CRIPTOGRAFIA MAT - FIB

Antonio Guilera Domingo

Lluís Marquès i Peñaranda

1. Enregistreu una connexió segura amb www.wikipedia.org que faci servir ECDHE-ECDSA.

(a) Doneu els noms de les corbes que es fan servir per acordar la clau DH i per la clau pública del servidor.

Curva clave pública: Named Curve: 1.2.840.10045.3.1.7 (secp256r1)

Diffie-Hellman: x25519 (0x001d)

(b) Comproveu que el nombre de punts (ordre) de la corba que es fa servir al certificat és primer.

```
#Params de la curva 256
p = 115792089210356248762697446949407573530086143415290314195533631308867097853951
n = 115792089210356248762697446949407573529996955224135760342422259061068512044369

#la a es -3
#la b es propia de la curva

a = -3
b = 0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b


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

```
115792089210356248762697446949407573529996955224135760342422259061068512044369
True
```

(c) Comproveu que la clau pública P de www.wikipedia.org, és realment un punt de la corba.

(d) Calculeu l'ordre del punt P.

Al hacer el cálculo del orden del punto ya estamos verificando que efectivamente

es un punto de la curva y su orden es el siguiente:

```
#Punto a partir de la clave publica
x = 0x23551f0b79e4822143c07d2b3f4570a67ed537197fe77ff14acdb3d220021724
y = 0x19b80ac6d1582b114cfafa89bab7ab1b8acaaeacbad7e3588fc94aa406279e9c
G = E([x,y])
G.order()
```

```
115792089210356248762697446949407573529996955224135760342422259061068512044369
```

si no fuese un punto de la curva G = E([x,y]) daría el siguiente error (hemos usado 123 como x y456 como y).

```
TypeError: Coordinates [123, 456, 1] do not define a point on Elliptic Curve defined by y^2 = x^3 +
115792089210356248762697446949407573530086143415290314195533631308867097853948*x + 41058363725152142129326129780047268409114441015993725554835256314039467401291
over Ring of integers modulo 115792089210356248762697446949407573530086143415290314195533631308867097853951
```

(e) Comproveu que la signatura ECDSA és correcta.

```
#Params de la curva 256
p = 115792089210356248762697446949407573530086143415290314195533631308867097853951
n = 115792089210356248762697446949407573529996955224135760342422259061068512044369

#la a es -3
#la b es propia de la curva

a = -3
b = 0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b


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

#Punto a partir de la clave publica
x = 0x23551f0b79e4822143c07d2b3f4570a67ed537197fe77ff14acdb3d220021724
y = 0x19b80ac6d1582b114cfafa89bab7ab1b8acaaeacbad7e3588fc94aa406279e9c
G = E([x,y])
#G.order()

#Verificacion de la signatura

F1 = 0x00880cddcd74b943a7ee9f1d774fb207160391881b7ffafe9c477f76094c29cd63
F2 = 0x363a9ef9afed040a05a373123a70f87456e4698c7bfebc9f834d60b2b34f4243

#m son los 256 primeros bits del sha384 de la concatenacion de los 6 binarios que hemos obtenido con wireshark
m = 0xB2A44C8DB04CEF601AEFE60F28BF4E7BD512D80E7E1EE060AA4BDCA617D321FF
mentera = 0xB2A44C8DB04CEF601AEFE60F28BF4E7BD512D80E7E1EE060AA4BDCA617D321FF7D76E01F7C5F6591D17854C9C0216938


#Punto q nos da el NIST

x1 = 0x6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296
y1 = 0x4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5
Punto = E([x1,y1])
#Punto.order()


w1 = mod(m*F2^-1,n)
w2 = mod(F1*F2^-1,n)

verificacion = Integer(w1)*Punto+G*Integer(w2)
mod(verificacion[0],n) == F1
```

```
True
```

2. Feu el mateix però amb una connexió amb google.com.

No es posible porqué la conexión con Google es mediante TLS 1.3 y los parámetros están encriptados.

3. Enregistreu una connexió segura amb www.fib.upc.edu. En aquesta connexió us faran arribar el certificat i el seu estatus.

(a) En el certificat es dóna un punt de distribució de la CRL de l'autoritat certificadora. Quants certificats revocats conté la CRL?

La CRL es TERENASSLCA3.crl y tiene 10158 certificados revocados

(b) Quin és l’estatus del certificat i fins quan és vàlid aquest estatus?

El estatus es válido hasta 18/11/2024