BLOCK CIPHERS
and PSEUDO-RANDOM FUNCTIONS
Recall: Block Cipher Definition

Let $E: \text{Keys} \times D \rightarrow R$ be a family of functions. We say that $E$ is a block cipher if

- $R = D$, meaning the input and output spaces are the same set.
- $E_K : D \rightarrow D$ is a permutation for every key $K \in \text{Keys}$, meaning has an inverse $E_K^{-1} : D \rightarrow D$ such that $E_K^{-1}(E_K(x)) = x$ for all $x \in D$.

We let $E^{-1} : \text{Keys} \times D \rightarrow D$, defined by $E^{-1}(K, y) = E_K^{-1}(y)$, be the inverse block cipher to $E$.

In practice we want that $E, E^{-1}$ are efficiently computable.

If $\text{Keys} = \{0, 1\}^k$ then $k$ is the key length as before. If $D = \{0, 1\}^{\ell}$ we call $\ell$ the block length.
Target Key Recovery: Informally

We consider two measures (metrics) for how well the adversary does at this key recovery task:

- Target key recovery (TKR)
- Consistent key recovery (KR)

Informally, let $E : \text{Keys} \times D \rightarrow R$ be a family of functions. It is known to the adversary $A$.

- A target key $K \leftarrow^\$ \text{Keys}$ is selected by the game, but not given to $A$.
- $A$ can submit a plaintext $M \in D$ to the game and get back $C = E(K, M)$, in this way gathering input-output examples $(M_1, C_1), \ldots, (M_q, C_q)$ of $E_K$.
- $A$ outputs a “guess” $K'$
- $A$ wins if $K'$ equals the target key $K$.
- $A$'s tkr advantage is the probability that it wins.
### Target Key Recovery Definitions: Game and Advantage

<table>
<thead>
<tr>
<th>Game $\text{TKR}_E$</th>
<th>procedure $\text{Fn}(M)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{procedure Initialize}$</td>
<td>Return $E(K, M)$</td>
</tr>
<tr>
<td>$K \leftarrow$ Keys</td>
<td>$\text{procedure Finalize}(K')$</td>
</tr>
<tr>
<td></td>
<td>Return $(K = K')$</td>
</tr>
</tbody>
</table>

**Definition:** $\text{Adv}^{\text{tkr}}_E(A) = \Pr[\text{TKR}_E^A \Rightarrow \text{true}]$.

- First Initialize executes, selecting *target key* $K \leftarrow$ Keys, but not giving it to $A$.
- Now $A$ can call (query) $\text{Fn}$ on any input $M \in D$ of its choice to get back $C = E_K(M)$. It can make as many queries as it wants.
- Eventually $A$ will halt with an output $K'$ which is automatically viewed as the input to $\text{Finalize}$.
- The game returns whatever $\text{Finalize}$ returns.
- The tkr advantage of $A$ is the probability that the game returns true.
Consistent Key Recovery Definitions: Game and Advantage

Let $E: \text{Keys} \times D \rightarrow R$ be a family of functions, and $A$ an adversary.

<table>
<thead>
<tr>
<th>Game $\text{KR}_E$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>procedure</strong> Initialize</td>
</tr>
<tr>
<td>$K \leftarrow $ \text{Keys}; i \leftarrow 0$</td>
</tr>
<tr>
<td><strong>procedure</strong> $\text{Fn}(M)$</td>
</tr>
<tr>
<td>$i \leftarrow i + 1; M_i \leftarrow M$</td>
</tr>
<tr>
<td>$C_i \leftarrow E(K, M_i)$</td>
</tr>
<tr>
<td>Return $C_i$</td>
</tr>
<tr>
<td><strong>procedure</strong> $\text{Finalize}(K')$</td>
</tr>
<tr>
<td>win $\leftarrow$ true</td>
</tr>
<tr>
<td>For $j = 1, \ldots, i$ do</td>
</tr>
<tr>
<td>If $E(K', M_j) \neq C_j$ then win $\leftarrow$ false</td>
</tr>
<tr>
<td>If $M_j \in {M_1, \ldots, M_{j-1}}$ then win $\leftarrow$ false</td>
</tr>
<tr>
<td>Return win</td>
</tr>
</tbody>
</table>

**Definition:** $\text{Adv}^{kr}_E(A) = \Pr[\text{KR}_E^A \Rightarrow \text{true}].$

The game returns true if (1) The key $K'$ returned by the adversary is consistent with $(M_1, C_1), \ldots, (M_q, C_q)$, and (2) $M_1, \ldots, M_q$ are distinct. $A$ is a $q$-query adversary if it makes $q$ distinct queries to its $\text{Fn}$ oracle.
kr advantage always exceeds tkr advantage

**Fact:** Suppose that, in game $KR_E$, adversary $A$ makes queries $M_1, \ldots, M_q$ to $Fn$, thereby defining $C_1, \ldots, C_q$. Then the target key $K$ is consistent with $(M_1, C_1), \ldots, (M_q, C_q)$.

**Proposition:** Let $E$ be a family of functions. Let $A$ be any adversary all of whose $Fn$ queries are distinct. Then

$$\text{Adv}_{E}^{kr}(A) \geq \text{Adv}_{E}^{tkr}(A).$$

**Why?** If the $K'$ that $A$ returns equals the target key $K$, then, by the Fact, the input-output examples $(M_1, C_1), \ldots, (M_q, C_q)$ will of course be consistent with $K'$.
Exhaustive Key Search attack

Let $E: \text{Keys} \times D \to R$ be a function family with $\text{Keys} = \{T_1, \ldots, T_N\}$ and $D = \{x_1, \ldots, x_d\}$. Let $1 \leq q \leq d$ be a parameter.

**adversary** $A_{\text{eks}}$

For $j = 1, \ldots, q$ do $M_j \leftarrow x_j$; $C_j \leftarrow Fn(M_j)$

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

if $(\forall j \in \{1, \ldots, q\} : E(T_i, M_j) = C_j)$ then return $T_i$

**Question:** What is $\text{Adv}^E_{kr}(A_{\text{eks}})$?
Exhaustive Key Search attack

Let \( E: \text{Keys} \times D \rightarrow R \) be a function family with \( \text{Keys} = \{ T_1, \ldots, T_N \} \) and \( D = \{ x_1, \ldots, x_d \} \). Let \( 1 \leq q \leq d \) be a parameter.

