PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Criptografía y Seguridad Computacional - IIC3253
Tarea 1
Plazo de entrega: Lunes 24 de abril, 20hrs

Instrucciones

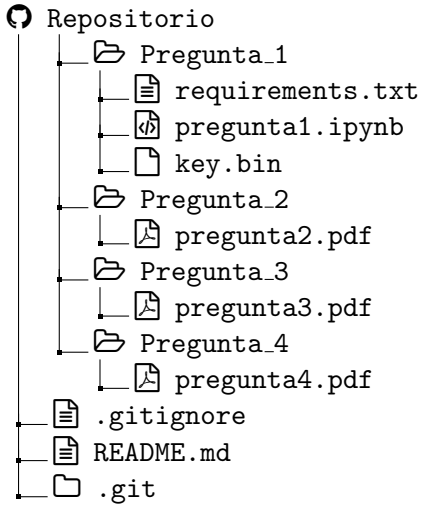
Cualquier duda sobre la tarea se deberá hacer en los *issues* del repositorio del curso. Si quiere usar alguna librería en sus soluciones debe preguntar primero si dicha librería está permitida. El foro es el canal de comunicación oficial para todas las tareas.

Configuración inicial. Para esta tarea utilizaremos *github classroom*. Para acceder a su repositorio privado debe ingresar al siguiente link, seleccionar su nombre y aceptar la invitación. El repositorio se creará automaticamente una vez que hagas esto y lo podrás encontrar junto a los repositorios del curso. Para la corrección se utilizará Python 3.10.

También deberá responder este formulario, en el que se le pedirá una **llave simétrica** que será utilizada para encriptar sus notas y correcciones usando el esquema AES. La llave debe tener como **máximo** un largo de 32 caracteres. En el repositorio del curso se publicará además el valor de hash de su llave utilizando SHA256. Si usted pierde dicha llave podrá recuperarla incurriendo en una penalización de dos décimas en el promedio de sus tareas. Si otra persona descubre su llave antes de que se publiquen las notas de la segunda tarea, el promedio de tareas de dicha persona aumentará en cinco décimas, mientras que el suyo disminuirá en cinco décimas.

Recomendación: **Use un administrador de contraseñas**.

Entrega. Al entregar esta tarea, su repositorio se deberá ver exactamente de la siguiente forma:



Deberá considerar lo siguiente:

- El archivo `requirements.txt` dentro de la carpeta de una pregunta deberá especificar todas las librerías que se necesitan instalar para ejecutar el código de su respuesta a dicha pregunta. Este archivo debe seguir la especificación de Pip, es decir se debe poder ejecutar el comando `pip install -r requirements.txt` suponiendo una versión de `Pip` mayor o igual a 22.0 que apunte a la versión 3.10 de Python. Si su respuesta no requiere librerías adicionales, este archivo debe estar vacío (pero debe estar en su repositorio).
- Se recomienda utilizar tipado para las variables y funciones en Python.
- Para cada problema cuya solución se deba entregar como un documento (en este caso las preguntas 2, 3 y 4), usted deberá entregar un archivo `.pdf` que, o bien fue construido utilizando LaTeX, o bien es el resultado de digitalizar un documento escrito a mano. En caso de optar por esta última opción, queda bajo su responsabilidad la legibilidad del documento. Respuestas que no puedan interpretar de forma razonable los ayudantes y profesores, ya sea por la caligrafía o la calidad de la digitalización, serán evaluadas con la nota mínima.

Preguntas

1. En esta pregunta usted deberá obtener una llave utilizada para encriptar mensajes con una variante de OTP bastante mala. Deberá entregar un archivo `Pregunta_1/key.bin` que contenga la llave que se utilizó para encriptar dichos mensajes, además de un notebook que explique el proceso que se siguió para obtener la llave.

   Los mensajes que debe decriptar serán subidos a su repositorio privado a lo más 24 horas después de su creación (ver Configuración Inicial más arriba).

2. Sea $\ell > 0$ un número entero, sea $\mathcal{M}$ el siguiente espacio de mensajes:

$$\mathcal{M} \quad = \quad \{\varepsilon\} \cup \{0,1\} \cup \{0,1\}^2 \cup \{0,1\}^3 \cup \cdots \cup \{0,1\}^\ell,$$

   donde ε es la palabra vacía, y sea $\mathcal{K} = \{0,1\}^{\ell+1}$. Defina un espacio de textos cifrados $\mathcal{C}$ que sea subconjunto de $\{0,1\}^*$, y un esquema criptográfico (Gen, Enc, Dec) sobre $\mathcal{K}$, $\mathcal{M}$ y $\mathcal{C}$ que sea perfectamente secreto.

   Nota: En la definición de (Gen, Enc, Dec) debe suponer que Gen es la distribución uniforme sobre $\mathcal{K}$.

3. Considere el juego que define una PRP con una ronda sobre OTP. Demuestre que todo adversario tiene una probabilidad de ganar de exactamente $1/2$.

   Bonus: ¿Es cierto lo anterior si consideramos cualquier protocolo que satisface la propiedad de perfect secrecy? Demuestre o de un contraejemplo.

4. En ayudantía fue demostrado que si una función de hash es resistente a colisiones, entonces esta función debe ser resistente a preimagen. En esta pregunta usted debe demostrar que la implicación inversa no es cierta. Vale decir, suponiendo que existe una función de hash que es resistente a preimagen, demuestre que existe una función de hash (Gen, h) que es resistente a preimagen y no es resistente a colisiones.