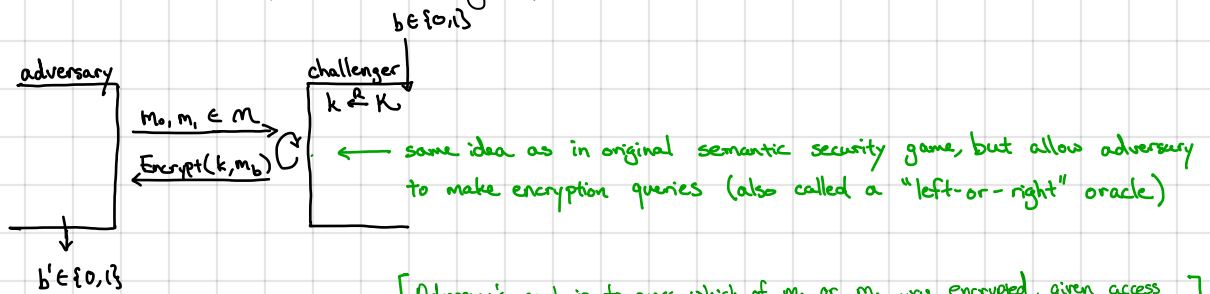<u>Definition</u>: An encryption scheme $\Pi_{SE} = (Encrypt, Decrypt)$ is secure against chosen-plaintext attacks (CPA-secure) if for all efficient adversaries A:

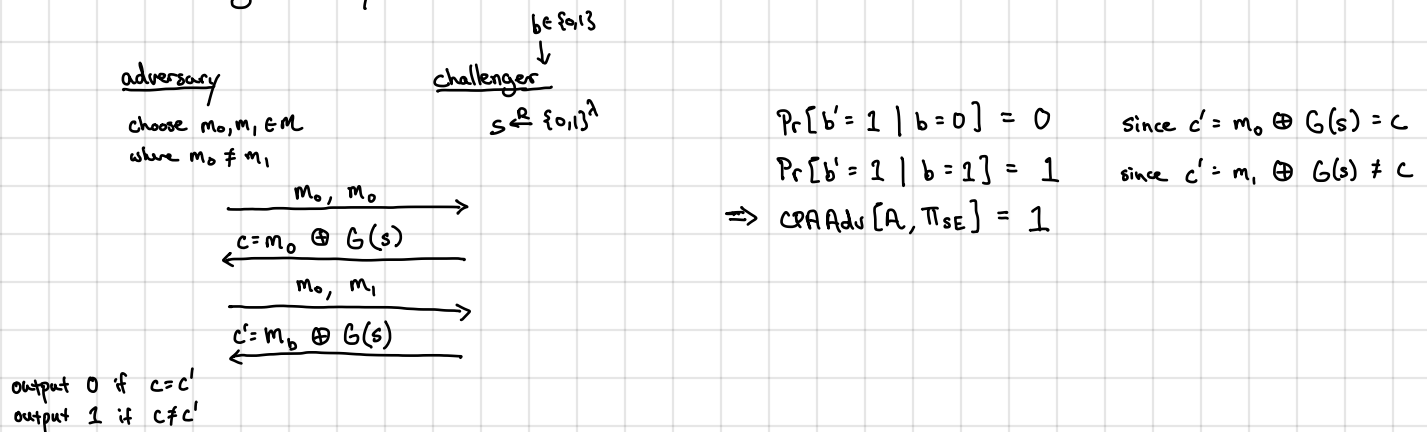$$CPAAdv[A, \Pi_{SE}] = \left| Pr[W_0 = 1] - Pr[W_1 = 1] \right| = negl.$$

Where W_b ($b \in \{0,1\}$) is the output of the following experiment:

same idea as in original semantic security game, but allows adversary to make encryption queries (also called a "left-or-right" oracle)

Adversary's goal is to guess which of m_0 or m_1 was encrypted, given access to an <u>encryption oracle</u> (i.e., adversary gets to see encryptions of messages of its choice.

<u>Claim</u>. A stream cipher is <u>not</u> CPA-secure.

<u>Proof</u>. Consider the following adversary:

$$Pr[b' = 1 \mid b = 0] = 0 \quad \text{since } c' = m_0 \oplus G(s) = c$$
$$Pr[b' = 1 \mid b = 1] = 1 \quad \text{since } c' = m_1 \oplus G(s) \neq c$$
$$\Rightarrow CPAAdv[A, \Pi_{SE}] = 1$$

<u>Observe</u>: Above attack works for any <u>deterministic</u> encryption scheme.

$\Rightarrow$ CPA-secure encryption must be <u>randomized</u>!

