

COMS BC3262: Introduction to Cryptography

# Lecture 5: CPA-Secure Encryption from PRFs, PRPs, and Block Ciphers

# Logistics

Office hours:

- **Eysa:** Mondays 3-5, Milstein 512
- **Mark:** Wednesday 4:30-6:30 this week, Tuesdays 6:30-8:30 starting next week, Milstein 503

PS1 is due Thursday (tomorrow), PS2 is released today

Lowest PS grade is dropped

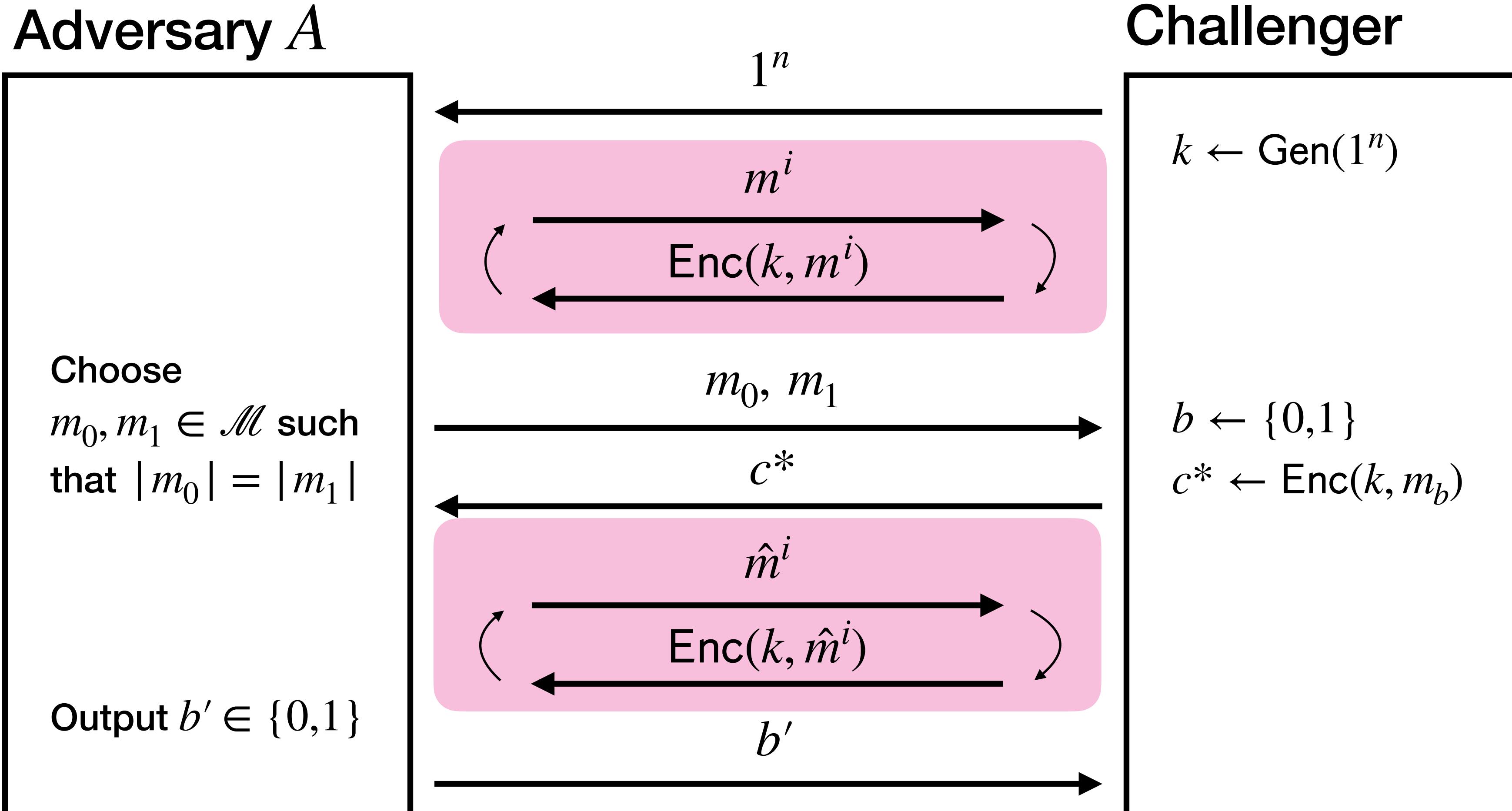
**Please see EdStem for clarifications on some of the questions (mostly notation)**

