Final Project Report: EECE2140 COMPUTING FUNDAMENTALS FOR ENGINEERS

Enocder/Decoder project

Student Names: Noah Yue and Isaac Rounds
Northeastern University
College of Engineering
Department of Electrical and Computer Engineering

Course Title: EECE 2140: COMPUTING FUNDAMENTALS FOR ENGINEERS
Instructor: Fatema Nafa
Spring Semester 2024

April 14, 2024

1 Abstract

The objective of our project was to create a program capable of reading data from a file, encoding or decoding it, and then writing the finished product to a new file. The program was written in Python and uses a basic command prompt user interface. The library cryptodome was used for creating cipher objects and the code is primarily structured using classes.

The program created successfully reads .txt and .pdf files (for reading .pdf files the library pypdf was used), encodes or decodes, and then writes to a .txt file

this program uses three ciphers: Shift, DES (Data Encryption Standard), and AES (Advanced Encryption Standard).

DES and AES encode and decode their data in bits - because of this this project is also capable of converting strings to bits and vice versa.

2 Objective and Introduction

This program was designed to read data from .pdf and .txt files, encode or decode that data (in AES, DES, or using shift encryption) and then write the results to a .txt file. Data encryption is an extraordinarily important part of the modern internet. Gaining an understanding of how data encryption works is very important to us, as future engineers who will be responsible for designing projects that are secure. To get an understanding of encryption from the ground up we designed an extraordinarily simple cipher first, the shift cipher. Then we made the DES cipher, which is now obsolete. Finally we designed the AES cipher, which is still in use today.

3 Theory and Experimental Methods

We have used descriptive function and variable naming (e.g., read_file, write_file, AESCipher, etc.) in our code to clearly indicate what they are used for. This naming follows a straightforward pattern that allows the user to clearly understand what the class or function does. Our comments are used to explain the function of the code block and it appears in all places in all programs. However, we should have paid more attention to the comments, including more details about parameters and expected results, as well as briefly describing exactly what the statement does; we didn't have enough of them. Code is organized into sections that deal with file input/output, data manipulation, and user interaction. Our code is almost always placed in functions or classes. This is done for two reasons - it improves readability, and it makes our code easy to segment, allowing us both to work on different files at the same time. Keeping our code highly segmented also allows us to easily deduce the source of errors and fix them. Because of the frequent use of these functions and classes, their naming is very important. All of the crypto classes are named [cipher type]Cipher, and these crypto classes have a method named encrypt and decrypt which, when called, encrypt and decrypt the data sent to them. We use GitHub for version control, whenever we make a change or addition to a program, we comment out the new code and push it to GitHub and create a new Branch for file exchange, eventually all of our files and uploads will be in the comments. We adhere to the PEP8 standard most of the time, which lets us use a uniform format for variable names and line lengths and functions, and helps us get our work handed off efficiently.

We use a hybrid structure of classes and functions to complete the project, our project contains four classes and four functions, the encapsulation of some functions can speed up the operational efficiency, but also easy to maintain and subsequent further expansion. The modular data structure facilitates the splitting of different parts, which makes their functions not be affected by each, and we can better manipulate the data. Regarding the four classes, they are MData class, AESCipyher class, DESCipher class and AlphCipher class, which encapsulate the processing of data and the three encryption methods, this object-oriented approach is suitable for maintaining and manipulating data state in different operations.
While writing the main program we faced many challenges, how to keep the code generic when dealing with different files and how to make sure that the data type is controllable during encryption, the former problem we solved with the read_file function, which can read the content in different ways according to different extensions, and the latter problem we solved with the MData class, which helps us before and after the encryption to transform the original data into the data type we need. Our extensibility is reflected in the structure of the program. If we need to add a new encryption method, we just need to add a new class or function, and we have added error handling at every step in case of unanticipated input.

