

Ejemplo-de-Examen-de-Practicas.pdf



Juandf03



Seguridad de la Información



3º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga**



La mejor escuela de negocios en energía, sostenibilidad y medio ambiente de España.

Más información
www.eoi.es

Formamos
talento para un futuro
Sostenible



100% Empleabilidad



Modalidad: Presencial u online



**Programa de Becas,
Bonificaciones y Descuentos**



X
SIBUYA
URBANSUSHIBAR

Hay rollos y rollos... Me vas a comparar tu rollo de ayer
con un Hot Roll, claramente el sushi está más bueno

HAZ CLIC PARA HACER TU

RESERVA



```
from Crypto.Cipher import PKCS1_OAEP, DES, AES
from Crypto.PublicKey import RSA
from Crypto.Hash import SHA256, HMAC
from Crypto.Signature import pss
from Crypto.Util.Padding import pad, unpad
import base64

# Nombre: Juan
# Apellidos: Diaz-Flores Merino
# Grado: Ingenieria Informatica

print("#=====")
print("# PREGUNTA 1 (2,5 puntos)")
print("#=====")
# Enunciado
#-----
# Se proporcionan las variables "key" (clave de 128 bits), "ciphertext"
# (texto cifrado con esa
# clave de 128 bits), y "hash" (hash del texto en claro, utilizando
# SHA256).
# Se pide descifrar el texto cifrado utilizando el modo de operacion
# ECB, y mostrar el texto en
# claro por pantalla utilizando print() si el hash de ese texto en
# claro y la variable "hash" coinciden.
# La primitiva de criptografia simetrica utilizada en este ejercicio
# debera ser deducida por el
# alumno en base a la informacion proporcionada.

# Ejercicio
#-----
BLOCK_SIZE = 16
key = b'0123456789ABCDEF' # Que mecanismo criptografico usa 128 bits
# como clave secreta...
ciphertext =
b"' \xa8\x08\x89(\xe2\x81\x9c9\x98\xeb\x0n\x16<\x04\x93:\xdaBX\xb8dP\x9
9\x9a?s\x85u&\xa6\xaf\xecx\xfeoo\xc2\xa4\xc4\xc2I\t&\xb0t\xafn&\x04\x9e
\xfb'\xbd\x9e\xaaw\x0f\xe3Jq\xc5\xae@\x82p\x82:86\xf5k\xc3\xae\xda\x01U
D!"
hash =
b'*\x9f\xa7\xffv\xc3\xf1\xcc\xe2\xc3R\xd1M\nT\x9bS\x94\x87j\xab\xbeI\xd
43\xdb\xde\x01\x9c}\xd0\xd9'

decipher = AES.new(key, AES.MODE_ECB)
```

