HASH FUNCTIONS
What is a hash function?

By a **hash function** we usually mean a map $h : D \rightarrow \{0, 1\}^n$ 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 $2^{64}$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD4</td>
<td>128</td>
</tr>
<tr>
<td>MD5</td>
<td>128</td>
</tr>
<tr>
<td>SHA1</td>
<td>160</td>
</tr>
<tr>
<td>RIPEMD</td>
<td>128</td>
</tr>
<tr>
<td>RIPEMD-160</td>
<td>160</td>
</tr>
<tr>
<td>SHA-256</td>
<td>256</td>
</tr>
<tr>
<td>Skein</td>
<td>256, 512, 1024</td>
</tr>
</tbody>
</table>
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 $h$. 

![Diagram showing collision resistance](image)
**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 $h$. 

![Diagram](image)
**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.
We consider a family $H : \{0, 1\}^k \times D \rightarrow \{0, 1\}^n$ of functions, meaning for each $K$ we have a map $h = H_K : D \rightarrow \{0, 1\}^n$ defined by

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

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

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

\[
\begin{array}{ccc}
  K \rightarrow & A & \rightarrow x_1, x_2 \\
\end{array}
\]

$A$ 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 $\text{Adv}^{cr}_H(A)$ the probability that $A$ wins.
Let $H : \{0, 1\}^k \times D \rightarrow \{0, 1\}^n$ be a family of functions and $A$ a cr-adversary for $H$.

Game $\text{CR}_H$

- Procedure Initialize
  $K \leftarrow \{0, 1\}^k$
  Return $K$

- Procedure Finalize($x_1, x_2$)
  Return $(x_1 \neq x_2 \land H_K(x_1) = H_K(x_2))$

Let

$$\text{Adv}_{H}^{\text{cr}}(A) = \Pr \left[ \text{CR}_H^A \Rightarrow \text{true} \right].$$
The measure of success

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

$$\text{Adv}^\text{cr}_H(A) = \Pr[\text{CR}^A_H \Rightarrow \text{true}] .$$

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
Adversary advantage depends on its
- strategy
- resources: Running time $t$

**Security:** $H$ is CR if $\text{Adv}^r_H(A)$ 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

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

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

$$H_K(x) = H_K(x[1]x[2]) = AES_K(x[1]) \oplus AES_K(x[2])$$

Is $H$ collision resistant?

Can you design an adversary $A$ such that $H_K(x_1) = H_K(x_2)$?
Let $H: \{0, 1\}^k \times \{0, 1\}^{256} \rightarrow \{0, 1\}^{128}$ be defined by

$$H_K(x) = H_K(x[1]x[2]) = AES_K(x[1]) \oplus AES_K(x[2])$$

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

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

and $A$ is efficient, so $H$ is not CR.
SHA1

algorithm SHA1(M)  // |M| < 2^{64}
    V ← SHF1(5A827999 || 6ED9EBA1 || 8F1BBDCE || CA62C1D6, M)
return V

algorithm SHF1(K, M)  // |K| = 128 and |M| < 2^{64}
    y ← shapad(M)
    Parse y as M_1 || M_2 || · · · || M_n where |M_i| = 512 (1 ≤ i ≤ n)
    V ← 67452301 || EFCDAB89 || 98BADCFE || 10325476 || C3D2E1F0
    for i = 1, . . . , n do
        V ← shf1(K, M_i || V)
    return V

