République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Ahmed Draia - Adrar

Faculté des Sciences et de la Technologie

Département des Mathématiques et Informatique

Vigenère Cryptosystem

PROJECT DOCUMENTAION REPORT

SUBMITTED BY:

Rabhi Zinelaabidine

Framed by:

Omari Mohammed

2020-2021

NOTICE

In the explanation, I only use Google Translate. Sorry if there are Mistakes … Note that this phrase is also translated

1.1 Introduction

Vigenère Cipher:

Vigenère Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets .The encryption of the original text is done using the *Vigenère square or Vigenère table*.

- The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers.
- At different points in the encryption process, the cipher uses a different alphabet from one of the rows.
- The alphabet used at each point depends on a repeating keyword.

Example:

Input: Plaintext: GEEKSFORGEEKS

Keyword: AYUSH

Output: Ciphertext: GCYCZFMLYLEIM

For generating key, the given keyword is repeated

in a circular manner until it matches the length of

the plain text.

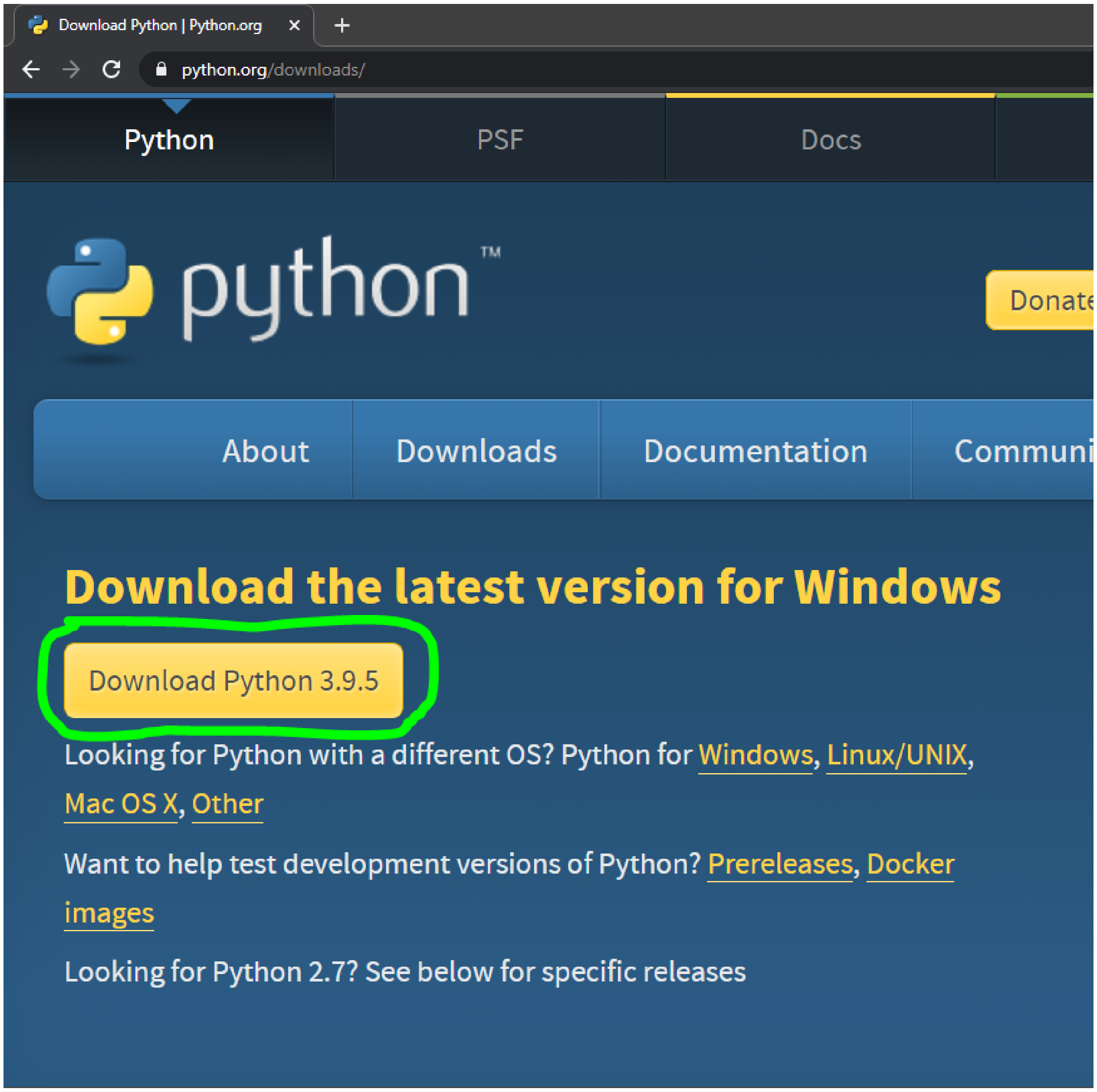
The keyword "AYUSH" generates the key "AYUSHAYUSHAYU"

The plain text is then encrypted using the process explained below.

