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.

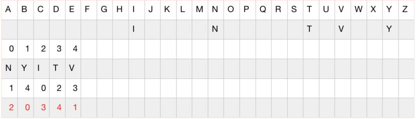
I. Introduction

The 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.

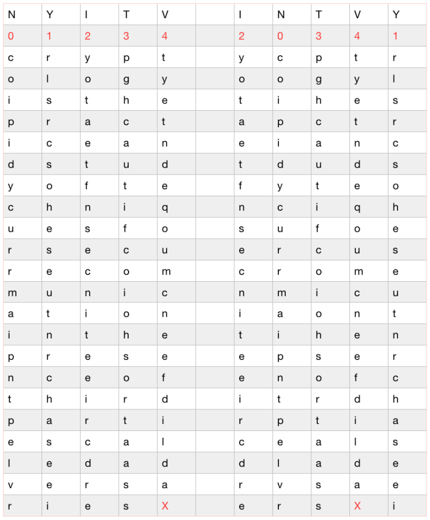


A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
								I					N						T		V			Y	
0	1	2	3	4																					
N	Y	I	T	V																					
1	4	0	2	3																					
2	0	3	4	1																					

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.

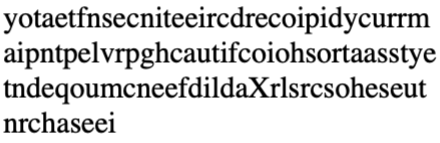
Regards for the current assignment, the empty space would be replaced by the capital letter 'X' to append the rows to form the ciphertext.



N	Y	I	T	V		I	N	T	V	Y
0	1	2	3	4		2	0	3	4	1
c	r	y	p	t		y	c	p	t	r
o	l	o	g	y		o	o	g	y	l
i	s	t	h	e		t	i	h	e	s
p	r	a	c	t		a	p	c	t	r
i	c	e	a	n		e	i	a	n	c
d	s	t	u	d		t	d	u	d	s
y	o	f	t	e		f	y	t	e	o
c	h	n	i	q		n	c	i	q	h
u	e	s	f	o		s	u	f	o	e
r	s	e	c	u		e	r	c	u	s
r	e	c	o	m		c	r	o	m	e
m	u	n	i	c		n	m	i	c	u
a	t	i	o	n		i	a	o	n	t
i	n	t	h	e		t	i	h	e	n
p	r	e	s	e		e	p	s	e	r
n	c	e	o	f		e	n	o	f	c
t	h	i	r	d		i	t	r	d	h
p	a	r	t	i		r	p	t	i	a
e	s	c	a	l		c	e	a	l	s
l	e	d	a	d		d	l	a	d	e
v	e	r	s	a		r	v	s	a	e
r	i	e	s	X		e	r	s	X	i

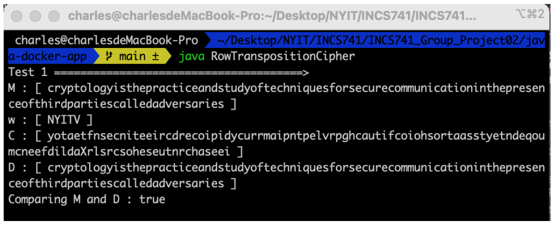
Figure 2: *Task1-Row Transcription Matrix*

The following figure 3 displays the encrypted message.



yotaetfnsecniteeircdrecoipidycurrm
aipntpelvrpghcautifcoiohsortaasstye
tndeqoumcneefdildaXrlsrcsoheseut
nrchaseei

Figure 3: *Task1-Encrypted Message*



charles@charlesdeMacBook-Pro:~/Desktop/NYIT/INCS741/INCS741...
⌥⌘2

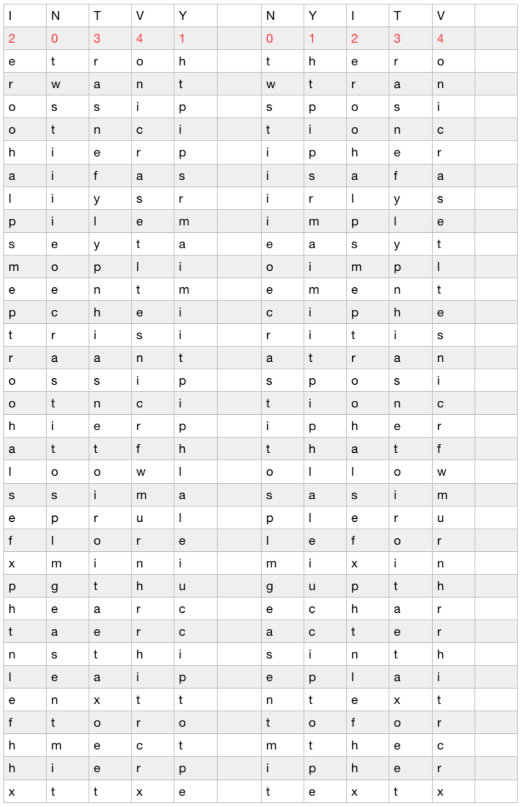
```
 charles@charlesdeMacBook-Pro  ~/Desktop/NYIT/INCS741/INCS741_Group_Project02/jav
a-docker-app  main ±  java RowTranspositionCipher
Test 1 ========================================>
M : [ cryptologyisthepracticeandstudyoftechniquesforsecurecommunicationinthepresen
ceofthirdpartiescalledadversaries ]
w : [ NYITV ]
C : [ yotaetfnsecniteeircdrecoipidycurrmaipntpelvrpghcautifcoiohsortaasstyetndeqou
mcneefdildaXrlsrcsoheseutnrchaseei ]
D : [ cryptologyisthepracticeandstudyoftechniquesforsecurecommunicationinthepresen
ceofthirdpartiescalledadversaries ]
Comparing M and D : true
```

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.

