7 Statistical Analysis of Ciphertext

Character Frequencies

Natural languages such as German, English, Russian, ..., and also artificial languages such as MS-DOS-EXE, ..., Pascal, ..., MS-Word, ..., show typical character frequencies that are

- nonuniformly distributed,
- characteristic for the language.

Texts of about 500 or 1000 letters in a natural language rareley show a significant deviation from the typical frequencies.

This allows automating the cryptanalysis based on letter frequencies to a large extent. The web offers several such programs, for example see the ACA Crypto Dropbox `[http://www.und.nodak.edu/org/crypto/crypto/]`.

Mathematical Model

The simplest mathematical model for statistical analysis of ciphertext is a probability distribution on the underlying (finite) alphabet Σ with atomic probabilities $p(s)$ for all letters $s \in \Sigma$. Thus we assume that plaintexts are streams of independent (but not uniformly distributed) random letters.

A closer approximation to the truth would account for dependencies of letters from their predecessors according to the typical bigram distribution.

There are further possible refinements, for example the most frequent initial letter of a word in English is T, in German, D.

Example: Byte Frequencies in MS-Word Files

Byte	**Frequency**
`00`	ca 7-70%
`01`	ca 0.8-17%
`20` = space	ca 0.8-12%
`65` = e	ca 1-10%
`FF`	ca 1-10%

Observations

- The variability is rather large, unexpected peaks occur frequently.
- The distribution depends on the software version.
- All bytes `00-FF` occur.
- We see long sequences of zero bytes. If the file is encrypted by XOR, large parts of the key shine through.

The last remark yields an efficient method for analysis of the XOR encryption of a WORD file with periodically repeated key. This not exactly a statistical cryptanalysis, it only uses the frequency of a single byte. To start with, pairwise add the blocks. If one of the plaintext blocks essentially consists of zeroes, then the sum is readable plaintext:

Plaintext	...	a_1	...	a_s	...	0	...	0	...
Key (repeated)	...	k_1	...	k_s	...	k_1	...	k_s	...
Ciphertext	...	c_1	...	c_s	...	c'_1	...	c'_s	...

where $c_i = a_i + k_i$ in the first block, and $c'_i = 0 + k_i$ in the second block for $i = 1, ..., s$ (s the blocksize).

Therefore $c_i + c'_i = a_i + k_i + k_i = a_i$,—one block of plaintext revealed and identified—; and $k_i = c'_i$—the key revealed.

If the addition of two cipher text blocks yields a zero block, then with high probability both plaintext blocks are zero blocks (or with small probability are identical nonzero blocks). Also in this case the key is revealed.