

# 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  
jjwu72@nyit.edu

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

February 27, 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 transposition cipher means that we change the order of the plan-text and rearrange to get ciphertext. Row transportation cipher is the plan-text that is written in rows of fixed length, and we write by column in key order. To enhance the complexity we can follow the procedure to get a more complex cipher-text.

## II. ENCRYPTION IMPLEMENTATION

To implement the row transposition encryption, we utilizes the key as a sequence to switch the columns in a two-dimension matrix to form a row transposition matrix(Figure 3).

Take the key 'NYITV' as an example (Figure 1), the algorithm uses the 26 English letters to find the number sequence '14023'. Then, the algorithm arranges the columns by the order of this number sequence. The encryption algorithm writes letters of message out in rows over a specified number of columns which equals to the key length '5' (Figure 2). Then, reorder columns in the matrix (Figure 3).

[illegible]

**Figure 1:** *Task1-Encryption Sequence Order*

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

**Figure 2:** *Task1-Message Matrix*

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

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

Figure 3: Task1-Row Transposition Matrix

The following figure 4 displays the encrypted message.

rtcyplyoogseithrtpaccnieasddtuoe  
yfhqcnieousfsurecemrcoucmmnitnaio  
neithrepescfneohdtiraiprtslecaedlda  
eavrsiXres

Figure 4: Task1-Encrypted Message

```
charles@charlesdeMBP:~/desktop/java-do...
charles@charlesdeMBP: ~/desktop/java-docker-app master
$ java RowTranspositionCipher
Test 1
M : [ cryptologyisthepracticeandstudyoftechniquesforsecureco
mmunicationintheabsenceofthirdpartiescalledadversaries ]
w : [ NYITV ]
C : [ rtycplyoogseithrtpaccnieasddtuoe yfhqcnieousfsurecemrc
oucmmnitnaio neithrepescfneohdtiraiprtslecaedlda eavrsiXres ]
D : [ cryptologyisthepracticeandstudyoftechniquesforsecureco
mmunicationintheabsenceofthirdpartiescalledadversaries ]
Comparing M and D : true
```

Figure 5: Task1-Output

Figure 5 shows the result of the encrypted plaintext. Further, we use the decryption algorithm to double check the answer. The result match to the original text.

### III. DECRYPTION IMPLEMENTATION

The decryption algorithm is the reverse order of the encryption algorithm. The algorithm first writes the encrypted message out in rows. Then, it reads off the message by recording columns.

0	1	2	3	4		
N	Y	I	T	V		
1	4	0	2	3		
R	E	O	R	D	E	R
2	0	3	4	1		

Figure 6: Task2-Decryption Sequence Order

The reorder sequence is 20341 to decryption by using the same row exchange method.

	0	1	2	3	4
e	r	o	o	h	
a	i	p	s	m	
e	p	t	r	o	
o	h	a	i	s	
e	f	x	p	h	
t	n	i	e	f	
h	h	x	t	w	
s	t	i	i	i	
i	e	o	e	c	
r	a	s	t	i	
t	o	s	p	i	
m	g	e	a	s	
e	n	t	m	i	
t	r	a	s	h	
e	f	y	i	y	
p	n	h	i	a	
s	n	e	t	o	
i	r	o	i	t	
a	e	t	a	x	
o	e	e	t	o	
n	i	c	r	a	
s	e	t	i	t	
e	s	n	i	c	
r	f	w	m	u	
r	n	h	r	r	
h	i	t	r	c	
r	x	h	t	p	
i	p	s	r	m	
a	i	m	i	i	
t	p	i	p	h	
i	a	i	e	i	
u	c	c	i	p	
t	o	t	p	e	

Figure 7: Task2-Message Matrix

	2	0	3	4	1
o	e	o	h	r	
p	a	s	m	i	
t	e	r	o	p	
a	o	i	s	h	
x	e	p	h	f	
i	t	e	f	n	
x	h	t	w	h	
i	s	i	i	t	
o	i	e	c	e	
s	r	t	i	a	
s	t	p	i	o	
e	m	a	s	g	
t	e	m	i	n	
a	t	s	n	r	
y	e	i	y	f	
h	p	i	a	n	
e	s	t	o	n	
o	i	i	t	r	
t	a	a	x	e	
e	o	t	o	e	
c	n	r	a	i	
t	s	i	t	e	
n	e	i	c	s	
w	r	m	u	f	
h	r	r	r	n	
t	h	r	c	i	
h	r	t	p	x	
s	i	r	m	p	
m	a	i	i	i	
i	t	p	h	p	
i	i	e	i	a	
c	u	i	p	c	
t	t	p	e	o	

