MysteryTwister C3

THE CRYPTO CHALLENGE CONTEST

HANDYCIPHER MADE IN LOVE – PART 2

Author: Bruce Kallick

May 2019

Introduction (1/4)

In solving George Theofanidis's recent "HANDYCIPHER MADE IN LOVE" challenge it's helpful to review the structure underlying this early version of Handycipher. The cipher operates on plaintext strings over the ordered 31-character alphabet A = {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 , . - ? ^} where ^ is used for the space character in plaintexts, and generates ciphertext strings over A*, the 40-character alphabet derived from A by omitting ^ and adding the ten decimal digits ∅-9.

Introduction (2/4)

- Some permutation of the 40 characters of A* and the ^ is chosen as a secret shared key K, and the 40 non-space characters of K are displayed as a 5 x 8 table T.
- A 31-character subkey P is derived from K by omitting the decimal digits.
- Plaintext characters are first encrypted into binary numbers by a simple substitution cipher using the subkey P.

Introduction (3/4)

- These binary numbers are then encrypted into k-tuples ($1 \leqslant k \leqslant 5$) of the characters contained in the first five columns of T by a nondeterministic homophonic substitution cipher using the twenty rows, columns, and diagonals of the matrix composed of those five columns.
- These k-tuples are then randomly interspersed with null characters chosen from the last three columns of T.
- In decryption, the ciphertext is first stripped of all null characters, the homophonic substitution is inverted, and then the simple substitution is inverted.

Introduction (4/4)

On reflection it should be clear that when Bob used the wrong key to decrypt the ciphertext produced with the correct key, only the simple substitution inversion went wrong (because he used an incorrect subkey) – i.e., the wrong plaintext produced with the wrong key is a simple substitution of the correct plaintext Alice encrypted.

So it turns out that knowing the correct contents of just the first five columns of the correct key's associated table and the position of ^ within the key is enough information to reconstruct the correct plaintext from the correct ciphertext.

Challenge (1/5)

Part Two of the challenge is identical to Part One except that the solution you are asked to provide is the correct key that Alice used to produce the ciphertext Bob received.

The wrong key Alice sent to Bob is
ONI6AHJLGTSDVMQR4B^1ZEUY2C8K,-357WPXFØ9.?
which is included in the file "wrong_key_txt" in the additional zip archive, and has the associated table:

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

Challenge (2/5)

Since you're given that Alice only rearranged the 15 letters in the last three columns, your task is to find a key whose associated table is:

O	N	I	6	A	_	_	_
G	T	S	D	V	_	_	_
4	B	1	Z	E	_	_	_
C	8	K	,	-	_	_	_
W	P	X	F	∅	_	_	_

where you are to fill in the blank positions with some permutation of those 15 letters. You will also have to determine where the ^ symbol occurs in the key.

Challenge (3/5)

Since the subkey P is obtained by omitting all decimal digits, it's clear that they just serve as placeholders and only the sequencing of the other 30 letters matters. Accordingly, any rearrangement of those 15 letters that preserves that sequencing will produce the same ciphertext and so there are many correct keys that Alice could have used to produce the ciphertext Bob received.

Challenge (4/5)

Please submit your solution in the following canonical form:

1. Replace each of the five decimal digits contained in the last three columns by the asterisk symbol (*).
2. Shift all asterisks in any one row of the last three columns to the end of that row.
3. Reconstitute a 41-character key from the resulting table, inserting the ^ character so that it will occupy the same position in the derived subkey as it does in the subkey derived from the wrong key.

Challenge (5/5)

For example, the wrong key above would be changed into canonical form as:

O	N	I	6	A	H	J	L
G	T	S	D	V	M	Q	R
4	B	1	Z	E	U	Y	*
C	8	K	,	-	*	*	*
W	P	X	F	Ø	.	?	*

and then submitted as:

ONI6AHJLGTSDVMQR4B^1ZEUY*C8K,-***WPXFØ.?*

Additional Files (1/2)

The additional zip archive contains the following files:

- wrong_key.txt
  - ➥ Contains the wrong key.
- correct_ciphertext.txt
  - ➥ Contains the ciphertext encrypted by Alice with the correct key.
- wrong_plaintext.txt
  - ➥ Contains the plaintext decrypted by Bob with the wrong key.

Additional Files (2/2)

- MTC3_Handycipher_Description.pdf
  - ➥ Detailed explanation of the early version of Handycipher used in this challenge, which employs only 15 nulls rather than the 25 employed by later versions used in subsequent Handycipher challenges.
- handycipher.zip
  - ➥ Python code and test files for the early version of Handycipher used in this challenge. The Python Handycipher implementation is not required to solve the challenge, but it might be useful for verifying the correctness of a solution to be submitted.