Marc Monfort Grau

Informe

ECC i certificats digitals

1. ECC - TLS 1.3 (wikipedia.org)

El fitxer ***captura_wikipedia_tls_1_3.pcapng*** conté la captura de totes les connexions en el moment de conectar-me a www.wikipedia.org, fent servir **Wireshark**.

Línies d'interès:

- **25:** Client Hello
- **28:** Server Hello, Change Cipher Spec, Application Data
- **30:** Certificate [TCP segment of a reassembled PDU]
- **32:** Certificate Verify, Finished

S'ha fet servir el fitxer ***keys.txt*** generat per Firefox com a ***(Pre)-Master-Secret log filename*** per poder llegir el protocol TLS a Wireshark.

I**nformació obtinguda**:

- Del paquet **Server Hello** (línia 28) obtenim el ***Cipher Suite*** utilitzat: **TLS_AES_256_GCM_SHA384**. Ara sabem que el servidor utilitzarà el **hash SHA384.**

- Del paquet **Certificate** (línia 30) obtenim la clau pública P de wikipedia.org com a punt de la corba el·líptica.
  SubjectPublicKey:
  04dedb39245f4d61ed2e9b2f892c9d2e7b9d56283c2e4feb71cf410839b825e15a175d2cb9e5def9e95e17028dd3e6a7a4b42542c4a98e134b4d0a50356e6f67b3
  Px (hex):
  dedb39245f4d61ed2e9b2f892c9d2e7b9d56283c2e4feb71cf410839b825e15a
  Py (hex):
  175d2cb9e5def9e95e17028dd3e6a7a4b42542c4a98e134b4d0a50356e6f67b3

Marc Monfort Grau

- Del paquet **Certificate Verify** (línia 32) obtenim la corba el·líptica utilitzada.
  Signature Algorithm: ecdsa_secp256r1_sha256
  i també obtenim la signatura:
  Signature:
  3046022100c5581c8fc4087eb656a0bab9b11b835e535dfcf77100a85d4341f548
  06abc432022100a216e731445535e7f28b380674c3a40ec0398fd464ce1ca0534
  d76d1a66296bf
  f1 (hex):
  00c5581c8fc4087eb656a0bab9b11b835e535dfcf77100a85d4341f54806abc432
  f2 (hex):
  00a216e731445535e7f28b380674c3a40ec0398fd464ce1ca0534d76d1a66296bf

- Del document **NIST.FIPS.186-4.PDF** obtenim la informació de la corba **p256** :
  p (hex): ffffffff00000001000000000000000000000000ffffffffffffffffffffffff
  q/n – ordre (hex):
  ffffffff00000000ffffffffffffffffbce6faada7179e84f3b9cac2fc632551
  b (hex):
  5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b
  G (Punt públic de la corba el·líptica):
  Gx (hex):
  6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296
  Gy (hex):
  4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5

Per obtenir el **missatge** firmat i comprovar la firma, he guardat 4 fitxers en binari amb els noms 1.bin, 2.bin, 3.bin i 4.bin. Cadascun correspon a:

- 1.bin = **Handshake Protocol: Client Hello** (linia 25)
- 2.bin = **Handshake Protocol: Server Hello** (linia 28)
- 3.bin = **Handshake Protocol: Encrypted Extensions** (linia 28)
- 4.bin = **Handshake Protocol: Certificate** (linia 30)

El fitxer **men.bin** contindrà la concatenació de 1.bin, 2.bin, 3.bin i 4.bin.

Marc Monfort Grau

2. Càlculs i comprovacions

Per fer els càlculs i les comprovacions he utilitzat SAGE online (sagecell.sagemath.org).