2 installation of tools used

2.1. Download and install Python:

First We enter the following link https://www.python.org/downloads/ Click on download if your Windows operating system from version 8 and above because the next version of Python does not support Windows 7 If you have Windows 7 download less version of version 3 for python When the download is complete, we install it.



Download the latest version for Windows

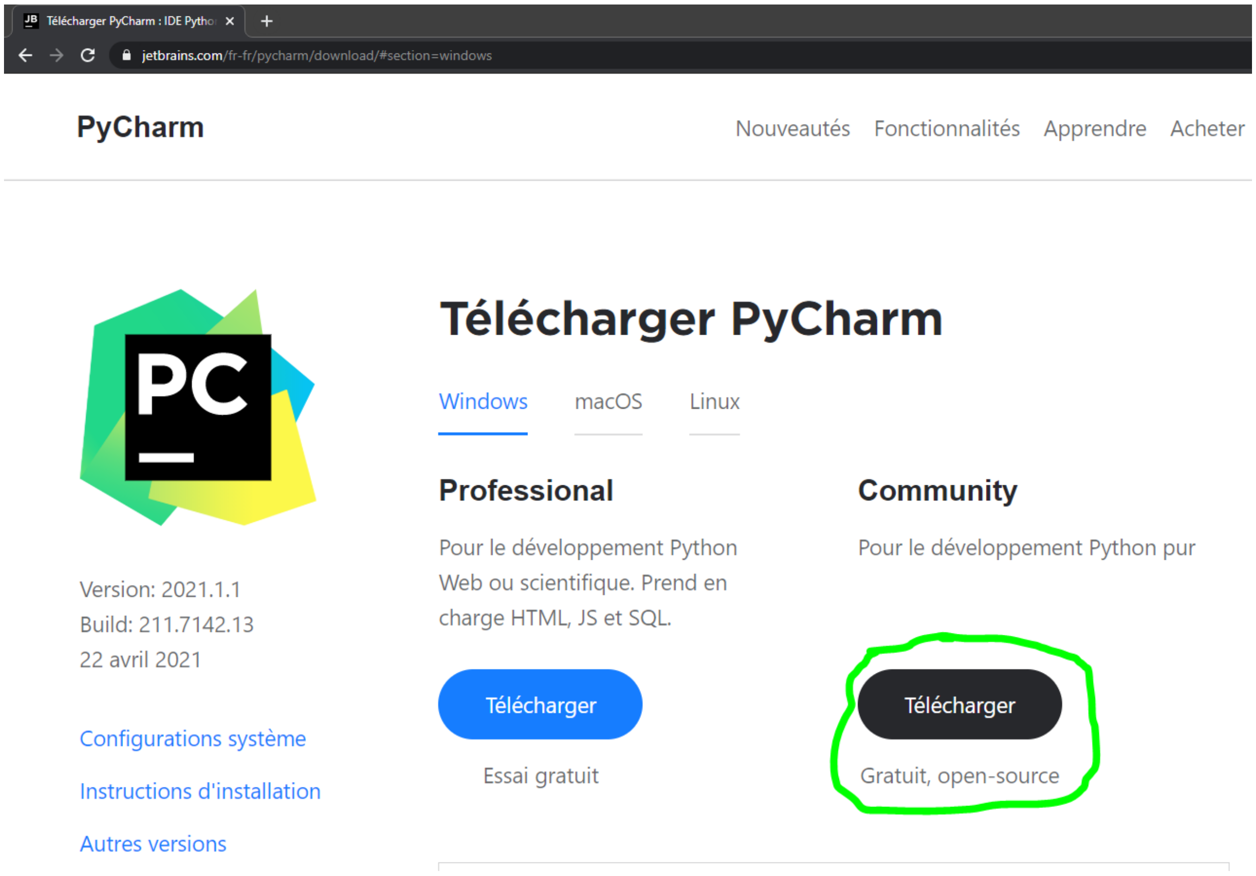
Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

2.2. Download and install PyCharm Community:

First We enter the following link https://www.jetbrains.com/frfr/pycharm/download/#section=windows Click on download When the download is complete, we install it.



