```
# Tran Anh Khoa
# B1913240

# -*- coding: utf8 -*-
from Crypto import Random
from Crypto.Cipher import PKCS1_v1_5
from Crypto.PublicKey import RSA
from tkinter import filedialog
from tkinter import *
import tkinter as tk
from Crypto.Cipher import DES
import base64


def pad(s):
    return s + (8 - len(s) % 8) * chr(8 - len(s) % 8)


def unpad(s):
    return s[:-ord(s[len(s)-1:])]


class MAHOA_DES(tk.Toplevel):
    def __init__(self, parent):
        self.parent = parent
        Toplevel.__init__(self)
        self.title("Chương trình mã hóa đối xứng")
        self.geometry('800x600')
        self.lbl = Label(self,
                         text="CHƯƠNG TRÌNH DEMO",
                         font=("Arial Bold", 20))
        self.lbl.grid(column=1, row=1)
        self.lb2 = Label(self,
                         text="MẬT MÃ ĐỐI XỨNG DES",
                         font=("Arial Bold", 15))
        self.lb2.grid(column=1, row=2)
        self.plainlb3 = Label(self,
                              text="Văn bản gốc",
font=("Arial", 14))
        self.plainlb3.grid(column=0, row=4)
        self.plaintxt = Entry(self, width=100)
        self.plaintxt.grid(column=1, row=4)
        self.lb4 = Label(self, text="Khóa", font=("Arial",
14))
        self.lb4.grid(column=0, row=5)
        self.keytxt = Entry(self, width=100)
        self.keytxt.grid(column=1, row=5)
        self.lb5 = Label(self,
                         text="Văn bản được mã hóa",
font=("Arial", 14))
        self.lb5.grid(column=0, row=6)
        self.ciphertxt = Entry(self, width=100)
        self.ciphertxt.grid(column=1, row=6)
        self.lb6 = Label(self,
                         text="Văn bản được giải mã",
font=("Arial", 14))
        self.lb6.grid(column=0, row=7)
        self.denctxt = Entry(self, width=100)
        self.denctxt.grid(column=1, row=7)
        self.btn_enc = Button(self, text="Mã Hóa",
                              command=self.mahoa_DES)
        self.btn_enc.grid(column=1, row=9)
```

```
        self.btn_dec = Button(self, text="Giải Mã ",
                              command=self.giaima_DES)
        self.btn_dec.grid(column=1, row=10)
        self.thoat = Button(self, text="Quay về màn hình
chính",
                            command=self.destroy)
        self.thoat.grid(column=1, row=11)

    def mahoa_DES(self):
        txt = pad(self.plaintxt.get()).encode()
        key = pad(self.keytxt.get()).encode()
        cipher = DES.new(key, DES.MODE_ECB)
        entxt = cipher.encrypt(txt)
        entxt = base64.b64encode(entxt)
        self.ciphertxt.delete(0, END)
        self.ciphertxt.insert(INSERT, entxt)

    def giaima_DES(self):
        txt = self.ciphertxt.get()
        txt = base64.b64decode(txt)
        key = pad(self.keytxt.get()).encode()
        cipher = DES.new(key, DES.MODE_ECB)
        detxt = unpad(cipher.decrypt(txt))
        self.denctxt.delete(0, END)
        self.denctxt.insert(INSERT, detxt)


def Char2Num(c):
    return ord(c)
```

```
def Num2Char(n):
    return chr(n)


def encryptAF(txt, a, b, m):
    r = ""
    for c in txt:
        e = (a*Char2Num(c)+b) % m
        r = r+Num2Char(e)
    return r


def xgcd(a, m):
    temp = m
    x0, x1, y0, y1 = 1, 0, 0, 1
    while m != 0:
        q, a, m = a // m, m, a % m
        x0, x1 = x1, x0 - q * x1
        y0, y1 = y1, y0 - q * y1
    if x0 < 0:
        x0 = temp+x0
    return x0


def decryptAF(txt, a, b, m):
    r = ""
    a1 = xgcd(a, m)
    for c in txt:
        e = (a1*(Char2Num(c)-b)) % m
        r = r+Num2Char(e)
    return r
```

```
class MAHOA_Affine(tk.Toplevel):
    def __init__(self, parent):
        self.parent = parent
        Toplevel.__init__(self)
        self.title("Welcome to Demo An Toàn Bảo Mật Thông
