DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
ESCUELA DE INGENIERÍA
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

IIC3253 - Criptografía y Seguridad Computacional (I/2023)

Ayudantía 3

Ayudantes: Susana Figueroa (sfigueroa3@uc.cl)

Pregunta 1: Funciones despreciables

(a) Muestre que 2^{-n} y $n^{-log(n)}$ son funciones despreciables.

Solución:

Una función $f : \mathbb{N} \to \mathbb{R}$ es despreciable si:

$$\big(\forall \text{polinomio } p\big)\big(\exists n_0 \in \mathbb{N}\big)\big(\forall n \geq n_0\big)\left(f(n) < \frac{1}{p(n)}\right)$$

Sin perdida de generalidad, podemos decir que para todo polinomio $p(n)$ existe un monomio n^c, tal que $p(n) < n^c$. Entonces al reescribir la definición nos queda como,

$$\big(\forall \text{polinomio } p\big)\big(\exists n_0 \in \mathbb{N}\big)\big(\forall n \geq n_0\big)\Big(f(n) < n^{-c}\Big)$$

1. $f(n) = 2^{-n}$

   Queremos demostrar que, existe un $n_0 \in \mathbb{N}$, tal que para todo $n \geq n_0$ se cumple que,

   $$2^{-n} < n^{-c}$$

   Podemos reemplazar 2^{-n} por,

   $$\begin{aligned} 2^{-n} = \big(2^{log_2 n}\big)^{\frac{-n}{log_2 n}} &< n^{-c} \\ \big(n\big)^{\frac{-n}{log_2 n}} &< n^{-c} \end{aligned}$$

   Calculamos el $log_n(\cdot)$ de cada lado,

   $$\begin{aligned} \frac{-n}{log_2 n} &< -c \\ \frac{n}{log_2 n} &> c \end{aligned}$$

Sabemos que $n \geq n_0$,

$$\begin{aligned} \frac{n}{log_2 n} \geq \frac{n_0}{log_2 n_0} &> c \\ \frac{n_0}{log_2 n_0} \geq \frac{n_0}{\sqrt{n_0}} &> c \\ \frac{n_0}{\sqrt{n_0}} = \sqrt{n_0} &> c \\ n_0 &> c^2 \end{aligned}$$

Entonces, para cualquier polinomio, existe un $n_0 > c^2$, tal que $2^{-n} \leq \frac{1}{p(n)}$. Por lo tanto, 2^{-n} es despreciable.

2. $f(n) = n^{-log(n)}$

   Queremos demostrar que, existe un $n_0 \in \mathbb{N}$, tal que para todo $n \geq n_0$ se cumple que,

$$n^{-log(n)} < n^{-c}$$

   Calculamos el $log_n(\cdot)$ de cada lado,

$$\begin{aligned} -log(n) &< -c \\ log(n) &> c \\ n &> 10^c \end{aligned}$$

   Sabemos que $n \geq n_0$, reemplazamos,

$$\begin{aligned} n \geq n_0 &> 10^c \\ n_0 &> 10^c \end{aligned}$$

   Entonces, para cualquier polinomio, existe un $n_0 > 10^c$, tal que $n^{-log(n)} \leq \frac{1}{p(n)}$. Por lo tanto, $n^{-log(n)}$ es despreciable.

(b) Demuestre que si f y g son funciones despreciables, entonces $f + g$ y $f \cdot g$ son funciones despreciables.

Solución:

1. $f + g$

   La idea es que la suma de dos funciones que son despreciables, sigue siendo despreciable. Si sumas dos funciones que son exponencialmente pequeñas, no puede sumar algo que es polinomialmente pequeño, entonces la suma es despreciable.

   Consideremos $f(n)$ y $g(n)$ dos funciones despreciables y $h(n) = f(n) + g(n)$.

   Como $f(n)$ y $g(n)$ son despreciables, existe un n_f y un n_g tal que,

   $$\forall n \geq n_f, \quad f(n) \leq n^{-(c+1)}$$

   Y,

   $$\forall n \geq n_g, \quad g(n) \leq n^{-(c+1)}$$

   Elegimos $n_0 = max(n_f, n_g, 2)$ (ya vamos a ver después por que el 2 esta ahí).

   Entonces, para cualquier $n \geq n_0$,

   $$\begin{aligned} h(n) = f(n) + g(n) &\leq n^{-(c+1)} + n^{-(c+1)} \\ h(n) &\leq 2n^{-(c+1)} \end{aligned}$$

   Como dijimos que $2 \leq n_0 \leq n$.

   $$\begin{aligned} h(n) &\leq 2n^{-(c+1)} \\ h(n) \leq 2n^{-(c+1)} &\leq n \cdot n^{-(c+1)} \\ h(n) \leq n \cdot n^{-(c+1)} &= n^{-c} \\ h(n) &\leq n^{-c} \end{aligned}$$

   Entonces,

   $$\begin{aligned} h(n) = f(n) + g(n) &\leq n^{-(c+1)} + n^{-(c+1)} \\ &= 2n^{-(c+1)} \\ &\leq n \cdot n^{-(c+1)} = n^{-c} \\ h(n) &\leq n^{-c} \end{aligned}$$

   Por lo que la suma de dos funciones despreciables, también es despreciable.

2. $f \cdot g$

Consideremos $f(n)$ y $g(n)$ dos funciones despreciables y $h(n) = f(n) \cdot g(n)$.

Como $f(n)$ y $g(n)$ son despreciables, existe un n_f y un n_g tal que,

$$\forall n \geq n_f, \quad f(n) \leq n^{-(\frac{c}{2})}$$

Y,

$$\forall n \geq n_g, \quad g(n) \leq n^{-(\frac{c}{2})}$$

Spdg. consideramos que c es par.

