**CRIPTOGRAFÍA APLICADA**

**LABORATORIO 1**

José Benavente y Daniel Aguayo

29 de mayo de 2023

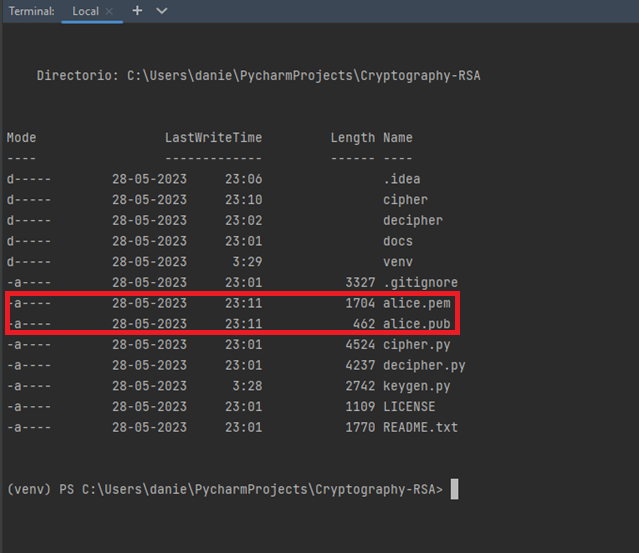
**DEMOSTRACIÓN**

En este punto se demuestra, a través de capturas de pantalla, la encriptación y desencriptación de mensajes utilizando los scripts programados en Python.

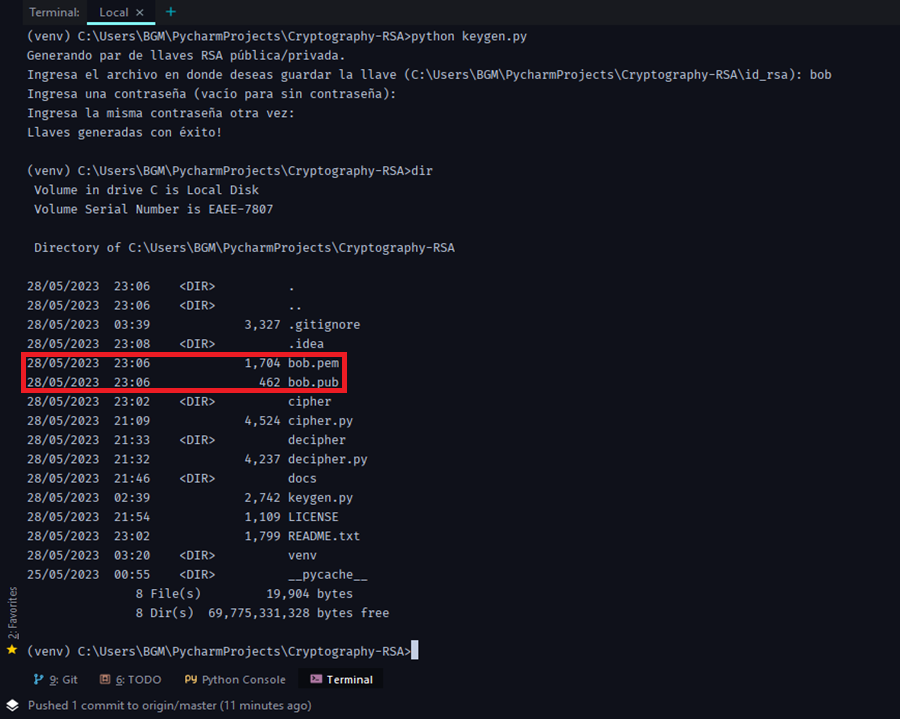
Para efectos de este ejemplo, Daniel será Alice y José será Bob, con lo que primero, Alice genera su par de llaves RSA:

A screenshot of a computer

Description automatically generated with medium confidence



Luego, Bob genera su par de llaves también:



Después de este paso, ambos usuarios intercambian sus llaves públicas. Una vez intercambiadas, Alice puede cifrar el primer mensaje usando su llave privada y la llave pública de Bob. En este caso, el mensaje en texto plano que Alice desea cifrar es “**Hola**”:

A screen shot of a computer

Description automatically generated with medium confidence

Luego de cifrar el mensaje, Alice le envía a Bob los archivos que componen al mensaje cifrado y su firma: **aes\_key.enc**, **ciphertext.txt**, **IV.iv** y **signature.sig**. Con estos archivos en su poder, Bob procede a desencriptar el mensaje usando su llave privada y la llave pública de Alice:



Como se puede apreciar en la imagen, el mensaje es genuino y la firma es válida, apareciendo así el contenido del mensaje en texto plano.

El proceso inverso es el mismo, pero los roles se invierten. Ahora Bob cifra el mensaje, y Alice lo descifra:

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated with medium confidence

Para demostrar la verificación de la firma, Bob genera otro mensaje de prueba completamente distinto “**aloH**”. Este mensaje es cifrado y guardado en otro archivo **ciphertext.txt**. Al intentar descifrarlo con los archivos **aes\_key.enc**, **IV.iv** y **signature.sig** asociados a la encriptación original del mensaje “**Hola**”, la verificación falla y la ejecución termina.



**EVIDENCIA DE ESTÁNDAR PEP8**

Para asegurarnos de que nuestros scripts cumplieran con las exigencias del estándar PEP8, utilizamos **plint**, librería de Python vista en clases.

Aquí están los resultados del escanéo de los tres scripts que componen nuestro proyecto:

