PSEUDO-RANDOM FUNCTIONS

Recall

We studied security of a block cipher against key recovery.

But we saw that security against key recovery is not sufficient to ensure that natural usages of a block cipher are secure.

We want to answer the question:

What is a good block cipher?

where "good" means that natural uses of the block cipher are secure.

We could try to define "good" by a list of necessary conditions:

- Key recovery is hard
- Recovery of M from $C = E_K(M)$ is hard
- ...

But this is neither necessarily correct nor appealing.

Turing Intelligence Test

Q: What does it mean for a program to be "intelligent" in the sense of a human?

Turing Intelligence Test

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!
-
-

Turing Intelligence Test

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!
-
-

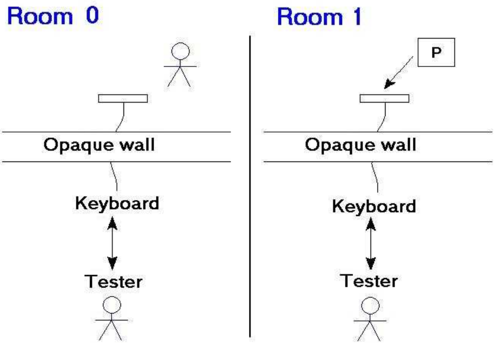
Clearly, no such list is a satisfactory answer to the question.

Turing Intelligence Test

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.

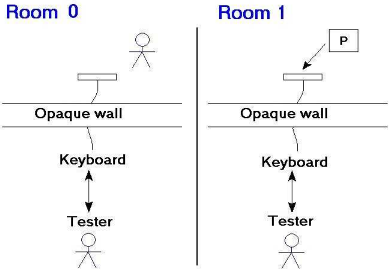
Turing Intelligence Test



Behind the wall:

- Room 1: The program P
- Room 0: A human

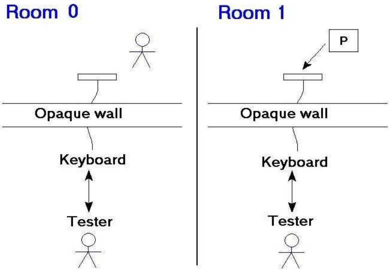
Turing Intelligence Test



Game:

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

Turing Intelligence Test

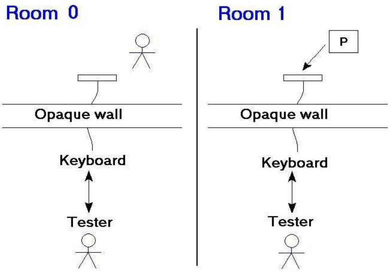


Game:

- Put tester in room 0 and let it interact with object behind wall
- Put tester in rooom 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.

Turing Intelligence Test



Game:

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

Clarification: Room numbers are in our head, not written on door!

Real versus Ideal

Notion	Real object	Ideal object
Intelligence	Program	Human
PRF	Block cipher	?

Real versus Ideal

Notion	Real object	Ideal object
Intelligence	Program	Human
PRF	Block cipher	Random function

Random functions

A random function with L-bit outputs is implemented by the following box **Fn**, where T is initially $\perp$ everywhere:

Fn

$$\text{Caller} \underset{\mathrm{T}[x]}{\overset{x}{\rightleftarrows}} \boxed{\begin{array}{l} \text{If } \mathrm{T}[x] = \perp \text{ then} \\ \quad \mathrm{T}[x] \xleftarrow{\$} \{0,1\}^L \\ \text{Return } \mathrm{T}[x] \end{array}}$$

Random function

Game $\mathrm{Rand}_{\{0,1\}^L}$

procedure Fn(x)

if $\mathsf{T}[x] = \perp$ then $\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^L$

return $\mathsf{T}[x]$

Adversary A

- Make queries to **Fn**
- Eventually halts with some output

We denote by

$$\Pr\left[\mathrm{Rand}^{A}_{\{0,1\}^l} \Rightarrow d\right]$$

the probability that A outputs d

Random function

Game $\mathrm{Rand}_{\{0,1\}^3}$
procedure Fn(x)
if $\mathsf{T}[x] = \perp$ then $\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^3$
return $\mathsf{T}[x]$

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

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

Random function

Game $\mathrm{Rand}_{\{0,1\}^3}$
procedure Fn(x)
if $\mathsf{T}[x] = \perp$ then $\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^3$
return $\mathsf{T}[x]$

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

$$\Pr\left[\mathrm{Rand}^{A}_{\{0,1\}^3} \Rightarrow \mathsf{true}\right] = 2^{-3}$$

Random function

Game $\mathrm{Rand}_{\{0,1\}^3}$
procedure Fn(x)
if $\mathsf{T}[x] = \perp$ then $\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^3$
return $\mathsf{T}[x]$

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

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

Random function

Game $\mathrm{Rand}_{\{0,1\}^3}$
procedure Fn(x)
if $\mathsf{T}[x] = \perp$ then $\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^3$
return $\mathsf{T}[x]$

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

$$\Pr\left[\mathrm{Rand}^{A}_{\{0,1\}^3} \Rightarrow \mathsf{true}\right] = 2^{-6}$$

Random function

Game $\mathrm{Rand}_{\{0,1\}^3}$
procedure Fn(x)
if $\mathsf{T}[x] = \perp$ then $\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^3$
return $\mathsf{T}[x]$

adversary A
$y_1 \leftarrow \mathbf{Fn}(00)$
$y_2 \leftarrow \mathbf{Fn}(11)$
return $(y_1 \oplus y_2 = 101)$

