4 Rotor Machines

General Description

Rotor machines are electromechanical devices that consist of several rotors in series connection. Figure 1 gives an impression of the electric flow through such a machine.

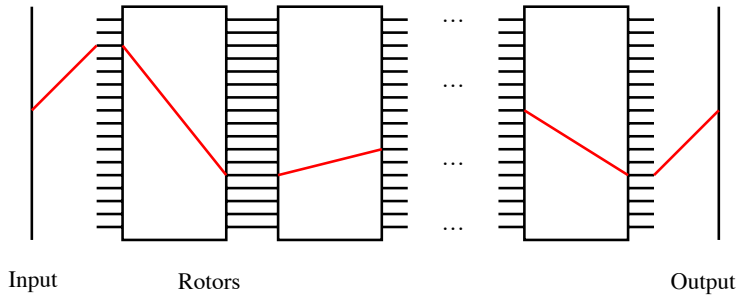

Figure 1: *Rotor machine circuit*

With each input letter the rotors move in individual ways, some by one position, some by several positions, some only after several steps. *The cryptographic security of a rotor machine depends* on the number of rotors, the multitude of different settings, and, in a crucial way, *on the complexity of the rotor movements.*

Operating a Rotor Machine

The operator hits a key on the keyboard that corresponds to the next plaintext letter. This action closes an electric circuit powering a light-bulb that corresponds to the ciphertext letter. Or it powers a type bar that prints the ciphertext letter. The rotors move according to their control logic, in general before the circuit is closed. See the FAQ at `http://www.staff.uni-mainz.de/pommeren/Cryptology/FAQ.html`.

Rotor machines are the state of the art in encryption during the period from 1920 until 1970. The mystic and irregularly rotating wheelwork that makes the desk tremble with each key hit looks very attractive and impresses the general or diplomat who wants to buy security.

Mathematical Description

The following abstract model describes an idealized rotor machine. Concrete historic machines each have their own peculiar details.

As before we identify the alphabet Σ with $\mathbb{Z}/n\mathbb{Z}$, the integers mod n. A rotor machine has the following characteristic parameters:

- A set $R \subseteq \mathcal{S}(\Sigma)$ of $p = \#R$ rotors. Each of these defines a primary alphabet, that is a permutation $\rho_i \in \mathcal{S}(\Sigma)$ that corresponds to the wiring of the rotor.

- A choice $\rho = (\rho_1, \ldots, \rho_q) \in \mathcal{S}(\Sigma)^q$ of q different rotors $\rho_i \in R$. There are $p \cdot (p-1) \cdots (p-q+1)$ choices if we assume that all rotors are differently wired ($q \leq p$). This choice serves as "primary key" and is usually fixed for several messages, say for an entire day.

- A state vector $z = (z_1, \ldots, z_q) \in (\mathbb{Z}/n\mathbb{Z})^q$ that describes the current rotor positions. The initial state $z^{(0)}$ serves as "secondary key" that usually changes with each message. The number of different initial states is n^q. Sometimes it is convenient to map the states to $\mathbb{Z}/n^q\mathbb{Z}$, the integers $\bmod n^q$, using the representation of integers in base n. The state vector $z = (z_1, \ldots, z_q) \in (\mathbb{Z}/n\mathbb{Z})^q$ then corresponds to the integer $\zeta = z_1 \cdot n^{q-1} + \cdots + z_q$.

- A state-transition function

  $$g \colon \mathbb{N} \times \Sigma^q \longrightarrow \Sigma^q$$

  that transforms the state at time i, $z^{(i)}$, to the state at time $i+1$, $z^{(i+1)} = g(i, z^{(i)})$, where "time" is discrete and simply counts the plaintext letters. This function g represents the control logic and is realized for example by more or less complex gear drives. In most rotor machines the state-transition function is independent of the time i.

- The substitution in state z:

  $$\sigma_z := \rho_q^{(z_q)} \circ \cdots \circ \rho_1^{(z_1)} \quad \text{where } \rho_j^{(z_j)} := \tau^{z_j} \circ \rho_j \circ \tau^{-z_j}$$

Ideally the map $\Sigma^q \longrightarrow \mathcal{S}(\Sigma)$, $z \mapsto \sigma_z$ would be injective, that is each state defines a different substitution. Unfortunately no useful general results seem to exist beyond the case $q = 1$ treated in Subsection 2.

Perl programs for encryption and decryption by rotor machines are in the web directory `http://www.staff.uni-mainz.de/pommeren/Cryptology/Classic/Perl/` as `rotmach.pl` and `rotdecr.pl`.

