# CRYPTOGRAPHY HASHING

A cryptographic hash function is a mathematical algorithm that maps data of arbitrary size to a bit array of fixed size. All the operations are public - such as the h(x) hash-function. No private keys and it is deterministic and random**.**

| **H : { 0, 1 }*** | **–>** | **{ 0, 1 }$^d$** |
|---|---|---|
| Data of arbitrary size | | Fixed Length string of d bits |

Examples :
MD5 - d is 128 bits.
SHA1 - d is 160 bits.
SHA2 - d is 256 bits.

## Properties of Hashing

1.**Preimage resistance**

In 1976, Diffie and Hellman introduced the use of a one-way hash function in cryptography. A one-way function is a computation function wherein its direction is straight forward, but if the computation reverses in direction, it becomes more difficult. Preimage resistance is in line with a one-way function, which makes it relatively easy to protect a file.
It is exponentially hard (computationally infeasible) to determine the m message from the h(m) message digest.

2. **Second preimage resistance**

If m1 is given then it is infeasible to find m2 such that h(m1) = h(m2).
Many times, second preimage resistance is mistaken as first preimage resistance because of the similarities they share. For instance, imagine a hash function is given and it is second preimage resistance, but the resistance is not preimage. The outcome of such a result might be contradictory which implies that you will have to get preimage resistance before you can get second preimage resistance.

3. **Collision Resistance**

It is infeasible to find m1 and m2 such that h(m1) = h(m2) .
Since a hash function is a compressing function with fixed hash length, it is impossible for a hash function not to have collisions. This property of collision free only confirms that these collisions should be hard to find. This property makes it very difficult for an attacker to find two input values with the same hash. Also, if a hash function is collision-resistant then it is second preimage resistant.

Birthday Paradox

Let's assume there are N people in a given room. What is the probability that nobody shares a common birthday?

$$P(N) = \prod (365\text{-}i)/365 \ (i \text{ from } 0 \text{ to } N\text{-}1)$$

Let's assume there are N people in a given room. What is the probability that at least 2 people share the same birthday?

$$p(N) = 1 - P(N)$$

For p(N) to be greater than 0.5, N> 23 ( $\approx \sqrt{365}$ ). It turns out that even with a small group of people it is quite common for at least two people to share the same birthday.

Rule Of Thumb:

If there are N different possibilities of something Then you need $\sqrt{N}$ randomly chosen items in order to have a 50% chance of collision**!**

**It is easier to break collision resistance than the second preimage resistance.**

Proof :

Let us consider a hash function h(m) of output length 128 bits. To break second preimage resistance, we need to find m2 such that it, h(m2) = h(m1). The running time for this approach will be one iteration in the best case scenario, and $2^{128}$ (total possible hash values) iterations In the worst case scenario. In an average case scenario it will take $2^{128}/2$ iterations, it is a huge number, So we may come to the conclusion that with the help of brute force approach, we are not able to break the cryptographic hash functions because these would take thousands and thousands of years**!.**

Breaking the collision resistance property would be easier because we just have to find any m1 and m2 such that h(m1) = h(m2).

"IT MAY SEEMS TO BE AS MUCH COMPLICATED BUT IT IS AN EASIER TASK BECAUSE OF THE **BIRTHDAY PARADOX**"

From above, how many hashes do I need to generate before a hash collision occurs with 50% probability? $\sqrt{2^{128}} = 2^{64}$.

Hence, there is a relatively high probability a hash collision may happen.

**Problem:**

Let $h : \{0, 1\}^* \rightarrow \{0, 1\}^n$ be a hash function that is second-preimage and collision resistant. Let $h' : \{0, 1\}^* \rightarrow \{0, 1\}^{n+1}$ be the hash function given by the rule

$h'(x) = 0||x \ x \in \{0, 1\}^n,$

$1||h(x)$ otherwise. ( || = OR )

Prove that h′ is not preimage resistant, but still second-preimage and collision resistant.

MD5 -

https://www.geeksforgeeks.org/what-is-the-md5-algorithm/

https://www.geeksforgeeks.org/md5-hash-python/

SHA -

https://www.encryptionconsulting.com/education-center/what-is-sha/#:~:text=SHA%20stands

%20for%20secure%20hashing,modular%20additions%2C%20and%20compression%20functions.
https://www.geeksforgeeks.org/sha-in-python/