Elegimos $n_0 = max(n_f, n_g)$.

Entonces, para cualquier $n \geq n_0$,

$$\begin{aligned} h(n) = f(n) \cdot g(n) &\leq n^{-(\frac{c}{2})} \cdot n^{-(\frac{c}{2})} \\ h(n) &\leq n^{-(\frac{c}{2})-(\frac{c}{2})} \\ h(n) &\leq n^{-c} \end{aligned}$$

Por lo que la multiplicación de dos funciones despreciables, también es despreciable.

Pregunta 2: Hash-Col

[2022 - Tarea 1] Considere el juego *Hash-Col*(n) mostrado en clases para definir la noción resistencia a colisiones. Utilizando este tipo de juegos, defina la noción de resistencia a preimagen para una función de hash (Gen, h). Además, demuestre que si (Gen, h) es resistente a colisiones, entonces (Gen, h) es resistente a preimagen.

Solución:

Considere una función de hash (Gen, h) tal que si $Gen(1^n) = s$, entonces $h^s : \{0,1\}^* \rightarrow \{0,1\}^{\ell(n)}$ donde $\ell(n)$ es un polinomio fijo. Además, suponga que h se puede calcular en tiempo polinomial en el largo de la entrada, vale decir, $h(m)$ se puede calcular en tiempo $O(|m|^c)$ para alguna constante fija c. Definimos un juego *Hash-Pre-Img*(n) dado por los siguientes pasos:

1. El verificador genera $s = Gen(1^n)$ y un hash $x \in \{0,1\}^{\ell(n)}$
2. El adversario elige $m \in \{0,1\}^*$ o $m = \perp$

3. El adversario gana el juego si alguna de las siguientes condiciones se cumple:
   - $m \in \{0,1\}^*$ y $h^s(m) = x$
   - $m = \perp$ y no existe $m' \in \{0,1\}^*$ tal que $h^s(m') = x$

   En caso contrario, el adversario pierde.

Además, decimos que (Gen, h) es resistente a preimagen si para todo adversario que funciona como un algoritmo aleatorizado de tiempo polinomial, la función
Pr(Adversario gane *Hash-Pre-Img*(n)) es despreciable (nótese que esta es una función de n).

Vamos a demostrar que si (Gen, h) es resistente a colisiones, entonces (Gen, h) es resistente a preimagen. De manera más precisa, vamos a hacer esto considerando el contrapositivo, vale decir, vamos a mostrar que si (Gen, h) no es resistente a preimagen, entonces (Gen, h) no es resistente a colisiones.

Suponga que (Gen, h) no es resistente a preimagen. Entonces existe un adversario $\mathcal{A}$ tal que $\mathcal{A}$ es un algoritmo aleatorizado de tiempo polinomial y $\Pr(\mathcal{A}$ gane *Hash-Pre-Img*(n)) no es una función despreciable. A partir del algoritmo $\mathcal{A}$, vamos a definir un algoritmo aleatorizado $\mathcal{B}$ tal que $\mathcal{B}$ funciona en tiempo polinomial y $\Pr(\mathcal{B}$ gane *Hash-Col*(n)) no es una función despreciable. Suponga que $\mathcal{A}$ funciona en tiempo $p(n)$, donde $p(n)$ es un polinomio fijo. Dado $s = \mathit{Gen}(1^n)$, el algoritmo $\mathcal{B}$ construye $m' = 0^{p(n)+1}$ (vale decir, m' tiene $p(n) + 1$ símbolos 0), se pone en el papel del verificador en el juego *Hash-Pre-Img*(n) y define $x = h^s(m')$ (nótese que $x \in \{0,1\}^{\ell(n)}$). Una vez que el algoritmo $\mathcal{A}$ responde con un mensaje $m \in \{0,1\}^*$ en el juego *Hash-Pre-Img*(n), el algoritmo $\mathcal{B}$ retorna el par de mensajes m, m'.

Dado que el mensaje m' es de largo $p(n)+1$, la función de hash h se puede calcular en tiempo polinomial (en el largo de la entrada) y $\mathcal{A}$ es un algoritmo aleatorizado de tiempo polinomial, se tiene que $\mathcal{B}$ es un algoritmo aleatorizado de tiempo polinomial. Para terminar la demostración solo necesitamos mostrar que $\Pr(\mathcal{B}$ gane *Hash-Col*(n)) no es una función despreciable. Nótese que si $\mathcal{A}$ gana el juego *Hash-Pre-Img*(n), entonces $\mathcal{A}$ genera un mensaje $m \in \{0,1\}^*$ tal que $h(m) = x$ (ya que $x = h(m')$ con $m' \in \{0,1\}^*$). Además, el algoritmo $\mathcal{A}$ ejecuta a lo más $p(n)$ pasos, por lo que $|m| \leq p(n)$ y se puede concluir que $m \neq m'$ ya que $|m'| = p(n) + 1$. Así, tenemos que si $\mathcal{A}$ retorna un mensaje $m \in \{0,1\}^*$ tal que $h(m) = x$, entonces m, m' es una colisión para la función de hash (Gen, h) y $\mathcal{B}$ gana el juego *Hash-Col*(n). En términos de probabilidades, lo que concluimos es que:

$$\Pr(\mathcal{B} \text{ gane } \textit{Hash-Col}(n)) \quad = \quad \Pr(\mathcal{A} \text{ gane } \textit{Hash-Pre-Img}(n)).$$

De esta forma, se deduce que $\Pr(\mathcal{B}$ gane *Hash-Col*(n)) es una función no despreciable, puesto que $\Pr(\mathcal{A}$ gane *Hash-Pre-Img*(n)) es una función no despreciable. Esto concluye la demostración de la propiedad.