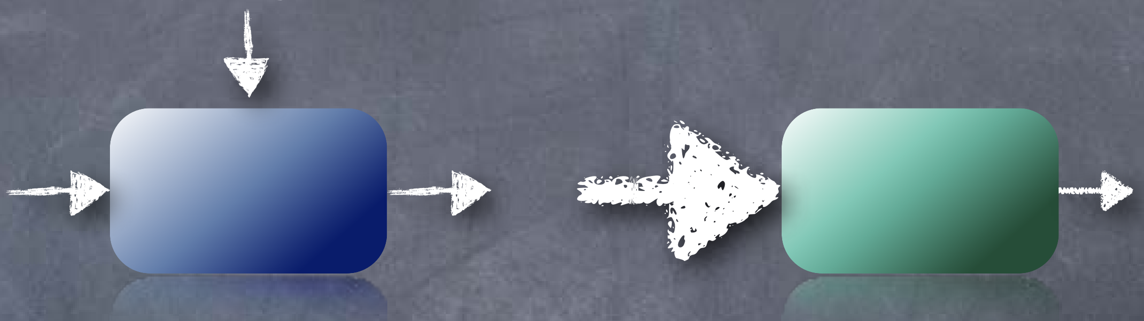
Hash Functions

Lecture 9

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

- "Randomised" 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 of theoretical interest (powerful enough for digital signatures, and can be based on OWF alone).
- Useful variants: A gets only oracle access to $h(\cdot)$ (weaker). Or, A gets any coins used for sampling h (stronger).

Degrees of Collision-Resistance

- Variants of CRHF 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 minimising the hash length (for efficiency)
  - Generic attack on a CRHF: birthday attack
    - Look for a collision in a set of random inputs (needs only oracle access to the hash function)
      - Expected size of the set before collision: $O(\sqrt{|range|})$
  - Birthday attack effectively halves the security (hash length) of a CRHF compared to a generic attack on UOWHF

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"
  - $\forall x,z\ \Pr_{h\leftarrow\mathcal{H}}[\ h(x)=z\ ] = 1/|Z|$ (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: $A \to (x,y)$; $h \leftarrow \mathcal{H}$. $h(x)=h(y)$ 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] = $? In a field, 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. 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
    - $Pr_h[\ Chop(h(x)) = w,\ Chop(h(y)) = z]$
      $= Pr_h[\ h(x) = w0 \text{ or } w1,\ h(y) = z0 \text{ or } z1] = 4/|Z|^2 = 1/|Z'|^2$

Cryptographic Hash Functions

- Combinatorial collision resistance depended on the hash function being randomly chosen after (independent of) adversary's pair (x,y)
- But if the hash function is known first, adversary can find collisions
- Often the hash function does have to be public
- Solution: OK if finding collisions is computationally infeasible
  - Cryptographic hash-functions
    - CRHF (and UOWHF)

CRHF: In Theory

- Collision-Resistant HF: $h \leftarrow \mathcal{H}$; $A(h) \rightarrow (x,y)$. $h(x)=h(y)$ w.n.p
- Not known to be possible from OWF/OWP alone
  - "Impossibility" (blackbox-separation) known
- Possible from "claw-free pair of permutations"
  - In turn from hardness of discrete-log, factoring, and from lattice-based assumptions
- Also from "homomorphic one-way permutations", and from homomorphic encryptions
- These candidates use mathematical operations that are fairly expensive (comparable to public-key encryption)

