David Martí Felip (1633953) Arnau Busquets Domingo (15533455)

Laboratori 3
–
Certificats

Criptografia i seguretat

ÍNDEX

Part 1: Obtenció d'un certificat legítim

Per a poder crear el nostre certificat legítim hem seguit els passos següents:

Primerament, hem executat el codi de l'arxiu sign_message.py que ens crea un parell de claus, publica i privada, a la vegada que utilitza la clau privada per xifrar el hash d'una cadena de caràcters que li demanem que sigui "#CHANGE GRADE TO 10#". D'aquesta manera ja hem xifrat la cadena de caràcters que haurem d'enviar amb la clau privada corresponent a la clau pública que contindrà el certificat il·legítim que crearem.

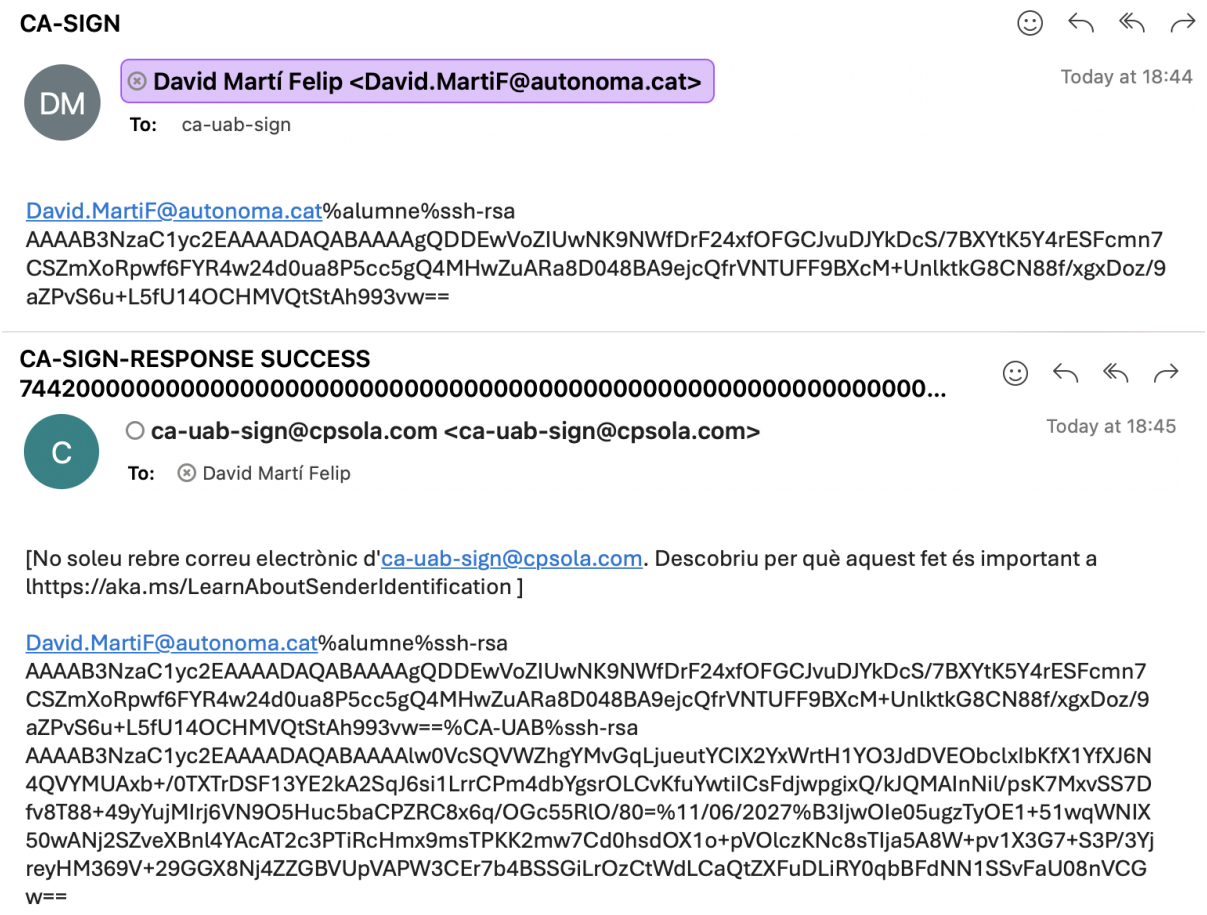
```python
# Generate a RSA key
student_key = RSA.generate(RSA_KEY_SIZE)
print(student_key.export_key().decode('utf-8'))

# Export the public key
student_pub_key = student_key.publickey()
print(student_pub_key.export_key('OpenSSH').decode('utf-8'))


# Hash the message to sign
str_to_sign = "#CHANGE GRADE TO 10#"
h = PrunedSHA256Hash().new(str_to_sign.encode('utf-8'))

# Sign the hash of the message and get b64 encoded signature
signer = PKCS1_v1_5.new(student_key)
signature = b64encode(signer.sign(h)).decode('utf-8')

# Print results
print(f"\tString to sign:\t\t\t\t\t\t{str_to_sign}")
print(f"\tHash of the string (SHA256 pruned):\t{h.hexdigest()}")
print(f"\tSignature:\t\t\t\t\t\t{signature}")

print(f"Public key: {student_key.publickey().export_key(format = 'OpenSSH').decode('utf-8')}")
print(f"Private key: {student_key.export_key().decode('utf-8')}")
```

