الجمهورية العربية السورية
جامعة تشرين
كلية الهندسة الميكانيكية والكهربائية
هندسة الاتصالات والإلكترونيات
السنة الخامسة

"تشفير وفك تشفير الرسائل بلغة البايثون باستخدام مكتبتي Tkinter وBase64"

Encode and Decode messages in Python with Tkinter and Base64 " "libraries

إشراف الدكتور:

مهند عيسى

إعداد:

هديل عبد المجيد محمد — 2824

سلافة عارف عيسى — 2395

من طلاب السنة الخامسة هندسة الاتصالات والإلكترونيات

ملخص:

سنقوم في هذا البحث بكتابة برنامج بلغة بايثون يقوم بتشفير وفك تشفير الرسائل باستخدام مفتاح مشترك وباستخدام مكتبة Tkinter والتي هي مكتبة واجهة المستخدم الرسومية القياسية لبايثون حيث يوفر Python عند دمجه مع Tkinter طريقة سريعة وسهلة لإنشاء تطبيقات واجهة المستخدم الرسومية. ومكتبة Base64 التي توفر توابع لتشفير البيانات الثنائية وتحويلها إلى محارف ASCII أو فك تشفير محارف ال ASCII وتحويلها إلى بيانات ثنائية.

مقدمة:

في هذه الأيام يرتبط كل نشاط بشري ارتباطا وثيقا بأنظمة الحوسبة حيث يتم استخدام هذه الأنظمة في كل تطبيق في مجال الرعاية الصحية والتعليم والخدمات المصرفية والبرمجيات والتسويق. لكن قد نتساءل كيف تقوم المؤسسات بتأمين معلوماتها وكيف يتم الحفاظ على سرية معاملاتنا المصرفية، الجواب على كل ذلك هو **"التشفير"**.

التشفير هو بشكل عام **فن أو تقنية إنشاء رسائل مشفرة باستخدام مفاتيح سرية** حيث لا يمكن فك تشفيرها إلا من قبل الشخص الذي يتم توجيهها إليه أو من يملك المفتاح.

بالتالي هو وحدة البناء الأساسية لأمن البيانات وهو أبسط الطرق وأهمها لضمان عدم سرقة معلومات نظام الحاسوب أو قراءتها من جانب شخص يريد استخدامها لأغراض ضارة.

فوائد البحث:

نتحدث في هذا المشروع عن إنشاء برنامج لتشفير وفك تشفير الرسائل والذي له الفوائد التالية:

١. برنامج مفتوح المصدر يمكن التعديل عليه
٢. البساطة والسهولة في التصميم
٣. الأمان وحماية الخصوصية
٤. التوفير المادي لأنه مجاني
٥. السرعة في التشفير وفك التشفير

الكود الموافق:

```
from tkinter import *
import base64
root = Tk()
root.geometry('500x300')
root.resizable(0,0)
root.title("DataFlair - Message Encode and Decode")
Label(root, text ='ENCODE DECODE', font = 'arial 20 bold').pack()
Label(root, text ='DataFlair', font = 'arial 20 bold').pack(side
=BOTTOM)
Text = StringVar()
private_key = StringVar()
mode = StringVar()
Result = StringVar()
def Encode(key,message):
    enc=[]
    for i in range(len(message)):
        key_c = key[i % len(key)]
        enc.append(chr((ord(message[i]) + ord(key_c)) % 256))
    return base64.urlsafe_b64encode("".join(enc).encode()).decode()
def Decode(key,message):
    dec=[]
    message = base64.urlsafe_b64decode(message).decode()
    for i in range(len(message)):
        key_c = key[i % len(key)]
        dec.append(chr((256 + ord(message[i])- ord(key_c)) % 256))
    return "".join(dec)
def Mode():
    if(mode.get() == 'e'):
        Result.set(Encode(private_key.get(), Text.get()))
    elif(mode.get() == 'd'):
        Result.set(Decode(private_key.get(), Text.get()))
    else:
        Result.set('Invalid Mode')
def Exit():
    root.destroy()
def Reset():
    Text.set("")
    private_key.set("")
    mode.set("")
    Result.set("")
Label(root, font= 'arial 12 bold', text='MESSAGE').place(x= 60,y=60)
Entry(root, font = 'arial 10', textvariable = Text, bg = 'ghost
white').place(x=290, y = 60)
Label(root, font = 'arial 12 bold', text ='KEY').place(x=60, y = 90)
Entry(root, font = 'arial 10', textvariable = private_key , bg
='ghost white').place(x=290, y = 90)
Label(root, font = 'arial 12 bold', text ='MODE(e-encode, d-
decode)').place(x=60, y = 120)
Entry(root, font = 'arial 10', textvariable = mode , bg= 'ghost
white').place(x=290, y = 120)
Entry(root, font = 'arial 10 bold', textvariable = Result, bg ='ghost
white').place(x=290, y = 150)
Button(root, font = 'arial 10 bold', text = 'RESULT'  ,padx =2,bg
='LightGray' ,command = Mode).place(x=60, y = 150)
Button(root, font = 'arial 10 bold' ,text ='RESET' ,width =6, command
= Reset,bg = 'LimeGreen', padx=2).place(x=80, y = 190)
Button(root, font = 'arial 10 bold',text= 'EXIT' , width = 6, command
= Exit,bg = 'OrangeRed', padx=2, pady=2).place(x=180, y = 190)
root.mainloop()
```