Télécharger PyCharm

Version: 2021.1.1
Build: 211.7142.13
22 avril 2021

Professional

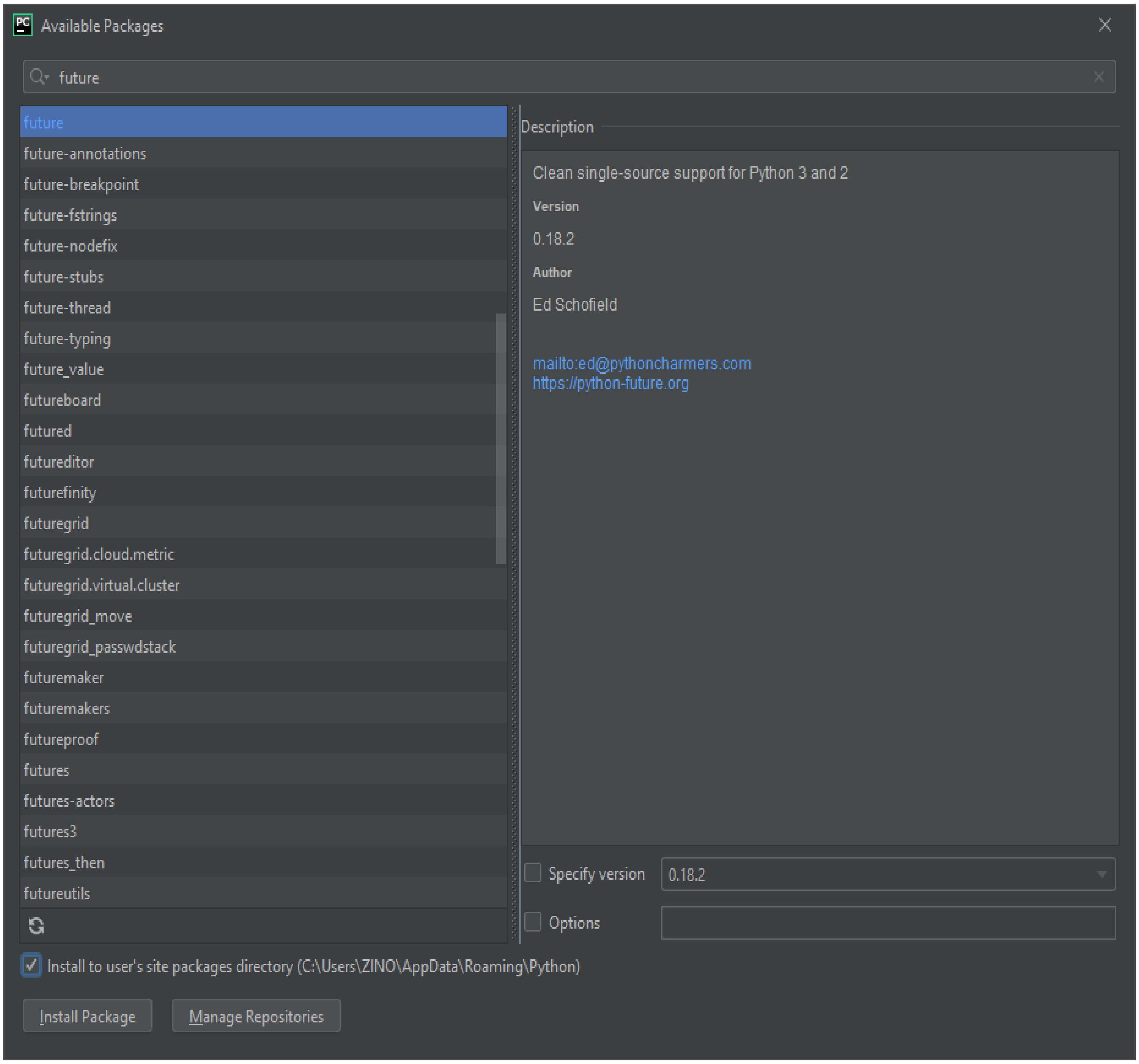
Pour le développement Python Web ou scientifique. Prend en charge HTML, JS et SQL.

Community

Pour le développement Python pur

2.3. Add Package Tkinter in PyCharm Community:

We go into Baicharm clicking on buttons alt+ctrl+s show us a window we choose a project and then a interpreter and click on icon **+** We write in the search box **future** and install package


Available Packages

future

future
future-annotations
future-breakpoint
future-fstrings
future-nodefix
future-stubs
future-thread
future-typing
future_value
futureboard
futured
futureditor
futurefinity
futuregrid
futuregrid.cloud.metric
futuregrid.virtual.cluster
futuregrid_move
futuregrid_passwdstack
futuremaker
futuremakers
futureproof
futures
futures-actors
futures3
futures_then
futureutils