**adversary** \( A_{\text{eks}} \)

For \( j = 1, \ldots, q \) do \( M_j \leftarrow x_j; \; C_j \leftarrow \text{Fn}(M_j) \)

For \( i = 1, \ldots, N \) do

\[
\text{if } (\forall j \in \{1, \ldots, q\} : E(T_i, M_j) = C_j) \text{ then return } T_i
\]

**Question:** What is \( \text{Adv}^\text{kr}_E(A_{\text{eks}}) \)?

**Answer:** It equals 1.

Because

- There is some \( i \) such that \( T_i = K \), and
- \( K \) is consistent with \( (M_1, C_1), \ldots, (M_q, C_q) \).
Exhaustive Key Search attack

Let $E : \text{Keys} \times D \to R$ be a function family with $\text{Keys} = \{ T_1, \ldots, T_N \}$ and $D = \{ x_1, \ldots, x_d \}$. Let $1 \leq q \leq d$ be a parameter.

**adversary $A_{eks}$**

For $j = 1, \ldots, q$ do $M_j \leftarrow x_j$; $C_j \leftarrow \text{Fn}(M_j)$

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

if $(\forall j \in \{1, \ldots, q\} : E(T_i, M_j) = C_j)$ then return $T_i$

**Question:** What is $\text{Adv}_{E}^{tkr}(A_{eks})$?
Exhaustive Key Search attack

Let $E: \text{Keys} \times D \rightarrow R$ be a function family with $\text{Keys} = \{ T_1, \ldots, T_N \}$ and $D = \{ x_1, \ldots, x_d \}$. Let $1 \leq q \leq d$ be a parameter.

**adversary $A_{\text{eks}}$**

For $j = 1, \ldots, q$ do $M_j \leftarrow x_j$; $C_j \leftarrow F_n(M_j)$

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

if $(\forall j \in \{1, \ldots, q\} : E(T_i, M_j) = C_j)$ then return $T_i$

**Question:** What is $\text{Adv}_{E}^{\text{tkr}}(A_{\text{eks}})$?

**Answer:** Hard to say! Say $K = T_m$ but there is a $i < m$ such that $E(T_i, M_j) = C_j$ for $1 \leq j \leq q$. Then $T_i$, rather than $K$, is returned.

In practice if $E: \{0, 1\}^k \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell$ is a “real” block cipher and $q > k/\ell$, we expect that $\text{Adv}_{E}^{\text{tkr}}(A_{\text{eks}})$ is close to 1 because $K$ is likely the only key consistent with the input-output examples.
How long does exhaustive key search take?

DES can be computed at 1.6 Gbits/sec in hardware.

DES plaintext = 64 bits

Chip can perform \((1.6 \times 10^9)/64 = 2.5 \times 10^7\) DES computations per second

Expect \(A_{\text{eks}} (q = 1)\) to succeed in \(2^{55}\) DES computations, so it takes time

\[
\frac{2^{55}}{2.5 \times 10^7} \approx 1.4 \times 10^9 \text{ seconds}
\]

\[
\approx 45 \text{ years!}
\]

Key Complementation \(\Rightarrow 22.5\) years

But this is prohibitive. Does this mean DES is secure?
Exhaustive key search is a generic attack: Did not attempt to “look inside” DES and find/exploit weaknesses.

The following non-generic key-recovery attacks on DES have advantage close to one and running time smaller than $2^{56}$ DES computations:

<table>
<thead>
<tr>
<th>Attack</th>
<th>when</th>
<th>$q$, running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differential cryptanalysis</td>
<td>1992</td>
<td>$2^{47}$</td>
</tr>
<tr>
<td>Linear cryptanalysis</td>
<td>1993</td>
<td>$2^{44}$</td>
</tr>
</tbody>
</table>
Exhaustive key search is a generic attack: Did not attempt to “look inside” DES and find/exploit weaknesses.

The following non-generic key-recovery attacks on DES have advantage close to one and running time smaller than $2^{56}$ DES computations:

<table>
<thead>
<tr>
<th>Attack</th>
<th>when</th>
<th>$q$, running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differential cryptanalysis</td>
<td>1992</td>
<td>$2^{47}$</td>
</tr>
<tr>
<td>Linear cryptanalysis</td>
<td>1993</td>
<td>$2^{44}$</td>
</tr>
</tbody>
</table>

But merely storing $2^{44}$ input-output pairs requires 281 Terabytes.

In practice these attacks were prohibitively expensive.
adversary $A_{eks}$

For $j = 1, \ldots, q$ do $M_j \leftarrow x_j$; $C_j \leftarrow \text{Fn}(M_j)$

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

if $(\forall j \in \{1, \ldots, q\} : E(T_i, M_j) = C_j)$ then return $T_i$
adversary $A_{\text{eks}}$

For $j = 1, \ldots, q$ do $M_j \leftarrow x_j; C_j \leftarrow \text{Fn}(M_j)$

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

if $(\forall j \in \{1, \ldots, q\} : E(T_i, M_j) = C_j)$ then return $T_i$

Observation: The $E$ computations can be performed in parallel!
**adversary** $A_{\text{eks}}$

For $j = 1, \ldots, q$ do $M_j \leftarrow x_j$; $C_j \leftarrow \text{Fn}(M_j)$

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

if ($\forall j \in \{1, \ldots, q\} : E(T_i, M_j) = C_j$) then return $T_i$

**Observation:** The $E$ computations can be performed in parallel!

In 1993, Wiener designed a dedicated DES-cracking machine:

- $1$ million
- 57 chips, each with many, many DES processors
- Finds key in 3.5 hours
**RSA DES challenges**

\[ K \leftarrow^\$ \{0, 1\}^{56}; \ Y \leftarrow \text{DES}(K, X); \text{ Publish } Y \text{ on website.} \]

Reward for recovering \( X \)

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Post Date</th>
<th>Reward</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>1997</td>
<td>$10,000</td>
<td>Distributed.Net: 4 months</td>
</tr>
<tr>
<td>II</td>
<td>1998</td>
<td>Depends how fast you find key</td>
<td>Distributed.Net: 41 days. EFF: 56 hours</td>
</tr>
<tr>
<td>III</td>
<td>1998</td>
<td>As above</td>
<td>(&lt; 28 \text{ hours})</td>
</tr>
</tbody>
</table>
DES is considered broken because its short key size permits rapid key search.

