C - Entrega 4 (Corbes el·líptiques)

Jordi Armengol Estapé

Desembre/Gener 2018-2019

Aquesta és la documentació corresponent a l'entrega 4. Els *scripts* complets, en *Sage*, es troben a dins dels directoris dels corresponents exercicis (1, 2 i 3). Aquí n'inclourem fragments rellevants per les respostes, tallant o modificant lleugerament línies si és necessari per brevetat o per fer la resposta més clara. Els paquets i certificats extrets amb *Wireshark* (o el propi navegador, en algun cas) es troben també als directoris dels exercicis corresponents.

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

   - **Doneu els noms de les corbes que es fan servir per acordar la clau DH i per la clau pública del servidor.**
     La corba utilitzada per la clau pública del servidor és la P-256:

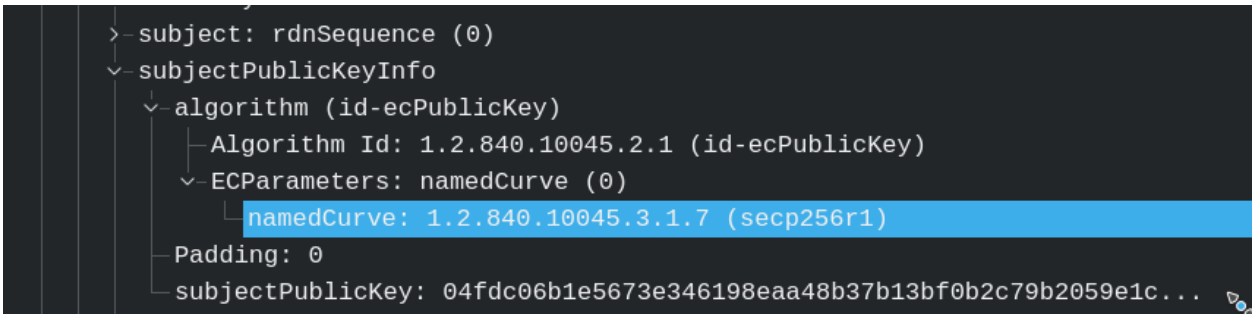

Figure 1: Nom de la corba per la clau pública del servidor.

La corba utilitzada per la clau pública de DH és la X25519 (Curve25519):

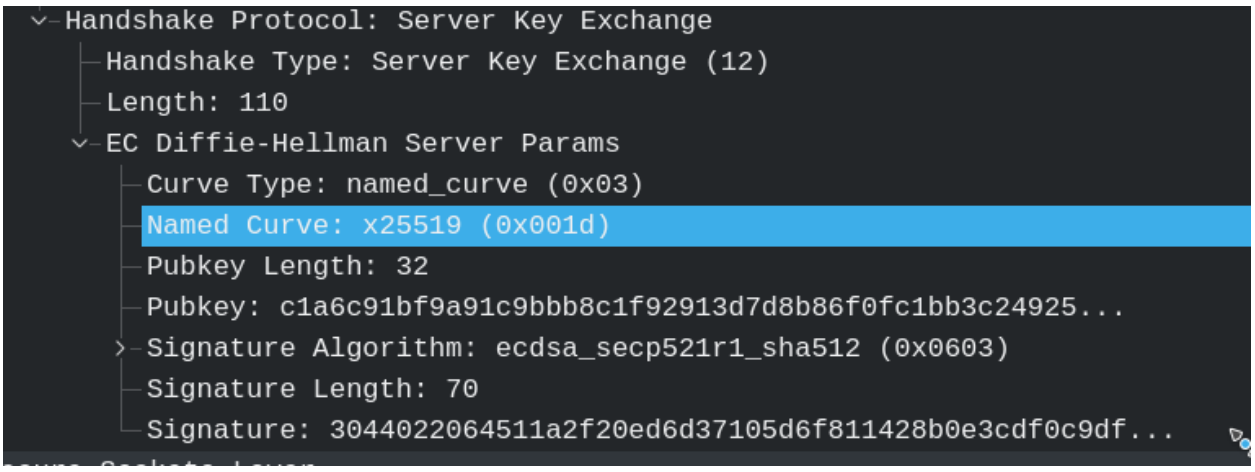

Figure 2: Nom de la corba per la clau pública de DH.