Listing 1: Python example

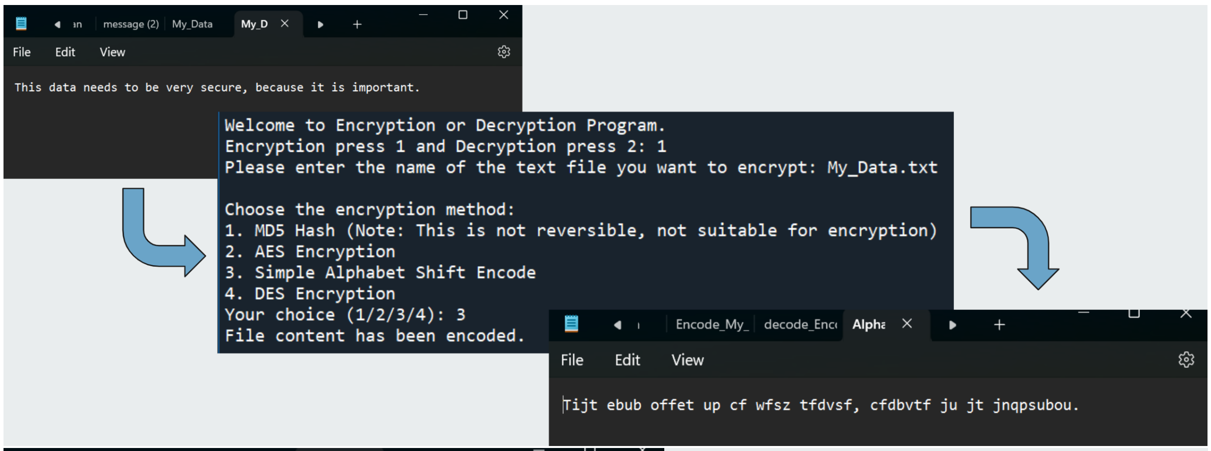
```python
class AESCipher:
    def __init__(self, key):
        self.key = key  # encryption key

    def encrypt(self, data):
        try:
            cipher = AES.new(self.key, AES.MODE_CBC)
            ciphertext = cipher.encrypt(data)
            return base64.b64encode(ciphertext).decode('utf-8')
        except Exception as e:
            raise ValueError(f"Encryption failed: {e}")

    def decrypt(self, data):
        try:
            data = base64.b64decode(data)
            cipher = AES.new(self.key, AES.MODE_CBC)
            plaintext = cipher.decrypt(data)
            return plaintext.decode('utf-8')
        except Exception as e:
            raise ValueError(f"Decryption failed: {e}")
```

4 Results

Code input and output:

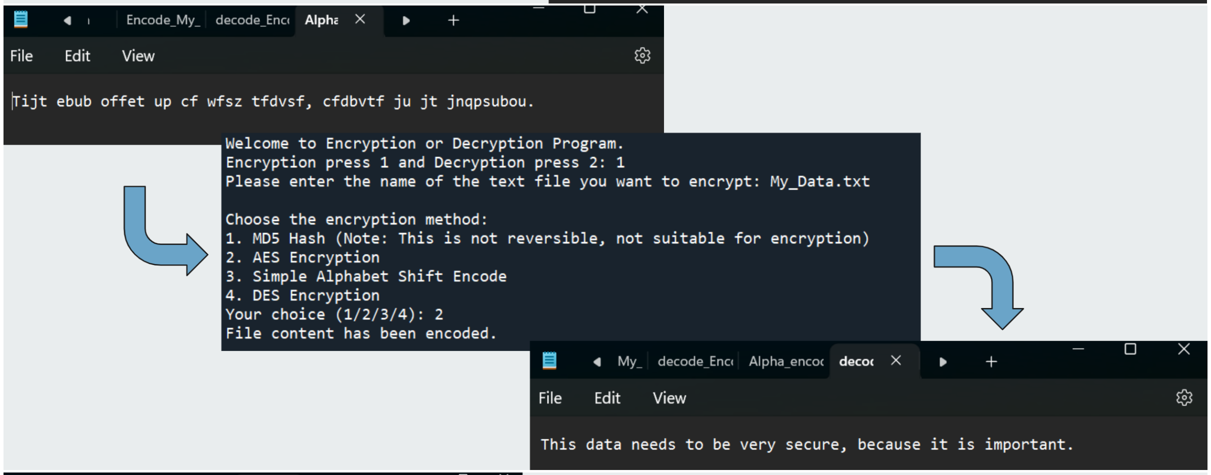


```
This data needs to be very secure, because it is important.
```

```
Welcome to Encryption or Decryption Program.
Encryption press 1 and Decryption press 2: 1
Please enter the name of the text file you want to encrypt: My_Data.txt

Choose the encryption method:
1. MD5 Hash (Note: This is not reversible, not suitable for encryption)
2. AES Encryption
3. Simple Alphabet Shift Encode
4. DES Encryption
Your choice (1/2/3/4): 3
File content has been encoded.
```

```
Tijt ebub offet up cf wfsz tfdvsf, cfdbvtf ju jt jnqpsubou.
```