But DES is a very strong design as evidenced by the fact that there are no practical attacks that exploit its structure.
Block cipher $2\text{DES} : \{0, 1\}^{112} \times \{0, 1\}^{64} \rightarrow \{0, 1\}^{64}$ is defined by

$$2\text{DES}_{K_1K_2}(M) = \text{DES}_{K_2}(\text{DES}_{K_1}(M))$$

- Exhaustive key search takes $2^{112}$ $\text{DES}$ computations, which is too much even for machines
- Resistant to differential and linear cryptanalysis.
Meet-in-the-middle attack on 2DES

Suppose $K_1 K_2$ is a target 2DES key and adversary has $M, C$ such that

$$C = 2DES_{K_1 K_2}(M) = DES_{K_2}(DES_{K_1}(M))$$

Then

$$DES^{-1}_{K_2}(C) = DES_{K_1}(M)$$
Meet-in-the-middle attack on 2DES

Suppose $DES^{-1}_{K_2}(C) = DES_{K_1}(M)$ and $T_1, \ldots, T_N$ are all possible DES keys, where $N = 2^{56}$.

\[
\begin{array}{|c|c|}
\hline
T_1 & DES(T_1, M) \\
\hline
T_i & DES(T_i, M) \\
\hline
T_N & DES(T_N, M) \\
\hline
\end{array}
\]

Table $L$

\[
\begin{array}{|c|c|}
\hline
DES^{-1}(T_1, C) & T_1 \\
\hline
DES^{-1}(T_j, C) & T_j \\
\hline
DES^{-1}(T_N, C) & T_N \\
\hline
\end{array}
\]

Table $R$

Attack idea:

- Build L,R tables
Meet-in-the-middle attack on 2DES

Suppose $\text{DES}_K^{-1}(C) = \text{DES}_K(M)$ and $T_1, \ldots, T_N$ are all possible DES keys, where $N = 2^{56}$.

$$
\begin{array}{|c|c|}
\hline
K_1 & \rightarrow \\
\hline
T_1 & \text{DES}(T_1, M) \\
\hline
T_i & \text{DES}(T_i, M) \\
\hline
T_N & \text{DES}(T_N, M) \\
\hline
\end{array}
$$

Table $L$

$$
\begin{array}{|c|c|}
\hline
\text{DES}^{-1}(T_1, C) & T_1 \\
\hline
\text{DES}^{-1}(T_j, C) & T_j \\
\hline
\text{DES}^{-1}(T_N, C) & T_N \\
\hline
\end{array}
$$

Table $R$

Attack idea:

- Build $L,R$ tables
- Find $i,j$ s.t. $L[i] = R[j]$
- Guess that $K_1K_2 = T_iT_j$
Meet-in-the-middle attack on 2DES

Let $T_1, \ldots, T_{2^{56}}$ denote an enumeration of DES keys.

 adversary $A_{\text{MinM}}$

\[
M_1 \leftarrow 0^{64}; \quad C_1 \leftarrow Fn(M_1); \quad \text{for } i = 1, \ldots, 2^{56} \text{ do } L[i] \leftarrow \text{DES}(T_i, M_1) \quad \text{for } j = 1, \ldots, 2^{56} \text{ do } R[j] \leftarrow \text{DES}^{-1}(T_j, C_1) \\
S \leftarrow \{(i, j) : L[i] = R[j]\} \quad \text{Pick some } (l, r) \in S \text{ and return } T_l \parallel T_r
\]

This uses $q = 1$ plaintext-ciphertext pair and is unlikely to return the target key. For that one should extend the attack to a larger value of $q$. 

Nadia Heninger

UCSD
Running time of Meet-in-the-middle attack

**adversary** $A_{\text{MinM}}$

$M_1 \leftarrow 0^{64}; \quad C_1 \leftarrow \text{Fn}(M_1)$

for $i = 1, \ldots, 2^{56}$ do $L[i] \leftarrow \text{DES}(T_i, M_1)$

for $j = 1, \ldots, 2^{56}$ do $R[j] \leftarrow \text{DES}^{-1}(T_j, C_1)$

$S \leftarrow \{ (i, j) : L[i] = R[j] \}$

Pick some $(l, r) \in S$ and return $T_l \parallel T_r$

Let $T_{\text{DES}}$ be the time to compute $\text{DES}$ or $\text{DES}^{-1}$.

Let $k = 56$ be the key length. Let $\ell = 64$ be the block length.

Each “for” loop takes $O(2^k \cdot T_{\text{DES}})$ time.

To create $S$, we can sort the tables and then compare entries. Recall that sorting a size $N$ list takes $O(N \log(N))$ comparisons. So the time for this step is $O(k\ell \cdot 2^k)$. Why? $N = 2^k$, and comparison is $O(\ell)$. 

Nadia Heninger  
UCSD  
24
Running time of Meet-in-the-middle attack

**adversary** $A_{\text{MinM}}$

$M_1 \leftarrow 0^{64}; C_1 \leftarrow \text{Fn}(M_1)$

for $i = 1, \ldots, 2^{56}$ do $L[i] \leftarrow \text{DES}(T_i, M_1)$

for $j = 1, \ldots, 2^{56}$ do $R[j] \leftarrow \text{DES}^{-1}(T_j, C_1)$

$S \leftarrow \{ (i, j) : L[i] = R[j] \}$

Pick some $(l, r) \in S$ and return $T_l \parallel T_r$

Let $T_{\text{DES}}$ be the time to compute DES or DES$^{-1}$.

