HASH FUNCTIONS

What is a hash function?

By a hash function we usually mean a map $h : D \to \{0,1\}^n$ that is compressing, meaning $|D| > 2^n$.

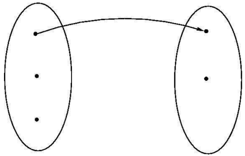
E.g. $D = \{0,1\}^{\leq 2^{64}}$ is the set of all strings of length at most 2^{64}.

h	n
MD4	128
MD5	128
SHA1	160
RIPEMD	128
RIPEMD-160	160
SHA-256	256
Skein	256, 512, 1024

Collision resistance (CR)

Definition: A collision for $h : D \rightarrow \{0,1\}^n$ is a pair $x_1, x_2 \in D$ of points such that $h(x_1) = h(x_2)$ but $x_1 \neq x_2$.

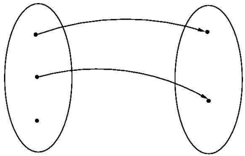
If $|D| > 2^n$ then the pigeonhole principle tells us that there must exist a collision for h.



Collision resistance (CR)

Definition: A collision for $h : D \rightarrow \{0, 1\}^n$ is a pair $x_1, x_2 \in D$ of points such that $h(x_1) = h(x_2)$ but $x_1 \neq x_2$.

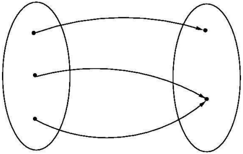
If $|D| > 2^n$ then the pigeonhole principle tells us that there must exist a collision for h.



Collision resistance (CR)

Definition: A collision for $h : D \to \{0,1\}^n$ is a pair $x_1, x_2 \in D$ of points such that $h(x_1) = h(x_2)$ but $x_1 \neq x_2$.

If $|D| > 2^n$ then the pigeonhole principle tells us that there must exist a collision for h.



Function h is collision-resistant if it is computationally infeasible to find a collision.

Function families

We consider a family $H : \{0,1\}^k \times D \to \{0,1\}^n$ of functions, meaning for each K we have a map $h = H_K : D \to \{0,1\}^n$ defined by

$$h(x) = H(K, x)$$

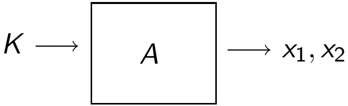
Usage: $K \xleftarrow{\$} \{0,1\}^k$ is made public, defining hash function $h = H_K$.

Note the key K is not secret. Both users and adversaries get it.

CR of function families

Let $H : \{0,1\}^k \times D \to \{0,1\}^n$ be a family of functions. A cr-adversary A for H

- Takes input a key $K \in \{0,1\}^k$
- Outputs a pair $x_1, x_2 \in D$ of points in the domain of H



A wins if x_1, x_2 are a collision for H_K, meaning

- $x_1 \neq x_2$, and
- $H_K(x_1) = H_K(x_2)$

Denote by $\mathbf{Adv}^{\mathrm{cr}}_H(A)$ the probability that A wins.

CR of function families

Let $H : \{0,1\}^k \times D \to \{0,1\}^n$ be a family of functions and A a cr-adversary for H.

Game CR_H

procedure Initialize	procedure Finalize(x_1, x_2)
$K \xleftarrow{\$} \{0,1\}^k$	Return $(x_1 \neq x_2 \wedge H_K(x_1) = H_K(x_2))$
Return K	

Let

$$\mathbf{Adv}^{\mathrm{cr}}_H(A) = \Pr\left[\mathrm{CR}^A_H \Rightarrow \mathsf{true}\right].$$

The measure of success

Let $H\colon \{0,1\}^k \times D \to \{0,1\}^n$ be a family of functions and A a cr adversary. Then

$$\mathbf{Adv}_H^{\mathrm{cr}}(A) = \Pr\left[\mathrm{CR}_H^A \Rightarrow \mathsf{true}\right].$$

is a number between 0 and 1.

A "large" (close to 1) advantage means

- A is doing well
- H is not secure

A "small" (close to 0) advantage means

- A is doing poorly
- H resists the attack A is mounting

CR security

Adversary advantage depends on its

- strategy
- resources: Running time t

Security: H is CR if $\mathbf{Adv}_H^{\mathrm{cr}}(A)$ is "small" for ALL A that use "practical" amounts of resources.

Insecurity: H is insecure (not CR) if there exists A using "few" resources that achieves "high" advantage.

In notes we sometimes refer to CR as CR-KK2.

Example

Let H: $\{0,1\}^k \times \{0,1\}^{256} \rightarrow \{0,1\}^{128}$ be defined by

$$H_K(x) = H_K(x[1]x[2]) = \mathsf{AES}_K(x[1]) \oplus \mathsf{AES}_K(x[2])$$

Is H collision resistant?

Example

Let H: $\{0,1\}^k \times \{0,1\}^{256} \rightarrow \{0,1\}^{128}$ be defined by

$$H_K(x) = H_K(x[1]x[2]) = \text{AES}_K(x[1]) \oplus \text{AES}_K(x[2])$$

Is H collision resistant?

Can you design an adversary A

$$K \longrightarrow \boxed{A} \longrightarrow \begin{array}{l} x_1 = x_1[1]x_1[2] \\ x_2 = x_2[1]x_2[2] \end{array}$$

such that $H_K(x_1) = H_K(x_2)$?

