Network Security

Lab1

1. Basic concepts
2. Who issued the digital certificate?

Let’s Encrypt

1. What is the validity of the certificate?

February 23, 2025 – May 25, 2025

1. How many bits is the public key defined in?

Hexadecimal string, each hex character has 4 bits  
So, 64 \* 4 = 256 bits

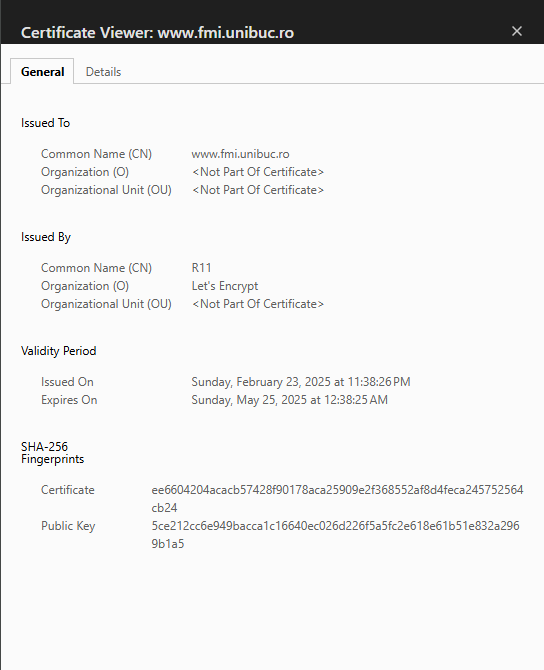
1. What is the value of the encryption exponents in the certificate and in the certificates that attest it in the chain? What do you notice? Does this impact security?

Public Exponent: 0x010001 → 65537

The same exponent (65537) is used in your certificate and in the certificates that attest it. This is intentional and it is the industry-standard value for e.

65537 is chosen for good security and performance.

* It's large enough to avoid attacks that affect small exponents like e = 3.
* It's small enough to allow fast operations like signature verification.



Conection is secure > Certificate is valid

1. Key generation using Putty
2. Download puttygen.exe.
3. Generate a 2048-bit RSA public key - private key pair. Press Generate.
4. Add PassPhrase. What is this for?

The PassPhrase encrypts your private key file on disk. Think of it like a password that protects your private key from unauthorized use.

1. Export the public key to the public\_key.pub file, the private key to the private\_key.ppk file. To do this, use Save public key, respectively Save private key.
2. Export the key in openssh format. To do this, go to Conversions and Export OpenSSH key.
3. Open and see what all the generated files contain.

