Gurukul International Multidisciplinary Research Journal (GIMRJ)*with* International Impact Factor 8.249
Peer Reviewed Journal
https://doi.org/10.69758/GIMRJ2406IIV12P006

e-ISSN No. 2394-8426
June-2024
Issue–II, Volume–XII

Key-Plain Mapping Algorithm: A Symmetric Key Cryptographic Technique for Secure Communication

Sutanuka Bhowmik
Department of Electronics and Communications Engineering
Year 3
Future Institute of Engineering and Management Sonarpur, Kolkata, India, PIN 700150
Email-sutanuka.bhowmik04@gmail.com
Ph-6291231802

Dr. Anirban Mandal
Department of Electronics and Communications Engineering
Associate Professor
Future Institute of Engineering and Management
Sonarpur, Kolkata, India, PIN 700150
Email- anirban.mandal@teamfuture.in
Ph-9064900298

Received on: *16 May ,2024* ***Revised on:*** *20 June ,2024* ***Published on:*** *30 June ,2024*

Abstract:- In the realm of cryptography, ensuring secure communication is paramount. Symmetric key cryptographic techniques offer a robust means of encryption and decryption. The Key-Plain Mapping Algorithm presents a novel approach to symmetric key cryptography, utilizing a fixed keyword and systematic mapping between plaintext and ciphertext alphabets. This thesis delves into the principles, operations, and security aspects of the Key-Plain Mapping Algorithm, demonstrating its efficacy in securing digital communication.

Keywords:- plain text; cipher text or encrypted message; encryption; decryption; symmetric key cryptography; decrypted message; keyword

1. INTRODUCTION

In the modern digital landscape, ensuring the security of sensitive information is crucial. Cryptography, the science of secure communication, offers various techniques for encryption and decryption. Among these, symmetric key cryptography, where both parties share a secret key, is widely used. The Key-Plain Mapping Algorithm is a novel symmetric key cryptographic technique that employs a fixed keyword and systematic mapping for encryption and decryption.This thesis explores the Key-Plain Mapping Algorithm's principles, mechanics, and cryptographic properties. It

begins by establishing a fixed keyword, which forms the basis of encryption and decryption. Two sets of alphabets are then created: the plain text alphabets (A to Z) and the cipher text alphabets, which include a permutation of the keyword followed by the remaining letters of the

alphabet. This systematic arrangement ensures a one-to-one correspondence between plaintext and ciphertext characters.In summary, this thesis aims to uncover the intricacies of the Key-Plain Mapping Algorithm, highlighting its potential as a symmetric key cryptographic technique for secure communication.Comparative analyses with established techniques provide insights into the algorithm's strengths and limitations. Through theoretical exploration, practical demonstrations, and security analyses, this research contributes to the advancement of cryptographic knowledge and enhances digital communication security.

2. THE CONCEPT

The Key-Plain Mapping Algorithm is a symmetric key cryptographic technique used for secure communication. It operates by utilizing a fixed keyword and establishing a mapping relation between plain text alphabets and cipher text alphabets. This technique ensures that each letter in the plain text alphabet has a corresponding letter in the cipher text alphabet, facilitating encryption and decryption processes.

The algorithm begins with the selection of a constant fixed keyword, which serves as the basis for both encryption and decryption. Next, two sets of alphabets are constructed: the plain text alphabets, consisting of the standard English alphabet from A to Z, and the cipher text alphabets. The cipher text alphabets are derived from the keyword, maintaining the original sequence of letters without repetition. The remaining letters of the alphabet are then appended to complete the cipher text alphabets.

Once the sets of alphabets are established, a mapping relation is established between the corresponding elements (letters) of each set. Each letter in the plain text alphabet is paired with its corresponding letter in the cipher text alphabet, forming the basis for encryption and decryption.For encryption, each letter in the user input plain text is replaced by its corresponding mapped letter in the cipher text alphabet set. This process generates the cipher text, effectively encrypting the user input plain text. Conversely, decryption involves replacing each letter in the user input cipher text with its corresponding plain text alphabet, utilizing the same key and mapping table. This decrypts the cipher text and retrieves the original plain message.