Example

Let H: $\{0,1\}^k \times \{0,1\}^{256} \to \{0,1\}^{128}$ be defined by

$$H_K(x) = H_K(x[1]x[2]) = \mathsf{AES}_K(x[1]) \oplus \mathsf{AES}_K(x[2])$$

Weakness:

$$H_K(x[1]x[2]) = H_K(x[2]x[1])$$

adversary $A(K)$
$x_1 \leftarrow 0^{128}1^{128}$; $x_2 \leftarrow 1^{128}0^{128}$; return x_1, x_2

Then

$$\mathbf{Adv}_H^{\mathrm{cr}}(A) = 1$$

and A is efficient, so H is not CR.

SHA1

algorithm $\mathsf{SHA1}(M)$ // $|M| < 2^{64}$

$V \leftarrow \mathsf{SHF1}(\texttt{5A827999} \| \texttt{6ED9EBA1} \| \texttt{8F1BBCDC} \| \texttt{CA62C1D6}, M)$

return V

---

algorithm $\mathsf{SHF1}(K, M)$ // $|K| = 128$ and $|M| < 2^{64}$

$y \leftarrow \mathsf{shapad}(M)$

Parse y as $M_1 \| M_2 \| \cdots \| M_n$ where $|M_i| = 512$ $(1 \leq i \leq n)$

$V \leftarrow \texttt{67452301} \| \texttt{EFCDAB89} \| \texttt{98BADCFE} \| \texttt{10325476} \| \texttt{C3D2E1F0}$

for $i = 1, \ldots, n$ do

$V \leftarrow \mathsf{shf1}(K, M_i \| V)$

return V

---

algorithm $\mathsf{shapad}(M)$ // $|M| < 2^{64}$

$d \leftarrow (447 - |M|) \bmod 512$

Let ℓ be the 64-bit binary representation of $|M|$

$y \leftarrow M \| 1 \| 0^d \| \ell$ // $|y|$ is a multiple of 512

return y

SHA1

algorithm $\mathsf{shf1}(K, B \parallel V)$ // $|K| = 128$, $|B| = 512$ and $|V| = 160$

Parse B as $W_0 \parallel W_1 \parallel \cdots \parallel W_{15}$ where $|W_i| = 32$ $(0 \le i \le 15)$

Parse V as $V_0 \parallel V_1 \parallel \cdots \parallel V_4$ where $|V_i| = 32$ $(0 \le i \le 4)$

Parse K as $K_0 \parallel K_1 \parallel K_2 \parallel K_3$ where $|K_i| = 32$ $(0 \le i \le 3)$

for $t = 16$ to 79 do $W_t \leftarrow \mathrm{ROTL}^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16})$

$A \leftarrow V_0$; $B \leftarrow V_1$; $C \leftarrow V_2$; $D \leftarrow V_3$; $E \leftarrow V_4$

for $t = 0$ to 19 do $L_t \leftarrow K_0$; $L_{t+20} \leftarrow K_1$; $L_{t+40} \leftarrow K_2$; $L_{t+60} \leftarrow K_3$

for $t = 0$ to 79 do

  if $(0 \le t \le 19)$ then $f \leftarrow (B \wedge C) \vee ((\neg B) \wedge D)$

  if $(20 \le t \le 39 \text{ OR } 60 \le t \le 79)$ then $f \leftarrow B \oplus C \oplus D$

  if $(40 \le t \le 59)$ then $f \leftarrow (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$

  $temp \leftarrow \mathrm{ROTL}^5(A) + f + E + W_t + L_t$

  $E \leftarrow D$; $D \leftarrow C$; $C \leftarrow \mathrm{ROTL}^{30}(B)$; $B \leftarrow A$; $A \leftarrow temp$

$V_0 \leftarrow V_0 + A$; $V_1 \leftarrow V_1 + B$; $V_2 \leftarrow V_2 + C$; $V_3 \leftarrow V_3 + D$; $V_4 \leftarrow V_4 + E$

$V \leftarrow V_0 \parallel V_1 \parallel V_2 \parallel V_3 \parallel V_4$

return V

Applications of hash functions

- primitive in cryptographic schemes
- tool for security applications
- tool for non-security applications

Password verification

- Client A has a password PW that is also held by server B
- A authenticates itself by sending PW to B over a secure channel (SSL)

$$A^{PW} \xrightarrow{\quad PW \quad} B^{PW}$$

Problem: The password will be found by an attacker who compromises the server.

Password verification

- Client A has a password PW and server stores $\overline{PW} = H(PW)$.
- A sends PW to B (over a secure channel) and B checks that $H(PW) = \overline{PW}$

$$A^{PW} \xrightarrow{\quad PW \quad} B^{\overline{PW}}$$

Server compromise results in attacker getting $\overline{PW}$ which should not reveal PW as long as H is one-way, which we will see is a consequence of collision-resistance.

But we will revisit this when we consider dictionary attacks!

Compare-by-hash

- A has a large file F_A and B has a large file F_B. For example, music collections.
- They want to know whether $F_A = F_B$
- A sends F_A to B and B checks whether $F_A = F_B$

$$A^{F_A} \xrightarrow{\quad F_A \quad} B^{F_B}$$

Problem: Transmission could take forever, particularly if the link is slow (DSL).

Compare-by-hash

- A has a large file F_A and B has a large file F_B and they want to know whether $F_A = F_B$
- A computes $h_A = H(F_A)$ and sends it to B, and B checks whether $h_A = H(F_B)$.

$$A^{F_A} \xrightarrow{\quad h_A \quad} B^{F_B}$$

Collision-resistance of H guarantees that B does not accept if $F_A \neq F_B$!

Compare-by-hash

- A has a large file F_A and B has a large file F_B and they want to know whether $F_A = F_B$
- A computes $h_A = H(F_A)$ and sends it to B, and B checks whether $h_A = H(F_B)$.

$$A^{F_A} \xrightarrow{\quad h_A \quad} B^{F_B}$$

Collision-resistance of H guarantees that B does not accept if $F_A \neq F_B$!

