University of Wollongong
Singapore Institute of Management

School of Computing and Information Technology (SCIT)

Lecturers: Distinguished Professor Willy Susilo and Mr. Japit Sionggo

CSCI361 - Session 3 2024

Assignment 1 (15 marks)

Due: **Saturday, 27 July 2024**, 09.00pm Singapore time. Submission via Moodle only.

Aim: To gain a basic familiarity with classical ciphers, including statistical methods for cryptanalysis of them, and to understand block cipher. This assignment includes seven tasks.

Standard Requirements for Assignments

- Submissions must be made via Moodle. No other submission method will attract any marks.
- Submission via email will result in getting ZERO.
- Identify your answer clearly by making a subdirectory for that solution. For example, makea subdirectory called "Task 1", "Task 2", etc. Follow the directions given by the tutor (Mr. Sionggo).
- At the top of your code, you will need to specify the version of the programming language that you use (you need to use C++/C/Java/Python - or else, please consult Mr. Sionggo).
- Students are to give batch / make files for compilation.
- Students are to place all compilation and instructions on how to run the program inside a README.TXT file. The markers will refer to this file when marking.
- Submission filenames are to be the same as the ones given in the assignment specifications, do not use your own filenames.

- **Plagiarized assignments will receive 0 marks immediately.**
- **DO NOT** post this assignment to any forum, or else you will receive 0 marks immediately.
- The penalty for the late assignment is **25% per day**, including weekend.

1 Task One: Cryptanalysis (2 marks)

Obtain two ciphertext files from Mr. Japit Sionggo **(don't miss it!)**. These files are `Ctext-1` and `Ctext-2`. They have been generated using a monoalphabetic cipher and a Vigènere cipher, respectively.

You need to:

- Apply cryptanalysis to each of `Ctext-1` and `Ctext-2`. You can use the `krypto` programprovided.
- Present a report describing what steps you took to break each cipher, and why. Justify the choices you have made.
- Include graphs (frequency distributions) as appropriate (in your Report1.pdf).
- Provide the plaintext and key for each cipher. You should include them as files `Ptext-1.txt`, `Key-1.txt`, `Ptext-2.txt` and `Key-2.txt`, and give them in your report (Report1.pdf).
- If you use any other tools or software, you must cite them in your report. If you only provide the final answer but without any proper analysis, then you will obtain0.

2 Task Two: The Flipped Kamasutra cipher (2 marks)

Implement the Flipped Kama Sutra cipher, in C, C++, Python or Java. The name of your programshould be `FKamasutra.c` or `FKamasutra.cpp` or `FKamasutra.java` or `FKamasutra.py`. Include inyour report instructions on how to compile your code. You must include a Makefile that can be used to build your program. To build your program, one can just type:

```
make all
```

The command syntax of your program should be as follows:

```
fkamasutra -k <keyfile.txt>
fkamasutra -e <keyfile.txt> <plaintext.txt> <ciphertext.txt>
fkamasutra -d <keyfile.txt> <ciphertext.txt> <plaintext.txt>
```

where the `-k` option is to generate the keypair in a file called keyfile.txt, `-e` option is associated with encryption and the `-d` option is associated with decryption. The encryption and decryption processes take the keyfile.txt and a plaintext (or a ciphertext, resp.) to produce a ciphertext (or a plaintext, resp.). You may assume all input is lower case without punctuation. The description of the Flipped Kama Sutra cipher is as follows. In the 4th century BC, the Indian text "Kama Sutra"proposed a method of encrypting text. Each letter of the alphabet was paired with one other letter. A ciphertext was formed by replacing each letter in the plaintext with its paired letter. When thisscheme is used in the English language, the number of possible keys is surprisingly high: around 7.9×10^{12}. An exhaustive attack on such a scheme would be unwieldly using a modern computer, and it was certainly infeasible at the time this scheme was suggested. For example, suppose the keyfile is just a regular alphabet as follows.

```
abcdefghijklmnopqrstuvwxyz
```

Then, suppose the plaintext contains of the following

```
abab bcbc cdcd
```

The resulting ciphertext would be

```
dcdc cbcb baba
```

Note that the order of the ciphertext is "flipped" from the very end. You also need to:

- First, generate a random keyfile `keyfile.txt`.
- Generate the ciphertext file Ctext-3.txt obtained by encrypting Ptext-1.txt under the key inkeyfile.txt.
- In your report, describe the statistical properties of Ctext-3 and discuss how they compare them with those of Ctext-1 and Ctext-2, remembering that Ctext-1 and Ctext-3 are associated with the same plaintext. Include a comparative graph of the letter frequency distributions. Write your report in a file called `Report2.pdf`.
- Discuss a way to decrypt the Kama Sutra cipher without using the key. Show your argument by decrypting the ciphertext Ctext-3 assuming that you don't know the key. The discussionwill need to be written in the same file `Report2.pdf`.
- To test the correctness of your program, one can just simply test with

  ```
  fkamasutra -d keyfile.txt Ctext-3.txt Output.txt
  ```

  and verify whether the file `Output.txt` is identical with `Ptext-1.txt` by executing

  ```
  diff Output.txt Ptext-1.txt
  ```

  in the Linux environment.