WUOLAH

```

texto = unpad(decipher.decrypt(ciphertext), BLOCK_SIZE)
h = SHA256.new()
h.update(texto)
if(h.digest() == hash):
    print(texto.decode("utf-8"))
else :
    print("Los hash no coinciden")

print("#=====")
print("# PREGUNTA 2 (2,5 puntos)")
print("#=====")
# Enunciado
#-----
# Se proporcionan las variables "private" (clave privada RSA), "public"
(clave publica RSA)
# y "signatureRSA" (firma del texto en claro utilizando RSASSA-PSS y la
clave privada).
# Se pide descifrar el texto cifrado, y mostrar el texto en claro por
pantalla utilizando print(),
# si el hash SHA256 de ese texto en claro puede ser verificado con la
variable "signatureRSA".

# Ejercicio
#-----
private = RSA.import_key(b'-----BEGIN RSA PRIVATE
KEY-----\nMIIEowIBAAKCAQEA\niD6Hta5ks4B3fyzfFPD5DdoC09O8MGC6Hcae\n
qiaCsQrao9\nL4VOQ1zAuD2+HuHtznLTXp+Svq7g3T+k88D7JTD1KpaP254rrMS/pJy4wCCNwVks\n
LqeKin6Onh9ybQFuSPURXSj3+V02qC7I1rD3zoBIOQhJTVeU3pKRgOAjZ4VQAmBh\n
nmicIgIKrftBdhiI9wf7GgIq5Kd8ajRbfTi7Yo9gYaiwpeUUYPLB2P57dKgDh8xEw4\n
nWRKMfjhvgHn7eJ9AW4/iTYi/GdEvqyINoSSw9U0mxkRh1BtwUAKnmMyhm71MNJwE\n
nlkFEce+xd8QVDZxZ/M3715Grts31NmboJP0ZWwIDAQABAoIBAEDyb3jaHbdHyKmK\n
nAJhIeVVTUncOuHAXMvLS9Hu7mNkVKxEBModBm96S5Q7nQR7DEd7w95LOPMH35uDI\n
norIBKcXj7Mo0s9pysSKRXts4CYPT+xUWUJjK9JKkn2Qfq2pNI6Rwj5SxXoQ7vla+\n
nfgG0Rtu4gbF3D1Bmb/0ouv1xR2ZFieqgkS1CMJwt+DiV8RYQ/9+BOFafPtISYYgM\n
nwk7qo7n88N/SfIK3AiQZbPg6vJIeKfj8yYtb9DaO14phXevRaoixrbCptDUPIK4m\n
n5C9G9tyjEZRaRCD/kng41j4YMWb+8dBO/o06th1hKKee75FfOCWw96QhLU0JmmfW\n
nwmRspXECgYEAx5vXOraKk0XJ9hoyZ73asavFHY6LpLiyUDLb6XQ21rZj7B8elJz2\n
\nnfKeMFIduyrA/HNDx5h1louj9DU5lamwvWTh6dPz+YREMTGo2tAhLf4qQa8e8qtly\n
\nngY/crl7N11lOq5C3WdIXL2+HRIFrMQy0aM/XF7UUjqcFLVQ70MH9eRECGYEAys02\n
\nngJ4zInq6CUPRE2ZgjQ6N2UeE5znYcMJSB47ze1xJvjhtj9b99YcmOUkvVN5y\n
aiPD\n\nnhZMKDthaIo3A2hjoMFkgFa0tzqV1h7C/lEtsQrShSlyWCqHL7oT/1ivdPCGIKQ1g\n
\nnYHLigdjpgpedsoMLWZlQMAZIqF4XcEZGEGcgjhi6sCgYAXnkCTPLsXvtpkTcDH3v7U\n
\nn+ZDnNv7pdGwG2Y2m65eCQVZ3ZiJyYgk4XUILWu4/D3KnjnOD0xcv1AhudSiaVnMzs\n
\nnTcjK+fS15

```



FORMACIÓN 100% PRÁCTICA EN CIBERSEGURIDAD

Estudia ahora y **paga al encontrar trabajo**. Accede al mundo laboral con nuestra formación de 6 meses en ciberseguridad



¡Transforma tu futuro en ciberseguridad!
Escanea el QR para descubrir cómo empezar



Seguridad de la Información



Comparte estos flyers en tu clase y consigue más dinero y recompensas



Banco de apuntes de la

WUOLAH

1

Imprime esta hoja

2

Recorta por la mitad

3

Coloca en un lugar visible para que tus compis puedan escanar y acceder a apuntes

