Autoperm Cipher

Alastair Horn

December 14, 2019

Introduction

This idea for a cipher popped into my head recently.

What follows is a short explanation of the cipher. Accompanying code is provided for enciphering and deciphering.

The Cipher

We are given as plaintext a string of letters A-Z which will here be represented by numbers 1-26 in alphabetical order. Let the plaintext be m letters in length, denoted by $a_1, a_2, \ldots, a_m$, and the target ciphertext be $b_1, b_2, \ldots, b_m$.

The keys are two permutations σ_0, τ_0 of $\{1, 2, \ldots, 26\}$. The keys and the plaintext together produce sequences of permutations as follows:

$$\begin{aligned}\sigma_{n+1} &= \sigma_n \circ (a_{2n}\ a_{2n+1})\\ \tau_{n+1} &= \tau_n \circ (a_{2n}\ a_{2n+1})\end{aligned}$$

for all n.

Round brackets are used here to write cycles, and $\circ$ is used to compose functions.

And of course we have a way to generate ciphertext from the plaintext and the sequences of permutations:

$$\begin{aligned}b_{2n} &= \sigma_n(a_{2n})\\ b_{2n+1} &= \tau_n(a_{2n+1})\end{aligned}$$

Cryptanalysis and Known Weaknesses

Over short distances within the plaintext, letters appearing more than once are likely to produce a similar result in the ciphertext. Consider for example some plaintext letters $a_{2n}, a_{2n+1}, a_{2n+2}, a_{2n+3}$, with $a_{2n+1} = a_{2n+2}$. We have

$$\begin{aligned}b_{2n} &= \sigma_n(a_{2n})\\ b_{2n+1} &= \tau_n(a_{2n+1})\\ b_{2n+2} &= \sigma_{n+1}(a_{2n+2})\\ &= (\sigma_n \circ (a_{2n}\ a_{2n+1}))(a_{2n+1})\\ &= b_{2n}\end{aligned}$$

and other similar formations of letters cropping up again can produce similar arrangements.