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

Random function

Game $\mathrm{Rand}_{\{0,1\}^3}$
procedure Fn(x)
if $\mathsf{T}[x] = \perp$ then $\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^3$
return $\mathsf{T}[x]$

adversary A
$y_1 \leftarrow \mathbf{Fn}(00)$
$y_2 \leftarrow \mathbf{Fn}(11)$
return $(y_1 \oplus y_2 = 101)$

$$\Pr\left[\mathrm{Rand}^{A}_{\{0,1\}^3} \Rightarrow \mathsf{true}\right] = 2^{-3}$$

Function families

A family of functions F : Keys(F) $\times$ Dom(F) $\to$ Range(F) is a two-argument map. For $K \in$ Keys(F) we let F_K : Dom(F) $\to$ Range(F) be defined by

$$\forall x \in \mathrm{Dom}(F) : F_K(x) = F(K, x)$$

Examples:

- DES: Keys(F) $= \{0, 1\}^{56}$, Dom(F) $=$ Range(F) $= \{0, 1\}^{64}$
- Any block cipher: Dom(F) $=$ Range(F) and each F_K is a permutation

Real versus Ideal

Notion	Real object	Ideal object
PRF	Family of functions (eg. a block cipher)	Random function

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.

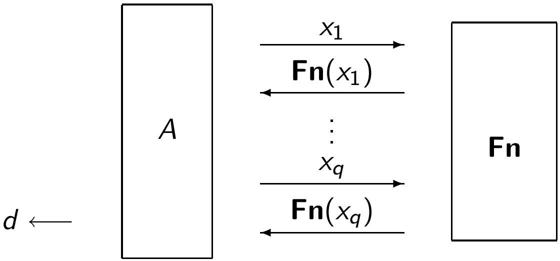
Tester does not get the key K!

PRF-adversaries

Let F: Keys$(F) \times$ Dom$(F) \rightarrow \mathrm{Range}(F)$ be a family of functions.

A prf-adversary (our tester) has an oracle **Fn** for a function from Dom(F) to Range(F). It can

- Make an oracle query x of its choice and get back **Fn**(x)
- Do this many times
- Eventually halt and output a bit d



Repeat queries

We said earlier that a random function must be consistent, meaning once it has returned y in response to x, it must return y again if queried again with the same x. This is why we have the "if" in the following: written as

Game $\mathrm{Rand}_{\mathsf{Range}(F)}$	**procedure Fn**(x) if $\mathsf{T}[x] \neq \perp$ then $\mathsf{T}[x] \xleftarrow{\$} \mathsf{Range}(F)$ Return $\mathsf{T}[x]$

Henceforth we make a rule:

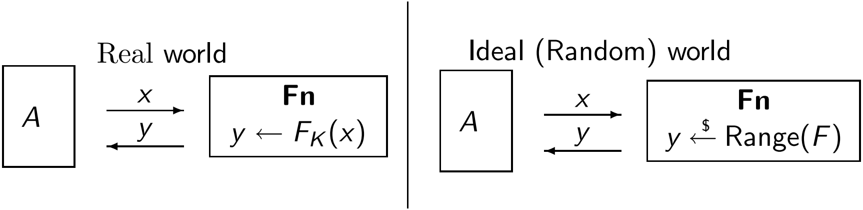
- A prf-adversary is not allowed to repeat an oracle query.

Then our game is:

Game $\mathrm{Rand}_{\mathsf{Range}(F)}$	**procedure Fn**(x) $\mathsf{T}[x] \xleftarrow{\$} \mathsf{Range}(F)$ Return $\mathsf{T}[x]$

PRF-adversaries

Let F: $\text{Keys}(F) \times \text{Dom}(F) \to \text{Range}(F)$ be a family of functions.



A's output d	Intended meaning: I think I am in the
1	Real world
0	Ideal (Random) world

The harder it is for A to guess world it is in, the "better" F is as a PRF.

The games

Let F: $\mathsf{Keys}(F) \times \mathsf{Dom}(F) \rightarrow \mathrm{Range}(F)$ be a family of functions.

Game Real_F
procedure Initialize
$K \xleftarrow{\$} \mathsf{Keys}(F)$
procedure Fn(x)
Return $F_K(x)$

Game $\mathrm{Rand}_{\mathsf{Range}(F)}$
procedure Fn(x)
$\mathsf{T}[x] \xleftarrow{\$} \mathsf{Range}(F)$
Return $\mathsf{T}[x]$

Associated to F, A are the probabilities

$$\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] \qquad \Big| \qquad \Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right]$$

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

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = \Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] - \Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right]$$

