Fixed Length Collision Resistant Hash Functions

Sarthak Mahajan, 2018111005

March 7, 2022

1 Hash functions:

- Hash functions are functions which compress strings of arbitrary length to a shorter length.
- We deal with a family of functions indexed by $s, H^s(x) = H(s, x)$
- A hash function is a pair of algorithms (Gen, H) where Gen (1^n) outputs the index s (for choosing $\mathrm{H^s}$)

2 Collision Resistance:

- Collisions for a hash function H occurs when $H(x) = H(y)$ when $x \neq y$, where x,y are two distinct input strings to the hash function.
- A hash function H is collision resistant if it is infeasible for any probabilistic polynomial-time algorithm to find a collision in H.
- If $\mathrm{H^s}$ is defined only for inputs x of a certain length, we say it is a **fixed length hash function**

3 Definition of Collision resistant Hash function:

3.1 Hash Setup:

The Hashing algorithm (Gen,H), sends the Hash function to the Adversary A, if A can find out x and y such that: $x \neq y$ and $H^s(x) = H^s(y)$, then and only then does the Hash setup output a 1.

3.2 Condition for security:

A hash function $(\mathrm{Gen}, \mathrm{H})$ is collision resistant if for all probabilistic polynomial time adversaries A:

$$\Pr[\text{ Hash-Setup output} = 1] \leq \text{ negl } (n)$$

4 Fixed Length Collision Resistant Hash Function from DLP:

4.1 Algorithm:

- Let P be a polynomial time algorithm that on input 1^n outputs a cyclic group G of order q (length of q is n) and generator g
- Define a fixed-length hash function (Gen, H) as follows:
- Gen: on input 1^n, run $\mathcal{P}(1^n)$ to obtain (G, q, g) and then select $h \leftarrow G$ randomly. Output $s := \langle G, q, g, h \rangle$ as the key. **(q's length is the same as the length of the key)**
- H : given a key $s = \langle \mathbb{G}, q, g, h \rangle$ and input $(x_1, x_2) \in \mathbb{Z}_q \times \mathbb{Z}_q$, output $H^s(x_1, x_2) := g^{x_1} h^{x_2}$.

4.2 Explanation:

4.2.1 THEOREM:

If the discrete logarithm problem is hard relative to $\mathcal{P}$, then the above Construction is a fixed-length collision-resistant hash function.

4.3 PROOF:

Lets assume that the Hash function gets a collision, i.e. $(x_1, x_2) \neq (x_1', x_2')$

$$\begin{aligned} H^s(x_1, x_2) = H^s(x_1', x_2') &\Rightarrow g^{x_1} h^{x_2} = g^{x_1'} h^{x_2'} \\ &\Rightarrow g^{x_1 - x_1'} = h^{x_2' - x_2} \end{aligned}$$

Since, $(x_1, x_2) \neq (x_1', x_2')$, atleast one of the pairs: (x_1, x_1') (x_2, x_2'), is not equal, lets assume that the pair that's definitely unequal is: (x_2, x_2'). (Note that the proof would follow symmetrically with Δ being defined on (x_1, x_1') if they were the definitely unequal pair)

$$\Delta \stackrel{\text{def}}{=} x_2' - x_2$$

$$g^{\left(x_1 - x_1'\right) \cdot \Delta^{-1}} = \left(h^{x_2' - x_2}\right)^{\left[\Delta^{-1} \bmod q\right]} = h^{\left[\Delta \cdot \Delta^{-1} \bmod q\right]} = h^1 = h$$

Since, the DLP is assumed as hard relative to P, thus its also hard to find this collision. Hence the Hash function H is collision resistant.