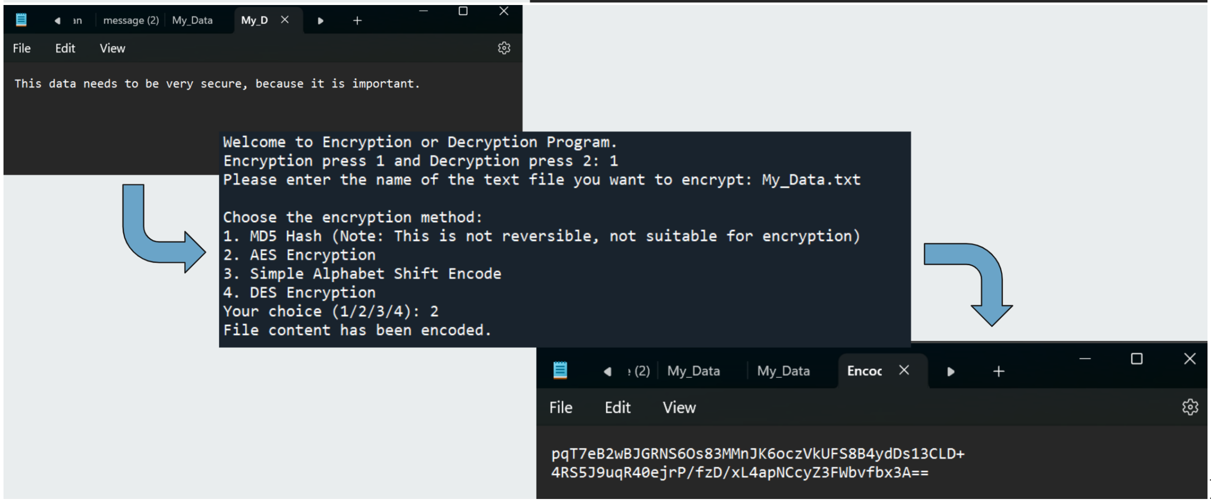


```
Tijt ebub offet up cf wfsz tfdvsf, cfdbvtf ju jt jnqpsubou.
```

```
Welcome to Encryption or Decryption Program.
Encryption press 1 and Decryption press 2: 1
Please enter the name of the text file you want to encrypt: My_Data.txt

Choose the encryption method:
1. MD5 Hash (Note: This is not reversible, not suitable for encryption)
2. AES Encryption
3. Simple Alphabet Shift Encode
4. DES Encryption
Your choice (1/2/3/4): 2
File content has been encoded.
```

```
This data needs to be very secure, because it is important.
```