4

Llévate dinero por cada descarga de los documentos descargados a través de tu QR



```

kn7WM++Uu2Jh8U8tYrlomSLZlqCEdjJTXTr2u5o6nu09BdY5R7jM3hJ\nMZkTMJUqnMQBr5
Wq3/4FMQKBgHqulg/MpCZxk+VS70ILJtFuQoV07INszPC5vSHx\nan3wAHRgcncXmh5QKz5
wdX+j6hcnd3pwzx7X5v8MPeQyORQ2dmBmmVVvXNNk+yBc\n2CsqVoBDrkjURCgQsSwA8R8V
MeeTvf/awAfJCW2TqHVAKK9SnMi+gVQlmFHQdA0A\nLmFtAoGBALN6kyvF+fPP/b698Q9N5
be+X43eIeROreLSf03L2g66mXq9xdHBj/4w\nylkMTlUbVv2dzBRfeL8Vj8/760Vi9Sngbh
C+xuvRsVbv/8u1UPcPajhhHLVtD760\nwn6ji09dDmUhh6ADfGtwhD3mOjtHZA4I8peUbA6
DXYkQF206Yzkr\n-----END RSA PRIVATE KEY-----', passphrase="password")
public = RSA.import_key(b'-----BEGIN PUBLIC
KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAniD6Hta5ks4B3fyzf
FPD\n5DdoC0908MGC6HcaeQIA+CsQrao9L4VOQlZuAuD2+HuHtznLTxp+Svq7g3T+k88D7\n
JTdlKpaP254rrMS/pJy4wCCNwVksLqeKin6Onh9ybQFuSPURXSj3+V02qC7I1rD3\nnzoBIO
QhJTVEU3pKRgOAJz4VQAmBhmIcIgIKrFbtDhiI9wf7GgIq5Kd8ajRbfTi7Y\nno9gYaiwpEU
YPLB2P57dKgDh8xEw4WRKMfjhvgHn7eJ9AW4/iTYi/GdEvqyINoSsW\n9U0mxkRh1BtwUAK
nmMyhm71MNJwElkFEce+xd8QVDZxZ/M3715GrTS3lNmboJP0Z\nWwIDAQAB\n-----END
PUBLIC KEY-----')
cipherRSA =
b'm\xdfsuZ\x02\x9cb\xaf\r\xfe\xbc\x9a8\x0e\xac\xe2\x10H$\xea\xc5\x0b\xd
5\x1d-\xa3\x82?d'\xf0\xfd^\xe4^8\x80\xed\xfd8\x9bd\x1e\xa0\x13p\xe8|CG)
8\x90ms\x9aZ\x0c\x13s\xc1\xd0\xa4\xflp\xb7\xdc\x8fQ\xb9\x11\xfb\xa7\x8a
\x98\xfe\xdeFt\x95)O\x89\xf2\xd7\x9f\xa9\xf0\xf1\x9c\x8b\x9f\x10\xc51\x
aa\x1a1\xd8\x06\x96\xbb\xf6\x1cy\xe6\x0e\x12\xb7\xe2\xa1\xc2\x08\x92\xb
d\xf0B\x02\xed]\xa5[\f\xfe6X#t^T\xaaep2\xfc\xaa[\x92\xc4\x90\xe8)k?t?\x
11\xcc\x8e*\xf1\xc4\xe0|\x042\xd5\x00\x95\xcc\x9\x85+`JY\xc3t\xbe\xdd\
xe6\x14[\xc0;\xc2\x0bM\xcdB\x08\x95H[\xf8\xc7\x0f\x1b\x0f0r&\xea\xfa\x14
v\xde\xd0
\xde\xc4Co\xd8\xceH\xb3a!*x04\xf8{\xb0\xd9\x86\x90\x90>\x8f\x1a\xf2)m\
xabf\xb7\xdeA\x87$\xb0\xc91\x80\xa5\xa5\xcb\xef\xb9\x1a\xdd\x02\x9bL\x0
3&a\xcf\xfb.\xc7\xc9#\xe8~+"
signatureRSA =
b'\x0b\xe7'\xeb\xdd+\x80F\xf0\x13^\x18M\xc5\x13v\x9aUa\xd5\xea\x9e\x16\
x01\x08`;#G\x82\xcd\xbf\x86|%=xddY\x06\xd2[q\xba\x842\x8ak\x1f\xeb1\xe
3(6bk"\xa5-\xe2\x05QU\t#\x83;
\x98\x97K\x17\xc4\x89*\xa6Pj\xf2sG]\xd5\x95/\xc3\xa0\x99\xd7\x0e\x15\xd
9\x8f/\xff\t\xf9\x9bw\xa2!%\xed\xa9y\t\xae\xc4\xfc\xb6\xe5\xa9\xf7\x13\
xa8\xcd\xf2\xb0\xca\xeci\x82c\xf1\x95ZP\xc5\x9d\xe3\x18\xff\xa9K\x93\xc
7^b30\xce\x9ab\xc7+h\x88\\\xe1\x05\x1fHv\x14\x01\xf0_6\xeb2b\xcdyd0~)U\
xa8X\xf8\xf2\xfc\xe8"\x81x\x81\xd1\xa9\xe7~0\xc1
WL\x92\xef\x1d\xe7\nf\x81oq\'Z\xfb\x8b\x0e\xb7\x13\xc8\xf3\xa0
a\x1b\xbe\x8f\x9brB\x060\x80@j\x1b\x1b\xd2\x17\xe6CN\xc8\xfej\xb6\xe7\x
bf\x91\xcd\x08\x8d\x0f\xa6\xe8\xe6\xfd`3YQ#\xabL\x08\x1b/\xf9\xf8\x1a\
xe3\xd0'

