1 Hash functions

1.1 What is hashing?

Hashing is essentially the process of running data through a formula that creates a hash as a result. The result is mostly of fixed length, despite the varying size of input data. A hash is designed to act as a one-way function.

1.2 Collision resistance

a hash collision is a random match in hash values that occurs when a hashing algorithm produces the same hash value for two distinct pieces of data. This is a dangerous scenario we would like to avoid, as otherwise the adversary can by-pass the authentication method.
Collisions are unavoidable, as every hash function with more inputs than outputs. Considering a hash function like SHA-256, which generates 256 bits of output from a huge input, the pigeonhole principle ensures that some inputs hash to the same result because it must generate one of 2256 outputs for each member of a much bigger set of inputs. Collision resistance does not suggest that there are no collisions; rather, it implies that they are hard to find.
LEMMA: Fix a positive integer N, and say $q \leq \sqrt{2N}$ elements $y_1, \ldots, y_9$ are chosen uniformly and independently at rondom from a set of size N. Then the probability that there exist distinct i, j with $y_i = y_j$ is at least $\frac{q(q-1)}{4N}$. That is,

$$\mathrm{col}(q, N) \geq \frac{q(q-1)}{4N}$$

PROOF:
$\Pr[\mathrm{NoCol}_q] = \Pr[\mathrm{NoColl}_1] \cdot \Pr[\mathrm{NoColl}_2 \mid \mathrm{NoColl}_1] \cdots \Pr[\mathrm{NoCol}_q \mid \mathrm{NoCol}_{q-1})$

$\Pr[\mathrm{NoColl}_1] = 1$

$\Pr[\mathrm{NoColl}_{i+1} \mid \mathrm{NoColl}_i] = 1 - \frac{i}{N}$

$\Pr[\mathrm{NoColColl}_q] = \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right)$

For all x *with* $0 \leq x \leq 1$ *it holds that*

$$e^{-x} \leq 1 - \left(1 - \frac{1}{e}\right) \cdot x \leq 1 - \frac{x}{2}.$$

$\Pr[\mathrm{NoColl}_q] \leq \prod_{i=1}^{q-1} e^{-i/N} = e^{-\sum_{i=1}^{q-1}(i/N)} = e^{-q(q-1)/2N}$

$\Pr[\text{ Coll }] = 1 - \Pr[\mathrm{NoColl}_q] \geq 1 - e^{-q(q-1)/2N} \geq \frac{q(q-1)}{4N}$

1.2.1 Building a Fixed length Collision Resistance Hash Function

Let G be as described in the text. Define a fixed-length hash function (Gen, H) as follows:

- Gen: on input 1^n, run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$ and then select $h \leftarrow G$. Output $s := (\mathbb{G}, q, g, h)$ as the key.

- H: given a key $s = (\mathbb{G}, q, g, h)$ and input $(x_1, x_2) \in \mathbb{Z}q \times \mathbb{Z}q$, output $H^s(x_1, x_2) := g^{x_1} h^{x_2}$.

If we assume that DLP is hard, we can proof that the above method of hash construction is collision resistant, as follows:

$H^s(x_1, x_2) = H^s(x_1', x_2')$

$\Rightarrow g^{x_1} h^{x_2} = g^{x_1'} h^{x_2'}$

$\Rightarrow g^{x_1 - x_1'} = h^{x_2' - x_2}$.

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

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