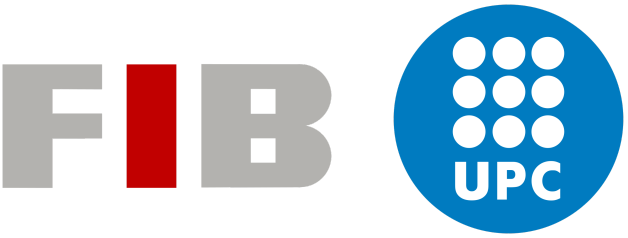
Criptografía 23-24 - Q1: Práctica 4 - ECC y certificados digitales

Miguel Moreno e Iván Risueño



Índice

1. Ejercicio 1

Capturad una conexión TLS 1.3 con www.wikipedia.org que use un certificado con una clave pública EC (Elliptic Curve).

a) **Comprobad que el número de puntos (orden) de la curva usada en el certificado es primo.**

In [16]:
```
# Ejercicio 1a): Comprobad que el número de puntos (orden) de la curva usada en el certificado es primo.

# Parámetros de la curva p-256(https://neuromancer.sk/std/nist/P-256#)
p = 0xffffffff00000001000000000000000000000000ffffffffffffffffffffffff
n = 0xffffffff00000000ffffffffffffffffbce6faada7179e84f3b9cac2fc632551
a = 0xffffffff00000001000000000000000000000000fffffffffffffffffffffffc
b = 0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b


E = EllipticCurve(Zmod(p),[a,b])
E.cardinality().is_prime()
```

Out[16]: True

b) **Comprobad que la clave pública P de www.wikipedia.org es realmente un punto de la curva.**

In [17]:
```
# Ejercicio 1b): Comprobad que la clave pública P de www.wikipedia.org es realmente un punto de la curva.

# Podemos comprobar si un punto P pertenece a la curva calculando su orden. Si éste
# existe, podemos afirmar que el punto pertenece a la curva (cálculo en la siguiente celda)
```

c) **Calculad el orden del punto P.**

In [18]:
```
# Ejercicio 1c): Calculad el orden del punto P.

# Este punto corresponde a la clave pública (Qx, Qy) obtenida en la captura de wireshark
Px = 0x3561f4211aff6ac43bfa0647c6196ebe7038f1dc16b1bc381412d4142b1c0b31
Py = 0x8159f567f6e72ad13c1efaaea7ed065dd66f5d894c6bc8b0e00f83cff5d38ada
P = E([Px,Py])
q = P.order()
q
```

Out[18]: 115792089210356248762697446949407573529996955224135760342422259061068512044369

d) **Comprobad que la firma ECDSA es correcta.**

In [19]:
```
# Ejercicio 1d): Comprobad que la firma ECDSA es correcta.

# Valores obtenidos de la captura de wireshark
f1 = 0x00a3eb0caf2ac3852d527034db8493ea2418c2c62a32606229f82d22c2e68db13a
f2 = 0x00f4e1203fff6a2cf02c4e65ccee949e3f01e705c0c616d86dae579135a8ec7f33

# Mensaje obtenido a partir del script en python proporcionado
m = 0x609e4c774f4cc24d3a787823904df5505977cb4e2da693e75becd0d4f2c84f24

# Punto obtenido en el documento NIST
x1 = 0x6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296
y1 = 0x4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5
P2 = E([x1,y1])

w1 = mod(m*f2^-1,q)
w2 = mod(f1*f2^-1,q)

x0y0 = Integer(w1)*P2+P*Integer(w2)
mod(x0y0[0],q) == f1
```

Out[19]: True

2. Ejercicio 2

Conectaros con www.fib.upc.edu. En esta conexión os enviarán el certificado del servidor de la FIB.

a) **Obtened el periodo de validez del certificado y la clave pública (módulo y exponente, en base 10) del web de la FIB. ¿Cuántos digitos tiene el módulo?**

La validez del certificado va desde el día 14/03/2023 a las 0:00 hasta el día 13/03/2024 a las 23:59.
El módulo y el exponente son los siguientes:

- Módulo(1233 dígitos):
  643353183200092094287771675345221470010077629570715127404791684658292676146258015596958199149307800248646412074806402713345405555081170237753105542354684208619132518842723761824119200367033102378675040596360890232050015927575092514853608500408281492709084825821114563373317007098338129291091372712149614623307473988341465977307145752705448273512348134128938358986609643424139880228535842593559221687695145242873591558628568048883643065876396362845755954465671178604922843812362868878247546278397207349196905507150170797294083377275301635094658499215617045568118385833417270628683712856537824875907530916265958057443067677151512997889742061364935925952606225668014307987049307339808449793109887341802097697797472132550044335014870194464476487381392865709669506030186544637694375179204618321954015650446276930911698051042892696086349759986571166856157686471982219042130923715005637691461222292252593992204681029950031304177950442270673330843986941368673704957836680903529694544395379801093854406668098582007713837797482211504933610635294907722785172663133317619642042809717727668127206553431567060606095326474493551870473353355219215124763132251033711614431811575092255558476963925483799431687150265952194102923507545349731288895302130 3

- Exponente: 65537

b) En el certificado encontraréis un enlace a la política de certificados (CPS) de la autoridad certificadora firmante. ¿Qué tipo de claves públicas y tamaños admite?

Claves RSA cuyo tamaño de módulo en bits sea divisible por 8, y de almenos 2048 bits (los certificados de firma de código para usuarios finales tendrán almenos 3072 bits). Además, las claves ECDSA estarán en las curvas P-256 o P-384.
Fragmento del documento (*https://www.sectigo.com/uploads/files/Sectigo_CPS_v5_3_7.pdf*):

For Root CA Certificates' key sizes, see section 6.3.2

Code Signing certificate key sizes SHALL be governed by NIST key management guidelines.

Root CA certificates and any certificates which chain up to them have:

- RSA keys whose modulus size in bits is divisible by 8, and is at least 2048 bits
  - Code Signing certificates for end users will have at least 3072 bits
- ECDSA keys on the P-256 or P-384 curves.

c) En el certificado encontraréis un enlace un punto de distribución de la CRL de la autoridad certificadora. ¿Cuántos certificados revocados contiene la CRL?

Habiendo ejecutado el comando
openssl crl -inform DER -text -in ./GEANTOVRSACA4.crl | grep -c "Revocation Date:",
la salida nos da un total de 18233 certificados cuya fecha de revocación existe, es decir, un total de 18233 certificados.

d) En el certificado encontraréis la dirección OCSP (Online Certificate Status Protocol) a la que se puede preguntar por el estatus del certificado. ¿Cuál es el estatus del certificado y hasta cuándo es válido dicho estatus?

En este caso, el certificado emisor corresponde al archivo *certificado_GEANT_OV_RSA_CA.pem*, de la entidad incluida en el nombre. Ejecutando entonces el comando
openssl ocsp -issuer certificado_GEANT_OV_RSA_CA.pem -cert certificadoFIB.pem -url http://GEANT.ocsp.sectigo.com -text,

éste nos genera una salida donde podemos consultar el estado del certificado y la fecha de su validez. En nuestro caso, el estado del certificado es *good* y la fecha de validez(o de *Next Update*, según la salida del comando) es Dec 20 18:34:36 2023 GMT(una semana después de la fecha *This Update*.