DAT 510: Assignment 1

Submission Deadline: 23:59, Friday, Sept. 16, 2022

Cryptanalysis of primitive ciphers

In this assignment, you will try your skills at cracking some encrypted messages.
Warning: Although the encryption techniques used in this assignment are extremely primitive compared to practical encryption schemes used in the real world, they are not necessarily easy to solve (even with computer assistance). Start early and deadline for submission is soon!

Part I. Poly-alphabetic Ciphers

For this part of the assignment, you are given enciphered English text and a hint about the encryption algorithm that was used. Your mission: Develop the necessary (software) tools and use them to help you produce plaintext.

Task 1 The text was enciphered using a polyalphabetic substitution cipher, where the key length was no larger than 10. Blank spaces were first deleted and then inserted at convenient locations.

```
FRRPU TIIYE AMIRN QLQVR BOKGK NSNQQ IUTTY
IIYEA WIJTG LVILA ZWZKT ZCJQH IFNYI WQZXH
RWZQW OHUTI KWNNQ YDLKA EOTUV XELMT SOSIX
JSKPR BUXTI TBUXV BLNSX FJKNC HBLUK PDGUI
IYEAM OJCXW FMJVM MAXYT XFLOL RRLAA JZAXT
YYWFY NBIVH VYQIO SLPXH ZGYLH WGFSX LPSND
UKVTR XPKSS VKOWM QKVCR TUUPR WQMWY XTYLQ
XYYTR TJJGO OLMXV CPPSL KBSEI PMEGC RWZRI
YDBGE BTMFP ZXVMF MGPVO OKZXX IGGFE SIBRX
SEWTY OOOKS PKYFC ZIEYF DAXKG ARBIW KFWUA
SLGLF NMIVH VVPTY IJNSX FJKNC HBLUK PDGUI
IYEAM HVFDY CULJS EHHMX LRXBN OLVMR
```

Hints In the ciphertext the letters were first converted to upper case—thus the alphabet substitutions consist of permutations of the 26 upper case letters A through Z. Spaces were removed before encryption and reinserted after encryption. Suppose that substitution 1 maps A to X, and substitution 2 maps B to Y. The plaintext message “AB ABAB ABA” (with spaces) might be converted to the ciphertext “XYXYX YXYX”. You might consider using the statistical analysis techniques discussed in class to crack these problems.

Task 2. Try shorter and Longer key lengths in your program and use the time package to find out program execution time. Does the program take longer time to decrypt the ciphertext? how does execution time arise by adding letters to the key?
Hints: Here are useful links about how to measure execution time in Python And Javascript, You can also find the function for your preferred programming language using search engines.

Task 3. This ciphertext has been encrypted with the same Key as previous ciphertext(Task1) with an addition in the encryption process. Is it possible to Produce the plaintext again using the same tool you created in Task1? Explain the differences you have encountered.

```
IRKPV YNZPT UFQZL ULCDI OEVWF ETBAW SHLGO
YQSXT UQRRK LRQUT FHUSE ZBFPR BEPHY DYEKF
ZSPPT VYQSY GKUHJ GNHXN UMWFF XIZFN NLWTJ
CKYHZ YDPDX KCOUO JEOMU AKVAU EGUEX RKHFC
SNHGG WRABW RASXJ IFJHO JRLLJ KOQLO UQRIT
YHVFV GZGRM TLRQJ ZGNNP NYJAE DFLQI SLYSV
RVKLE AJUNL MHDGE IFFQN FKEKT NJGQN OPOXM
VVRRC JGHEH FEVGB QDAEI FDHTA AWFYG ZLLVO
AUXFV JRPGV DYOYK BFMQA TWFMS WUQEB PQHXC
WWEUP LGSGL NYMTM RXOWK FZFOE FUBFG QFNVI
OVLHZ NETBS AIBBT PEIHQ DRTAU EGUEX RKHFC
SNHGG PDDHY OBGOV CJBXG DVEIZ LWMJS
```

Hints: Here same encryption mechanism has been used with same key. ”**with an addition in the encryption process**”.

Part II. KES

A spy uses a product cipher named KES, whose encryption consists of a column-transposition-cipher at first and a Caesar-cipher in second, and the keys are located in address of a house, where the house telephone number is the key to the row-transposition-cipher the house-number is the key of the Caesar-cipher. According to the schedule, today's keys are associated with John Smith in the phone book: John Smith, Lagardsveien 3 , 4010 Stavanger Tel: 51 63 47 82 Today's message to the spy is:

HSQQD XHDRP YFKWV HNHDL OULLQ DDWVW BDWWA RJULS

Task 1. Present your cryptanalysis on this KES.

