

# **ENCRYPTION AND DECRYPTION USING MONOALPHABETIC CIPHER**

## **A COURSE PROJECT REPORT**

By

**MADAVARAM REDDI VINATHI(RA2011030010133)**  
**GUVVALA SAI AMARNADH SATVIC REDDY(RA2011030010124)**

Under the guidance of

**Dr. Mary Subaja Christo**

(Associate Professor, Department of Networking and Communication)

*In partial fulfilment for the Course*

of

**18CSC381T - CRYPTOGRAPHY**

in Networking and Communication



**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Chenpalattu District**

**NOVEMBER 2022**

# **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

## **BONAFIDE CERTIFICATE**

Certified that this mini project report "**Encryption and Decryption using Monoalphabetic Cipher** " is the bonafide work of **Satvic (RA2011030010124)**, **MR Vinathi (RA2011030010133)** who carried out the project work under my supervision.

### **SIGNATURE**

Dr. Mary Subaja Christo  
**Associate Professor**  
**Networking and Communication**  
SRM Institute of Science and Technology

## **ABSTRACT**

In the digital era, being hacked is a common happening worldwide. With communications over the cloud, the privacy of data sent and received, is vulnerable. Cryptography is being a protector by safeguarding the data

communicating between sender and receiver, in such situation we do want anyone else to access our data or private messages. With digital currencies a.k.a. cryptocurrencies on the rise, it is of utmost priority to build a stronger anti-hack mechanism to protect them.

The block-chain, that protects the digital currencies, is fundamentally based on cryptography. The process of converting ordinary and plain text into unintelligible text and vice versa is known as cryptography. In this method, data is stored and transmitted in a specific form in order to make it available for only particular people to read and process.

Its importance lies in the fact that it protects data from hacking and alteration, while making it useful for user authentication. Phil Zimmermann defines cryptography as the science of using mathematics to encrypt and decrypt data Bruce Schneier says,

Cryptography is the art and science of keeping messages secure.

## ACKNOWLEDGEMENT

We express our heartfelt thanks to our honorable **Vice Chancellor Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy**, for his encouragement

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal**, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman**, for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professor Dr. Annapurani Panaiyappan, Professor and Head, Department of Networking and Communications** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **Dr. Mary Subaja Christo , Associate Professor , Networking and Communication**, for his/her assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **Dr. Annapurani Panaiyappan, Course Coordinator, Professor and Head, Department of Networking and Communication** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

## **TABLE OF CONTENTS**

<b>CHAPTERS</b>	<b>CONTENTS</b>
<b>1.</b>	<b>ABSTRACT</b>
<b>2.</b>	<b>INTRODUCTION</b>
<b>3.</b>	<b>DESIGN &amp; IMPLEMENTATION</b>
<b>4.</b>	<b>EXPERIMENT RESULTS &amp; ANALYSIS</b>
	4.1. RESULTS
	4.2. RESULT ANALYSIS
<b>5.</b>	<b>CONCLUSION &amp; FUTURE ENHANCEMENT</b>
<b>6.</b>	<b>REFERENCES</b>

# 1. INTRODUCTION

The substitution cipher is the oldest forms of encryption algorithms according to creates each character of a plaintext message and require a substitution process to restore it with a new character in the ciphertext.

This substitution method is deterministic and reversible, enabling the intended message recipients to reverse-substitute ciphertext characters to retrieve the plaintext.

The specific form of substitution cipher is the Monoalphabetic Substitution Cipher, is known as “Simple Substitution Cipher”. Monoalphabetic Substitution Ciphers based on an individual key mapping function  $K$ , which consistently replaces a specific character  $\alpha$  with a character from the mapping  $K(\alpha)$ .

A mono-alphabetic substitution cipher is a type of substitution ciphers in which the equivalent letters of the plaintext are restored by the same letters of the ciphertext. Mono, which defines one, it signifies that each letter of the plaintext has a single substitute of the ciphertext.

In Monoalphabetic cipher, the substitute characters symbols supports a random permutation of 26 letters of the alphabet.  $26!$  Permutations of the alphabet go up to  $4 \times 10^{26}$ . This creates it complex for the hacker to need brute force attack to gain the key.

Mono-alphabetic cipher is a type of substitution where the relationship among a symbol in the plaintext and a symbol in the cipher text is continually one-to-one and it remains fixed throughout the encryption process.

These ciphers are considered largely susceptible to cryptanalysis. For instance, if ‘T’ is encrypted by ‘J’ for any number of appearance in the plain text message, then ‘T’ will continually be encrypted to ‘J’.

