



CryptoCoder: A Polyalphabetic Substitution Encryption tool

1. Introduction

Our crypto-coder is a powerful encryption built with Python and Tkinter framework GUI. An application that utilizes a sophisticated polyalphabetic substitution method to encode messages securely. With this CryptoCoder, it can protect your sensitive information or details by replacing characters in your plaintext with carefully chosen characters from multiple alphabets. Hence, our intuitive graphical interface streamlines the encryption processes, allowing it to input text and customize the encryption parameters effortlessly in a user-friendly environment including the functions of encrypting, decrypting, txt.file insertion, can save an encrypted & decrypted file and resetting the text

2. Features:

2.1 Method: POLYALPHABETIC SUBSTITUTION

- Our cryptocoder utilizes a polyalphabetic substitution method, a cryptographic technique that replaces characters in the plaintext with a diverse character from a multiple letter. This approach significantly strengthens the encryption process, making it more resistant to a various decryption attack

2.2 Framework: TKINTER GUI

- The inclusion of a tkinter-based GUI in CryptoCoder offers a seamless and intuitive interface. Through GUI, users can easily access and input their plaintext, customize encryption parameters, and will visualize the encoded output, facilitating smooth and user-friendly encryption experiences.

2.3 Service: DATA PROTECTION:

- With this CryptoCoder, users now can confidently safeguard their sensitive information by leveraging the power of polyalphabetic substitution, this tool ensures that encoded messages are simple but highly resistant to unauthorized access, providing a robust layer of security and with a symmetric algorithm that carries a private key to obfuscate the attackers.

2.4 **Algorithm:** SYMMETRIC ALGORITHM ENCRYPTION

- In addition to the polyalphabetic substitution method, this cryptographic incorporates a powerful symmetric algorithm that ensures both encryption and decryption processes utilizing the same key, providing a high level of confidentiality and data integrity.

3. How to use Polyalphabetic Substitution CryptoCoder

1 **Launch the CryptoCoder application**

- a. Open the Tkinter GUI.

2 **Encryption:**

- a. Input the plaintext message into the provided text input field.
- b. Input the key "1234" to operate the encryption and decryption process
- c. Customize encryption parameters, such as the number of alphabets and their arrangement, through the GUI options.
- d. Click the "Encrypt" button to initiate the encryption process.
- e. The encoded ciphertext will be displayed in the output field.
- f. The encrypted ciphertext can save the text file on your desktop, and users may/can decrypt the saved encrypted file message after inserting it in the GUI CryptoCoder which will display the result.

3 **Decryption:**

- a. Input the encoded ciphertext into the text input field.
- b. Ensure that the encryption parameters and symmetric key used during encryption are correctly applied.
- c. Click the "Decrypt" button to begin the decryption process.
- d. The decrypted plaintext will be displayed in the output field.
- e. The decrypted ciphertext can save the text file on your desktop, and users may/can encrypt the saved decrypted file message after inserting it in the GUI CryptoCoder which will display the result.

4 **File Insertion:**

- a. Locate the "Open File" option in the GUI.
- b. Click on the option to select a file from your system.

- c. Choose the desired encryption parameters and a symmetric key. iv. Click the "Encrypt" button to encrypt the contents of the file.
- d. The encrypted file will be saved in a specified location.
- e. Vice versa when saving a decrypted file - this will also save in a specified location.

5 File Reset:

- a. If you desire to clear your text input and output fields, click the "Reset" Button.
- b. The input and output fields will be cleared, allowing it to start the process with a fresh message.