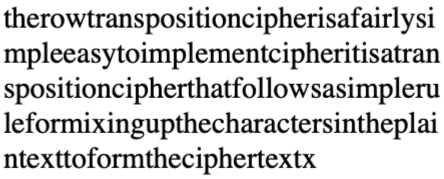
III. Decryption Implementation

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.



I	N	T	V	Y		N	Y	I	T	V	
2	0	3	4	1		0	1	2	3	4	
e	t	r	o	h		t	h	e	r	o	
r	w	a	n	t		w	t	r	a	n	
o	s	s	i	p		s	p	o	s	i	
o	t	n	c	i		t	i	o	n	c	
h	i	e	r	p		i	p	h	e	r	
a	i	f	a	s		i	s	a	f	a	
l	i	y	s	r		i	r	l	y	s	
p	i	l	e	m		i	m	p	l	e	
s	e	y	t	a		e	a	s	y	t	
m	o	p	l	i		o	i	m	p	l	
e	e	n	t	m		e	m	e	n	t	
p	c	h	e	i		c	i	p	h	e	
t	r	i	s	i		r	i	t	i	s	
r	a	a	n	t		a	t	r	a	n	
o	s	s	i	p		s	p	o	s	i	
o	t	n	c	i		t	i	o	n	c	
h	i	e	r	p		i	p	h	e	r	
a	t	t	f	h		t	h	a	t	f	
l	o	o	w	l		o	l	l	o	w	
s	s	i	m	a		s	a	s	i	m	
e	p	r	u	l		p	l	e	r	u	
f	l	o	r	e		l	e	f	o	r	
x	m	i	n	i		m	i	x	i	n	
p	g	t	h	u		g	u	p	t	h	
h	e	a	r	c		e	c	h	a	r	
t	a	e	r	c		a	c	t	e	r	
n	s	t	h	i		s	i	n	t	h	
l	e	a	i	p		e	p	l	a	i	
e	n	x	t	t		n	t	e	x	t	
f	t	o	r	o		t	o	f	o	r	
h	m	e	c	t		m	t	h	e	c	
h	i	e	r	p		i	p	h	e	r	
x	t	t	x	e		t	e	x	t	x	

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 :

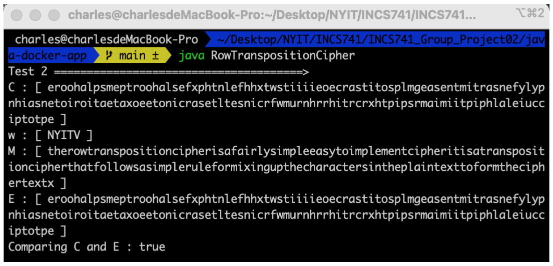
The first part finds the decoding sequence from the key 'NYITV'. Then, we write the encrypted message into a 2-dimension RowTranspositionMatrix. Next, we use the matrix to record the rearrangement of the RowTranspositionMatrix. In the last step, we utilize the StringBuilder to build the encrypted message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

Row Transposition Decryption Algorithm 1

```
input: 'w' : Key and 'C' Encrypted plain-text
output: Decrypted plain-text
function RTCDECRYPTION($w, C$)

    $keylen \leftarrow w.length()$
    $keyArray \leftarrow key.toCharArray()$
    $messageArray \leftarrow C.toCharArray()$
    $keyPosition \leftarrow int[keylen]$

    Sort the keyArray                            ▷ sort the keyArray and assign it to a string
    $Strings \leftarrow String.valueOf(keyArray)$

    $x \leftarrow 0$
    for each char c in dArray do
        $keyPosition[x] \leftarrow s.indexOf(c)$
        Increament x by 1

    $cols \leftarrow keylen$
    $rows \leftarrow 0$
    if C's length mod cols equals 0 then        ▷ calculate rows
        $rows \leftarrow C.length()/cols$
    else                                         ▷ Get the right order to decrypt the message in the matrix
        $rows \leftarrow C.length()/cols + 1$

    $RowMatrix \leftarrow char[rows][cols]$
    $k \leftarrow 0$
    for i to cols do
        for j to rows do
            if count k equals message C's length then
                break
            assign $RowMatrix[j][keyPosition[i]]$
            from $messageArray[k]$
            Increament k by 1

    $StringBuilder\ str \leftarrow StringBuilder()$
    for i to rows do
        for j to cols do
            if $RowMatrix[i][j]$ unequal to 'X' then
                $str.append(RowMatrix[i][j])$
    return str                                   ▷ Decrypted message is str
```

The following figure 7 shows the result of 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...
⌥⌘2

```
 charles@charlesdeMacBook-Pro  ~/Desktop/NYIT/INCS741/INCS741_Group_Project02/jav
a-docker-app  main ±  java RowTranspositionCipher
Test 2 ======================================>
C : [ eroohalpsmeptroohalsefxphtnlefhhxtwstiiiieoecrastitosplmgeasentmitrasnefylyp
nhiasnetoiroitaetaxoeetonicrasetltesnicrfwmurnhrrhitrcrxhtpipsrmaimiitpiphlaleiucc
iptotpe ]
w : [ NYITV ]
M : [ therowtranspositioncipherisafairlysimpleeasytoimplementcipheritisatransposit
ioncipherthatfollowsasimpleruleformixingupthecharactersintheplaintexttoformtheciph
ertextx ]
E : [ eroohalpsmeptroohalsefxphtnlefhhxtwstiiiieoecrastitosplmgeasentmitrasnefylyp
nhiasnetoiroitaetaxoeetonicrasetltesnicrfwmurnhrrhitrcrxhtpipsrmaimiitpiphlaleiucc
iptotpe ]
Comparing C and E : true
```

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.

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].