Figure 8: Task2-Decrypted Row Transposition Matrix

The first part finds the decoding sequence from the key 'NYITV'. Then, write the encrypted message into a 2D matrix 'plainTex-

tArray'. Next, use the RowTranspositionMatrix to record the rearrangement of the 'plainTex-tArray'. The last step, utilize the stringbuilder to build the encrypted message line by line through the RowTranspositionMatrix. The following is the pseudocode for the row transposition cipher algorithm :

---

#### Row Transposition Decryption Algorithm 1

---

**input:** 'w' : Key and 'C' Encrypted plain-text

**output:** Decrypted plain-text

```

1: function RTCDECRYPTION(w, C)
2:   keylen ← w.length()
3:   keyArray ← key.toCharArray()
4:   messageArray ← C.toCharArray()
5:   keyPos ← int[keylen]
6:
7:   Sort the keyArray
8:   Strings ← String.valueOf(keyArray)
9:   keyPos ← int[keylen]
10:
11:   for each char c in dArray do
12:     keyPosition[x] ← s.indexOf(c)
13:     Increment x by 1
14:
15:   cols ← keylen
16:   rows ← 0
17:   if C's length mod cols equals 0 then ▷ calculate rows
18:     rows ← C.length()/cols
19:   else ▷ calculate columns
20:     rows ← C.length()/cols + 1
21:
22:   plainTextArray ← char[rows][cols]
23:   k ← 0
24:   for i to rows do
25:     for j to cols do
26:       if count k equals message C's length then
27:         while k equals message's length and j less than cols keep
28:           add 'X' to plainTextArray[i][j]
29:         break
30:       assign plainTextArray[i][j] from messageArray[k]
31:       Increment k by 1
32:
33:   rowTranspositionMatrix ← char[rows][cols]
34:   for i to rows do
35:     for j to cols do ▷ each column read by the key order
36:       assign RowTranspositionMatrix[i][j]
37:       from plainTextArray[i][keyPosition[j]]
38:
39:   StringBuilder str ← StringBuilder()
40:   for i to rows do
41:     for j to cols do
42:       str.append(RowTranspositionMatrix[i][j])
43:   return str ▷ Decrypted message is str

```

---

message.

oeohrpasmilteropaolshxephfltefnxh  
twhisiitoiecesrtiastploemasgtemina  
tsnryelyfhpianestonoiiirtaaxeetoec  
cnraitslteneicswrmufhrrnrthrcihrtpx  
sirmpmaiitphlleiucipcttpeo

Figure 9: Task2-Decrypted Message

```

charles@charlesdeMBP:~/Desktop/NYIT/INCS741/INCS741_Grou...
charles@charlesdeMBP:~/Desktop/NYIT/INCS741/INCS741_Group_Project02/java-doc
ker-app $ java RowTranspositionCipher
Test 2
C : [ eroohalpsmeptroohalsefxphntlefhxhtwtstiiieocrastitospmgeasentmitrasnefy
lyphnasnetoiroitaetaxoeetonicsrateltesnicrfwmurnhrhritrcrxhtpsrmaiitphila
leiucipcttpe ]
w : [ NYITV ]
M : [ oeohrpasmilteropaolshxephfltefnxhtwhisiitoiecesrtiastploemasgteminatsnryel
yfhpianstonoiiirtaaxeetoecnraitslteneicswrmufhrrnrthrcihrtpxsirmpmaiitphl
leiucipcttpe ]
row : 33 cols : 5
E : [ eroohalpsmeptroohalsefxphntlefhxhtwtstiiieocrastitospmgeasentmitrasnefy
lyphnasnetoiroitaetaxoeetonicsrateltesnicrfwmurnhrhritrcrxhtpsrmaiitphila
leiucipcttpe ]
Comparing C and E : true

```

Figure 10: Task2-Output

Figure 10 shows the result of the decrypted plaintext. Further, we use the encryption algorithm to double check the answer. The result match to the original encrypted text.

#### IV. CONCLUSION

With the growing use of computers and the Internet, and an increasing need to transmit information quickly and securely, the use of encryption through existing protocols (AES, RSA, 3DES, etc.) information security that uses the two types of transposition mentioned (transposition and substitution).

In this example, we can see that using only one round of encryption 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 use the types of transposition, it becomes almost impossible 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

#### REFERENCES

- [1]G. Newell, "An introduction to linux access control lists (acls)," Enable Sysadmin, 07-Jan-2022. [Online]. Available: <https://www.redhat.com/sysadmin/linux-access-control-lists>. [Accessed: 21-Feb-2022].