المناقشة وطرائق البحث:

```
from tkinter import *
import base64
```

- الخطوة الأولى هي استيراد مكتبات Tkinterو Base64

```
root = Tk()
root.geometry('500x300')
root.resizable(0,0)
root.title("DataFlair - Message Encode and Decode")
```

- Tk () تهيئة مكتبة Tkinter تعني إنشاء النافذة
- Geometry () تعيين عرض وارتفاع النافذة
- resizable (0,0) ضبط الحجم الثابت للنافذة
- title () تحديد عنوان النافذة

```
• Label(root, text ='ENCODE DECODE', font = 'arial 20
  bold').pack()
  Label(root, text ='DataFlair', font = 'arial 20
  bold').pack(side =BOTTOM)
```

- Label () عنصر واجهة المستخدم (اللائحة) الذي يستخدم لعرض سطر واحد أو أكثر من النص الذي لا يستطيع المستخدمون تعديله
- Root هو الاسم الذي يشير إلى نافذة المستخدم
- Text النص الذي نعرضه على اللائحة
- Font الخط الذي يكتب فيه النص
- Pack تنظم اللائحة ضمن النافذة

```
• Text = StringVar()
  private_key = StringVar()
  mode = StringVar()
  Result = StringVar()
```

- Text متغير يخزن الرسالة لتشفيرها وفك تشفيرها
- Private key متغير يخزن المفتاح الخاص بالمستخدم للتشفير وفك التشفير
- mode يستخدم لتحديد ما إذا كان الخيار المستخدم هو تشفير أو فك تشفير
- Result يستخدم لتخزين النتيجة

```
def Encode(key,message):
    enc=[]
    for i in range(len(message)):
        key_c = key[i % len(key)]
        enc.append(chr((ord(message[i]) +
ord(key_c)) % 256))
    return
base64.urlsafe_b64encode("".join(enc).encode()).deco
de()
```

- نعرف تابع Encode يحتوي على بارامترين هما الرسالة المراد تشفيرها ومفتاح التشفير
- enc= [] هي قائمة فارغة
- نقوم بتشغيل حلقة بطول الرسالة
- key_c = key [i % Len(key) باقي قسمة i على طول المفتاح والناتج هو فهرس المفتاح ، قيمة الفتاح التي يدل عليها هذا الفهرس تخزن في key_c
- ord () تابع يأخذ بارامتر من النوع سلسلة من حرف Unicode واحد ويعيد قيمة Unicode الصحيحة
- Chr () تابع يأخذ بارامتر من نوع عدد صحيح ويعيد سلسلة
- Ord (message[i] تحويل قيمة الرسالة في الفهرس i إلى قيمة عدد صحيح
- Ord (key_c)) تحويل قيمة key_c إلى قيمة عدد صحيح
- Ord (message[i]) + ord (key_c)) % 256) يعطي باقي قسمة مجموع Ord (message[i] و Ord (key_c)) على العدد 256 ويمرر ناتج هذا الباقي إلى التابع Chr ()
- Chr () تابع يحول تلك القيمة الصحيحة إلى سلسلة ويخزنها في القائمة الفارغة enc
- base64.urlsafe_b64encode تشفير سلسلة
- join () طريقة الانضمام: يضم كل عنصر من عناصر list, string, tuple بواسطة فاصل سلسلة ويعيد السلسلة المتسلسلة
- Encode () يقوم هذا التابع بإرجاع الرسالة المشفرة من السلسلة utf-8
- Decode () تابع يقوم بفك تشفير السلسلة
- Return يعطي نتيجة السلسلة المشفرة

```
def Decode(key,message):
    dec=[]
    message =
base64.urlsafe_b64decode(message).decode()
    for i in range(len(message)):
        key_c = key[i % len(key)]
        dec.append(chr((256 + ord(message[i])-
ord(key_c)) % 256))
    return "".join(dec)
```

