

## ECC y certificados digitales

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.
  - (b) Comprobad que la clave pública  $P$  de **www.wikipedia.org** es realmente un punto de la curva.
  - (c) Calculad el orden del punto  $P$ .
  - (d) Comprobad que la firma ECDSA es correcta.
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 (RSA: módulo y exponente; ECC: curva y coordenadas X e Y del punto) del web de la FIB.
  - (b) Indicad las marcas de tiempo de los certificados, hasta el milisegundo (Signed Certificate Timestamp).
  - (c) En el certificado encontraréis los DNS para los que es válido, indicadlos.
  - (d) En el certificado encontraréis un enlace un punto de distribución de la CRL de la autoridad certificadora. ¿Cuántos certificados revocados contiene la CRL?, ¿cuántos hay sido revocados por Key Compromise?
  - (e) 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?

## Entrega

Un único fichero **zip**, **tar**,... con:

1. Los ficheros con las capturas de las conexiones y los ficheros adicionales necesarios para descifrar los paquetes cifrados y comprobar la validez de la firma. Se han de explicitar claramente los paquetes involucrados en la conexión con **www.wikipedia.org**.
2. Los cálculos y las comprobaciones pedidas en primer punto. Podéis usar <https://pypi.org/project/ECPy> y <https://docs.sympy.org/latest/modules/ntheory.html>
3. Las respuestas a las preguntas del segundo punto, la CRL y un pequeño **script** (y los ficheros necesarios) para comprobar el estatus del certificado del web de la FIB y la validez de la respuesta del OCSP. Podéis usar **openssl**.

## Referencias

- Wireshark network protocol analyzer
- Wireshark: Using the (Pre)-Master-Secret in TLS
- The New Illustrated TLS Connection
- The Transport Layer Security (TLS) Protocol Version 1.3, <https://tools.ietf.org/html/rfc8446>
- TLS 1.3: **Certificate Verify** message
- FIPS 186-5 Digital Signature Standard, <https://csrc.nist.gov/pubs/fips/186-5/final>
- NIST SP 800-186 Recommendations for Discrete Logarithm-based Cryptography
- RFC 5480: Elliptic Curve Cryptography Subject Public Key Information, <https://tools.ietf.org/html/rfc5480#section-2.2>
- Standards for Efficient Cryptography Group (SECG), "SEC 1: Elliptic Curve Cryptography", <http://www.secg.org/sec1-v2.pdf>
- Standards for Efficient Cryptography Group (SECG), "SEC 2: Recommended Elliptic Curve Domain Parameters", <https://www.secg.org/sec2-v2.pdf>
- Openssl x509 - Certificate utility, <https://docs.openssl.org/master/man1/openssl-x509/>
- Openssl crl - CRL utility, <https://docs.openssl.org/master/man1/openssl-crl/>
- Openssl ocsp - Online Certificate Status Protocol utility, <https://docs.openssl.org/master/man1/openssl-ocsp/>