Cybersecurity MSc: Applied Cryptography 2024-25

Practical Exercises: Classical cryptography and cryptanalysis

September 19, 2024 Due date: no date

Changelog

- v1.0 - Initial version.

1 Introduction

In classical cryptography, the information subject to cryptographic transformations was just the sequence of letters extracted from a text, excluding all non-alphabetical symbols (spaces, digits, punctuation, etc.). Also, characters with diacritics were used without them (thus, á was transformed to a, ç was transformed to c, etc.). Thus, the plaintext presented to the encryption algorithm was a sequence of letters, disregarding case, and the resulting ciphertext was of the same nature.

Those plaintext contents maintain the relative frequency of letters, as well as digrams, trigrams, etc., but can create some less evident n-grams, caused by space suppressions (e.g. "six xylophones" becomes "sixxylophones", which contains an "xx" digram that does not exist in English.

2 Preparation

Select a set of texts from a given language (Portuguese, English) and create a tool to prepare them for encryption with classical cryptographic tools. Them, create tools to evaluate their letter frequency, as well as to evaluate the more frequent n-grams (digrams, trigrams, etc.). You will get different results for different languages. Choose large texts for getting more accurate statistics.

3 Classical encryption with monoalphabetic substitution

In the classical monoalphabetic encryption using the substitution of symbols, the first step to take is to create a single substitution alphabet, or table, for a given key.

Use some strategy to create it, and use it to perform a successful encryption and decryption of the prepared texts.

Use the tools previously developed to analyse the statistics of the cryptograms, and verify that the frequency of letters and n-grams is kept, though for different letters and n-grans. This fact is the starting point for a successful cryptanalysis of a classical cryptogram performed with a monoalphabetic cipher.

4 Classical encryption with polialphabetic substitution

Polialphabetic substitution encryption attempts to mitigate the weaknesses of monoalphabetic techniques, namely to hide the frequencies of letters and n-grams of the original plaintext that are observable in the ciphertext.

Use a Vigenère square to implement a polialphabetic cipher and decipher. Note: you do not have to create the complete square, you can compute its transformations.

Encrypt texts with the Vigenère cipher, with a given key, and use the analysis tools previously developed to observe the statistics of the cryptograms produced for different keys with the same length and different key lengths. Compare these statistics with the ones observed with cryptograms created with monoalphabetic ciphers.

4.1 Determination of the key length

The cryptanalysis of polialphabetic ciphers that use a short-term, cyclic behaviour can benefit from knowing the number of alphabets used to implement it (usually, in a round-robin fashion). This information allows a cryptanalyst to slice the cryptogram in N parts, where N is the number of alphabets used consecutively, and to process each part as the result of a monoalphabetic cipher. The cryptanalysis of a slice can also facilitate the cryptanalysis of other adjacent slices, because tentative substitutions may be used to find suitable n-grams on the recovered text.

Create a tool to find possible sizes of an encryption key for a Vigenère cipher. For that purpose, use the Kasiski and the index of coincidence (autocorrelation) methods. Verify the tool efficacy with the cryptograms previously generated.

4.2 Determination of the original language of a cryptogram created with a polialphabetic cipher

Once you know the size of a key (say, N), you can perform a statistical analysis of each of the N slices of the cryptogram. By displaying the frequency of the most frequent symbols in each slice, you should be able to guess the original language (if you know the symbol frequency of that language in advance). Note that the frequencies should be similar in all slices (for large texts). Then, you can guess the encryption operation enforced on the original symbols of each slice, which yields the part of the key that was used to process that slice.

Note, however, that this strategy works for Vigenère ciphers, and not for all polialphabetic ciphers.

5 Non-cyclic polialphabetic ciphers

The analysis tools previously referred fail once the number of alphabets used has no fixed cycle. This was the strategy used by machines such as Enigma or Hagelin. However, we can use the Vigenère square to implement it as well.

Use the Vigènere square in the following way. Select an initial N-letter key. Encrypt the first N plaintext characters and produce the first N cryptogram characters. Then, rotate the key, such that the first character becomes the last, the second the first, and so on. Finally, change the last character (the one that was before the first) by incrementing it by one (modulo 26 plus `'a'`). For each N-letter block, modify the key as previously done.

Study the statistics of a cryptogram created this way with a large text and show that the Kasiski test may still succeed with short keys (e.g. with 5 letters). Explain why.

To finalize this task, you should submit for evaluation (as a bonus) a plaintext with more than $10,000$ characters (in any language), and the corresponding ciphertext encrypted with the concatenation of your first and last name. In the submission include as well the statistics of the cryptogram (letters' frequency) and the code you have produced for encryption, decryption and statistical analysis. Finally, explain the relevance of the key length for the robustness of this encryption algorithm against the Kasiski test.

References

- Substitution cipher, https://en.wikipedia.org/wiki/Substitution_cipher
- Vigenère cipher, https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher
- Kasiski examination, https://en.wikipedia.org/wiki/Kasiski_examination
- Index of coincidence, https://en.wikipedia.org/wiki/Index_of_coincidence