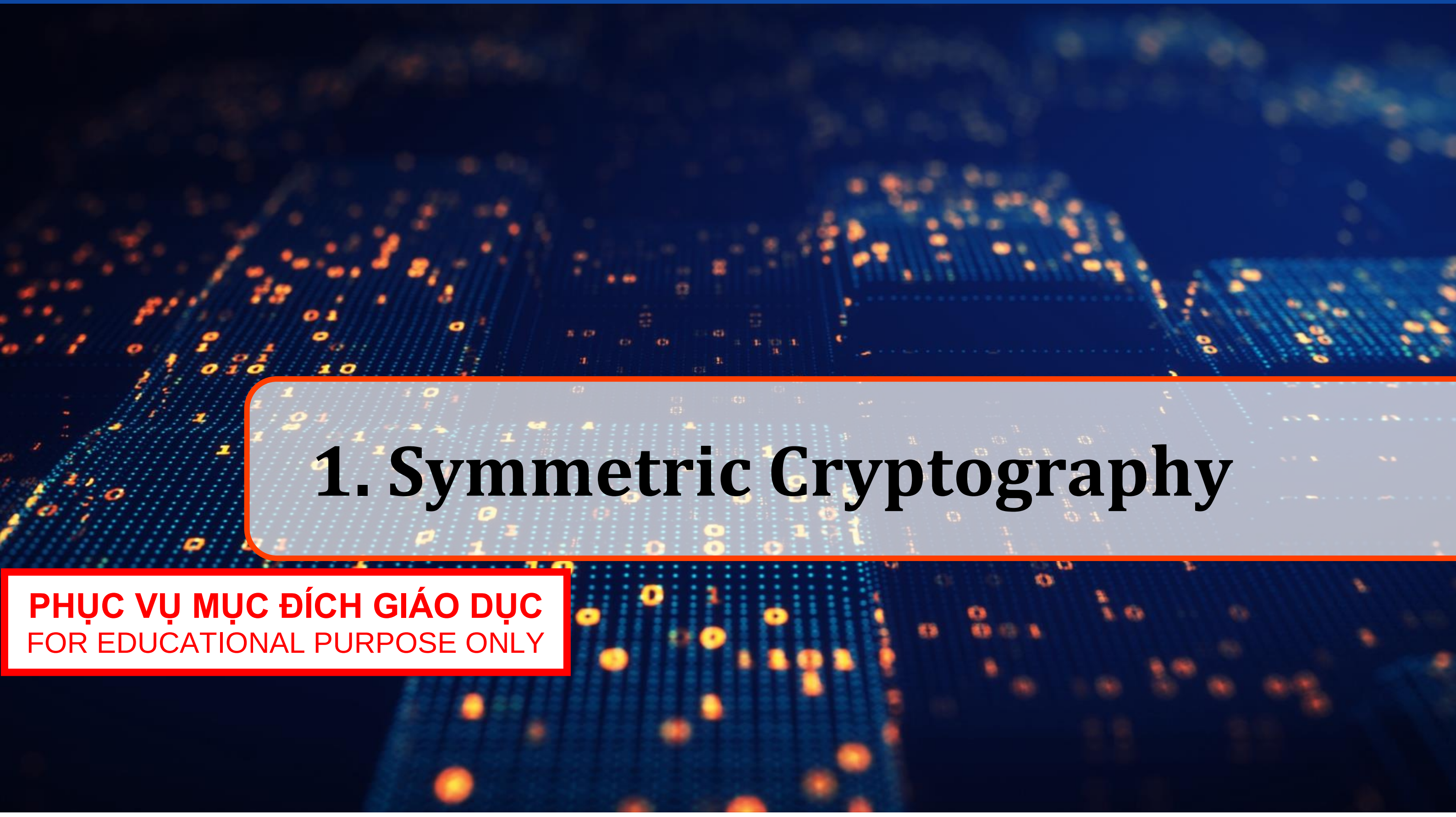


1. Symmetric Cryptography

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 and block ciphers. 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:

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

2. Practical environment 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. Besides, programming skill (with your preferred language, e.g., Python, C/C++, Golang) is also necessary.

This lab requires you to have the following preparations:

- A Linux-based computer (using Ubuntu or CentOS operating system) that preinstalled OpenSSL for encryption and decryption.
- An IDE of your preferred programming language.

3. Backgrounds and Prerequisites

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 reference 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.

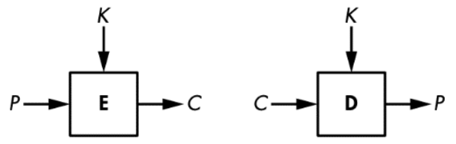


Figure 1: Basic encryption and decryption

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).

