# Hash

# Hash Function Motivation

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

# Hash Function Motivation

- So, Alice signs $h(M)$
  - That is, Alice computes $S = [h(M)]_{Alice}$
  - Alice then sends $(M, S)$ to Bob
  - Bob verifies that $h(M) = \{S\}_{Alice}$
- What properties must $h(M)$ satisfy?
  - Suppose Trudy finds $M'$ so that $h(M) = h(M')$
  - Then Trudy can replace $(M, S)$ with $(M', S)$
- Does Bob detect this tampering?
  - No, since $h(M') = h(M) = \{S\}_{Alice}$

# Crypto Hash Function

- Crypto hash function $h(x)$ must provide
  - **Compression** — output length is small
  - **Efficiency** — $h(x)$ easy to compute 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 ***any***

# Pre-Birthday Problem

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

# Birthday Problem

- How many people must be in a room before probability is ≥ 1/2 that any two (or more) have same birthday?
  - $1 - 365/365 \cdot 364/365 \cdot \cdot \cdot (365\text{-}N+1)/365$
  - 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
  - And there are 365 possible birthdays

# Of Hashes and Birthdays

- If $h(x)$ is $N$ bits, $2^N$ different hash values are possible
- So, if you hash about $2^{N/2}$ random values then you expect to find a collision
  - Since $\text{sqrt}(2^N) = 2^{N/2}$
- **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"
  - Exhaustive search attacks, that is

# Non-crypto Hash (1)

- Data $X = (X_0, X_1, X_2, \ldots, X_{n-1})$, each $X_i$ is a byte
- Define $h(X) = X_0 + X_1 + X_2 + \ldots + X_{n-1}$
- Is this a secure cryptographic hash?
- Example: $X = (10101010, 00001111)$
- Hash is $h(X) = 10111001$
- If $Y = (00001111, 10101010)$ then $h(X) = h(Y)$
- Easy to find collisions, so **not** secure…

# Non-crypto Hash (2)

- Data $X = (X_0, X_1, X_2, \ldots, X_{n-1})$
- Suppose hash is defined as

$$h(X) = nX_0 + (n-1)X_1 + (n-2)X_2 + \ldots + 1 \cdot X_{n-1}$$

- Is this a secure cryptographic hash?
- Note that

$$h(10101010, 00001111) \neq h(00001111, 10101010)$$

- But hash of $(00000001, 00001111)$ is same as hash of $(00000000, 00010001)$
- Not "secure", but this hash is used in the (non-crypto) application rsync

# Non-crypto Hash (3)

- Cyclic Redundancy Check (CRC)
- Essentially, CRC is the remainder in a long division calculation
- Good for detecting burst **errors**
  - Random errors unlikely to yield a collision
- But easy to construct collisions
- CRC has been mistakenly used where crypto integrity check is required (e.g., WEP)

# Popular Crypto Hashes

- **MD5** — invented by Rivest
  - 128 bit output
  - Note: MD5 collisions easy to find
- **SHA-1** — A U.S. government standard, inner workings similar to MD5
  - 160 bit output
- Many other hashes, but MD5 and SHA-1 are the most widely used
- Hashes work by hashing message in blocks