1. Classical Cryptography

A. OVERVIEW

1. Introduction and learning objective

Cryptography plays a vital role in modern digital communication systems. The Oxford Dictionary[1] defines *cryptography* as *"the art of writing or solving codes"* with *"codes"* elsewhere defined as *"a system of prearranged signals, especially used to ensure secrecy in transmitting messages."* Historically, cryptography focused exclusively on ensuring private communication between two parties sharing secret information in advance using code. It was used primarily for military, government, and a few niche industry applications for centuries.

But cryptography nowadays is much more of a science. We would say that **modern cryptography** involves studying *mathematical techniques for securing information, systems, and distributed computations against adversarial attacks.* Cryptography has gone from *"an **art** form that dealt with secret communication for the military"* to *"a **science** to secure systems for ordinary people all across the globe"*. It deals with mechanisms for ensuring integrity, techniques for exchanging secret keys, protocols for authenticating users, electronic auctions and elections, digital currency, and more.

[1] https://www.oxfordlearnersdictionaries.com

The **learning objective** of this lab is for students to get familiar with the concepts in secret-key encryption, particularly in classical cryptography. After finishing the lab, students should gain first-hand experience with encryption algorithms. Moreover, students may use crypto tools and write simple programs to encrypt/decrypt messages. This lab will cover the following topics regarding classical ciphers:

1. Monoalphabetic substitution ciphers - Frequency analysis
2. Polyalphabetic ciphers
3. Permutation ciphers

2. Backgrounds and Prerequisites

To ensure everything goes smoothly and you achieve better results in this lab, you are expected to be familiar with cryptography concepts and gain enough background knowledge about symmetric cryptography, particularly in classical ciphers. Besides, programming skill (with your preferred language, e.g., Python, C/C++, Golang) is also necessary.

The following section summarizes some of the key aspects to help you review the knowledge that you may already have obtained before getting started. You can also read the textbooks and related references that I listed in section 1.4 for more details.

Concepts

Before beginning, we define some terms. When we're encrypting a message,

- **The plaintext (p)** refers to the original or unencrypted message
- **The ciphertext (C)** refers to the coded or encrypted message.

A cipher is therefore composed of two functions:

- **Encryption** or **Enciphering (E)** turns plaintext into ciphertext.
- **Decryption** or **Deciphering (D)** turns a ciphertext back into plaintext.

Written as functions:
$C=E(K_e,p)$
$p=D(K_d,C)$
where K_e and K_d are encryption and decryption keys, respectively.

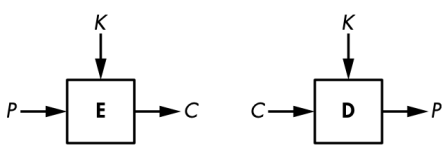

Figure 1: Basic encryption and decryption

Techniques used for deciphering a message without any knowledge of the enciphering details fall into the area of **cryptanalysis** (breaking the code). The areas of cryptography and cryptanalysis together are called **cryptology**.

Taxonomy

Concerning encryption methodologies, there are two types of ciphers:

1. **Symmetric ciphers** *(or Secret-key ciphers)*: Using a single key for encryption and decryption. It was the only type of encryption in use before the development of public-key encryption in the 1970s (Figure 2).