- **Comproveu que el nombre de punts (ordre) de la corba que es fa servir és primer.**
  Consultem la n de la P-256 a la documentació del NIST i comprovem que efectivament és primera:

```
sage: n = 115792089210356248762697446949407573529996955224135760342422259061068512044369
sage: print(n in Primes())
True
```

- **Comproveu que la clau pública P de www.wikipedia.org és realment un punt de la corba.**
  El punt P és 0xfdc06b1e5673e346198eaa48b37b13bf0b2c79b2059e1c707c97b7726da48b82,0x23d232de50231c3dba6d2b7178f5028bb0840de136553ae4e504ad6e7a6e87df (més la tercera coordenada que és 1). Utilitzant el codi d'exemple de l'enunciat podem comprovar que el punt P (que llegim del certificat de Wikipedia) compleix l'equació de la corba, és a dir, que la resta següent dóna 0 (la resta surt de l'equació):

```
sage: p = 115792089210356248762697446949407573530086143415290314195533631308867097853951
sage: n = 115792089210356248762697446949407573529996955224135760342422259061068512044369
sage: a = -3
sage: b = 0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b
sage: with open("raw/certificate_subjectPublicKey.bin", "rb") as f:
certificate = f.read()
sage: Px = int(certificate[1:((len(certificate) - 1)//2 + 1)].encode('hex'),16)
sage: Py = int(certificate[((len(certificate) - 1)//2 + 1):
           2*(((len(certificate) - 1)//2)+1)].encode('hex'),16)
sage: mod(Py**2,p)-mod(Px**3+a*Px+b,p)
0
```

- **Calculeu l'ordre del punt P.** Construïm la corba el·líptica i utilitzem el mètode de *Sage* corresponent per extreure'n l'ordre. Noti's que l'ordre obtingut és la mateixa n definida per la corba.

```
sage: Zp = Zmod(p)
sage: E = EllipticCurve(Zp,[a,b]);
sage: P = E([Px,Py])
sage: print(P.order())
115792089210356248762697446949407573529996955224135760342422259061068512044369
```

- **Comproveu que la signatura ECDSA és correcta.**
  Un cop extrets els camps necessaris de Wireshark (exportats com a *raw bytes* i disponibles als directoris de l'entrega als directoris /raw), he seguit els següents passos. Primer he llegit els bytes de la firma de DF i l'he extret a F1 i F2. El criteri que he seguit per assignar F1 i F2 ha estat que el camp seguia l'*Abstract Syntax Notation*, i als primers bytes i als del mig s'indicaven aspectes com ara la mida, el tipus,...

```
with open("raw/server_key_exchange_signature.bin", "rb") as f:
    signature = f.read()
f1 = int(signature[4:36].encode('hex'),16)
f2 = int(signature[38:70].encode('hex'),16)
```

  A continuació, he concatenat els respectius camps *Random* dels paquets de *Hello Client* i *Hello server* amb els camps *CurveType*, *NamedCurve*, *pubkeyLength* i *pubkey* de DF. N'hem fet el hash

512 (que era el que s'havia fet en el protocol com indicava Wireshark) i n'he agafat els 256 bits més significatius. Finalment ho he passat a enter.

```
with open("raw/client_hello_random.bin","rb") as f:
 m = f.read()
with open("raw/server_hello_random.bin","rb") as f:
 m += f.read()
with open("raw/server_key_exchange_curve_type.bin","rb") as f:
 m += f.read()
with open("raw/server_key_exchange_named_curve.bin","rb") as f:
 m += f.read()
with open("raw/server_key_exchange_pubkey_length.bin","rb") as f:
 m += f.read()
with open("raw/server_key_exchange_pubkey.bin","rb") as f:
 m += f.read()

hashed = hashlib.sha512(m).hexdigest()
hashed = hashed[:64]
hashed = int(hashed,16)
```

Un cop tenim les dues parts de la firma i el hash, he seguit el procediment indicat a les diapostives per verificar la signatura:

```
inv2 = sympy.invert(f2,n)
w1 = int((hashed*inv2)%n)
w2 = int((f1*inv2)%n)
P = E([Gx,Gy])
Q = E([Px,Py])
P_ = w1*P + w2*Q
print("e) Verificar la firma:")
print(f1 == int(P_[0])%n)
e) Verificar la firma:
True
```

on Gx i Gy són les components del generador i Px i Py són les components del punt de la clau pública. El resultat ha estat que efectivament la firma era vàlida.