```

A resolver por el alumno



```
engine_descifrado = PKCS1_OAEP.new(private)
datos = engine_descifrado.decrypt(cipherRSA)
h = SHA256.new(datos)
texto = datos.decode("utf-8")
verifier = pss.new(public)

try:
    verifier.verify(h, signatureRSA)
    print(texto)
except (ValueError, TypeError):
    print("no se pudo verificar")

#####
#####
print("#=====")
print("# PREGUNTA 3 (2,5 puntos)")
print("#=====")
# Enunciado
#-----
# Implementar el siguiente protocolo:
#
# - A: Mostrar por pantalla M
# - A: K_AT(M) = Cifrar_K_AT(M)
# - A->T: K_AT(M)
# - T: M = Descifrar_K_AT(K_AT(M))
# - T: K_BT(M) = Cifrar_K_BT(M)
# - T->B: K_BT(M)
# - B: M = Descifrar_K_BT(K_BT(M))
# - B: Mostrar por pantalla M
#
# (cuya version simplificada siguiendo el formato de las transparencias
es:
# - A->T: K_AT(M)
# - T->B: K_BT(M)
# )
#
# utilizando las claves DES "K_AT" y "K_BT", y el texto en claro "M"
mostrado en el codigo fuente.
# La clave K_AT se comparte entre A y T, y la clave K_BT se comparte
entre B y T.
# El mensaje se cifra y descifra con el modo de operacion ECB de DES.
# El codigo para "transmitir" el mensaje se proporcionara en el codigo
fuente de la practica.
```

```

# Ejercicio
#-----

M = "Este es el texto del tercer apartado (12/11/2018), formato string
(cadena)"
K_AT = b'01234567'
K_BT = b'76543210'
BLOCK_SIZE_8 = 8

# A: Mostrar por pantalla M #
print(M)

# A ##### A realizar por el alumno ##
des_cipher = DES.new(K_AT, DES.MODE_ECB)
cifrado_K_AT = des_cipher.encrypt(pad(M.encode("utf-8"), BLOCK_SIZE_8))
#print("cifrado_k_AT")
#print(cifrado_K_AT)

# A-> T ##### No cambiar este codigo ####
M = None
# A envia: cifrado_K_AT
file_out = open("a_t.bin", "wb")
file_out.write(cifrado_K_AT)
file_out.close()
cifrado_K_AT = None
# T recibe: msg_K_AT
file_in = open("a_t.bin", "rb")
msg_K_AT = file_in.read()
file_in.close()

# T ##### A realizar por el alumno ##
des_decipher = DES.new(K_AT, DES.MODE_ECB)
texto = unpad(des_decipher.decrypt(msg_K_AT), BLOCK_SIZE_8)

des_cipher = DES.new(K_BT, DES.MODE_ECB)
cifrado_K_BT = des_cipher.encrypt(pad(texto, BLOCK_SIZE_8))
#print("cifrado_K_BT")
#print(cifrado_K_BT)

# T -> B ##### No cambiar este codigo ##
descifrado_K_AT = None
# T envia: cifrado_K_BT

