Collision Resistant Hashing

Recap: secure MAC (S, V):

existentially unforgeable under chosen msg attack.

PRF gives a secure MAC (e.g. AES)

Goal: PRF with small domain

$\Rightarrow$ PRF for large domain

ex 1: CBC-MAC or CMAC

ex 2: PMAC

Today: ex 3: HMAC (hash MAC)

First: collision resistant hashing.

Let $H: M \to T$ be a hash function $(|T| \ll |M|)$

a <u>collision</u> for H is a pair $m_0, m_1 \in M$

s.t. $m_0 \neq m_1$ and $H(m_0) = H(m_1)$

$|T| < |M| \Rightarrow$ collisions must exist !!

<u>Def:</u> A function $H: M \to T$ is <u>collision resistant</u>

if for all (explicit) "eff" algs. A:

$$CRadv[A, H] = Pr[A \text{ outputs collision for } H]$$

is negligible.

$\Rightarrow$ hard to find an explicit collision.

<u>Std examples:</u> SHA256, SHA384, SHA512 (2001)

SHA3-256, SHA3-384, SHA3-512 (2014)

An immediate application: small-mac $\Rightarrow$ Big-mac

(S, V) a secure MAC over $(\mathcal{K}, M, T)$ for short msgs.

$H: M^{big} \to M$ a CRH

Define: (S', V') a MAC over $(\mathcal{K}, M^{big}, T)$ where:

$$S'(K, m) := S(K, H(m))$$
$$V'(K, m, t) := V(K, H(m), t)$$

Thm: (S, V) a secure MAC, H a CRH.

Then (S', V') is a secure MAC.

"Proof:" Suppose an adv. $\mathcal{A}$ attacks (S', V'):

(1) requests tag on $m_1, m_2, \ldots \in M$ and get $t_1, t_2, \ldots$.

(2) outputs a Forgery $(m, t) \neq (m_i, t_i)$, $i = 1, 2, \ldots$

Then either:

(1) $\exists i : H(m) = H(m_i)$ and $m \neq m_i \Rightarrow$ Attack on CRHF.

or

(2) $\forall i : (H(m), t) \neq (H(m_i), t_i) \Rightarrow$ Attack on (S, V)

Why CRHF is necessary!

Suppose adv. has $m_0 \neq m_1 \in M$ s.t. $H(m_0) = H(m_1)$,

attack on (S', V'): request tag on m_0, get t_0.

output Forgery (m_1, t_0).

How to construct a CRH ?

First: a general attack on CRH - the birthday attack

The birthday paradox:

Let $r_0, \ldots, r_n \in \{1, \ldots, B\}$ be indep. uniform random variables.

Thm: when $n \geq 1.2\sqrt{B}$ then $\Pr[\exists i \neq j : r_i = r_j] \geq \frac{1}{2}$

Proof:

$$\Pr[\exists i \neq j : r_i = r_j] = 1 - \Pr[\forall i \neq j : r_i \neq r_j] =$$

$$= 1 - (1 - \tfrac{1}{B})(1 - \tfrac{2}{B}) \cdots (1 - \tfrac{n}{B}) =$$

$$= 1 - \prod_{i=1}^{n} (1 - \tfrac{i}{B}) \geq$$

$\forall x \in \mathbb{R}$: $\left(e^{-x} \geq 1 - x \Rightarrow\right)$

$$\geq 1 - \prod_{i=1}^{n} e^{-i/B} \geq 1 - e^{-\frac{1}{2}(n^2/B)}$$

$$\geq 1 - e^{-\frac{1}{2}\cdot(1.2)^2} = 1 - \left(\tfrac{1}{e}\right)^{0.72} \approx 0.53 \geq \tfrac{1}{2}$$ ∎

Note: what if $r_0, \ldots, r_n$ are indep. but non-uniform random variables?

Then # samples to get a collision is less than $1.2\sqrt{B}$.

$\Rightarrow$ the uniform distr. is the worst case for thm.

So: if $H: M \to T$ outputs ℓ-bit string

$\Rightarrow$ can find collision in time $\approx 2^{\ell/2}$

Birthday attack:
- Choose random $m_0, m_1, \ldots, m_{2^{\ell/2}} \in M$
- Compute $H(m_0), \ldots, H(m_{2^{\ell/2}})$
- Look for collision

$\Rightarrow$ time $\approx 2^{\ell/2}$.