Description

Clean single-source support for Python 3 and 2

Version

0.18.2

Author

Ed Schofield

mailto:ed@pythoncharmers.com
https://python-future.org

☐ Specify version 0.18.2

☐ Options

☑ Install to user's site packages directory (C:\Users\ZINO\AppData\Roaming\Python)

1. PROGRAM STRUCTURE

1.1 Global/Local Variables:

Variable name	*Type*
alphabet	String List [a……..z]
P, message	Plain text String
K, key	Keyword string
c	Cipher text string

1.2 Fonction & Procedures:

Fonction & Procedures	**Description**
Encrypt(p, k)	This Fonction Encrypt the plain text with using two variable p(plaintext) K(keyword) In the beginning, we extract the first letter from the word key and call the variable i with the value 0 Now we put a condition if the variable i is equal to the key length, the value of i does not change We extract the order of the first letter from the text and add the key order to it, so we get the first letter we add at the beginning of the c string In the event that the order is above the value of 25, we will decrease the value of 25 from this order.

Decrypt(Key, message)	In the decryption process we know a variable in the form of a list and we put an episode starting from the first code of the encrypted text to the last code within it provided that if the code is a letter belonging to the previously defined series or does not belong if it belongs we subtract its order from the order of the first letter of the key and complete our instructions to the end of the episode and at the end of the episode we return the value of the text without blanks
Take_input(),Take_input1()	In these two functions I enter two variables in each function the first function we enter the key and the text to be encrypted and in the second function we enter the key and the encrypted text to decrypt it and also we enter the result in the text box in both cases when decryption and have encryption and return the vacuum in the two texts as entered in the first
Cryptanalysis(),V_Dictionary(),Is_English() ,GetEnglishCount(),Remove_Non_Aphabets(), AddDictionary()	The function cryptoanalysis is a link between the functions you use. The function calls the function a V_Dictionary() in which we open a dictionary file that contains a lot of potential keys that he uses to decrypt the text starts from the first word when he decrypts it with the first key comes the role of the function is_English() to make sure the ratio of words

	in English to 40 percent if this condition is realized we print the key in the key text box and print the text in the text box There is also a function Remove_Non_Alphabets() that deletes from the encrypted text the symbols And the function GetEnglishCount() that calculates the number of words in English