$\Rightarrow$ To be reusable, cannot be deterministic. Encrypting the same message twice should not reveal that identical messages were encrypted.

To build a CPA-secure encryption scheme, we will use a "block cipher"

- Block cipher is an <u>invertible</u> keyed function that takes a block of n input bits and produces a block of n output bits
- Examples include 3DES (key size 168 bits, block size 64 bits)
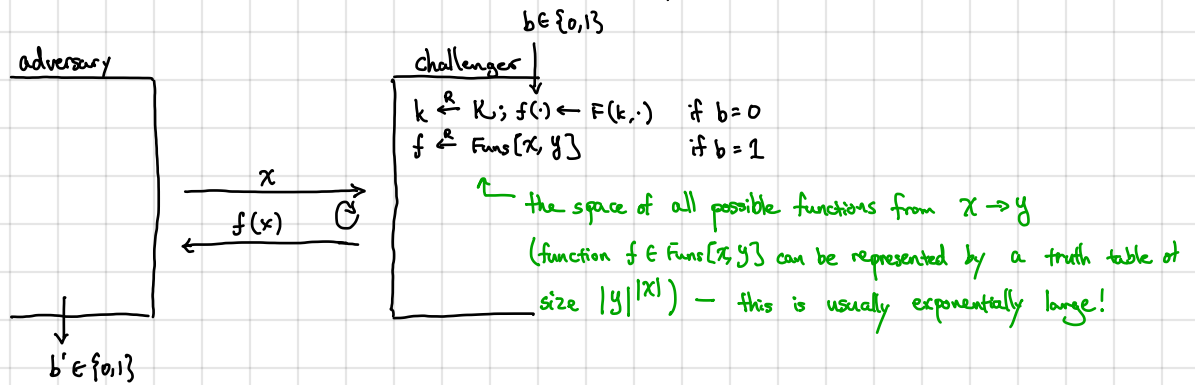 AES (key size 128 bits, block size 128 bits)   block ciphers

Will define block ciphers abstractly first: pseudorandom functions (PRFs) and pseudorandom permutations (PRPs)

$\hookrightarrow$ <u>General idea</u>: PRFs behave like <u>random</u> functions
  PRPs behave like <u>random</u> permutations

Definition. A function $F: K \times X \to Y$ with key-space K, domain X, and range Y is a pseudorandom function (PRF) if for all efficient adversaries A, $|W_0 - W_1| = \text{negl.}$, where W_b is the probability the adversary outputs 1 in the following experiment:

$b \in \{0,1\}$

adversary | challenger

$k \xleftarrow{R} K; \; f(\cdot) \leftarrow F(k, \cdot)$ if $b = 0$

$f \xleftarrow{R} \text{Funs}[X, Y]$ if $b = 1$

$\xrightarrow{\quad x \quad}$

$\xleftarrow{\quad f(x) \quad}$ $\circlearrowleft$

$b' \in \{0,1\}$

↳ the space of all possible functions from $X \to Y$
(function $f \in \text{Funs}[X, Y]$ can be represented by a truth table of size $|Y|^{|X|}$) — this is usually exponentially large!

$$\text{PRFAdv}[A, F] = |W_0 - W_1| = \left| \Pr[A \text{ outputs } 1 \mid b=0] - \Pr[A \text{ outputs } 1 \mid b=1] \right|$$

Intuitively: input-output behavior of a PRF is indistinguishable from that of a random function (to any computationally-bounded adversary)

3DES: $\{0,1\}^{168} \times \{0,1\}^{64} \to \{0,1\}^{64}$ $|K| = 2^{168}$ $|\text{Funs}[X,Y]| = (2^{64})^{(2^{64})}$

AES: $\{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{128}$ $|K| = 2^{128}$ $|\text{Funs}[X,Y]| = (2^{128})^{(2^{128})}$

} space of random functions is exponentially-larger than key-space!

Definition: A function $F: K \times X \to X$ is a pseudorandom permutation (PRP) if

– for all keys k, $F(k, \cdot)$ is a permutation and moreover, there exists an efficient algorithm to compute $F^{-1}(k, \cdot)$:

$$\forall k \in K : \forall x \in X : F^{-1}(k, F(k,x)) = x$$

– for $k \xleftarrow{R} K$, the input-output behavior of $F(k, \cdot)$ is computationally indistinguishable from $f(\cdot)$ where $f \xleftarrow{R} \text{Perm}[X]$ and $\text{Perm}[X]$ is the set of all permutations on X (analogous to PRF security)

Note: a block cipher is another term for PRP (just like stream ciphers are PRGs)

Observe that a block cipher can be used to construct a PRG:

$F : \{0,1\}^\lambda \times \{0,1\}^n \longrightarrow \{0,1\}^n$ be a block cipher

Define $G : \{0,1\}^\lambda \longrightarrow \{0,1\}^{\ell n}$ as

$$G(k) = F(k,1) \| F(k,2) \| \cdots \| F(k,\ell) \qquad \leftarrow \text{this stream cipher allows } \underline{random} \text{ access!}$$
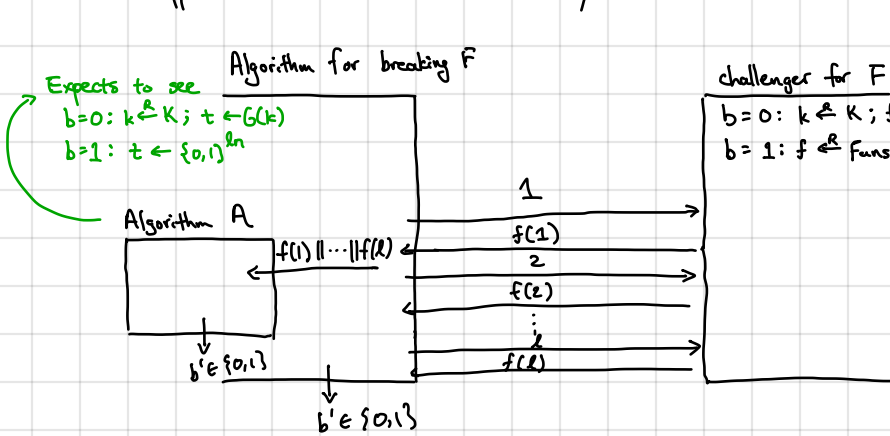
↑ string concatenation   ↑ write input as an n-bit string (just require that $n > \log \ell$)

we said PRP above (will revisit this) ↴

<u>Theorem.</u> If F is a secure PRF, then G is a secure PRG.

<u>Proof.</u> As usual, we show the contrapositive: if G is not a secure PRG, then F is not a secure PRF.

Suppose we have efficient adversary A for G. We use A to build adversary for F:

Expects to see
$b=0: k \xleftarrow{R} K; t \leftarrow G(k)$
$b=1: t \leftarrow \{0,1\}^{\ell n}$

Algorithm for breaking F

Algorithm A

$f(1) \| \cdots \| f(\ell)$

$b' \in \{0,1\}$

$b' \in \{0,1\}$

1 $f(1)$
2 $f(2)$
⋮   ⋮
ℓ $f(\ell)$

$b \in \{0,1\}$

challenger for F
$b=0: k \xleftarrow{R} K; f \leftarrow F(k, \cdot)$
$b=1: f \xleftarrow{R} \text{Funs}[\{0,1\}^n, \{0,1\}^n]$

1. If $\ell = $ poly, then B is efficient
2. If $b=0$: B sends $G(k)$ to A
   where k is a uniformly random key
   If $b=1$: B sends uniformly random string (f is <u>random</u> function) to A
3. $\text{PRFAdv}[B,F] = |\Pr[b'=1 \mid b=0] - \Pr[b'=1 \mid b=1]|$
   $= |\Pr[A \text{ outputs } 1 \mid b=0] - \Pr[A \text{ outputs } 1 \mid b=1]|$
   $= \text{PRGAdv}[A,G]$
   which is non-negligible by assumption.

But... we used a block cipher (PRP) in our construction above. Does the proof still go through?

Not quite...

for a random function, $f(1) = f(2)$ with probability $\frac{1}{2^n}$ ⎫ but 2^{-n} might be very very small...
for a random permutation, $f(1) = f(2)$ with probability 0 ⎭ adversary won't notice unless it sees a "collision" [i.e., two values x, y where $f(x) = f(y)$]

<u>PRF Switching Lemma.</u> Let $F : K \times X \longrightarrow X$ be a secure PRP. Then, for any Q-query adversary A:
$$|\text{PRPAdv}[A,F] - \text{PRFAdv}[A,F]| \leq \frac{Q^2}{2|X|}$$

<u>Proof Idea.</u> Adversary essentially cannot tell the difference unless it sees a collision. If there is no collision, then it is just seeing random values. How many queries before there is a collision? Birthday paradox: $Q \sim \sqrt{|X|}$

<u>Take-away:</u> If $|X|$ is large (e.g., exponential), then we can use a PRP as a PRF.
- 3DES: $n = 64$ so $|X| = 2^{64}$ [if adversary makes $\ll 2^{32}$ queries, then can use it as a PRF]
- AES: $n = 128$ so $|X| = 2^{128}$ [if adversary makes $\ll 2^{64}$ queries, then can use it as a PRF]