Added bonus: This to some extent protects privacy of F_A, F_B. But be careful: not in the strong IND-CPA sense we have studied.

Virus protection

An executable may be available at lots of sites $S_1, S_2, \ldots, S_N$. Which one can you trust?

- Provide a safe way to get the hash $h = H(X)$ of the correct executable X.
- Download an executable from anywhere, and check hash.

General collision-finding attacks

We discuss attacks on $H : \{0,1\}^k \times D \to \{0,1\}^n$ that do no more than compute H. Let $D_1, \ldots, D_d$ be some enumeration of the elements of D.

Adversary $A_1(K)$
$x_1 \xleftarrow{\$} D; y \leftarrow H_K(x_1)$
For $i = 1, \ldots, q$ do
  If $(H_K(D_i) = y \wedge x_1 \neq D_i)$ then
    Return x_1, D_i
Return FAIL

Adversary $A_2(K)$
$x_1 \xleftarrow{\$} D; y \leftarrow H_K(x_1)$
For $i = 1, \ldots, q$ do
  $x_2 \xleftarrow{\$} D$
  If $(H_K(x_2) = y \wedge x_1 \neq x_2)$ then
    Return x_1, x_2
Return FAIL

Now:

- A_1 could take $q = d = |D|$ trials to succeed.
- We expect A_2 to succeed in about 2^n trials.

But this still means 2^{160} trials to find a SHA1 collision.

Birthday attacks

Let $H : \{0,1\}^k \times D \to \{0,1\}^n$ be a family of functions with $|D| > 2^n$.
The q-trial birthday attack finds a collision with probability about

$$\frac{q^2}{2^{n+1}}.$$

So a collision can be found in about $q = \sqrt{2^{n+1}} \approx 2^{n/2}$ trials.

Recall Birthday Problem

for $i = 1, \ldots, q$ do $y_i \xleftarrow{\$} \{0,1\}^n$
if $\exists i, j$ $(i \neq j$ and $y_i = y_j)$ then $\mathsf{COLL} \leftarrow \mathsf{true}$

$$\begin{aligned} \Pr[\mathsf{COLL}] &= C(2^n, q) \\ &\approx \frac{q^2}{2^{n+1}} \end{aligned}$$

Birthday attack

Let $H : \{0,1\}^k \times D \rightarrow \{0,1\}^n$.

adversary $A(K)$
for $i = 1, \ldots, q$ do $x_i \xleftarrow{\$} D$; $y_i \leftarrow H_K(x_i)$
if $\exists i, j$ $(i \neq j$ and $y_i = y_j$ and $x_i \neq x_j)$ then return x_i, x_j
else return FAIL

Analysis of birthday attack

Let $H : \{0,1\}^k \times D \to \{0,1\}^n$.

adversary $A(K)$
for $i = 1, \ldots, q$ do $x_i \xleftarrow{\$} D$; $y_i \leftarrow H_K(x_i)$
if $\exists i, j$ $(i \neq j$ and $y_i = y_j$ and $x_i \neq x_j)$ then return x_i, x_j
else return FAIL

What is the probability that this attack finds a collision?

adversary $A(K)$
for $i = 1, \ldots, q$ do $x_i \xleftarrow{\$} D$; $y_i \leftarrow H_K(x_i)$
if $\exists i, j$ $(i \neq j$ and $y_i = y_j)$ then COLL $\leftarrow$ true

We have dropped things that don't much affect the advantage and focused on success probability. So we want to know what is

$$\Pr[\text{COLL}] \,.$$

Analysis of birthday attack

Birthday	Adversary A

for $i = 1, \ldots, q$ do
$\quad y_i \xleftarrow{\$} \{0,1\}^n$
if $\exists i, j\ (i \neq j$ and $y_i = y_j)$ then
$\quad \mathsf{COLL} \leftarrow \mathsf{true}$
	for $i = 1, \ldots, q$ do
	$\quad x_i \xleftarrow{\$} D\ ;\ y_i \leftarrow H_K(x_i)$
	if $\exists i, j (i \neq j$ and $y_i = y_j)$ then
	$\quad \mathsf{COLL} \leftarrow \mathsf{true}$

$\Pr[\mathsf{COLL}] = C(2^n, q)$	$\Pr[\mathsf{COLL}] = ?$

Are the two collision probabilities the same?

Analysis of birthday attack

Birthday	Adversary A

for $i = 1, \ldots, q$ do
$\quad y_i \stackrel{\$}{\leftarrow} \{0,1\}^n$
if $\exists i, j\ (i \neq j$ and $y_i = y_j)$ then
$\quad \mathsf{COLL} \leftarrow \mathsf{true}$
	for $i = 1, \ldots, q$ do
	$\quad x_i \stackrel{\$}{\leftarrow} D\,;\ y_i \leftarrow H_K(x_i)$
	if $\exists i, j (i \neq j$ and $y_i = y_j)$ then
	$\quad \mathsf{COLL} \leftarrow \mathsf{true}$

$\Pr[\mathsf{COLL}] = C(2^n, q)$	$\Pr[\mathsf{COLL}] =?$

Are the two collision probabilities the same?

Not necessarily, because

- on the left $y_i \stackrel{\$}{\leftarrow} \{0,1\}^n$
- on the right $x_i \stackrel{\$}{\leftarrow} D\,;\ y_i \leftarrow H_K(x_i)$

Analysis of birthday attack

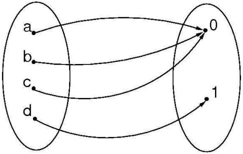
Consider the following processes

Process 1	Process 2
$y \xleftarrow{\$} \{0,1\}^n$	$x \xleftarrow{\$} D;\ y \xleftarrow{\$} H_K(x)$
return y	return y

Process 1 certainly returns a random n-bit string. Does Process 2?