Example

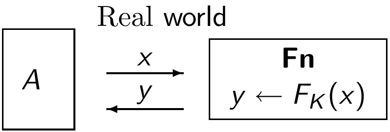
Let $F\colon \{0,1\}^k \times \{0,1\}^{128} \to \{0,1\}^{128}$ be defined by $F_K(x) = x$. Let prf-adversary A be defined by

adversary A
if $\mathbf{Fn}(0^{128}) = 0^{128}$ then Ret 1 else Ret 0

Game Real_F

procedure Initialize
$K \xleftarrow{\$} \{0,1\}^k$

procedure Fn(x)
Return $F_K(x)$



Example

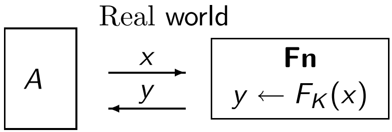
Let $F\colon \{0,1\}^k \times \{0,1\}^{128} \to \{0,1\}^{128}$ be defined by $F_K(x) = x$. Let prf-adversary A be defined by

adversary A
if $\mathbf{Fn}(0^{128}) = 0^{128}$ then Ret 1 else Ret 0

Game Real_F

procedure Initialize
$K \xleftarrow{\$} \{0,1\}^k$

procedure Fn(x)
Return $F_K(x)$



Then

$$\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] =$$

Example

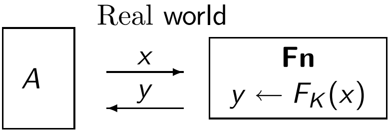
Let $F\colon \{0,1\}^k \times \{0,1\}^{128} \to \{0,1\}^{128}$ be defined by $F_K(x) = x$. Let prf-adversary A be defined by

adversary A
if $\mathbf{Fn}(0^{128}) = 0^{128}$ then Ret 1 else Ret 0

Game Real_F

procedure Initialize
$K \xleftarrow{\$} \{0,1\}^k$

procedure Fn(x)
Return $F_K(x)$



Then

$$\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] = 1$$

because the value returned by **Fn** will be $\mathbf{Fn}(0^{128}) = F_K(0^{128}) = 0^{128}$ so A will always return 1.

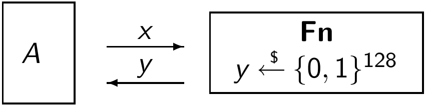
Example

Let $F\colon \{0,1\}^k \times \{0,1\}^{128} \to \{0,1\}^{128}$ be defined by $F_K(x) = x$. Let prf-adversary A be defined by

adversary A
if $\mathbf{Fn}(0^{128}) = 0^{128}$ then Ret 1 else Ret 0

Game $\mathrm{Rand}_{\mathsf{Range}(F)}$
procedure $\mathbf{Fn}(x)$
$\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^L$
Return $\mathsf{T}[x]$

Ideal (Random) world



Then

$$\Pr\left[\mathrm{Rand}^{A}_{\mathsf{Range}(F)} \Rightarrow 1\right] =$$

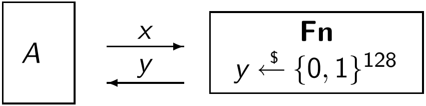
Example

Let $F\colon \{0,1\}^k \times \{0,1\}^{128} \to \{0,1\}^{128}$ be defined by $F_K(x) = x$. Let prf-adversary A be defined by

adversary A
if $\mathbf{Fn}(0^{128}) = 0^{128}$ then Ret 1 else Ret 0

Game $\mathrm{Rand}_{\mathsf{Range}(F)}$
procedure Fn(x)
$\mathrm{T}[x] \xleftarrow{\$} \{0,1\}^L$
Return $\mathrm{T}[x]$

Ideal (Random) world



Then

$$\Pr\left[\mathrm{Rand}^{A}_{\mathsf{Range}(F)} \Rightarrow 1\right] = \Pr\left[\mathbf{Fn}(0^{128}) = 0^{128}\right] = 2^{-128}$$

because $\mathbf{Fn}(0^{128})$ is a random 128-bit string.

Example: Advantage computation.

Let F: $\{0,1\}^k \times \{0,1\}^{128} \to \{0,1\}^{128}$ be defined by $F_K(x) = x$. Let prf-adversary A be defined by

adversary A
if $\mathbf{Fn}(0^{128}) = 0^{128}$ then Ret 1 else Ret 0

Then

$$\begin{aligned}\mathbf{Adv}_F^{\mathrm{prf}}(A) &= \overbrace{\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right]}^{1} - \overbrace{\Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right]}^{2^{-128}} \\ &= 1 - 2^{-128}\end{aligned}$$

The measure of success

Let $F\colon \text{Keys}(F) \times \mathrm{Domain}(F) \to \text{Range}(F)$ be a family of functions and A a prf adversary. Then

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

is a number between -1 and 1.

A "large" (close to 1) advantage means

- A is doing well
- F is not secure

A "small" (close to 0 or ≤ 0) advantage means

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

PRF security

Adversary advantage depends on its

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

Security: F is a (secure) PRF if $\mathbf{Adv}_F^{\mathrm{prf}}(A)$ 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

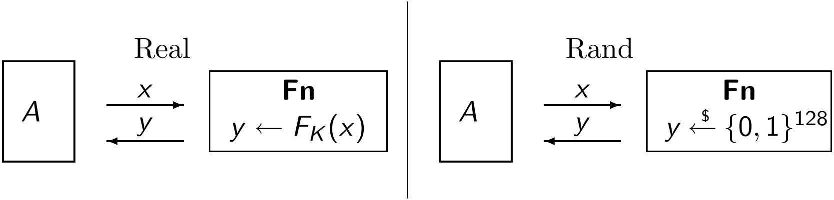
$$\mathbf{Adv}_F^{\mathrm{prf}}(A) \leq 2^{-n}$$

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

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

Example 1

Define $F : \{0,1\}^k \times \{0,1\}^{128} \to \{0,1\}^{128}$ by $F_K(x) = x$ for all k, x.
Is F a secure PRF?