- Decode () تابع يحتوي على بارامترين هما الرسالة المراد تشفيرها ومفتاح التشفير
- Dec= [] هي قائمة فارغة
- فك تشفير المحتوى من الدخل وكتابة النتيجة بالثنائي في الخرج
- نقوم بتشغيل حلقة بطول الرسالة
- (256 + ord(message[i])- ord(key_c)) % 256 يعطي باقي مجموع 256 مع ناتج طرح ل ord(message[i])- ord(key_c) ثم القسمة على 256 ويمرر الباقي إلى التابع Chr ()
- Chr () تابع يحول تلك القيمة الصحيحة إلى سلسلة ويخزنها في القائمة الفارغة Dec
- return "". join(dec) إرجاع النتيجة

```
def Mode():
    if(mode.get() == 'e'):
        Result.set(Encode(private_key.get(), Text.get()))
    elif(mode.get() == 'd'):
        Result.set(Decode(private_key.get(), Text.get()))
    else:
        Result.set('Invalid Mode')
```

- إذا كان الوضع الذي حدده المستخدم هو "e" فسيتم استدعاء التابع Encode ()
- إذا تم ضبط الوضع على "d" فسيتم استدعاء التابع Decode ()
- وإلا يطبع 'Invalid Mode'
- private_key.get () و Text.get () قيم يتم تمريرها إلى بارامترات التوابع Encode () و Decode ()

```
def Exit():
    root.destroy()
```

- root. Destroy () سيقوم بإنهاء البرنامج عن طريق إيقاف ال mainloop

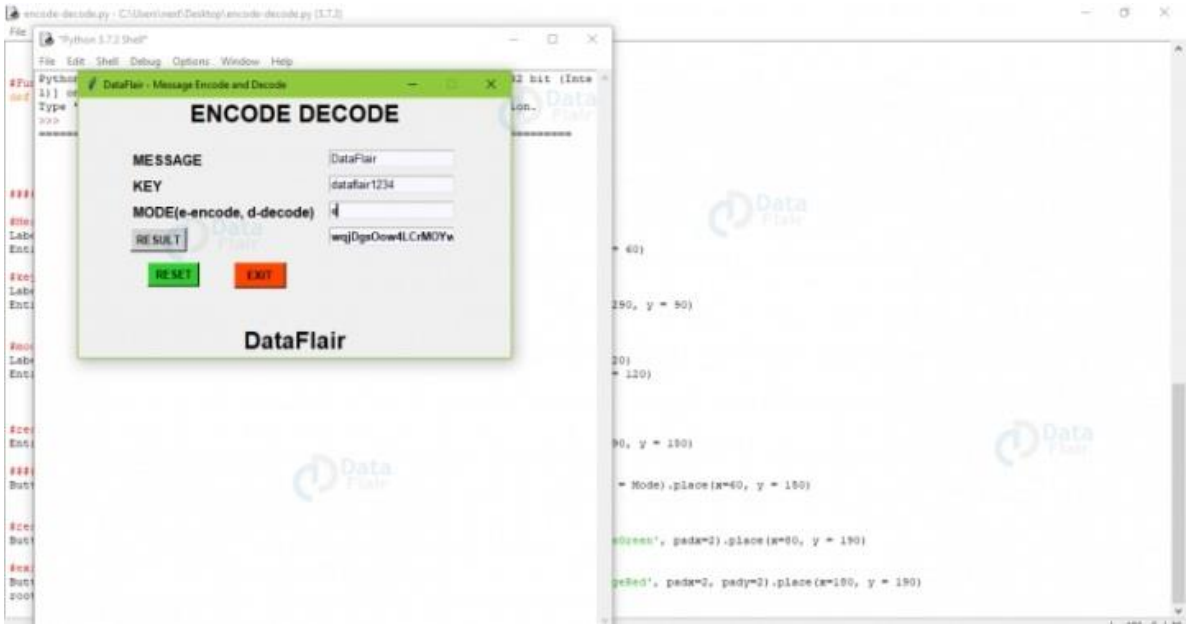
```
def Reset():
    Text.set("")
    private_key.set("")
    mode.set("")
    Result.set("")
```