algorithm shapad(M)  // |M| < 2^{64}
    d ← (447 − |M|) mod 512
    Let ℓ be the 64-bit binary representation of |M|
    y ← M || 1 || 0^d || ℓ  // |y| is a multiple of 512
return y
algorithm sha1(K, B || V)  // |K| = 128, |B| = 512 and |V| = 160
Parse B as W0 || W1 || · · · || W15 where |Wi| = 32 (0 ≤ i ≤ 15)
Parse V as V0 || V1 || · · · || V4 where |Vi| = 32 (0 ≤ i ≤ 4)
Parse K as K0 || K1 || K2 || K3 where |Ki| = 32 (0 ≤ i ≤ 3)
for t = 16 to 79 do Wt ← ROTL1(Wt−3 ⊕ Wt−8 ⊕ Wt−14 ⊕ Wt−16)
A ← V0; B ← V1; C ← V2; D ← V3; E ← V4
for t = 0 to 19 do Lt ← K0; Lt+20 ← K1; Lt+40 ← K2; Lt+60 ← K3
for t = 0 to 79 do
  if (0 ≤ t ≤ 19) then f ← (B ∧ C) ∨ ((¬B) ∧ D)
  if (20 ≤ t ≤ 39 OR 60 ≤ t ≤ 79) then f ← B ⊕ C ⊕ D
  if (40 ≤ t ≤ 59) then f ← (B ∧ C) ∨ (B ∧ D) ∨ (C ∧ D)
  temp ← ROTL5(A) + f + E + Wt + Lt
  E ← D; D ← C; C ← ROTL30(B); B ← A; A ← temp
V0 ← V0 + A; V1 ← V1 + B; V2 ← V2 + C; V3 ← V3 + D; V4 ← V4 + E
V ← V0 || V1 || V2 || V3 || V4
return V
Applications of hash functions

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

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

**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 $H(PW) = \overline{PW}$

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

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 $F_A = F_B$

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

**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 $h_A = H(F_B)$.

\[ A^{F_A} \xrightarrow{h_A} B^{F_B} \]

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 $h_A = H(F_B)$.

\[ A^{F_A} \xrightarrow{h_A} B^{F_B} \]

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

<table>
<thead>
<tr>
<th>Adversary $A_1(K)$</th>
<th>Adversary $A_2(K)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 \leftarrow D; y \leftarrow H_K(x_1)$</td>
<td>$x_1 \leftarrow D; y \leftarrow H_K(x_1)$</td>
</tr>
<tr>
<td>For $i = 1, \ldots, q$ do</td>
<td>For $i = 1, \ldots, q$ do</td>
</tr>
<tr>
<td>If ($H_K(D_i) = y \land x_1 \neq D_i$) then</td>
<td>$x_2 \leftarrow D$</td>
</tr>
<tr>
<td>Return $x_1, D_i$</td>
<td>If ($H_K(x_2) = y \land x_1 \neq x_2$) then</td>
</tr>
<tr>
<td>Return FAIL</td>
<td>Return $x_1, x_2$</td>
</tr>
<tr>
<td>Return FAIL</td>
<td>Return FAIL</td>
</tr>
</tbody>
</table>

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 $H : \{0, 1\}^k \times D \rightarrow \{0, 1\}^n$ 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.
for $i = 1, \ldots, q$ do $y_i \leftarrow \{0, 1\}^n$
if $\exists i, j \ (i \neq j$ and $y_i = y_j)$ then COLL $\leftarrow$ true

$$\Pr[\text{COLL}] = C(2^n, q)$$
$$\approx \frac{q^2}{2^{n+1}}$$
Birthday attack

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

**adversary** $A(K)$

for $i = 1, \ldots, q$ do $x_i \leftarrow D$; $y_i \leftarrow H_K(x_i)$

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 \( H : \{0, 1\}^k \times D \rightarrow \{0, 1\}^n. \)

**adversary** \( A(K) \)

for \( i = 1, \ldots, q \) do \( x_i \leftarrow D ; y_i \leftarrow H_K(x_i) \)

if \( \exists i, j \ (i \neq j \text{ and } y_i = y_j \text{ and } x_i \neq x_j) \) 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 \( x_i \leftarrow D ; y_i \leftarrow H_K(x_i) \)

if \( \exists i, j \ (i \neq j \text{ and } y_i = y_j) \) then \( \text{COLL} \leftarrow \text{true} \)

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

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

$\Pr\left[\text{COLL}\right] = C(2^n, q)$

Adversary $A$

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

$\Pr\left[\text{COLL}\right] = ?$

Are the two collision probabilities the same?
Analysis of birthday attack

Birthday

for \( i = 1, \ldots, q \) do
\[
\begin{align*}
y_i &\leftarrow \{0, 1\}^n \\
\text{if } &\exists i, j \ (i \neq j \text{ and } y_i = y_j) \text{ then} \\
&\text{COLL } \leftarrow \text{true}
\end{align*}
\]
\[
\Pr[\text{COLL}] = C(2^n, q)
\]