If the plaintext is “TREE”, thus the cipher text can be “ADOO” and this showcases that the cipher is possibly mono-alphabetic as both the “O”s in the plaintext are encrypted with “E”s in the cipher text.

## 2. DESIGN AND IMPLEMENTATION\

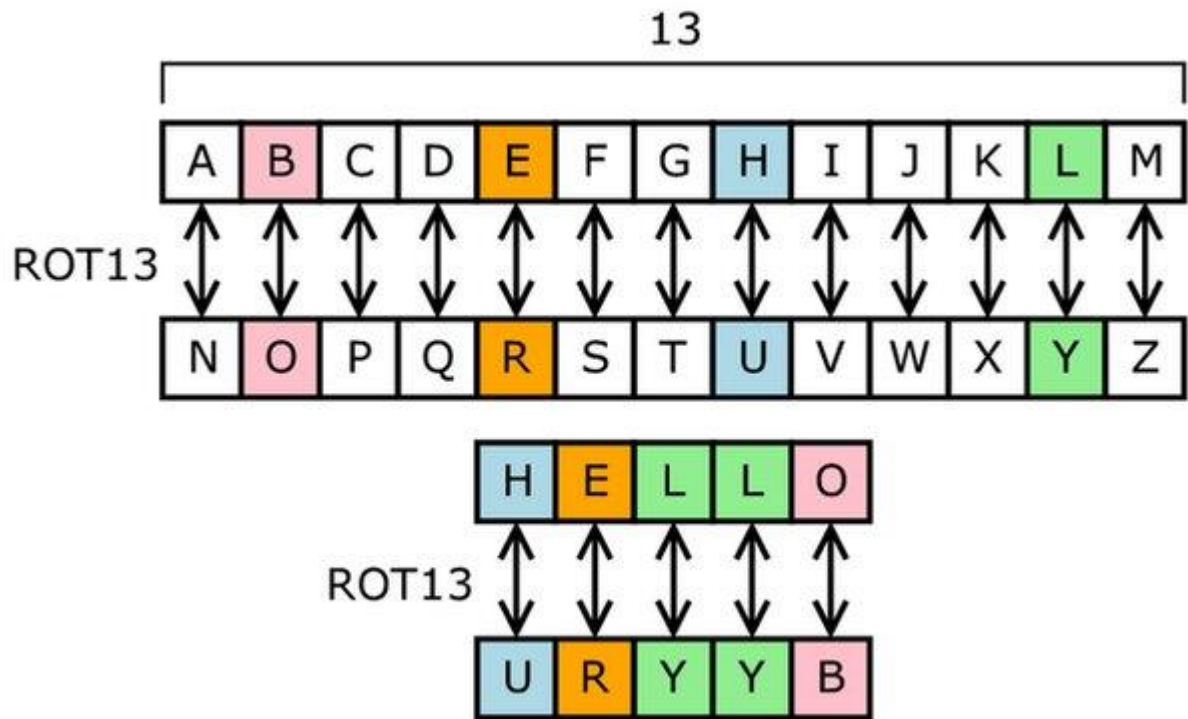


Fig no 2.1: Mapping of letters in monoalphabetic

- Import important libraries
- Define variables and dictionaries
- Create important functions
- Import important libraries

```
from string import letters, digits
from random import shuffle
```

- define a function “message” which takes the input entered by the user
- initialize input alphabets and key logic

- perform encryption by mapping key to the value
- perform decryption by mapping cipher value to the key

```
def message():
    keys='abcefg hijklmnopqrstuvw xyz !'
    value=keys[-1]+keys[0:-1]
    encryptDict=dict(zip(keys,value))
    decryptDict=dict(zip(value,keys))
```

