Transposition

1. Permutation Cipher
2. Column Transposition
3. Double Transposition

I am using the descriptions from Classical and Contemporary Cryptology by Spillman

1. Permutation

A permutation is a reordering of a list of integers. If you look in math books, usually they're reorder 1, 2, 3, …, n, but for computer science, I usually teach by reordering 0,1,2…, n-1 because arrays are 0-based in most languages and most implementations we would do for class are on the computer

N = 6

Example permutation: 3, 1, 6, 2, 4, 5 (1 based)

2, 0, 5, 1, 3, 4

F(0) = 2, f(1) = 0, f(2) = 5, f(3) = 1, f(4) =3, f(5) = 4

Plaintext: WEWENT TOTHES TOREHI

Ciphertext: EEWNTW OHTEST OETHIR

$f^{-1}(2) = 0$, $f^{-1}(0) = 1$, $f^{-1}(5) = 2$, $f^{-1}(1) = 3$, $f^{-1}(3) = 4$, $f^{-1}(4) = 5$ → (made output of old function input, and input of old function output)

$f^{-1}(0) = 1$, $f^{-1}(1) = 3$, $f^{-1}(2) = 0$, $f^{-1}(3) = 4$, $f^{-1}(4) = 5$, $f^{-1}(5) = 2$

Decryption key is [1, 3, 0, 4, 5, 2]

Transposition

Exactly phase 2 of ADFGVX

Secret key: FLUTE (sort in alpha order and under each letter put its rank. If there are ties, break them going left to right.)

F	L	U	T	E
2	3	5	4	1
T	H	I	S	I
S	A	L	I	T
T	L	E	M	E
S	S	A	G	E
T	O	R	E	A
D	NONE	NONE	NONE	NONE

Read the columns in numerical order

ITEEATSTSTDHALSOSIMGEILEAR

Guess at the keyword length. So if msg is 26 letters and we guess a keylength of 5, then most of the columns have 5 letters and one column has 6 letters.

Make sure you know how to decrypt regular transposition

Double Transposition

Use two keywords instead of one. Encrypt with the first key word, then take that temporary ciphertext, and encrypt it with the second keyword.

Does doing this twice add to the security?

Let's look at some of our old systems:

1) Shift – add some number to each letter. If we do two shifts, ultimately two numbers add to another number mod 26, so two shifts is NOT better than one, because any two shifts can be replaced with an equivalent shift.
2) Affine – you proved for your homework that the composition of two affine encryption functions produces a single valid affine encryption, so this is NOT better either.

3) Substitution – each letter maps to another one. When we compose two substitutions, we just get a single substitution.
4) Vigenere – (1) same keyword length (because we can just add the two keywords to give us the resultant key), (2) different keyword length –

MAPMAPMAPMAP

HOMEHOMEHOME

With a keyword of size 3 and one of size 4, I have created a resultant keyword of size 12, which is presumably a larger keyspace, so yes, Vigenere used twice can make something more secure.

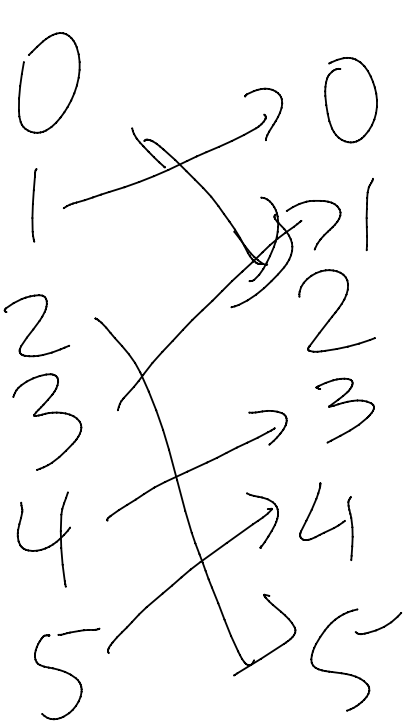
Transposition ends up being similar to Vigenere in this analysis. Two transpositions, can move a character around to a greater number of locations in the ciphertext than a single transposition. Complexity wise, it's likely that two transpositions of different keyword lengths is as complex as using a single keyword of the length of the product…

Beauty of this codewise is that you can just call the same function twice!!!

