University of San Carlos
School of Arts and Sciences
Department of Computer, Information Sciences and Mathematics
Talamban Campus, Cebu City, Philippines

CIS 3106 - Information Assurance and Security

Cryptography Project:
MARV

Submitted by:
Juma-ang, Curtney Sealtiel Mata

Submitted to:
Mr. Godwin Monserate

May 2025

Project: Cryptography

MARV

(Monoalphabetic.Atbash.RSA.Vignere)

Introduction:

This algorithm is developed using python and it is a combination of hybrid encryption creating an onion layer which helps the users encrypt their data and store them in a local system using file handling.

Algorithms Used:

1. **Monoalphabetic Cipher**
    - Third Layer. This uses a substitution rule and each letter in the encrypted text is replaced by another letter based on the fixed mapping. This basically just scrambles the once encrypted ciphertext.

2. **Atbash Cipher**
    - Second Layer. Its function is basically to mirror the encrypted ciphertext . That means the counter part of A is the last letter of the alphabet which is Z making it unpredictable for the attacker to expect.

3. **RSA Cipher**
    - Last Layer. At the start of MARV users are required to generate two keys namely: Private(for encryption) and Public(for decryption). This is basically the hardest algorithm to crack since it makes use of factoring large prime numbers in which you have to use the Euler’s Formula in getting the Totient.

4. **Vigenere Cipher**
    - The first layer. Basically it uses a keyword phrase instead of a value shift phrase which starts the layer the security of the said algorithm. This basically just helps the algorithm with its onion layer.

‘

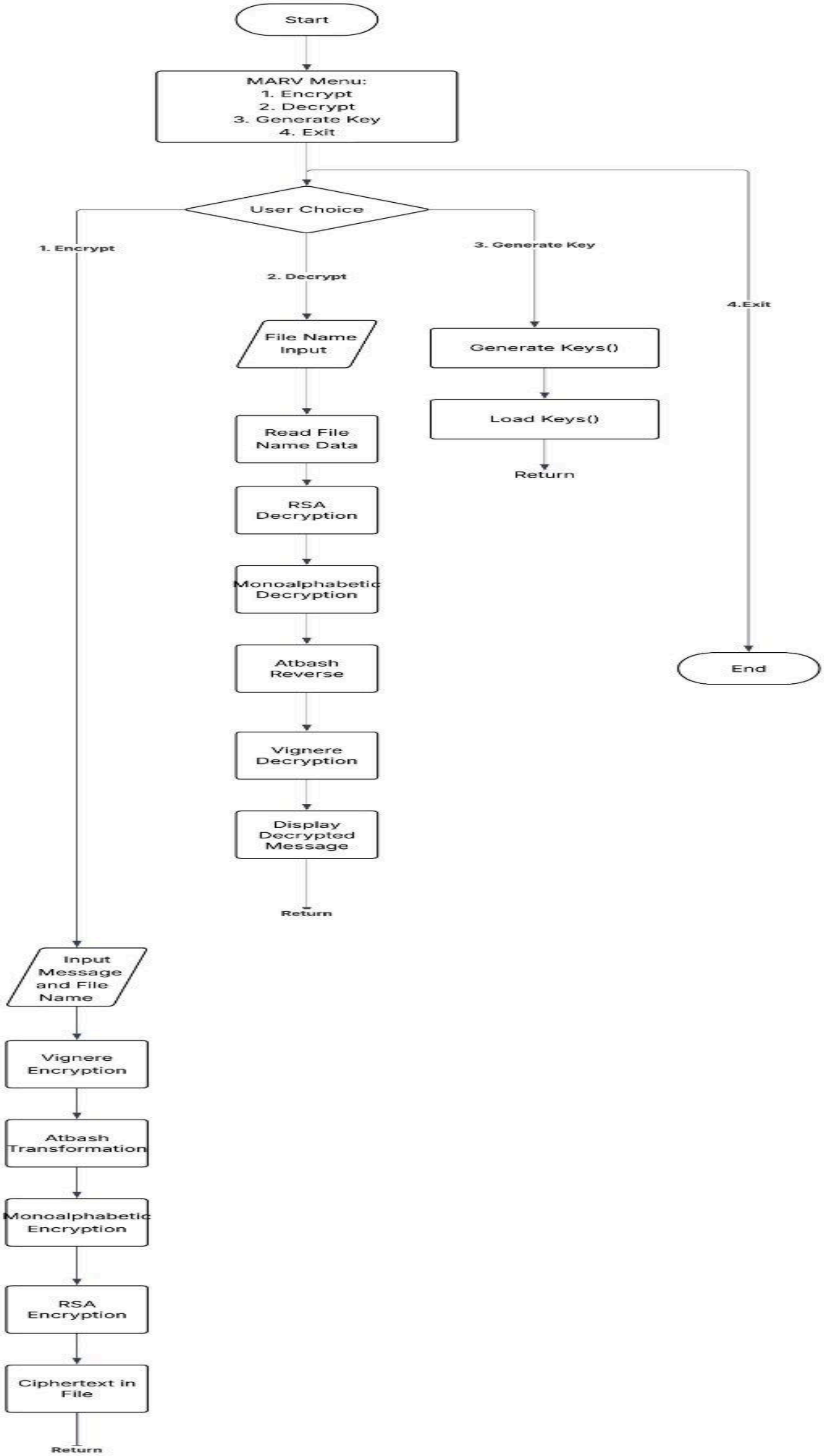
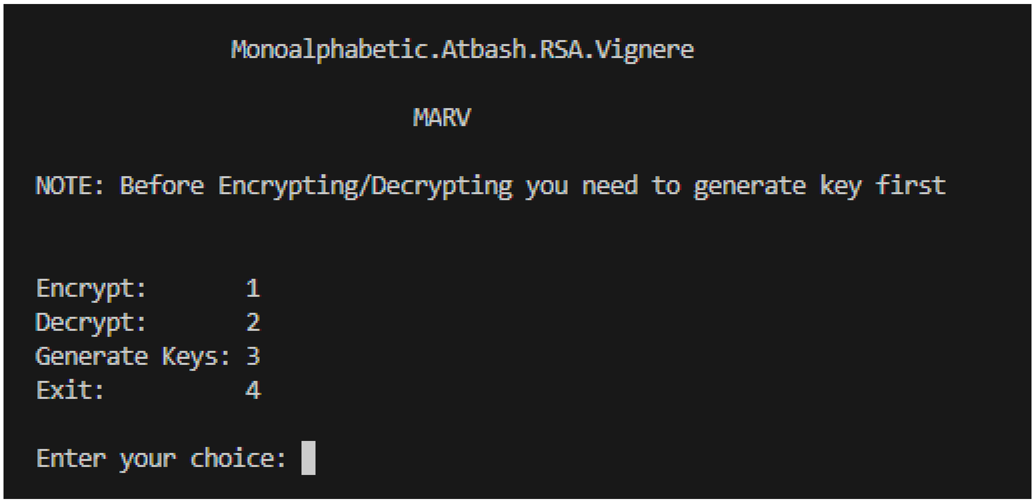