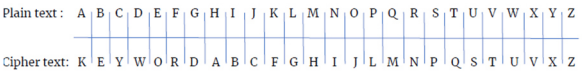
3. THE ALGORITHM

- Select a fixed keyword for both encryption and decryption. For example, consider the keyword "KEYWORD".
- Construct the plain text alphabets set, comprising the standard English alphabet from A to Z,

  Plain text alphabets:-A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z

- Derive the cipher text alphabets set from the keyword. This involves arranging the letters of the keyword in their original sequence without repetition of any letter, followed by appending the remaining letters of the alphabet.

  Cipher text alphabets:-
  K,E,Y,W,O,R,D,A,B,C,F,G,H,I,J,L,M,N,P,Q,S,T,U,V,X,Z
- Establish a mapping relation between the corresponding elements (letters) of the plain text and cipher text alphabets sets.
- Visualize or write both sets vertically besideeach other or horizontally one below another for mapping.



- Encryption:
  Take the user input plain text.
  Replace each letter in the plain text with its corresponding mapped letter in the cipher text alphabet set.
  Generate the cipher text, which represents the encrypted message.
- Decryption:
  Take the user input cipher text.
  Replace each letter in the cipher text with its corresponding plain text alphabet, utilizing the same key and mapping table.
  PROGRAM IN PYTHON:

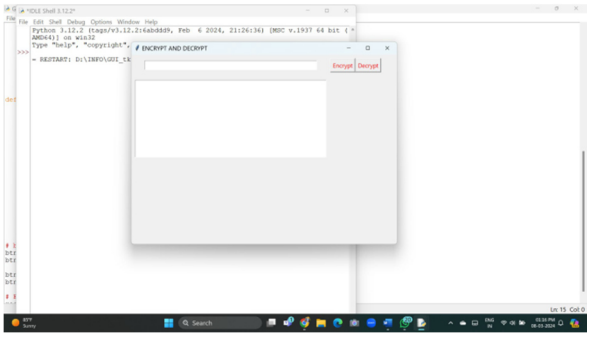
```
from tkinter import *
root = Tk()
root.title("ENCRYPT AND DECRYPT")
# adding a label to the root window
root.geometry('700x500')
txt = Entry(root, width=50)
txt.grid(column=300, row=50, padx=10, pady=10, columnspan=3)
# function to display text when
# button is clicked
plain_alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
keyword = "KEYWORD"
cipher = "KEYWORDABCFGHIJLMNPQSTUVXZ"
s = StringVar()
# create a Text widget to display the result
result_text = Text(root, height=10, width=50)
result_text.grid(column=300, row=90, padx=10, pady=10, columnspan=3)
def Input():
        S = txt.get()
        return S
def en_clicked():
        s = Input()
        l = len(s)
        E = ""
```

```
    for i in range(l):
        character = s[i]
        z=ord(character)
        if  z>=97 and z<=122:
            z=z-32
            character=chr(z)
        if character in plain_alpha:
            index = plain_alpha.find(character)
            c = cipher[index]
            E = E + c
          else:
            E = E + character
    # insert the encrypted text into the Text widget
    result_text.delete(1.0, END) # clear previous content
result_text.insert(END, "Encrypted: " + E)
def de_clicked():
    d = Input()
    l = len(d)
    D = ""
    for i in range(l):
        character = d[i]
        z=ord(character)
        if z>=97 and z<=122:
            z=z-32
            character=chr(z)
        if character in cipher:
            index = cipher.find(character)
            e = plain_alpha[index]
            D = D + e
        else:
            D = D + character
    result_text.delete(1.0, END) # clear previous content
```

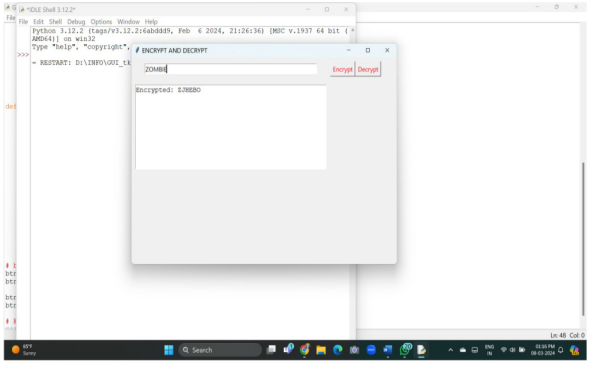
```
        result_text.insert(END, "Decrypted: " + D)
# button widgets with red color text
btn1 = Button(root, text="Encrypt", fg="red", command=en_clicked)
btn1.grid(column=310, row=50, pady=10)
btn2 = Button(root, text="Decrypt", fg="red", command=de_clicked)
btn2.grid(column=320, row=50, pady=10)
# Execute Tkinter
        root.mainloop()
```

