ROW TRANSPOSITION CIPHER IMPLEMENTATION

Shiyun hu Student ID 1298223 New York Institute of Technology shu10@nyit.edu Jun Jie Wu Student ID 1298381 New York Institute of Technology jwu72@nyit.edu

Leonardo Amorim de Lemos Student ID 1292678 New York Institute of Technology lamori01@nyit.edu

March 2, 2022

Abstract

The cipher appeared thousands of years ago but was heavily exploited from the 2nd World War. In our group project, we have learn one of the cryptography method in the transposition ciphers filed called Row Transposition cipher. Our group implements the Row Transposition cipher's encryption and decryption algorithm by using Java programming language.

I. Introduction

■ He cipher appeared thousands of years ago but was heavily exploited from the 2nd World War in which information exchanged between enemies, even encrypted, was intercepted and deciphered. There are two widely explored types of encryptions (transposition and substitution) that, when worked independently, can be solved quickly. Still, protocols have been created and improved that fundamentally use these two techniques repetitively. By definition, transposition cipher changes the plaintext order and rearranges to get ciphertext. In this group project, we used "Row Transposition Cipher," where you write your plaintext in rows of fixed length (key size), and we write by columns in key order. We can use the procedure to enhance the complexity of a more complex cipher-text.

II. ENCRYPTION IMPLEMENTATION

To implement the row transposition encryption, we utilize the key as a sequence to either read or fill the columns in a two-dimension matrix

as we encrypt or decrypt the input message (Figure 2 & Figure 5).

Take the key 'NYITV' as an example (Figure 1), the algorithm uses the 26 English letters to find the number sequence '14023', which means we read the 0 in the matrix's Col2 1st and 1 in the matrix Col0 2nd and so on. The length of the key defines the number of columns we use in the algorithm.

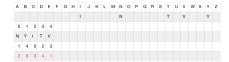


Figure 1: Task1-Encryption Sequence Order

The sequence reordering is '20341' (Figure 1) for decryption in reading the columns from 2D Matrix that transfers from the key order '14023'. In Figure 2, we read column by column in the reordering sequence '20341' from the matrix to produce our encrypted message.

Regards for the current assignment, the empty space would be replaced by the capital letter 'X' to append the rows to form the ciphertext.

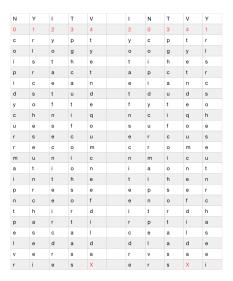


Figure 2: *Task1-Row Transcription Matrix*

The following figure 3 displays the encrypted message.

yotaetfnsecniteeircdrecoipidycurrm aipntpelvrpghcautifcoiohsortaasstye tndeqoumcneefdildaXrlsrcsoheseut nrchaseei

Figure 3: Task1-Encrypted Message



Figure 4: Task1-Output

The Figure 4 shows the result of the encrypted message. Further, we use the decrytion algorithm to double check the answer. The result matches to the original text.

III. DECRYPTION IMPLEMENTATION

The decryption algorithm put the letters in columns by the key order '14023'; however, as we fill the columns by following the order of '20341' (Figure 5, left). Take '0' as an example, the '0' is filled up 1st in the RowMatrix's 2nd column. Then, it reads off the message by '01234' row by row(Figure 5, right). The result decrypted message shows in Figure 6.

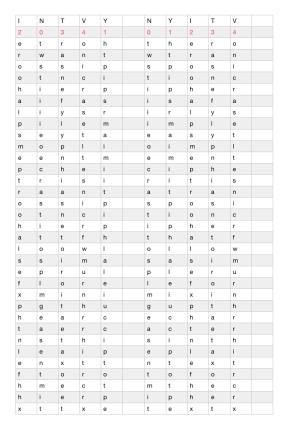


Figure 5: Task2-Decrypted Row Transposition Matrix

therowtranspositioncipherisafairlysi mpleeasytoimplementcipheritisatran spositioncipherthatfollowsasimpleru leformixingupthecharactersintheplai ntexttoformtheciphertextx

Figure 6: Task2-Decrypted Message

The following is the pseudocode for the row transposition cipher algorithm :

The first part finds the decoding sequence from the key 'NYITV'. Then, we write the encrypted message into a 2D RowTranspositionMatrix. Next, we use the matrix to record the rearrangement of the RowTranspositionMatrix. In the last step, we utilize the StringBuilder to build the encrypted message line by line through the RowMatrix.