Analysis of birthday attack

Process 1	Process 2
$y \stackrel{\$}{\leftarrow} \{0,1\}$	$x \stackrel{\$}{\leftarrow} \{\text{a,b,c,d}\}$; $y \leftarrow H_K(x)$
return y	return y



$\Pr[y = 0] =$

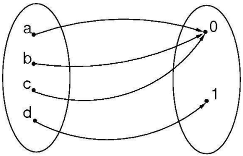
$\Pr[y = 1] =$

$\Pr[y = 0] =$

$\Pr[y = 1] =$

Analysis of birthday attack

Process 1	Process 2
$y \xleftarrow{\$} \{0,1\}$	$x \xleftarrow{\$} \{a,b,c,d\}$; $y \leftarrow H_K(x)$
return y	return y



$$\Pr[y=0] = \frac{1}{2}$$

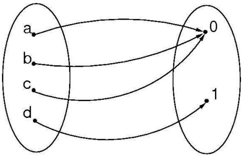
$$\Pr[y=1] = \frac{1}{2}$$

$\Pr[y=0] =$

$\Pr[y=1] =$

Analysis of birthday attack

Process 1	Process 2
$y \xleftarrow{\$} \{0,1\}$	$x \xleftarrow{\$} \{\mathrm{a,b,c,d}\}\ ;\ y \leftarrow H_K(x)$
return y	return y

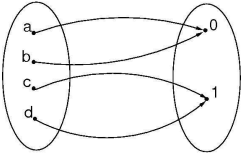


$$\Pr[y=0] = \frac{1}{2} \qquad \Pr[y=0] = \frac{3}{4}$$

$$\Pr[y=1] = \frac{1}{2} \qquad \Pr[y=1] = \frac{1}{4}$$

Analysis of birthday attack

Process 1	Process 2
$y \xleftarrow{\$} \{0,1\}$	$x \xleftarrow{\$} \{\mathrm{a,b,c,d}\}$; $y \leftarrow H_K(x)$
return y	return y



$\Pr[y = 0] =$

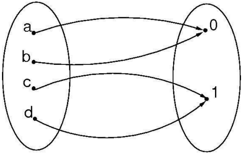
$\Pr[y = 1] =$

$\Pr[y = 0] =$

$\Pr[y = 1] =$

Analysis of birthday attack

Process 1	Process 2
$y \xleftarrow{\$} \{0,1\}$	$x \xleftarrow{\$} \{\mathsf{a,b,c,d}\}\ ;\ y \leftarrow H_K(x)$
return y	return y



$$\Pr[y=0] = \frac{1}{2}$$

$$\Pr[y=1] = \frac{1}{2}$$

$$\Pr[y=0] = \frac{1}{2}$$

$$\Pr[y=1] = \frac{1}{2}$$

The processes are the same if every range point has the same number of pre-images.

Analysis of birthday attack

We say that $H : \{0,1\}^k \times D \to \{0,1\}^n$ is regular if every range point has the same number of pre-images under H_K. That is if we let

$$H_K^{-1}(y) = \{x \in D : H_K(x) = y\}$$

then H is regular if

$$|H_K^{-1}(y)| = \frac{|D|}{2^n}$$

for all K and y. In this case the following processes both result in a random output

Process 1	Process 2
$y \xleftarrow{\$} \{0,1\}^n$	$x \xleftarrow{\$} D;\ y \xleftarrow{\$} H_K(x)$
return y	return y

Analysis of birthday attack

If H: $\{0,1\}^k \times D \rightarrow \{0,1\}^n$ is regular then the birthday attack finds a collision in about $2^{n/2}$ trials.

Analysis of birthday attack

If H: $\{0,1\}^k \times D \rightarrow \{0,1\}^n$ is regular then the birthday attack finds a collision in about $2^{n/2}$ trials.

If H is not regular, the attack may succeed sooner.

So we want functions to be "close to regular".

It seems MD4,MD5,SHA1,RIPEMD,... have this property.

Birthday attack times

Function	n	T_B
MD4	128	2^{64}
MD5	128	2^{64}
SHA1	160	2^{80}
RIPEMD-160	160	2^{80}
SHA256	256	2^{128}

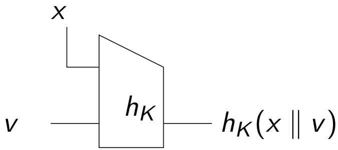
T_B is the number of trials to find collisions via a birthday attack.

Compression functions

A compression function is a family $h : \{0,1\}^k \times \{0,1\}^{b+n} \to \{0,1\}^n$ of hash functions whose inputs are of a fixed size $b+n$, where b is called the block size.

E.g. $b = 512$ and $n = 160$, in which case

$$h : \{0,1\}^k \times \{0,1\}^{672} \to \{0,1\}^{160}$$



The MD transform

Design principle: To build a CR hash function

$$H : \{0,1\}^k \times D \to \{0,1\}^n$$

where $D = \{0,1\}^{\leq 2^{64}}$:

- First build a CR compression function $h : \{0,1\}^k \times \{0,1\}^{b+n} \to \{0,1\}^n$.
- Appropriately iterate h to get H, using h to hash block-by-block.