Tin")
        self.geometry('1000x600')
        self.lb0 = Label(self, text=" ", font=("Arial Bold",
10))
        self.lb0.grid(column=0, row=0)
        self.lbl = Label(self, text="CHƯƠNG TRÌNH DEMO",
                         font=("Arial Bold", 20))
        self.lbl.grid(column=1, row=1)
        self.lb2 = Label(self, text="MẬT MÃ AFFINE",
font=("Arial Bold", 15))
        self.lb2.grid(column=0, row=2)
        self.plainlb3 = Label(self, text="PLAIN TEXT",
font=("Arial", 14))
        self.plainlb3.grid(column=0, row=3)
        self.plaintxt = Entry(self, width=20)
        self.plaintxt.grid(column=1, row=3)
        self.KEYlb4 = Label(self, text="KEY PAIR",
font=("Arial", 14))
        self.KEYlb4.grid(column=2, row=3)
        self.KEYA1 = Entry(self, width=3)
        self.KEYA1.grid(column=3, row=3)
        self.KEYB1 = Entry(self, width=5)
        self.KEYB1.grid(column=4, row=3)
        self.plainlb4 = Label(self, text="CIPHER TEXT",
font=("Arial", 14))
        self.plainlb4.grid(column=0, row=4)
        self.ciphertxt3 = Entry(self, width=20)
        self.ciphertxt3.grid(column=1, row=4)
        self.encryptAFtxt = Entry(self, width=20)
        self.encryptAFtxt.grid(column=3, row=4)

        self.AFbtn = Button(self, text="Mã Hóa",
command=self.mahoa)
        self.AFbtn.grid(column=5, row=3)

        self.DEAFbtn = Button(self, text="Giải Mã",
command=self.runDescriptAF)
        self.DEAFbtn.grid(column=2, row=4)

        self.thoat = Button(self, text="Quay về màn hình
chính",
                            command=self.destroy)
        self.thoat.grid(column=1, row=6)

        self.geometry('1000x600')

    def mahoa(self):
        a = int(self.KEYA1.get())
        b = int(self.KEYB1.get())
        m = 127
        entxt = encryptAF(self.plaintxt.get(), a, b, m)
        self.ciphertxt3.delete(0, END)
        self.ciphertxt3.insert(INSERT, entxt)
```

```
    def runDescriptAF(self):
        a = int(self.KEYA1.get())
        b = int(self.KEYB1.get())
        m = 127
        temptxt = decryptAF(self.ciphertxt3.get(), a, b, m)
        self.encryptAFtxt.delete(0, END)
        self.encryptAFtxt.insert(INSERT, temptxt)


def save_file(content, _mode, _title, _filetypes,
_defaultextension):
    f = filedialog.asksaveasfile(mode=_mode,
                                 initialdir="D:/",
                                 title=_title,
                                 filetypes=_filetypes,
                                 defaultextension=_defaultexte
nsion)
    if f is None:
        return
    f.write(content)
    f.close()


def get_key(key_style):
    filename = filedialog.askopenfilename(initialdir="D:/",
                                          title="Open " +
key_style,
                                          filetypes=(("PEM
files", "*.pem"), ("All files", "*.*")))
    if filename is None:
        return
    file = open(filename, "rb")
    key = file.read()
    file.close()
    return RSA.importKey(key)


class MAHOA_RSA(tk.Toplevel):
    def __init__(self, parent):
        self.parent = parent
        Toplevel.__init__(self)
        self.title("Welcome to Demo An Toàn Bảo Mật Thông
Tin")

        # Them cac control
        self.lb0 = Label(self, text=" ", font=("Arial Bold",
10))
        self.lb0.grid(column=0, row=0)
        self.lbl = Label(self, text="CHƯƠNG TRÌNH DEMO",
                         font=("Arial Bold", 20))
        self.lbl.grid(column=1, row=1)
        self.lb2 = Label(self, text="MẬT MÃ ĐỐI XỨNG RSA",
                         font=("Arial Bold", 15))
        self.lb2.grid(column=1, row=2)

        widthEntry = 80
        height = 5

        self.plainlb3 = Label(self, text="Văn bản gốc",
font=("Arial", 14))
        self.plainlb3.grid(column=0, row=3)
