Lab Assignment:
Cryptanalysis of the (modified) Vigenère cipher

Prepared by Johannes Borgström

Cryptology (1DT075) – Spring 2018

Introduction

This lab assignment consists of four parts, (A), (B), (C) and (D) below. For part (A) you will be required to encrypt a Swedish text using the Vigenère cipher adapted to the Swedish alphabet. For (B), you will get to break texts encrypted using this cipher. For (C), you will have to reflect on modifications to the Vigenère cipher. Part (D) concerns the oral examination. You must succeed in all parts to pass the lab assignment.

General Instructions

This lab assignment is solved in groups of (usually) three students. You have been assigned to a group on the Student Portal. If your group partners are unresponsive, or if you are not planning to do this assignment, please notify a teaching assistant **as soon as possible**.

All group members should be involved in the work and discussions leading to the solution and should understand all parts of the solution.

It is acceptable to discuss solution methods with other groups, but you must construct your own solution and submit your own code. See our Ethics Rules for further guidance.

Parts of this assignment—in particular part (B)—may require a substantial amount of computing time in addition to the work that needs to be done. Please start working on **all parts** of the assignment early, i.e., now!

While this lab assignment asks you to work with Swedish texts, we understand that many students do not speak Swedish. If you have difficulties because of the language—or any other questions about the assignment—please contact one of our teaching assistants.

Submission Instructions

Hand in your solution for each part before the respective deadline in the designated file area on the Student Portal. (In case of technical problems with the Student Portal, send your solution to a teaching assistant by email before the respective deadline.)

(A) Encryption

Encrypt a Swedish text of your choice using the Vigenère cipher modified as follows. In addition to the 26 letters of the English alphabet (`a`, `b`, ..., `z`), the Swedish alphabet contains three additional letters, namely `å`, `ä` and `ö`. Moreover, we allow the space character (␣), comma (`,`) and period (`.`).

We thus use an alphabet of 32 characters instead of 26, and all arithmetic operations will be performed modulo 32. We assign values 0 to 31 to the characters: `a` is 0, `b` is 1, ..., `z` is 25, `å` is 26, `ä` is 27, `ö` is 28, space (␣) is 29, comma (`,`) is 30, period (`.`) is 31.

As an example, `varsågod` is encrypted with the key `hej` into `öeåz,pvh`. (We do not require that the message length is a multiple of the key length.)

You can encrypt the text by hand or write a program that does this for you (which may be a better idea). In either case, you must follow these specifications:

- Your plaintext must be in Swedish and between 200 to 600 characters in length (including spaces, commas and periods). If you do not speak Swedish, you can find suitable texts on the Internet, for instance at Project Runeberg.
- If your original plaintext contains uppercase characters, these must be replaced by the corresponding lowercase characters before encryption.
- If, after conversion to lowercase, the plaintext contains any characters that are not included in the 32 characters listed above, those unsupported characters must be removed before encryption. (Choose your original plaintext so that the plaintext still consists of least 200 characters after unsupported characters have been removed.)
- Your key must be no more than 16 characters in length, including spaces. It may only contain the 32 characters listed above.

Your plaintext and key will (eventually) become known to other students. Please refrain from using offensive language.

Handing in Part (A)

You must hand in a single ZIP archive named `vig_groupX.zip`, with `X` replaced by your group number. The archive must contain the following three text files (again, with `X` replaced by your group number):

1. Your (original *or* sanitized) plaintext as `vig_groupX.plain`.
2. Your key as `vig_groupX.key`.
3. Your ciphertext as `vig_groupX.crypto`.

Please follow these naming conventions precisely!

All text files must use the UTF-8 character encoding, and must not contain any additional characters beyond those detailed above. In particular, make sure there is no byte order mark at the start of your files, and no newline character at the end.

The submission deadline for part (A) is **Friday, January 27 at 18:00**. Your encrypted text will then be distributed to the other groups, who will try to break it in part (B) of the lab assignment.

(B) Breaking the Cipher

Write a program for breaking the Vigenère cipher. You may use any programming language of your choice. (We recommend Python or Java; we may not be able to provide technical assistance with other languages.) Your program must compile and run on the Ubuntu x86_64 lab servers, e.g., `siegbahn.it.uu.se`.

The lecture slides suggest approaches for breaking the cipher. It may also help to read the corresponding Wikipedia entry, especially the section on Cryptanalysis. Notations may differ between these sources and the course book, but the principles for breaking the cipher are the same.

You will be given ciphertexts and your task is to break them (i.e., to find the corresponding plaintexts) using your program. Some encrypted messages are the ones generated by other students and have a key no longer than 16 characters. The other messages have been encrypted by our teaching assistants and use a longer key. However, this key is *the same* for all such messages. You might need a slightly different approach to deal with those.

The table of letter frequencies for the Swedish alphabet is not provided. You will need to find a way to get it, either by generating it yourself or by finding a reliable reference.

The instructions for handing in part (B) are given below, together with those for part (C).

(C) Reasoning about Ciphers

Answer the following questions. Use the knowledge you have gained while solving parts (A) and (B) of the lab assignment, during the lectures, and/or elsewhere (e.g., on the Internet—but remember to give credit when you use someone else's ideas). You are required to motivate your answers, but we do not ask for formal proofs.

1. You have been asked to break a modified version of the Vigenère cipher, with 32 characters instead of 26. Do you think this version is easier to break, more difficult, or of the same level of difficulty?

2. How does the length of the key affect the security of the Vigenère cipher? Are there other characteristics that can impact the security of a particular key?

3. How could you break the Vigenère cipher if the language of the plaintext is (one of the world's major languages, but precisely which one is) not known?