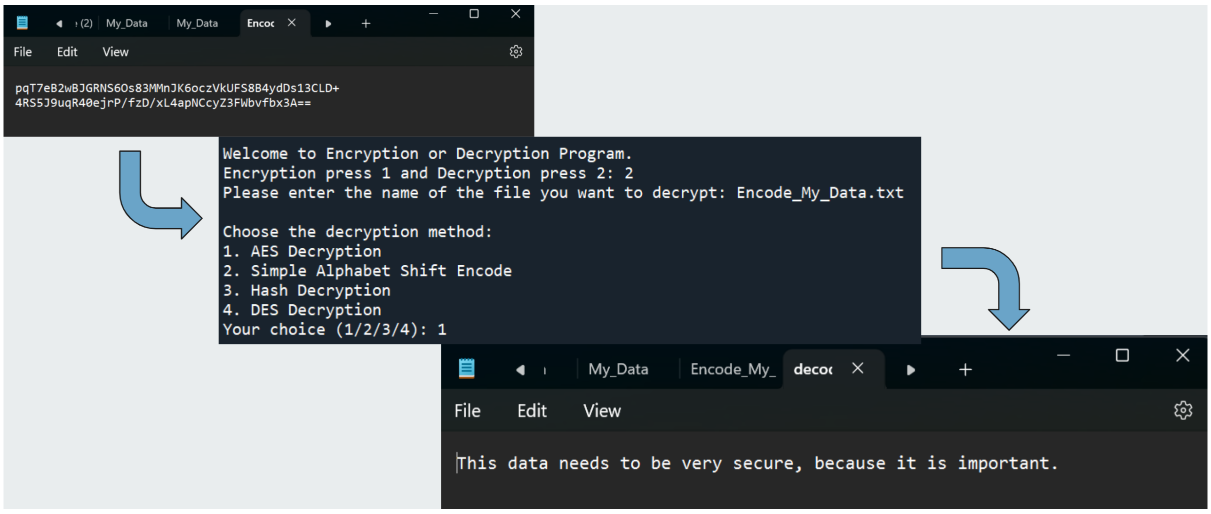
```
This data needs to be very secure, because it is important.
```

```
Welcome to Encryption or Decryption Program.
Encryption press 1 and Decryption press 2: 1
Please enter the name of the text file you want to encrypt: My_Data.txt

Choose the encryption method:
1. MD5 Hash (Note: This is not reversible, not suitable for encryption)
2. AES Encryption
3. Simple Alphabet Shift Encode
4. DES Encryption
Your choice (1/2/3/4): 2
File content has been encoded.
```

```
pqT7eB2wBJGRNS6Os83MMnJK6oczVkUFS8B4yDs13CLD+
4RS5J9uqR40ejrP/fzD/xL4apNCcyZ3FWbvfbx3A==
```

1



```
pqT7eB2wBJGRNS6Os83MMnJK6oczVkUFS8B4ydDs13CLD+
4RS5J9uqR40ejrP/fzD/xL4apNCcyZ3FWbvfbx3A==
```

```
Welcome to Encryption or Decryption Program.
Encryption press 1 and Decryption press 2: 2
Please enter the name of the file you want to decrypt: Encode_My_Data.txt

Choose the decryption method:
1. AES Decryption
2. Simple Alphabet Shift Encode
3. Hash Decryption
4. DES Decryption
Your choice (1/2/3/4): 1
```

```
This data needs to be very secure, because it is important.
```

5 Conclusions

The program was successful. It is able to encode and decode data as well as reading and writing to .txt files. We learned more about how ciphers function, including ciphers used in the modern day. We now understand more about how ciphers take binary data and scramble/unscramble the bits.

6 Limitations and Future Work

the program is unable to write to .pdf files. This is because the pypdf library does not support writing to .pdf files. The pypdf library is also not 100 percent reliable when reading from pdfs. The program also does not save new lines in the encrypted data, so when data is run through the program it will not have the new line's the original did. A future feature that would improve the usefulness of the program is a way for the program to attempt to figure out what cipher the encoded data needs to be decoded.

7 References

- https://blog.csdn.net/chouzhou9701/article/details/12201996
- https://sectigostore.com/blog/what-is-des-encryption-a-look-at-the-des-algorithm/
- https://pypi.org/project/pycryptodome/
- https://pypi.org/project/pypdf/
- https://docs.python.org/3/library/hashlib.html
- https://docs.python.org/3/library/base64.html