Adversary \( A \)

for \( i = 1, \ldots, q \) do
\[
\begin{align*}
x_i &\leftarrow D \; ; \; y_i \leftarrow H_K(x_i) \\
\text{if } &\exists i, j \ (i \neq j \text{ and } y_i = y_j) \text{ then} \\
&\text{COLL } \leftarrow \text{true}
\end{align*}
\]
\[
\Pr[\text{COLL}] = ?
\]

Are the two collision probabilities the same?
Not necessarily, because

- on the left \( y_i \leftarrow \{0, 1\}^n \)
- on the right \( x_i \leftarrow D \; ; \; y_i \leftarrow H_K(x_i) \)
Analysis of birthday attack

Consider the following processes

Process 1
\[ y \leftarrow \{0, 1\}^n \]
return \( y \)

Process 2
\[ x \leftarrow D; \quad y \leftarrow H_K(x) \]
return \( y \)

Process 1 certainly returns a random \( n \)-bit string. Does Process 2?
Analysis of birthday attack

Process 1
\[ y \leftarrow \{0, 1\} \]
return \( y \)

Process 2
\[ x \leftarrow \{a, b, c, d\}; \ y \leftarrow H_K(x) \]
return \( y \)

\[
\begin{align*}
\Pr[y = 0] &= \\
\Pr[y = 1] &=
\end{align*}
\]

\[
\begin{align*}
\Pr[y = 0] &= \\
\Pr[y = 1] &=
\end{align*}
\]
Analysis of birthday attack

Process 1
\[ y \leftarrow \{0, 1\} \]
return \( y \)

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

Process 2
\[ x \leftarrow \{a, b, c, d\} \; ; \; y \leftarrow H_K(x) \]
return \( y \)

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

Pr\[y = 0\] = \( \frac{3}{4} \)
Pr\[y = 1\] = \( \frac{1}{4} \)
Analysis of birthday attack

Process 1
\[ y \leftarrow \{0, 1\} \]
return \( y \)

Process 2
\[ x \leftarrow \{a, b, c, d\}; \quad y \leftarrow H_K(x) \]
return \( y \)

Pr[\( y = 0 \)] =
Pr[\( y = 1 \)] =
Pr[\( y = 0 \)] =
Pr[\( y = 1 \)] =
\[
\frac{27}{62}
\]
Analysis of birthday attack

Process 1
\[ y \leftarrow \{0, 1\} \]
return \( y \)

Process 2
\[ x \leftarrow \{a, b, c, d\} ; y \leftarrow H_K(x) \]
return \( y \)

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

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

We say that $H : \{0, 1\}^k \times D \rightarrow \{0, 1\}^n$ 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

$y \leftarrow \{0, 1\}^n$

return $y$

Process 2

$x \leftarrow D$; $y \leftarrow H_K(x)$

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

<table>
<thead>
<tr>
<th>Function</th>
<th>$n$</th>
<th>$T_B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD4</td>
<td>128</td>
<td>$2^{64}$</td>
</tr>
<tr>
<td>MD5</td>
<td>128</td>
<td>$2^{64}$</td>
</tr>
<tr>
<td>SHA1</td>
<td>160</td>
<td>$2^{80}$</td>
</tr>
<tr>
<td>RIPEMD-160</td>
<td>160</td>
<td>$2^{80}$</td>
</tr>
<tr>
<td>SHA256</td>
<td>256</td>
<td>$2^{128}$</td>
</tr>
</tbody>
</table>

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

\[ H : \{0, 1\}^k \times D \rightarrow \{0, 1\}^n \]