Start
MARV Menu:
1. Encrypt
2. Decrypt
3. Generate Key
4. Exit
User Choice
1. Encrypt
2. Decrypt
3. Generate Key
4.Exit
File Name Input
Generate Keys()
Load Keys()
Return
Read File Name Data
RSA Decryption
Monoalphabetic Decryption
Atbash Reverse
Vignere Decryption
Display Decrypted Message
Return
End
Input Message and File Name
Vignere Encryption
Atbash Transformation
Monoalphabetic Encryption
RSA Encryption
Ciphertext in File
Return

Figure. 1 MARV Flowchart of processes.

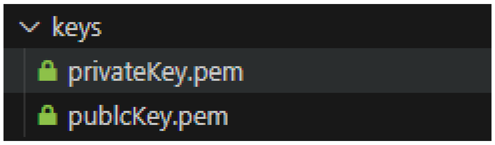
Process:

1. Marv starts by letting the user choose between the four choices namely: Encrypt, Decrypt, Generate Keys, and Exit.

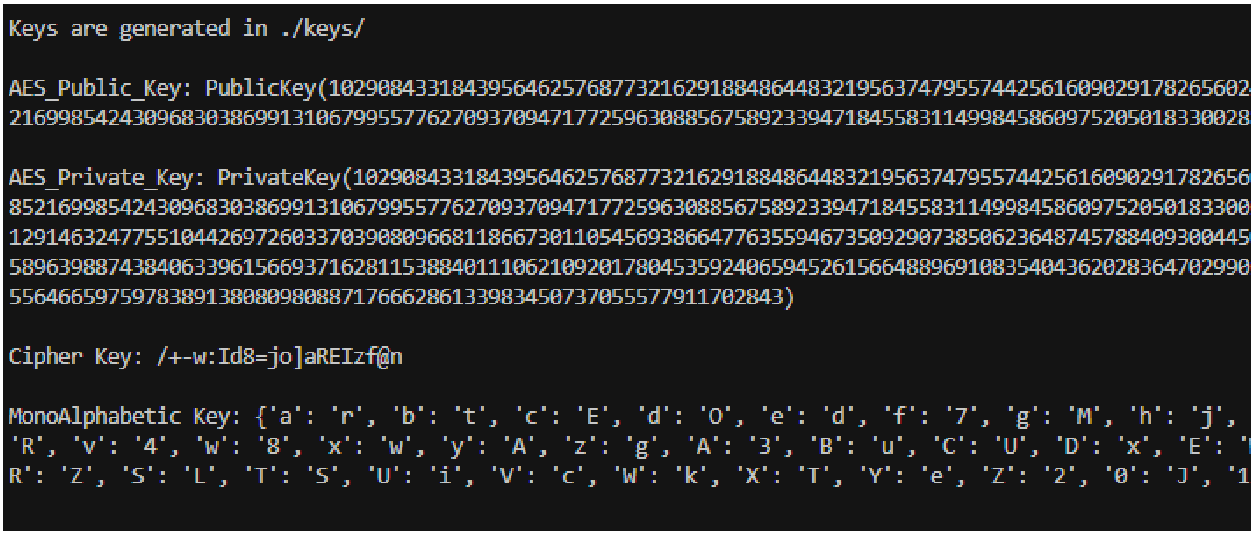


```
                 Monoalphabetic.Atbash.RSA.Vignere

                              MARV

NOTE: Before Encrypting/Decrypting you need to generate key first


Encrypt:       1
Decrypt:       2
Generate Keys: 3
Exit:          4

Enter your choice: 
```

2. As said in the note it's required for the user to generate the key first so press 3 and generate the key. Once the key has been generated it will be stored in the folder of keys and will be stored in the said directory.

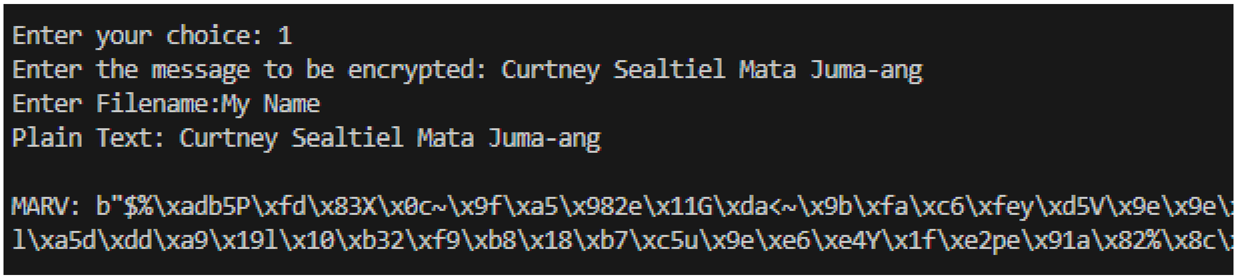


```
keys
  privateKey.pem
  publcKey.pem
```



```
Keys are generated in ./keys/

AES_Public_Key: PublicKey(102908433184395646257687732162918848644832195637479557442561609029178265602
21699854243096830386991310679955776270937094717725963088567589233947184558311499845860975205018330028

AES_Private_Key: PrivateKey(1029084331843956462576877321629188486448321956374795574425616090291782656
85216998542430968303869913106799557762709370947177259630885675892339471845583114998458609752050183300
12914632477551044269726033703908096681186673011054569386647763559467350929073850623648745788409300445
58963988743840633961566937162811538840111062109201780453592406594526156648896910835404362028364702990
556466597597838913808098088717666286133983450737055577911702843)

Cipher Key: /+-w:Id8=jo]aREIzf@n

MonoAlphabetic Key: {'a': 'r', 'b': 't', 'c': 'E', 'd': 'O', 'e': 'd', 'f': '7', 'g': 'M', 'h': 'j',
'R', 'v': '4', 'w': '8', 'x': 'w', 'y': 'A', 'z': 'g', 'A': '3', 'B': 'u', 'C': 'U', 'D': 'x', 'E':
R': 'Z', 'S': 'L', 'T': 'S', 'U': 'i', 'V': 'c', 'W': 'k', 'X': 'T', 'Y': 'e', 'Z': '2', '0': 'J', '1
```

