IFSCR

Cryptography Summer internship

PESU

Bangalore

Single columnar transposition cryptanalysis

Nikhil M Adyapak PES2UG19CS257

Contents

1. Problem description

Design and implement a python program that analyses given ciphertext and produces possible plaintext. (Ciphers produced by using single columnar encryption).

2. Introduction

Single columnar transposition **[#Ref4] [#Ref7] [#Ref9]**

is a type of cipher that follows a simple rule for mixing up characters in the plain text form the cipher text. The message is written out in rows of a fixed length, and read out again column by column, and the columns are chosen by some scrambled order. Both the length of the rows and the permutations of the columns are usually defined by a keyword.
The columnar transposition requires both the encoder and the decoder to know the keyword when encoding the plain text to cipher text and vice versa.

This technique was invented by the ancient Greeks used by the Spartans to send secret messages.

This technique was used extensively in the early 1900's by the Germans during the world war to send encrypted texts to military bases of their allies.

The key was shared to all the cipher operators at the beginning of the day and was supposed to be discarded after memorizing it.

The columnar transposition is very secure when encoded with long keys (around length 20), but much weaker if shorter keywords are used. If the length of the keyword can be known, by permutations and combinations, the plain text can be obtained by brute force technique.

3. Literature Survey of the problem statement

[#Ref2] [#Ref6] [#Ref7]

Examples of a columnar transposition

Example 1:

Plain text – attack postponed until three am.

Key – dcab

The key dcab can be translated to the numbers 4 3 1 2.

This numbering is given according to the letters in the key appearing in alphabetic order.

4	3	1	2
a	t	t	a
c	k	p	o
s	t	p	o
n	e	d	u
n	t	i	l
l	t	h	r
e	e	a	m

The cipher text is read column wise according to the order of the key specified.

Cipher text – tppdiha aooulrm tktette acsnnle

This so happens to be an ideal case where the length of the plain text happens to fit perfectly in the matrix.

In some cases, some dummy characters need to be inserted to completely fill up the matrix.

Example 2:

Plain text – attack postponed until two am.

Key – dcabefg

The key dcab can be translated to the numbers 4 3 1 2 5 6 7.

4	3	1	2	5	6	7
a	t	t	a	c	k	p
o	s	t	p	o	n	e
d	u	n	t	i	l	t
w	o	a	m	X	Y	Z

Here, since the matrix is not filled up completely, X Y Z are dummy characters used to fill it up.

The dummy characters can be spaces or any other randomly generated characters so that the cipher cannot be easily decoded.

Cipher text when dummy characters are replaced by '_' is

ttna aptm tsuo aodw coi_ knl_ pet_

Decoding columnar transposition

By knowing the length of the cipher and the key permutation 4 3 1 2 5 6 7, example 2 can be converted back to plain text. **[#Ref7]**

By dividing the total length of the cipher by the length of the key, number rows originally present in the matrix while encrypting can be calculated.

Here, the number of rows is 4 that can be got from dividing 28 (length of text) by 7 (length of key).

Rearranging the cipher text by taking single letters at a time from the key permutations and representing the same as columns, the plain text is obtained.

Cipher text - ttna aptm tsuo aodw coi_ knl_ pet_

From the cipher text and the key 4 3 1 2 5 6 7,

We take the first letters from element 4 which is 'a', element 3 which is 't', element 1 which is 't', element 2 which is 'a', element 5 which is 'c', element 6 which is 'k' element 7 which is 'p'.

We repeat this row number of times for the next characters in the same order of the key permutation.

In the problem statement only the cipher text is given to us. **[#Ref6]**

The key nor the size of the key is given to us.

Therefore, the length of the key must be found such that the length of the key perfectly divides the length of the cipher text.

The first step in attacking a columnar transposition cipher is to try all possible short keywords. If all keywords are checked up to a length of 9, the compute time is not very long. The number of possible rearrangements of a length N key is N! (N factorial). This number grows very quickly as N gets larger. The number of possible keys for various length keywords is shown below:

Key Length	No. of permutations	Examples
2	2	AB, BA
3	6	ABC, BAC, CBA, ...
4	24	ABCD, ABDC, ACBD, ...
5	120	ABCDE, ABCED, ...
6	720	ABCDEF, ABDCFE, ...
7	5,040	ABCDEFG, ABDCGEF, ...
8	40,320	ABCDEFGH, ...
9	362,880	ABCDEFGHI, ...
10	3,628,800	ABCDEFGHIJ, ...
11	39,916,800	ABCDEFGHIJK, ...
12	479,001,600	ABCDEFGHIJKL, ...

