5 The Cipher Disk Algorithm

Mathematical Notation

Take the alphabet $\Sigma = \{s_0, \ldots, s_{n-1}\}$, and interpret (or code) it as the additive group of the ring $\mathbb{Z}/n\mathbb{Z}$. The key $(\sigma, k) \in \mathcal{S}(\Sigma) \times \Sigma^l$ of a disk cipher consists of a primary alphabet (represented by the permutation σ) and a keyword $k = (k_0, \ldots, k_{l-1}) \in \Sigma^l$. Our notation for the corresponding encryption function is

$$f_{\sigma,k}\colon \Sigma^* \longrightarrow \Sigma^*$$

Special case: The BELLASO cipher with keyword k is $f_{\varepsilon,k}$ where $\varepsilon \in \mathcal{S}(\Sigma)$ denotes the identity permutation.

The Alphabet Table

We arrange the alphabets for the polyalphabetic substitution in form of the usual table:

s_0	s_1	s_2	$\ldots$	s_{n-1}
t_0	t_1	t_2	$\ldots$	t_{n-1}
t_1	t_2	t_3	$\ldots$	t_0
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
t_{n-1}	t_0	t_1	$\ldots$	t_{n-2}

where $t_i = \sigma s_i$ for $0 \le i \le n-1$.

Note that whenever we refer to an alphabet table we implicitely use an order on the alphabet Σ. This order manifests itself by indexing the letters as $s_0, \ldots, s_{n-1}$.

The Encryption Function

Now we encrypt a text $a = (a_0, a_1, a_2, \ldots) \in \Sigma^r$ using this notation. Let $a_i = s_q$ and $k_i = t_p$ as letters of the alphabet. Then we read the ciphertext letter c_i off from row p and column q of the table:

$$c_i = t_{p+q} = \sigma s_{p+q} = \sigma(s_p + s_q) \quad \text{[sums in } \mathbb{Z}/n\mathbb{Z}\text{]}.$$

We have

$$k_i = t_p = \sigma(s_p), \quad s_p = \sigma^{-1}(k_i), \quad \text{hence } c_i = \sigma(a_i + \sigma^{-1}(k_i)).$$

If we denote by f_σ the monoalphabetic substitution corresponding to σ, then this derivation proves:

Theorem 1 *The disk cipher $f_{\sigma,k}$ is the composition (or "superencryption") of the* BELLASO *encryption $f_{\varepsilon,k'}$, where $k' = f_\sigma^{-1}(k)$, with the monoalphabetic substitution f_σ,*

$$f_{\sigma,k} = f_\sigma \circ f_{\varepsilon,k'}$$

Algorithm

The naive straightforward algorithm for the disk cipher is

- Take the next plaintext letter.
- Take the next alphabet.
- Get the next ciphertext letter.

From Theorem 1 we derive an algorithm that is a bit more efficient:

1. Take $k' = f_\sigma^{-1}(k)$, in coordinates $k'_i = \sigma^{-1}(k_i)$ for $0 \leq i < l$.
2. Add a and (the periodically extended) k' over $\mathbb{Z}/n\mathbb{Z}$, and get b, in coordinates $b_j = a_j + k'_{j \bmod l}$
3. Take $c = f_\sigma(b) \in \Sigma^r$, in coordinates $c_j = \sigma(b_j)$.

A Perl program implementing this algorithm is on the web page `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/porta.pl`, the corresponding program for decryption on `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/portadec.pl`. They can be called online from the pages `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/2_Polyalph/portaenc.html` and `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/2_Polyalph/portadec.html`