Hash functions

Hash fn: \( H: \mathbb{Z}_2^* \rightarrow \mathbb{Z}_2^k \)
- Compressing
- Efficiently computable
- Want some security property; depends on application

1) One-wayness
   - A probabilistic polynomial-time adversary
   - Given \( y = H(x) \) for random \( x \), infeasible to find \( x' \) s.t. \( H(x') = y \).

2) Preimage resistance
   - "Given any \( y \in \mathbb{Z}_2^k \), infeasible to find \( x' \) s.t. \( H(x') = y \)."
   - Wait! Does this make sense? What if I give you \( H(0) \)?
   - In fact, adversary \( A \) could have a hardwired output \( +H(0) \):
     - Then clearly not true that \( \forall y \in \mathbb{Z}_2^k \), \( \Pr[A(y) \in H^{-1}(y)] = \) small.
   - So, often we talk about hash fn families \( H = \{ h_s \}_{s \in \mathbb{Z}_2^k} \).
     - "Seed" is picked randomly
   - Then we have: \( \forall y \in \mathbb{Z}_2^k \), \( \Pr[A(s,y) \in h_s^{-1}(y)] = \) negligible.
     - (i.e., probability is over random choice of fn from family)

3) Second preimage resistance
   - "Given any \( x \), infeasible to find \( x' \neq x \) s.t. \( H(x') = H(x) \)," more precisely: \( \forall A, \forall x \in \mathbb{Z}_2^k \), \( \Pr[H_s(A(s,x)) = h_s(x)] = \) negligible.

4) Collision resistance
   - "Infeasible to find any \( (x, x') \) s.t. \( H(x) = H(x') \) and \( x \neq x' \)."

5) Random oracle

Birthday paradox: how hard is it to find collisions?
- \( \Pr[\text{two students have same birthday}] = \frac{1}{365} \) (1/2 or not?)

Floyd's cycle-finding algorithm: a better strategy than random guessing for collision finding.

Hash fn standards:
- MDS [Rivest, 1991]
  - Used in an attack in 2012! "Flame 1.05"
  - Google used SHA-1 in 2012!
  - Google "Flame 1.05"

Collisions found in 1996, 2004 so not recommended.

Floyd's cycle-finding algorithm: a better strategy than random guessing for collision finding.

Floyd's cycle-finding algorithm: a better strategy than random guessing for collision finding.

But still fine for some purposes (e.g., HMAC).

- SHA-1 [NSA, 1995]
  - Phasing out. All major browsers to stop accepting SHA-1 SSL certs by this year.
  - \( 2^{69} \)-time algo to find collisions as of 2005.

- SHA-2 [NSA, 2001]
  - Closely related to SHA-1. Widely used.
  - SHA-3 (Keccak) [5-year NIST contest, 2012]