

16-10-2024

**Nombre:** Álvarez García Brandon Azarael

**Nombre de la materia:** Selected topics in cryptography

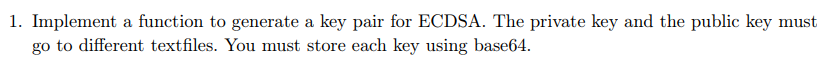
**Grupo:** 7CM1

**Nombre del profesor:** Dra. Sandra Díaz Santiago

Session 6:

ECDSA for real implementations

**Ejercicios de programación**



Para esta implementación utilizaremos la librería Crypto con sus diferentes módulos, la cual nos ayudara a implementar lo necesario para esta practica

Texto

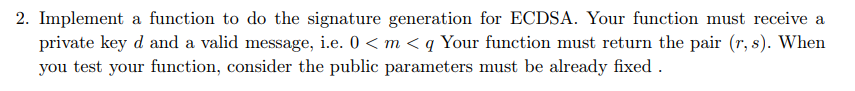
Descripción generada automáticamente

Esta función nos ayudara a crear el par de llaves tanto publica como privada, todo se realiza mientras la función ECC.generate , la cual lleva como parámetro una curva estandarizada, yo escogí la P-256, sin embargo soporta otras, como P-192, P-224 etc, de ahí extraemos la llave publica y privada, y las guardamos en un archivo de texto por separado, así como la codificamos en base 64.

Adicionalmente, para los ejercicios posteriores utilizamos la cardinalidad, sin embargo, dentro de la librería no hay algún método que te retorne esta, así que cree una función que devuelve la cardinalidad de la curva P-256

Imagen que contiene Texto

Descripción generada automáticamente

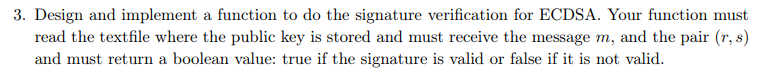


Texto

Descripción generada automáticamente

En este función leemos el archivo de texto donde contenemos nuestra llave privada y hacemos uso de la función creada previamente para obtener el orden, esta función lleva como parámetro un mensaje m representado en enteros y utilizamos una función hash , la más conocida es la SHA256 y creamos la firma, posteriormente obtenemos los parámetros r y s , sin embargo la función que devuelve la firma, los devuelve pero de forma concatenada, así que lo separamos en variables independientes y las retornamos.

Texto

Descripción generada automáticamente

Para esta función vamos a usar loa parámetros previamente calculados, r,s y el mensaje, posteriormente obtenemos la llave publica de nuestro archivo y utilizamos la función hash256, definida a base del mensaje y vovlemos a generar la firma como la habíamos obtenido inicialmente (sin desestructurarla en dos parámetros) y procedemos a verificar con la función, pasando la firma y el valor de nuestro hash.

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente con confianza media

Texto

Descripción generada automáticamente

Para el punto 4,5 cree una función que junta las funciones previamente creadas y el proceso se repite las 5 veces que se mencionan.

**Resultados.**

Para los resultados, cree otro archivo Python, en donde simplemente mande a llamar la última función creada test\_program(), para no juntar todo en un solo archivo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente