Encryption Algorithms

Transposition ciphers rearrange characters or bits of plaintext to produce ciphertext.

Example: The key to the cipher is the number of rows and columns of a matrix. Encipher a message by writing the plaintext into the matrix by rows and reading the ciphertext out of the matrix by columns.

There is a fixed period, d, say. If we assume all $d!$ permutations to be equally likely, then one can prove that the the shortest length of ciphertext needed to break it is

$$N = \frac{d \log_2(d/e)}{3.2} \approx 0.3 d \log_2(d/e).$$

Example: With a 3×9 matrix we have $d = 27$ and $N = 27.9$.

Use digrams to crack transposition ciphers. The process is called *anagramming*.

Substitution Ciphers

Alphabets: Plain $\{m_i\}$, Cipher $\{c_i\}$.

Replace (blocks of) characters by other characters. Four types:

1. Simple: Replace m_i by c_i. Permute alphabet.

2. Homophonic: Replace m_i by a random one of several possible c_j.

3. Polyalphabetic: Use multiple maps from the plaintext alphabet to the ciphertext alphabet.

4. Polygram: Make arbitrary substitution for blocks of characters.

Use frequency counts to distinguish Transposition ciphers from Substitution ciphers.

1. Simple: Replace m_i by c_i. Write $f(m_i) = c_i$.

Example. Caesar cipher–rotate the alphabet: $f(m) = (m + k) \bmod n$, where n is the alphabet size. Then one can show that the shortest length of ciphertext needed to break it is $(\log_2 26)/3.2 \approx 1.5$ letters.

If all $n!$ permutations of the alphabet are equally likely (the best case) in a simple substitution cipher, and the language is English (with $n = 26$), then the shortest length of ciphertext needed to break it would be $\log(26!)/3.2 \approx 27.6$. This is the case for the Cryptoquote in the *Exponent*.

These ciphers may be broken with frequency analysis and trial and error.

An *affine cipher*, $f(m) = (am+b) \bmod n$, guess some two-letter pairs and solve two congruences in two unknowns a and b.

2. Homophonic: Replace m_i by a random one of several possible c_j.

To confound the frequency analysis which succeeds so well for simple substitution ciphers, one might use a ciphertext alphabet larger than the plaintext alphabet and assign each plaintext letter a to a subset (*homophone*) $f(a)$ of the ciphertext alphabet. To permit deciphering, require $f(a) \cap f(b) = \emptyset$ when $a \neq b$. Encipher each m_i in the plaintext as a randomly chosen $c_j \in f(m_i)$.

Usually, the ciphertext alphabet is much larger than the plaintext alphabet and the size of $f(a)$ is proportional to the frequency of occurrence of a in normal English. Then the letters of the ciphertext alphabet have a uniform distribution in the ciphertext. Use digrams to break.

One can define f via a standard text using the number of an instance of the letter as its cipher.

3. Polyalphabetic: Use multiple maps f_i from the plaintext alphabet to the ciphertext alphabet.

Encipher $M = m_0 m_1 \ldots$ as $C = f_0(m_0) f_1(m_1) \ldots$.

Let n be the length of the alphabet. The sequence $\{f_i\}$ may be periodic, perhaps defined by a *keyword* $K = k_0 \ldots k_{d-1}$.

Example: Vigenère cipher: $f_i(a) = (a + k_i) \bmod n$.

M: `RENA ISSA NCE`

K: `BAND BAND BAN`

C: `SEAD JSFD OCR`

Example: Beaufort cipher: $f_i(a) = (k_i - a)$ mod n.

If the period of the key is d, then one can show that the the shortest length of ciphertext needed to break it is

$$\log_2(n^d)/3.2 = (d/3.2)\log_2 n.$$

For English (with $n = 26$), this is $d \log_2(26)/3.2 \approx 1.47d$.

The way to break a periodic polyalphabetic substitution cipher is to first find the period d. Then break it by solving d interlaced simple substitution ciphers by letter frequency, guessing words or using digraphs.

There are two basic methods to find the period of a periodic polyalphabetic substitution cipher.

Kasiski Method: Look for repetitions in cipher text. The difference between the starting locations of a repeated ciphertext might be a multiple of the period d. Look at the gcd of some of these differences.

The Index of Coincidence Method of William F Friedman: Measure frequency variations of letters to guess the period d. Let $\{a_0, a_1, \ldots, a_{n-1}\}$ be the (plain or ciphertext) alphabet. Let F_i be the frequency of occurrence of a_i in a ciphertext of length N. Define the Index of Coincidence as

$$IC = \left(\sum_{i=0}^{n-1} \frac{F_i(F_i - 1)}{2} \right) / \left(\frac{N(N-1)}{2} \right).$$

Then IC represents the probability that two letters chosen at random in the ciphertext are the same.

One can estimate IC theoretically in terms of the period d. For English and a polyalphabetic cipher with period d, the expected value of IC is

$$\frac{1}{d}\,\frac{N-d}{N-1}(0.066)+\frac{d-1}{d}\,\frac{N}{N-1}(0.038).$$

Note: $1/26 \approx 0.038$.

Guess d by comparing the IC with this table: (d, IC): (1, 0.066), (2, 0.052), (3, 0.047), (4, 0.045), (5, 0.044), (10, 0.041), (infinity, 0.038).

The IC tells you the approximate size of the period d. The Kasiski method complements the IC method by telling you a number (the gcd) that probably divides d.

Example. If $IC = 0.043$, then d is probably 6 or 7. If a Kasiski analysis finds several examples of repeated ciphertext occurring at multiples of 3 in the position of the letters, then d is probably a multiple of 3. These two pieces of information suggest that $d = 6$.