Some Project Ideas

- Read & Write about something not covered in class
  - Constructions: e.g., CCA secure PKE schemes, lattice-based PKE, more block-cipher modes, ...
  - Concepts: e.g., Key management, Double-Ratcheting, Searchable Encryption, Onion Routing/Mix-Nets, Homomorphic Encryption, ...
  - Proofs: e.g., Goldreich-Levin predicate, Fujisaki-Okamoto, security of TLS,...
- Implementation project
  - Make something
    - Slow and secure crypto (e.g., SKE and/or Digital Signatures from OWP, full-domain CRHF from DL,...)
    - Higher-level applications (e.g., "simple-TLS", Off-the-record messaging, things you can do with a block-cipher...)
    - A library with a cleaner API for encryption/authentication
  - Break something
    - e.g., use a constraint-solver to break (broken) block-ciphers

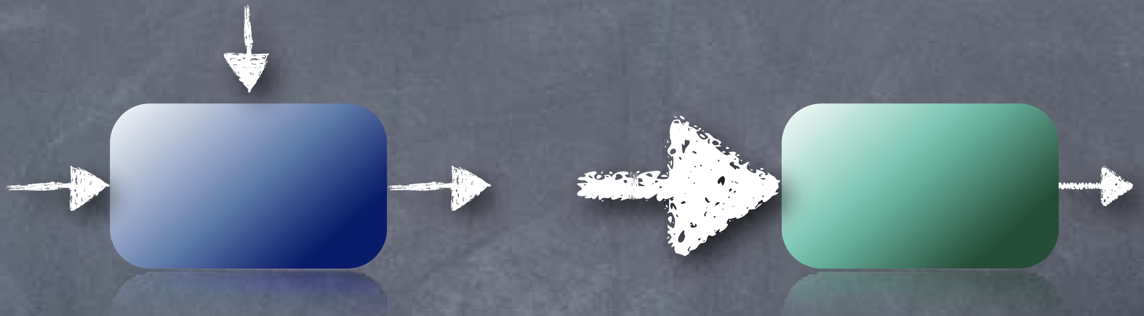
Hash Functions

Lecture 14

Flavours of collision resistance

A Tale of Two Boxes

- The bulk of today's applied cryptography works with two magic boxes
  - Block Ciphers
  - Hash Functions



- Block Ciphers: Best modeled as (strong) Pseudorandom Permutations, with inversion trapdoors
  - Often more than needed (e.g. SKE needs only PRF)
- Hash Functions:
  - Some times modelled as Random Oracles!
    - Use at your own risk! No guarantees in the standard model.
  - Today: understanding security requirements on hash functions

Hash Functions

- "Randomized" mapping of inputs to shorter hash-values
- Hash functions are useful in various places
  - In data-structures: for efficiency
    - Intuition: hashing removes worst-case effects
  - In cryptography: for "integrity"
- Primary use: Domain extension (compress long inputs, and feed them into boxes that can take only short inputs)
  - Typical security requirement: "collision resistance"
    - Different flavours: some imply one-wayness
  - Also sometimes: some kind of unpredictability

Hash Function Family

- Hash function $h:\{0,1\}^{n(k)} \rightarrow \{0,1\}^{t(k)}$
  - Compresses
- A family
  - Alternately, takes two inputs, the index of the member of the family, and the real input
- Efficient sampling and evaluation
- Idea: when the hash function is randomly chosen, "behaves randomly"
  - Main goal: to "avoid collisions". Will see several variants of the problem

x	$h_1(x)$	$h_2(x)$	$h_3(x)$	$h_4(x)$		$h_N(x)$
000	0	0	0	1	...	1
001	0	0	1	1		1
010	0	1	0	1		1
011	0	1	1	0		1
100	1	0	0	1		1
101	1	0	1	0		1
110	1	1	0	1		1
111	1	1	1	0		1

Hash Functions in Crypto Practice

- A single fixed function
  - e.g. SHA-3, SHA-256, SHA-1, MD5, MD4
  - Not a family ("unkeyed")
  - (And no security parameter knob)
- Not collision-resistant under any of the following definitions
- Alternately, could be considered as having already been randomly chosen from a family (and security parameter fixed too)
  - Usually involves hand-picked values (e.g. "I.V." or "round constants") built into the standard

Degrees of Collision-Resistance

- If for all PPT A, Pr[$x \neq y$ and $h(x)=h(y)$] is negligible in the following experiment:
  - $A \rightarrow (x,y)$; $h \leftarrow \mathcal{H}$: Combinatorial Hash Functions (even non-PPT A)
  - $A \rightarrow x$; $h \leftarrow \mathcal{H}$; $A(h) \rightarrow y$: Universal One-Way Hash Functions
  - $h \leftarrow \mathcal{H}$; $A(h) \rightarrow (x,y)$: Collision-Resistant Hash Functions
- CRHF the strongest; UOWHF still powerful (will be enough for digital signatures)
- Useful variants: A gets only oracle access to $h(\cdot)$ (weak). Or, A gets any coins used for sampling h (strong).

