Objetivo: Criptografía Asimétrica y Funciones de Hashing

Trabaje con un compañero de grupo y basado en el material expuesto en horario de laboratorio sobre criptografía asimétrica se le pide realizar tres programas en python (extensión py):

- El primer programa debe crear un par de llaves asimétricas RSA para ambos integrantes del grupo.
    - Las parejas de llaves deben ser almacenadas en un archivo formato PEM y etiquetadas con el nombre de su dueño. Ejemplo: llave_privada_Alice.key y llave_publica_Alice. key (de manera equivalente para Bob)
    - Intercambien los archivos que contienen sus llaves públicas
- El segundo programa debe permitir que dos personas intercambien mensajes cifrados. Por simplicidad piense que usted es Alice desee enviar un mensaje a su compañero Bob:
    1. Solicitar un texto desde teclado a Alice,
    2. Lee llave privada de Alice y llave pública de Bob
    3. Firma texto plano con llave privada de Alice. Escriba firma en un archivo (Por ejemplo "Signature_Alice. sig")
    4. Genera una llave AES y cifra texto plano en modo CBC con AES (no cifre la firma) y escribe texto cifrado en un archivo. También escribe vector IV en un archivo (IV.iv)
    5. Cifra la llave AES con llave pública de Bob y almacena llave AES cifrada en otro archivo. (Ejemplo llave_AES_cifrada.key)
- Tercer programa debe leer varios archivos: archivo del mensaje, archivo con la firma del mensaje llave pública del emisor (Alice) del mensaje, vector de inicialización, llave privada del destino (Bob) y llave AES cifrada.
    - Descifre llave AES cifrada con llave privada de Bob
    - Desencripte texto cifrado de Alice con llave AES
    - Verifique que el mensaje es genuino y señala si la firma es válida. Genere un archivo de prueba distinto al mensaje original para probar esta situación.
    - Solo si el mensaje es genuino, muestre el contenido del texto plano en pantalla

Condiciones de entrega:

1. Códigos debidamente comentados
2. Archivos de texto de prueba, parejas de llave, vector IV
3. Capturas de pantalla (en un archivo word), donde muestre un mensaje cifrado que es generado por usted a su compañero y viceversa
4. Breve archivo "README.txt" que describa como se ejecutan los archivos Python desarrollados e indique los nombres del grupo (2 alumnos) NO USAR archivos pynb (jupyter notebooks)
5. Debe mostrar evidencia (capturas de pantalla) que se ajusto al estándar python PEP8 (use pylint para ello)
6. Debe compactar estos archivos en un SOLO ARCHIVO, etiquetado como *Laboratorio2_\<Apellido_alumno1\>_\<Nombre_alumno1\>_\<Apellido_alumno2\>_\<Nombre_alumno2\>.zip* y subirlo en módulo de ADECCA (basta con una entrega por grupo).
7. Bonus (opcional) se premiará tarea que emplee el menor número de archivos (por supuesto no mezcle las llave publicas y privadas de Alice y Bob)