```



```

file_out = open("t_b.bin", "wb")
file_out.write(cifrado_K_BT)
file_out.close()
cifrado_K_BT = None
# B recibe: msg_K_BT
file_in = open("t_b.bin", "rb")
msg_K_BT = file_in.read()
file_in.close()

# B ##### A realizar por el alumno ##
des_decipher = DES.new(K_BT, DES.MODE_ECB)
mensaje = unpad(des_decipher.decrypt(msg_K_BT),
BLOCK_SIZE_8).decode("utf-8")
print("Mensaje recibido")
print(mensaje)

#####
#####
print("#=====")
print("# PREGUNTA 4 (2,5 puntos)")
print("#=====")
# Enunciado
#-----
# Implementar el siguiente protocolo:
# - A: Mostrar por pantalla M
# - A: Cipher_M = Cifrado_Publica_B(M)
# - A: HMAC_M = HMAC(M, SHA256, K_AB)
# - A->B: Cipher_M, HMAC_M
# - B: Texto_M = Descifrado_Privada_B(Cipher_M)
# - B: Si HMAC_M se verifica con HMAC(Texto_M, SHA256, K_AB)
# - Mostrar por pantalla Texto_M
#
# (cuya version simplificada siguiendo el formato de las transparencias
es:
# - A->B: Cifrado_Publica_B(M), HMAC_SHA256(M,K_AB)
# )
#
# utilizando las variables "private" (clave privada RSA de B), "public"
(clave publica RSA de B)
# , y "K_AB" (una clave compartida entre A y B).
# Para el cifrado se utilizara PKCS1_OAEP, y para el HMAC se utilizara
SHA256.

```



X
SIBUYA
URBANSUSHIBAR

Hay rollos y rollos... Me vas a comparar tu rollo de ayer
con un Hot Roll, claramente el sushi está más bueno

HAZ CLIC PARA HACER TU

RESERVA



Ejercicio

#-----

```
private = RSA.import_key(b'-----BEGIN RSA PRIVATE
KEY-----\nMIIEowIBAAKCAQEAniD6Hta5ks4B3fyzfFPD5DdoC0908MGC6HcaeqiA+CsQr
ao9\nL4VOQlZaUD2+HuHtznLTxp+Svq7g3T+k88D7JTD1KpaP254rrMS/pJy4wCCNwVks\n
LqeKin6Onh9ybQFuSPURXSj3+V02qC7IlrD3zoBIOQhJTVeU3pKRGOAJz4VQAmBh\nnmicIg
IKrfBtDhiI9wf7GgIq5Kd8ajRbfTi7Yo9gYaiwpeUYPLB2P57dKgDh8xEw4\nnWRKMfjhvgH
n7eJ9AW4/iTYi/GdEvqyINoSsW9U0mxkRh1BtwUAKnmMyhm71MNJwE\nnlkFEce+xd8QVDZx
Z/M3715Grts31NmboJP0ZWwIDAQABAOIBAEDyb3jaHbdHyKmK\nnAJhIeVVTUncOuHAXMvLS
9Hu7mNkVKxEBModBm96S5Q7nQR7DED7w95LOPMH35uDI\nnorIBKcXj7Mo0s9pysSKRXts4C
YPT+xUWUJjK9JKkn2Qfq2pNI6Rwj5SxXoQ7vla+\nfGG0Rtu4gbF3D1Bmb/0ouv1xR2ZFIE
ggks1CMJWt+DiV8RYQ/9+BOFafPtISYyGM\nnwK7qo7n88N/SfIK3AiQZbPg6vJIEkfj8yYt
b9DaO14phXevRaoixrbCptDUPIK4m\nn5C9G9tyjEZRaRCD/kng41j4YMWb+8dBO/o06th1h
KKEe75FfOCWW96QhLU0JmmfW\nnwmRspXECgYEAX5vXOraKk0XJ9hoyZ73asavFHY6LpLiyU
DLb6XQ2lrZj7B8elJz2\nnfKeMFIduyrA/HNDx5h1louj9DU5lamwvWth6dPz+YREMTGo2tA
hLf4qQa8e8qt1y\nngY/crl7N110q5C3WdIXL2+HRIFrMQy0aM/XF7UUjqcFLVQ70MH9eRE
CgYEAys02\nngJ4zInq6CUPRE2ZgjQ6N2UeE5znYcMJSB47ze1xJvjhtj9b99YcMOukvVN5y
aiPD\nnhZMKDthaIo3A2hjoMFkgFa0tzqV1H7C/1EtsQrShSlyWCqHL7oT/1ivdPCGIKQ1g\
nYHLigdjgpedsoMLWZ1QMAZIQF4XcEZGECgjh16sCgYAXnkCTPLsXvtpkTcDH3v7U\nn+ZDn
Nv7pdGwG2Y2m65eCQVZ3ZiYjyGk4XUILWu4/D3KnjnOD0xcv1AhudSiaVnMzs\nnTcjK+fs15
kn7WM++Uu2Jh8U8tYr1omSLZ1qCEdjTjTXTr2u5o6nuO9BdY5R7jM3hJ\nnMZkTMJUqnMQBr5
Wq3/4FMQKBgHqu1g/MpCZxk+VS70ILJtFuQoV07INSzPC5vSHx\nnan3wAHRgcncXmh5QKz5
wdX+j6hcnD3pwzx7X5v8MPEQyORQ2dmBmmVVvXNNk+yBc\nn2CsQVoBDrkjURCgQsSwA8R8V
MeeTvf/awAfJCW2TqHVAKK9SnMi+gVQ1mFhQda0A\nnLmFtAoGBALN6kyvF+fPP/b698Q9N5
be+X43e1ERoreLSf03L2g66mXq9xdHBj/4w\nnylKMT1UbVv2dzBRfeL8Vj8/760Vi9Sngbh
C+xuvRsVbv/8u1UPcPajhhHLVtD760\nnwn6ji09dDmUhh6ADfGtwhD3mOjthZA4I8peUbA6
DXYkQF206Yzkr\nn-----END RSA PRIVATE KEY-----', passphrase="password")
public = RSA.import_key(b'-----BEGIN PUBLIC
KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAniD6Hta5ks4B3fyzf
FPD\nn5DdoC0908MGC6HcaeqiA+CsQrao9L4VOQlZaUD2+HuHtznLTxp+Svq7g3T+k88D7\nn
JTD1KpaP254rrMS/pJy4wCCNwVksLqeKin6Onh9ybQFuSPURXSj3+V02qC7IlrD3\nnzO BIO
QhJTVeU3pKRGOAJz4VQAmBhmicIgIKrfBtDhiI9wf7GgIq5Kd8ajRbfTi7Y\nno9gYaiwpeU
YPLB2P57dKgDh8xEw4WRKMfjhvgHn7eJ9AW4/iTYi/GdEvqyINoSsW\nn9U0mxkRh1BtwUAK
nmMyhm71MNJwElkFEce+xd8QVDZxZ/M3715Grts31NmboJP0Z\nnWwIDAQAB\nn-----END
PUBLIC KEY-----')
M = "Texto del cuarto apartado, cifrado RSA y HMAC (12/11/2018),
formato String (cadena)"
K_AB = b'01234567890ABCDEF'
BLOCK_SIZE_8 = 16

# A: Mostrar por pantalla M #
print(M)
```