MD setup

Assume for simplicity that $|M|$ is a multiple of b. Let

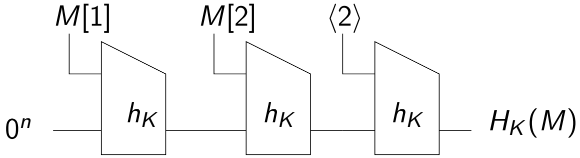
- $\|M\|_b$ be the number of b-bit blocks in M, and write $M = M[1] \dots M[\ell]$ where $\ell = \|M\|_b$.
- $\langle i \rangle$ denote the b-bit binary representation of $i \in \{0, \dots, 2^b - 1\}$.
- D be the set of all strings of at most $2^b - 1$ blocks, so that $\|M\|_b \in \{0, \dots, 2^b - 1\}$ for any $M \in D$, and thus $\|M\|_b$ can be encoded as above.

MD transform

Given: Compression function $h : \{0,1\}^k \times \{0,1\}^{b+n} \rightarrow \{0,1\}^n$.

Build: Hash function $H : \{0,1\}^k \times D \rightarrow \{0,1\}^n$.

Algorithm $H_K(M)$
$m \leftarrow \|M\|_b$; $M[m+1] \leftarrow \langle m \rangle$; $V[0] \leftarrow 0^n$
For $i = 1, \ldots, m+1$ do $v[i] \leftarrow h_K(M[i] \| V[i-1])$
Return $V[m+1]$



MD preserves CR

Assume

- h is CR
- H is built from h using MD

Then

- H is CR too!

This means

- No need to attack H! You won't find a weakness in it unless h has one
- H is guaranteed to be secure assuming h is.

For this reason, MD is the design used in many current hash functions. Newer hash functions use other iteration methods with analogous properties.

MD preserves CR

Theorem: Let $h : \{0,1\}^k \times \{0,1\}^{b+n} \to \{0,1\}^n$ be a family of functions and let $H : \{0,1\}^k \times D \to \{0,1\}^n$ be obtained from h via the MD transform. Then for any cr-adversary A_H there exists a cr-adversary A_h such that

$$\mathbf{Adv}_H^{\mathrm{cr}}(A_H) \leq \mathbf{Adv}_h^{\mathrm{cr}}(A_h)$$

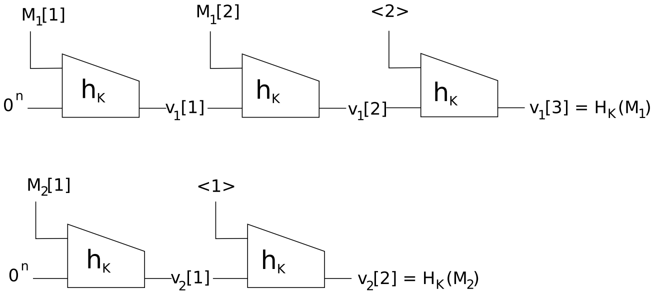
and the running time of A_h is that of A_H plus the time for computing h on the outputs of A_H.

Implication:

$$\begin{aligned} h \text{ CR} &\Rightarrow \mathbf{Adv}_H^{\mathrm{cr}}(A_h) \text{ small} \\ &\Rightarrow \mathbf{Adv}_H^{\mathrm{cr}}(A_H) \text{ small} \\ &\Rightarrow H \text{ CR} \end{aligned}$$

How A_h works

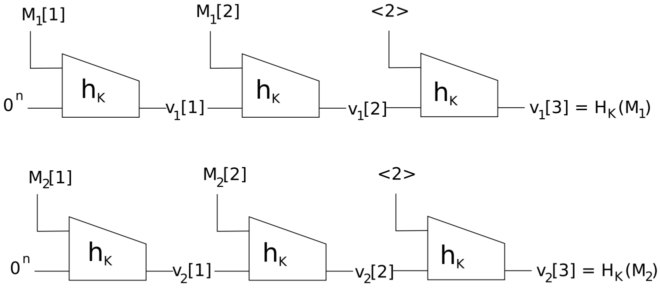
Let (M_1, M_2) be the H_K-collision returned by A_H. The A_h will trace the chains backwards to find an h_k-collision.



Let $x_1 = \langle 2 \rangle || V_1[2]$ and $x_2 = \langle 1 \rangle || V_2[1]$. Then

- $h_K(x_1) = h_K(x_2)$ because $H_K(M_1) = H_K(M_2)$.
- But $x_1 \neq x_2$ because $\langle 1 \rangle \neq \langle 2 \rangle$.

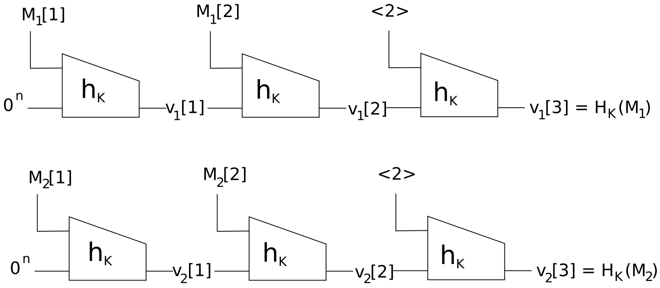
Case 2: $\|M_1\|_b = \|M_2\|_b$



$x_1 \leftarrow \langle 2\rangle || V_1[2]$; $x_2 \leftarrow \langle 2\rangle || V_2[2]$

If $x_1 \neq x_2$ then return x_1, x_2

Case 2: $\|M_1\|_b = \|M_2\|_b$

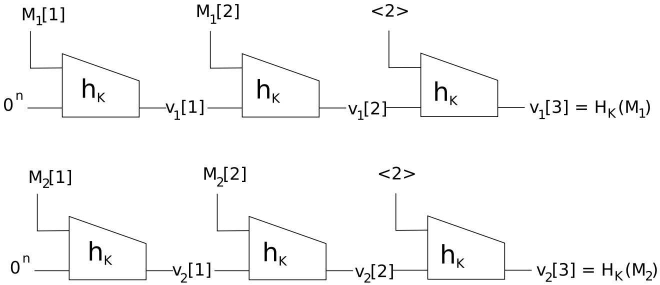


$x_1 \leftarrow \langle 2 \rangle || V_1[2]$; $x_2 \leftarrow \langle 2 \rangle || V_2[2]$