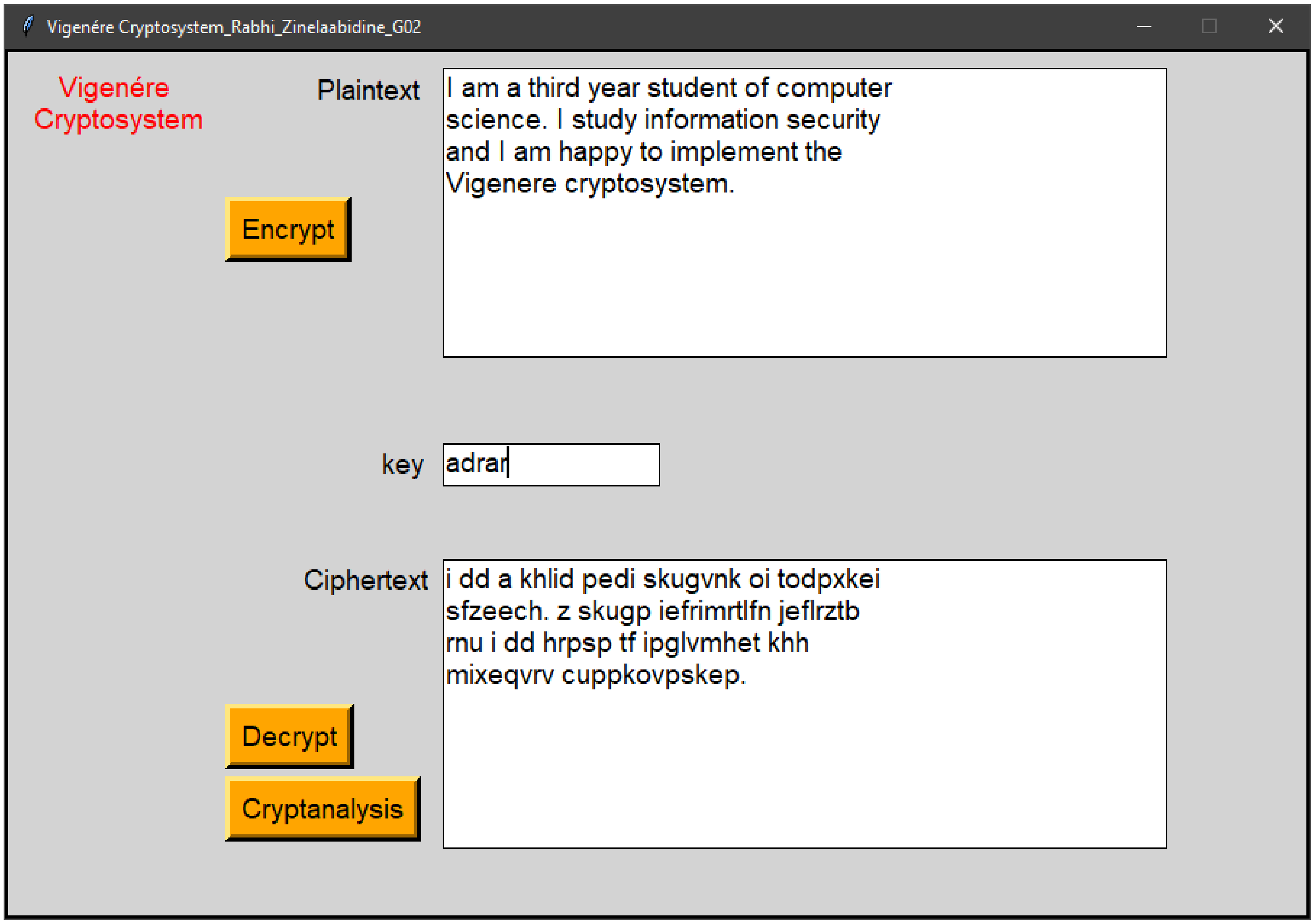
3.3 Source Code

```
# Rabhi_Zinelaabidine_code
from tkinter import *
from tkinter import messagebox
import re

vg = Tk()
vg.geometry("900x600")
vg.title("Vigenére Cryptosystem_Rabhi_Zinelaabidine_G02")
vg.resizable(width="false", height="false")
vg.config(bg="lightgrey",relief='solid',borderwidth=2)
alphabets = "abcdefghijklmnopqrstuvwxyz"

# les Fonction
def Encrypt(p, k):  #The encryption process requires the key and the 
Plaintext
    c = ""
    kpos = []
    k=k.lower()
    for x in k:
        kpos.append(alphabets.find(x))
    i = 0
    for x in p:

            if i == len(kpos):
                i = 0
            if not alphabets.find(x) == -1:   # If it x is found in the list 
alphabet
                pos = alphabets.find(x) + kpos[i]
                if pos > 25:
                    pos = pos - 26

                c += alphabets[pos]
                i += 1
            else:
                c += x
```

```python
    return c

def Decrypt(key, message):

    p = []
    keyIndex = 0
    key = key.lower()

    for symbol in message:
        num = alphabets.find(symbol.lower())
        if num != -1:
            num -= alphabets.find(key[keyIndex])
            num %= len(alphabets)
            if symbol.islower():
                p.append(alphabets[num])
            elif symbol.islower():
                p.append(alphabets[num].lower())

            keyIndex += 1
            if keyIndex == len(key):
                keyIndex = 0
        else:
            p.append(symbol)

    return ''.join(p)


def cryptanalysis():
    ciphertext = entry4.get("1.0", "end-1c")
    fo = open('dictionary.txt')
    words = fo.readlines()
    fo.close()
    for word in words:
        word = word.strip()  # Remove the newline at the end.
        decryptedText = Decrypt(word, ciphertext)
        if Is_English(decryptedText, wordPercentage=40):
            entry2.delete("1.0", "end")
            entry3.delete("1.0", "end")
            entry2.insert(END, decryptedText)
            entry3.insert(END, str(word))


Upperalphabets = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
alphabets_AND_SPACE = Upperalphabets + Upperalphabets.lower() + ' \t\n'

def AddDictionary():
    DictionaryFile = open('dictionary.txt')
    englishWords = {}
    for word in DictionaryFile.read().split('\n'):
        englishWords[word] = None
    DictionaryFile.close()
    return englishWords

ENGLISH_WORDS = AddDictionary()

def GetEnglishCount(message):
    message = message.upper()
    message = Remove_Non_Alphabets(message)
    PossibleWords = message.split()

    if PossibleWords == []:
        return 0.0 # No words at all, so return 0.0.

    matches = 0
    for word in PossibleWords:
        if word in ENGLISH_WORDS:
            matches += 1
    return float(matches) / len(PossibleWords)

def Remove_Non_Alphabets(message):
    AlphabetsOnly = []
    for symbol in message:
        if symbol in alphabets_AND_SPACE:
            AlphabetsOnly.append(symbol)
    return ''.join(AlphabetsOnly)

def Is_English(message, wordPercentage=20, letterPercentage=85):