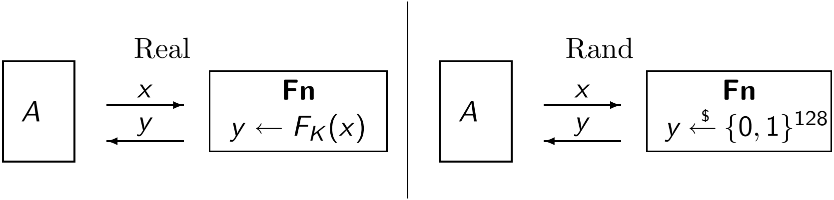
Can we design A so that

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = \Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] - \Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right]$$

is close to 1?

Example 1

Define $F : \{0,1\}^k \times \{0,1\}^{128} \to \{0,1\}^{128}$ by $F_K(x) = x$ for all k, x.
Is F a secure PRF?



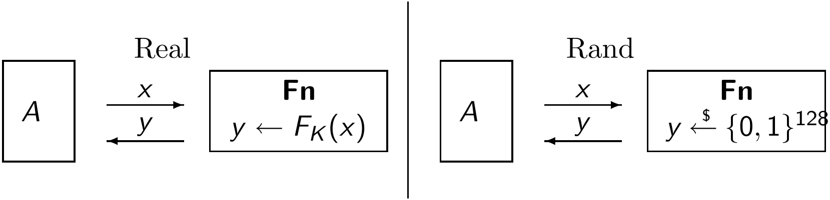
Can we design A so that

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = \Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] - \Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right]$$

is close to 1?

Exploitable weakness of F: $F_K(0^{128}) = 0^{128}$ for all K. We can determine which world we are in by testing whether $\mathbf{Fn}(0^{128}) = 0^{128}$.

Example 1

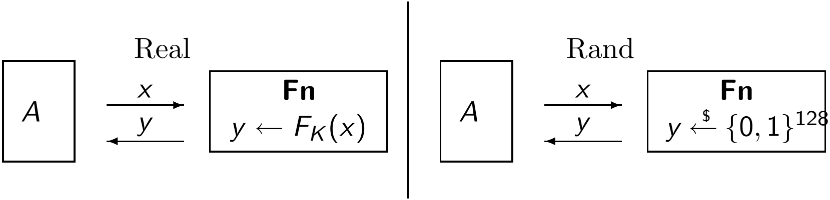


Now F is defined by $F_K(x) = x$.

adversary A
if $\mathbf{Fn}(0^{128}) = 0^{128}$ then return 1 else return 0

F is defined by $F_K(x) = x$.

adversary A
if $\mathbf{Fn}(0^{128}) = 0^{128}$ then return 1 else return 0



We already analysed this and saw that

$$\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] = 1 \qquad \Big| \qquad \Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right] = 2^{-128}$$

Example 1: Conclusion

F is defined by $F_K(x) = x$.

adversary A
if $\mathbf{Fn}(0^{128}) = 0^{128}$ then return 1 else return 0

Then

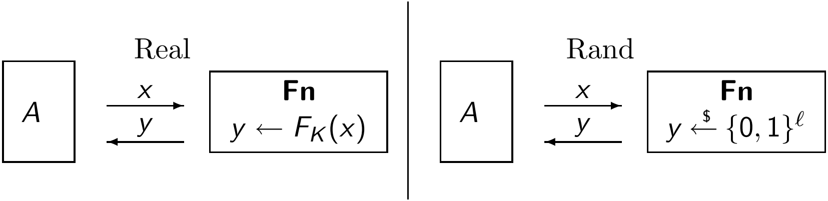
$$
\begin{aligned}
\mathbf{Adv}_F^{\mathrm{prf}}(A) &= \overbrace{\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right]}^{1} - \overbrace{\Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right]}^{2^{-128}} \\
&= 1 - 2^{-128}
\end{aligned}
$$

and A is efficient.

Conclusion: F is not a secure PRF.

Example 2

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



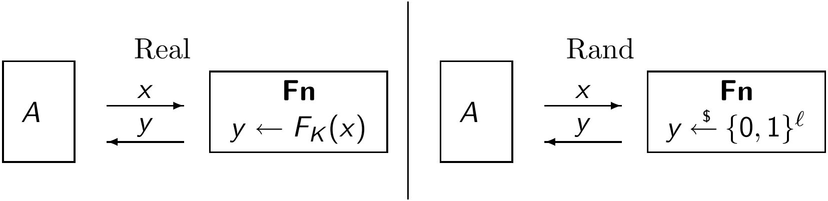
Can we design A so that

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = \Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] - \Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right]$$

is close to 1?

Example 2

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



Can we design A so that

$$\mathbf{Adv}_F^{\mathrm{prf}}(A) = \Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] - \Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right]$$

is close to 1?

Exploitable weakness of F:

$$F_K(0^\ell) \oplus F_K(1^\ell) = (K \oplus 0^\ell) \oplus (K \oplus 1^\ell) = 1^\ell$$

for all K. We can determine which world we are in by testing whether

$$\mathbf{Fn}(0^\ell) \oplus \mathbf{Fn}(1^\ell) = 1^\ell.$$

Example 2: The adversary

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

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

Example 2: Real world analysis

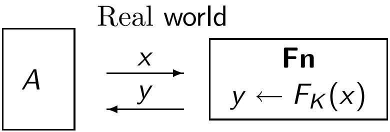
$F\colon \{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^\ell$ is defined by $F_K(x) = K \oplus x$.