Trying to test all possible combinations of a length 6 keyword is easy, 720 trial decryptions can be done very quickly. However, there is little hope of trying to enumerate all possible length 12 keywords, as it would take far too long. This is why we stop at around length 9 keywords.

By assuming a key length that divides the cipher text completely, that is the assumption that the given cipher text fits perfectly into a matrix, brute force is one of the ways in finding out the possible plain text without the use of a keyword by trying out all possible permutations in the key length that is got that divides the entire cipher text into a perfect matrix. This is assuming that while encoding the plain text in the single columnar transposition technique, it is properly padded. Padding refers to completely filling the matrix according to a specific keyword or key length.

Cryptanalysis

The key is the most vulnerable part of this cipher because attackers just knowing the key length can decipher the cipher text into plain text by using brute force technique. **[#Ref5] [#Ref8]**

For a key length of n, the number of possible key permutations is n! (n factorial).

In example 2, dummy characters X Y Z are added. Since the matrix was not filled completely, the strength of the columnar transposition cipher is increased by adding random dummy characters.

Cryptanalysis is the study of ciphertext, ciphers and cryptosystems with the aim of understanding how they work and finding and improving techniques for defeating or weakening them.

All substitution and transposition ciphers so far has been decoded and there are methods being devised by researchers and cryptography experts in this field where they use and a combination of different ciphers like hill ciphers, play fair cipher, Caeser cipher and different transposition ciphers like columnar transposition, rail fence transposition and double transposition.

The columnar transposition cipher on its own is not that difficult to decode by attackers. In combination with other ciphers, it becomes really difficult to decode the cipher into plain text.

In research papers **[#Ref3] [#Ref4],** there is a new bit level approach for encoding and decoding columnar transposition ciphers, combined columnar transposition with hill cipher, and **[#Ref1]** published its research in encoding plain text initially with rail fence or columnar transposition cipher and then applying another layer of security be coding the cipher text again with Caeser cipher.

Columnar transposition according to some trusted articles, is very secure when key lengths above 20 are chosen. This is because to decode the cipher text, there is at least 20! Permutations of keys to be used. Hence it is really hard for attackers to decode the cipher.

In some of the cryptanalysis studies **[#Ref8] [#Ref5]**, some researchers used frequencies to determine the dimensions of the matrix and then used digraph frequencies to arrange the columns.

Taking an example, using the cipher text,

ASAIR ITFNM IMTKL SOIEE M

The ‘key’ to cryptanalyzing ciphertext is to determine the number of columns, that is the length of the keyword. There are 21 letters in the cipher text. Because the message completely fills the rectangle (since the cipher text has been padded), this suggests either a 3x7 or a 7x3 array.

Arranging the ciphertext in columns,

A	I	T	M	T	S	E		A	F	L
S	R	F	I	K	O	E	or	S	N	S
A	I	N	M	L	I	M		A	M	O
								I	I	I.
								R	M	E
								I	T	E
								T	K	M

This cryptanalysis study had a solution by anagramming that is making a word or portions of words by rearranging letters.

By analyzing both the array possibilities, the 7x3 matrix can be rejected since there is no word with ‘MKT’ or ‘III’. Therefore, the study points to the 3x7 matrix.

A	I	T	M	T	S	E
S	R	F	I	K	O	E
A	I	N	M	L	I	M

This cryptanalysis had another approach for determining the dimensions of the rectangle or matrix. Frequencies of vowels helped their study. In English approximately 40% of plaintext consisted of vowels. Hence in the same 3x7 matrix,

							Number of vowels	Difference
A	I	T	M	T	S	E	3	0.2
S	R	F	I	K	O	E	3	0.2
A	I	N	M	L	I	M	3	0.2

The sum of the differences is 0.6

			Number of vowels	Difference
A	F	L	1	0.2
S	N	S	0	1.2
A	M	O	2	0.8
I	I	I	3	1.8
R	M	E	1	0.2
I	T	E	2	0.8
T	K	M	0	1.2

The sum of differences is 6.2, therefore the 3x7 matrix is more likely to produce the plain text. Using digraph frequencies to arrange the columns, it helps in the cryptanalysis in place of looking for reasonable columns for the same example,

The researcher's study was that he had to study the other pairings and then manually apply them as in a trial-and-error way and concluded that they would most probably get the right pairing on the first try. Once they got a probable pairing, they would then continue the same process of using digraph frequencies to select columns to add on to the left or right.