Let $k = 56$ be the key length. Let $\ell = 64$ be the block length.

Overall attack takes time $O(2^k \cdot (T_{\text{DES}} + k\ell))$.

In practice this should be around $2^{57}$ DES/DES$^{-1}$ operations, which is about the same as the cost of exhaustive key search on DES itself.
3DES

Block ciphers

3DES3 : \(\{0, 1\}^{168} \times \{0, 1\}^{64} \rightarrow \{0, 1\}^{64}\)

3DES2 : \(\{0, 1\}^{112} \times \{0, 1\}^{64} \rightarrow \{0, 1\}^{64}\)

are defined by

\[
3DES_3^{K_1 \parallel K_2 \parallel K_3}(M) = DES_{K_3}(DES^{-1}_{K_2}(DES_{K_1}(M)))
\]

\[
3DES_2^{K_1 \parallel K_2}(M) = DES_{K_2}(DES^{-1}_{K_1}(DES_{K_2}(M)))
\]

Meet-in-the-middle attack on 3DES3 reduces its “effective” key length to 112.
Later we will see “birthday” attacks that “break” a block cipher
$E : \{0, 1\}^k \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell$ in time $2^{\ell/2}$

For DES this is $2^{64/2} = 2^{32}$ which is small, and this is unchanged for 2DES and 3DES.

Would like a larger block size.
1998: NIST announces competition for a new block cipher

- key length 128
- block length 128
- faster than DES in software

Submissions from all over the world: MARS, Rijndael, Two-Fish, RC6, Serpent, Loki97, Cast-256, Frog, DFC, Magenta, E2, Crypton, HPC, Safer+, Deal
1998: NIST announces competition for a new block cipher

- key length 128
- block length 128
- faster than DES in software

Submissions from all over the world: MARS, Rijndael, Two-Fish, RC6, Serpent, Loki97, Cast-256, Frog, DFC, Magenta, E2, Crypton, HPC, Safer+, Deal

2001: NIST selects Rijndael to be AES.
function $\text{AES}_K(M)$

$\left(K_0, \ldots, K_{10}\right) \leftarrow \text{expand}(K)$

$s \leftarrow M \oplus K_0$

for $r = 1$ to $10$ do

\begin{align*}
    s & \leftarrow S(s) \\
    s & \leftarrow \text{shift-rows}(s) \\
    \text{if } r \leq 9 \text{ then } s & \leftarrow \text{mix-cols}(s) \text{ fi} \\
    s & \leftarrow s \oplus K_r
\end{align*}

end for

return $s$

- Fewer tables than DES
- Finite field operations
### Implementing AES

<table>
<thead>
<tr>
<th></th>
<th>Code size</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-compute and store round function tables</td>
<td>largest</td>
<td>fastest</td>
</tr>
<tr>
<td>Pre-compute and store S-boxes only</td>
<td>smaller</td>
<td>slower</td>
</tr>
<tr>
<td>No pre-computation</td>
<td>smallest</td>
<td>slowest</td>
</tr>
</tbody>
</table>

**AES-NI**: Hardware for AES, now present on most processors. Your laptop has it! Can run AES at around 1 cycle/byte. VERY fast!
Best known key-recovery attack [BoKhRe11] takes $2^{126.1}$ time, which is only marginally better than the $2^{128}$ time of EKS.

There are attacks on reduced-round versions of AES as well as on its sibling algorithms AES192, AES256. Many of these are “related-key” attacks. There are also effective side-channel attacks on AES such as “cache-timing” attacks [Be05,OsShTr05].
Limitations of security against key recovery

So far, a block cipher has been viewed as secure if it resists key recovery, meaning there is no efficient adversary $A$ having $\text{Adv}_E^{kr}(A) \approx 1$.

Is security against key recovery enough?

Not really. For example define $E: \{0, 1\}^{128} \times \{0, 1\}^{256} \rightarrow \{0, 1\}^{256}$ by


This is as secure against key-recovery as AES, but not a “good” blockcipher because half the message is in the clear in the ciphertext.
Possible reaction: But DES, AES are not designed like $E$ above, so why does this matter?

Answer: It tells us that security against key recovery is not, as a block-cipher property, sufficient for security of uses of the block cipher.

As designers and users we want to know what properties of a block cipher give us security when the block cipher is used.
So what is a “good” block cipher?

<table>
<thead>
<tr>
<th>Possible Properties</th>
<th>Necessary?</th>
<th>Sufficient?</th>
</tr>
</thead>
<tbody>
<tr>
<td>security against key recovery</td>
<td>YES</td>
<td>NO!</td>
</tr>
<tr>
<td>hard to find $M$ given $C = E_K(M)$</td>
<td>YES</td>
<td>NO!</td>
</tr>
</tbody>
</table>

We can’t define or understand security well via some such (indeterminable) list.

We want a single “master” property of a block cipher that is sufficient to ensure security of common usage of the block cipher.
Q: What does it mean for a program to be “intelligent” in the sense of a human?

Possible answers:

• It can be happy
• It recognizes pictures
• It can multiply
• But only small numbers!

Clearly, no such list is a satisfactory answer to the question.
Q: What does it mean for a program to be “intelligent” in the sense of a human?

Turing’s answer: A program is intelligent if its input/output behavior is indistinguishable from that of a human.
Behind the wall:

- **Room 1**: The program $P$
- **Room 0**: A human
Turing Intelligence Test

Game:
- Put tester in room 0 and let it interact with object behind wall
- Put tester in room 1 and let it interact with object behind wall
- Now ask tester: which room was which?