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

Game Real_F

procedure Initialize
$K \stackrel{\$}{\leftarrow} \{0,1\}^k$

procedure Fn(x)
Return $F_K(x)$



Example 2: Real world analysis

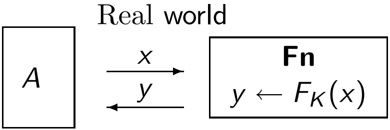
F: $\{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^\ell$ is defined by $F_K(x) = K \oplus x$.

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

Game Real_F

procedure Initialize
$K \xleftarrow{\$} \{0,1\}^k$

procedure Fn(x)
Return $F_K(x)$



Then

$$\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] =$$

Example 2: Real world analysis

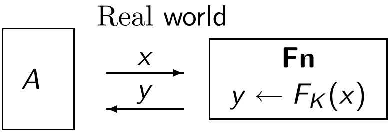
F: $\{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}^\ell$ is defined by $F_K(x) = K \oplus x$.

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

Game Real_F

procedure Initialize
$K \xleftarrow{\$} \{0,1\}^k$

procedure Fn(x)
Return $F_K(x)$



Then

$$\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] = 1$$

because

$$\mathbf{Fn}(0^\ell) \oplus \mathbf{Fn}(1^\ell) = F_K(0^\ell) \oplus F_K(1^\ell) \quad = \quad (K \oplus 0^\ell) \oplus (K \oplus 1^\ell) = 1^\ell$$

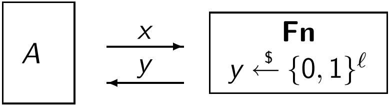
Example 2: Ideal world analysis

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

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

Game $\mathrm{Rand}_{\mathsf{Range}(F)}$
procedure Fn(x)
$\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^\ell$ return $\mathsf{T}[x]$

Ideal (random) world



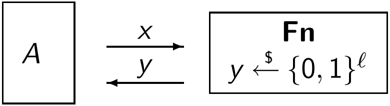
Example 2: Ideal world analysis

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

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

Game $\mathrm{Rand}_{\mathsf{Range}(F)}$
procedure Fn(x)
$\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^\ell$ return $\mathsf{T}[x]$

Ideal (random) world



Then

$$\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] =$$

Example 2: Ideal world analysis

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

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

Game $\mathrm{Rand}_{\mathsf{Range}(F)}$
procedure Fn(x)
$\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^\ell$ return $\mathsf{T}[x]$

Ideal (random) world

$A \xrightarrow{x}$ **Fn**: $y \xleftarrow{\$} \{0,1\}^\ell$ $\xrightarrow{y} A$

Then
$$\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] = \Pr\left[\mathbf{Fn}(1^\ell) \oplus \mathbf{Fn}(0^\ell) = 1^\ell\right] =$$

Example 2: Ideal world analysis

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

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

Game $\mathrm{Rand}_{\mathsf{Range}(F)}$
procedure $\mathbf{Fn}(x)$
$\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^\ell$ return $\mathsf{T}[x]$

Ideal (random) world

$$\boxed{A} \quad \underset{\xleftarrow{\quad y \quad}}{\xrightarrow{\quad x \quad}} \quad \boxed{\begin{array}{c}\mathbf{Fn} \\ y \xleftarrow{\$} \{0,1\}^\ell\end{array}}$$

Then

$$\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] = \Pr\left[\mathbf{Fn}(1^\ell) \oplus \mathbf{Fn}(0^\ell) = 1^\ell\right] = 2^{-\ell}$$

because $\mathbf{Fn}(0^\ell), \mathbf{Fn}(1^\ell)$ are random ℓ-bit strings.

Example 2: Conclusion

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

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

Then

$$\begin{aligned} \mathbf{Adv}_F^{\mathrm{prf}}(A) &= \overbrace{\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right]}^{1} - \overbrace{\Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right]}^{2^{-\ell}} \\ &= 1 - 2^{-\ell} \end{aligned}$$

and A is efficient .

Conclusion: F is not a secure PRF.

Birthday Problem

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

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

- 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

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

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

- 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

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

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

- 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

q	$C(365, q)$
15	0.253
18	0.347
20	0.411
21	0.444
23	0.507
25	0.569
27	0.627
30	0.706
35	0.814
40	0.891
50	0.970

Birthday Problem

Pick $y_1, \ldots, y_q \xleftarrow{\$} \{1, \ldots, N\}$ and let

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

Birthday setting: $N = 365$

Birthday Problem

Pick $y_1, \ldots, y_q \xleftarrow{\$} \{1, \ldots, N\}$ and let

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

Birthday setting: $N = 365$

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

Birthday collisions formula

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

$$
\begin{aligned}
1 - C(N,q) &= \Pr\left[y_1, \ldots, y_q \text{ all distinct}\right] \\
&= 1 \cdot \frac{N-1}{N} \cdot \frac{N-2}{N} \cdot \cdots \cdot \frac{N-(q-1)}{N} \\
&= \prod_{i=1}^{q-1} \left(1 - \frac{i}{N}\right)
\end{aligned}
$$

so

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

Birthday bounds

Let

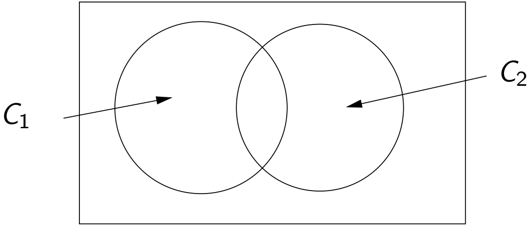
$$C(N, q) = \Pr\left[y_1, \ldots, y_q \textbf{ not} \text{ all distinct}\right]$$

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 $1 \leq q \leq \sqrt{2N}$.

