MysteryTwister C3

THE CRYPTO CHALLENGE CONTEST

DOUBLE-COLUMN TRANSPOSITION/GRANIT – PART 1

Author: Joerg Drobick

May 2015 (description updated Oct 2015)

Introduction (1/2)

While nowadays one encrypts almost exclusively by computer systems, in the centuries before the development of IT, simpler methods had to be used. For example, in the 19th and early 20th centuries, the double-column transposition (DTP) was often used. DTP applies a columnar transposition twice.
This method can be done manually, and up-to-now no ciphertext-only attack is known, provided that the method has been parameterized correctly: The two words used as permutation keys are each at least 20 characters long, the plaintext is unknown and the wordlengths are co-prime.
See the challenges "DCT Reloaded":
https://www.mysterytwisterc3.org/en/challenges/level-iii/double-column-transposition-reloaded-part-1

Introduction (2/2)

Among others, the spy Guenter Guillaume used the double-column transposition for his communication with the Ministry of State Security of the former GDR until about 1960.
More precisely, he used an even stronger two-stage method known as GRANIT E160: In the first step the plaintext is converted into a numerical sequence with the help of a codebook and a matrix (a variant of the Polybius square), and in the second step the actual double-column transposition is run.

This challenge series is about that GRANIT method.

Challenge Description (1/2)

Part 1 of this GRANIT series is a ciphertext-only challenge. Your task is to recover the plaintext from the following ciphertext:

```
69554 69554 04881 88790 79599
75901 65181 55790 09480 47890
57849 90419 00997 40082
```

Due to historical usage, we refer to this ciphertext message as a telegram.

Before performing the double-column transposition at least one word has been substituted with a 3-digit number (using the codebook on page 6).

Challenge Description (2/2)

To obtain the correct plaintext, you will need to undo the double-column transposition and the substitutions (with the help of the matrix and the codebook). Moreover, possible padding at the end of the text should be removed.

As solution, submit the meeting place from your putative plaintext in capital letters. Make sure to enter the complete term.

Hint: All challenges of this series use the very same keyword W for the generation of the key matrix (used in the substitution step): `MEINHAUS`. The empty fields of the first row are also identical.

Codebook

3.7. Das dazugehörige Codebuch TITAN-Z

000 abgesandt	253 Deckadresse	505 laufend	758 Stimmung
019 Adresse	262 Dokument	514 Legende	767 TBK
028 Änderung	271 dringend	523 lesbar	776 Termin
037 Anleg-en/ung	280 Einsatz, einsetzen	532 Maßnahme	785 Treff
046 Antwort-en	299 Einschätz-en/ung	541 Material	794 Treff wie vereinbart
055 Anweis-en/ung	307 einverstanden (mit)	550 Mikrat	802 Treffart
064 Arbeit-en	316 Empfang-en	569 Militär-isch	811 Treffttermin
073 Arbeitsstelle	325 Entleer-en/ung	578 mitbringen	820 über
082 Aufenthalt	334 entleert	587 Mitteil-en/ung	839 Übergabe, übergeben
091 Aufgabe, aufgeben	343 Ergebnis	596 Nachricht	848 Überprüf-en/ung
109 Aufklär-en/ung	352 Erhalt-en/ung	604 nächst	857 unbedingt
118 Aufnahme, aufgeben	361 Ermittel-n/ung	613 negativ	866 Unterstütz-en/ung
127 Auftrag	370 Erwart-n/ung	622 normal	875 Verbind-en/ung
136 ausführlich	389 Feststell-en/ung	631 notwendig	884 Vereinbar-en/ung
145 Bahnhof	398 Frequenz	640 Objekt	893 Vernicht-en/ung
154 Beginn-en	406 Funk	659 operativ	901 voraussichtlich
163 Beleg-en/ung	415 Geheimschreibmittel	668 Päckchen	910 Vorbereit-en/ung
172 belegt	424 Grenzübergang	677 Politik, politisch	929 vorläufig
181 benötigen	433 Information, informieren	686 Post	938 Vorschlag-en
190 Beobacht-en/ung	442 Instrukteur	695 Post noch nicht erhalten	947 Westberlin
208 Bericht-en	451 Interesse, interessieren	703 Reaktion (auf)	956 Westdeutschland
217 Berlin	460 Karte	712 Send-en/ung	965 Wiederholung-en/ung
226 Bestätig-en/ung	479 Kontakt	721 Sicherheit	974 Wirtschaft-lich
235 Brief	488 Kontroll-en/ieren	730 sofort	983 Zeichen
244 Chiffre	497 Kurier	749 Spruch	992 Zentrale