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

- First build a CR compression function \( h : \{0, 1\}^k \times \{0, 1\}^{b+n} \rightarrow \{0, 1\}^n \).
- 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 $M = M[1] \ldots M[\ell]$ where $\ell = \|M\|_b$.
- $\langle i \rangle$ denote the $b$-bit binary representation of $i \in \{0, \ldots, 2^b - 1\}$.
- $D$ be the set of all strings of at most $2^b - 1$ blocks, so that $\|M\|_b \in \{0, \ldots, 2^b - 1\}$ 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 \( V[m + 1] \)
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.
Theorem: Let $h : \{0, 1\}^k \times \{0, 1\}^{b+n} \rightarrow \{0, 1\}^n$ be a family of functions and let $H : \{0, 1\}^k \times D \rightarrow \{0, 1\}^n$ be obtained from $h$ via the MD transform. Then for any cr-adversary $A_H$ there exists a cr-adversary $A_h$ such that

$$Adv^c_H(A_H) \leq Adv^c_h(A_h)$$

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:

$$h \text{ CR } \Rightarrow Adv^c_H(A_h) \text{ small}$$

$$\Rightarrow Adv^c_H(A_H) \text{ small}$$

$$\Rightarrow H \text{ CR}$$
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.
Case 1: $\|M_1\|_b \neq \|M_2\|_b$

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

\[ M_1[1] \xrightarrow{h_k} v_1[1] \xrightarrow{h_k} v_1[2] \xrightarrow{\langle 2 \rangle} v_1[3] = H_K(M_1) \]

\[ M_2[1] \xrightarrow{h_k} v_2[1] \xrightarrow{h_k} v_2[2] \xrightarrow{\langle 2 \rangle} v_2[3] = H_K(M_2) \]

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

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 $x_1, x_2$

Case 2: \( \| M_1 \|_b = \| M_2 \|_b \)

\[
\begin{align*}
\text{If } x_1 \neq x_2 & \text{ then return } x_1, x_2 \\
& \text{If } x_1 \neq x_2 \text{ then return } x_1, x_2
\end{align*}
\]
Case 2: \( \| M_1 \|_b = \| M_2 \|_b \)

\[
\begin{align*}
M_1[1] & \xrightarrow{h_K} v_1[1] \xrightarrow{v_1[2]} h_K \xrightarrow{v_1[3]} = H_K(M_1) \\
M_2[1] & \xrightarrow{h_K} v_2[1] \xrightarrow{v_2[2]} h_K \xrightarrow{v_2[3]} = H_K(M_2)
\end{align*}
\]

\[
x_1 \leftarrow \langle 2 \rangle \| V_1[2] \ ;
\]

If \( x_1 \neq x_2 \) then return \( x_1, x_2 \)


\[
x_1 \leftarrow M_1[2] \| V_1[1] \ ;
\]

If \( x_1 \neq x_2 \) then return \( x_1, x_2 \)

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 $x_1, x_2$


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

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 $E : \{0, 1\}^b \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ 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?
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 $E : \{0, 1\}^{512} \times \{0, 1\}^{160} \to \{0, 1\}^{160}$. 
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

<table>
<thead>
<tr>
<th>When</th>
<th>Against</th>
<th>Time</th>
<th>Who</th>
</tr>
</thead>
<tbody>
<tr>
<td>1993,1996</td>
<td>md5</td>
<td>$2^{16}$</td>
<td>[dBBo,Do]</td>
</tr>
<tr>
<td>2005</td>
<td>RIPEMD</td>
<td>$2^{18}$</td>
<td></td>
</tr>
<tr>
<td>2004</td>
<td>SHA0</td>
<td>$2^{51}$</td>
<td>[JoCaLeJa]</td>
</tr>
<tr>
<td>2005</td>
<td>SHA0</td>
<td>$2^{40}$</td>
<td>[WaFeLaYu]</td>
</tr>
<tr>
<td>2005</td>
<td>SHA1</td>
<td>$2^{69}, 2^{63}$</td>
<td>[WaYiYu,WaYaYa]</td>
</tr>
<tr>
<td>2009</td>
<td>SHA1</td>
<td>$2^{52}$</td>
<td>[MHP]</td>
</tr>
<tr>
<td>2005,2006</td>
<td>MD5</td>
<td>1 minute</td>
<td>[WaFeLaYu, LeWadW, Kl]</td>
</tr>
</tbody>
</table>

- md5 is the compression function of MD5
- SHA0 is an earlier, weaker version of SHA1
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:
  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/
National Institute for Standards and Technology (NIST) is holding a world-wide competition to develop a new hash function standard.

