CADT IDT វិទ្យាស្ថានបច្ចេកវិទ្យាឌីជីថល Institute of Digital Technology

Department of Telecoms and Networking

Course Title: Cryptography

Term 1 | Year 3

Assigment Title: Mini-CyberChef

Lecturer: Mr. Meas Sothearath

Submission date: [20/12/2025]

Name: Savon Chanserey

ID: IDTB100030

Group: 2

I. Introduction

1. Overview

In the field of cybersecurity of the present-day world, cryptography is a critical part that assists in data protection by encoding, encryption, and secure communication methods. However, many beginners find it difficult to understand how cryptographic operations work because they often require writing code or manually applying complex algorithms. This forms a learning barrier to those students that are only beginning to learn about cybersecurity and cryptography.

2. Problem

Simple encoding and ciphering, like Base64, Hex, ROT13, simple substitution, and more, are the most basic methods that most beginners have a hard time identifying and using. They might not have the clue of what method was applied and how to decode unknown encoded text when they are presented with unknown encoded text. Moreover, the application of cryptographic functions in the code may be difficult among the learners who are not intensive learners of programming yet.

3. Solution

In this project, Mini-CyberChef is presented, and that is a lightweight and user-friendly tool that is easy to learn because the tool is simplified. The application has a graphical interface (GUI) that allows the users to encode, decode, and analyze the text without writing any type of code. It also has a magic feature which attempts to automatically find common encodings and tries to decode them.

4. Motivation

Mini-CyberChef was developed with the goal of providing easier and more enjoyable access to cryptography by amateurs. The project provides users with a visual representation of how text is transformed with various encoding and cipher strategies by providing an interactive and hands-on tool. This strategy promotes trial and error, develops hunch, and decreases the intricacy commonly attached to the acquisition of cryptographic functions. The tool can be a useful initial step to learners who are interested in cybersecurity, and CTF (Capture The Flag) challenges.

5. Related Cryptographic Concepts

Mini-CyberChef presents a number of some basic concepts including Base64 encoding, hexadecimal representation, Caesar/ROT ciphers, and basic decoding detection. The knowledge of these concepts forms a base to further complex concepts such as hashing, block ciphers and secure communication protocols.

II. System Design / Architecture

Mini-CyberChef is designed in a modular and pipeline-based design. The system enables users to execute several cryptographic and text-processing functions one after another, analogous to an actual workflow of data transformation.

1. System Architecture Overview

The application is a combination of three key components:

- **Graphical User Interface (GUI)**

  Gives the system communication with the user. Users type, choose functions and see results.

- **Controller / Recipe Engine**

  In charge of the order of execution of the operations chosen by the user. It transfers the data of one process to another.