CRHF: In Theory

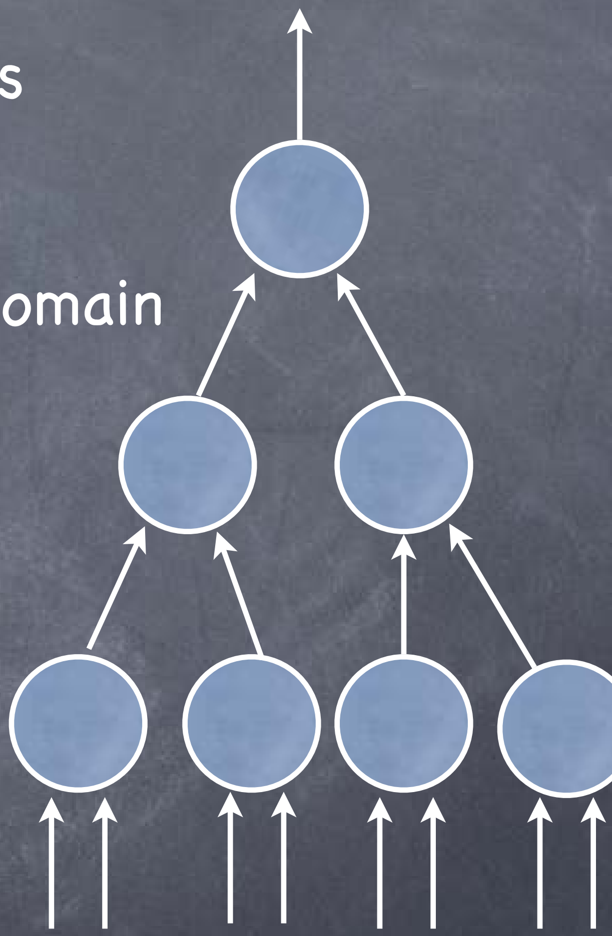
- CRHF from discrete log assumption:
  - Suppose $\mathbb{G}$ a group of prime order q, where DL is considered hard (e.g. $\mathbb{QR}_p^*$ for p=2q+1 a safe prime — i.e., q prime)
  - $h_{g1,g2}(x_1,x_2) = g_1^{x1} g_2^{x2}$ (in $\mathbb{G}$) where g_1, $g_2 \neq 1$ (hence generators)
  - A collision: $(x_1,x_2) \neq (y_1,y_2)$ s.t. $h_{g1,g2}(x_1,x_2) = h_{g1,g2}(y_1,y_2)$
    - Collision ⇒ $x_1 \neq y_1$ and $x_2 \neq y_2$ [Why?]
    - Then $g_2 = g_1^{(x1-y1)/(x2-y2)}$ (exponents in $\mathbb{Z}_q^*$)
      - i.e., w.r.t. a random base g_1, can compute DL of a random element g_2. Breaks DL!
  - Hash halves the size of the input

Domain Extension

- Full-domain hash: hash arbitrarily long strings to a single hash value
  - So far, UOWHF/CRHF which have a fixed domain
- First, simpler goal: extend to a larger, fixed domain
  - Assume we are given a hash function from two blocks to one block (a block being, say, k bits)
    - What if we can compress only slightly — say, by one bit?
      - Can just apply repeatedly to compress by k bits