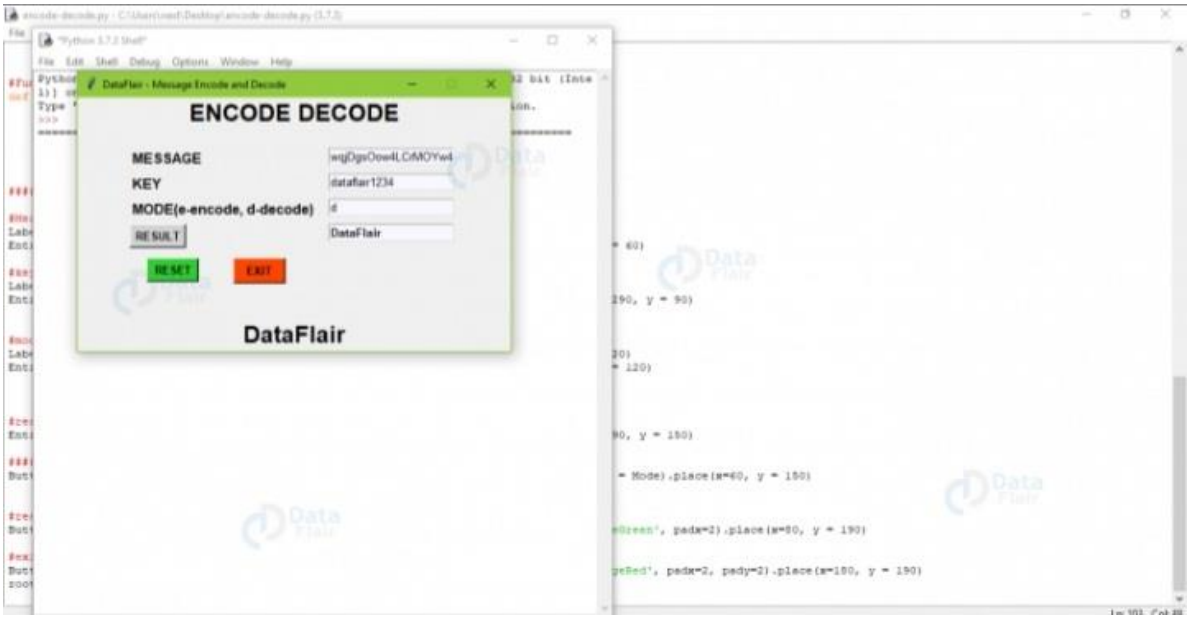
- يقوم هذا التابع بتعيين جميع المتغيرات على سلسلة فارغة

```
Label(root, font= 'arial 12 bold',
text='MESSAGE').place(x= 60,y=60)
Entry(root, font = 'arial 10', textvariable = Text, bg =
'ghost white').place(x=290, y = 60)
Label(root, font = 'arial 12 bold', text
='KEY').place(x=60, y = 90)
Entry(root, font = 'arial 10', textvariable = private_key
, bg ='ghost white').place(x=290, y = 90)
Label(root, font = 'arial 12 bold', text ='MODE(e-encode,
d-decode)').place(x=60, y = 120)
Entry(root, font = 'arial 10', textvariable = mode , bg=
'ghost white').place(x=290, y = 120)
Entry(root, font = 'arial 10 bold', textvariable =
Result, bg ='ghost white').place(x=290, y = 150)
Button(root, font = 'arial 10 bold', text = 'RESULT'
,padx =2,bg ='LightGray' ,command = Mode).place(x=60, y =
150)
Button(root, font = 'arial 10 bold' ,text ='RESET' ,width
=6, command = Reset,bg = 'LimeGreen', padx=2).place(x=80,
y = 190)
Button(root, font = 'arial 10 bold',text= 'EXIT' , width
= 6, command = Exit,bg = 'OrangeRed', padx=2,
pady=2).place(x=180, y = 190)
root.mainloop()
```

- Label () عنصر واجهة المستخدم (اللائحة) الذي يستخدم لعرض سطر واحد أو أكثر من النص الذي لا يستطيع المستخدمون تعديله
- Entry () عنصر واجهة المستخدم يستخدم لإنشاء حقل إدخال نصي
- Button () عنصر واجهة المستخدم يستخدم لعرض زر على نافذتنا
- Root هو الاسم الذي يشير إلى نافذة المستخدم
- Text النص الذي نعرضه على اللائحة
- Font الخط الذي يكتب فيه النص
- width استخدام عرض الدخل لتعيين عرض مؤشر الإدراج
- Bg مجموعات لون الخلفية
- command هو الاتصال عند النقر على الزر
- textvariable يستخدم لاسترداد النص الحالي إلى عنصر واجهة المستخدم
- root. Mainloop () تابع يتم تنفيذه عندما نريد تشغيل برنامجنا

نتيجة خرج كود تشفير وفك تشفير الرسائل:





المراجع:

- https://data-flair.training/blogs/python-message-encode-decode/
- https://youtu.be/0NfDGNcuyAQ
- https://arabicprogrammer.com
- https://www.krsan4learn.com