Example (1/17) – Overview of the 4 parts

1. Preparation for encryption
   - Generate the key matrix for the substitution (step 1) using the **secret** keyword W
   - Select a random place (a kind of sessionkey) in the **secret** book
   - Generate two permutation keys for the DTP (step 2) using the found word sequence in the book
2. Perform the encryption
   - Step 1: Substitution: Apply the key matrix to the given plaintext M_0 and generate M_1
   - Step 2: Transposition: Apply the permutation keys R_1 and R_2 to M_1 and generate M_2
   - Step 3: Prepare the telegram to be transferred
     - Create the key-id from the chosen place in the book
     - Create the key-group from the key-id and the **secret** number Z
     - Generate the telegram by merging the doubled key-group with the ciphertext M_2

Example (2/17) – Overview

3. Preparation for decryption
   - Generate the key matrix using W (same as in the preparation for encryption)
   - Reconstruct the key-id using Z in order to get the place in the book
   - Generate the two permutation keys (grids) from the word sequences found at the place in the book (same as in the preparation for encryption)
4. Perform the decryption
   - Step 1: Revert the 2^{nd} transposition, then revert the 1^{st} transposition
   - Step 2: Revert the substitution and get the plaintext

Example (3/17) – Preparation: Generation of the Key Matrix for Step 1

Agreement of the initial parameters (initial keys)

First, sender and recipient agree on the secret parameters: a keyword W (for the substitution step), a book (for the permutation step), and a 5-digit key number Z (encrypting the place in the book which is used find the permutation grids).
In our example: the book 'Mockingjay' from S. Collins (there are different editions, but all necessary information is given here), the word W = "Katniss", and the number Z = 67914.

Example (4/17) – Preparation: Generation of the Key Matrix for Step 1

Generation of the key matrix using W

Enter all different characters of W into a $3 * 10$ matrix and then all remaining characters of the alphabet (without "J"). In the first row two fields are left empty: Their column numbers serve as labels for the two other rows. The matrix contains 28 different characters. You need the matrix to substitute the characters with digits.

	0	1	2	3	4	5	6	7	8	9
	K	A	T	N	I	S	B	C		
8	D	E	F	G	H	L	M	O	P	Q
9	R	U	V	W	X	Y	Z	zs	.	,

Example (5/17) – Preparation: Generation of the Permutation Keys (Grids) for Step 2

Usage of the book

To generate the grids R_1 and R_2 we choose 10 words from an arbitrary page and row of the book, here page 133, row 04: "*the paint that cannot cover the bags under his eyes*". The first five words "*the paint that cannot cover*" form R_1, the second five form R_2:

R_1: (Length of the 1^{st} word sequence = number of columns = 23)

t	h	e	p	a	i	n	t	t	h	a	t	c	a	n	n	o	t	c	o	v	e	r
18	8	6	16	1	10	11	19	20	9	2	21	4	3	12	13	14	22	5	15	23	7	17

R_2: (Length of the 2^{nd} word sequence = number of columns = 19)

t	h	e	b	a	g	s	u	n	d	e	r	h	i	s	e	y	e	s
17	9	4	2	1	8	14	18	12	3	5	13	10	11	15	6	19	7	16

The numbers in the respective second row correspond to the numbering of the characters in the order of their occurrence in the alphabet, starting with 1.

Example (6/17) – Properties of the Permutation Keys (Grids)

For the grids R_1 and R_2 in GRANIT the following must hold:

- Number of columns $\geqslant 15$
- Number of columns in $R_1 \neq$ number of columns in R_2
  In case of equal number of columns the first character of the sequence in R_2 will be added at the end of R_2.

Remark: Since the two word sequences normally are different, the length normally will also be different. Here, R_1 has a length of 23 and R_2 a length of 19, so both conditions hold.

In step 2, R_1 and R_2 are used as permutation keys: R_1 for the first transposition, R_2 for the second one.

Example (7/17) – Step 1 (Substitution): Generating the Modified Plaintext M_1

After creating the keys during the "preparation", now they are applied for encryption.