2. **Trobeu a la xarxa un lloc web, diferent de www.wikipedia.org, que presenti un certificat amb una clau pública que sigui un punt d'una corba el·líptica. Doneu el punt i la corba que fa servir.**
   He escollit YouTube (youtube.com). També utilitza la corba P-256, i el punt, llegit del certificat utilitzant el mateix codi del cas anterior, és:
   26290164927358189317303931243902068566833940247834251915977646536943439527456,
   589162984046817826902479073634608909457394926848511966219205891134450277420 38 (més la tercera coordenada, que és 1).
   Fent servir aquesta corba:

   - **Genereu un parell clau privada/clau pública a partir de la corba el·líptica que es fa servir al certificat.**
     Utilitzant el procediment explicat a classe per generar claus privades:

     ```
     Zp = Zmod(p)
     E = EllipticCurve(Zp,[a,b]);
     G = E([Gx,Gy])
     r = random.randint(1,n)
     ```

```
public_key = r*G
```

La clau privada és l'enter r, generat aleatòriament. Al l'*script* l'he deixat *hardcodejat* a $r = 32903829104380230318795649424761085390545183133709242454697246715961201857515$ per fer els resultats reproduïbles. Per altra banda, Gx i Gy són les coordenades del generador de la P-256.

- **Demana a un company que signi dos missatges diferents amb la seva clau pública però fent servir el mateix nombre aleatori i troba la seva clau privada.**
  El meu company Erik Miedes Bragado m'ha enviat el següent:

```
sha(m1) = 1af8a350161756100593aa0534e82ba3bbced07e0d2795d8dfaf0b74a9720f35
sha(m2) = eaa75c66efe2f4e56aa472c64053848e810d5003d6530a80ee5d8717b800fd31
(f11, f12) = (565152197906911714131090579040116886954248101558029299735264813213098562420 40,
8876801693421390096363116482548108121968470671853380804365777977091432063018 9)
(f21, f22) = (565152197906911714131090579040116886954248101558029299735264813213098562420 40,
1994466196642620024209026263383348591572447109913784515707310862432928553865 0)
```

  També m'ha dit que la seva clau pública era el punt:

```
Qx,Qy = (926765709390531455995935574289488509410305262003149659596609246010764465936 29,
699993725804771059838060889185048541206980497368430306278849144388025973871 3)
```

Utilizant el següent codi:

```
sinv = sympy.invert((f2_msg1 - f2_msg2) % n, n)
rinv = sympy.invert(f1_msg1, n)
k = int(sinv*(hash_msg1 - hash_msg2) % n)
r_company = int(rinv*(k*f2_msg1 - hash_msg1) % n)
```

He trobat que havia utilitzat el nombre aleatori $k = 2$ i la seva clau privada era:

```
r = 576617832886074017058704696679545395802631845618740872857975571059836357032 70
```

Cosa que m'ha confirmat i es pot comprovar a la seva entrega.

La conclusió és que tot i que aquest criptosistema basat en les corbes el·líptiques és extremadament segur (el problema és encara més intractable que en el cas de RSA i no es coneix cap algorisme mínimament eficient), al final la seguretat es pot veure compromesa si el generador de nombres aleatoris no és de prou qualitat o simplement no s'utilitza correctament.

Per altra banda, el company Enrique González m'ha demanat que li firmi dos missatges, i per fer-ho he fet el següent:

```
m1 = b"Cryptography is easy"
m2 = b"Fake news!"
hash_m1 = hashlib.sha256(m1).hexdigest()
hash_m2 = hashlib.sha256(m2).hexdigest()
k = 2019
kinv = sympy.invert(k,n)
punt = k*public_key
kinv = sympy.invert(k,n)
f1_m1 = int(punt[0]) % n
f2_m1 = kinv * (int(hash_m1,16) + f1_m1 * r) % n
f1_m2 = int(punt[0]) % n
f2_m2 = kinv * (int(hash_m2,16) + f1_m2 * r) % n
```

```
print("sha256 m1 = " + hash_m1)
print("firma m1 = " + str(f1_m1) + ", " + str(f2_m1))
print("sha256 m2 = " + hash_m2)
print("firma m2 = " + str(f1_m2) + ", " + str(f2_m2))
```

He hashejat (*sha256*) els dos missatges i he utilitzat el procediment que vam veure per firmar, amb 2019 com a nombre aleatori i la clau privada que havia generat anteriorment.