This study had a very unique cryptanalysis technique of predicting the matrix and figuring a way of rearranging the columns based on statistics from different pairings in English. Since this technique however is based on trial and error and is almost similar to trying out every single combination of key length permutation, the brute force technique seems more viable with the restrictions being unknown key length and permutations to the decoder and the prerequisite being that the matrix is padded completely or filled completely with random characters or spaces for a secret keyword from the encoder.

Research papers **[#Ref1] [#Ref3] [#Ref6]**

and cryptanalysis of this cipher tried key lengths up to 16 characters long and some tried up to key lengths of 9. As seen in the table earlier, the number of permutations increases exponentially with increase in key lengths as the number of permutations for a given key length N is equal to N! (N factorial). Due to long computing time required, some researchers in their papers have restricted their cryptanalysis to key lengths of 9 or 16.

1)Single transposition ciphers with key lengths above 20 are secure

2)Encoding the plain text with multiple ciphers with moderate key lengths for columnar transposition is also very secure.

4. Current Scope of Columnar transposition cipher problem

The current scope of the project considers the following two approaches

1) The columnar transposition has plain texts written with spaces instead of the compressed plain text. This helps to decipher the cipher text with trial-and-error method where we can try out all the permutations of keyword lengths from 1 to some n where we choose the value of the key length if it is perfectly dividing the cipher text length. This is an assumption however that the key lengths are size 9 or smaller.

For example,

Plain text – attack postponed until two am

Compressed Plain text – attackpostponeduntiltwoam

Here, if we used the plain text as it is and filled the matrix by spaces with the configuration of some secret key and key length with padding, then it is possible to reverse engineer the exact plain text by trying out different key lengths and permutations and seeing if each word reconstructed using this method is present in the particular plain text language's dictionary.

Taking the same example as before,

Plain text – attack postponed until two am

Key – dcabefg

This key will be known only to the person encoding the cipher and will be unknown to the person trying to decode it.

The key dcabefg translates to 4 3 1 2 5 6 7

4	3	1	2	5	6	7
a	t	t	a	c	k	(space)
p	o	s	t	p	o	n
e	d	(space)	u	n	t	i
l	(space)	t	w	o	(space)	a
m	(padding)	(padding)	(padding)	(padding)	(padding)	(padding)

The padding can be any random computer-generated character, but in order to maintain simplicity, we can have spaces.

The cipher text would be –

ts t atuw tod apelmcpno kot nia .

When this is reconstructed, with keyword and key length unknown, we can check if all the words in this text is present in the English dictionary to verify that we have indeed arrived at the correct plain text.

2) The columnar transposition can be applied to plain texts for a secretly generated key.

In this method, the compressed cipher text would be used to fill the matrix for the secret key known only to the encoder and will be padded to complete the matrix.

This technique will generate plain texts from permutations of all possible key lengths up to size 5 or smaller.

5. Software Requirements

- Operating System – Windows 10
- Processor - Intel Core i5-7Y54 CPU @ 1.20GHz 1.61 GHz
- Language – Python 3.8.8
- Tools used – Pycharm IDE (2019 community edition)
- Packages – Pandas, numpy and cryptography python packages

6. References

1. A Novel Approach for Encryption of TextMessages, Analysis and Implementation of Simple Columnar Transposition Cipher with Ceasar Cipher and Rail Fence Cipher in C/C++ (by 1 Jawad Ahmad Dar, 2Amit Verma 1Research Scholar, Computer Science and Engineering, Kurukshetra University Kurukshetra, Haryana, India 2HOD CSE, NNSS Samalkha Group of Institution, Kurukshetra University Kurukshetra, Haryana, India) (International Journal of Science and Research (IJSR))

2. Evaluative Study on Substitution and Transposition Ciphers (by Dr. Sumathy Kingslin, R.Saranya, Associate Professor, M.Phil Research ScholarPG & Research Dept. of Computer Science,Quaid-E-Millath Government College for Women (Autonomous), Anna Salai, Chennai 600002, Tamil Nadu, India)

3. A New Bit-level Columnar Transposition Encryption Algorithm (by Sayantan Majumdar, Abhisek Maiti Biswarup Bhattacharyya Asoke Nath, Department of Computer Science, St. Xavier's College, Kolkata, India) (International Journal of Advance Research in Computer Science and Management Studies)

4. Columnar Transportation: https://studydriver.com/columnar-transportation/

5. Cryptanalysis of columnar transpositions from https://www.uni-mainz.de/

6. Articles at http://practicalcryptography.com/

7. Transposition ciphers from https://wikipedia.org/

8. Cryptanalysis of transposition ciphers by christen

9. Modern cryptanalysis, techniques for advanced code breaking from https://books.google.com/