Hash

h: {0,1}* -> {0,1}^{l}

mapping an arbitrary-length string to a fixed-length string

advantage

- efficient — polynomial time to length of input
- public — not using key

properties

- P1. (**one-way, pre-image resistance**) For an arbitrary digest **h(m)**, it is *computationally infeasible* to find message **m**
  - but it is ***not*** infeasible to find ***some*** **bits** of **m**
  - ex: **f(m)** is one-way **=> h(m_1||m_2) = m_1||f(m_2)** is one-way, **|m_1| = |m_2|**
  - — pre-image resistance is like have no idea of the origin password
- P2. (**collision resistance**) It is *computationally infeasible* to find two *distinct* messages, **m**, **m'** such that **h(m) = h(m').** — for **every** m, there is no a m' such that h(m) = h(m')
- P3. (**weak collision resistance, second pre-image resistance**) Given a message **m** (and **h(m)**), it is *computationally infeasible* to find *another* message **m' ≠ m** s. t. **h(m) = h(m').** — given a m, can't find a m' such that h(m) = h(m')
- P4. (**pseudo-random**) It is *computationally infeasible* to distinguish **h()** from a *random (oracle) function*
- P5. (**non-malleable**) Given **h(m),** it is *computationally infeasible* to find **h(m')** where **m** and **m '** are *related* (e.g., **m' = 2m**)
- P6. (**entropy preservation**) Given *pseudo-random* input **m** (*computationally infeasible* to distinguish **m** from random bits), such that **|m| = |h(m)|,** the entropy of **h(m)** must be no smaller than that of **m**.

relationships

- P2 => P3 and P3 =/=> P2
  If a hash function is *collision resistant,* then *it is second-preimage resistant.*
- P2 <=/=> P1 (**P2 => P1** when |range| << |domain| of the hash function)
  Collision resistance* and *pre-image resistance are separate properties.
- P3 <=/=> P1 (**P3 => P1** when |range| << |domain| of the hash function)
  Second pre-image resistance and pre-image resistance are separate properties

applications

- password storage — need P1
- manipulation detection codes — need P3
- hash trees — need P2
- digital signatures — need P3
- commitment protocols
  - e.g. a bid in an auction
  - need P1, P2, P5
- key derivation — need P4
- seeding Pseudo-random Number Generator (PRNG) — need P6
- …

vulnerability

1. collision finding attack
   - birthday paradox

"Birthday Paradox"

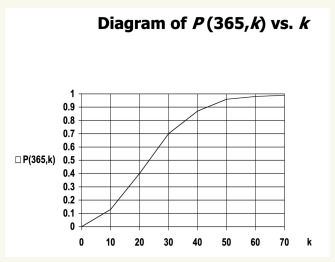
Find the minimum value of k such that:
P\{at least one pair of k people have same birthday\} = 0.5

General problem

Let $P(n,k) = P$\{there is at least a pair of duplicates among k instances of a uniformly distributed random variable with values in [1,n]\}.
Find the minimum values of k such that $P(n,k) = 0.5$.

$P(365,k)$=0.5
$Q(365,k) = P$\{no pair of people have same birthday\}=1-$P(365,n)$.
Suppose k<=365 (otherwise there are duplicates).
Let N = number of ways to choose k values in [1,365] with no duplicates.
$N = 365*364*...*(365-k+1) = 365!/(365-k)!$
The total number of ways to choose k values in [1,365] is $T = 365^k$.
Thus, $Q(365,k) = N/T = 365!/(365-k)!/365^k$, and
$P(365,k) = 1 - 365!/(365-k)!/365^k$.