Contest webpage:

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


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

http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo
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 $H(PW) = \overline{PW}$.

\[
A^{PW} \quad PW \quad B^{\overline{PW}}
\]

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

$$\begin{array}{c}
K \rightarrow I \\
y \rightarrow I \\
\rightarrow x'
\end{array}$$
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 \( H : \{0, 1\}^k \times D \rightarrow \{0, 1\}^n \) be a family of functions with \( D \) finite, and \( A \) a OW-adversary.

**Game** \( \text{OW}_H \)

**procedure** Initialize

\[
K \leftarrow \{0, 1\}^k; \\
x \leftarrow D; \\
y \leftarrow H_K(x) \\
\text{return } K, y
\]

**procedure** Finalize\((x')\)

\[
\text{return } (H_K(x') = y)
\]

The ow-advantage of \( A \) is

\[
\text{Adv}^\text{ow}_H(A) = \Pr[\text{OW}^A_H \Rightarrow \text{true}].
\]
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 $B(K)$**

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

return $x_1, x_2$

$A succeeds \Rightarrow H_K(x_2) = y$

$\Rightarrow H_K(x_2) = H_K(x_1)$

$\Rightarrow B succeeds?$
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 \( B(K) \)**

\[ x_1 \leftarrow D; \quad y \leftarrow H_K(x_1); \quad x_2 \leftarrow A(K, y) \]

return \( x_1, x_2 \)

\[ \text{A succeeds} \implies H_K(x_2) = y \]

\[ \implies H_K(x_2) = H_K(x_1) \]

\[ \implies B \text{ succeeds?} \]

**Problem:** May have \( x_1 = x_2 \).
Counter example: Let $H : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ 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** $B(K)$

\[
x_1 \leftarrow D; \quad y \leftarrow H_K(x_1); \quad x_2 \leftarrow A(K, y)
\]

return $x_1, x_2$

**Intuition:** 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.
**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

$$\text{Adv}_H^{\text{OW}}(A) \leq 2 \cdot \text{Adv}_H^{\text{CR}}(B) + \frac{2^n}{|D|}.$$ 

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

**Implication:** CR $\Rightarrow$ OW as long as $2^n/|D|$ is small.
Proof of Theorem

**Adversary** $B(K)$

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

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 $B(K)$**

\[
x_1 \leftarrow \$ D; \quad y \leftarrow H_K(x_1); \quad x_2 \leftarrow \$ A(K, y)
\]

return $x_1, x_2$

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

\[
= \Pr[H_K(x_2) = y \land x_1 \neq x_2]
\]

\[
= \Pr[H_K(x_2) = y \land x_1 \neq x_2 \land x_1 \notin S_K]
\]

\[
= \Pr[x_1 \neq x_2 \mid H_K(x_2) = y \land x_1 \notin S_K] \cdot \Pr[H_K(x_2) = y \land x_1 \notin S_K]
\]

\[
1 - \frac{1}{|H_K^{-1}(y)|} \geq 1 - \frac{1}{2} = \frac{1}{2}
\]

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

Adversary $B(K)$

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

return $x_1, x_2$

$$\text{Adv}_{H}^{\text{cr}}(B) \geq \frac{1}{2} \Pr[H_K(x_2) = y \land x_1 \notin S_K]$$

Fact: $\Pr[E \land \overline{F}] \geq \Pr[E] - \Pr[F]$

Proof: $\Pr[E \land \overline{F}] = \Pr[E] - \Pr[E \land F] \geq \Pr[E] - \Pr[F]$

Apply with

$E : H_K(x_2) = y$ and $F : x_1 \in S_K$

$$\text{Adv}_{H}^{\text{cr}}(B) \geq \frac{1}{2} (\Pr[H_K(x_2) = y] - \Pr[x_1 \in S_K])$$
Advantage of $B$

Adversary $B(K)$

$x_1 \leftarrow^S D; \quad y \leftarrow H_K(x_1); \quad x_2 \leftarrow^S A(K, y)$

return $x_1, x_2$

$$
\text{Adv}^\text{cr}_{H}(B) \geq \frac{1}{2} \text{Adv}^\text{ow}_{H}(A) - \frac{\Pr[x_1 \in S_K]}{2}
$$

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

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