3. **En el certificat trobat al segon apartat es dóna un punt de distribució de la CRL de l'autoritat certificadora i una adreça on fer peticions OCSP.**

   - **Quants certificats revocats conté la CRL?**
     He extret la URL amb la CRL del camp CRL Distribution Point trobat en una de les extensions. Era la següent: http://crl.pki.goog/GTSGIAG3.crl En aquesta adreça es pot baixar l'arxiu CRL, *GTSGIAG3.crl*. Amb la següent comanda de *Bash*, utilitzant la utilitat *openssl*:

     ```
     openssl crl -inform DER -in GTSGIAG3.crl -noout -text
     ```

     sobté la següent sortida:

     ```
     Certificate Revocation List (CRL):
             Version 2 (0x1)
         Signature Algorithm: sha256WithRSAEncryption
             Issuer: /C=US/O=Google Trust Services/CN=Google Internet Authority G3
             Last Update: Dec 24 01:00:21 2018 GMT
             Next Update: Jan  3 01:00:21 2019 GMT
             CRL extensions:
                 X509v3 Authority Key Identifier:
                     keyid:77:C2:B8:50:9A:67:76:76:B1:2D:C2:86:D0:83:A0:7E:A6:7E:BA:4B

                 X509v3 CRL Number:
                     548
     Revoked Certificates:
         Serial Number: 3BC1FF92F32FC6DF
             Revocation Date: Apr 25 10:53:43 2018 GMT
             CRL entry extensions:
                 X509v3 CRL Reason Code:
                     Cessation Of Operation
         Serial Number: 1CAF5665073F79C3
             Revocation Date: Jun 26 05:31:48 2018 GMT
             CRL entry extensions:
                 X509v3 CRL Reason Code:
                     Key Compromise
         Serial Number: 67168E04F45AAF5B
             Revocation Date: Feb 26 18:19:45 2018 GMT
             CRL entry extensions:
                 X509v3 CRL Reason Code:
                     Key Compromise
         Signature Algorithm: sha256WithRSAEncryption
              87:51:df:12:64:22:85:a4:b7:77:bc:9e:34:33:91:5b:67:bb:
              93:22:12:96:1e:16:4e:89:51:a4:e4:67:30:f9:9a:61:65:0a:
              06:b0:9d:66:ab:26:0a:eb:6d:e9:2d:6b:e6:38:67:9a:42:a1:
              c9:33:9d:2c:7d:8d:ae:f2:e5:d3:d6:2b:3a:d6:b6:6d:07:41:
              fc:7d:75:08:0c:c9:0d:94:14:72:65:62:ee:22:83:dd:e5:d4:
              f4:8c:2c:ab:0a:bb:23:35:8b:b0:0f:e4:0e:a4:44:28:db:bd:
     ```

```
cf:58:8b:68:24:3e:6d:36:82:24:7d:3c:bf:a0:fd:bb:06:d8:
34:59:89:54:9c:fe:78:d1:b9:19:ec:14:ad:2a:c2:93:48:f5:
f9:8b:83:56:8c:7a:09:59:09:f8:01:29:9e:4e:68:a3:c4:45:
d7:02:7b:fd:95:8a:ed:a9:e2:f1:4f:1e:78:15:34:25:41:1e:
3b:68:d0:49:4c:28:b7:89:1c:a9:df:6e:c7:a2:ed:22:cb:52:
2d:7e:53:b9:39:64:af:38:c8:db:6f:46:cf:9b:b1:12:c2:c7:
bb:54:63:4f:89:c4:8f:d3:9c:93:42:78:f1:58:06:cb:3b:bc:
5a:c9:db:0c:a4:a2:f5:36:2c:c4:da:2a:36:2a:ef:bb:c3:db:
79:7e:3b:57
```

S'hi identifiquen **3** certificats revocats, el primer d'ells per cessament d'operació i els altres dos perquè les claus es van veure compromeses.

- **Pregunteu a l'OCSP l'estatus del certificat que heu trobat.**
  Al camp *Authority Info Access Syntax* hi trovem l'URL: http://ocsp.pki.goog/GTSGIAG3. Després de consultar la documentació i baixar els certificats de l'autoritat, m'ha acabat funcionant la següent comanda:

```
openssl ocsp -no_nonce -issuer Google_Internet_Authority_G3.pem -cert
youtube_cert.pem -url http://ocsp.pki.goog/GTSGIAG3 -header Host
ocsp.pki.goog -noverify
```

  La sortida obtinguda és:

```
youtube_cert.pem: good
This Update: Dec 24 13:25:14 2018 GMT
Next Update: Dec 31 13:25:14 2018 GMT
```

  Significa que el certificat és vàlid, que s'acaba de renovar poca estona després que cridés la comanda i que s'haurà de renovar el dia 31 de desembre.