The given example plaintext M_0 is: "*Projekt Edelweiß gefährdet? Neuer Termin am 15. Juni 18 Uhr, Treffpunkt bleibt.*"

a) Editing the plaintext M_0 with the help of the codebook: Using the codebook we substitute some words of the plaintext with 3-digit numbers. This way, we get M_{1a} = "*Projekt Edelweiß gefährdet? Neuer 776 am 15. Juni 18 Uhr, 785 punkt bleibt.*"

Example (8/17) – Step 1 (Substitution): Generating the Modified Plaintext M_1

b) Restricting the plaintext to the 28 characters of the matrix:
In M_{1a}, we substitute every 'J' with 'II', every German umlaut with its 2 letters, every 'ß' with 'ss', and '?' with '..'.
All punctuation marks which are not necessary for the understanding of the text are left out.
Then we write the symbol 'zs' in front of and after every number in M_{1a} that is made of digits.
M_{1b} = "*proiiekt edelweiss gefaehrdet.. neuer zs 776 zs am zs 15 zs iiuni zs 18 zs uhr, zs 785 zs punkt bleibt.*"

c) Every digit of numbers NOT originating from the codebook is tripled:
M_{1c} = "*proiiekt edelweiss gefaehrdet.. neuer zs 776 zs am zs 111555 zs iiuni zs 111888 zs uhr, zs 785 zs punkt bleibt.*"

Example (9/17) – Step 1 (Substitution): Generating the Modified Plaintext M_1

d) With the key matrix we substitute the characters in M_{1c} (e.g. the symbol 'zs' becomes 97):

M_{1d} = "*88 90 87 4 4 81 0 2 81 80 81 85 93 81 4 5 5 83 81 82 1 81 84 90 80 81 2 98 98 3 81 91 81 90 zs 776 zs 1 86 zs 111 555 zs 4 4 91 3 4 zs 111 888 zs 91 84 90 99 zs 785 zs 88 91 3 0 2 6 85 81 4 6 2 98*".

With this we get the modified plaintext M_1 as input for the double-column transposition in step 2 (spaces are ignored):

M_1 = "*889087448102818081859381455838182181849080812989838191819097776971869711155597449134971118889791849099977859788913026858146298*"

Example (10/17) – Step 2: Performing the 1st Transposition

Inserting the modified plaintext M_1 into the grid R_1

We enter the modified plaintext M_1 (numerical sequence) line by line and from the left to the right into R_1:

18	8	6	16	1	10	11	19	20	9	2	21	4	3	12	13	14	22	5	15	23	7	17
8	8	9	0	8	7	4	4	8	1	0	2	8	1	8	0	8	1	8	5	9	3	8
1	4	5	5	8	3	8	1	8	2	1	8	1	8	4	9	0	8	0	8	1	2	9
8	9	8	3	8	1	9	1	8	1	9	0	9	7	7	7	6	9	7	1	8	6	9
7	1	1	1	5	5	5	9	7	4	4	9	1	3	4	9	7	1	1	1	8	8	8
9	7	9	1	8	4	9	0	9	9	9	7	7	8	5	9	7	8	8	9	1	3	0
2	6	8	5	8	1	4	6	2	9	8	9	4	9	5								

Applying the 5-package rule for R_1

The number of digits in R_1 has to be divisible by 5. Also R_2 must not have a full last line, that means the number of digits in R_1 must not be divisible by the number of columns of R_2 (19 in our example). Additionally, for better security the remainder shouldn't be too small. Hence, padding characters (arbitrary filling characters) are added at the end, in our case: XY = 9495. So we raise the number of characters from 126 to 130.

Example (11/17) – Step 2: Performing the 2^{nd} Transposition

Reading the characters of R_1 column by column (with increasing column number) we get the input for the 2^{nd} transposition:
M_2 = "*88858801...*"

Entering the result M_2 of the 1^{st} transposition into the grid R_2:

17	9	4	2	1	8	14	18	12	3	5	13	10	11	15	6	19	7	16
8	8	8	5	8	8	0	1	9	4	9	8	1	8	7	3	8	9	8
1	9	1	7	4	8	0	7	1	8	9	5	8	1	9	8	3	2	6
8	3	8	4	9	1	7	6	1	2	1	4	9	9	7	3	1	5	4
1	4	8	9	5	9	4	8	4	7	4	5	5	0	9	7	9	9	8
0	6	7	7	5	8	1	1	9	0	5	3	1	1	5	8	9	9	8
0	8	1	8	7	9	2	4	1	1	9	0	6	8	8	8	7	9	2
2	8	0	9	7	9	1	8	9	1	8	9	1	8	8	1			

Then we read the characters column by column in the order of the numbering again and divide them into groups of 5.

Example (12/17) – Step 3: Transmitting Page Number, Row and Ciphertext

At last, we form the key-group (here 13304) out of the arbitrary chosen position in the book, that means out of the 3 digits long number of the page (133) and the 2 digits long one of the row (04), and sum the single digits modulo 10 with Z:

$$\begin{array}{rr} & 13304 \\ + & 67914 \\ \hline & 70218 \end{array}$$

Adding this **key-id** (German: **Kenngruppe**) in front of the ciphertext twice, we get the telegram:

```
70218 70218 84955 77574 97894 82701 18188
71099 14598 38378 81925 99988 19899 89346
88189 51618 19018 89114 91985 45309 00741
21797 95888 64882 81810 02176 81488 31997
```

The recipient needs this key-id to find the 10 words from the book by which he can generate the same permutation keys the sender used.