Union bound



$$
\begin{aligned}
\Pr[C_1 \vee C_2] &= \Pr[C_1] + \Pr[C_2] - \Pr[C_1 \wedge C_2] \\
&\leq \Pr[C_1] + \Pr[C_2]
\end{aligned}
$$

More generally

$$\Pr[C_1 \vee C_2 \vee \cdots \vee C_q] \leq \Pr[C_1] + \Pr[C_2] + \cdots \Pr[C_q]$$

Arithmetic sums

$$0 + 1 + 2 + \cdots + (q - 1) =$$

Arithmetic sums

$$0 + 1 + 2 + \cdots + (q-1) = \frac{q(q-1)}{2}$$

Birthday bounds

Let

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

Then

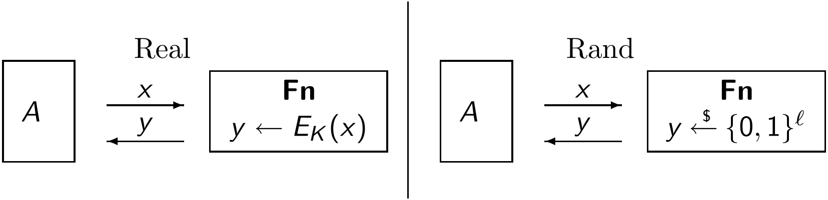
$$C(N, q) \leq 0.5 \cdot \frac{q(q-1)}{N}$$

Proof of this upper bound: Let C_i be the event that $y_i \in \{y_1, \ldots, y_{i-1}\}$. Then

$$\begin{aligned} C(N, q) &= \Pr[C_1 \vee C_2, \ldots, \vee C_q] \\ &\leq \Pr[C_1] + \Pr[C_2] + \ldots + \Pr[C_q] \\ &\leq \frac{0}{N} + \frac{1}{N} + \ldots + \frac{q-1}{N} \\ &= \frac{q(q-1)}{2N}. \end{aligned}$$

Block ciphers as PRFs

Let $E\colon \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$ be a block cipher.



Can we design A so that

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

is close to 1?

Block ciphers as PRFs

Defining property of a block cipher: E_K is a permutation for every K

So if $x_1, \ldots, x_q$ are distinct then

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

Let us turn this into an attack.

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 $y_i \leftarrow \mathbf{Fn}(x_i)$
if $y_1, \ldots, y_q$ are all distinct then return 1
else return 0

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

Game Real_E
procedure Initialize
$K \xleftarrow{\$} \{0,1\}^k$
procedure Fn(x)
Return $E_K(x)$

adversary A

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

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

if $y_1, \ldots, y_q$ are all distinct

then return 1 else return 0

Then

$$\Pr\left[\mathrm{Real}_E^A \Rightarrow 1\right] =$$

Real world analysis

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

Game Real_E
procedure Initialize
$K \xleftarrow{\$} \{0,1\}^k$
procedure Fn(x)
Return $E_K(x)$

adversary A
Let $x_1, \ldots, x_q \in \{0,1\}^\ell$ be distinct
for $i = 1, \ldots, q$ do $y_i \leftarrow \mathbf{Fn}(x_i)$
if $y_1, \ldots, y_q$ are all distinct
then return 1 else return 0

Then

$$\Pr\left[\mathrm{Real}_E^A \Rightarrow 1\right] = 1$$

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

Ideal world analysis

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

Game $\mathrm{Rand}_{\{0,1\}^\ell}$
procedure Fn(x)
$\mathsf{T}[x] \xleftarrow{\$} \{0,1\}^\ell$
Return $\mathsf{T}[x]$

adversary A
Let $x_1, \ldots, x_q \in \{0,1\}^\ell$ be distinct
for $i = 1, \ldots, q$ do $y_i \leftarrow \mathbf{Fn}(x_i)$
if $y_1, \ldots, y_q$ are all distinct
then return 1 else return 0

Then

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

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 $y_i \leftarrow \mathbf{Fn}(x_i)$
if $y_1, \ldots, y_q$ are all distinct then return 1 else return 0

$$\begin{aligned}
\mathbf{Adv}_E^{\mathrm{prf}}(A) &= \overbrace{\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right]}^{1} - \overbrace{\Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right]}^{1 - C(2^\ell, q)} \\
&= C(2^\ell, q) \\
&\geq 0.3 \cdot \frac{q(q-1)}{2^\ell}
\end{aligned}$$

so

$$q \approx 2^{\ell/2} \Rightarrow \mathbf{Adv}_E^{\mathrm{prf}}(A) \approx 1 \,.$$

Birthday attack on a block cipher

Conclusion: If $E : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$ 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!

	ℓ	$2^{\ell/2}$	Status
DES, 2DES, 3DES3	64	2^{32}	Insecure
AES	128	2^{64}	Secure

KR-security versus PRF-security

We have seen two possible metrics of security for a block cipher E

- KR-security: It should be hard to get K from input-output examples of E_K
- PRF-security: It should be hard to distinguish the input-output behavior of E_K from that of a random function.

Question: Is it possible for E to be

- PRF-secure, but
- NOT KR-secure?

KR-security versus PRF-security

Question: Is it possible for a block cipher E to be PRF-secure but not KR-secure?

Why do we care? Because we

- agreed that KR-security is necessary
- claim that PRF-security is sufficient

for secure use of E, so a YES answer would render our claim false.