    wordsMatch = GetEnglishCount(message) * 100 >= wordPercentage
    numalphabets = len(Remove_Non_Alphabets(message))
    messagealphabetsPercentage = float(numalphabets) / len(message) * 100
    alphabetsMatch = messagealphabetsPercentage >= letterPercentage
    return wordsMatch and alphabetsMatch


def Take_input():   # Function that allows me to enter Plaintext and the key
    INPUT = entry2.get("1.0", "end-1c")
    d=INPUT
    INPUT=INPUT.replace(" ","")
    INPUT = INPUT.lower()
    INPUT2 = entry3.get("1.0", "end-1c")
    INPUT2 = INPUT2.replace(" ", "")
    if not INPUT2.isalpha():
            messagebox.showerror('Only alphabets', 'Only alphabets are
allowed in the key!')
    else:
            entry4.delete("1.0", "end")
            c = Encrypt(INPUT, INPUT2)
            indexes = [x.start() for x in re.finditer(' ', d)]     #Add white
spaces in the plaintext
            for i in indexes:
                c = c[:i] + " " + c[i:]
            entry4.insert(END, c)

def Take_input1():   # Function that allows me to enter Ciphertext and the
key
    INPUT = entry4.get("1.0", "end-1c")
    d = INPUT
    INPUT = INPUT.replace(" ", "")
    INPUT = INPUT.lower()
    INPUT2 = entry3.get("1.0", "end-1c")
    INPUT2 = INPUT2.replace(" ", "")
    if not INPUT2.isalpha():
            messagebox.showerror('Only alphabets', 'Only alphabets are
allowed in the key!')
    else:
            entry2.delete("1.0", "end")
            c = Decrypt(INPUT2, INPUT)
            indexes = [x.start() for x in re.finditer(' ', d)]  #Add white 
spaces in the ciphertext
            for i in indexes:
                c = c[:i] + " " + c[i:]

            entry2.insert(END, c)

# Began Interface Graphics
# position label && Text Box
l1 = Label(text="Vigenére\n Cryptosystem", font=("serif", 14), fg="red", 
bg="lightgrey")
l1.pack()
l1.place(x=10, y=10)

l2 = Label(text="Plaintext", font=("serif", 14), bg="lightgrey")
l2.pack()
l2.place(x=210, y=12)
entry2 = Text(vg, font=("serif", 14), relief='solid')
entry2.pack()
entry2.place(width=500, height=200, x=300, y=11)

l3 = Label(text="key", font=("serif", 14), bg="lightgrey")
l3.pack()
l3.place(x=255, y=270)
entry3 = Text(vg, font=("serif", 14), relief='solid')
entry3.pack()
entry3.place(width=150, height=30, x=300, y=270)

l4 = Label(text="Ciphertext", font=("serif", 14), bg="lightgrey")
l4.pack()
l4.place(x=201, y=350)
entry4 = Text(vg, font=("serif", 14), relief='solid')
entry4.pack()
entry4.place(width=500, height=200, x=300, y=350)

# button

b1 = Button(vg, text="Encrypt", font=("serif", 14), bg="orange", 
command=lambda: Take_input(),relief='raised', borderwidth=5)
b1.pack()
b1.place(x=150, y=100)

b2 = Button(vg, text="Decrypt", font=("serif", 14), bg="orange", 
command=lambda: Take_input1(),relief='raised', borderwidth=5)
b2.pack()
b2.place(x=150, y=450)

b3 = Button(vg, text="Cryptanalysis", font=("serif", 14), bg="orange", 
command=lambda: cryptanalysis(), relief='raised', borderwidth=5)
b3.pack()
b3.place(x=150, y=500)
#End Button
# End_Interface Graphics//

vg.mainloop()
```