# Today's Lecture

- CPA-secure encryption from PRFs
  - A brief overview of hybrid arguments
- PRPs
- Block ciphers

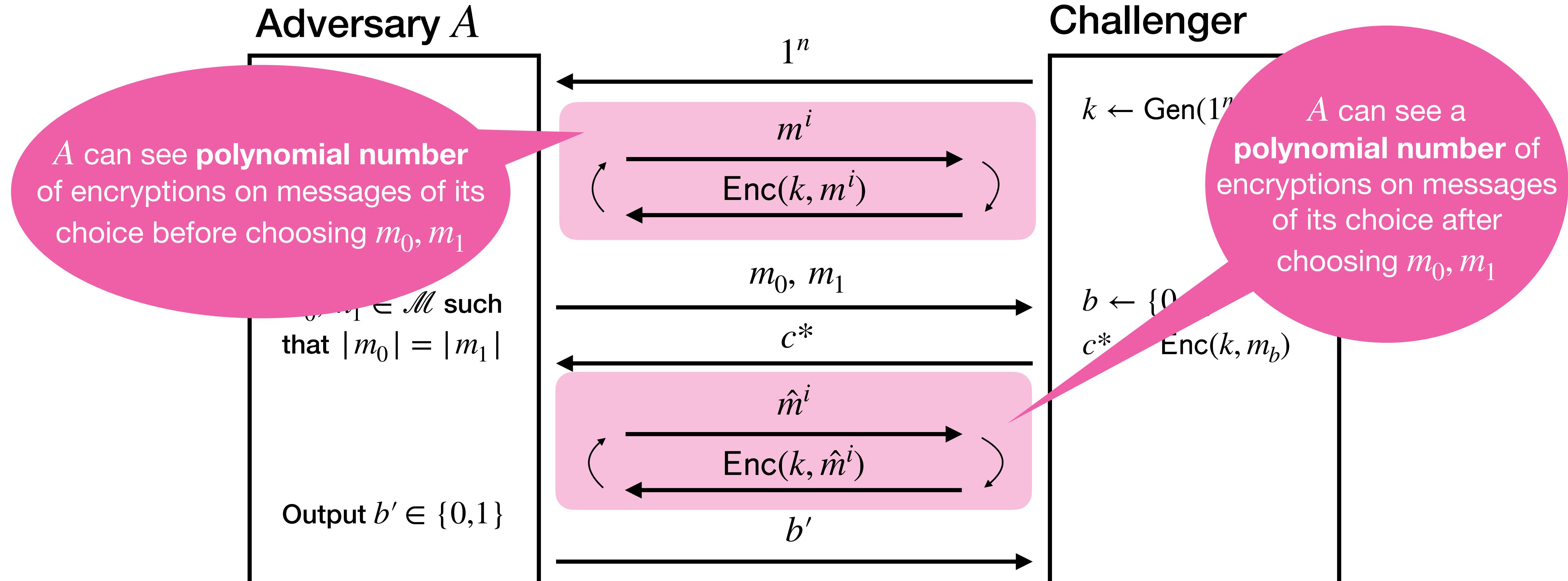
# Chosen-Plaintext Attack (CPA)

# Chosen-Plaintext Attack (CPA)



$$\begin{aligned}\text{PrivK}_{\Pi, A}^{\text{CPA}}(n) &= 1 \text{ if } b' = b \\ &\quad \text{and } 0 \text{ otherwise}\end{aligned}$$

# Chosen-Plaintext Attack (CPA)



$$\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1 \text{ if } b' = b \text{ and } 0 \text{ otherwise}$$