RUNNING THE PROGRAM:

On executing the program a graphical user interface terminal is created by the tkinter:



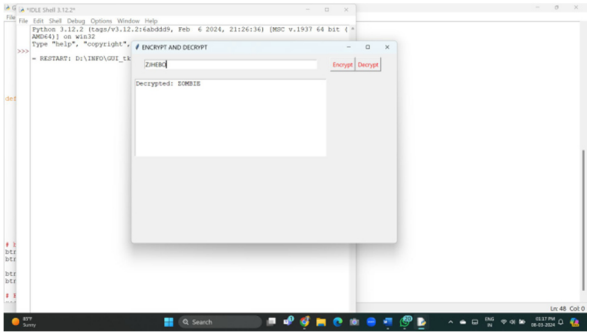
Test Case 1 of Encryption:
Input: plain text = “ZOMBIE”
Output: cipher text = “ZJHEBO”



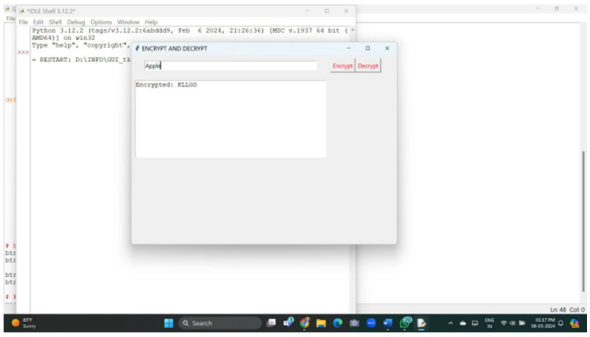
Test Case 1 of Decryption:
Input: cipher text = “ZJHEBO”
Output: cipher text = “ZOMBIE”

Gurukul International Multidisciplinary Research Journal (GIMRJ)*with* International Impact Factor 8.249
Peer Reviewed Journal
https://doi.org/10.69758/GIMRJ2406IIV12P006

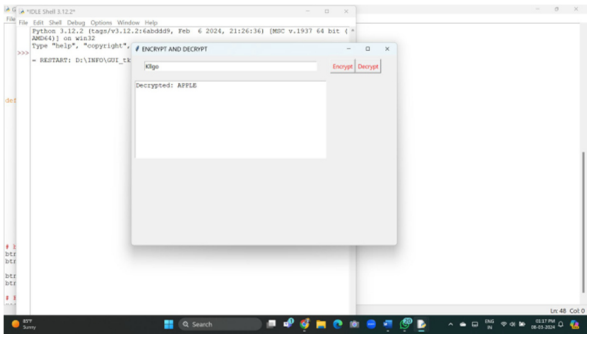
e-ISSN No. 2394-8426
June-2024
Issue–II, Volume–XII



Test Case 2 of Encryption:
Input: plain text = “Apple”
Output: cipher text = “KLLGO”



Test Case 2 of Decryption:
Input: plain text = “Kllgo”
Output: cipher text = “APPLE”



4.CONCLUSION

Using Python3 IDLE and the Tkinter module of Python the Graphical User Interface is created to encrypt and decrypt messages. Cryptography plays a vital role in securing communication, data, and transactions in the digital world. It provides methods for encoding information in such a way that only authorized parties can access it, thus ensuring confidentiality, integrity, authentication, and non-repudiation. As technology advances, the importance of cryptography continues to grow, especially with the rise of digital currencies, cloud computing, and Internet of Things (IoT) devices.

In conclusion, cryptography serves as a cornerstone of modern cybersecurity, enabling secure communication and transactions over insecure channels. It empowers individuals, organizations, and governments to safeguard sensitive information, preserve privacy, and maintain trust in digital interactions. As threats evolve, ongoing research and innovation in cryptography are crucial to stay ahead of adversaries and uphold the security and privacy of digital systems.

7. REFERENCES

[1] Behrouz Forouzan "Data Communications and Networking", Tata McGraw Hill, 2001
[2] Bruce Schneier, " Applied Cryptography", Wiley India, 1993