INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Project for the ETS of Selected Topics in Cryptography

Coordinadora: Sandra Díaz Santiago
email: sdiazsa@ipn.mx;

1 General specifications

To evaluate ETS especial examination of Selected Topics in Cryptography, in addition to solve the written exam, you must develop the project described below. The evaluation will be as follows:

- **Written exam:** 80%
- **Project** 20%

You can use C/C++, Python or Java to develop the project.

We will check your project at the end of the written exam.

During your project review, you must be able to answer questions about your implementation and about the details of the cryptographic algorithms that you implemented. In particular you must be able to argue about the security of the cryptographic algorithms.

2 Description

The CEO of an important company requires to digitally sign and verify the signature of important documents. Help the CEO to design and implement a computer program to solve his problem, using cryptography.

To sign and verify you must do your own implementation of ECDSA, and the toy hash function described below. It is **mandatory** that your implementation satisfies the following requirements:

1. Design and implement functions to add points and double points in the elliptic curve. You must use prime numbers of 64 bits or bigger. Use a cryptographic library of the programming language to generate a prime number and to do arithmetic with large numbers.

2. Use three coordinates to represent a point the elliptic curve (x, y, z). The infinity point will be represented as $\mathcal{O} = (0, 1, 0)$ any other point in the curve will have $z = 1$

3. Implement a program that receives the parameters of an elliptic curve a, b, p, q, where $y^2 = x^3 + ax + b \bmod p$. Your program must randomly generate a key pair for ECDSA. Store the public key $K_{pub} = (p, a, b, q, G, B = dq)$ in textfile. Store the private key and store it in a different file. The filename of each file must be chosen by the user.

4. Implement a program to generate an ECDSA signature of a textfile. This program, must receive the parameters of an elliptic curve p, a, b, q, a generator G for the elliptic curve, the filename of a private key and the name of the textfile to be signed. The output must be the signature in base64. The size of the textfile to be signed must be at least of 500 words.

5. Implement a program to verify an ECDSA signature of a textfile. This program, must receive the parameters of an elliptic curve a, b, p, q, a generator G for the elliptic curve, the filename of a public key and the name of the textfile that was signed and the signature. Your program must indicate if the signature is valid or not.

6. You must implement the following toy hash function and use it as part of your program to sign and verify.

Algorithm 1: toyhash(m, n)

Input : $m = m_0, \ldots, m_{t-1}$ where $|m_i| = 16$ bits; n an integer of 32 bits
Output: $h(m)$ a digest of 32 bits

```
1 sum ← 0 ;
2 for i ← 0 to t − 1 do
3 |   sum ← sum + m_i ;                    /* m_i will represent an integer */
4 end
5 return sum mod n
```

3 Products

To evaluate your project, you must present the following:

- Your application running without errors. It is very important that your program your program must be able to sign and verify the signature.
- Your source code
- A written report

3.1 Report

You must write a report containing the following:

1. The source code of the cryptographic functions you implemente and a brief explanation of how each of these functions works.
2. Screen shots, showing each important part of your application running.
3. If you used an artificial intelligence to generate source code. You must specify which part of the code you implemented and which part was generated with the IA. Also you must justify why did you require to use AI.
4. References properly written. Here you must include at least two books of cryptography that you have used.