



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

Criptografía y Seguridad Computacional - IIC3253

Tarea 2

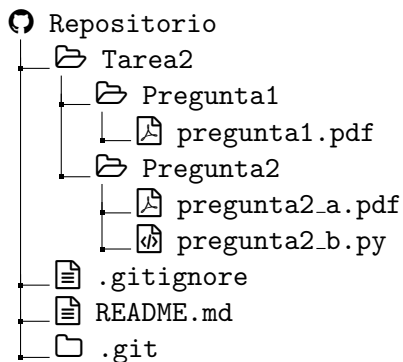
Plazo de entrega: martes 20 de mayo

Instrucciones

Cualquier duda sobre la tarea se deberá hacer en los *issues* del [repositorio del curso](#). Los issues son 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 automáticamente una vez que haga esto y lo podrá encontrar junto a los [repositorios del curso](#). Para la corrección se utilizará Python 3.11.

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



Para cada problema cuya solución se deba entregar como un documento (en este caso la pregunta 1), 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. Sea $\{h^n\}_{n \in \mathbb{N}}$ una familia de funciones de compresión resistente a colisiones tal que $h^n : \{0, 1\}^{2n} \rightarrow \{0, 1\}^n$. Supondremos también que esta familia es *puzzle friendly*, lo que significa que no existe un algoritmo eficiente que es capaz de encontrar una palabra x que resuelve un puzzle $h^n(u||x) = v$. Formalmente, un puzzle es un par $(u, v) \in \{0, 1\}^{2n}$, y una solución para el puzzle es una palabra $x \in \{0, 1\}^n$ tal que $h(u||x) = v$. Si existe tal x , se dice que el puzzle (u, v) tiene solución. Con esta notación, la familia $\{h^n\}_{n \in \mathbb{N}}$ se dice *puzzle friendly* si para cada algoritmo aleatorizado \mathcal{A} tal que:

- con entrada $(u, v) \in \{0, 1\}^{2n}$, el algoritmo \mathcal{A} retorna una palabra $\mathcal{A}(u, v) \in \{0, 1\}^n$ o el símbolo \perp ,
- \mathcal{A} realiza $o(n \cdot 2^n)$ operaciones □ para cada entrada $u, v \in \{0, 1\}^n$,

se tiene que la siguiente función de n es despreciable:

$$\max_{v \in \{0, 1\}^n} \Pr_{u \sim \mathbb{U}(\{0, 1\}^n)} [\mathcal{A} \text{ soluciona el puzzle } (u, v)],$$

donde $u \sim \mathbb{U}(\{0, 1\}^n)$ indica que u es escogido al azar con distribución uniforme desde el conjunto $\{0, 1\}^n$, y \mathcal{A} soluciona el puzzle (u, v) si $\mathcal{A}(u, v) \in \{0, 1\}^n$ y $h(u||\mathcal{A}(u, v)) = v$ en caso de que el puzzle (u, v) tenga solución, y $\mathcal{A}(u, v) = \perp$ en caso de que el puzzle (u, v) no tenga solución.

A partir de la familia $\{h^n\}_{n \in \mathbb{N}}$, definimos el protocolo **EstablecerClave**(1^n) que permite a dos usuarios A y B establecer una clave de n bits en un canal público, sin la necesidad de juntarse físicamente.

EstablecerClave(1^n)

- (1) A escoge con distribución uniforme $s \in \{0, 1\}^n$, y se lo envía a B .
- (2) Sea P el conjunto de las primeras n^2 palabras en $\{0, 1\}^n$, ordenadas por orden lexicográfico (definido por $0 < 1$), y sea $m = n \cdot \lceil \log n \rceil$. Por ejemplo, si $n = 5$, entonces $P = \{00000, 00001, 00010, \dots, 10110, 10111, 11000\}$ y $m = 15$.
 - (2.1) A escoge m palabras distintas x_1, \dots, x_m desde el conjunto P , calcula $a_i = h(s||x_i)$ para cada $i \in \{1, \dots, m\}$, y envía $(1, a_1), \dots, (m, a_m)$ a B .
 - (2.2) B escoge m palabras distintas y_1, \dots, y_m desde el conjunto P , calcula $b_i = h(s||y_i)$ para cada $i \in \{1, \dots, m\}$, y envía $(1, b_1), \dots, (m, b_m)$ a A .
- (3) Sea $I = \{(i, j) \mid a_i = b_j\}$. Si $I = \emptyset$, entonces el protocolo falla. En otro caso, sea (k, ℓ) el menor elemento en I en el orden lexicográfico sobre $\{1, \dots, m\}^2$ definido por $1 < 2 < \dots < m$.
 - (3.1) A establece como clave x_k .
 - (3.2) B establece como clave y_ℓ (que debería ser igual a x_k).

¹Recuerde que una función $f(n)$ es $o(g(n))$ si se cumple que $(\forall c \in \mathbb{R}, c > 0)(\exists n_0 \in \mathbb{N})(\forall n \in \mathbb{N}, n \geq n_0)(f(n) \leq c \cdot g(n))$.