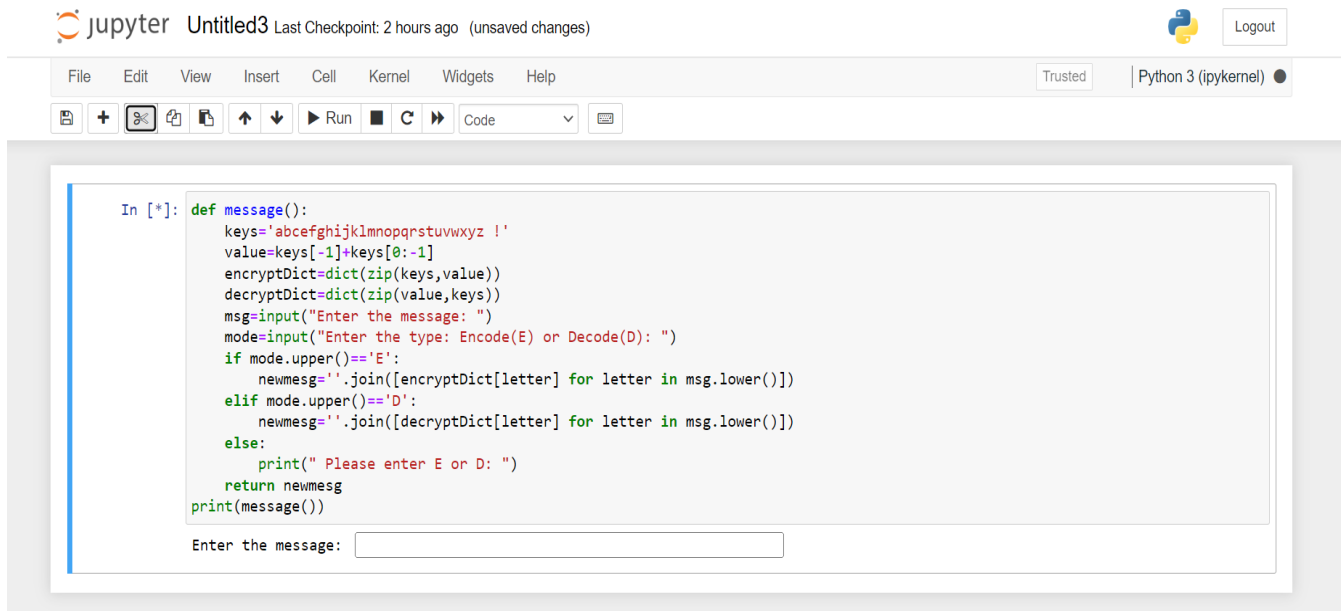
- to give better user experience , create Encryption as “E” and Decryption “D” and leave it to user upon his choice
- based on user’s selection encrypt or decrypt the message

```
if mode.upper()=='E':
    newmesg=''.join([encryptDict[letter] for letter in msg.lower()])
elif mode.upper()=='D':
    newmesg=''.join([decryptDict[letter] for letter in msg.lower()])
```



### 3. RESULTS AND DISCUSSION

#### Encryption:



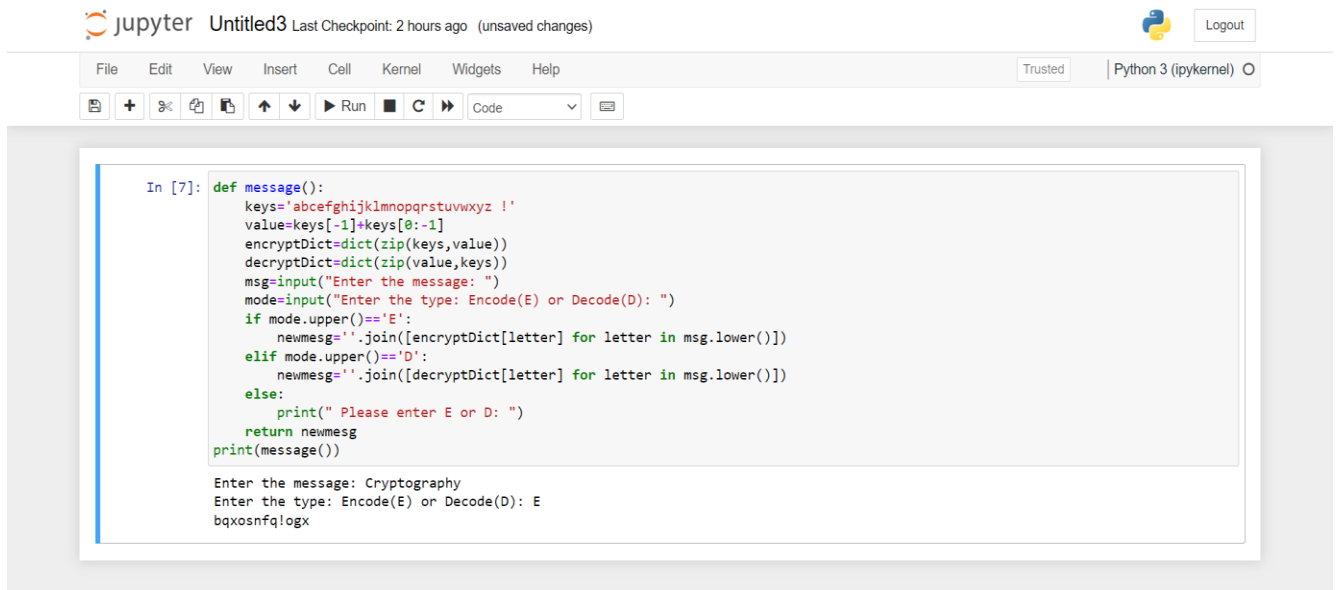
The image shows a Jupyter Notebook interface with the title 'Untitled3' and a status bar indicating 'Last Checkpoint: 2 hours ago (unsaved changes)'. The notebook is running Python 3 (ipykernel). The code in the cell is as follows:

```
In [*]: def message():
        keys='abcdefghijklmnopqrstuvwxyz !'
        value=keys[-1]+keys[0:-1]
        encryptDict=dict(zip(keys,value))
        decryptDict=dict(zip(value,keys))
        msg=input("Enter the message: ")
        mode=input("Enter the type: Encode(E) or Decode(D): ")
        if mode.upper()=='E':
            newmesg=''.join([encryptDict[letter] for letter in msg.lower()])
        elif mode.upper()=='D':
            newmesg=''.join([decryptDict[letter] for letter in msg.lower()])
        else:
            print(" Please enter E or D: ")
        return newmesg
    print(message())
```