The measure of “intelligence” of $P$ is the extent to which the tester fails.
# Real versus Ideal

<table>
<thead>
<tr>
<th>Notion</th>
<th>Real object</th>
<th>Ideal object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligence</td>
<td>Program Block cipher</td>
<td>Human ?</td>
</tr>
<tr>
<td>PRF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Real versus Ideal

<table>
<thead>
<tr>
<th>Notion</th>
<th>Real object</th>
<th>Ideal object</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intelligence</td>
<td>Program</td>
<td>Human</td>
</tr>
<tr>
<td>PRF</td>
<td>Block cipher</td>
<td>Random function</td>
</tr>
</tbody>
</table>
Random functions

Game $\text{Rand}_R$  // here $R$ is a set

procedure $\text{Fn}(x)$
if $T[x] = \bot$ then $T[x] \leftarrow_R R$
return $T[x]$

Adversary $A$

• Make queries to $\text{Fn}$
• Eventually halts with some output

We denote by

$$\Pr \left[ \text{Rand}_R^A \Rightarrow d \right]$$

the probability that $A$ outputs $d$
Game $\text{Rand}_{\{0,1\}^3}$

procedure $\text{Fn}(x)$
if $T[x] = \bot$ then $T[x] \leftarrow \{0, 1\}^3$
return $T[x]$

adversary $A$
$y \leftarrow \text{Fn}(01)$
return $(y = 000)$

$$\Pr \left[ \text{Rand}^A_{\{0,1\}^3} \Rightarrow \text{true} \right] =$$
Game $\text{Rand}_{\{0,1\}^3}$

**procedure** $\text{Fn}(x)$

if $T[x] = \perp$ then $T[x] \leftarrow \{0,1\}^3$

return $T[x]$}

**adversary** $A$

$y \leftarrow \text{Fn}(01)$

return $(y = 000)$

$$\Pr\left[\text{Rand}^A_{\{0,1\}^3} \Rightarrow \text{true}\right] = 2^{-3}$$
Random function

Game $\text{Rand}_{\{0,1\}^3}$

**procedure** $\text{Fn}(x)$

if $T[x] = \bot$ then $T[x] \leftarrow \{0, 1\}^3$

return $T[x]$

adversary $A$

$y_1 \leftarrow \text{Fn}(00)$

$y_2 \leftarrow \text{Fn}(11)$

return $(y_1 = 010 \land y_2 = 011)$

$$
\Pr \left[ \text{Rand}_{\{0,1\}^3}^A \Rightarrow \text{true} \right] =
$$
Game $\text{Rand}_{\{0,1\}^3}$

procedure $\text{Fn}(x)$
if $T[x] = \bot$ then $T[x] \leftarrow \{0,1\}^3$
return $T[x]$

adversary $A$
$y_1 \leftarrow \text{Fn}(00)$
$y_2 \leftarrow \text{Fn}(11)$
return $(y_1 = 010 \land y_2 = 011)$

$$\Pr \left[ \text{Rand}^A_{\{0,1\}^3} \Rightarrow \text{true} \right] = 2^{-6}$$
Game $\text{Rand}_{\{0,1\}^3}$

**procedure** $F_n(x)$

if $T[x] = \bot$ then $T[x] \leftarrow \{0, 1\}^3$

return $T[x]$

adversary $A$

$y_1 \leftarrow F_n(00)$

$y_2 \leftarrow F_n(11)$

return $(y_1 \oplus y_2 = 101)$

$$\Pr \left[ \text{Rand}^A_{\{0,1\}^3} \Rightarrow \text{true} \right] =$$
Game \( \text{Rand}_{\{0,1\}^3} \)

**procedure** \( \text{Fn}(x) \)

if \( T[x] = \bot \) then \( T[x] \leftarrow \{0,1\}^3 \)

return \( T[x] \)

\[
\text{adversary } A \quad \\
y_1 \leftarrow \text{Fn}(00) \quad \\
y_2 \leftarrow \text{Fn}(11) \quad \\
\text{return } (y_1 \oplus y_2 = 101)
\]

\[
\Pr \left[ \text{Rand}_{\{0,1\}^3} \Rightarrow \text{true} \right] = 2^{-3}
\]
A family of functions (also called a function family) is a two-input function $F : \text{Keys} \times D \rightarrow R$. For $K \in \text{Keys}$ we let $F_K : D \rightarrow R$ be defined by $F_K(x) = F(K, x)$ for all $x \in D$.

**Examples:**

- **DES:** Keys = \{0, 1\}^{56}, D = R = \{0, 1\}^{64}
- Any block cipher: D = R and each $F_K$ is a permutation
### Real versus Ideal

<table>
<thead>
<tr>
<th>Notion</th>
<th>Real object</th>
<th>Ideal object</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRF</td>
<td>Family of functions (eg. a block cipher)</td>
<td>Random function</td>
</tr>
</tbody>
</table>

*F* is a PRF if the input-output behavior of $F_K$ looks to a tester like the input-output behavior of a random function.

Tester does **not** get the key $K$!
Games defining prf advantage of an adversary against $F$

Let $F: \text{Keys} \times D \rightarrow R$ be a family of functions.

<table>
<thead>
<tr>
<th>Game $\text{Real}_F$</th>
<th>Game $\text{Rand}_R$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>procedure</strong> Initialize</td>
<td><strong>procedure</strong> $\text{Fn}(x)$</td>
</tr>
<tr>
<td>$K \leftarrow \text{Keys}$</td>
<td>if $T[x] = \bot$ then $T[x] \leftarrow R$</td>
</tr>
<tr>
<td><strong>procedure</strong> $\text{Fn}(x)$</td>
<td>Return $T[x]$</td>
</tr>
<tr>
<td>Return $F_K(x)$</td>
<td></td>
</tr>
</tbody>
</table>

Associated to $F$, $A$ are the probabilities

$$\Pr \left[ \text{Real}_F^A \Rightarrow 1 \right] \quad \text{and} \quad \Pr \left[ \text{Rand}_R^A \Rightarrow 1 \right]$$

that $A$ outputs 1 in each world. The advantage of $A$ is

$$\text{Adv}_{F}^{\text{prf}}(A) = \Pr \left[ \text{Real}_F^A \Rightarrow 1 \right] - \Pr \left[ \text{Rand}_R^A \Rightarrow 1 \right]$$
PRF advantage

<table>
<thead>
<tr>
<th>$A$’s output $d$</th>
<th>Intended meaning: I think I am in game</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Real</td>
</tr>
<tr>
<td>0</td>
<td>Random</td>
</tr>
</tbody>
</table>

$\text{Adv}^\text{prf}_F(A) \approx 1$ means $A$ is doing well and $F$ is not prf-secure.