If $x_1 \neq x_2$ then return x_1, x_2

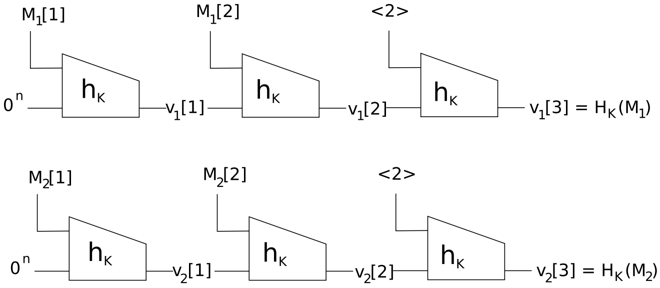
Else // $V_1[2] = V_2[2]$

Case 2: $\|M_1\|_b = \|M_2\|_b$



$x_1 \leftarrow \langle 2 \rangle || V_1[2]$; $x_2 \leftarrow \langle 2 \rangle || V_2[2]$
If $x_1 \neq x_2$ then return x_1, x_2
Else // $V_1[2] = V_2[2]$
  $x_1 \leftarrow M_1[2] || V_1[1]$; $x_2 \leftarrow M_2[2] || V_2[1]$
  If $x_1 \neq x_2$ then return x_1, x_2

Case 2: $\|M_1\|_b = \|M_2\|_b$



$x_1 \leftarrow \langle 2 \rangle || V_1[2]$; $x_2 \leftarrow \langle 2 \rangle || V_2[2]$

If $x_1 \neq x_2$ then return x_1, x_2

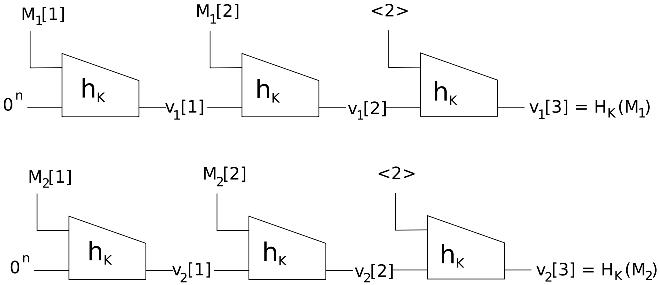
Else // $V_1[2] = V_2[2]$

$\quad x_1 \leftarrow M_1[2] || V_1[1]$; $x_2 \leftarrow M_2[2] || V_2[1]$

$\quad$ If $x_1 \neq x_2$ then return x_1, x_2

$\quad$ Else // $V_1[1] = V_2[1]$

Case 2: $\|M_1\|_b = \|M_2\|_b$



$x_1 \leftarrow \langle 2 \rangle || V_1[2]$; $x_2 \leftarrow \langle 2 \rangle || V_2[2]$

If $x_1 \neq x_2$ then return x_1, x_2

Else // $V_1[2] = V_2[2]$

- $x_1 \leftarrow M_1[2] || V_1[1]$; $x_2 \leftarrow M_2[2] || V_2[1]$
- If $x_1 \neq x_2$ then return x_1, x_2
- Else // $V_1[1] = V_2[1]$
  - $x_1 \leftarrow M_1[1] || 0^n$; $x_2 \leftarrow M_2[1] || 0^n$
  - Return x_1, x_2

How are compression functions designed?

Let $E : \{0,1\}^b \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Let us design keyless compression function

$$h : \{0,1\}^{b+n} \to \{0,1\}^n$$

by

$$h(x||v) = E_x(v)$$

Is H collision resistant?

How are compression functions designed?

Let $E : \{0,1\}^b \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a block cipher. Let us design keyless compression function

$$h : \{0,1\}^{b+n} \rightarrow \{0,1\}^n$$

by

$$h(x||v) = E_x(v)$$

Is H collision resistant?

NO!

adversary A

Pick some x_1, x_2, v_1 with $x_1 \neq x_2$

$y \leftarrow E_{x_1}(v_1)\,;\; v_2 \leftarrow E_{x_2}^{-1}(y)$

return $x_1 \parallel v_1, x_2 \parallel v_2$

Then

$$E_{x_1}(v_1) = y = E_{x_2}(v_2)$$

How are compression functions designed?

Let $E : \{0,1\}^b \times \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Keyless compression function

$$h : \{0,1\}^{b+n} \to \{0,1\}^n$$

may be designed as

$$h(x||v) = E_x(v) \oplus v$$

The compression function of SHA1 is underlain in this way by a block cipher $E : \{0,1\}^{512} \times \{0,1\}^{160} \to \{0,1\}^{160}$.

Cryptanalytic attacks

So far we have looked at attacks that do not attempt to exploit the structure of H.

Can we do better than birthday if we do exploit the structure?

Ideally not, but functions have fallen short!

Cryptanalytic attacks against hash functions

When	Against	Time	Who
1993,1996	md5	2^{16}	[dBBo,Do]
2005	RIPEMD	2^{18}	
2004	SHA0	2^{51}	[JoCaLeJa]
2005	SHA0	2^{40}	[WaFeLaYu]
2005	SHA1	$2^{69}, 2^{63}$	[WaYiYu,WaYaYa]
2009	SHA1	2^{52}	[MHP]
2005,2006	MD5	1 minute	[WaFeLaYu,LeWadW,KI]

md5 is the compression function of MD5
SHA0 is an earlier, weaker version of SHA1

Security of MD5

MD5 is used in 720 different places in Microsoft Windows OS.

