table>
<thead>
<tr>
<th>72</th>
<th>101</th>
<th>108</th>
<th>108</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>32</td>
<td>83</td>
<td>105</td>
</tr>
<tr>
<td>110</td>
<td>103</td>
<td>97</td>
<td>112</td>
</tr>
<tr>
<td>111</td>
<td>114</td>
<td>101</td>
<td>42</td>
</tr>
</tbody>
</table>

First three keys key1, key2, and key3 are used for substitution purpose to get a new matrix. The maximum length of these keys is two digits (0-99). The key1 is added to the diagonal elements, key2 is added to the upper triangle elements, and key3 is added to the lower triangle elements. For example key1 = 10, key2 = 20 and key3 = 40, then new matrix will be as follows:

![Figure 1(b)](image1.png)

Let’s call this new matrix as “mid-matrix”.

The last key, key4 will be used to apply transposition on mid-matrix to achieve the final matrix. The length of the key4 depends on the length of the plaintext and is derived from the diagonal size of the matrix (or matrix size (MxM)). Here, the diagonal size is 4 (the matrix size (4x4)), the length of the key4 will be four. For example, let’s take key4 = 2031, where each digit in this key need to be less than its length, i.e no digit is greater than 3 as we start with zero. Also, digits in this key can’t be repeated.

The columns of the mid-matrix will be swapped according to the digits in the key4 and stored in another matrix called final-matrix. That is, in our example, column 2 from mid-matrix will be placed at first column in the final-matrix, and then column 0 from mid-matrix will be placed at second column in the final-matrix, column 3 from mid-matrix will be placed at third column in the final-matrix, and finally, column 1 from mid-matrix will be placed at last column in the final-matrix.

![Figure 2](image2.png)
Now, read the final-matrix row wise to get the cipher text. This will get following result in taken example.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>82</td>
<td>128</td>
<td>121</td>
</tr>
<tr>
<td>103</td>
<td>151</td>
<td>125</td>
<td>42</td>
</tr>
<tr>
<td>107</td>
<td>150</td>
<td>132</td>
<td>143</td>
</tr>
<tr>
<td>141</td>
<td>151</td>
<td>52</td>
<td>154</td>
</tr>
</tbody>
</table>

Figure (4)

Read outcomes: 
“128, 82, 128, 121, 103, 151, 125, 42, 107, 150, 132, 143, 141, 52, 154”

Finally, each ASCII value from the read outcomes will be converted to its equivalent character by comparing it with the ASCII table. So, the cipher text generated here will be “R y g } * k “. Here, only printable characters from 32 to 127 (ASCII value) will be displayed and rest will be white spaces in the cipher. This cipher can be stored into the database along with a unique identity number (at least 5 digits long) to add more security to the cipher.

**Encryption Algorithm**

STEP 1: Read the plain text.
STEP 2: Count the number of characters (N) in the plain text including white spaces.
STEP 3: Convert the plain text into equivalent ASCII code, character by character.
STEP 4: Create a square matrix (M) such that it can accommodate ASCII value of all N characters row wise from left to right and then fill remaining cells with 42 (ASCII of ‘*’).
STEP 5: Read the elements of the matrix into three parts, diagonal, upper triangle and lower triangle.
STEP 6: Apply three different keys k1, k2 and k3 on diagonal, upper triangle and lower triangle parts respectively for encryption (i.e. add these keys on respective parts)
STEP 7: Arrange the matrix column by column according to the key k4 and then read the matrix row wise.
STEP 8: Convert the ASCII values into characters to obtain the cipher text.

**Decryption:**
The process of decrypting cipher into original text is exactly in the reverse order of its encryption process. The only difference here is user has to apply unique identity number first to prove himself/herself as an authentic person for further implementation of the keys. The things to be taken care are sharing the unique identity number, first three keys (maximum two digits), and fourth key (size depends on the length of plaintext). Furthermore, the decrypted message will carry the ‘*’ for the number of empty spaces in the matrix which appended at the time of creation of matrix during encryption.

1. **ANALYSIS**

Here in the proposed algorithm, only numbers are considered to form the keys. We have three different (can be same) keys, k1, k2 and k3 of two digits in length and key k4 whose size depends on the size of matrix created from the plain text.

<table>
<thead>
<tr>
<th>Key</th>
<th>Length of the key</th>
<th>Possible Combinations</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>10</td>
<td>(10)^2</td>
</tr>
<tr>
<td>k2</td>
<td>10</td>
<td>(10)^2</td>
</tr>
<tr>
<td>k3</td>
<td>10</td>
<td>(10)^2</td>
</tr>
<tr>
<td>k4</td>
<td>M=5</td>
<td>(M)^4 = (5)^4</td>
</tr>
</tbody>
</table>

Table 6.1 Possible combinations of keys

Here, M denotes the size of the transpose matrix. So, the possible combinations of all the keys K1, K2, K3 and K4 of the proposed algorithm will be [(10)^6 x (5)^4] = 3.125 x 10^10 possible combinations.

**REFERENCES**

An Approach of Hybrid Transposition Method based on ASCII Value in Cryptography