        self.plaintxt = Entry(self, width=widthEntry)
```

```
        self.plaintxt.grid(column=1, row=3)

        self.cipherlb5 = Label(
            self, text="Văn bản được mã hoá", font=("Arial",
14))
        self.cipherlb5.grid(column=0, row=4)
        self.ciphertxt = Entry(self, width=widthEntry)
        self.ciphertxt.grid(column=1, row=4)

        self.denclb6 = Label(
            self, text="Văn bản được giải mã", font=("Arial",
14))
        self.denclb6.grid(column=0, row=5)
        self.denctxt = Entry(self, width=widthEntry)
        self.denctxt.grid(column=1, row=5)

        self.privateKeylb4 = Label(
            self, text="Khoá Cá Nhân", font=("Arial", 14))
        self.privateKeylb4.grid(column=0, row=6)
        self.privateKeytxt = Text(self, width=50,
height=height)
        self.privateKeytxt.grid(column=1, row=6)

        self.publicKeylb4 = Label(
            self, text="Khoá Công Khai", font=("Arial", 14))
        self.publicKeylb4.grid(column=0, row=7)
        self.publicKeytxt = Text(self, width=50,
height=height)
        self.publicKeytxt.grid(column=1, row=7)

        self.createKeybtn = Button(
            self, text="Tạo Khóa", command=self.generate_key)
        self.createKeybtn.grid(column=1, row=8)

        self.AFbtn = Button(self, text="Mã Hóa",
command=self.mahoa_rsa)
        self.AFbtn.grid(column=1, row=9)

        self.DEAFbtn = Button(self, text="Giải Mã",
command=self.giaima_rsa)
        self.DEAFbtn.grid(column=1, row=10)

        self.geometry('800x500')

        self.thoat = Button(self, text="Quay về màn hình
chính",
                            command=self.destroy)
        self.thoat.grid(column=1, row=11)

    def generate_key(self):
        key = RSA.generate(1024)
        pri = save_file(key.exportKey('PEM'),
                        'wb',
                        'Lưu khóa cá nhân',
                        (("All files", "*.*"), ("PEM files",
"*.pem")),
                        ".pem")
        pub = save_file(key.publickey().exportKey('PEM'),
                        'wb',
                        'Lưu khóa công khai',
                        (("All files", "*.*"), ("PEM files",
"*.pem")),
```

```
                        ".pem")
        self.privateKeytxt.delete('1.0', END)
        self.privateKeytxt.insert(END, key.exportKey('PEM'))
        self.publicKeytxt.delete('1.0', END)
        self.publicKeytxt.insert(END,
key.publickey().exportKey('PEM'))

    def mahoa_rsa(self):
        txt = self.plaintxt.get().encode()
        publicKeytxt = get_key("Public Key")
        cipher = PKCS1_v1_5.new(publicKeytxt)
        entxt = cipher.encrypt(txt)
        entxt = base64.b64encode(entxt)
        self.ciphertxt.delete(0, END)
        self.ciphertxt.insert(INSERT, entxt)

    def giaima_rsa(self):
        txt = self.ciphertxt.get()
        txt = base64.b64decode(txt)
        privateKeytxt = get_key("Private Key")
        cipher = PKCS1_v1_5.new(privateKeytxt)
        dsize = 128
        sentinel = Random.new().read(dsize)
        entxt = cipher.decrypt(txt, sentinel)
        self.denctxt.delete(0, END)
        self.denctxt.insert(INSERT, entxt)


class MainWindow(tk.Frame):
    def __init__(self, parent):
        self.parent = parent
        tk.Frame.__init__(self)

        self.mahoa_Affine = Button(text="Mã hóa Affine",
                                   font=("Times New Roman",
11),
                                   command=self.affine)
        self.mahoa_Affine.pack()

        self.mahoa_DES = Button(text="Mã hóa DES",
                                font=("Times New Roman", 11),
                                command=self.des)
        self.mahoa_DES.pack()

        self.mahoa_RSA = Button(text="Mã hóa RSA",
