#encoding=utf-8

#Copyright HXBer

from tkinter import \*

import tkinter.messagebox

import base64

import rsa

'''可设定指定RSA公司钥

with open('./rsa\_public\_key.pem', 'r') as f:

pubkey = rsa.PublicKey.load\_pkcs1(f.read().encode('utf-8'))

with open('./rsa\_private\_key.pem', 'r') as f:

privkey = rsa.PrivateKey.load\_pkcs1(f.read().encode('utf-8'))

'''

(pubkey, privkey) = rsa.newkeys(1024) #获取随机RSA公私钥

def rsa\_encrypt(): #加密

inp2.delete('1.0', 'end')

plain = inp1.get('1.0', 'end')[:-1]

plain = plain.encode('utf-8')

print(plain)

val\_list = []

for i in range(0, len(plain), 117): #进行分段加密并合并

tpl = plain[i:i + 117]

val = rsa.encrypt(tpl, pubkey)

val\_list.append(val)

cipher = b''.join(val\_list)

cipher = base64.b64encode(cipher)

inp2.insert(END, cipher)

print(cipher)

def rsa\_sign(): #签名

inp3.delete('1.0', 'end')

cipher = inp1.get('1.0', 'end')[:-1]

crypto = rsa.sign(cipher.encode('utf-8'), privkey, 'SHA-1') #进行签名

crypto = base64.b64encode(crypto)

print(crypto)

inp3.insert(END, crypto)

def transl(): #传输

inp5.delete('1.0', 'end')

content = inp2.get('1.0', 'end')[:-1]

inp5.insert(END, content)

inp6.delete('1.0', 'end')

content = inp3.get('1.0', 'end')[:-1]

inp6.insert(END, content)

def rsa\_decrypt(): #解密

inp4.delete('1.0', 'end')

cipher = base64.b64decode(inp5.get('1.0', 'end')[:-1]) #获取密文

val\_list = []

for i in range(0, len(cipher), 128): #进行分段解密并合并

tpl = cipher[i:i + 128]

val = rsa.decrypt(tpl, privkey)

val\_list.append(val)

plain = b''.join(val\_list) #合并明文

plain = plain.decode('utf-8')

inp4.insert(END, plain)

print(plain)

def rsa\_verif(): #校验

indata = inp4.get('1.0', 'end')[:-1]

indata = indata.encode('utf-8')

signature = base64.b64decode(inp6.get('1.0', 'end')[:-1]) #获取签名

print("indata", indata)

print("signature", signature)

try:

rsa.verify(indata, signature, pubkey) #校验签名

tkinter.messagebox.showinfo('Result', 'correct')

except rsa.VerificationError:

tkinter.messagebox.showinfo('Result', 'incorrect')

raise ('Verification failed.')

root = tkinter.Tk()

root.geometry('900x600')

root.title('RSA')

#设定文本

Label(root, text="plain", anchor=NW).grid(row=0, column=0)

Label(root, text="plain", anchor=NE).grid(row=0, column=1)

Label(root, text="cipher", anchor=W).grid(row=3, column=0)

Label(root, text="cipher", anchor=E).grid(row=3, column=1)

Label(root, text="sign", anchor=W).grid(row=5, column=0)

Label(root, text="sign", anchor=E).grid(row=5, column=1)

inp1 = Text(root, width=60, height=15, relief=GROOVE)

inp2 = Text(root, width=60, height=9, relief=GROOVE)

inp3 = Text(root, width=60, height=9, relief=GROOVE)

##设定输入框

inp1.grid(row=1, column=0)

inp2.grid(row=4, column=0)

inp3.grid(row=6, column=0)

inp4 = Text(root, width=60, height=15, relief=GROOVE)

inp5 = Text(root, width=60, height=9, relief=GROOVE)

inp6 = Text(root, width=60, height=9, relief=GROOVE)

inp4.grid(row=1, column=1)

inp5.grid(row=4, column=1)

inp6.grid(row=6, column=1)

##设定按钮

btn1 = Button(root, width=5, text='Encode', command=rsa\_encrypt)

btn2 = Button(root, width=5, text='Sign', command=rsa\_sign)

btn3 = Button(root, width=5, text='Transl', command=transl)

btn4 = Button(root, width=5, text='Decode', command=rsa\_decrypt)

btn5 = Button(root, width=5, text='Verify', command=rsa\_verif)

btn1.place(relx=0.08, rely=0.9)

btn2.place(relx=0.28, rely=0.9)

btn3.place(relx=0.48, rely=0.9)

btn4.place(relx=0.68, rely=0.9)

btn5.place(relx=0.88, rely=0.9)

mainloop()

通过RSA加密对明文进行加密，然后对明文生成签名，然后传送。对密文进行解密，然后验证签章。