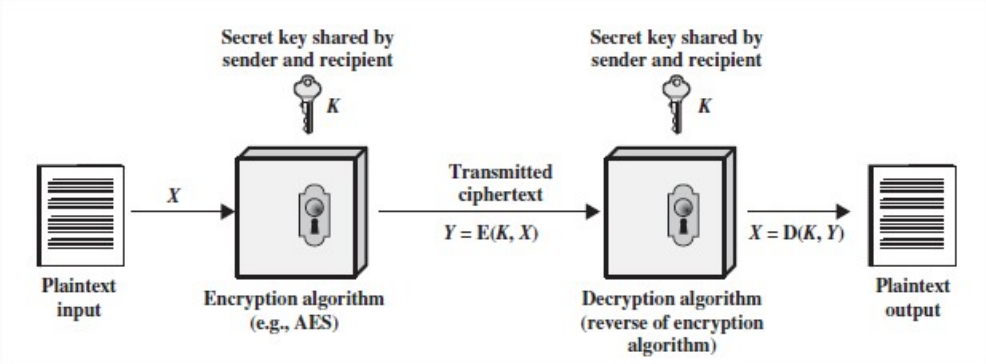

Figure 2: Symmetric Encryption model

2. **Asymmetric ciphers** *(or Public-key ciphers)*: Use two related keys, a public key and a private key, to perform complementary operations, such as encryption and decryption or signature generation and signature verification.

Classical ciphers predate computers; therefore, they work on letters rather than bits. Almost all classical ciphers are symmetric ciphers.

The two basic building blocks of all encryption techniques are *substitution* and *transposition* (Figure 3)

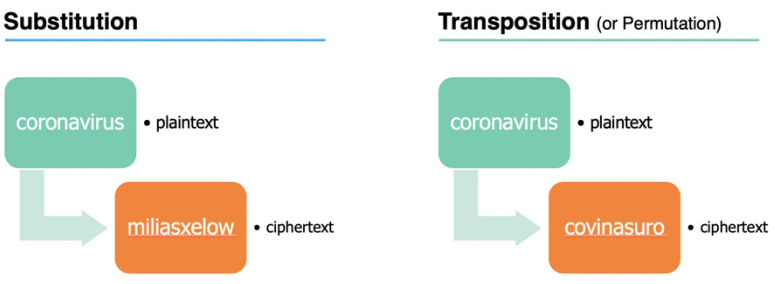

Figure 3: Substitution and Transposition example

B. LAB TASKS

1. Kickoff: Crack the code

Let's begin with a straightforward task that does not use any cipher algorithm. Try to solve the following codes:

a. We need to find the code to open the lock in Figure 4. The lock has a three-digit pin that satisfies five conditions (hints). Can you crack this code? If it's possible, explain how.

b. Find the corresponding encoding for the numbers 1 to 9 according to the clues provided in Table 1:
   - Each symbol in the set (✪✪✪✪✪✪✪✪✪) **unique** encoding for one of the numbers from 1 to 9.
   - The rightmost column is the sum of the numbers in each row.
   - The bottom row is the sum of the number in each column.
   - Each symbol "?" can represent any one-digit or a two-digit number and could be the same or different from each other.

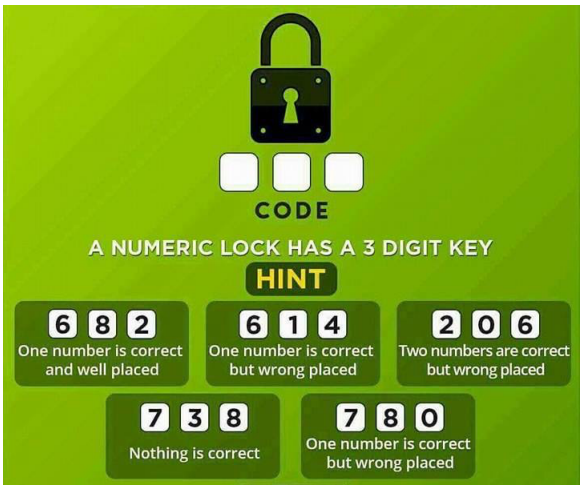

Figure 4: Crack the code to open the lock

Table 1: Find the corresponding encoding for each number

2. Caesar cipher

In this task, you must write an application using your chosen programming language to encrypt and decrypt a message using Caesar cipher. Your application should satisfy the following requirements:

- Allow to input a key and a plaintext to encrypt or ciphertext to decrypt using a given key.
- Allow brute-force all possible keys k to find the plaintext of given ciphertext without its key.

