Chapter 5
Hash Functions++

Hash Function Motivation

- Suppose Alice signs M
  - Alice sends M and $S = [M]_{\text{Alice}}$ to Bob
  - Bob verifies that $M = \{S\}_{\text{Alice}}$
  - Is it OK to just send S?
- If M is big, $[M]_{\text{Alice}}$ is costly to compute
- Suppose instead, Alice signs $h(M)$, where $h(M)$ is much smaller than M
  - Alice sends M and $S = [h(M)]_{\text{Alice}}$ to Bob
  - Bob verifies that $h(M) = \{S\}_{\text{Alice}}$

Crypto Hash Function

- Crypto hash function $h(x)$ must provide
  - **Compression** — output length is small
  - **Efficiency** — $h(x)$ easy to computer for any x
  - **One-way** — given a value y it is infeasible to find an x such that $h(x) = y$
  - **Weak collision resistance** — given x and $h(x)$, infeasible to find $y \neq x$ such that $h(y) = h(x)$
  - **Strong collision resistance** — infeasible to find any x and y, with $x \neq y$ such that $h(x) = h(y)$
  - Lots of collisions exist, but hard to find one

Pre-Birthday Problem

- Suppose N people in a room
- How large must N be before the probability someone has same birthday as me is $\geq 1/2$
  - Solve: $1/2 = 1 - (364/365)^N$ for N
  - Find $N = 253$

Birthday Problem

- How many people must be in a room before probability is $\geq$ 1/2 that two or more have same birthday?
    - $1 - 365/365 \cdot 364/365 \cdot \cdot \cdot (365-N+1)/365$
    - Set equal to 1/2 and solve: **N = 23**
- Surprising? A paradox?
- Maybe not: "Should be" about sqrt(365) since we compare all **pairs** x and y

Of Hashes and Birthdays

- If $h(x)$ is N bits, then 2^N different hash values are possible
- $\mathrm{sqrt}(2^N) = 2^{N/2}$
- Therefore, hash about $2^{N/2}$ random values and you expect to find a collision
- **Implication:** secure N bit symmetric key requires 2^{N-1} work to "break" while secure N bit hash requires $2^{N/2}$ work to "break"

Next
Non-crypto & Crypto Hashes, Design, Tiger, HMAC,...