$\text{Adv}^\text{prf}_F(A) \approx 0$ (or $\leq 0$) means $A$ is doing poorly and $F$ resists the attack $A$ is mounting.
Adversary advantage depends on its

- strategy
- resources: Running time $t$ and number $q$ of oracle queries

**Security:** $F$ is a (secure) PRF if $\text{Adv}^\text{prf}_F(A)$ is “small” for ALL $A$ that use “practical” amounts of resources.

**Example:** 80-bit security could mean that for all $n = 1, \ldots, 80$ we have

$$\text{Adv}^\text{prf}_F(A) \leq 2^{-n}$$

for any $A$ with time and number of oracle queries at most $2^{80-n}$.

**Insecurity:** $F$ is insecure (not a PRF) if we can specify an $A$ using “few” resources that achieves “high” advantage.
Example

Define $F: \{0, 1\}^\ell \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell$ by $F_K(x) = K \oplus x$ for all $K, x \in \{0, 1\}^\ell$. Is $F$ a secure PRF?

<table>
<thead>
<tr>
<th>Game Real$_F$</th>
<th>Game Rand$_{{0,1}^\ell}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>procedure</strong> Initialize</td>
<td><strong>procedure</strong> Fn($x$)</td>
</tr>
<tr>
<td>$K \leftarrow {0, 1}^\ell$</td>
<td>if $T[x] = \bot$ then $T[x] \leftarrow {0, 1}^\ell$</td>
</tr>
<tr>
<td><strong>procedure</strong> Fn($x$)</td>
<td>Return $T[x]$</td>
</tr>
<tr>
<td>Return $K \oplus x$</td>
<td></td>
</tr>
</tbody>
</table>

So we are asking: Can we design a low-resource $A$ so that

$$\text{Adv}^\text{prf}_F (A) = \Pr \left[ \text{Real}^A_F \Rightarrow 1 \right] - \Pr \left[ \text{Rand}^A_{\{0,1\}^\ell} \Rightarrow 1 \right]$$

is close to 1?
Define $F$: $\{0, 1\}^\ell \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell$ by $F_K(x) = K \oplus x$ for all $K, x \in \{0, 1\}^\ell$. Is $F$ a secure PRF?

**Game $Real_F$**

**procedure** Initialize

$K \leftarrow \{0, 1\}^\ell$

**procedure** $Fn(x)$

Return $K \oplus x$

**Game $Rand_{\{0,1\}^\ell}$**

**procedure** $Fn(x)$

if $T[x] = \bot$ then $T[x] \leftarrow \{0, 1\}^\ell$

Return $T[x]$

So we are asking: Can we design a low-resource $A$ so that

$$Adv_F^{prf}(A) = \Pr \left[ Real^A_F \Rightarrow 1 \right] - \Pr \left[ Rand^A_{\{0,1\}^\ell} \Rightarrow 1 \right]$$

is close to 1?

Exploitable weakness of $F$: For all $K$ we have

$$F_K(0^\ell) \oplus F_K(1^\ell) = (K \oplus 0^\ell) \oplus (K \oplus 1^\ell) = 1^\ell$$
Example: The adversary

\( F: \{0, 1\}^\ell \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell \) is defined by \( F_K(x) = K \oplus x \).

adversary \( A \)

if \( F_n(0^\ell) \oplus F_n(1^\ell) = 1^\ell \) then return 1 else return 0
Example: Real game analysis

\[ F: \{0,1\}^\ell \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell \] is defined by \( F_K(x) = K \oplus x \).

adversary \( A \)

if \( F_n(0^\ell) \oplus F_n(1^\ell) = 1^\ell \) then return 1 else return 0
Example: Real game analysis

\[ F: \{0, 1\}^\ell \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell \text{ is defined by } F_K(x) = K \oplus x. \]

adversary \( A \)

if \( F_n(0^\ell) \oplus F_n(1^\ell) = 1^\ell \) then return 1 else return 0

\[
\begin{align*}
\text{Game Real}_F \\
\text{procedure } \text{Initialize} \\
K \leftarrow \{0, 1\}^\ell \\
\text{procedure } F_n(x) \\
\text{Return } K \oplus x
\end{align*}
\]

\[
\Pr \left[ \text{Real}_F^A \rightarrow 1 \right] = 1
\]

because

\[
F_n(0^\ell) \oplus F_n(1^\ell) = F_K(0^\ell) \oplus F_K(1^\ell) = (K \oplus 0^\ell) \oplus (K \oplus 1^\ell) = 1^\ell
\]
Example: Rand game analysis

\[ F: \{0, 1\}^\ell \times \{0, 1\}^\ell \to \{0, 1\}^\ell \text{ is defined by } F_K(x) = K \oplus x. \]

adversary \( A \)
if \( Fn(0^\ell) \oplus Fn(1^\ell) = 1^\ell \) then return 1 else return 0

\[
\text{Game } \text{Rand}_{\{0,1\}^\ell} \\
\text{procedure } Fn(x) \\
\text{if } T[x] = \perp \text{ then } T[x] \leftarrow \{0, 1\}^\ell \\
\text{Return } T[x]
\]

\[ \Pr \left[ \text{Rand}_{\{0,1\}^\ell}^A \Rightarrow 1 \right] = \]
Example: Rand game analysis

\( F: \{0,1\}^\ell \times \{0,1\}^\ell \rightarrow \{0,1\}^\ell \) is defined by \( F_K(x) = K \oplus x \).

**adversary** \( A \)

if \( F_n(0^\ell) \oplus F_n(1^\ell) = 1^\ell \) then return 1 else return 0

---

<table>
<thead>
<tr>
<th>Game Rand_{0,1}^\ell</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>procedure</strong> ( F_n(x) )</td>
</tr>
<tr>
<td>if ( T[x] = \bot ) then ( T[x] \leftarrow $ {0,1}^\ell )</td>
</tr>
<tr>
<td>Return ( T[x] )</td>
</tr>
</tbody>
</table>

\[
\Pr \left[ \text{Rand}_{0,1}^\ell \Rightarrow 1 \right] = \Pr \left[ F_n(1^\ell) \oplus F_n(0^\ell) = 1^\ell \right] =
\]
Example: Rand game analysis

\[ F: \{0, 1\}^\ell \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell \] is defined by \( F_K(x) = K \oplus x \).