What can current attacks do against MD5?

- Find 2 random-looking messages that only differ in 3 bits (boring)
- Find two PDF documents whose hashes collide (more exciting)
- Find two Win32 executables whose hashes collide (very exciting)
- Break deployed cryptographic protocols (very exciting)

Finding collisions

How do attacks work in reality against MD5? Examples:

- Find 2 random-looking messages that only differ in 3 bits
  Cochran's code for MD5:
  `http://www.cs.colorado.edu/~jrblack/md5toolkit.tar.gz`
  Work's in a few minutes on laptop...try it!

- Find 2 Win32 executables whose hashes collide
  Swiss group:
  `http://www.win.tue.nl/hashclash/SoftIntCodeSign/`
  Takes 2 days on a Playstation 3

Status of SHA-1

No collisions yet...

Status of SHA-1

No collisions yet...

You can help find the first ever messages that collide under SHA-1!

http://boinc.iaik.tugraz.at/

SHA3

National Institute for Standards and Technology (NIST) is holding a world-wide competition to develop a new hash function standard.

Contest webpage:
`http://csrc.nist.gov/groups/ST/hash/index.html`

Requested parameters:

- Design: Family of functions with 224, 256, 384, 512 bit output sizes
- Compatibility: existing cryptographic standards
- Security: CR, one-wayness, near-collision resistance, others...
- Efficiency: as fast or faster than SHA-256

SHA3

Submissions: 64

Round 1: 51 **Round 2:** 14

The round 2 functions: BLAKE, Blue Midnight Wish, CubeHash, ECHO, Fugue, Grostl, Hamsi, JH, Keccak, Luffa, Shabal, SHAvite-3, SIMD, Skein.

Final round candidates to be announced in 2010 and winner in 2012.

http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo

One-wayness

Let $H : \{0,1\}^k \times D \to \{0,1\}^n$ be a family of functions.

We say that $x' \in D$ is a pre-image of $y \in \{0,1\}^n$ under H_K if $H_K(x') = y$.

Informally: H is one-way if given y and K it is hard to find a pre-image of y under H_K.

Password verification

- Client A has a password PW and server stores $\overline{PW} = H(PW)$.
- A sends PW to B (over a secure channel) and B checks that $H(PW) = \overline{PW}$

$$A^{PW} \xrightarrow{\quad PW \quad} B^{\overline{PW}}$$

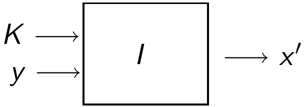
Server compromise results in attacker getting $\overline{PW}$ which should not reveal PW as long as H is one-way, which we will see is a consequence of collision-resistance.

But we will revisit this when we consider dictionary attacks!

One-wayness adversaries

Let $H : \{0,1\}^k \times D \to \{0,1\}^n$ be a family of functions. A OW - adversary I

- gets input a key K
- gets input some $y = H_K(x) \in D$
- Tries to compute a pre-image of y under H_K



Issues in formalizing one-wayness

Suppose $H_K(0^n) = 0^n$ for all K. Then it is easy to invert H_K at $y = 0^n$ because we know a pre-image of 0^n under H_K: it is simply $x' = 0^n$.

Should this mean H is not one-way?

Turns out what is useful is to ask that it be hard to find a pre-image of the image of a random point.

Formal definition of one-wayness

Let $H : \{0,1\}^k \times D \to \{0,1\}^n$ be a family of functions with D finite, and A a OW-adversary.

Game OW_H

procedure Initialize	**procedure** Finalize(x')
$K \xleftarrow{\$} \{0,1\}^k$; $x \xleftarrow{\$} D$; $y \leftarrow H_K(x)$ return K, y	return $(H_K(x') = y)$

The ow-advantage of A is

$$\mathbf{Adv}_H^{\mathrm{ow}}(A) = \Pr[\mathrm{OW}_H^A \Rightarrow \mathit{true}].$$

Generic attacks on one-wayness

For any $H : \{0,1\}^k \times D \rightarrow \{0,1\}^n$

- There is an attack that inverts H in about 2^n trials
- But the birthday attack does not apply.

Does CR imply OW?

Let $H : \{0,1\}^k \times D \rightarrow \{0,1\}^n$.

Given: Adversary A attacking one-wayness of H, meaning $A(K,y)$ returns x_2 satisfying $H_K(x_2) = y$.

Want: Adversary B attacking collision resistance of H, meaning $B(K)$ returns x_1, x_2 satisfying $H_K(x_1) = H_K(x_2)$ and $x_1 \neq x_2$.

Adversary $B(K)$

$x_1 \stackrel{\$}{\leftarrow} D;\ \ y \leftarrow H_K(x_1);\ x_2 \stackrel{\$}{\leftarrow} A(K,y)$

return $\ x_1, x_2$

$$\begin{aligned} A \textit{ succeeds} &\Rightarrow H_K(x_2) = y \\ &\Rightarrow H_K(x_2) = H_K(x_1) \\ &\Rightarrow B \textit{ succeeds?} \end{aligned}$$

Does CR imply OW?

Let $H : \{0,1\}^k \times D \to \{0,1\}^n$.

Given: Adversary A attacking one-wayness of H, meaning $A(K, y)$ returns x_2 satisfying $H_K(x_2) = y$.

Want: Adversary B attacking collision resistance of H, meaning $B(K)$ returns x_1, x_2 satisfying $H_K(x_1) = H_K(x_2)$ and $x_1 \neq x_2$.