A continuació, vegem les dues claus juntament amb la signatura de l'string.

```
(.venv) davidmarti@davids-MacBook-Air Criptografia-AES % "/Users/davidmarti/Library/CloudStorage/OneDrive-UAB/3r curs/se
B/3r curs/semestre 2/Criptografia/labs/Criptografia-AES/certificats/sign_message.py"
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQDCDqUezfYL9n7zPg6j98nozhW7HilmrEWJxPQz4nU8jllL2m1k
4RsGOJMBEErBXzm+KvgbFUVg5Zn0esOxBO9aUMngOzuanQYaHNcjTLXJE14mlcYc
nSUwOP9j801Nr4aZ6dTmq2qs2XntYicumYHdrYyOHgKdz7ZPcgJEBQrgjwIDAQAB
AoGAEHS4eN/DVSleg5cIG8x38UeEnT0GCsBiFzg98yZphayWkvGfZDYYmhLzGFx+
0l4WYfN33agUnY5q20oLzTebalm3J4Iusev8b1YdjlZYR/4gezE0ThCBAX/K/aNO
KZuIzrPZivSTEcx0bjTq1Cgd92Or1tD+q2DDvXHAgjkHN+UCQQDHxzinMDcIBuC/
Rh8DCzp87d6fR7EndTljShCpa1DmR9K9NLXEPoEeWDz79klaNi0XncQqzqO/ZjGv
cjPMT6BbAkEA+KtC2BgwoIUAc4VXe9MIFdo8gegCQjUuSC1wqel/94sYTwBKDGn+
kHFfQ8GxopqNBJ0eQSqzZ58hE7TYtoL23QJAXzh4q0+S21OT6fc9LtE6RnOGSbr9
GEaKXOXffTD2dVxEeQRXorr8Bf+eyaWBHut1oK0E5DZr2/2UrwAq6jYFwwJBANo9
C/w1NrAaKiw4kW/l5Mmw77ufECRlIgpMfv1QJXPuilYDTfmZKMZCCbagjrLDaxHq
hrvwquU9Xj06UZvUUq0CQFyAIeQuLSxU0bNcy6Mev/P6c8Hg9FdOHvv9qR8XMNk/
b5I3lbOg65ZNS8tUeJEfBI5DPQBCTZqTPnafi4Nl6gQ=
-----END RSA PRIVATE KEY-----
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDCDqUezfYL9n7zPg6j98nozhW7HilmrEWJxPQz4nU8jllL2m1k4RsGOJMBEErBXzm+KvgbFUVg5Zn0esOx
        String to sign:                                         #CHANGE GRADE TO 10#
        Hash of the string (SHA256 pruned):     76ba000000000000000000000000000000000000000000000000000000000000
        Signature:                                              Z5RcQbTA9zgwhMUz9Hhuq1PHIGJ/d6Pwi+gheLW6FuvmNa/C
GoY/WhNtXc=
Public key: ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDCDqUezfYL9n7zPg6j98nozhW7HilmrEWJxPQz4nU8jllL2m1k4RsGOJMBEErBXzm+Kvgb
```

Per a l'obtenció del nostre certificat lícit farem un correu a amb destinació ca-uab-sign@eines.uab.es, amb tema CA-SIGN i una línia amb tots els camps sol·licitats:

- Correu
- Professio
- Clau pública



Com podem observar, la CA ha respost el nostre correu amb èxit, i ens retorna el certificat complert, amb el nom de la CA, la seva clau pública, la data d'expedició i la signatura annexats.