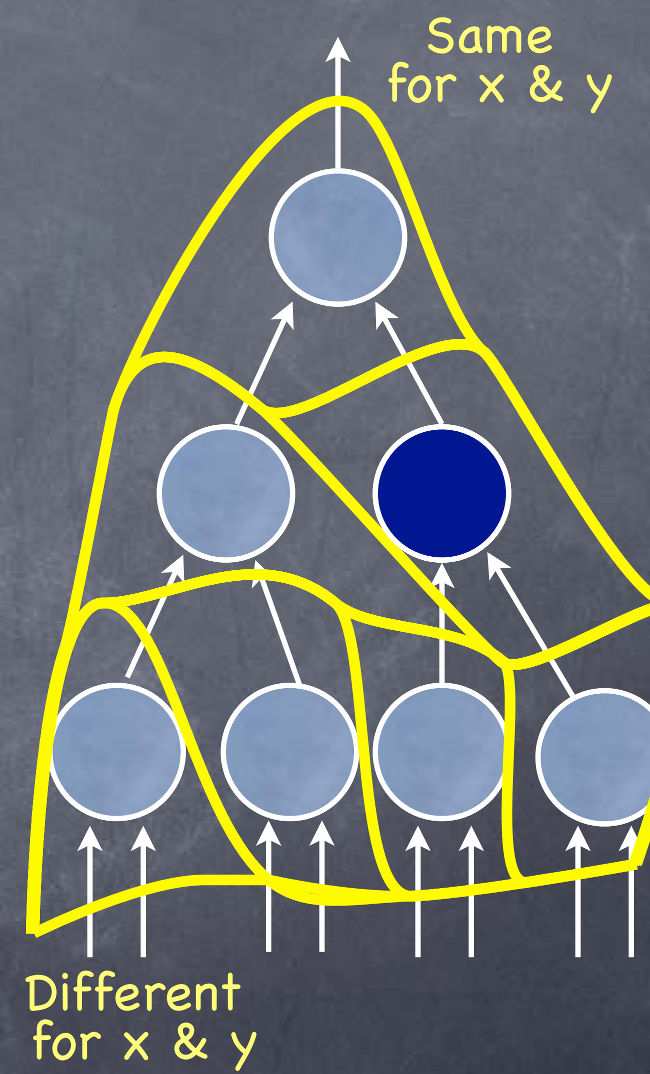
CRHF Domain Extension

- Full-domain hash: hash arbitrarily long strings to a single hash value
- First, simpler goal: extend to a larger, fixed domain
- Can compose hash functions more efficiently, using a "Merkle tree"
  - Uses a basic hash from $\{0,1\}^{2k}$ to $\{0,1\}^k$
  - Example: A hash function from $\{0,1\}^{8k}$ to $\{0,1\}^k$ using a tree of depth 3
  - Any tree can be used, with consistent I/O sizes
  - Same basic hash used at every node in the Merkle tree. Hash description same as for a single basic hash



Domain Extension for CRHF

- If a collision ($(x_1...x_n)$, $(y_1...y_n)$) over all, then some collision (x',y') for basic hash
  - Consider moving a "frontline" from bottom to top. Look for equality on this front.
    - Collision at some step (different values on i^{th} front, same on $i+1^{st}$); gives a collision for basic hash
- A*(h): run A(h) to get $(x_1...x_n)$, $(y_1...y_n)$. Move frontline to find (x',y')

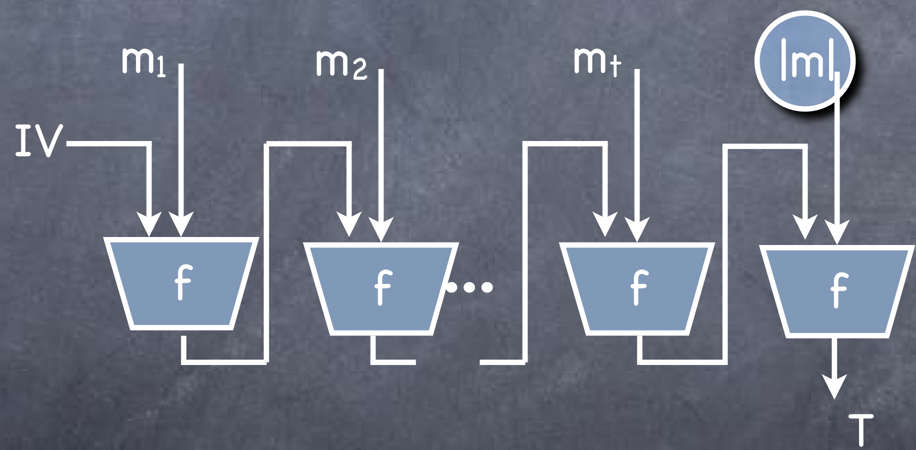


Domain Extension for CRHF

- Full-domain hash: hash arbitrarily long strings to a single hash value
  - Merkle-Tree construction extends the domain to any fixed input length
- Hash the message length (number of blocks) along with the original hash
  - Collision in the new hash function gives either collision at the top level, or if not, collision in the original Merkle tree and for the same message length

CRHF in Practice

- A single function, not a family (e.g. SHA-3, SHA-256, MD4, MD5)
- Often based on a fixed input-length compression function
- Merkle-Damgård iterated hash function, MD^f:



Collision resistance even with variable input-length.

Note: Unlike CBC-MAC, here "length-extension" is OK, as long as it results in a different hash value

If f is not keyed, but "concretely" collision resistant, so is MD^f

- If f "concretely" collision resistant then so is MD^f (for any IV)

Today

- Combinatorial hash functions, UOWHF and CRHF
  - (And weaker variants of CRHF: pre-image collision resistance and second-pre-image collision resistance)
- Collision-resistant combinatorial HF from 2-Universal Hash Functions
- A candidate CRHF construction based on Discrete Log assumption
- Domain extension: Merkle Tree, Merkle-Damgård iterated hash