public\_key.pub - base64-encoded representation of the public key

private\_key.ppk - contains both public and private parts

private\_openssh.key - openssh private key

1. Digital Certificate Generation using OpenSSL
2. Generate an RSA key.

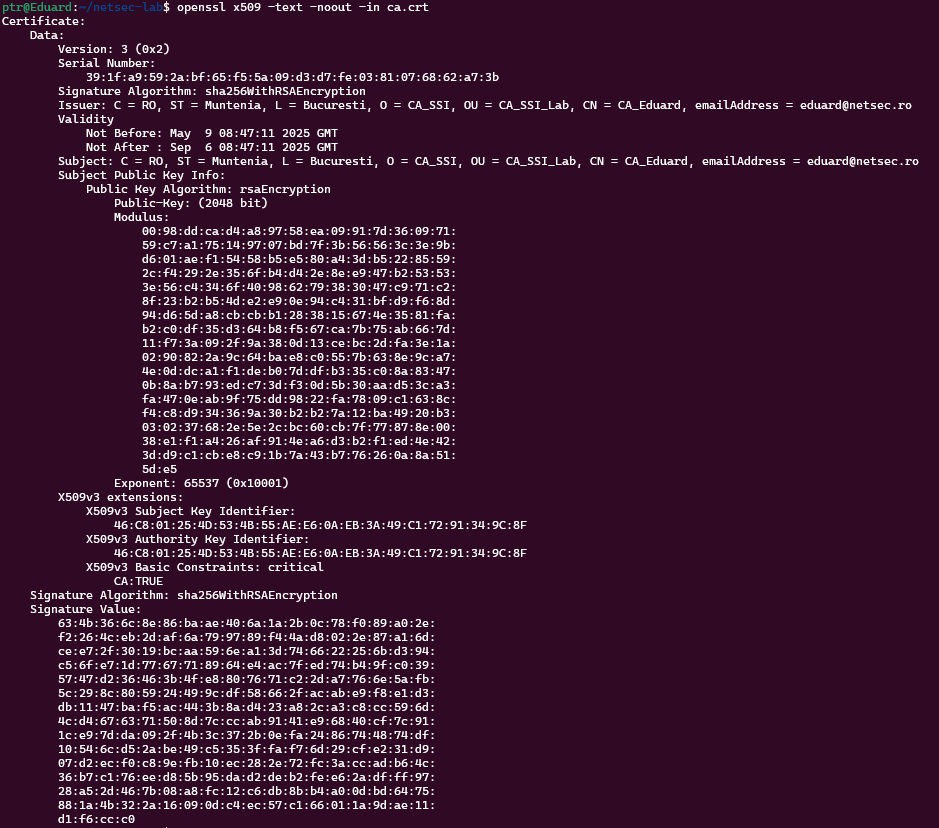
openssl genrsa -out ca.key 2048

1. Use the previously generated key in a *self-signed* certificate, valid for 120 days, stored as *ca.crt.*

openssl req -new -x509 -days 120 -key ca.key -out ca.crt

1. Read about *X.509* [6]. Have a look at the digital certificated that was created

openssl x509 -text -noout -in ca.crt



1. Use this CA certificate to sign/issue another certificate of a subordinate entity SUB\_SLA.

openssl genrsa -out sub.key 2048

1. Initiate a *Certificate Signing Request (CSR) sub.csr*

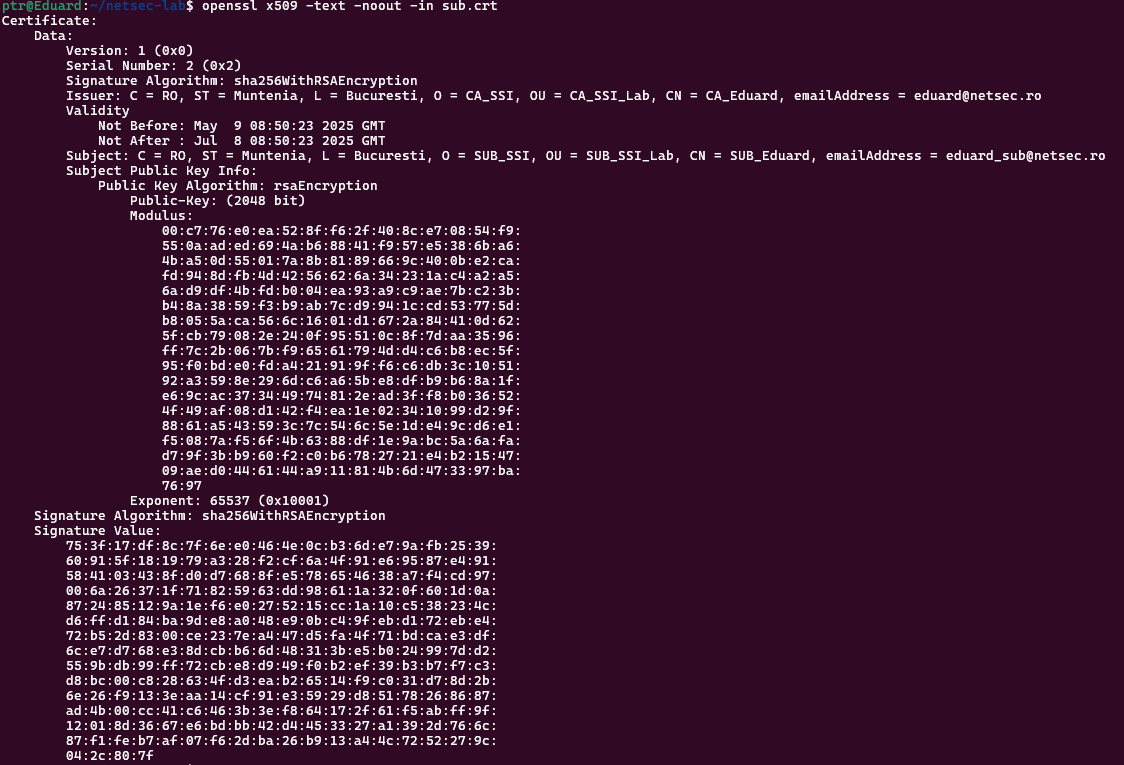
openssl req -new -key sub.key -out sub.csr

1. Then create a certificate for SUB\_SSI sub.crt signed by the CA authority, valid for 60 days, with serial number 02:

openssl x509 -req -days 60 -in sub.csr -CA ca.crt -CAkey ca.key -set\_serial 02 -out sub.crt

1. View the created digital certificate

openssl x509 -text -noout -in sub.crt



1. Transform this certificate to *PKCS#12*

openssl pkcs12 -export -out sub.p12 -inkey sub.key -in sub.crt -chain -CAfile ca.crt

1. Verify the content of *sub.p12* by using

openssl pkcs12 -info -in sub.p12