Luckily the answer to the above question is NO.

KR-security versus PRF-security

Fact: PRF-security implies

- KR-security
- Many other security attributes

Key recovery security, formally

Let F : Keys$(F) \times \mathrm{Domain}(F) \rightarrow$ Range(F) a family of functions

Let B be an adversary

Game KR_F	**procedure Fn**(x)
	$\mathrm{return}\ F_K(x)$
procedure Initialize	**procedure Finalize**(K')
$K \xleftarrow{\$}$ Keys(F)	$\mathrm{return}\ (K = K')$

The *kr-advantage* of B is defined as

$$\mathbf{Adv}_F^{\mathrm{kr}}(B) \quad = \quad \Pr\left[\mathrm{KR}_F^B \Rightarrow \mathsf{true}\right]$$

The oracle allows a chosen message attack.

F is secure against key recovery if $\mathbf{Adv}_F^{\mathrm{kr}}(B)$ is "small" for all B of "practical" resources.

Example

Let $k = L\ell$ and define $F = \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L$ by

$$F_K(X) = \begin{bmatrix} K[1,1] & K[1,2] & \cdots & K[1,\ell] \\ K[2,1] & K[2,2] & \cdots & K[2,\ell] \\ \vdots & & & \vdots \\ K[L,1] & K[L,2] & \cdots & K[L,\ell] \end{bmatrix} \cdot \begin{bmatrix} X[1] \\ X[2] \\ \vdots \\ X[\ell] \end{bmatrix} = \begin{bmatrix} Y[1] \\ Y[2] \\ \vdots \\ Y[L] \end{bmatrix}$$

Here the bits in the matrix are the bits in the key, and arithmetic is modulo two.

Question: Is F secure against key-recovery?

Example

Let $k = L\ell$ and define $F = \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L$ by

$$F_K(X) = \begin{bmatrix} K[1,1] & K[1,2] & \cdots & K[1,\ell] \\ K[2,1] & K[2,2] & \cdots & K[2,\ell] \\ \vdots & & & \vdots \\ K[L,1] & K[L,2] & \cdots & K[L,\ell] \end{bmatrix} \cdot \begin{bmatrix} X[1] \\ X[2] \\ \vdots \\ X[\ell] \end{bmatrix} = \begin{bmatrix} Y[1] \\ Y[2] \\ \vdots \\ Y[L] \end{bmatrix}$$

Here the bits in the matrix are the bits in the key, and arithmetic is modulo two.

Question: Is F secure against key-recovery?

Answer: NO

Example

For $1 \leq i \leq \ell$ let:

$$e_j = \begin{bmatrix} \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} & j-1 \\ 1 & \\ \left.\begin{matrix} 0 \\ \vdots \\ 0 \end{matrix}\right\} & \ell - j \end{bmatrix}$$

be the j-th unit vector.

$$F_K(e_j) = \begin{bmatrix} K[1,1] & K[1,2] & \cdots & K[1,\ell] \\ K[2,1] & K[2,2] & \cdots & K[2,\ell] \\ \vdots & & & \vdots \\ K[L,1] & K[L,2] & \cdots & K[L,\ell] \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} K[1,j] \\ K[2,j] \\ \vdots \\ K[L,j] \end{bmatrix}$$

KR attack on example

Adversary B

$K' \leftarrow \varepsilon$ // ε is the empty string

for $j = 1, \ldots, \ell$ do $y_j \leftarrow \mathbf{Fn}(e_j)$; $K' \leftarrow K' \| y_j$

return K'

Then

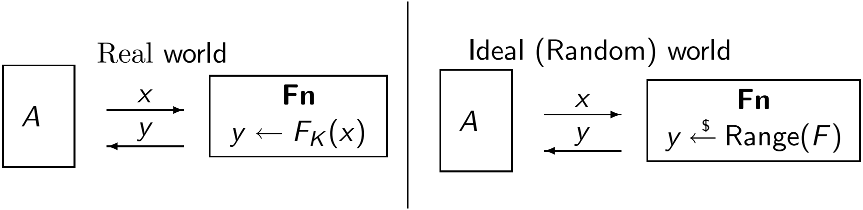
$$\mathbf{Adv}_F^{\mathrm{kr}}(B) = 1 .$$

The time-complexity of B is $t = O(\ell^2 L)$ since it makes $q = \ell$ calls to its oracle and each computation of $\mathbf{Fn} = F_K$ takes $O(\ell L)$ time.

So F is insecure against key-recovery.

Why does PRF-security imply KR-security?

Claim: KR-insecurity $\Rightarrow$ PRF-insecurity



If you give me a method B to defeat KR-security I can design a method A to defeat PRF-security.

What A does:

- Use B to find key K'
- Test whether $\mathbf{Fn}(x) = F_{K'}(x)$ for some new point x
- If this is true, decide it is in the Real world

Why does PRF-security imply KR-security?

Issues: To run B, adversary A must give it input-output examples under F_K.

We have A give B input-output examples under **Fn**. This is correct in the real world but not in the random world. Nonetheless we can show it works.

If F is a PRF then it is KR-secure

Our first example of a proof by reduction!