Degrees of Collision-Resistance

- Variants of CRHF/UOWHF where x is random
  - $h \leftarrow \mathcal{H}$; $x \leftarrow X$; $A(h,h(x)) \rightarrow y$ ($y=x$ allowed)
    - Pre-image collision resistance if $h(x)=h(y)$ w.n.p
    - i.e., $f(h,x) := (h,h(x))$ is a OWF (and h compresses)
  - $h \leftarrow \mathcal{H}$; $x \leftarrow X$; $A(h,x) \rightarrow y$ ($y \neq x$)
    - Second Pre-image collision resistance if $h(x)=h(y)$ w.n.p
  - Incomparable (neither implies the other) [Exercise]
- CRHF implies second pre-image collision resistance and, if compressing, then pre-image collision resistance [Exercise]

A.k.a One-Way Hash Function

Hash Length

- If range of the hash function is too small, not collision-resistant
  - If range poly(k)-size (i.e. hash is logarithmically long), then non-negligible probability that two random x, y provide collision
- In practice interested in minimizing the hash length (for efficiency)
  - Generic collision-finding attack: birthday attack
    - Look for a collision in a set of random hashes (needs only oracle access to the hash function)
      - Expected size of the set before collision: $O(\sqrt{|range|})$
  - Birthday attack effectively halves the hash length (say security parameter) over "naïve attack"

Universal Hashing

- Combinatorial HF: $A \rightarrow (x,y)$; $h \leftarrow \mathcal{H}$. $h(x)=h(y)$ w.n.p
- Even better: 2-Universal Hash Functions
  - "Uniform" and "Pairwise-independent"
  - $\forall x,z \ Pr_{h\leftarrow\mathcal{H}} [\ h(x)=z\ ] = 1/|Z|$ (where $h:X \rightarrow Z$)
  - $\forall x\neq y,w,z \ Pr_{h\leftarrow\mathcal{H}} [\ h(x)=w,\ h(y)=z\ ] = Pr_{h\leftarrow\mathcal{H}} [\ h(x)=w\ ] \cdot Pr_{h\leftarrow\mathcal{H}} [\ h(y)=z\ ]$
    - $\Rightarrow \forall x\neq y \ Pr_{h\leftarrow\mathcal{H}} [\ h(x)=h(y)\ ] = 1/|Z|$

x	$h_1(x)$	$h_2(x)$	$h_3(x)$	$h_4(x)$
0	0	0	1	1
1	0	1	0	1
2	1	0	0	1

Negligible collision-probability if super-polynomial-sized range

- k-Universal:
  - $\forall x_1..x_k$ (distinct), $z_1..z_k$, $Pr_{h\leftarrow\mathcal{H}} [\forall i\ h(x_i)=z_i\ ] = 1/|Z|^k$
- Inefficient example: $\mathcal{H}$ set of all functions from X to Z
  - But we will need all $h\in\mathcal{H}$ to be succinctly described and efficiently evaluable

Universal Hashing

- Combinatorial HF: $A \rightarrow (x,y)$; $h \leftarrow \mathcal{H}$. $h(x)=h(y)$ w.n.p
- Even better: 2-Universal Hash Functions
  - "Uniform" and "Pairwise-independent"
  - $\forall x \neq y, w, z$ $\Pr_{h \leftarrow \mathcal{H}}[\ h(x)=w,\ h(y)=z\ ] = 1/|Z|^2$
    - $\Rightarrow \forall x \neq y$ $\Pr_{h \leftarrow \mathcal{H}}[\ h(x)=h(y)\ ] = 1/|Z|$

x	$h_1(x)$	$h_2(x)$	$h_3(x)$	$h_4(x)$
0	0	0	1	1
1	0	1	0	1
2	1	0	0	1

Negligible collision-probability if super-polynomial-sized range

- e.g. $h_{a,b}(x) = ax+b$ (in a finite field, X=Z)
  - Uniform
    - $\Pr_{a,b}[\ ax+b = z\ ] = \Pr_{a,b}[\ b = z-ax\ ] = 1/|Z|$
  - $\Pr_{a,b}[\ ax+b = w,\ ay+b = z] = ?$ Exactly one (a,b) satisfying the two equations (for $x \neq y$)
    - $\Pr_{a,b}[\ ax+b = w,\ ay+b = z] = 1/|Z|^2$
  - But does not compress!

Universal Hashing

- Combinatorial HF: A→(x,y); h←$\mathcal{H}$. h(x)=h(y) w.n.p
- Even better: 2-Universal Hash Functions
  - "Uniform" and "Pairwise-independent"
  - $\forall x \neq y, w, z \; \Pr_{h \leftarrow \mathcal{H}} [\; h(x)=w,\; h(y)=z \;] = 1/|Z|^2$
    - $\Rightarrow \forall x \neq y \; \Pr_{h \leftarrow \mathcal{H}} [\; h(x)=h(y) \;] = 1/|Z|$

x	$h_1(x)$	$h_2(x)$	$h_3(x)$	$h_4(x)$
0	0	0	1	1
1	0	1	0	1
2	1	0	0	1

Negligible collision-probability if super-polynomial-sized range

- e.g. $h'_h(x)$ = Chop(h(x)) where h from a (possibly non-compressing) 2-universal HF
  - Chop a t-to-1 map from Z to Z'
  - e.g. with $|Z|=2^k$, removing last bit gives a 2-to-1 mapping
    - $\Pr_h$ [Chop(h(x)) = w, Chop(h(y)) = z]
      $= \Pr_h$ [h(x) = w0 or w1, h(y) = z0 or z1] $= 4/|Z|^2 = 1/|Z'|^2$