El codi utilitzat és el següent (el fitxer ***sage.txt*** conte el mateix codi):

```
p = 0xffffffff00000001000000000000000000000000ffffffffffffffffffffffff
a = -3
b = 0x5ac635d8aa3a93e7b3ebbd55769886bc651d06b0cc53b0f63bce3c3e27d2604b
q = 0xffffffff00000000ffffffffffffffffbce6faada7179e84f3b9cac2fc632551 # eq. n (ordre)
E = EllipticCurve(Zmod(p), (a, b))

# (a) Comproveu que el nombre de punts (ordre) de la corba que es fa servir al certificat és primer.
print("(a):", is_prime(q))

# (b) Comproveu que la clau pública P de www.wikipedia.org és realment un punt de la corba.
Px = 0xdedb39245f4d61ed2e9b2f892c9d2e7b9d56283c2e4feb71cf410839b825e15a
Py = 0x175d2cb9e5def9e95e17028dd3e6a7a4b42542c4a98e134b4d0a50356e6f67b3
print("(b):", [Px,Py] in E)

# (c) Calculeu l'ordre del punt P .
P = E(Px,Py)
print("(c):", P.order())

# (d) Comproveu que la signatura ECDSA és correcta.
f1 = 0x00c5581c8fc4087eb656a0bab9b11b835e535dfcf77100a85d4341f54806abc432
f2 = 0x00a216e731445535e7f28b380674c3a40ec0398fd464ce1ca0534d76d1a66296bf

Gx = 0x6b17d1f2e12c4247f8bce6e563a440f277037d812deb33a0f4a13945d898c296
Gy = 0x4fe342e2fe1a7f9b8ee7eb4a7c0f9e162bce33576b315ececbb6406837bf51f5
G = E(Gx,Gy)

mensaje = 0x213de96e0297454748d543c43e5e80e01aafbb8392a39f34b30278e9ded70102

w1 = mensaje * f2^(-1) % q
w2 = f1 * f2^(-1) % q

X = w1*G + w2*P
print("(d):", int(X[0]) % q == f1)
```

Marc Monfort Grau

Per cada apartat retorna el seu resultat. **Resultat obtingut:**

(a): True

(b): True

(c): 1157920892103562487626974469494075735299969552241357603424222590610685120443 69

(d): True

Queda comprovat que el nombre de punts (ordre) de la corba és primer, que la clau pública P és un punt de la corba, que l'ordre del punt P es:
1157920892103562487626974469494075735299969552241357603424222590610685120443 69
(el mateix valor que l'ordre de la corba **q (n)**)
i finalment he verificat la firma del missatge.

Per obtenir el valor del missatge he utilitzat el codi del fitxer ***mensaje.py***:

```
import hashlib

preambulo = 64*'20'
preambulo += ''.join(format(ord(c),'x') for c in 'TLS 1.3, server CertificateVerify')
preambulo += '00'

with open("./men.bin", 'rb') as f:

    men384 = hashlib.sha384(f.read())
    m = preambulo + men384.hexdigest()
    mensaje = hashlib.sha256(bytes.fromhex(m))

    print('mensaje (hex):', mensaje.hexdigest())
```

On el *preambulo* sempre és el mateix, i la resta del missatge l'he obtingut a partir del fitxer ***men.bin*** que és la concatenació de 1.bin, 2.bin, 3.bin i 4.bin.

Per obtenir el valor final del missatge, s'ha de realitzar el **sha256** de la concatenació del *preambulo* amb el **sha384** del fitxer **men.bin**.

S'utilitza el **sha384** ja que és el hash especificat en *el **Cipher Suite*** enviat per Wikipedia.

S'utilitza el **sha256** ja que és el hash especificat en el **Signature Algorithm** enviat també per Wikipedia.

Marc Monfort Grau

3. CRL - Certificat Digital (fib.upc.edu)

M'he connectat al la pàgina fib.upc.edu i he obtingut a través de Firefox els seguents documents:

- TERENASSLCA3.crl
- TERENASSLCA3.crt
- DigiCertAssuredIDRootCa.crt

a)

Per veure el nombre de certificats revocats que conté la CRL, he convertit el fitxer TERENASSLCA3.crl a un fitxer de text amb la comanda:

```
openssl crl -inform DER -text -noout -in TERENASSLCA3.crl -out crl.txt
```

A partir del fitxer de text he pogut comptar el nombre de "Revoked Cerificates", amb un resultat de **7112** certificats.

b)

Per poder preguntar sobre l'estatus del certificat, primer he hagut de convertir els dos certificats **.crt** a l'extensió **.PEM** utilitzant les comandes:

```
openssl x509 -inform DER -in TERENASSLCA3.crt -out terena.pem -text
```

```
openssl x509 -inform DER -in DigiCertAssuredIDRootCa.crt -out digicert.pem -text
```

Ara ja tinc la cadena (digicert.pem) i el certificat (terena.pem). Amb la següent comanda pregunto l'estat del certificat a la OSCP correponent (http://ocsp.digicert.com):

```
openssl ocsp -issuer digicert.pem -cert terena.pem -url http://ocsp.digicert.com
```

i el resultat obtingut és:

Marc Monfort Grau

WARNING: no nonce in response

Response verify OK

terena.pem: good

This Update: Dec 8 20:11:48 2020 GMT

Next Update: Dec 15 20:11:48 2020 GMT

Podem veure que l'estatus del certificat és **good** i que aquest és **vàlid** fins al **Next Update** que serà el 15/12/2020 a les 20:11:48 GMT.