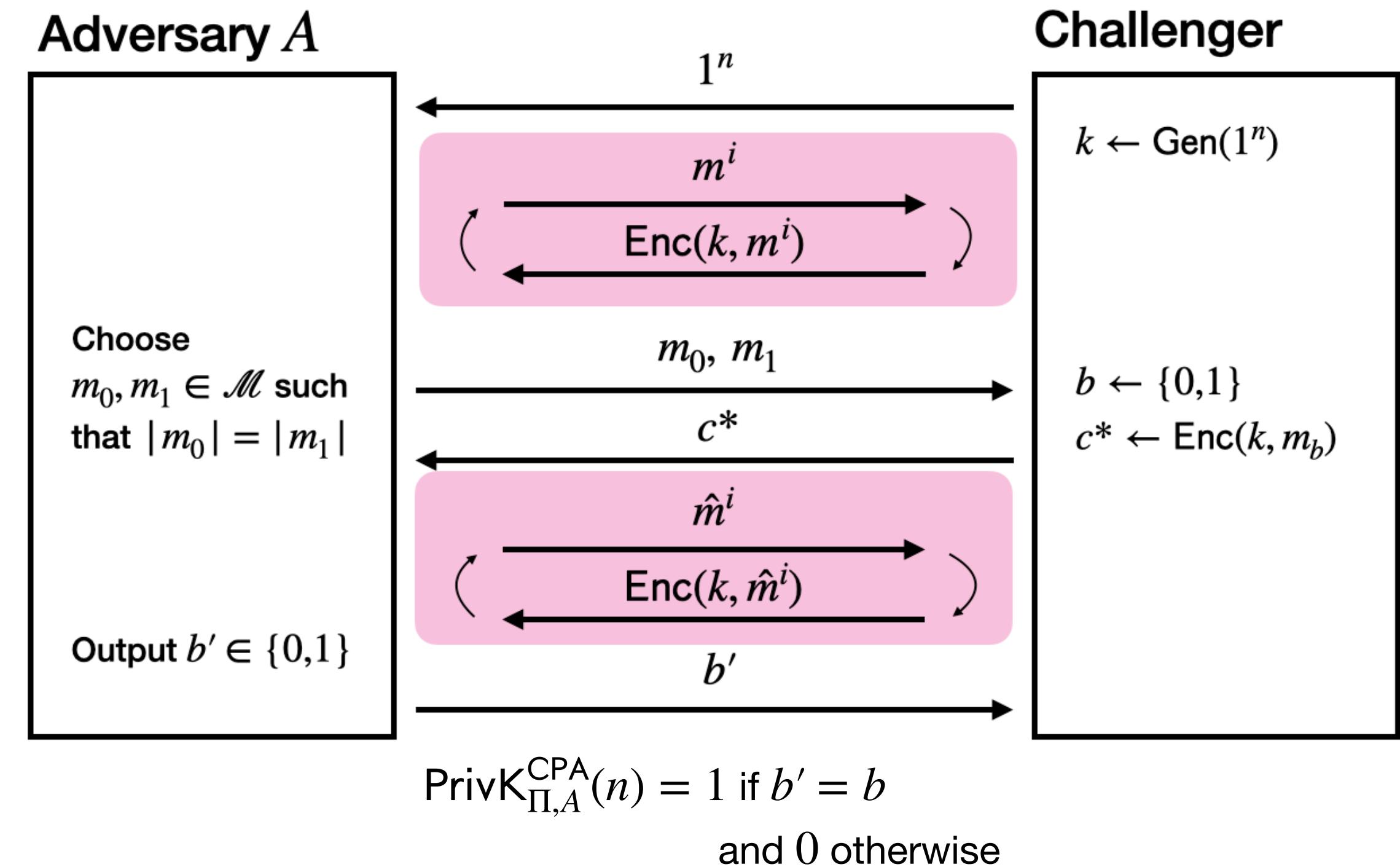
We typically refer to these types of queries as having “oracle access”

# Chosen-Plaintext Attack (CPA)

## Definition:

$\Pi$  has **indistinguishable encryptions under chosen-plaintext attack** (or CPA-security) if for every PPT adversary  $A$  there exists a negligible function  $\epsilon(\cdot)$  such that

$$\Pr[\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1] \leq \frac{1}{2} + \epsilon(n)$$



## Notes:

- CPA Security implies multiple message security
- Any CPA secure private key encryption scheme must have a **randomized** encryption algorithm