3.4 Execution of the program:

1 Encryption using key and text :

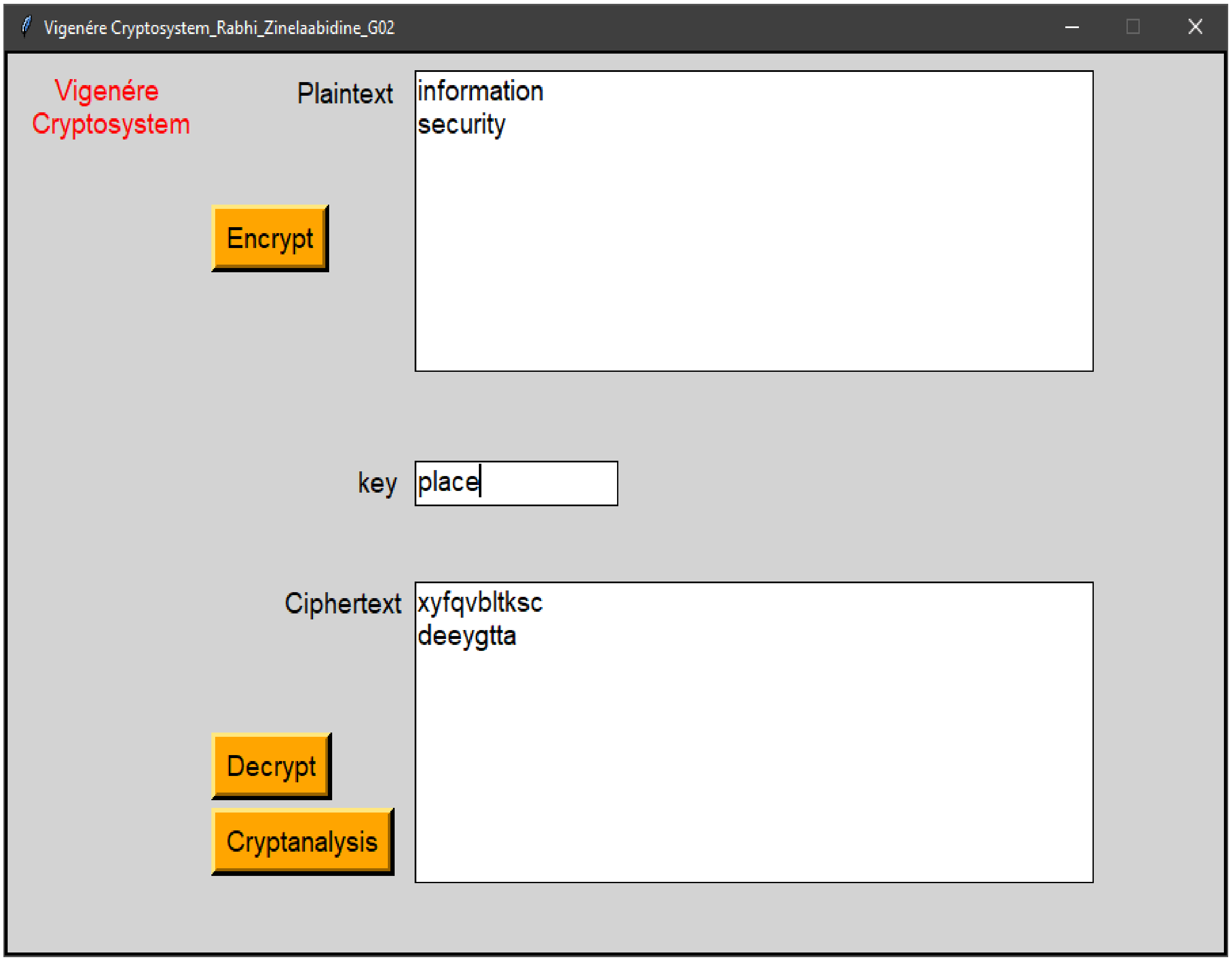


Plaintext: I am a third year student of computer science. I study information security and I am happy to implement the Vigenere cryptosystem.

key: adrar

Ciphertext: i dd a khlid pedi skugvnk oi todpxkei sfzeech. z skugp iefrimrtlfn jeflrztb rnu i dd hrpsp tf ipglvmhet khh mixeqvrv cuppkovpskep.

Decryption using key and encrypted text :



