2 Cryptanalytic Approaches to Running-Text Ciphers

Cryptanalysis of running-text ciphers is laborious. There are several approaches that should be combined in practice. Automated procedures are proposed in

> E. Dawson and L. Nielsen: Automated cryptanalysis of XOR plaintext strings. Cryptologia XX (1996), 165–181.
>
> A. Griffing: Solving the running key cipher with the Viterbi algorithm. Cryptologia XXX (2006), 361–367.

The first of these considers running-text ciphers where plaintext and key are combined via binary addition (XOR) instead of addition mod 26. This distinction not essential for the method (but of course for the use of the program).

Approach 0: Exhaustion

Exhaustion of all possible keytexts is practically infeasible when there is no a priori idea what the keytext could be. Exhaustion is feasible when the attacker knows the source of the keytext, say a certain book. If the source text has length q and the ciphertext has length r, then there are only $q - r$ choices for the start of the key text. This is troublesome for the pencil and paper analyst, but easy with machine support.

Approach 1: Probable Word and Zigzag Exhaustion

When in the example above the attacker guesses the probable word "arrive" in the plaintext and shifts it along the ciphertext, already for the second position she gets the keytext `FYOUCA`. With a little imagination she guesses the phrase `IFYOUCAN`, yielding the plaintext fragment `IARRIVET`, and expands this fragment to `IARRIVETOMORROW`. This in turn expands the keytext to `IFYOUCANKEEPYOU`. Proceeding in this way alternating between plaintext and keytext is called **zigzag exhaustion** (or cross-ruff method). For some time during this process it may be unclear whether a partial text belongs to plaintext or key.

A dictionary is a useful tool for this task. Or a pattern search in a collection of literary texts may lead to success.

Approach 2: Frequent Word Fragments

If the attacker cannot guess a probable word she might try common word fragments, bearing in mind that plaintext as well as keytext are meaningful texts. Shifting words or word fragments such as

```
THE AND FOR WAS HIS NOT BUT ARE ING ION ENT
THAT THIS FROM WITH HAVE TION
```

along the ciphertext will result in many meaningful trigrams or tetragrams that provide seed crystals for a zigzag exhaustion. Recognizing typical combinations such as

```
THE + THE = MOI
ING + ING = QAM
THAT + THAT = MOAM
```

may be useful.

Approach 3: Frequency Analysis

Let $p_0, \ldots, p_{n-1}$ be the letter frequencies of the (stochastic) language M over the alphabet $\Sigma = \{s_0, \ldots, s_{n-1}\}$. Then running-key ciphertexts will exhibit the typical letter frequencies

$$q_h = \sum_{i+j=h} p_i \cdot p_j \quad \text{for } 0 \le h \le n-1.$$

Even though the distribution is much more flat compared with plain language, it is not completely uniform, and therefore leaks some information on the plaintext. For example it gives a hint at the method of encryption.

Example: Letter frequencies of running-text cryptograms in **English** (values in percent). Coincidence index = 0.0400.

A	B	C	D	E	F	G	H	I	J	K	L	M
4.3	3.5	3.2	2.5	4.7	3.8	4.4	4.4	4.8	2.9	3.5	4.5	4.3
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
3.1	3.2	3.6	3.0	4.4	4.5	4.0	3.2	4.9	4.7	3.8	3.3	3.5

Example: Letter frequencies of running-text cryptograms in **German** (values in percent). Coincidence index = 0.0411.

A	B	C	D	E	F	G	H	I	J	K	L	M
4.2	2.6	2.3	2.4	5.0	3.7	3.7	4.3	5.8	2.9	3.7	4.4	4.9
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
3.2	3.0	3.1	3.3	5.7	3.4	3.2	3.4	5.9	4.5	3.9	3.9	3.6

Even more helpful is the distribution of bigrams and trigrams. Each bigram in the ciphertext has $26^2 = 676$ different possible sources whose probabilities however show large differences. For trigrams most sources even have probabilities 0.

A systematic description of this approach is in

> Craig Bauer and Christian N. S. Tate: A statistical attack on the running key cipher. Cryptologia XXVI (2002), 274–282.

Approach 4: Frequent Letter Combinations

Frequency analysis (approach 3) is cumbersome, at least for manual evaluation. FRIEDMAN refined this approach in a systematic way that doesn't need known plaintext. See the next section.