Adversary \( A \)

if \( F_n(0^\ell) \oplus F_n(1^\ell) = 1^\ell \) then return 1 else return 0

Game \( \text{Rand}_{\{0,1\}^\ell} \)

procedure \( F_n(x) \)

if \( T[x] = \bot \) then \( T[x] \leftarrow \{0, 1\}^\ell \)

Return \( T[x] \)

\[ \Pr \left[ \text{Rand}_{\{0,1\}^\ell}^A \Rightarrow 1 \right] = \Pr \left[ F_n(1^\ell) \oplus F_n(0^\ell) = 1^\ell \right] = 2^{-\ell} \]

because \( F_n(0^\ell), F_n(1^\ell) \) are random \( \ell \)-bit strings.
$F$: $\{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^\ell$ is defined by $F_K(x) = K \oplus x$.

**adversary** $A$

if $F_n(0^\ell) \oplus F_n(1^\ell) = 1^\ell$ then return 1 else return 0

Then

$$\text{Adv}_{F}^{\text{prf}}(A) = \Pr[\text{Real}_{F}^A \Rightarrow 1] - \Pr[\text{Rand}_{\{0,1\}^\ell}^A \Rightarrow 1] = 1 - 2^{-\ell}$$

and $A$ is efficient.

**Conclusion:** $F$ is not a secure PRF.
Birthday Problem

We have $q$ people $1, \ldots, q$ with birthdays $y_1, \ldots, y_q \in \{1, \ldots, 365\}$. Assume each person’s birthday is a random day of the year. Let

$$C(365, q) = \Pr[\text{2 or more persons have same birthday}]$$

$$= \Pr[y_1, \ldots, y_q \text{ are not all different}]$$

• What is the value of $C(365, q)$?
• How large does $q$ have to be before $C(365, q)$ is at least $1/2$?
Birthday Problem

We have \( q \) people \( 1, \ldots, q \) with birthdays \( y_1, \ldots, y_q \in \{1, \ldots, 365\} \). Assume each person’s birthday is a random day of the year. Let

\[
C(365, q) = \Pr[2 \text{ or more persons have same birthday}] \\
= \Pr[y_1, \ldots, y_q \text{ are not all different}]
\]

• What is the value of \( C(365, q) \)?
• How large does \( q \) have to be before \( C(365, q) \) is at least 1/2?

Naive intuition:
• \( C(365, q) \approx q/365 \)
• \( q \) has to be around 365
Birthday Problem

We have \( q \) people \( 1, \ldots, q \) with birthdays \( y_1, \ldots, y_q \in \{1, \ldots, 365\} \). Assume each person’s birthday is a random day of the year. Let

\[
C(365, q) = \Pr \{\text{2 or more persons have same birthday}\} \\
= \Pr \{y_1, \ldots, y_q \text{ are not all different}\}
\]

• What is the value of \( C(365, q) \)?
• How large does \( q \) have to be before \( C(365, q) \) is at least 1/2?

Naive intuition:
• \( C(365, q) \approx q/365 \)
• \( q \) has to be around 365

The reality
• \( C(365, q) \approx q^2/365 \)
• \( q \) has to be only around 23
Birthday collision bounds

$C(365, q)$ is the probability that some two people have the same birthday in a room of $q$ people with random birthdays

<table>
<thead>
<tr>
<th>$q$</th>
<th>$C(365, q)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0.253</td>
</tr>
<tr>
<td>18</td>
<td>0.347</td>
</tr>
<tr>
<td>20</td>
<td>0.411</td>
</tr>
<tr>
<td>21</td>
<td>0.444</td>
</tr>
<tr>
<td>23</td>
<td>0.507</td>
</tr>
<tr>
<td>25</td>
<td>0.569</td>
</tr>
<tr>
<td>27</td>
<td>0.627</td>
</tr>
<tr>
<td>30</td>
<td>0.706</td>
</tr>
<tr>
<td>35</td>
<td>0.814</td>
</tr>
<tr>
<td>40</td>
<td>0.891</td>
</tr>
<tr>
<td>50</td>
<td>0.970</td>
</tr>
</tbody>
</table>
Pick $y_1, \ldots, y_q \leftarrow \{1, \ldots, N\}$ and let

$$C(N, q) = \Pr[y_1, \ldots, y_q \text{ not all distinct}]$$

Birthday setting: $N = 365$
Pick $y_1, \ldots, y_q \leftarrow \{1, \ldots, N\}$ and let

$$C(N, q) = \Pr [y_1, \ldots, y_q \text{ not all distinct}]$$

**Birthday setting:** $N = 365$

**Fact:** $C(N, q) \approx \frac{q^2}{2N}$
Birthday collisions formula

Let $y_1, \ldots, y_q \leftarrow \{1, \ldots, N\}$. Then

$$1 - C(N, q) = \Pr[y_1, \ldots, y_q \text{ all distinct}]$$

$$= 1 \cdot \frac{N - 1}{N} \cdot \frac{N - 2}{N} \cdots \frac{N - (q - 1)}{N}$$

$$= \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right)$$

So

$$C(N, q) = 1 - \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right)$$
Birthday bounds

Let

\[ C(N, q) = \Pr \left[ y_1, \ldots, y_q \text{ not all distinct} \right] \]

**Fact:** Then

\[ 0.3 \cdot \frac{q(q - 1)}{N} \leq C(N, q) \leq 0.5 \cdot \frac{q(q - 1)}{N} \]

where the lower bound holds for \( 1 \leq q \leq \sqrt{2N} \).
Let $E: \{0, 1\}^k \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell$ be a block cipher.