Diagram of $P(365,k)$ vs. k



2. …

designing hash functions

Generating large number variants of a message

— larger than 2^128, suggested for hash function output

- generating 2^32 variants is currently feasible
- generating 2^64 is not feasible now

hash based on block ciphers

- concerns

Let **E :** {0,1}^{k} x {0,1}^{l} -> {0,1}^{l} be a **PRF** family and construct hash function
H : {0,1}^{k} x ({0,1}^{l})$^+$ -> {0,1}^{l} as follows:

The hash of message M = M_1,..,M_m is $\mathbf{H_m}$ where

(1) $\mathbf{H_0} = 0^l$, $\mathbf{H_i} = \mathbf{E_K}(\mathbf{H_{i-1}} \oplus M_i)$, for i ≥ 1 and key **K** <-$- {0,1}^{k} is public

(2) $\mathbf{H_0} = 0^l$, $M_0 = 0^l$, $\mathbf{H_i} = \mathbf{E_K}(\mathbf{H_{i-1}} \oplus M_i \oplus M_{i-1})$, for i ≥ 1 and key **K** <-$- {0,1}^{k} is public

(3) $\mathbf{H_0} = 0^l$, $M_0 = 0^l$, $\mathbf{H_i} = \mathbf{E_K}(\mathbf{H_{i-1}} \oplus M_i) \oplus M_{i-1}$, for i ≥ 1 and key **K** <-$- {0,1}^{k} is public

Assume **E** is a secure block cipher.
Are hash functions (1) – (3) secure (that is, collision resistant) ?

What about

(4) $\mathbf{H_0} = IV$, $\mathbf{H_i} = \mathbf{E_{Mi}}(\mathbf{H_{i-1}}) \oplus \mathbf{H_{i-1}}$, for i ≥ 1 and

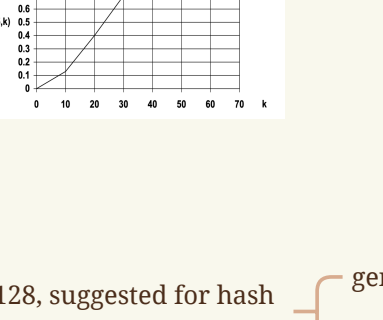
(5) $\mathbf{H_0} = IV$, $\mathbf{H_i} = \mathbf{E_{Hi-1}}(\mathbf{M_i}) \oplus \mathbf{M_i}$, for i ≥ 1 ?

  - 1) CBC — bit flip
  - 2) PCBC — swap
  - 3) IGE — swap after CPA
  - 4) secure
    5) secure
    - 7 encryption function is chosen by undetermined element, and added some noise
    - having P1, P2, P4, P6

- block ciphers are usually too small to act as a hash function

Compression Functions

- Merkle-Damgard Scheme



***h*: {0,1}^{k} × {0,1}^{n} → {0,1}^{k}** is a compression function
IV = *initialization vector, |IV| = k bits*
l* = |*m*|** encoded as an ***n-bit integer
***padded m' = m* || 0^t || *l*,** where ***t* = *n*⌈|*m*|/*n*⌉−|*m*|**
m* ' -> m_1 || m_2 || … || m_m,** where ***m* = |*m'*|/*n
$H_0 \leftarrow IV$
do *i* **to** $m \Rightarrow h_i \leftarrow h(h_{i-1}, m_i)$ **od**
output H = h_m

  - padding is necessary for this scheme to preserve collision resistance
  - 7 or the length might be shorter

- Compression Functions

Compression Functions

- **block ciphers** (does their OWF property hide all input bits?)

  E: {0,1}^{n} × {0,1}^{n} → {0,1}^{n} be a PRF/PRP family.

  Davies-Meyer:
  $\mathbf{h_0} = IV$, $\mathbf{h_i} = \mathbf{E_{m_i}}(\mathbf{h_{i-1}}) \oplus \mathbf{h_{i-1}}$, for i ≥ 1 and

  Matyas-Meyer-Oseas:
  $\mathbf{h_0} = IV$, $\mathbf{h_i} = \mathbf{E_{h_{i-1}}}(\mathbf{m_i}) \oplus \mathbf{m_i}$, for i ≥ 1

- **special functions**
  - e.g., MD-5, SHA-1, etc.

Note: padding is necessary. It allows Merkle-Damgård scheme to preserve collision resistance.

Presented with xmind