Plaintext

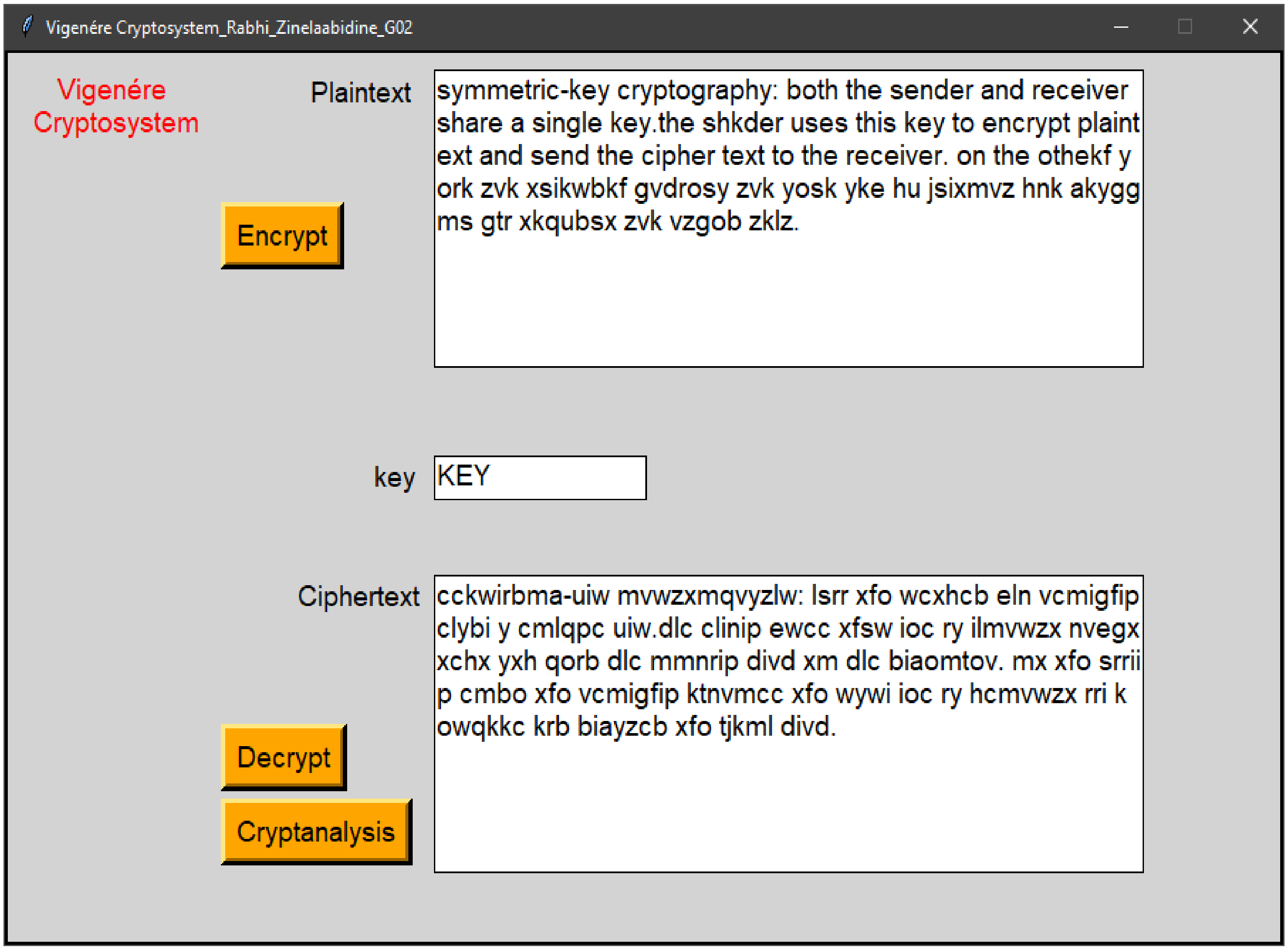
```
information
security
```

key: place

Ciphertext

```
xyfqvbltksc
deeygtta
```

Cryptanalysis

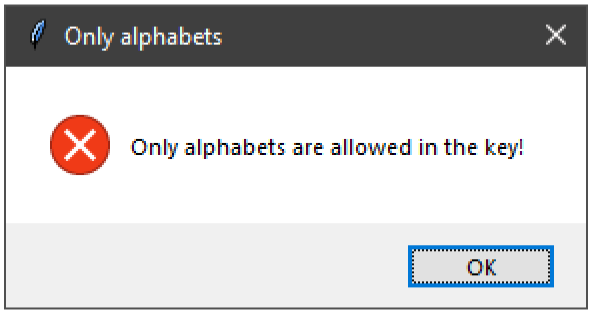

Vigenére Cryptosystem_Rabhi_Zinelaabidine_G02

Plaintext: symmetric-key cryptography: both the sender and receiver share a single key.the shkder uses this key to encrypt plaint ext and send the cipher text to the receiver. on the othekf y ork zvk xsikwbkf gvdrosy zvk yosk yke hu jsixmvz hnk akygg ms gtr xkqubsx zvk vzgob zklz.

key: KEY

Ciphertext: cckwirbma-uiw mvwzxmqvyzlw: lsrr xfo wcxhcb eln vcmigfip clybi y cmlqpc uiw.dlc clinip ewcc xfsw ioc ry ilmvwzx nvegx xchx yxh qorb dlc mmnrip divd xm dlc biaomtov. mx xfo srrii p cmbo xfo vcmigfip ktnvmcc xfo wywi ioc ry hcmvwzx rri k owqkkc krb biayzcb xfo tjkml divd.

If the key contains a number or symbol without the space between the letters or before it is created MessageBox


Only alphabets

Only alphabets are allowed in the key!

Documentation Two Files

vigenerCrypto_Rabhi_Zinelaabidine.py

dictionary.txt

CONCLUSIONS:

1. If you can identify the key, you find the solution to decrypt

2. Encryption analysis is not 100% effective in decryption