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 $h$. 
**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 \( h \).
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 $h$.

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

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

\[ K \xrightarrow{A} x_1, x_2 \]

$A$ 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 $\text{Adv}_{H}^{\text{cr}}(A)$ 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**\( (x_1, x_2) \)

  \( \text{Return } (x_1 \neq x_2 \land H_K(x_1) = H_K(x_2)) \)

Let

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

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

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

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

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

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

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 \( A \)

\[
K \rightarrow A \quad \rightarrow \quad x_1 = x_1[1]x_1[2] \\
\quad \quad x_2 = x_2[1]x_2[2]
\]

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 $A$**

$x_1 \leftarrow 0^{128}1^{128}$; $x_2 \leftarrow 1^{128}0^{128}$; return $x_1, x_2$

Then

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

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

algorithm SHA1(M)  // |M| < 2^{64}
    V ← SHF1(5A827999 || 6ED9EBA1 || 8F1BFCDC || 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 || EFCDBA58 || 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 shf1($K$, $B$ || $V$) // $|K| = 128$, $|B| = 512$ and $|V| = 160$

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

Parse $V$ as $V_0$ || $V_1$ || ··· || $V_4$ where $|V_i| = 32$ ($0 \leq i \leq 4$)

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

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

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

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 ($0 \leq t \leq 19$) then $f \leftarrow (B \land C) \lor ((\neg B) \land D)$
  if ($20 \leq t \leq 39$ OR $60 \leq t \leq 79$) then $f \leftarrow B \oplus C \oplus D$
  if ($40 \leq t \leq 59$) then $f \leftarrow (B \land C) \lor (B \land D) \lor (C \land D)$

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

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

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

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

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)

\[ A^{PW} \xrightarrow{PW} B^{PW} \]

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

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!
• 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 $F_A = F_B$

\[ A^{F_A} \quad F_A \quad B^{F_B} \]

**Problem:** Transmission could take forever, particularly if the link is slow (DSL).
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 $h_A = H(F_B)$.

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.
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 $A_1(K)$**

$x_1 \leftarrow D; y \leftarrow H_K(x_1)$

For $i = 1, \ldots, q$ do

If $(H_K(D_i) = y \land x_1 \neq D_i)$ then

Return $x_1, D_i$

Return FAIL

**Adversary $A_2(K)$**

$x_1 \leftarrow D; y \leftarrow H_K(x_1)$

For $i = 1, \ldots, q$ do

$x_2 \leftarrow D$

If $(H_K(x_2) = y \land x_1 \neq x_2)$ 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.
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 $\text{COLL} \leftarrow \text{true}$

$$\Pr[\text{COLL}] = C(2^n, q) \approx \frac{q^2}{2^{n+1}}$$
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 $\exists i, j$ ($i \neq j$ and $y_i = y_j$ and $x_i \neq x_j$) then return $x_i, x_j$

else return $\text{FAIL}$

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

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

Adversary $A$

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

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

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

### Birthday

<table>
<thead>
<tr>
<th>for $i = 1, \ldots, q$ do</th>
<th>for $i = 1, \ldots, q$ do</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_i \leftarrow {0, 1}^n$</td>
<td>$x_i \leftarrow D$; $y_i \leftarrow H_K(x_i)$</td>
</tr>
<tr>
<td>if $\exists i, j (i \neq j$ and $y_i = y_j)$ then</td>
<td>if $\exists i, j (i \neq j$ and $y_i = y_j)$ then</td>
</tr>
<tr>
<td>$\text{COLL} \leftarrow \text{true}$</td>
<td>$\text{COLL} \leftarrow \text{true}$</td>
</tr>
</tbody>
</table>

Pr[COLL] = $C(2^n, q)$

### Adversary $A$

Pr[COLL] = ?

Are the two collision probabilities the same?

Not necessarily, because

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

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

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

\[
\begin{align*}
\Pr[y = 0] &= \frac{26}{62} \\
\Pr[y = 1] &= \frac{26}{62}
\end{align*}
\]
Analysis of birthday attack

Process 1

\[ y \leftarrow \{0, 1\} \]

\[ \text{return } y \]

Process 2

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

\[ \text{return } y \]

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

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

\[ \Pr[y = 0] = \]

\[ \Pr[y = 1] = \]
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
\[ x \leftarrow \{a, b, c, d\}; y \leftarrow H_K(x) \]
return \( y \)

\[
\Pr[y = 0] = \frac{27}{62}
\]
\[
\Pr[y = 1] = \frac{27}{62}
\]
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**

\[
\begin{align*}
  y & \gets \{0, 1\}^n \\
  \text{return } y
\end{align*}
\]

**Process 2**

\[
\begin{align*}
  x & \gets D; \\
  y & \gets H_K(x) \\
  \text{return } y
\end{align*}
\]
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 \( h : \{0, 1\}^k \times \{0, 1\}^{b+n} \rightarrow \{0, 1\}^n \) 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

\[
h : \{0, 1\}^k \times \{0, 1\}^{672} \rightarrow \{0, 1\}^{160}
\]
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.
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 $h : \{0, 1\}^k \times \{0, 1\}^{b+n} \to \{0, 1\}^n$.