Example (13/17) – Decrypting Preparation: Reconstructing the Key-group

First, we generate the key matrix using W – exactly as for the encryption.

Then we reconstruct the key-group out of the key-id in order to get the used page and row of the book:
Subtract the single digits of Z from those of the key-id modulo 10:

```
    70218
 -  67914
-------
    13304
```

The key-id is the place in the book to find the word sequences for the two grids R_1 and R_2: "*the paint that cannot cover the bags under his eyes*"
With this we can – exactly as for the encryption – create the two grids (see page 11).

Example (14/17) – Decrypting Preparation: Reconstructing the Grids

a) Initially, delete the two key-ids from the telegram.

b) Draw the schemes of the grids as follows:
Divide the number of digits of the telegram by the number of columns of the grid – the quotient stands for the number of full rows in the grid, and the remainder quantifies the number of filled fields in the last row which is not completely filled:

- $130/23 = 5$ remainder 15
  So, R_1 has 5 full rows, and 15 filled fields in the 6^{th} row.
- $130/19 = 6$ remainder 16
  So, R_2 has 6 full rows, and 16 filled fields in the 7^{th} row.

Example (15/17) – Decrypting Step 1: Reverting the 2nd Transposition

Enter the digits of the telegram (84955 77574 ...) into R_2 column by column and top down according to the numbering of the columns. In doing so, the columns 1 - 16 each get 7 digits, and the columns 17 - 19 each get 6 digits – as we computed on the previous page:

17	9	4	2	1	8	14	18	12	3	5	13	10	11	15	6	19	7	16
8	8	8	5	8	8	0	1	9	4	9	8	1	8	7	3	8	9	8
1	9	1	7	4	8	0	7	1	8	9	5	8	1	9	8	3	2	6
8	3	8	4	9	1	7	6	1	2	1	4	9	9	7	3	1	5	4
1	4	8	9	5	9	4	8	4	7	4	5	5	0	9	7	9	9	8
0	6	7	7	5	8	1	1	9	0	5	3	1	1	5	8	9	9	8
0	8	1	8	7	9	2	4	1	1	9	0	6	8	8	8	7	9	2
2	8	0	9	7	9	1	8	9	1	8	9	1	8	8	1			

Example (16/17) – Decrypting Step 1: Reverting the 1st Transposition

Now we read the digits from R_2 row by row and from left to right (88858...), and enter them into R_1 column by column and top down according to the numbering of the columns. In doing so, the columns 1 - 10 each get 6 digits, and the columns 11 - 23 each get 5 digits – as we computed on page 20:

18	8	6	16	1	10	11	19	20	9	2	21	4	3	12	13	14	22	5	15	23	7	17
8	8	9	0	8	7	4	4	8	1	0	2	8	1	8	0	8	1	8	5	9	3	8
1	4	5	5	8	3	8	1	8	2	1	8	1	8	4	9	0	8	0	8	1	2	9
8	9	8	3	8	1	9	1	8	1	9	0	9	7	7	7	6	9	7	1	8	6	9
7	1	1	1	5	5	5	9	7	4	4	9	1	3	4	9	7	1	1	1	8	8	8
9	7	9	1	8	4	9	0	9	9	9	7	7	8	5	9	7	8	8	9	1	3	0
2	6	8	5	8	1	4	6	2	9	8	9	4	9	5								

Again we read the digits row by row and get the text M_1':

M_1' = "*889087448102818081859381455838182181849080812989838191819097776971869711155597449134971118889791849099977859788913026858146298949 5*"

Example (17/17) – Decrypting Step 2: Substitution and Getting the Plaintext

Substituting the digits using the key matrix (88 = p, 90 = r, 87 = o, 4 = i etc.) we get:
"*proiiektedelweissgefaehrdet..*
neuerzs776zsamzs15zsiiunizs18zsuhr,zs785zspunktbleibt.xy"
Out of this intermediate plaintext we determine the actual plaintext using the codebook (and delete the obvious padding 'xy' at the end):
"*Projekt Edelweiß gefährdet?*
Neuer Termin am 15. Juni 18 Uhr, Treffpunkt bleibt."

In this example the solution to be entered would be date and time:
"15JUNI18UHR"

Links

- Website with nearly all codes and encryption mechanisms used in the former GDR – (only German): http://scz.bplaced.net/m.html#dwa
- An interesting thread about double-column transposition – (only German): http://www.buha.info/showthread.php?10834-Verfahren-des-Doppelw%FCrfel/page2