4. Code:

```

1  from tkinter import *
2  from tkinter import messagebox
3  from tkinter import filedialog
4
5
6  1 usage
7  def decrypt():
8      password = code.get()
9
10     if password == "1234":
11         screen2 = Toplevel(screen)
12         screen2.title("Decryption")
13         screen2.geometry("400x200")
14         screen2.configure(bg="#E9A178")
15
16         message = text1.get(1.0, END)
17         decrypted_message = polyalphabetic_decrypt(message)
18
19         Label(screen2, text="DECRYPT", font="arial", fg="white", bg="#E9A178").place(x=10, y=0)
20         text2 = Text(screen2, font="Rpbote 10", bg="white", relief=GROOVE, wrap=WORD, bd=0)
21         text2.place(x=10, y=40, width=380, height=150)
22
23         text2.insert(END, decrypted_message)
24
25     elif password == "":
26         messagebox.showerror("Decryption", "Input Password")
27
28     elif password != "1234":
29         messagebox.showerror("Decryption", "Invalid Password")
30
31 polyalphabetic_decrypt()

```

```

23
24     elif password == "":
25         messagebox.showerror("Decryption", "Input Password")
26
27     elif password != "1234":
28         messagebox.showerror("Decryption", "Invalid Password")
29
30     def save_file():
31         file_path = filedialog.asksaveasfilename(defaulttextextension=".txt", filetypes=[("Text Files", "*.txt")])
32         if file_path:
33             with open(file_path, "w") as f:
34                 f.write(text2.get("1.0", "end-1c"))
35
36     Button(text="SAVE DECRYPTED FILE", height="2", width=50, bg="#E9A178", fg="white", bd=0, command=save_file).place(
37         x=10, y=400)
38
39
40 1 usage
41 def encrypt():
42     password = code.get()
43
44     if password == "1234":
45         screen1 = Toplevel(screen)
46         screen1.title("Encryption")
47         screen1.geometry("400x200")
48         screen1.configure(bg="#820000")
49
50         message = text1.get(1.0, END)
51         encrypted_message = polyalphabetic_encrypt(message)
52
53     decrypt()

```

```

45     screen1.title("Encryption")
46     screen1.geometry("400x200")
47     screen1.configure(bg="#820000")
48
49     message = text1.get(1.0, END)
50     encrypted_message = polyalphabetic_encrypt(message)
51
52     Label(screen1, text="ENCRYPT", font="arial", fg="white", bg="#820000").place(x=10, y=0)
53     text2 = Text(screen1, font="Rpbote 10", bg="white", relief=GR00VE, wrap=WORD, bd=0)
54     text2.place(x=10, y=40, width=380, height=150)
55
56     text2.insert(END, encrypted_message)
57
58     elif password == "":
59         messagebox.showerror("Encryption", "Input Password")
60
61     elif password != "1234":
62         messagebox.showerror("Encryption", "Invalid Password")
63
64     def save_file():
65         file_path = filedialog.asksaveasfilename(defaulttextextension=".txt", filetypes=[("Text Files", "*.txt")])
66         if file_path:
67             with open(file_path, "w") as f:
68                 f.write(text2.get("1.0", "end-1c"))
69
70     Button(text="SAVE ENCRYPTED FILE", height="2", width=50, bg="#820000", fg="white", bd=0, command=save_file).place(
71         x=10, y=400)
72
73
74     decrypt()

```

```

74 def polyalphabetic_decrypt(ciphertext):
75     key = code.get()
76     key_length = len(key)
77     plaintext = ""
78
79     for i in range(len(ciphertext)):
80         char = ciphertext[i]
81         if char.isalpha():
82             if char.islower():
83                 decrypted_char = chr(
84                     (ord(char) - ord('a') - (ord(key[i % key_length].lower()) - ord('a')) % 26 + ord('a'))
85                 )
86             else:
87                 decrypted_char = chr(
88                     (ord(char) - ord('A') - (ord(key[i % key_length].upper()) - ord('A')) % 26 + ord('A'))
89                 )
90             plaintext += decrypted_char
91         else:
92             plaintext += char
93     return plaintext
94
95 1 usage
96 def polyalphabetic_encrypt(plaintext):
97     key = code.get()
98     key_length = len(key)
99     ciphertext = ""
100     for i in range(len(plaintext)):
101         char = plaintext[i]
102         if char.isalpha():