                                font=("Times New Roman", 11),
                                command=self.rsa)
        self.mahoa_RSA.pack()

        self.thoat = Button(text="Kết Thúc",
                            font=("Times New Roman", 11),
                            command=quit)
        self.thoat.pack()

    def des(self):
        MAHOA_DES(self)

    def affine(self):
        MAHOA_Affine(self)

    def rsa(self):
        MAHOA_RSA(self)
```

```
def main():
    window = tk.Tk()
    window.title("Chương trình chính")
    window.geometry('300x200')
    MainWindow(window)
    window.mainloop()


main()
```

```
# Lab 04 Bài 1
# Ho va ten sinh vien: Tran Anh Khoa
# Ma so sinh vien: B1913240

from tkinter import *
from tkinter import filedialog
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Hash import MD5, SHA1, SHA256, SHA512
from Crypto.Cipher import PKCS1_v1_5
import base64

def hashing():
    content = plaintxt.get().encode()
    func = hashmode.get()
    if func == 0:
        result = MD5.new(content)
    if func == 1:
        result = SHA1.new(content)
    if func == 2:
        result = SHA256.new(content)
    if func == 3:
        result = SHA512.new(content)
    # Học viên tự cài đặt các phương thức cho SHA256 và SHA512
    rs = result.hexdigest().upper()
    hashvalue.delete(0,END)
    hashvalue.insert(INSERT,rs)

window = Tk()
window.title("Welcome to Demo An Toàn Bảo Mật Thông Tin")
app_name = Label(window, text="CHƯƠNG TRÌNH BĂM", font=("Arial
Bold", 20))
app_name.grid(column=1, row=1)
plainlb0 = Label(window, text="Văn bản", font=("Arial", 14))
plainlb0.grid(column=0, row=3)
plaintxt = Entry(window, width=95)
plaintxt.grid(column=1, row=3)

# radio
radioGroup = LabelFrame(window, text = "Hàm băm")
radioGroup.grid(row=4, column=1)
hashmode = IntVar()
hashmode.set(-1)

# md5
md5_func = Radiobutton(radioGroup,
    text="Hash MD5",
    font=("Times New Roman", 11),
    variable=hashmode,
    value=0,
    command=hashing)
md5_func.grid(row=4, column=0)

# sha1
sha1_func = Radiobutton(radioGroup,
    text="Hash SHA1",
    font=("Times New Roman", 11),
    variable=hashmode,
    value=1,
    command=hashing)
sha1_func.grid(row=5, column=0)
```

```
# sha256
sha1_func = Radiobutton(radioGroup,
    text="Hash SHA256",
    font=("Times New Roman", 11),
    variable=hashmode,
    value=2,
    command=hashing)
sha1_func.grid(row=6, column=0)

# sha512
sha1_func = Radiobutton(radioGroup,
    text="Hash SHA512",
    font=("Times New Roman", 11),
    variable=hashmode,
    value=3,
    command=hashing)
sha1_func.grid(row=7, column=0)

# hash out put
hash_out_put = Label(window, text="Giá trị băm",
font=("Arial", 14))
hash_out_put.grid(column=0, row=5)
hashvalue = Entry(window, width=95)
hashvalue.grid(column=1, row=5)

# Tương tự đối với sha256 và sha512
window.geometry('800x500')
window.mainloop()

# Lab 04 Bài 2
# Ho va ten sinh vien: Tran Anh Khoa
# Ma so sinh vien: B1913240

import tkinter as tk
from tkinter import *
from tkinter import filedialog, messagebox
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Hash import MD5, SHA1, SHA256, SHA512
from Crypto.Cipher import PKCS1_v1_5
import base64
import os
import csv
import random
from pathlib import Path

def hashing(username):
    content = password.get().encode()
    func = random.randint(0, 3)
    if func == 0:
        result = MD5.new(content)
    if func == 1:
        result = SHA1.new(content)
    if func == 2:
        result = SHA256.new(content)
    if func == 3:
        result = SHA512.new(content)
    # Học viên tự cài đặt các phương thức cho SHA256 và SHA512
    rs = result.hexdigest().upper()
    # create records
    directory = os.path.abspath(os.path.join(os.path.curdir))