Row Transposition Decryption Algorithm 1

```
input: 'w': Key and 'C' Encrypted plain-text
output: Decrypted plain-text
 1: function RTCDECRYPTION(w, C)
 2:
 3:
         keylen \leftarrow w.length()
 4:
         keyArray \leftarrow key.toCharArray()
 5:
         messageArray \leftarrow C.toCharArray()
 6:
         keyPosition \leftarrow int[keylen]
 7:
8:
                                > sort the keyArray and assign it to a string
         Sort the keyArray
 9:
         Strings \leftarrow String.valueOf(keyArray)
10:
11:
         x \leftarrow 0
12:
         for each char c in dArray do
13:
             keyPosition[x] \leftarrow s.indexOf(c)
14:
             Increament x by 1
15:
16:
         cols \leftarrow keylen
17:
18:
         rows \leftarrow 0 if C's length mod cols equals 0 then \triangleright calculate rows
19:
            rows \leftarrow C.length()/cols
20:

    □ Get the right order to decrypt the message in the matrix

21:
            rows \leftarrow C.length()/cols + 1
22:
23:
         RowMatrix \leftarrow char[rows][cols]
24:
         k \leftarrow 0
25:
         for i to rows do
26:
             for j to cols do
27:
                 if count k equals message C's length then
28:
                     while k equals message's length and j
29:
                     less than cols keep add 'X' to
30:
                     RowMatrix[i][j]
31:
                     break
32:
                     assign RowMatrix[j][keyPosition[i]] from
33:
                     messageArray[k]
34:
                     Increament k by 1
35:
36:
         StringBuilder\ str \leftarrow StringBuilder()
37:
         for i to rows do
38:
             for j to cols do
39:
                 if RowMatrix[i][j] unequal to 'X' then
                     str.append(RowMatrix[i][j])
40:
41:
         return str
                                              Decrypted message is str
```

the decrypted plaintext. Further, we use the encryption algorithm to double-check the answer. The result matches the original encrypted message.

```
charles@charlesdeMacBook-Pro:-/Desktop/NYIT/INCS741/INCS741.... C#2
charles@charlesdeMacBook-Pro:-/Desktop/NYIT/INCS741/INCS741....
charles@charlesdeMacBook-Pro:-/Desktop/NYIT/INCS741/INCS741....
charles@charlesdeMacBook-Pro:-/Desktop/NYIT/INCS741/INCS741....
charles@charlesdeMacBook-Pro:-/Desktop/NYIT/INCS741/INCS741....
project 2
caded charles of comparing contact conta
```

Figure 7: Task2-Output

IV. Conclusion

With the growing use of computers and the internet, and an increasing need to transmit information quickly and securely, encryption through existing protocols (AES, RSA, 3DES, etc.) became essential. In the project, we can see that using only one round of encryption (row transportation) and a minor key (5 letters), the information is already quite challenging to decipher, and with the use of the protocols mentioned above that repeatedly (using transportation, substitution, and other resources), it becomes harder to decipher the messages. We also demonstrate in the project that the information is decrypted, just doing the inverse of the encryption procedure that needs to be done by the person who will receive the message.

The following figure 7 shows the result of

REFERENCES

[1]N. Hamza, "Row transposition ciphers - ppt download," SlidePlayer. [Online]. Available: https://slideplayer.com/slide/13205094/. [Accessed: 28-Feb-2022].

[2]"Transposition cipher," Wikipedia, 23-Feb-2022. [Online]. Available: https://en.wikipedia.org/wiki/Transposition_cipher. [Accessed 28-Feb-2022].