Task 2. What is the clear-text of the message above?

Task 3. How would you improve this KES?

Task 4. Implement your improved KES and encrypt the same message.

Task 5. Webserver using improved KES

Create a simple webserver which stores Raw keys used in the previous tasks and can decrypt any ciphertext coming to it by that key using KES which you have already created and show the result in the browser. This is a simple end to end encryption. How strong is the security in this type of communication?

Example:

Browser input(bits are just for demo):

http://localhost:5000/index.js?cipher=1011011010111011111001011110111000010111

Output: PLAIN MESSAGE

Hints. here are useful links on how to make simple webserver using Python(Flask)and Javascript (Nodejs)

Assignment Approval (by TA and SA)

Assignment approval will have a weight on your grade for the assignment. If you are not going to get the approval before deadline, your assignment will not be evaluated and you will fail the assignment.

What need to be done to get the approval of the assignment:

1. Show all parts of assignment is working i.e, show the code with proper comments, results.
2. Code should have a proper README file that describes the contents of the directory and any special instructions needed to run your programs (i.e. if it requires and packages, commands to install the package. describe any command line arguments with the required parameters).
3. Source code submitted for the assignment should be your own code. If you have used sources from the internet everything should be added to the references. If you used someone's code without reference, that will also be treated as plagiarism.
4. Provide the references in Code and Report, show these parts for TA's and Student Assistants.
5. You should **NOT** use available libraries/packages/classes for implementing the core functionality of the assignment.

You may use any "reasonable" programming language for part one of the assignment. Reasonable languages include: Java, C, C++, Python, MatLab, R and others with permission of Jayachander Surbiryala (Email: `jayachander.surbiryala@uis.no`) or Chunming Rong (Email: `chunming.rong@uis.no`).

Assignment Submission

Deadline: 23:59, Friday, Sept. 16, 2022 (submit your assignment through canvas)
Final submission:

1. Source Code
   - Source code submitted for the assignment should be your own code. If you have used sources from the internet everything should be added to the references. If you used someone's code without reference, that will also be treated as plagiarism.
   - Source code should be single, compressed directory in .tar.gz or .zip format.
   - Directory should contain a file called README that describes the contents of the directory and any special instructions needed to run your programs (i.e. if it requires and packages, commands to install the package. describe any command line arguments with the required parameters).
   - You should **NOT** use available libraries/packages/classes for implementing the core functionality of the assignment.
2. A **separate** report with PDF format
   - Texts in the report should be readable by human, and recognizable by machine;
   - Other formats will **NOT** be opened, read, and will be considered missing;
   - Report should follow the formal report style guide in next page.
   - Each student should write an individual report. Each report will be checked for plagiarism. If it is copied from some where else, you will fail the assignment.

NOTE: Please upload the archive file in *.zip, *.tar only and report in *.pdf format only to the website https://stavanger.instructure.com/.

Note: The assignment is individual and can **NOT** be solved in groups.

Project Title

Abstract

A one-paragraph summary of the entire assignment - your procedure, results, and analysis.

1. Introduction

Describe the background for the project. If you are building on top of any existing resources highlight them in this section and cite them in your references.

2. Design and Implementation

Detail description of the design, procedure, and implementation of your project along with the following details from Part I and Part II.

2.1 Part I

- The plaintext message you managed to decipe r;
- Describe the strategy you employed, show the details for each of the steps of that strategy, describe any programs you wrote, show sample output of these programs, and show how you transformed that output into your solution.
- Describe the Execution time and impact of the key length on it.

2.2 Part II

- Present your crypt analysis on this KES **Tasks 1 and 2**;
- How would you improve this KES? **Tasks 3**
- Implement your improved KES and encrypt the same message **Task 4**.
- Describe **Task 5** implementation and how strong is the security in this type of communication?

3. Test Results

Results of testing the software, as you observed/recorded them. Note that this section is only for observations you make during testing. Your analysis belongs in the Discussion section.

4. Discussion

Your analysis of what your testing results mean, and your error analysis.

5. Conclusion

A short paragraph that restates the objective from your introduction and relates it to your results and discussion and describes any future improvements on your techniques that you would recommend.

References

A bibliography of all of the sources you got information from in your report.