```

```
    file = Path(directory + "/CSDL.csv")
    file.touch(exist_ok=True)
    write_able = True
    with open(directory + "/CSDL.csv", "r") as file_obj:
        reader_obj = csv.reader(file_obj)
        for row in reader_obj:
            try:
                if(row[0] == username):
                    tk.messagebox.showinfo("Thông báo thất
bại.",  "Tài khoản đã tồn tại")
                    write_able = FALSE
            except IndexError:
                print('except block ran')
                continue
    if(write_able == True):
        tk.messagebox.showinfo("Thông báo thành công.",  "Thêm
tài khoản thành công")
        with open(directory + "/CSDL.csv", "a") as file_obj:
            csvWriter = csv.writer(file_obj, delimiter=',')
            csvWriter.writerow([username, rs])
    return rs

window = Tk()
window.title("Welcome to Demo An Toàn Bảo Mật Thông Tin")
app_name = Label(window, text="Tạo tài khoản", font=("Arial
Bold", 20))
app_name.grid(column=1, row=1)

usernamelb = Label(window, text="Tên đăng nhập",
font=("Arial", 14))
usernamelb.grid(column=0, row=3)
username = Entry(window, width=50)
username.grid(column=1, row=3)

passwordlb = Label(window, text="Mật khẩu", font=("Arial",
14))
passwordlb.grid(column=0, row=4)
password = Entry(window, width=50)
password.grid(column=1, row=4)

create_account = Button(window, text="Tạo tài khoản", command=
lambda: hashing(username.get()))
create_account.grid(column=1, row=9)

window.geometry('500x200')
window.mainloop()
```

```
# Lab04 Bài 3
# Ho va ten sinh vien: Tran Anh Khoa
# Ma so sinh vien: B1913240

import tkinter as tk
from tkinter import *
from tkinter import filedialog, messagebox
from Crypto.PublicKey import RSA
from Crypto import Random
from Crypto.Hash import MD5, SHA1, SHA256, SHA512
from Crypto.Cipher import PKCS1_v1_5
import base64
import os
import csv
import random
from pathlib import Path
def decrypt(index, content):
    if index == 0:
        result = MD5.new(content)
    if index == 1:
        result = SHA1.new(content)
    if index == 2:
        result = SHA256.new(content)
    if index == 3:
        result = SHA512.new(content)
    return result.hexdigest().upper()

def hashing(username):
    content = password.get().encode()
    directory = os.path.abspath(os.path.join(os.path.curdir))
    file = Path(directory + "/CSDL.csv")
    file.touch(exist_ok=True)
    write_able = True
    with open(directory + "/CSDL.csv", "r") as file_obj:
        reader_obj = csv.reader(file_obj)
        for row in reader_obj:
            try:
                if(row[0] == username):
                    for i in range(0, 4):
                        if(row[1] == decrypt(i, content)):
                            tk.messagebox.showinfo("Thông
báo.",  "Đăng nhập thành công")
                            write_able = FALSE
            except IndexError:
                print('except block ran')
                continue
    if(write_able == True):
        tk.messagebox.showinfo("Thông báo thất bại.",  "Đăng
nhập thất bại")

window = Tk()
window.title("Welcome to Demo An Toàn Bảo Mật Thông Tin")
app_name = Label(window, text="Đăng nhập", font=("Arial Bold",
20))
app_name.grid(column=1, row=1)

usernamelb = Label(window, text="Tên đăng nhập",
font=("Arial", 14))
usernamelb.grid(column=0, row=3)
username = Entry(window, width=50)
username.grid(column=1, row=3)
```

```
passwordlb = Label(window, text="Mật khẩu", font=("Arial",
14))
passwordlb.grid(column=0, row=4)
password = Entry(window, width=50)
password.grid(column=1, row=4)

create_account = Button(window, text="Đăng nhập", command=
lambda: hashing(username.get()))
create_account.grid(column=1, row=9)

window.geometry('500x200')
window.mainloop()
```