University of Wollongong
Singapore Institute of Management

School of Computing and Information Technology (SCIT)

Lecturers: Distinguished Professor Willy Susilo and Mr. Sionggo Japit

CSCI361 – Session 3 2025

Assignment 1 (15 marks)

Due: Sunday, 27 July 2025 by 9:00 pm 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

- Submission 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, make a 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.
- Penalty for the late assignment is 25% per day.

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 program provided.
- Present a report describing what steps you took to break each cipher, and why. Justify 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 also 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 obtain 0.

2. Task Two: The Substitution Cipher (2marks)

A substitution cipher by a keyword works as follows. If a keyword is "STRAWBERRY", we remove the repeating characters in it to get "STAWBEY". Then, we append the rest of the alphabet characters in reverse order (from 'Z' to 'A') to the keyword to construct a key i.e., "STAWBEYZXVURQPONMLKJIHGFDC" to form a complete substitution key. Finally, we encrypt a message by substituting its characters with the characters in the key as follows:

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
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
S	T	A	W	B	E	Y	Z	X	V	U	R	Q	P	O	N	M	L	K	J	I	H	G	F	D	C

Your task is to implement the substitution cipher. In particular, your program

- should be a command-line program;
- take a keyword, which is typed through *command-line interface*, as input;
- take a plaintext (message) *file* as input;
- output a ciphertext *as a file*;
- take a ciphertext *as a file;*
- output a plaintext *as a file;*
- handle any possible errors;
- write in *C/C++, Java, or Python*. (You need to consult Mr. Sionggo otherwise);
- be submitted with a clear *readme.txt file*.

NB: Ignore case. That is, you can choose whether you will input and output uppercase or lowercase characters. Also, keep (do not encrypt) special character such as full stops and commas.

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 $f(x) = x^k\ (mod\ 26)$ 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: The Flipped Kamasutra cipher (2 marks)

Implement the Kama Sutra cipher, in C, C++, Java or Python. The name of your program should be Kamasutra.c or Kamasutra.cpp or Kamasutra.java or Kamasutra.py. Include in your 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:

```
kamasutra -k <keyfile.txt>
kamasutra -e <keyfile.txt> <plaintext.txt> <ciphertext.txt>
kamasutra -d <keyfile.txt> <ciphertext> <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 ciphtertext, resp.) to produce a ciphertext (or a plaintext, resp.). You may assume all input is lower case without punctuation. The description of the 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. However, when a letter 'f' is found, then the paired letter will not be replaced. This is representing the 'flipped' version of Kamasutra. When this schema is used in the English language, the number of possible keys is surprisingly high: around 7.9×10^{12}. An exhaustive attack kon such a scheme would be unwielly 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 effe

The resulting ciphertext would be

baba adad dcdc effe

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 in keyfile.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 if you don't know the key. The discussion will need to be written in the same file Report2.pdf.
- To test the correctness of your program, one can just simply test with

  ```
  kamasutra -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.)

  fc Output.txt Ptext-1.txt (in the Windows environment using Windows Command Prompt or PowerShell.)

5.Task Five: 3-bit CFB (3 marks)

This task comprises three parts. Part one. This task is to implement a 5-bit CFB TEA algorithm. See the code for TEA in the lecture notes. Part two. Encrypt your student number using 5-bit CFB TEA 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 CFB TEA algorithm and encrypt your student number using this algorithm. Compare the time needed to encrypt using 5-bit CFB TEA algorithm and c-bit CFB TEA algorithm and write your result in Task5Report.pdf. Explain what has happened. **Remark:** If the result of the modulo operation is 0 or 5, that is c = 0 or c = 5, then you need to implement 4-bit CFB TEA instead.

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 $Z_{26} = \{0, 1, 2, ..., 25\}$ in the usual way. For a key $k \in Z_{26}$, a key stream $k_1k_2k_3 \cdots$ is generated by the following rule:

- $(k_0, k_1) \in Z_{(26)}$
- $k_i = k_{n-1} \times k_{n-2}\ (mod\ 26), where\ i = 2,3,4 \ldots n$

For a message $m = m_1m_2 \cdots m_t$, where $m_i \in Z_{26}$, the ciphertext $c = c_1c_2 \cdots c_t$ is calculated as follows

$$c_i = m_i + k_i \pmod{26}$$

Your tasks are:

- Describe the decryption algorithm.
- Implement the above algorithm, for both encryption and decryption.
- Encrypt the message “I LOVE WOLLONGONG" with key = $(k_0 = 7,\ k_1 = 11)$.
- Decrypt the ciphertext MQJJ with key = $(k_0 = 7,\ k_1 = 11)$.

7.Task Seven: Take the best of Two (2 marks)

This question has several steps:

1. Modify Task 5 to achieve 1-bit CFB.
2. Find a document which is about 200 MB and then try to encrypt with 1-bit CFB and then decrypt it. Also, encrypt the same document with the synchronous cipher from Task 6 (encrypt and decrypt). Measure the time and put it in Report7.pdf.
3. Now, combine the implementation from Task 5 and Task 6, so for all the odd characters, we will use 1-bit CFB and all the even characters will use the synchronous cipher from Task 6.
4. Encrypt the same document from part 2 and report the encryption and decryption time.
5. Explain the phenomena that happens and try to justify why this occurs.

Submission

You need to submit one ZIP file and upload it to Moodle. In this ZIP file, you need to create seven 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. If you 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 subject entirely.