Adversary $B(K)$

$x_1 \stackrel{\$}{\leftarrow} D;\ \ y \leftarrow H_K(x_1);\ x_2 \stackrel{\$}{\leftarrow} A(K, y)$

$\mathrm{return}\quad x_1, x_2$

$$\begin{aligned} A \textit{ succeeds } &\Rightarrow\ H_K(x_2) = y \\ &\Rightarrow\ H_K(x_2) = H_K(x_1) \\ &\Rightarrow\ B \textit{ succeeds?} \end{aligned}$$

Problem: May have $x_1 = x_2$.

CR $\not\Rightarrow$ OW

Counter example: Let $H : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be defined by

$$H_K(x) = x$$

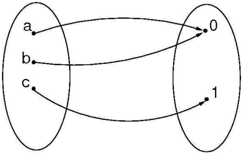
Then

- H is CR since it is impossible to find $x_1 \neq x_2$ with $H_K(x_1) = H_K(x_2)$.
- But H is not one-way since the adversary A that given K, y returns y has ow-advantage 1.

Does CR imply OW?

Adversary $B(K)$
$x_1 \xleftarrow{\$} D$; $y \leftarrow H_K(x_1)$; $x_2 \xleftarrow{\$} A(K, y)$
return x_1, x_2

Inuition: If $|D|$ is sufficiently larger than 2^n, meaning H is compressing, then y is likely to have more than one pre-image, and we are likely to have $x_2 \neq x_1$.



In this case, H being CR will imply it is one way

$\mathrm{CR} \Rightarrow \mathrm{OW}$ for functions that compress

Theorem: Let $H : \{0,1\}^k \times D \to \{0,1\}^n$ be a family of functions. Let A be a ow-adversary with running time at most t. Then there is a cr-adversary B such that

$$\mathbf{Adv}_H^{\mathrm{ow}}(A) \leq 2 \cdot \mathbf{Adv}_H^{\mathrm{cr}}(B) + \frac{2^n}{|D|}.$$

Furthermore the running time of B is about that of A.

Implication: $\mathrm{CR} \Rightarrow \mathrm{OW}$ as long as $2^n/|D|$ is small.

Proof of Theorem

Adversary $B(K)$
$x_1 \xleftarrow{\$} D$; $y \leftarrow H_K(x_1)$; $x_2 \xleftarrow{\$} A(K, y)$
return x_1, x_2

Definition: x_1 is a sibling of x_2 under H_K if x_1, x_2 form a collision for H_K.

For any $K \in \{0,1\}^k$, let

$$S_K = \{x \in D : |H_K^{-1}(H_K(x))| = 1\}$$

be the set of all domain points that have no siblings.

Advantage of B

Adversary $B(K)$

$x_1 \xleftarrow{\$} D;\ \ y \leftarrow H_K(x_1);\ x_2 \xleftarrow{\$} A(K, y)$

return $\ \ x_1, x_2$

Then $\mathbf{Adv}_H^{\mathrm{cr}}(B)$

$$
\begin{aligned}
&= \Pr\left[H_K(x_2) = y \wedge x_1 \neq x_2\right] \\
&= \Pr\left[H_K(x_2) = y \wedge x_1 \neq x_2 \wedge x_1 \notin S_K\right] \\
&= \underbrace{\Pr\left[x_1 \neq x_2 \mid H_K(x_2) = y \wedge x_1 \notin S_K\right]}_{1-\frac{1}{\left|H_K^{-1}(y)\right|} \ \geq\ 1-\frac{1}{2} \ =\ \frac{1}{2}} \cdot \Pr\left[H_K(x_2) = y \wedge x_1 \notin S_K\right]
\end{aligned}
$$

Because A has no information about x_1, barring the fact that $H_K(x_1) = y$.

Advantage of B

Adversary $B(K)$

$x_1 \xleftarrow{\$} D$; $y \leftarrow H_K(x_1)$; $x_2 \xleftarrow{\$} A(K, y)$

return x_1, x_2

$$\mathbf{Adv}_H^{\mathrm{cr}}(B) \geq \frac{1}{2}\Pr\left[H_K(x_2) = y \wedge x_1 \notin S_K\right]$$

Fact: $\Pr\left[E \wedge \overline{F}\right] \geq \Pr[E] - \Pr[F]$

Proof: $\Pr\left[E \wedge \overline{F}\right] = \Pr[E] - \Pr[E \wedge F] \geq \Pr[E] - \Pr[F]$

Apply with

$$E : H_K(x_2) = y \quad \text{and} \quad F : x_1 \in S_K$$

$$\mathbf{Adv}_H^{\mathrm{cr}}(B) \geq \frac{1}{2}\left(\Pr\left[H_K(x_2) = y\right] - \Pr\left[x_1 \in S_K\right]\right)$$

Advantage of B

Adversary $B(K)$

$x_1 \xleftarrow{\$} D$; $y \leftarrow H_K(x_1)$; $x_2 \xleftarrow{\$} A(K, y)$

return x_1, x_2

$$\mathbf{Adv}_H^{\mathrm{cr}}(B) \geq \frac{1}{2}\mathbf{Adv}_H^{\mathrm{ow}}(A) - \frac{\Pr[x_1 \in S_K]}{2}$$

Recall S_K is the set of domain points that have no siblings, so if $\alpha_1, \alpha_2, \ldots, \alpha_s$ are in S_K then $H_K(\alpha_1), H_K(\alpha_2), \ldots, H_K(\alpha_s)$ must be distinct. So

$$|S_K| \leq |\{0,1\}^n| = 2^n.$$

So

$$\Pr[x_1 \in S_K] \leq \frac{2^n}{|D|}.$$