Test your program with a message of at least 100 words and compare the result with other cryptography tools (like Cryptool 2) to verify. Then use your program to crack the following ciphertext:

```
Gurer ner gjb xvaqf bs crbcyr va guvf jbeyq: gubfr jub ner ybbxvat
sbe n ernfba naq gubfr jub ner svaqvat fhpprff. Gubfr jub ner
ybbxvat sbe n ernfba nyjnlf frrxvat gur ernfbaf jul gur jbex vf
abg svavfurq. Naq crbcyr jub svaq fhpprff ner nyjnlf ybbxvat sbe
ernfbaf jul gur jbex pna or pbzcyrgrq.
```

Do you find any special concerning the key used to encrypt this ciphertext?

Tips: *Using the Caesar algorithm*

Encryption : $C = E(k, p) = (p + k) \ mod \ 26$

Decryption : $p = D(k,C) = (C - k) \ mod \ 26$

where C = Ciphertext, p = plaintext, k is key

3. Mono-alphabetic substitution cipher and frequency analysis

(This task is based on a lab in SEED Labs materials by Wenliang Du, Syracuse University.)

It is well-known that monoalphabetic substitution cipher (also known as the monoalphabetic cipher) is not secure because it can be subjected to frequency analysis. You are given a ciphertext encrypted using a monoalphabetic cipher in this lab. Each letter in the original text is replaced by another letter, where the replacement does not vary (i.e., a letter is always replaced by the same letter during the encryption.

Click here to download the ciphertext file.
Your job is to find out the original text using the frequency analysis technique. It is known that the original text is an English article. Describe how to find the plaintext in detail (step-by-step).

Please note that you are not allowed to use the automatic mode of any tools (like CrypTool, dCode, quipqiup,…) to decrypt.

Tips: *Using the frequency analysis, you can easily find the plaintext for some of the characters. For those characters, you may want to change them back to their plaintext, as you may be able to get more clues. It's better to use capital letters for plaintext, so for the same letter, we know which is plaintext and which is ciphertext.*

If you use Linux or macOS, you can use the ***tr*** *command. For example, in the following, we replace letters* ***a, e*** *and* ***t*** *in* ***in.txt*** *with letters* ***X, G, E****, respectively; the results are saved in out.txt.*

```
$ tr 'aet' 'XGE' < in.txt > out.txt
```

You can also use Cryptool or dCode to decrypt it manually by analyzing and replacing letters. There are many online resources that you can use. Two valuable links as the following:

- ***Cryptool Online N-gram analysis:*** *This website can produce the statistics from sequence), trigram frequencies (3-letter sequence), etc.*
- *http://norvig.com/mayzner.html: This web page provides frequencies for a typical English plaintext, including unigram, bigram, and trigram frequency.*

Advanced task 1:

Decrypt a ciphertext from Edgar Allan Poe's – **The Gold-Bug** and explain how to do:

```
53‡‡†305))6*;4826)4‡.)4‡);806*;48†8¶60))85;1‡(;:‡*8†83(88)
5*†;46(;88*96*?;8)*‡(;485);5*†2:*‡(;4956*2(5*-4)8¶8*;40692
85);)6†8)4‡‡;1(‡9;48081;8:8‡1;48†85;4)485†528806*81(‡9;48
;(88;4(‡?34;48)4‡;161;:188;‡?;
```

It's known that:

- The original text is an English article.
- The ciphertext does not include punctuation and spaces.
- Each symbol corresponds to a letter in the English alphabet.

4. Playfair cipher

Write an application with your own programming language to encrypt and decrypt a message using **Playfair cipher**. Your application should satisfy the following requirements:

- Allow you to input a key and a plaintext to encrypt or a ciphertext to decrypt using the given key.
- Display the Playfair matrix (5x5) corresponding with the given key.

a. Test your program with a message of at least 100 words and compare the result with other cryptography tools (like Cryptool 2) to verify (include the test case in your report)

b. Prepare a message with the full name of all members of your group and some words to get at least 50 characters. Use the Playfair matrix below (Figure 5) to encrypt your message:

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

Figure 5:Playfair matrix for task

5. Polyalphabetic cipher – Vigenère

In this task, write an application using your chosen programming language to encrypt and decrypt a message using Vigenère cipher.