```

polyalphabetic_decrypt() → for i in range(len(ciphertext))

```

94 def polyalphabetic_encrypt(plaintext):
95     key = code.get()
96     key_length = len(key)
97     ciphertext = ""
98     for i in range(len(plaintext)):
99         char = plaintext[i]
100         if char.isalpha():
101             if char.islower():
102                 encrypted_char = chr(
103                     (ord(char) - ord('a') + (ord(key[i % key_length].lower()) - ord('a')) % 26 + ord('a'))
104                 )
105             else:
106                 encrypted_char = chr(
107                     (ord(char) - ord('A') + (ord(key[i % key_length].upper()) - ord('A')) % 26 + ord('A'))
108                 )
109             ciphertext += encrypted_char
110         else:
111             ciphertext += char
112     return ciphertext
113
114 1 usage
115 def main_screen():
116     global screen
117     global code
118     global text1

```

! usage

```
114 def main_screen():
115     global screen
116     global code
117     global text1
118
119     screen = Tk()
120     screen.geometry("375x470")
121     screen.title("CryptoCoder")
122
123     def load_file():
124         file_path = filedialog.askopenfilename(filetypes=[("Text Files", "*.txt")])
125         if file_path:
126             with open(file_path, "r") as f:
127                 text1.delete("1.0", "end")
128                 text1.insert("1.0", f.read())
129
130     def reset():
131         code.set("")
132         text1.delete(1.0, END)
133
134     Label(text="Enter text for encryption and decryption", fg="black", font=("calibri", 13)).place(x=10, y=10)
135     text1 = Text(font="Roboto 20", bg="white", relief=GRROOVE, wrap=WORD, bd=0)
136     text1.place(x=10, y=50, width=355, height=100)
137
138     Label(text="Enter secret key for encryption and decryption", fg="black", font=("calibri", 13)).place(x=10, y=170)
139     code = StringVar()
140     Entry(textvariable=code, width=19, bd=0, font=("arial", 25), show="*").place(x=10, y=200)
```

```
137
138     Label(text="Enter secret key for encryption and decryption", fg="black", font=("calibri", 13)).place(x=10, y=170)
139     code = StringVar()
140     Entry(textvariable=code, width=19, bd=0, font=("arial", 25), show="*").place(x=10, y=200)
141
142     Label(text="Created by: Justine Lenard Sabawil of G35", fg="black", font=("calibri", 10)).place(x=10, y=445)
143
144     Button(text="ENCRYPT", height="2", width=23, bg="#820000", fg="white", bd=0, command=encrypt).place(x=10, y=250)
145     Button(text="DECRYPT", height="2", width=23, bg="#E9A178", fg="white", bd=0, command=decrypt).place(x=200, y=250)
146     Button(text="RESET", height="2", width=50, bg="#4E6C50", fg="white", bd=0, command=reset).place(x=10, y=300)
147     Button(text="OPEN FILE", height="2", width=50, bg="#4E6C50", fg="white", bd=0, command=load_file).place(x=10, y=350)
148
149     screen.mainloop()
150
151
152 main_screen()
153
```


OUTPUT:

MAIN INTERFACE



The image shows a screenshot of a web application titled "CryptoCoder". The interface is clean and modern, with a light gray background. At the top, there is a header bar with the application name and standard window controls (minimize, maximize, close). Below the header, the main content area is divided into sections. The first section is labeled "Enter text for encryption and decryption" and contains a large, empty white text input field. The second section is labeled "Enter secret key for encryption and decryption" and contains a smaller, empty white text input field. Below these input fields, there are four buttons: "ENCRYPT" (dark red), "DECRYPT" (orange), "RESET" (dark green), and "OPEN FILE" (dark green). At the bottom of the interface, there is a footer section that reads "Created by: Justine Lenard Sabawil of G35".

