8 Similarity of Ciphers

Let Σ be an alphabet, $M \subseteq \Sigma^*$ a language, and K a finite set (to be used as keyspace).

Definition [Shannon 1949]. Let $F = (f_k)_{k\in K}$ and $F' = (f'_k)_{k\in K}$ be ciphers on M with encryption functions

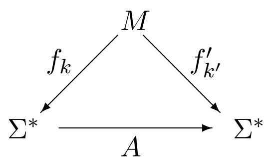
$$f_k, f'_k : M \longrightarrow \Sigma^* \quad \text{for all } k \in K.$$

Let $\tilde{F}$ and $\tilde{F}'$ be the corresponding sets of encryption functions. Then F is called **reducible** to F' if there is a bijection $A: \Sigma^* \longrightarrow \Sigma^*$ such that

$$A \circ f \in \tilde{F}' \quad \text{for all } f \in \tilde{F}.$$

That is, for each $k \in K$ there is a $k' \in K$ with $A \circ f_k = f'_{k'}$, see the diagram below.

F and F' are called **similar** if F is reducible to F', and F' is reducible to F.



Application. Similar ciphers F and F' are cryptanalytically equivalent—provided that the transformation $f \mapsto f'$ is efficiently computable. That means an attacker can break F if and only if she can break F'.

Examples

1. **Reverse** Caesar. This is a monoalphabetic substitution with a cyclically shifted exemplar of the reverse alphabet `Z Y ... B A`, for example

```
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
W V U T S R Q P O N M L K J I H G F E D C B A Z Y X
```

We have $K = \Sigma = \mathbb{Z}/n\mathbb{Z}$. Let $\rho(s) := n - s$ the reversion of the alphabet. Then encryption is defined by

$$f_k(s) := k - s \quad \text{for all } k \in K.$$

This encryption function is involutory: $f_k \circ f_k(s) = k - (k - s) = s$. The ordinary Caesar encryption is

$$f'_k(s) := k + s \quad \text{for all } k \in K.$$

Then

$$\rho \circ f_k(s) = \rho(k - s) = n + s - k = (n - k) + s = f'_{n-k}(s),$$

whence $\rho \circ f_k = f'_{\rho(k)}$. Because also the corresponding converse equation holds CAESAR *and Reverse* CAESAR *are similar.*

2. **The** BEAUFORT **cipher** [SESTRI 1710]. This is a periodic polyalphabetic substitution with a key $k = (k_0, \ldots, k_{l-1}) \in \Sigma^l$ (periodically continued):

$$f_k(a_0, \ldots, a_{r-1}) := (k_0 - a_0, k_1 - a_1, \ldots, k_{r-1} - a_{r-1}).$$

Like Reverse CAESAR it is involutory. The alphabet table over the alphabet $\Sigma = \{\mathtt{A,\ldots,Z}\}$ is in Figure 1. Compare this with TRITHEMIUS-BELLASO encryption:

$$f'_k(a_0, \ldots, a_{r-1}) := (k_0 + a_0, k_1 + a_1, \ldots, k_{r-1} + a_{r-1}).$$

Then as with Reverse CAESAR we have $\rho \circ f_k = f'_{\rho(k)}$, and in the same way we conclude: *The* BEAUFORT *sipher is similar with the* TRITHEMIUS-BELLASO *cipher.*

3. **The Autokey cipher.** As alphabet we take $\Sigma = \mathbb{Z}/n\mathbb{Z}$. We write the encryption scheme as:

$$\begin{array}{rcl|rcl|rcl}
c_0 & = & a_0 + k_0 & & & & & & \\
c_1 & = & a_1 + k_1 & & & & & & \\
\vdots & & & & & & & & \\
c_l & = & a_l + a_0 & c_l - c_0 & = & a_l - k_0 & & & \\
\vdots & & & & & & & & \\
c_{2l} & = & a_{2l} + a_l & c_{2l} - c_l & = & a_{2l} - a_0 & c_{2l} - c_l + c_0 & = & a_{2l} + k_0 \\
\vdots & & & & & & & &
\end{array}$$

Let

$$A(c_0, \ldots, c_i, \ldots, c_{r-1}) = (\ldots, c_i - c_{i-l} + c_{i-2l} - \ldots, \ldots).$$

In explicit form the i-th component of the image vector looks like:

$$\sum_{j=0}^{\lfloor i \rfloor} (-1)^j \cdot c_{i-jl}.$$

and as a matrix A looks like

$$\begin{pmatrix} 1 & & -1 & & 1 & \\ & \ddots & & \ddots & & \ddots \\ & & 1 & & -1 & \\ & & & \ddots & & \ddots \\ & & & & 1 & \\ & & & & & \ddots \end{pmatrix}$$

Then

$$A \circ f_k(a) = f'_{(k,-k)}(a),$$

where $f'_{(k,-k)}$ is the TRITHEMIUS-BELLASO cipher with key $(k_0, \ldots, k_{l-1}, -k_0, \ldots, -k_{l-1}) \in \Sigma^{2l}$. Hence *the Autokey cipher is reducible to the* TRITHEMIUS-BELASO *cipher with period* twice *the key length.* [FRIEDMAN und SHANNON] The converse is not true, the ciphers are not similar: This follows from the special form of the BELLASO key of an autokey cipher.

Note that A depends only on l. The reduction of the autokey cipher to the TRITHEMIUS-BELASO cipher is noteworthy but practically useless: The encryption algorithm and the cryptanalysis are both more complicated when using this reduction. And the reduction is possible only after the keylength l is known.