Ara ja tenim un certificat legítim.

Part 2: Tripijocs amb un certificat il·legítim

En aquest pas hem de aconseguir un certificat signat per la CA-UAB que tingui com a identitat el correu d'un dels professors de l'assignatura. Però és evident que no podem demanar-lo a la CA, ja que aquesta comprovaria que l'adreça de correu electrònic coincideixi i denegaria el certificat.

La nostra estratègia consistirà en trobar un certificat amb el nostre nom amb el que obtinguem el mateix hash que amb el certificat del professor.

Seguirem la següent estructura, sempre usant "%" com a delimitador entre camps:

Certificat del professor: "carlos.borrego@uab.cat" + "profesor" + clau pública del nostre certificat + "CA-UAB" + clau pública CA + data d'expiració certificat modificat

Una vegada construït el certificat del professor buscarem un valor aleatori que poguem substituir en el segon paràmetre del nostre certificat per tal de crear un nou certificat amb el nostre correu però el mateix hash.

Ho farem amb el següent codi:

```
certificatprofe = "carlos.borrego@uab.cat%profesor%ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDBh03TNOYQzyE6FgGn6gIdh1WBUsXax2kz57KvM7jo
clau_pubCA = "%CA-UAB%ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAlw0VcSQVWZhgYMvGqLjueutYCIX2YxWrtH1YO3JdDVEObclxIbKfX1YfXJ6N4QVYMUAxb+/0T
data_expedicio = "%11/06/2027"
certificatdavid2= "David.MartiF@autonoma.cat%alumne%ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDCDqUezfYL9n7zPg6j98nozhW7HilmrEWJxPQz4nU

from base64 import b64encode, b64decode
from sign_message import PrunedSHA256Hash, PRUNED_HASH_SIZE, RSA_KEY_SIZE


# Hash the message to sign

h = PrunedSHA256Hash().new(str(certificatprofe+clau_pubCA+data_expedicio).encode('utf-8'))
print(h.hexdigest())

# Try to get a certificate with my name with the same hash

for i in range(100000000):
    prehash = 'David.MartiF@autonoma.cat%'
    posthash = '%ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDCDqUezfYL9n7zPg6j98nozhW7HilmrEWJxPQz4nU8jllL2m1k4RsGOJMBEErBXzm+KvgbFUVg5Z
    hash = PrunedSHA256Hash().new((prehash + str(i) + posthash).encode('utf-8'))
    if hash.hexdigest() == h.hexdigest():
        print("i: ", i)
        print("hash:", hash.hexdigest())
        print("certificat:", prehash + str(i) + posthash)
        break
```

I obtenim :