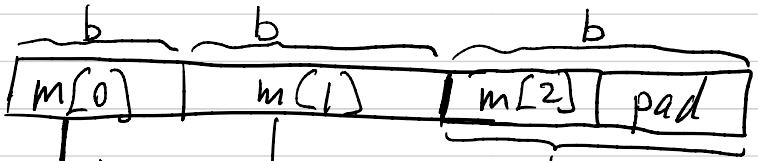
So: 128-bit hash $\Rightarrow$ 2^{64}-time attack

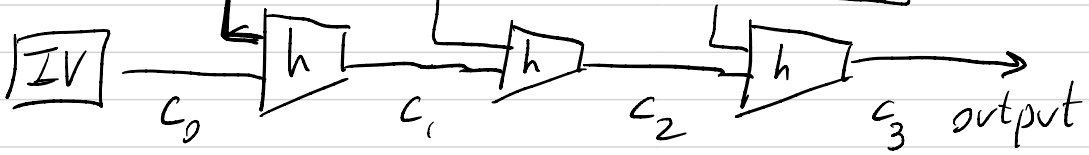
$\Rightarrow$ Typical hash value: 256-bit $\Rightarrow$ 2^{128} security.

Naively: memory $O(2^{\ell/2})$. Puzzle: constant memory.

Quantum: Some evidence that it can be done in time $O(2^{\ell/3})$, but still open.

Constructing a CRH : step 1: Merkle-Damgard

message: m = 

H_{MD}: 

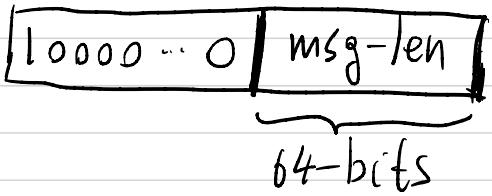
terminology: (1) $h: \{0,1\}^b \times T \to T$ compression function

(2) c_0, c_1, c_2, c_3 : chaining variables

(3) IV : fixed initial value

SHA256: block size $b = 512$ bits.

pad: ensures msg-len is multiple of b bits.

pad = 

if no space for msg-len, add dummy block.

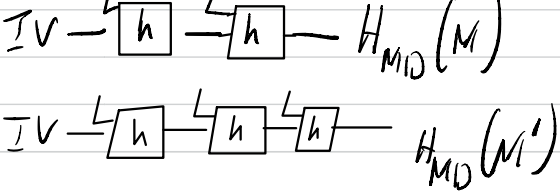
Lemma: h CRH $\Rightarrow$ H_{MD} CRH

So: to construct a large CRH, sufficient to construct a small one.

Proof: Suppose $H_{MD}(M) = H_{MD}(M')$. Goal: find F-collision

M: $IV = c_0, c_1, c_2, \ldots, c_t$

M': $IV = c'_0, c'_1, c'_2, \ldots, c'_r$



$H_{MD}(M) = H_{MD}(M') \Rightarrow c_t = c'_r \Rightarrow$

$$h(M[t-1], c_{t-1}) = c_t = c'_r = h(M[r-1], c_{r-1}).$$

So: if $M[t-1] \neq M'[r-1]$ or $c_t \neq c'_{t-1}$

$\Rightarrow$ Found collision for h. Done.

So: suppose $M[t-1] = M'[r-1] \Rightarrow t = r$ & $c_{t-1} = c'_{t-1}$.

Then: $h(M[t-2], c_{t-2}) = c_{t-1} = c'_{t-1} = h(M'[t-2], c'_{t-2})$

if $M[t-2] \neq M'[t-2]$ or $c_{t-2} \neq c'_{t-2}$ we are done

So: suppose $M[t-1] = M'(t-1]$ & $M[t-2] = M'[t-2)$ & $c_{t-2} = c'_{t-2}$

keep going until first block.

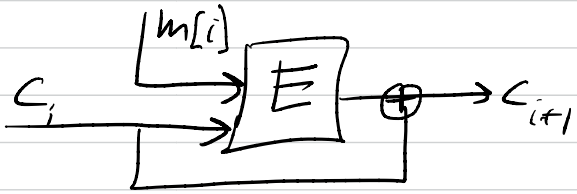
$\Rightarrow$ either find collision on h or $M = M'$

Constructing compression Function: Davies-Meyer

Let $E(k, x)$ be a block cipher over $(\mathcal{K}, X)$

where $X = \{0,1\}^n$

$$h(m, c) = E(m, c) \oplus c$$



"Thm:" if E is "ideal" (random)

then finding a collision takes time $\geq 2^{n/2}$.

Best possible!

SHA256 uses a cipher called SHACAL2.

A MAC from a hash function

How to build a PRF F (and MAC) from hash function $H: M \to T$??

Attempt 1: $F(k, m) := H(k \| m)$

bad idea: extension attack on MD hash

Given $y = F(k, m)$ anyone can compute $y' = F(k, m \| pad \| x)$

For all one-block msgs x.

std method: HMAC

$$F_{HMAC}(k, m) := H\Big((k \oplus opad) \,\|\, H(k \oplus ipad \,\|\, m)\Big)$$

outer pad, inner pad

fixed values

Thm: if compr. func. $h(m, c)$ is a secure PRF (with either input as key) then HMAC is a secure PRF.

$\Rightarrow$ A crypto library that implements SHA256 can use HMAC as a MAC.