INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Proyecto para el ETS fuera de calendario

Selected Topics in Cryptography

Coordinator: Sandra Díaz Santiago
email: sdiazsa@ipn.mx; sds.escom@gmail.com

1. General specifications

To evaluate ETS (fuera de calendario) 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:** 70 %
- **Project** 30 %
  - Key generation, signature and verification with ECDSA : 15pts
  - Key generation, encipher/decipher with **Chacha20-Poly1305** : 5pts
  - Key generation, encipher/decipher with RSA-OAEP: 5pts
  - Written report: 5pts

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

If you are going to present the exam in the morning, we will check your project at the end of the written exam. If you are going to present the exam in the evening, we will check your project just before the written exam.**In any case, please send an email to make an appointment to review your project.**

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 must share sensitive documents with his employees in such a way that only authorized people has access to them. Thus he needs to encipher/sign sensitive documents and an employee must be able to decipher and verify the signature. Help the CEO to design and implement a computer program to solve his problem, using cryptography.

It is **mandatory** that your implementation satisfies the following requirements:

- Use **Chacha20 and Poly1305** to provide authenticated encryption.
- Only the CEO must be able to generate a key and nonce for Chacha20 and Poly1305. Both the key and the nonce must be randomly generated.
- To share the secret key for Chacha20 in a secure way with an employee, use key-wrapping with RSA-OAEP to encipher it . For this purpose, the CEO must generate his key pair for RSA-OAEP. Also each employee must generate his own key pair for RSA-OAEP.
- You can use a cryptographic library for key generation, encipher and decipher with Chacha20-Poly1305 and RSA-OAEP. The private key for RSA-OAEP must be store in a file and the public key in a different file. Both keys must be stored in file using base 64.
- The ciphertext and the enciphered key for Chacha20-Poly1305, must be stored in a file using base64.
- To sign and verify you must do your own implementation for ECDSA and a toy hash function that produces digest of 64 bits (you can propose one, it does not have to be collision resistant).

Your implementation for ECDSA, must fulfill the following:

1. Design and implement functions to add points and double points in the elliptic curve:

   $$y^2 = x^3 + x + 1844674407370954114 \text{ mód } 18446744073709551629$$

   with primitive element $G = (476675671136392426, 5963031211007724569, 1)$. The cardinality of this elliptic curve is a prime number: 18446744080022336321

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

3. You must randomly generate a key pair for ECDSA and stored them in a text file.

3. Products

To evaluate your project, you must present the following:

- Your application running without errors. It is very important that your program properly enciphers and deciphers files. Also your program must be able to sign and verify the signature.
- Your source code
- A written report and a user manual
- **You must provide correct answers during the oral evaluation of your project. In particular, you must know how Chacha20-Poly1305 work.**

3.1. Report

You must write a report containing the following:

1. The architecture of your system, that shows the most important blocks of your systems, specifying the inputs and outputs for each block.
2. A brief introduction explaining what the application does, and describing the most important parts of your system.
3. Description in your own words of the cryptographic algorithms that you use.
4. The most important functions of your code and a brief explanation of how each of these functions works.
5. Screen shots, showing each important part of your application running.
6. A user manual, here you must explain how to use your application.
7. References properly written. Here you must include at least two books of cryptography that you have used.