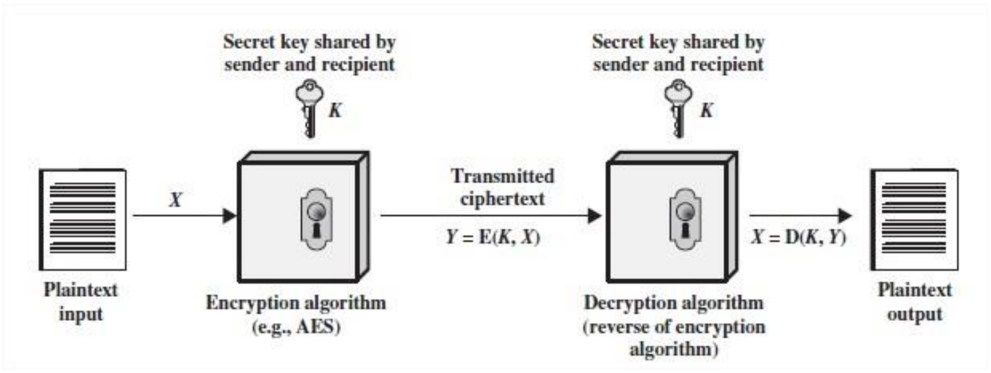

Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
Y = E(K, X)
X = D(K, Y)
Plaintext input
Encryption algorithm (e.g., AES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Figure 2: Symmetric Encryption model

Classical ciphers*: are ciphers that predate computers, therefore work on letters rather than on bits and almost are symmetric ciphers.* The two basic building blocks of all encryption techniques are *substitution* and *transposition* (Figure 3)

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.

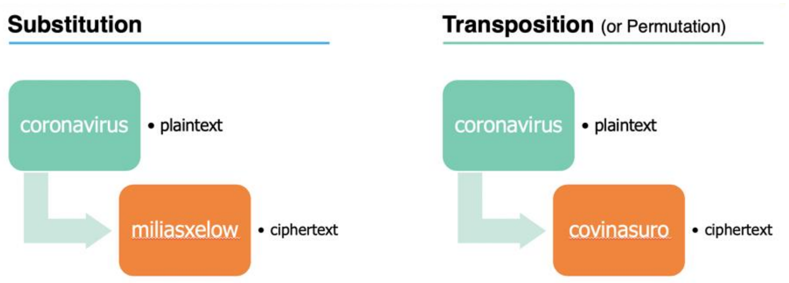


Figure 3: Substitution and Transposition example

B. LAB TASKS

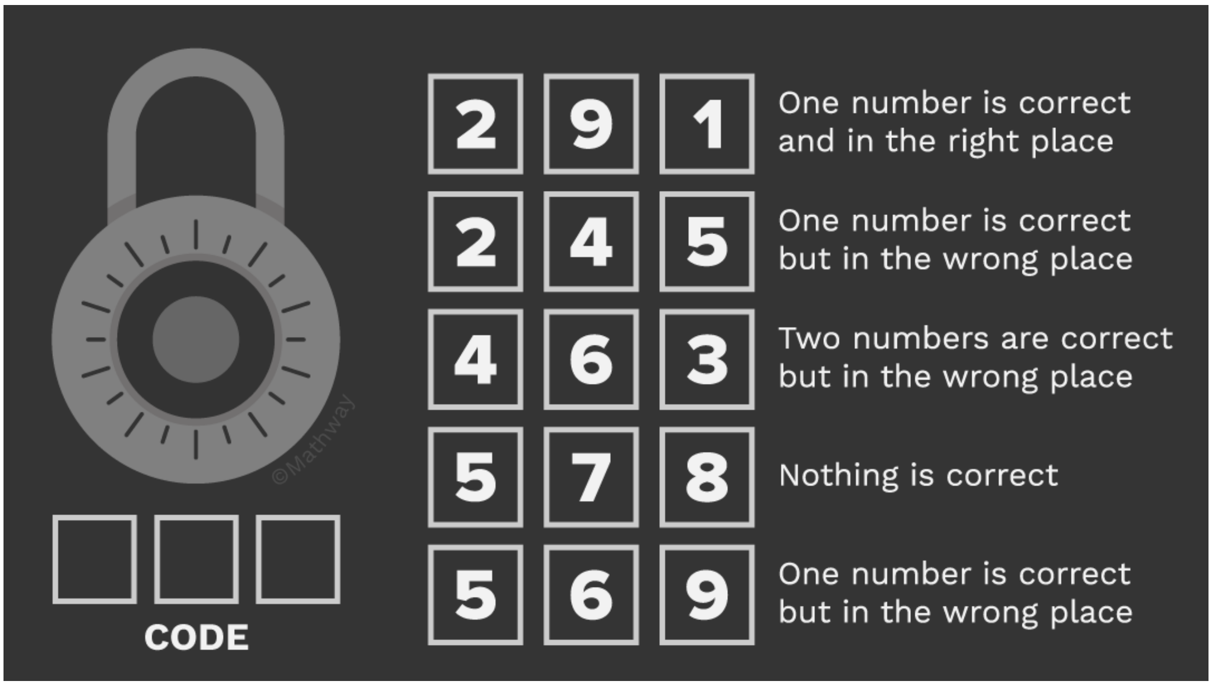
1. Kickoff: Crack the code

Task 1.1

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 inFigure 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 either a one-digit number or a two-digit number. Thay can be different from each other.