3 Task Three: Analysis of Substitution Cipher (2 marks)

Consider the following mapping

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Can the function $f(x) = x^k \ (mod\ 26) - 1$ be used as a cipher where $k > 1$ is the key and x is a letter to be encrypted? Justify your answer.

4 Task Four: 4-bit OFB (2 marks)

This task comprises three parts. Part one. This task is to implement a 4-bit OFB TEA algorithm. See the code for TEA in the lecture notes. Part two. Encrypt your student number using the 4-bit OFBTEA algorithm as devised above. Part three. Add all the digits of your student number mod 7. Let the result be *c*. Then implement c-bit OFB TEA algorithm and encrypt your student numberusing this algorithm. Compare the time needed to encrypt using 4-bit OFB TEA algorithm and *c*-bit OFB TEA algorithm and write your result in Task4Report.PDF. Explain what has happened. **Remark:** If the result of the modulo operation is 0 or 4, that is c = 0 or c = 4, then you need to implement 3-bit OFB TEA instead.

5 Task Five: Symmetric Cryptosystem Analysis (3 marks)

Alice and Bob use a cryptosystem in which their private key is a (large) prime k and their plain-texts and ciphertexts are integers. Bob encrypts a message m by computing the product

$$c = km$$

The enemy, Eve, intercepts two ciphertexts c_1 and c_2.

- Can you help Eve to find Bob and Alice's private key? Show the mathematics formula todemonstrate this. (1 mark)
- Write a program to implement your method in the earlier part of this question. Then, enterthe following ciphertexts:

  $$c_1 = 12849217045006222$$

  and

  $$c_2 = 6485880443666222$$

  Can you find the private key? (2 marks)

  Write your report for this section in Task5.pdf.

6 Task Six: Synchronous Cipher (2 marks)

The following stream cipher operates on character A-Z one at a time. A character A-Z is mapped to $Z_{26} = \{0,1,2, \ldots,25\}$ in the usual way. For a key $k \in Z_{26}$, a key stream $k_1 k_2 k_3 \ldots$ is generated by the following rule: $k_1 = (k^3 + 2)\ mod\ 26, and\ k_n = (k_{n-1}^5 + n)\ mod\ 26$.

For a message $m = m_1 m_2 \ldots m_t$, where $m_i \in Z_{26}$, the ciphertext $c = c_1 c_2 \ldots c_t$ is calculated as follows:

$$c_i = m_i + k_i\ (mod\ 26)$$

Your tasks are:

- Describe the decryption algorithm.
- Implement the above algorithm, for both encryption and decryption.
- Encrypt the message "I LOVE WOLLONGONG" with key = 3.
- Decrypt the ciphertext $MQJJ$ with key = 3.

7 Task Seven: Comparing block cipher modes (2 marks)

In this task, we are to explore a simple image encryption. An example of an image encryption iswritten in Python as follows:

```
# Simple Python Code

ifile, ofile = sys.arg[1:3]
with open(ifile, "rb") as
   reader: with open(ofile,
   "wb+") as writer:
      image_data = reader.read()
      header, body = image_data[:54],
      image_data[54:] body += b"\x00"*(16-
      (len(body)%16)) writer.write(header +
      aesEncryptor.update(body))
```

Your first task is to understand the idea of the above code. This is a very basic AES encrypting program, and the key that is used for encrypting it does not really matter much. It means, you can just choose any random key for this, or just use a fixed test key. We will need to first read a binary file, encrypt everything *except the first 54 bytes*, and then write it out to a new file. The reason we are not encrypting the first 54 bytes is because this program is going to encrypt the contents of a bitmap file (BMP) and the header is 54 bytes in length. You should try to use an image editor of your choice to create a large image with text that

takes up most of the space. Then save the encrypted file to something encrypted_image.bmp, then open the file with an image viewer. Finally, write a decryption program to decrypt the encrypted image. If you can see the original image, then it means you have done this first step correctly.

The main idea of this task is to understand the difference between AES-CBC and AES-CTR mode. Do the following steps.

First introduce an error into the ciphertext and decrypt the modified bytes. Try for example picking the byte right in the middle of the encrypted image data and setting it to 0. After corrupting the data, call the decryption program and view the restored image. Compare the impact between CBC and CTR mode. Based on this experiment, write a report to demonstrate the difference and impact between CBC and CTR mode.

Hints: You can try with an all-white image. If you still cannot figure it out, change 50 bytes or so to figure out where the changes are happening. Once you find where the changes are happening, go back to changing a single byte to view the differences between CTR and CBC. Can you explain what is happening there clearly in the report?

You can choose to use C++ or Java to implement this part too. It is not necessary to use Python.

Submission

You need to submit one ZIP file and upload it to Moodle. In this ZIP file, you need to createseven subdirectories, in which each subdirectory will have the answer to each task. For each programming task, write a README file that explains the compiler setting. Ideally, you should make a Makefile for each of the tasks.

DISCLAIMER: This assignment contains an intellectual property that is owned by the University of Wollongong. Please do not use it without the permission from the University of Wollongong. Ifyou have any questions, please contact the author on wsusilo@uow.edu.au or sjapit@uow.edu.au.

DISCLAIMER: By submitting your assignment, you **DECLARE** that the assignment is your own work and you did not obtain it from any third party or even purchase it from someone else or ask someone else to do it for you. If you violate this rule, then you may end up failing the subjectentirely.