WUOLAH

```

# A ##### A realizar por el alumno ##
cipher = PKCS1_OAEP.new(public)
Cipher_M = cipher.encrypt(M.encode("utf-8"))
#print("Cipher_M")
#print(Cipher_M)

h = HMAC.new(K_AB, digestmod=SHA256)
h.update(M.encode("utf-8"))
HMAC_M = h.digest()
#print("HMAC_M")
#print(HMAC_M)

# A-> B ##### No cambiar este codigo #####
file_out = open("encrypted.bin", "wb")
[ file_out.write(x) for x in (HMAC_M, Cipher_M) ]
file_out.close()
HMAC_M = None
Cipher_M = None
file_in = open("encrypted.bin", "rb")
msg_HMAC_M, msg_Cipher_M = [ file_in.read(x) for x in (32, -1) ]

# B ##### A realizar por el alumno ##
cipher = PKCS1_OAEP.new(private)
Texto_M = cipher.decrypt(msg_Cipher_M).decode("utf-8")
#print("Texto_M")
#print(Texto_M)

try:
    h.verify(msg_HMAC_M)
    print("mensaje enviado")
    print(Texto_M)
except ValueError:
    print("no se verifica")

```



FORMACIÓN 100% PRÁCTICA EN CIBERSEGURIDAD

Estudia ahora y **paga al encontrar trabajo**. Accede al mundo laboral con nuestra formación de 6 meses en ciberseguridad



¡Transforma tu futuro en ciberseguridad!
Escanea el QR para descubrir cómo empezar