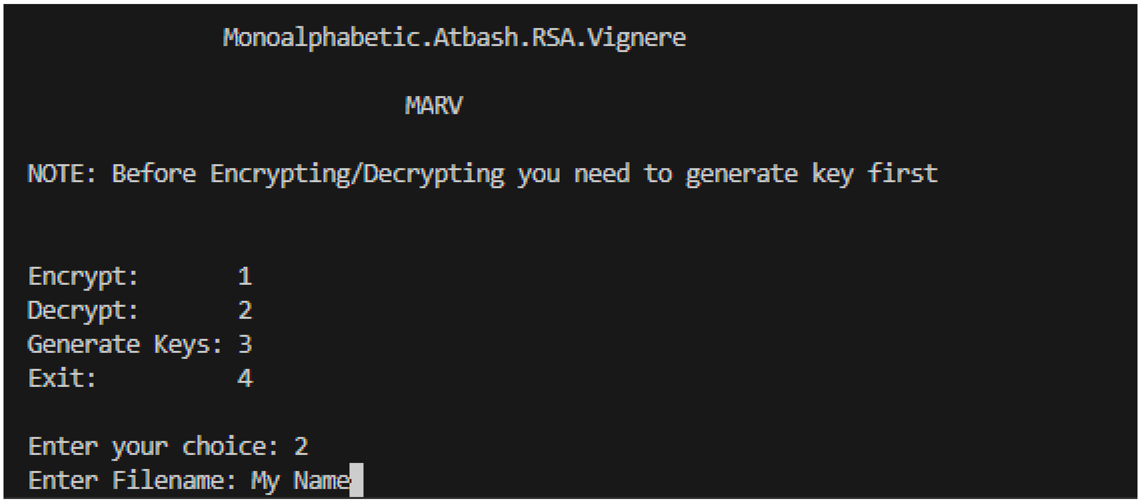
3. Once the keys are created the users can now proceed to encrypt a specific file. They should enter the message, and the filename.



```
Enter your choice: 1
Enter the message to be encrypted: Curtney Sealtiel Mata Juma-ang
Enter Filename:My Name
Plain Text: Curtney Sealtiel Mata Juma-ang

MARV: b"$%\xadb5P\xfd\x83X\x0c~\x9f\xa5\x982e\x11G\xda<~\x9b\xfa\xc6\xfey\xd5V\x9e\x9e\
l\xa5d\xdd\xa9\x19l\x10\xb32\xf9\xb8\x18\xb7\xc5u\x9e\xe6\xe4Y\x1f\xe2pe\x91a\x82%\x8c\
```

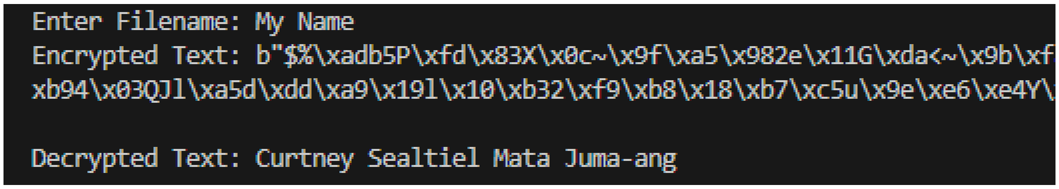
As you can see after putting in both of the input fields, it will show the ciphertext MARV has created.

4. MARV also has a file handling feature which means that it could store both the plain text and encrypted text in a file. That way we can easily input the filename once we try to decrypt the ciphertext.



```
               Monoalphabetic.Atbash.RSA.Vignere

                            MARV

NOTE: Before Encrypting/Decrypting you need to generate key first


Encrypt:        1
Decrypt:        2
Generate Keys: 3
Exit:           4

Enter your choice: 2
Enter Filename: My Name
```

5. Once the user has inputted the file name correctly it will then display the decrypted plaintext along with the once encrypted file.



```
Enter Filename: My Name
Encrypted Text: b"$%\xadb5P\xfd\x83X\x0c~\x9f\xa5\x982e\x11G\xda<~\x9b\xf
xb94\x03QJl\xa5d\xdd\xa9\x19l\x10\xb32\xf9\xb8\x18\xb7\xc5u\x9e\xe6\xe4Y\

Decrypted Text: Curtney Sealtiel Mata Juma-ang
```