Below the code, the prompt 'Enter the message:' is displayed, followed by an empty text input field.

**Fig no 3.1: Output of the code which takes users input**

Run the code , and it displays “Enter the message” which takes user input .



The image shows the same Jupyter Notebook interface as Fig no 3.1, but with the code executed. The output of the code is displayed below the code cell:

```
In [7]: def message():
        keys='abcdefghijklmnopqrstuvwxyz !'
        value=keys[-1]+keys[0:-1]
        encryptDict=dict(zip(keys,value))
        decryptDict=dict(zip(value,keys))
        msg=input("Enter the message: ")
        mode=input("Enter the type: Encode(E) or Decode(D): ")
        if mode.upper()=='E':
            newmesg=''.join([encryptDict[letter] for letter in msg.lower()])
        elif mode.upper()=='D':
            newmesg=''.join([decryptDict[letter] for letter in msg.lower()])
        else:
            print(" Please enter E or D: ")
        return newmesg
    print(message())

Enter the message: Cryptography
Enter the type: Encode(E) or Decode(D): E
bqxosnfqlgix
```

**Fig no 3.2: Encryption of the word ‘cryptography’**

As soon as the user enters the message ( let’s say ‘cryptography’) it gives two options Encryption(E) or Decryption(D) to the user . If user selects the ‘E’ , then it encrypts the

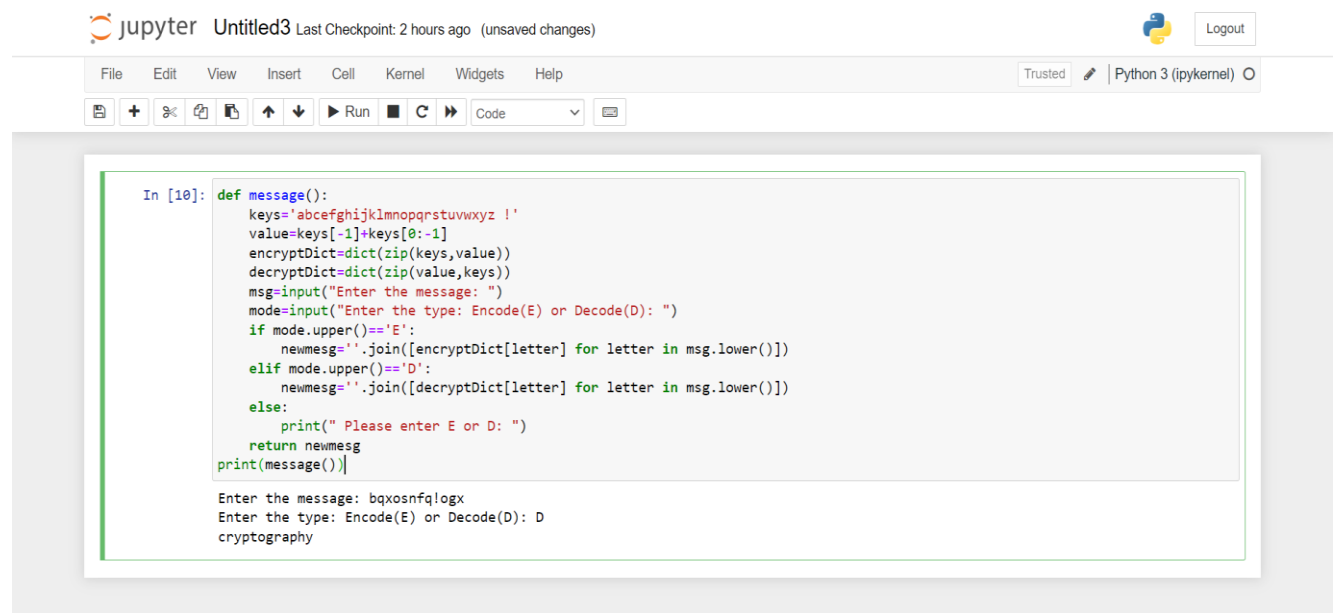
message based on key value which is [-1] i.e it maps the alphabets preceding alphabet and if alphabet is 'a', it maps to '!' .

