Criptografía y Seguridad Computacional - IIC3253
Tarea 1
Solución pregunta 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.

Solución. Suponga que (Gen, h') es una función de hash resistente a preimagen. En particular, para cada $n \geq 0$, si $Gen(1^n) = s$, entonces $(h')^s : \{0,1\}^* \to \{0,1\}^{\ell(n)}$ donde $\ell(n)$ es un polinomio fijo. A partir de esta función, definimos una función de hash (Gen, h) de la siguiente forma. Suponiendo que $n \geq 0$ y $Gen(1^n) = s$, para cada $m \in \{0,1\}^*$ se tiene que:

$$h^s(m) \quad = \quad \begin{cases} (h')^s(\varepsilon) & \text{si } m = \varepsilon \\ (h')^s(u) & \text{si } m = uv \text{ con } |v| = 1 \end{cases}$$

Vamos a demostrar que (Gen, h) es resistente a preimagen y no es resistente a colisiones.

Suponga primero que (Gen, h) no es resistente a preimagen, de lo cual esperamos llegar a una contradicción. Dado que (Gen, h) no es resistente a preimagen, existe un algoritmo aleatorizado $\mathcal{A}$ de tiempo polinomial que gana el siguiente juego con una probabilidad no despreciable. Dado $n \geq 0$, se ejecutan 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.

A partir del algoritmo $\mathcal{A}$, definimos un algoritmo aleatorizado $\mathcal{A}'$ de la siguiente forma. Dado $n \geq 0$ y $x \in \{0,1\}^{\ell(n)}$, el algoritmo $\mathcal{A}'$ se pone en el papel del verificador en el juego anterior, y le pide a $\mathcal{A}$ una preimagen para x. Si $\mathcal{A}$ responde con $m = \perp$ o $m = \varepsilon$, entonces $\mathcal{A}'$ responde con el mismo string m como una preimagen para x bajo la función (Gen, h'). Si $\mathcal{A}$ responde con $m = uv$ con $m \in \{0,1\}^*$ y $|v| = 1$, entonces entonces $\mathcal{A}'$ responde con u como una preimagen para x bajo la función (Gen, h'). Tenemos que $\mathcal{A}'$ es un algoritmo aleatorizado de tiempo polinomial ya que $\mathcal{A}$ es un algoritmo aleatorizado de tiempo polinomial. Además, $\mathcal{A}'$ genera una preimagen de x con la misma probabilidad que $\mathcal{A}$, puesto que por definición de h tenemos que:

- si $m = \perp$ y no existe m' tal que $h^s(m') = x$, entonces no existe m' tal que $(h')^s(m') = x$;
- si $m = \varepsilon$ y $h^s(m) = x$, entonces $(h')^s(\varepsilon) = h^s(\varepsilon) = x$; y

- si $m = uv$, con $m \in \{0,1\}^*$ y $|v| = 1$, y $h^s(m) = x$, entonces $(h')^s(u) = h^s(uv) = x$.

La existencia del algoritmo $\mathcal{A}'$ nos muestra que la función de hash (Gen, h') no es resistente a preimagen, lo cual contradice nuestro supuesto inicial.

Para demostrar que (Gen, h) no es resistente a colisiones, nos basta considerar que si $n \geq 0$ y $Gen(1^n) = s$, entonces $h^s(0) = h^s(1) = (h')^s(\varepsilon)$. Esto concluye el ejercicio.