4. Below are some suggestions for "ciphers" based on the Vigenère cipher. For each of them, discuss if they indeed define a cryptosystem; and if so, whether it is more secure than Vigenère or less, what the computational cost of a brute force attack would be, and what cryptanalysis approaches you would try (do not actually implement them, just briefly describe the ideas).

   Note: you have to consider each proposed cipher independently of the other ones.

   For each, we give an example, based on the text `Tänka␣fritt␣är␣stort␣men␣tänka␣rätt␣är␣större` and the key `Uppsala`, to help you understand the different modifications (highlighted in red below).

(a) Starting from the observation that the one-time pad is unconditionally secure, the idea is to use the reverse of the plaintext as the key. Thus, the key is as long as the message and is only used for this particular message.

Plaintext	t	ä	n	k	a	␣	f	r	i	t	t	...
Plaintext values	19	27	13	10	0	29	5	17	8	19	19	...
Key	e	r	r	ö	t	s	␣	r	ä	␣	t	...
Key values	4	17	17	28	19	18	29	17	27	29	19	...
Ciphertext values	23	12	30	6	19	15	2	2	3	16	6	...
Ciphertext	x	m	,	g	t	p	c	c	d	q	g	...

(b) Instead of adding plaintext characters to corresponding key characters, we subtract plaintext characters from key characters. In this fashion, encryption and decryption are the very same operation.

Plaintext	t	ä	n	k	a	␣	f	r	i	t	t	...
Plaintext values	19	27	13	10	0	29	5	17	8	19	19	...
Key	u	p	p	s	a	l	a	u	p	p	s	...
Key values	20	15	15	18	0	11	0	20	15	15	18	...
Ciphertext values	1	20	2	8	0	14	27	3	7	28	31	...
Ciphertext	b	u	c	i	a	o	ä	d	h	ö	.	...

(c) Before encrypting with the Vigenère cipher, we first transform the plaintext. Each character is replaced by the difference (modulo 32) between this character and the previous character. The first character is left unmodified.

Plaintext	t	ä	n	k	a	␣	f	r	i	t	t	...
Plaintext values	19	27	13	10	0	29	5	17	8	19	19	...
Differences	19	8	18	29	22	29	8	12	23	11	0	...
Key	u	p	p	s	a	l	a	u	p	p	s	...
Key values	20	15	15	18	0	11	0	20	15	15	18	...
Ciphertext values	7	23	1	15	22	8	8	0	6	26	18	...
Ciphertext	h	x	b	p	w	i	i	a	g	å	s	...

(d) Instead of repeating the initial key, we append the plaintext after it to continue the key stream.

Plaintext	t	ä	n	k	a	␣	f	r	i	t	t	...
Plaintext values	19	27	13	10	0	29	5	17	8	19	19	...
Key	u	p	p	s	a	l	a	t	ä	n	k	...
Key values	20	15	15	18	0	11	0	19	27	13	10	...
Ciphertext values	7	10	28	28	0	8	5	4	5	0	29	...
Ciphertext	h	k	ö	ö	a	i	f	e	d	a	␣	...

(e) Instead of repeating the initial key, we append the ciphertext after it to continue the key stream.

Plaintext	t	ä	n	k	a	␣	f	r	i	t	t	...
Plaintext values	19	27	13	10	0	29	5	17	8	19	19	...
Key	u	p	p	s	a	l	a	h	k	ö	ö	...
Key values	20	15	15	18	0	11	0	7	10	28	28	...
Ciphertext values	7	10	28	28	0	8	5	24	18	15	15	...
Ciphertext	h	k	ö	ö	a	i	f	y	s	p	p	...

Handing in Parts (B) and (C)

You must hand in two files:

1. your program (source code) for part (B), and
2. a written lab report. The report must be submitted as a PDF or text file. (Other formats, e.g., MS Word or ODF, will *not* be accepted.)

Your program must (as a minimum) be

- working. Please test your program on one of the Ubuntu x86_64 lab servers, e.g., `siegbahn.it.uu.se`, before handing it in!
- well commented. This means that one should be able to figure out what the program does even without being familiar with the programming language that you used.

In your lab report, you must include the following information:

- Your group number and the names of all group members.
- Instructions on how to compile and run your program on the Ubuntu x86_64 lab servers, assuming the reader has access to your program sources.
- A general explanation of the techniques you used to break the ciphertexts.
- The frequency table of Swedish letters that you used and an explanation of how you obtained it.
- Approximate running times for breaking each ciphertext. If you were unable to break one or more of the ciphertexts, please write this in your report, along with some reasoning on why you think you were unsuccessful.
- For part (C): motivated answers to the questions.
- A short evaluation of the lab: which parts were easy or hard to solve? How was the collaboration in your group? What did you like most and least in this lab?

To reduce the size of your report, please do *not* include the plaintexts or keys that are computed by your program.

The submission deadline for parts (B) and (C) is **Friday, February 3 at 18:00**.

(D) Examination

The oral examination will take place on **Tuesday, February 7**. The exact schedule will be announced later. At the examination session, we will discuss your answers to parts (B) and (C). You are expected to demonstrate an understanding of *all parts* of your group's answers through active participation in the examination.

Lab Rooms

You can work on this assignment using your own computer, or you can use any of the lab rooms at the department when they are not booked for other courses. The Ubuntu x86_64 servers (see `http://www.it.uu.se/datordrift/maskinpark/linux` for a full list) can be accessed via SSH.

Grading and Completions

The assignment is graded on a pass/fail scale. Groups that submit (at least) a serious attempt for parts (B) and (C) will be given the opportunity to resubmit if their answers were not mostly correct. Completions may be handed in via the Student Portal any time before the final exam on **March 16**.