Ministry of Education, Culture and Research of the Republic of Moldova

Technical University of Moldova

Department of Software and Automation Engineering

**REPORT**

Laboratory work No. 5

**Discipline**: Cryptography and Security

Elaborated: Macrii Danu FAF-222,

Checked: asist. univ. Dumitru Nirca,

Chișinău 2024

Topic: **Cifruri bloc. Algoritmul DES**

Tasks:

Create an internal PKI using the OpenSSL tool. The generation of the root private

key and the initialization of a Certified Authority (CA) are required. A self-signed

certificate is created for the CA. System must be able to issue and revoke the private

key for the user so that he can subsequently generate a digital signature. Each

user/entity that obtains a signature will be able to sign the document/file and verify

this signature. For the realization of this laboratory, the use of any language is

allowed, including script languages such as Bash, PowerShell, zsh etc.

Theoretical notes:

Creating self-signed certificates and keys with OpenSSL is a common task for setting up secure communication channels, particularly in development or testing environments where a fully-trusted certificate authority (CA) is not necessary. OpenSSL is a widely-used open-source toolkit for working with Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols.

Here’s a step-by-step guide on how to create self-signed certificates and keys using OpenSSL:

Install OpenSSL: Ensure that OpenSSL is installed on your system. If not, you can download and install it from the official OpenSSL website or use your operating system’s package manager to install it.

Generate a Private Key: The first step is to generate a private key. You can do this using the following OpenSSL command:

**Implementation()**

**#!/bin/bash**

PATH\_TO\_PKI="/Users/macriidanu/Desktop/csLab/lab5"

mkdir -p "$PATH\_TO\_PKI"

read -p "Press Enter to continue..."

echo "1. Generate a private key using RSA Algorithm with 4096 bits"

openssl genpkey -algorithm RSA -out "$PATH\_TO\_PKI/private\_key.pem" -pkeyopt rsa\_keygen\_bits:4096

read -p "Press Enter to continue..."

echo "2. Generate X509 Certificate for CA"

openssl req -new -x509 -noenc -days 3650 -key "$PATH\_TO\_PKI/private\_key.pem" -out "$PATH\_TO\_PKI/root\_cert.pem"

read -p "Press Enter to continue..."

echo "3. Display the content of the certificate"

openssl x509 -in "$PATH\_TO\_PKI/root\_cert.pem" -text -noout

read -p "Press Enter to continue..."

echo "4. Generate a Private Key for the User"

openssl genpkey -algorithm RSA -out "$PATH\_TO\_PKI/user\_private\_key.pem" -pkeyopt rsa\_keygen\_bits:2048

read -p "Press Enter to continue..."

echo "5. Generate a Certificate Signing Request for the User"

openssl req -new -key "$PATH\_TO\_PKI/user\_private\_key.pem" -out "$PATH\_TO\_PKI/user\_cert\_req.csr"

read -p "Press Enter to continue..."

echo "6. Sign the User Certificate Request"

openssl x509 -req -in "$PATH\_TO\_PKI/user\_cert\_req.csr" -CA "$PATH\_TO\_PKI/root\_cert.pem" -CAkey "$PATH\_TO\_PKI/private\_key.pem" -CAcreateserial -out "$PATH\_TO\_PKI/user\_cert.crt" -days 365

read -p "Press Enter to continue..."

echo "7. Display the content of the certificate"

openssl x509 -in "$PATH\_TO\_PKI/user\_cert.crt" -text -noout

read -p "Press Enter to continue..."

echo "8. Sign a text file with the User Private Key"

openssl dgst -sha256 -sign "$PATH\_TO\_PKI/user\_private\_key.pem" -out "$PATH\_TO\_PKI/signature.txt" "$PATH\_TO\_PKI/user\_text.txt"

read -p "Press Enter to continue..."

echo "9. Verify the signature"

echo "9.1 Extract public key from the certificate"

openssl x509 -in "$PATH\_TO\_PKI/user\_cert.crt" -pubkey -noout > "$PATH\_TO\_PKI/public\_key.pem"

echo "9.2 Verify the signature"

openssl dgst -sha256 -verify "$PATH\_TO\_PKI/public\_key.pem" -signature "$PATH\_TO\_PKI/signature.txt" "$PATH\_TO\_PKI/user\_text.txt"

read -p "Press Enter to continue..."

echo "10. Modify the text file"

echo "This is a modified text file" > "$PATH\_TO\_PKI/user\_text.txt"

read -p "Press Enter to continue..."

echo "11. Verify the signature again"

openssl dgst -sha256 -verify "$PATH\_TO\_PKI/public\_key.pem" -signature "$PATH\_TO\_PKI/signature.txt" "$PATH\_TO\_PKI/user\_text.txt"

read -p "Press Enter to finish..."

rm private\_key.pem

rm root\_cert.pem

rm user\_private\_key.pem

rm user\_cert\_req.csr

rm user\_cert.crt

rm root\_cert.srl

rm public\_key.pem

rm signature.txt

clear