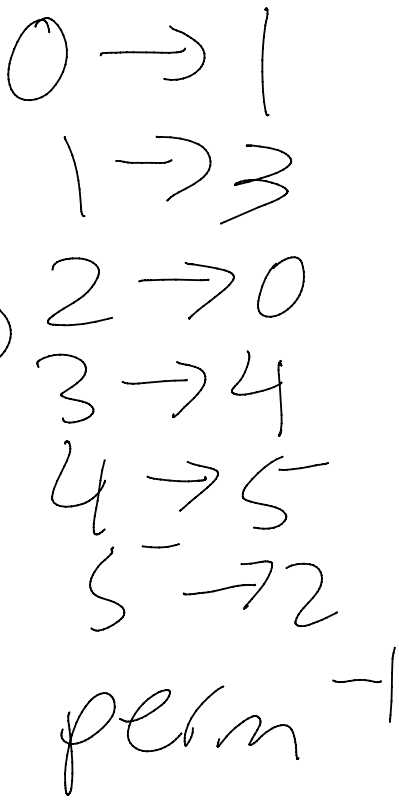
Open question for code: How would you code transposition if there was NO PADDING ADDED?

Transposition (Extra Drawings) 9/21/2020

Friday, September 18, 2020 3:46 PM

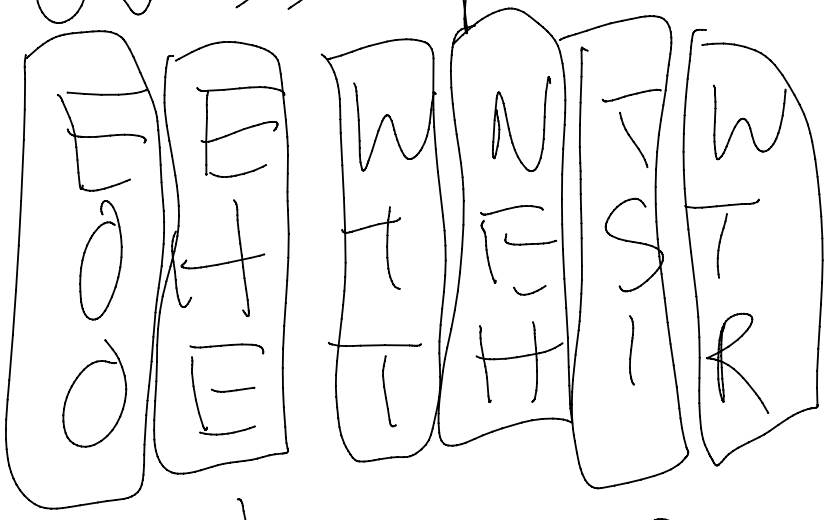


f^{-1} ⟹ do grows backwards



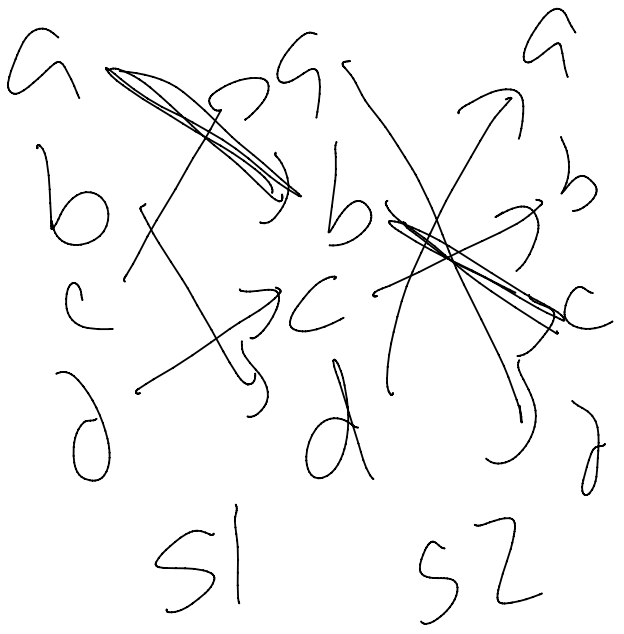
Breaking ① Guess perm

Perm length



randomly swap

columns until you see words. Use language patterns!



S1 S2

S1 followed by S2

a → c
b → a
c → d
d → b