2	9	1	One number is correct and in the right place
2	4	5	One number is correct but in the wrong place
4	6	3	Two numbers are correct but in the wrong place
5	7	8	Nothing is correct
5	6	9	One number is correct but in the wrong place

Figure 4 Crack the code to open the lock

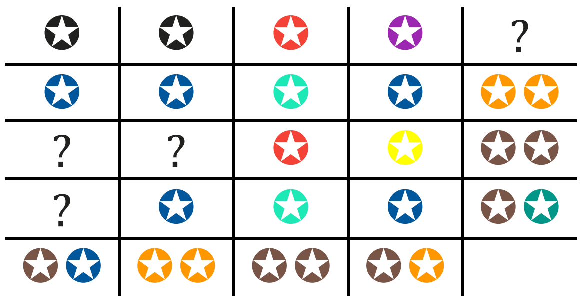


Table 1: Find the corresponding encoding for each number

2. Caesar cipher

Task 2.1

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.

Task 3.1

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 public tools to support 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.*
- ***Mayzner****: This web page provides frequencies for a typical English plaintext, including unigram, bigram, and trigram frequency.*

Advanced Task 3.2

Decrypt a cipher-text from Edgar Allan Poe's - The Gold-Bug and explain how to do

53‡‡†305))6*;4826)4‡.)4‡);806*;48†8¶60))85;;]8*;:‡*8†83(88)5*†;46(;88*96*?;8)*‡(;485);5*
†2:*‡(;4956*2(5*-
4)8¶8*;4069285);)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 is known that:

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

4. Playfair cipher

Task 4.1

Encrypt and decrypt using Playfair cipher. You are able to use public tools or your own tool that was developed in the advanced task 4.2.

Given the cipher text:
ARYWYPHCBVEBYGMPNCYGCNTDNCWTMGRMFTQPLEWTMLREFBEBQEBIYGBFL
PHVOAEHKDHEUNGQFEROLEWTMLOPHEQGOSBEROQDWTLCMTHBWLNRKXRYL
ORYYPHCBVEBYRLGYDMKYGGWKLROANDBWGNERMNGYRLGHEWRTRLMBRHM
UDGVODVTEGMCHLGWCMTFODNRRYCMZKODDUTDXGEOPOYRMFRMGUKXRYGH
ABROVTGQMCEHPRPEOTSEGEQLARYWYPOTMGQDOEXGOAUDHGUTULTNEHFTF
HPGXGVPHGURBDMEGWKLETCBOTNTFQLTAEHMTUGEOAHEVEROXGVPHGDEW
TEWGQIEDLPILERWPMOATNGQKQEAHBMVRFKBRMKLXODXFREBHMNUKXRYKL
RMFLWDDNCN

The key used was ***Harry Potter***. Let’s decrypt the message.

Advanced Task 4.2

In this extra task, you have to 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 athe given key. plain-text to encrypt or a cipher-text to decrypt using
- Display the Playfair matrix (5x5) corresponding with the given key

Test your program with an message of at least 100 words and compare the result with other cryptography tools (like Cryptool 2) to verify.

5. Polyalphabetic ciphers – Vigenère

Task 5.1

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

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

Tips: Using Vigenère algorithm:

Encrypt *i*i^{th} *letter in plain-text:*

$$C_i = (p_i + k_{i \bmod m}) \bmod 26$$

Decrypt i^{th} *letter in cipher-text:*

$$p_i = (C_i - k_{i \bmod m}) \bmod 26$$

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

Advanced Task 5.2

How to crack Vigenere cipher without knowing the key? Explain how and try to crack the following ciphertext which was encrypted using Vigenere cipher

cv vvobobxy uocmgjg, olgiaqsliioa dynxyu fi axjwdy qgmqdgimpqz uvw jqygcgpemnqhu vqwpgpsgya wltupmw mtag utajqgimpemf khueqjbl hpp u axa qr lcel-dmmmw jcxwcehvuivl jcxfmw hnsizbajym bh atmhayvty gmlzcsya bu ymsa mocf uzx ocdx bh kgocxalt. fbmll fqnmktkzcampe mfohykfbul htq oaxk hal kkfrfiokhrtck dla syvxycfcwg hpp xqzpvmf abnpuho tuf hyzbmkoubbvp fi xkvvqwb whvm jzbccos, exi ddielpps iv mog uhbxypqn igk eahnbkgznqts eagunukoubbvpe mcvo ce wzxkkf wikk vduvlhefcwgz czx mfhkx

6. Advanced Task

Advanced Task 6.1

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

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 two or three to get the highest efficiency.

Your submission must meet the following requirements:

- You need to submit a **detailed lab report in .pdf** format, **using the report template** provided on the UIT Courses website.
- Either Vietnamese or English report is accepted, that's up to you. The report written in the mixing of multiple languages 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 in your report. Simply attaching code without any explanation will not receive points.
- 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 the owner and the 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, chapter 4, chapter 6, chapter 7*

[2] Wenliang Du (Syracuse University), *SEED Cryptography Labs*
https://seedsecuritylabs.org/Labs_20.04/Crypto/

[3] Wenliang Du (Syracuse University), *SEED Cryptography Labs*
https://seedsecuritylabs.org/Labs_20.04/Files/Crypto_Encryption

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!*