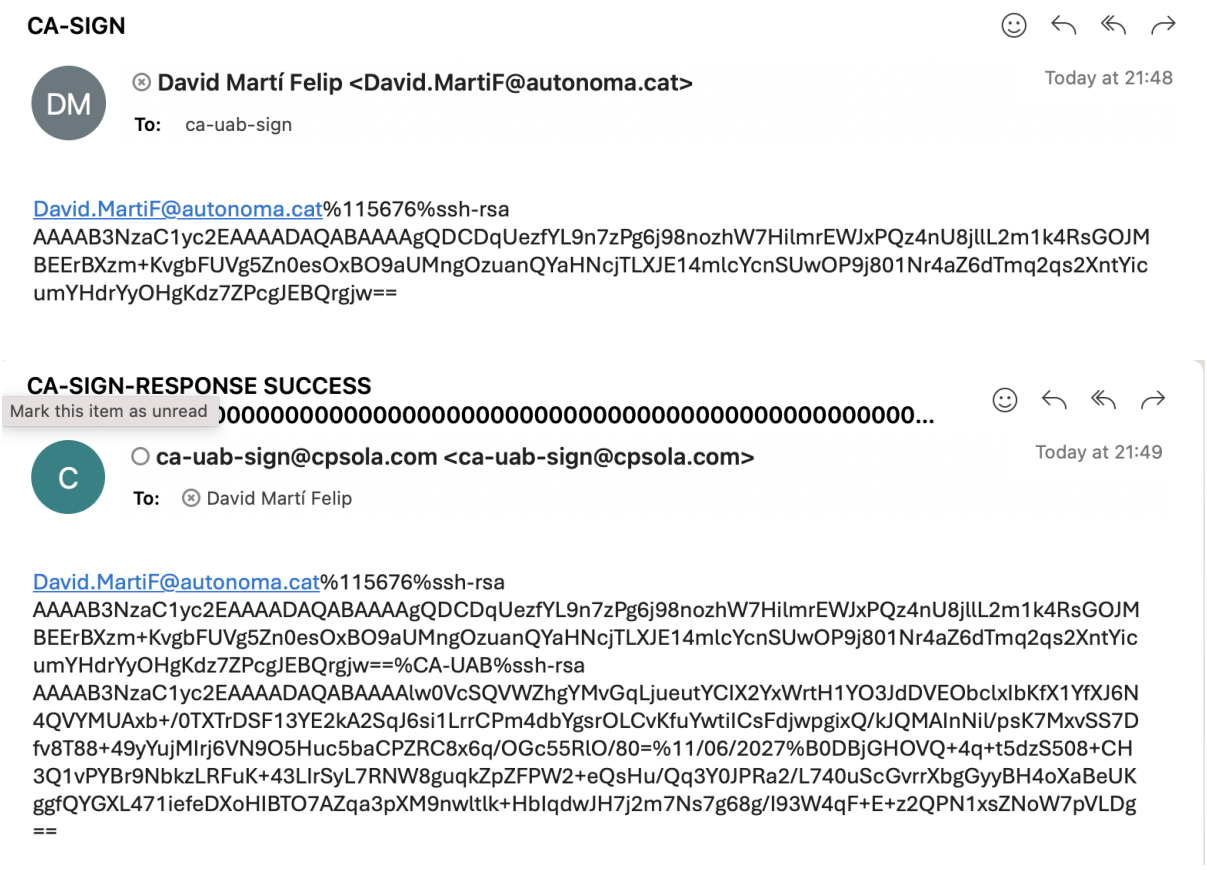
```
(.venv) davidmarti@davids-MacBook-Air Criptografia-AES % "/Users/davidmarti/Libr
B/3r curs/semestre 2/Criptografia/labs/Criptografia-AES/certificats/lab3.py"
c8410000000000000000000000000000000000000000000000000000000000000000
i:  115676
hash: c8410000000000000000000000000000000000000000000000000000000000000000
certificat: David.MartiF@autonoma.cat%115676%ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAA
rYyOHgKdz7ZPcgJEBQrgjw==%CA-UAB%ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAlw0VcSQVWZhg
jMIrj6VN9O5Huc5baCPZRC8x6q/OGc55RlO/80=%11/06/2027
```

Per tant canviarem el camp de professió per el valor “115676”.

Així ens quedarà el certificat modificat:

Certificat modificat : David.MartiF@autonoma.cat + valor trobat(“115676”) + clau pública del nostre certificat + “CA-UAB” + clau publica CA + data d’expiració

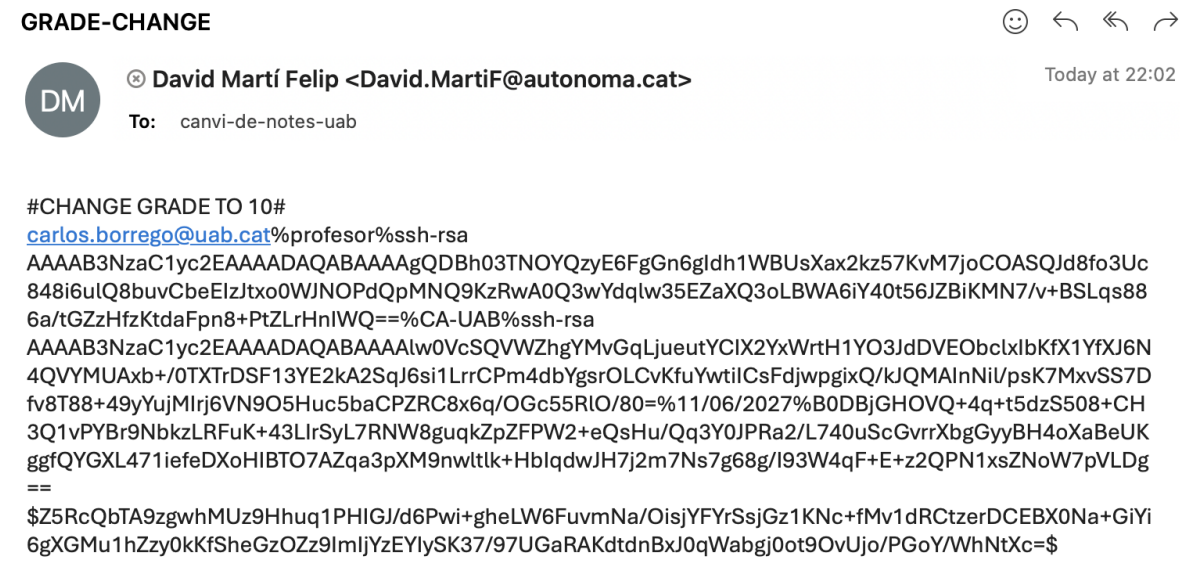
Ara si, enviarem aquest certificat de nou a la CA per a que ens el firmi, ja que aquest si que conté el nostre correu. Rebrem un certificat firmat del que només en volem la firma:



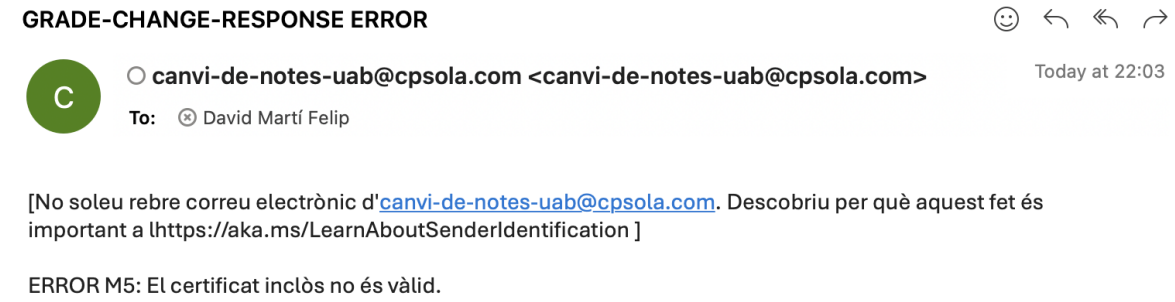
Ara usem la firma rebuda per afegir-la al certificat creat del professor

Certificat del professor: “carlos.borrego@uab.cat” + “professor” + clau pública del nostre certificat + “CA-UAB” + clau pública CA + data d’expiració + Signatura del certificat modificat

Aquest certificat l'enviem a l'adreça canvi-de-notes-uab@eines.uab.es amb l'string "#CHANGE TO GRADE 10#" a la primera línia, el certificat a la segona i la cadena del principi xifrada amb la nostra clau privada al final. D'assumpte del correu posem "GRADE-CHANGE".



Després de repetir diverses vegades el procés complert i provar canvis en el codi, escrivint a mà part dels correus i canviant els encoders al script python, sempre acabem rebent la resposta següent:



Així doncs, hem decidit deixar el nostre procediment durant la pràctica detallat en aquest document per a que quedi reflectit el nostre coneixement dels passos a seguir i quin ha estat el nostre camí.

Part 3: Dels certificats a la PKI

Els certifcats X.509 són un estàndard que es fa servir en la majoria de protocols que requereixen certicats digitals. Investigueu quins són els camps bàsics que conté un certifcat X.509 i indiqueu quins canvis caldria fer als certifcats d'aquesta pràctica per tal d'adaptar-los a l'estàndard.

Un certificat X.509 conté els següents camps:

- Version: *Indica la verió de standard*
- Serial number: *és un nombre únic (com un ID) assignat al certificat per la CA*
- Signature Algorithm: *és l'algoritme que s'utilitza per a firmar el certificat*
- Issuer: *entitat de la CA ("CA-UAB" en el nostre cas)*
- Validity: *No només tenim data de caducitat sinó que també una data d'inici de validesa*.
    - Not before
    - Not After
- Subject: *Identitat del propietari (en el nostre cas el correu)*
- Subject Public Key Info
    - Public Key Algorithm: *Algoritme de clau pública (p.e. RSA)*
    - Subject Public Key: *Clau pública del propietari*
- Extensions (optional)
- Signature

Per a adaptar el nostre certificat als standards X.509 caldria afegir-hi els camps de **Version, Serial Number, Signature Algorithm,** la data **Not before, Public key algorithm** i un espai per **extensions.** A més d'odenar-lo tal i com ho fa l'Standard.