```python
# alice.py
# c. Cargar la clave privada de Alice y la clave pública de Bob.
# d. Cifrar el texto “Hola amigos de la seguridad” utilizando la clave de Bob.
# e. Firmar el texto “Hola amigos de la seguridad” utilizando la clave de Alice.
# f. Guardar en unos ficheros, el texto cifrado y la firma digital.

from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
from Crypto.Signature import pss
from Crypto.Hash import SHA256

def crear_RSAKey():
    key = RSA.generate(2048)
    return key

def guardar_RSAKey_Privada(fichero, key, password):
    key_cifrada = key.export_key(passphrase=password, pkcs=8, protection="scryptAndAES128-CBC")
    file_out = open(fichero, "wb")
    file_out.write(key_cifrada)
    file_out.close()

def cargar_RSAKey_Privada(fichero, password):
    key_cifrada = open(fichero, "rb").read()
    key = RSA.import_key(key_cifrada, passphrase=password)
    return key

def guardar_RSAKey_Publica(fichero, key):
    key_pub = key.publickey().export_key()
    file_out = open(fichero, "wb")
    file_out.write(key_pub)
    file_out.close()

def cargar_RSAKey_Publica(fichero):
    keyFile = open(fichero, "rb").read()
    key_pub = RSA.import_key(keyFile)
    return key_pub

def cifrarRSA_OAEP(cadena, key):

    datos = cadena.encode("utf-8")
    engineRSACifrado = PKCS1_OAEP.new(key)
    cifrado = engineRSACifrado.encrypt(datos)
    return cifrado

def descifrarRSA_OAEP(cifrado, key):
    engineRSADescifrado = PKCS1_OAEP.new(key)
    datos = engineRSADescifrado.decrypt(cifrado)
    cadena = datos.decode("utf-8")
    return cadena

def firmarRSA_PSS(texto, key_private):
# La firma se realiza sobre el hash del texto (h)
    h = SHA256.new(texto.encode("utf-8"))
    print(h.hexdigest())
    signature = pss.new(key_private).sign(h)
    return signature

def comprobarRSA_PSS(texto, firma, key_public):
# Comprobamos que la firma coincide con el hash (h)
    h = SHA256.new(texto.encode("utf-8"))
```

```
    print(h.hexdigest())
    verifier = pss.new(key_public)
    try:
        verifier.verify(h, firma)
        return True
    except (ValueError, TypeError):
        return False

def savefile (file, data):
    file_out = open(file, "wb")
    file_out.write(data)
    file_out.close()


password = "1234"

#carga la clave privada de Alice
alice_Priv_Key = cargar_RSAKey_Privada("A_priv.pkcs", password)

#carga la clave pública de Bob
bob_Pub_Key = cargar_RSAKey_Publica("B_pub.pkcs")

#cadena a cifrar y firmar
cadena = "Hola amigos de la seguridad"

#cifra la cadena y guarda el resultado en un archivo cadenaCifradaAliceBob.bin
#solo Bob con su clave privada podrá descifrar el contenido del erchivo
cifrado = cifrarRSA_OAEP(cadena, bob_Pub_Key)
print("el texto cifrado es:", cifrado)
savefile("cadenaCifradaAliceBob.bin", cifrado)

#firma la cadena y la guarda en un archivo cadenaFirmadaAlice.bin
#la función proporcionada se encarga de calcular el hash y firmarlo
aliceSignedData = firmarRSA_PSS(cadena, alice_Priv_Key)
savefile("cadenaFirmadaAlice.bin", aliceSignedData)


```