Formalizando la noción de resistencia a colisiones

Considere una función de hash (Gen, h)

Definimos el juego $\mathit{Hash}\text{-}\mathit{Col}(n)$:

1. El verificador genera $s = Gen(1^n)$, y se lo entrega al adversario
2. El adversario genera al azar m_1 y m_2 con $m_1 \neq m_2$
3. El adversario gana el juego si $h^s(m_1) = h^s(m_2)$, y en caso contrario pierde

Formalizando la noción de resistencia a colisiones

Una función de hash (Gen, h) se dice resistente a colisiones si para todo adversario que funciona en tiempo polinomial, existe una función despreciable $f(n)$ tal que:

$$\Pr(\text{Adversario gane } \mathit{Hash}\text{-}\mathit{Col}(n)) \leq f(n)$$

Nótese que el adversario es un algoritmo *aleatorizado* de tiempo polinomial

¿Cómo se formaliza la noción de ser resistente a preimagen usando las ideas anteriores?

Usted va a contestar esta pregunta en la tarea 1

Y además usted va a demostrar que ser resistente a colisiones implica ser resistente a preimagen

¿Cuántas propiedades más vamos a definir para las funciones de hash?

¿Cuál es la propiedad más fuerte que podríamos pedir?

Como la propiedad límite esperamos que una función de hash se vea como un **random oracle**

- Aunque en la práctica no podamos lograr esto

El modelo de random oracle

Considere una función de hash $h : \mathcal{M} \to \mathcal{H}$

h es un random oracle si para cada secuencia de mensajes m_1, ..., m_k

- Si $m_i \neq m_j$ para cada $j < i$, entonces $h(m_i)$ es escogido al azar con distribución uniforme desde $\mathcal{H}$
- Si $m_i = m_j$ para $j < i$, entonces $h(m_i) = h(m_j)$

El modelo de random oracle

Esperamos que las funciones de hash que vamos a definir **no puedan ser distinguidas** de un random oracle

Construyendo una función de hash

Primero vamos a definir una función de hash de largo fijo

- Los mensajes en $\mathcal{M}$ tienen largo fijo

Después vamos a ver una forma general que permite utilizar de manera iterativa las funciones de hash de largo fijo para construir funciones de hash para mensajes de largo arbitrario

- En esta construcción, las funciones de hash de largo fijo son llamadas *funciones de compresión*

Una función de compresión

Queremos construir una función de hash de largo fijo:

$$h : \{0,1\}^{2n} \to \{0,1\}^n$$

Para construir esta función suponemos que tenemos un esquema criptográfico (Gen, Enc, Dec) sobre los espacios $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0,1\}^n$

- Tenemos que $Enc : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$

La construcción de Davies–Meyer

Dados $u, v \in \{0,1\}^n$, defina $h(uv) = Enc(u, v)$

Este es nuestro primer intento para definir una función de hash $h : \{0,1\}^{2n} \rightarrow \{0,1\}^n$

- ¿Es esta una buena alternativa?

Muestre que h no es resistente a preimagen

La construcción de Davies–Meyer

Un segundo intento $h(uv) = Enc(u, v) + u$

- El símbolo $+$ representa a la suma en módulo 2, dado que estamos operando con bits

Esta es la construcción de Davies–Meyer

- Se puede demostrar que h es resistente a colisiones si (Gen, Enc, Dec) es un esquema criptográfico *ideal*