Can we design $A$ so that

$$\text{Adv}_{E}^{\text{prf}}(A) = \Pr[\text{Real}_E^A \Rightarrow 1] - \Pr[\text{Rand}_{\{0,1\}^\ell}^A \Rightarrow 1]$$

is close to 1?
Defining property of a block cipher: \( E_K \) is a permutation for every \( K \)

So if \( x_1, \ldots, x_q \) are distinct then

- \( \text{Fn} = E_K \Rightarrow \text{Fn}(x_1), \ldots, \text{Fn}(x_q) \) distinct
- \( \text{Fn} \) random \( \Rightarrow \text{Fn}(x_1), \ldots, \text{Fn}(x_q) \) not necessarily distinct

This leads to the following attack:

**adversary** \( A \)

Let \( x_1, \ldots, x_q \in \{0, 1\}^\ell \) be distinct

for \( i = 1, \ldots, q \) do \( y_i \leftarrow \text{Fn}(x_i) \)

if \( y_1, \ldots, y_q \) are all distinct then return 1
else return 0
Real world analysis

Let $E: \{0, 1\}^k \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell$ be a block cipher

<table>
<thead>
<tr>
<th>Game $\text{Real}_E$</th>
<th>adversary $A$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>procedure</strong> Initialize $K \leftarrow {0, 1}^k$</td>
<td>Let $x_1, \ldots, x_q \in {0, 1}^\ell$ be distinct</td>
</tr>
<tr>
<td><strong>procedure</strong> $\text{Fn}(x)$</td>
<td>for $i = 1, \ldots, q$ do $y_i \leftarrow \text{Fn}(x_i)$</td>
</tr>
<tr>
<td>Return $E_K(x)$</td>
<td>if $y_1, \ldots, y_q$ are all distinct</td>
</tr>
<tr>
<td></td>
<td>then return 1 else return 0</td>
</tr>
</tbody>
</table>

Then

$$\Pr \left[ \text{Real}_E^A \Rightarrow 1 \right] =$$
Let $E : \{0, 1\}^k \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell$ be a block cipher

<table>
<thead>
<tr>
<th>Game Real$_E$</th>
<th>adversary $A$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>procedure</strong> Initialize</td>
<td>Let $x_1, \ldots, x_q \in {0, 1}^\ell$ be distinct</td>
</tr>
<tr>
<td>$K \leftarrow {0, 1}^k$</td>
<td>for $i = 1, \ldots, q$ do $y_i \leftarrow \text{Fn}(x_i)$</td>
</tr>
<tr>
<td><strong>procedure</strong> Fn($x$)</td>
<td>if $y_1, \ldots, y_q$ are all distinct</td>
</tr>
<tr>
<td>Return $E_K(x)$</td>
<td>then return 1 else return 0</td>
</tr>
</tbody>
</table>

Then

$$\Pr \left[ \text{Real}^A_E \Rightarrow 1 \right] = 1$$

because $y_1, \ldots, y_q$ will be distinct because $E_K$ is a permutation.
Rand world analysis

Let $E : \{0, 1\}^K \times \{0, 1\}^\ell \to \{0, 1\}^\ell$ be a block cipher

**Game Rand_{\{0,1\}^\ell}**

**procedure** $\text{Fn}(x)$

if $T[x] = \bot$ then $T[x] \leftarrow \{0, 1\}^\ell$

Return $T[x]$

**adversary $A$**

Let $x_1, \ldots, x_q \in \{0, 1\}^\ell$ be distinct

for $i = 1, \ldots, q$ do $y_i \leftarrow \text{Fn}(x_i)$

if $y_1, \ldots, y_q$ are all distinct then return 1 else return 0

Then

$$\Pr \left[ \text{Rand}^A_{\{0,1\}^\ell} \Rightarrow 1 \right] = \Pr [y_1, \ldots, y_q \text{ all distinct}] = 1 - C(2^\ell, q)$$

because $y_1, \ldots, y_q$ are randomly chosen from $\{0, 1\}^\ell$. 
Birthday attack on a block cipher

\[ E : \{0, 1\}^k \times \{0, 1\}^\ell \to \{0, 1\}^\ell \] a block cipher

**adversary** \( A \)

Let \( x_1, \ldots, x_q \in \{0, 1\}^\ell \) be distinct

for \( i = 1, \ldots, q \) do \( y_i \leftarrow F_n(x_i) \)

if \( y_1, \ldots, y_q \) are all distinct then return 1 else return 0

\[
\text{Adv}_{E}^{\text{prf}} (A) = \Pr \left[ \text{Real}_E^A \Rightarrow 1 \right] - \Pr \left[ \text{Rand}_A^{\{0,1\}^\ell} \Rightarrow 1 \right] = C(2^\ell, q) \geq 0.3 \cdot \frac{q(q - 1)}{2^\ell}
\]

so

\[
q \approx 2^{\ell/2} \Rightarrow \text{Adv}_{E}^{\text{prf}} (A) \approx 1.
\]
Conclusion: If \( E : \{0, 1\}^k \times \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell \) is a block cipher, there is an attack on it as a PRF that succeeds in about \(2^{\ell/2}\) queries.

Depends on block length, not key length!

<table>
<thead>
<tr>
<th>( \ell )</th>
<th>(2^{\ell/2})</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>(2^{32})</td>
<td>Insecure</td>
</tr>
<tr>
<td>128</td>
<td>(2^{64})</td>
<td>Secure</td>
</tr>
</tbody>
</table>

\( \ell \) refers to the block length.
We have seen two possible metrics of security for a block cipher $E$

- **(T)KR-security**: It should be hard to find the target key, or a key consistent with input-output examples of a hidden target key.
- **PRF-security**: It should be hard to distinguish the input-output behavior of $E_K$ from that of a random function.

**Fact**: PRF-security of $E$ implies

- KR (and hence TKR) security of $E$
- Many other security attributes of $E$

This is a validation of the choice of PRF security as our main metric.
Our Assumptions

DES, AES are good block ciphers in the sense that they are PRF-secure up to the inherent limitations of the birthday attack and known key-recovery attacks.

You can assume this in designs and analyses.

But beware that the future may prove these assumptions wrong!