Hash Functions & Birthday Paradox

'Unbirthdays'

Cryptographic hash functions

Hash functions

- A cryptographic hash function is a map $H : \{0,1\}^{anything} \longrightarrow \{0,1\}^n$, that take as input **arbitrarily long messages** and outputs **fixed size bit strings** (usually $n = 160, 256$).
- The hash function H should be efficiently computable and **one-way**, i.e. a hash output h it should be **infeasible** to find the original message b such that $H(m) = h$ (**pre-image resistant property**).
- For any given message m_1 it should be computationally infeasible to find $m_2 \neq m_1$ such that $H(m_1) = H(m_2)$ (**weak collision resistance property**).
- It should be computationally infeasible to find a pair of messages (m_1, m_2) such that $H(m_1) = H(m_2)$ (known as (strong) **collision resistance property**).
- The map H should be indistinguishable from a truly random function.

Properties of Hash Functions

Attention

- For any hash function H, collisions must exist (simply because *anything* $>>$ n)!!
- Also MACs map large messages into a fix-size tag. The **difference** between a **hash function** and a **MAC** is that the MAC takes also a key in input.

Attacks against Hash functions

Attack 1: given a **hash value** h, find a message m, such that $H(m) = h$.
Security if brute force is the best attack, we get n bits security (it takes $O(2^n)$ number of attempts).

Attack 2: find a collision, i.e. find m_1 and $m_2 \neq m_1$ such that $H(m_1) = H(m_2)$.
Security: By the *birthday paradox*, with high probability, you can find a collision in after $O(2^{\frac{n}{2}})$ trials!

The birthday paradox

The birthday paradox

Let $r_1, \ldots, r_k \in \{0,1\}^n$ be k random n-bit values (chosen uniformly at random).
When $k = 1.2 \cdot 2^{n/2}$ then $Prob[\exists i \neq j : r_i = r_j] \geq 1/2$

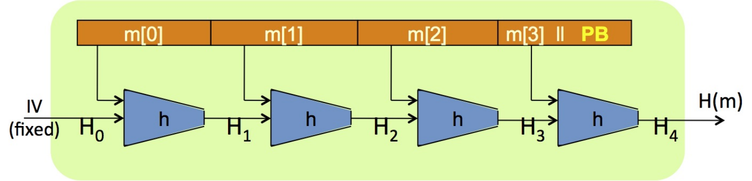
In other words, when k(=number of trials), is large enough we will find a collusion with high probability. The paradox lies in that k is smaller than what you expect!

Example

Let $n = 128$, by the birthday paradox, after sampling about 2^{64} **random** messages from $\{0,1\}^{128}$, it is very likely that two sampled messages have the same hash value.

Question: Given a collision resistant function for *short* messages, can we construct collision resistant function for *long* messages?

The Merkle-Damgard iterated construction



Theorem: h collision resistant $\Longrightarrow$ H collision resistant.

Example of collision resistance hash function is SAH-256