Given: $F : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L$
Given: efficient KR-adversary B
Construct: efficient PRF-adversary A such that:

$$\mathbf{Adv}_F^{\mathrm{kr}}(B) \le \mathbf{Adv}_F^{\mathrm{prf}}(A) + \boxed{\cdot}$$

How to infer that PRF-secure $\Rightarrow$ KR-secure:

$$\begin{aligned} F \text{ is PRF secure} \quad &\Rightarrow \mathbf{Adv}_F^{\mathrm{prf}}(A) \text{ is small} \\ &\Rightarrow \mathbf{Adv}_F^{\mathrm{kr}}(B) \text{ is small} \\ &\Rightarrow F \text{ is KR-secure} \end{aligned}$$

If F is a PRF then it is KR-secure

Our first example of a proof by reduction!

Given: $F : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L$
Given: efficient KR-adversary B
Construct: efficient PRF-adversary A such that:

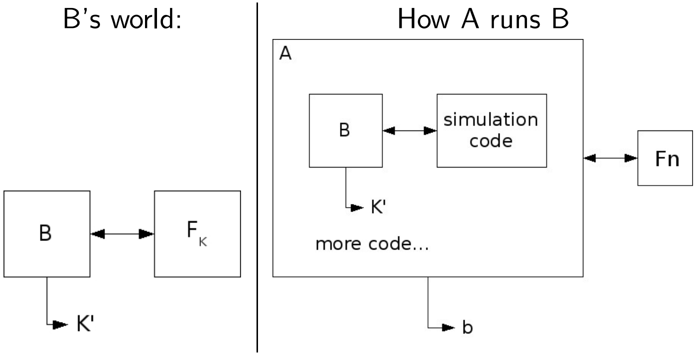
$$\mathbf{Adv}_F^{\mathrm{kr}}(B) \leq \mathbf{Adv}_F^{\mathrm{prf}}(A) + \boxed{\cdot}$$

Contrapositive:

$$\begin{aligned} F \text{ not KR-secure} \quad & \Rightarrow \mathbf{Adv}_F^{\mathrm{kr}}(B) \text{ is big} \\ & \Rightarrow \mathbf{Adv}_F^{\mathrm{prf}}(A) \text{ is big} \\ & \Rightarrow \text{F is not PRF-secure} \end{aligned}$$

How reductions work

A will run B as a subroutine



A itself answers B's oracle queries, giving B the impression that B is in its own correct world.

If F is a PRF then it is KR-secure

Given: $F : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L$
Given: efficient KR-adversary B
Construct: efficient PRF-adversary A such that:

$$\mathbf{Adv}_F^{\mathrm{kr}}(B) \le \mathbf{Adv}_F^{\mathrm{prf}}(A) + \boxdot$$

Idea:

- A uses B to find key K'
- Tests whether K' is the right key

Issues:

- B needs an F_K oracle, which A only has in the real world
- How to test K'?

How they are addressed:

- A gives B its **Fn** oracle
- Test by seeing whether $F_{K'}$ agrees with **Fn** on a new point.

If F is a PRF then it is KR-secure

Given: $F : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L$
Given: efficient KR-adversary B
Construct: efficient PRF-adversary A such that:

$$\mathbf{Adv}_F^{\mathrm{kr}}(B) \le \mathbf{Adv}_F^{\mathrm{prf}}(A) + \boxdot$$

adversary A
$i \leftarrow 0$
$K' \leftarrow B^{\mathrm{FnKRSim}}$
$x \xleftarrow{\$} \{0,1\}^\ell - \{x_1, \ldots, x_i\}$
if $F_{K'}(x) = \mathbf{Fn}(x)$ then return 1
else return 0

subroutine $\mathrm{FnKRSim}(x)$
$i \leftarrow i + 1$
$x_i \leftarrow x$
$y_i \leftarrow \mathbf{Fn}(x)$
return y_i

Analysis

adversary A
$i \leftarrow 0$
$K' \leftarrow B^{\mathrm{FnKRSim}}$
$x \xleftarrow{\$} \{0,1\}^{\ell} - \{x_1, \ldots, x_i\}$
if $F_{K'}(x) = \mathbf{Fn}(x)$ then return 1
else return 0

subroutine $\mathrm{FnKRSim}(x)$
$i \leftarrow i + 1$
$x_i \leftarrow x$
$y_i \leftarrow \mathbf{Fn}(x)$
return y_i

- If $\mathbf{Fn} = F_K$ then $K' = K$ with probability the KR-advantage of B, so
$$\Pr\left[\mathrm{Real}_F^A \Rightarrow 1\right] \geq \mathbf{Adv}_F^{\mathrm{kr}}(B)$$
- If $\mathbf{Fn}$ is a random function, then due to the fact that $x \notin \{x_1, \ldots, x_i\}$,
$$\Pr\left[\mathrm{Rand}_{\mathsf{Range}(F)}^A \Rightarrow 1\right] = 2^{-L}$$

So $\mathbf{Adv}_F^{\mathrm{prf}}(A) \geq \mathbf{Adv}_F^{\mathrm{kr}}(B) - 2^{-L}$

If F is PRF-secure then it is KR-secure

Proposition: Let $F : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L$ be a family of functions, and B a kr-adversary making q oracle queries. Then there is a PRF adversary A making $q+1$ oracle queries such that:

$$\mathbf{Adv}_F^{\mathrm{kr}}(B) \leq \mathbf{Adv}_F^{\mathrm{prf}}(A) + 2^{-L}$$

The running time of A is that of B plus $O(q(\ell + L))$ plus the time for one computation of F.

Implication:

F PRF-secure $\Rightarrow$ F is KR-secure.

Our Assumptions

DES, AES are good block ciphers in the sense of being PRF-secure to the maximum extent possible.