Thus from the above figure, we can clearly see that message 'Cryptography' → 'BQXOSNFG!OGX' as follows:

$c \rightarrow B$  ,  $r \rightarrow Q$  ,  $y \rightarrow X$  ,  $p \rightarrow O$  ,  $t \rightarrow S$  ,  $o \rightarrow N$  ,  $g \rightarrow F$  ,  $r \rightarrow G$  ,  $a \rightarrow !$  ,  $p \rightarrow O$  ,  $h \rightarrow G$  ,  $y \rightarrow X$

## Decryption:

In decryption , the cipher text alphabets will shift to its next alphabet [:+1] and '!' will be written as 'a'.



```

In [10]: def message():
          keys='abcdefghijklmnopqrstuvwxyz !'
          value=keys[-1]+keys[0:-1]
          encryptDict=dict(zip(keys,value))
          decryptDict=dict(zip(value,keys))
          msg=input("Enter the message: ")
          mode=input("Enter the type: Encode(E) or Decode(D): ")
          if mode.upper()=='E':
              newmsg=''.join([encryptDict[letter] for letter in msg.lower()])
          elif mode.upper()=='D':
              newmsg=''.join([decryptDict[letter] for letter in msg.lower()])
          else:
              print(" Please enter E or D: ")
          return newmsg
          print(message())

Enter the message: bqxsfnf!ogx
Enter the type: Encode(E) or Decode(D): D
cryptography
  
```

**Fig no 3.3 : Decryption of word 'cryptography'**

From the above figure we can clearly see that cipher text 'BQXOSNFG!OGX' → 'Cryptography' as follows:

$B \rightarrow c$  ,  $Q \rightarrow r$  ,  $X \rightarrow y$  ,  $O \rightarrow p$  ,  $S \rightarrow t$  ,  $N \rightarrow o$  ,  $F \rightarrow g$  ,  $G \rightarrow r$  ,  $! \rightarrow a$  ,  $O \rightarrow p$  ,  $G \rightarrow h$  ,  $X \rightarrow y$

#### **4. CONCLUSION AND FUTURE ENHANCEMENT**

So , in this way we can encrypt and decrypt the message by using certain key values using monoalphabetic algorithm . Also we can improve the security or hardness of the code/algorithm by changing key values or using complexity crypto algorithms.

Additional features we may include such as simultaneous encryption and decryption, also wide variety of algorithms have been produced until today. We can use asymmetric-key and symmetric-key algorithms such as Blowfish, AES, RSA, ElGamal which can be implemented and compared to this.

## REFERENCES

1. <https://www.tutorialspoint.com/what-is-monoalphabetic-cipher-in-information-security>
2. <https://russell.ballestrini.net/monoalphabetic-cipher-and-inverse-written-in-python/>
3. <https://crypto.interactive-maths.com/monoalphabetic-substitution-ciphers.html>
4. <https://www.techtarget.com/searchsecurity/definition/cryptography>

## APPENDIX

### CODE:

```
def message():
    keys='abcefghijklmnopqrstuvwxyz !'
    value=keys[-1]+keys[0:-1]
    encryptDict=dict(zip(keys,value))
    decryptDict=dict(zip(value,keys))
    msg=input("Enter the message: ")
    mode=input("Enter the type: Encode(E) or Decode(D): ")
    if mode.upper()=='E':
        newmesg="".join([encryptDict[letter] for letter in msg.lower()])
    elif mode.upper()=='D':
        newmesg="".join([decryptDict[letter] for letter in msg.lower()])
    else:
        print(" Please enter E or D: ")
    return newmesg
print(message())
```