- **Operation Modules**

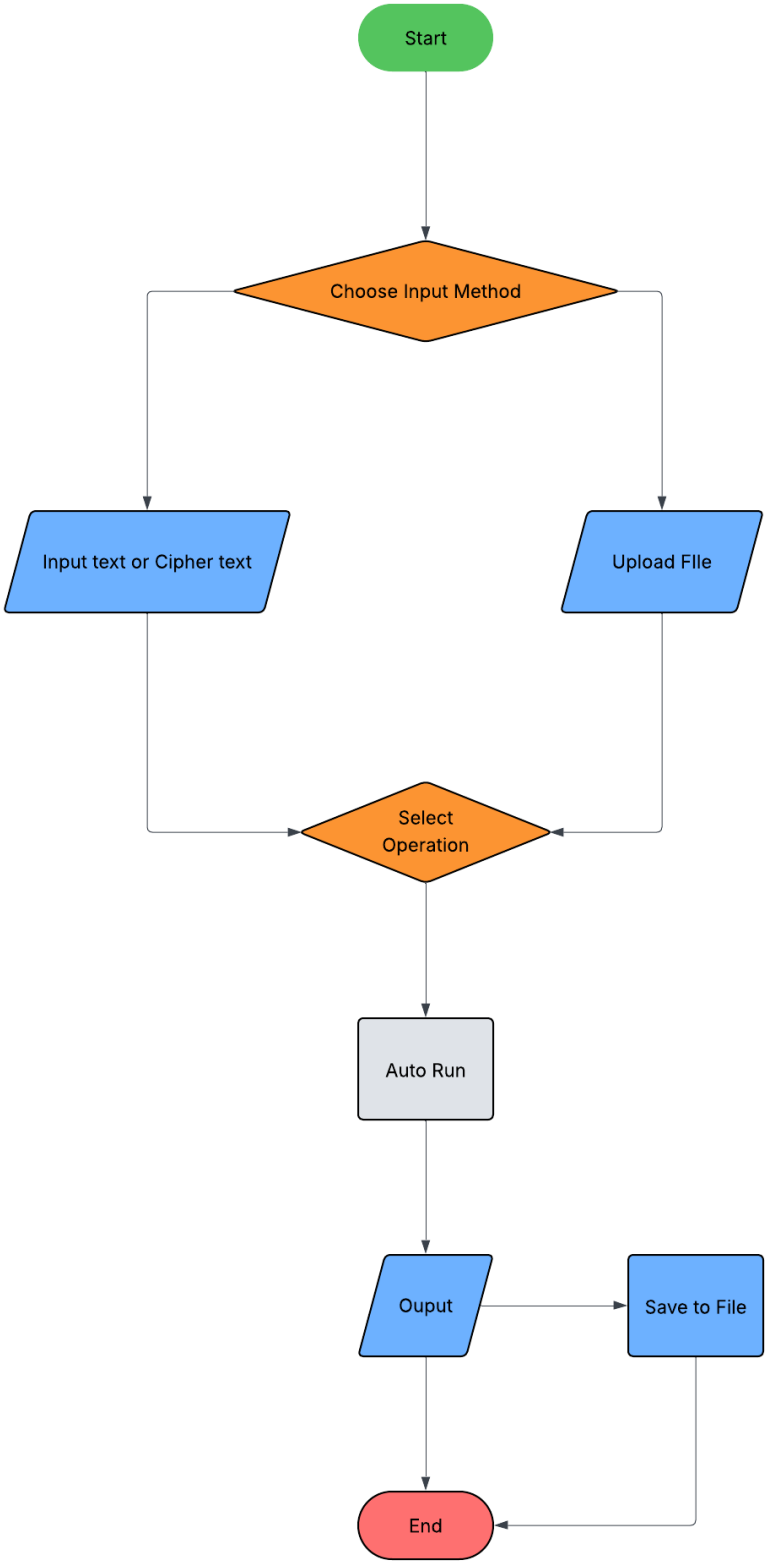
  Encoding, decoding, hashing, ciphering, text processing modules, which are independent Python modules.

1. Data Flow Architecture

The data flow is linear, and it is a pipeline model:

1. The user will input a text in the input box or upload file
2. User picks operations and puts them to the recipe list
3. The recipe engine holds the operations in a Python list
4. The output of one operation is processed by the next one
5. The overall output is presented in the output area

2. Data Flow Diagram



III. Implementation Details

Mini-CyberChef is built using Python 3. The code is organized in a simple and clear way so it is easy to understand, fix, or add new features later.

1. Programming Language and Libraries

- **Python3:** Main Programming Language

- **Tkinter:** Used to Create Graphical User Interface

- **Python Standard Libraries**:

    + **base64:** Base32, Base64, Base85 encoding and decoding.

    + **binascii:** Hexadecimal and binary conversions.

    + **hashlib:** Cryptographic hash functions.

    + **re:** Pattern matching for extraction features.

    + **html:** HTML entity encoding and decoding.

    + **zlib:** CRC32 checksum.

- **External Libraries**:

    + **base58:** Base58 encoding and decoding.

    + **base45:** Base45 encoding and decoding.

    + **base92:** Base92 encoding and decoding

2. Key Components of the Code

a) main.py

This is the file that contains an entry point to the application. It initializes the Tkinter window and loads the main GUI.

b) gui/app.py

This module handles:

- Input and output text areas
- Drag-and-drop operation list
- Recipe execution logic
- Clear button
- Auto run
- Upload and save file

c) Operations/ (Directory)

The file in this directory are Python files representing one independent operation each. Each operation is implemented as function that:

- Accepts text (or bytes) as input
- Process the data
- Returns the result

3. Data Structure Used

- Python List

Used to store the ordered recipe of selected operations

- Strings and Byte Objects

Used to represent input and output data depending on the operation type

4. Crpytographic and Alogorithm Methods Used

Mini-CyberChef implements several fundamental cryptographic and data transformation methods.

a) Encoding and Decoding Modules

Encoding schemes are reversible transformations used to represent data in different formats:

- Base32, Base45, Base58, Base62, Base64, Base92
- Binary, Octal, Decimal, Hexadecimal
- URL and HTML entity encoding/decoding
- Morse code and Character code conversions

b) Classical Cipher Algorithms

Classical ciphers are implemented to demonstrate basic encryption concepts:

- ROT13 and ROT47: Fixed letter-shifting ciphers
- Vignere Cipher: Polyalphabetic substitution cipher
- Bacon Cipher: Binary representation using A/B patterns

c) Hashing Alogrithms

Hashing algorithms are one-way cryptographic functions used to generate fixed-length digests:

- MD5
- SHA1
- SHA256
- SHA512

d) Text Processing and Utility Operations

- Uppercase / Lowercase
- Reverse string / Reverse line
- Remove duplicates / Remove whitespace
- Count characters
- Extracts IP addresses and URLs
- CRC32 checksum

e) Magic Detection Algorithm

The magic feature (magic.py) attempts automatically detect unknown encodings by:

- Checking character patterns
- Comparison of various approaches to decoding
- Evaluating output readability
- Outputttting the most probable outcome of the decoding process and its type

f) Error Health and Safety.

Every operation therein is enclosed with try-except block to:

- Avoid crashing of applications.
- Process invalid input forms.
- Give messages on error which are easy to read.

This makes the application stable even when it is provided with erroneous input.

5. Usage Guide

1. System Requirements

- Python 3.8 or later
- Windows, macOS or Linux

2. Installation Steps

a) Clone the Repository:

- git clone https://github.com/SavonChanserey/Mini-CyberChef.git
- cd path/to/Mini-CyberChef

b) Create a Virtual Environment:

- python3 -m venv venv

c) Activate the Virtual Environment

- For macOS/Linux:

+ source venv/bin/activate

- For Window:

+ venv\Scripts\activate

d) Install dependencies

- pip install -r requirements.txt

(In requirements.txt, it contains base58, base45, and base92.)

e) Run the application

- python3 main.py

To exit the virtual environment:

- deactivate

3. How to Use the Program

- Launch the application (python3 main.py)

- Enter the text or load file into the Input Area

- Drag & drop the operation into the Recipe

- Auto run the process

- View the result in the Output Area

4. Examples

a) Base64

- Input: SGVsbG8gd29ybGQh

- Operation: From Base64

- Output: Hello world!

b) Magic

- Input: ;K_$aOB

- Operation: Magic

- Output: [Base92] Hello

c) Hashing

- Input: Password123

- Operation: MD5

- Output: 42f749ade7f9e195bf475f37a44cafcb

d) Binary

- Loaf file: flag.txt (it contains some binary)

- Operation: From Binary

- Output: CTF{First_Flag}

IV. Conclusion and Future work

1. Conclusion

Mini-CyberChef can be an entry-level and user-friendly learning tool in basic cryptography. The tool can eliminate the complexity of programming by providing a graphical interface and a modular design that helps users to concentrate on the operation of cryptographic operations.

The system of recipes that can be drag-dropped and the Magic detection feature make the process of learning interactive and realistic. Students, cybersecurity beginners, and those who prepare to take part in CTF may find this project particularly handy.

2. Future Work

Even though Mini-CyberChef currently offers effective cryptography-related learning functionality, its practical application and educational potential can still be enhanced in future upgrades.

a) Modern Cryptography Algorithms

The next generation of Mini-CyberChef can add current encryption algorithms, like AES and RSA. This would serve to teach the users both symmetric and asymmetric encryption schemes in the real world.

b) Better Image Rendering and Visualization.

The application can be improved by displaying images more clearly and improving the visual presentation of encoded or decoded results, making the tool easier to use and understand.

c) **Increased Magic Detection Accuracy**

The Magic feature can be enhanced to better detect by incorporating more encoding types and to better detect by incorporating more discerning ways of detection.

d) **Improvements to Graphical User Interface**

To make the application easier to use, it can be enhanced by including improved layouts, icons, themes as well as tooltips to make the application more user-friendly and attractive.

e) **Single Packaging of Applications**

Mini-CyberChef can be compiled as a standalone installation package on Windows, macOS, and Linux, whilst the user is provided the ability to execute the tool without needing to install Python or other dependencies.

V. References

1. https://gchq.github.io/CyberChef/
2. https://docs.python.org/3/library/tkinter.html
3. https://docs.python.org/3/library/base64.html
4. https://docs.python.org/3/library/hashlib.html
5. https://docs.python.org/3/library/zlib.html
6. https://docs.python.org/3/library/re.html
7. https://www.geeksforgeeks.org/python/morse-code-translator-python/