CryptoCoder

Enter text for encryption and decryption

Enter secret key for encryption and decryption

ENCRYPT

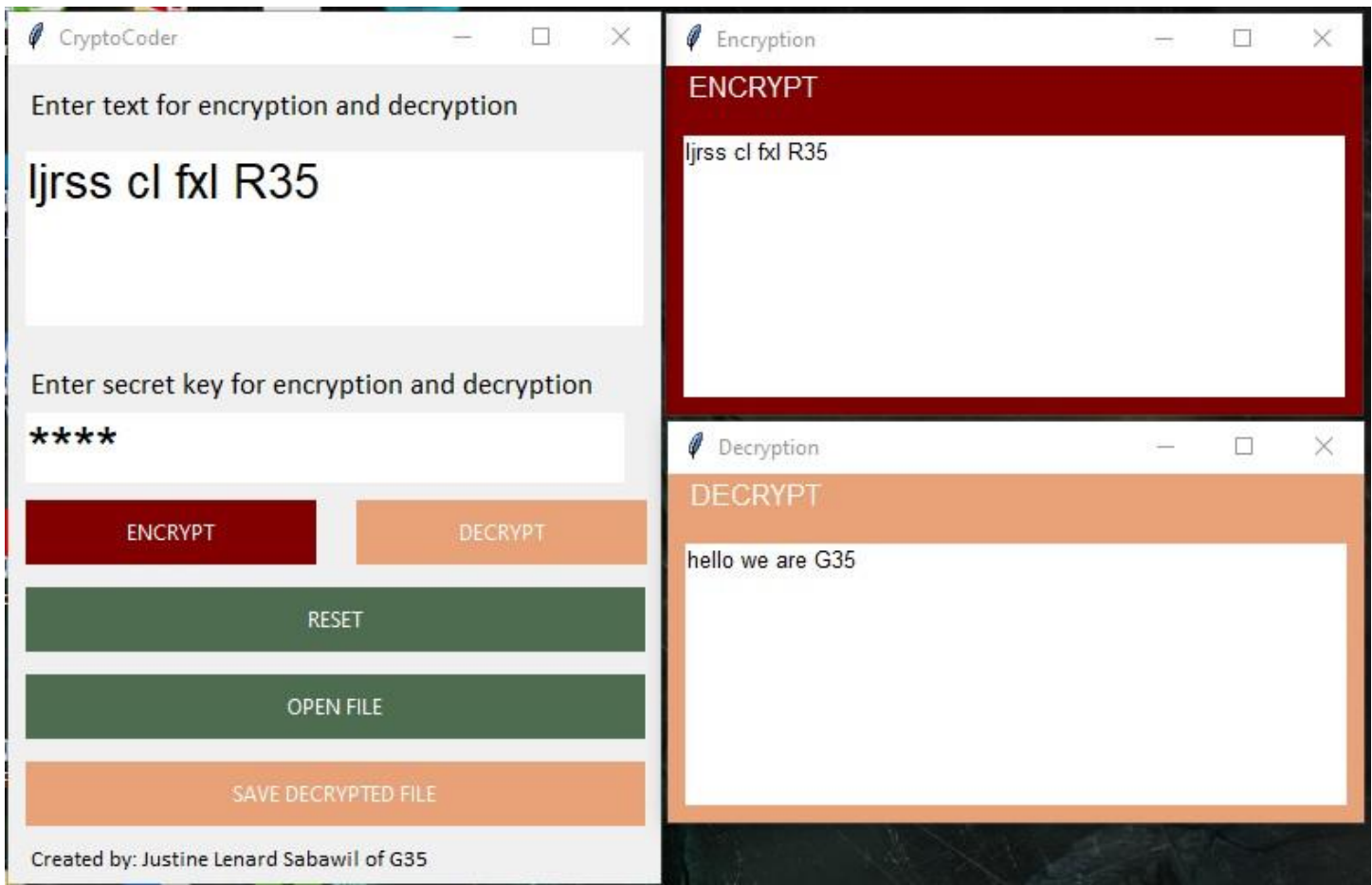
DECRYPT

RESET

OPEN FILE

Created by: Justine Lenard Sabawil of G35

OUTPUT INTERFACE/ RESULT



5. Members:

Justine Lenard Sabawil - Developer
Mae Escurel - documentary
Dona Balaoro
Francis Jhonrel Gilos
Alyanna Mae Garbida