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

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.
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
\[
\text{Adv}^{\text{cr}}_H(A_H) \leq \text{Adv}^{\text{cr}}_h(A_h)
\]
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:

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

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

$\Rightarrow H \text{ CR }$
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 $x_1 = \langle 2 \rangle \| V_1[2] \text{ and } x_2 = \langle 1 \rangle \| V_2[1]$. 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: \( \| M_1 \|_b = \| M_2 \|_b \)

\[
\begin{align*}
&M_1[1] \quad M_1[2] \quad \langle 2 \rangle \\
&h_k \quad h_k \quad h_k \\
\text{0}^n \quad v_1[1] \quad v_1[2] \quad v_1[3] = H_K(M_1)
\end{align*}
\]

\[
\begin{align*}
&M_2[1] \quad M_2[2] \quad \langle 2 \rangle \\
&h_k \quad h_k \quad h_k \\
\text{0}^n \quad v_2[1] \quad v_2[2] \quad v_2[3] = H_K(M_2)
\end{align*}
\]

\[
x_1 \leftarrow \langle 2 \rangle \| V_1[2] \right ; \quad x_2 \leftarrow \langle 2 \rangle \| V_2[2] \\
\text{If } x_1 \neq x_2 \text{ then return } x_1, x_2 \\
\text{Else } // \ V_1[2] = V_2[2]
\]
Case 2: $\|M_1\|_b = \|M_2\|_b$

\[
x_1 \leftarrow \langle 2 \rangle \| V_1[2] \; \; ; \; x_2 \leftarrow \langle 2 \rangle \| V_2[2]
If \; x_1 \neq x_2 \; \text{then return} \; x_1, x_2
\]

\[
x_1 \leftarrow M_1[2]\| V_1[1] \; \; ; \; x_2 \leftarrow M_2[2]\| V_2[1]
If \; x_1 \neq x_2 \; \text{then return} \; x_1, x_2
\]
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 $x_1, x_2$


$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 // $V_1[1] = V_2[1]$
Case 2: \( \| M_1 \|_b = \| M_2 \|_b \)

\[
\begin{align*}
M_1[1] & \quad \quad \quad M_1[2] & \quad \quad \quad <2> \\
\quad \quad \quad h_K & \quad \quad \quad \quad \quad \; v_1[1] & \quad \quad \; v_1[2] & \quad \quad \; v_1[3] = H_K(M_1) \\
0^n & \quad \quad \quad \quad \; h_K & \quad \quad \; v_1[2] & \quad \quad \; v_1[3] = H_K(M_1)
\end{align*}
\]

\[
\begin{align*}
M_2[1] & \quad \quad \quad M_2[2] & \quad \quad \quad <2> \\
\quad \quad \quad h_K & \quad \quad \quad \quad \quad \; v_2[1] & \quad \quad \; v_2[2] & \quad \quad \; v_2[3] = H_K(M_2) \\
0^n & \quad \quad \quad \quad \; h_K & \quad \quad \; v_2[2] & \quad \quad \; v_2[3] = H_K(M_2)
\end{align*}
\]

\[
\begin{align*}
x_1 & \leftarrow <2>\| V_1[2] ; \quad x_2 \leftarrow <2>\| V_2[2] \\
\text{If } x_1 \neq x_2 \text{ then return } x_1, x_2 \\
\text{Else } \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \text{// } V_1[2] = V_2[2] \\
\quad x_1 & \leftarrow M_1[2]\| V_1[1] ; \quad x_2 \leftarrow M_2[2]\| V_2[1] \\
\text{If } x_1 \neq x_2 \text{ then return } x_1, x_2 \\
\text{Else } \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \text{// } V_1[1] = V_2[1] \\
\quad x_1 & \leftarrow M_1[1]\| 0^n ; \quad x_2 \leftarrow M_2[1]\| 0^n \\
\text{Return } x_1, x_2
\end{align*}
\]
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 \( x_1 \neq x_2 \)

\[
y \leftarrow E_{x_1}(v_1) ; \quad v_2 \leftarrow E_{x_2}^{-1}(y)
\]

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

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

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 $E : \{0, 1\}^{512} \times \{0, 1\}^{160} \rightarrow \{0, 1\}^{160}$. 
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
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:
  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
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, SHA\text{v}ite-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
Let $H : \{0, 1\}^k \times D \rightarrow \{0, 1\}^n$ 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!
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 \( H_K \)

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

**Game $OW_H$**

**procedure** Initialize

$K \leftarrow \{0, 1\}^k$;

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

return $K, y$

**procedure** Finalize($x'$)

return $(H_K(x') = y)$

The ow-advantage of $A$ is

$$Adv_{H}^{OW}(A) = \Pr[OW_{H}^{A} \Rightarrow true].$$
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 \leftarrow \$ D; $y \leftarrow H_K(x_1)$; $x_2 \leftarrow \$ A(K, y)

return $x_1, x_2$

$$A \text{ succeeds } \Rightarrow H_K(x_2) = y$$

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

$$\Rightarrow B \text{ succeeds?}$$
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; $y \leftarrow H_K(x_1); x_2 \leftarrow \$ A(K, y)$

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?

**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 $B(K)$

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

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

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

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** $B(K)$

\[
\begin{align*}
    &x_1 \leftarrow D; \ y \leftarrow H_K(x_1); \ x_2 \leftarrow A(K, y) \\
    &\text{return } x_1, x_2
\end{align*}
\]

**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 $B(K)$**

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

return $x_1, x_2$

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

\[
= \Pr [H_K(x_2) = y \land x_1 \neq x_2] \\
= \Pr [H_K(x_2) = y \land x_1 \neq x_2 \land x_1 \notin S_K] \\
= \Pr [x_1 \neq x_2 \mid H_K(x_2) = y \land x_1 \notin S_K] \cdot \Pr [H_K(x_2) = y \land x_1 \notin S_K] \\
\geq 1 - \frac{1}{|H_{K^{-1}}(y)|} \geq 1 - \frac{1}{2} = \frac{1}{2}
\]

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 $x_1, x_2$

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

**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

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

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

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