The Key Space

By the description above a key of our idealized rotor machine consists of

- a choice of rotors

- an initial state

Therefore the key space K has

$$\#K = n^q \cdot \frac{p!}{(p-q)!}$$

elements. In a typical example (HEBERN machine) we have $p = q = 5$, $n = 26$, $\#K = 120 \cdot 26^5 = 712882560$, and the effective key length is $d(F) \approx 29.4$. That was good enough in 1920. Today, against an enemy with a computer, this is much too little.

> In fact the HEBERN machine was not good enough even in 1920 because it allows attacks far more efficient than exhaustion.

Encryption and Decryption

The plaintext $a = (a_1, \ldots, a_r) \in \Sigma^r$ is encrypted by the formula

$$c_i = \sigma_{z^{(i)}}(a_i)$$

At full length this formula reads

$$c_i = \tau^{z_q^{(i)}} \circ \rho_q \circ \tau^{z_{q-1}^{(i)} - z_q^{(i)}} \circ \cdots \circ \tau^{z_1^{(i)} - z_2^{(i)}} \circ \rho_1 \circ \tau^{-z_1^{(i)}}(a_i)$$

Decryption follows the formula

$$a_i = \tau^{z_1^{(i)}} \circ \rho_1^{(-1)} \circ \tau^{z_2^{(i)} - z_1^{(i)}} \circ \cdots \circ \tau^{z_q^{(i)} - z_{q-1}^{(i)}} \circ \rho_q^{(-1)} \circ \tau^{-z_q^{(i)}}(c_i)$$

Technically for decryption we simply have to route the current through the machine in the reverse direction, of course interchanging the keyboard and lightbulbs. The sequence of states is identical for encryption and decryption.

The Rotor Machine as a Finite-State Automaton

Figure 2 shows an abstract model of a rotor machine.

Usually the state-transition function is independent of the step i. Then it has the simpler form

$$g \colon \Sigma^q \longrightarrow \Sigma^q$$

This makes the states periodic as shown in the next subsection.

Periods of State Changes

Let M be a finite set with $m = \#M$. We may think of the elements of M as "states". Consider a map ("state transition")

$$g : M \longrightarrow M.$$

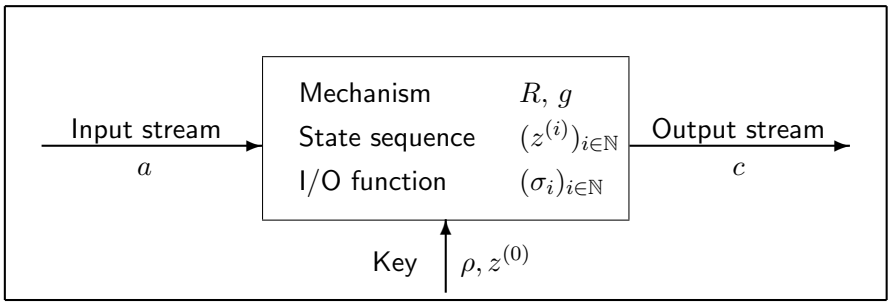

Figure 2: *Rotor machine as finite-state automaton*

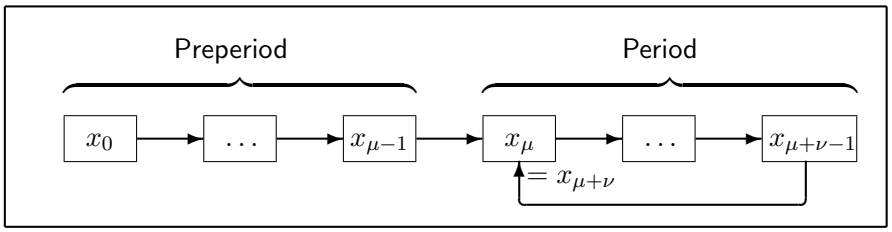

Figure 3: *Period and preperiod*

For each element ("initial state") $x_0 \in M$ we define a sequence $(x_i)_{i\in\mathbb{N}}$ in M by the recursion formula $x_i = g(x_{i-1})$ for $i \geq 1$. After a preperiod of length μ this sequence becomes periodic with a period of ν, see Figure 3, an explanation follows below.

Since M is finite there are smallest integers $\mu \geq 0$ and $\nu \geq 1$ such that $x_{\mu+\nu} = x_\mu$: Take for μ the smallest index such that the element x_μ reappears somewhere in the sequence, and for $\mu+\nu$ the index where the first repetition occurs. Then also

$$x_{i+\nu} = x_i \quad \text{for } i \geq \mu.$$

Obviously $0 \leq \mu \leq m-1$, $1 \leq \nu \leq m$, $\mu+\nu \leq m$. The values $x_0, \ldots, x_{\mu+\nu-1}$ are all distinct, and the values $x_0, \ldots, x_{\mu-1}$ never reappear in the sequence.

Definition: μ is called (length of the) **preperiod**, ν is called (length of the) **period**.