Test your application by a message with at least 100 words and a key of about 10-20 letters. Then verify the result with other cryptography tools (e.g., Cryptool 2, dCode, etc.)

6. Other ciphers:

a. Find the flag of the following message (describe how to find the flag in detail – step-by-step):

```
TXpNek5ETXpNek16TXpNMU16TXpNak16TXpVek16TTVNek16TlRNek16QXp
Nek0xTXpNek5ETXpNelF6TkRNMk16TXpORE16TXpjek16TTFNek16TWpNek
16UXpNek14TXpNek5UTXpNekF6TXpNME16TXpPRE16TXpVek16TTU=
```

b. Describe another classical cipher that was not mentioned in this lab. Write an application and give a demonstration to illustrate how it works.

7. Prepare for the next lab

Programming cryptography application using Crypto++ (CryptoPP) library in C++:

- Introduction to Crypto++ library: https://www.cryptopp.com/
- Setup the environment and get ready to write C++ applications with Crypto++
- Write a simple C++ application with Crypto++ library to solve any problems relevant to cryptography.
- Input, output in Crypto++
  - in: *https://www.cryptopp.com/wiki/Source*
  - out: *https://www.cryptopp.com/wiki/Sink*

- String in Crypto++
  - in: StringSource (https://www.cryptopp.com/wiki/StringSource)
  - out: StringSink (https://cryptopp.com/docs/ref/class_string_sink_template.html)
  - Convert string:
  - wstring, Hex, Base 64,...
  - Vietnamese supports
- Array in Crypto++
  - in: ArraySource (byte arrays) (https://www.cryptopp.com/wiki/ArraySource)
  - out: ArraySink (byte arrays) (https://www.cryptopp.com/wiki/ArraySink)
- File in Crypto++
  - in: FileSource (https://www.cryptopp.com/wiki/FileSource)
  - out: FileSink (https://www.cryptopp.com/wiki/FileSink)
- Work with large numbers
  - Integer class (https://www.cryptopp.com/wiki/Integer)
  - ModularArithmetic (https://cryptopp.com/docs/ref/class_modular_arithmetic.html)
  - - nbtheory class
  - Convert string <-> integer

C. REQUIREMENTS

You are expected to complete all tasks in section B (Lab tasks). Advanced tasks are optional, and you could get bonus points for completing those tasks. We prefer you work in a team of 2 members to get the highest efficiency.

Your submission must meet the following requirements:

- You need to submit a **detailed lab report in .docx** *(Word Document)* format, **using the report template** provided on the UIT Courses website.

- Either Vietnamese or English report is accepted. That's up to you. Using more than one language in the report is not allowed (except for the untranslatable keywords).
- When it comes to **programming tasks** *(require you to write an application or script),* please attach all source-code and executable files (if any) in your submission. Please also list the important code snippets followed by explanations and screenshots when running your application. Simply attaching code without any explanation will not receive points.
- Submit work you are proud of – don't be sloppy and lazy!

  Your submissions must be your own. You are free to discuss with other classmates to find the solution. However, copying reports is prohibited, even if only a part of your report. Both reports of owner and copier will be rejected. Please remember to cite any source of the material (website, book,…) that influences your solution.

Notice: Combine your lab report and all related files into a single **ZIP file (.zip)**, name it as follow:

StudentID1_StudentID2_ReportLabX.zip

D. REFERENCES

[1] William Stallings, *Cryptography and network security: Principles and practice, 7th ed*, Pearson Education, 2017. *Chapter 3. Classical Encryption Techniques*

[2] Wenliang Du (Syracuse University), *SEED Cryptography Labs* https://seedsecuritylabs.org/Labs_16.04/Crypto/.

Training platforms and related materials

- ASecuritySite-https://asecuritysite.com
- Cryptopals-https://cryptopals.com

Attention: *Don't share any materials (slides, readings, assignments, labs, etc..) out of our class without my permission!*