# Pseudorandom Functions (PRFs)

# Pseudorandom Functions (PRFs)

Idea: A function that “looks like” a **truly random function** (but is efficient)

- We want the benefits of a random function but **polynomial-length representation** (i.e., polynomial-length key)
- No PPT adversary should be able to tell the difference between your polynomial length key and a truly random function
- This is the idea behind a **pseudorandom function (PRF)**

# Pseudorandom Functions (PRFs)

## Definition:

Let  $F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$  be an efficient, length-preserving, keyed function. We say that  $F$  is a **pseudorandom function (PRF)** if for all PPT  $D$  there exists a negligible function  $\epsilon(\cdot)$  such that

$$\left| \Pr_{k \leftarrow \{0,1\}^n} [D^{F_k}(1^n) = 1] - \Pr_{f \leftarrow \mathcal{F}_n} [D^f(1^n) = 1] \right| \leq \epsilon(n)$$

# Pseudorandom Functions (PRFs)

## Definition:

Let  $F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$  be an efficient, length-preserving, keyed function. We say that  $F$  is a **pseudorandom function (PRF)** if for all PPT  $D$  there exists a negligible function  $\epsilon(\cdot)$  such that

$$\left| \Pr_{k \leftarrow \{0,1\}^n} [D^{F_k}(1^n) = 1] - \Pr_{f \leftarrow \mathcal{F}_n} [D^f(1^n) = 1] \right| \leq \epsilon(n)$$

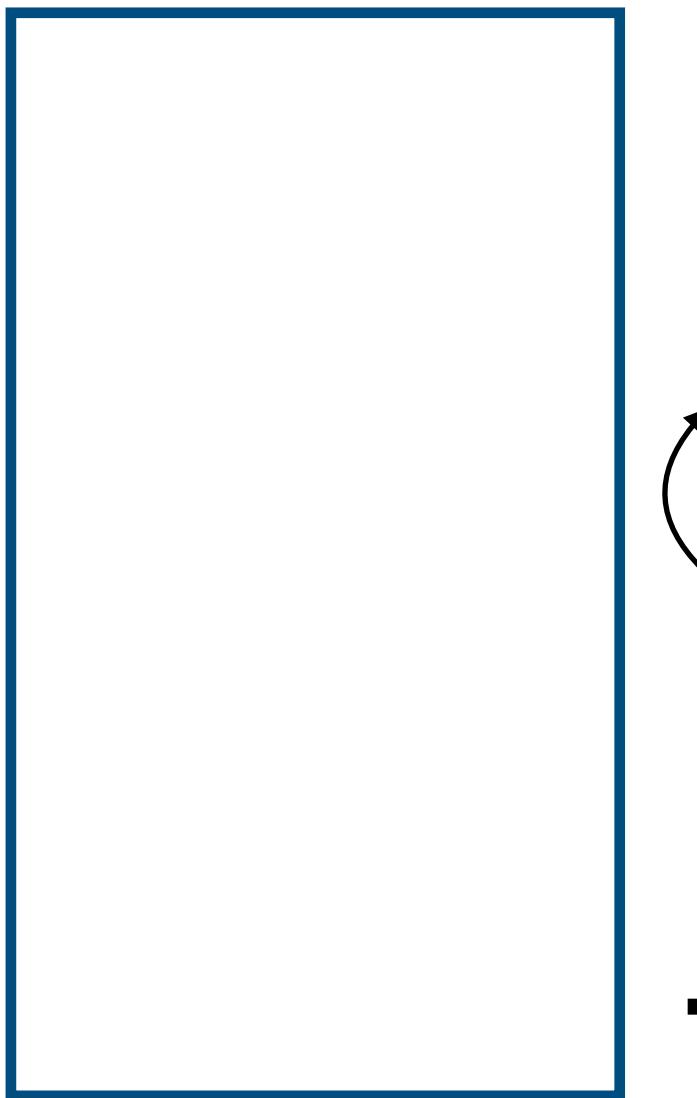
*D* is given oracle access to a keyed function  $F$ , where  $k$  is sampled at random (key is not given to  $D$ )

*D* is given oracle access to a truly random function  $f$

# PRF Distinguisher

PRF World

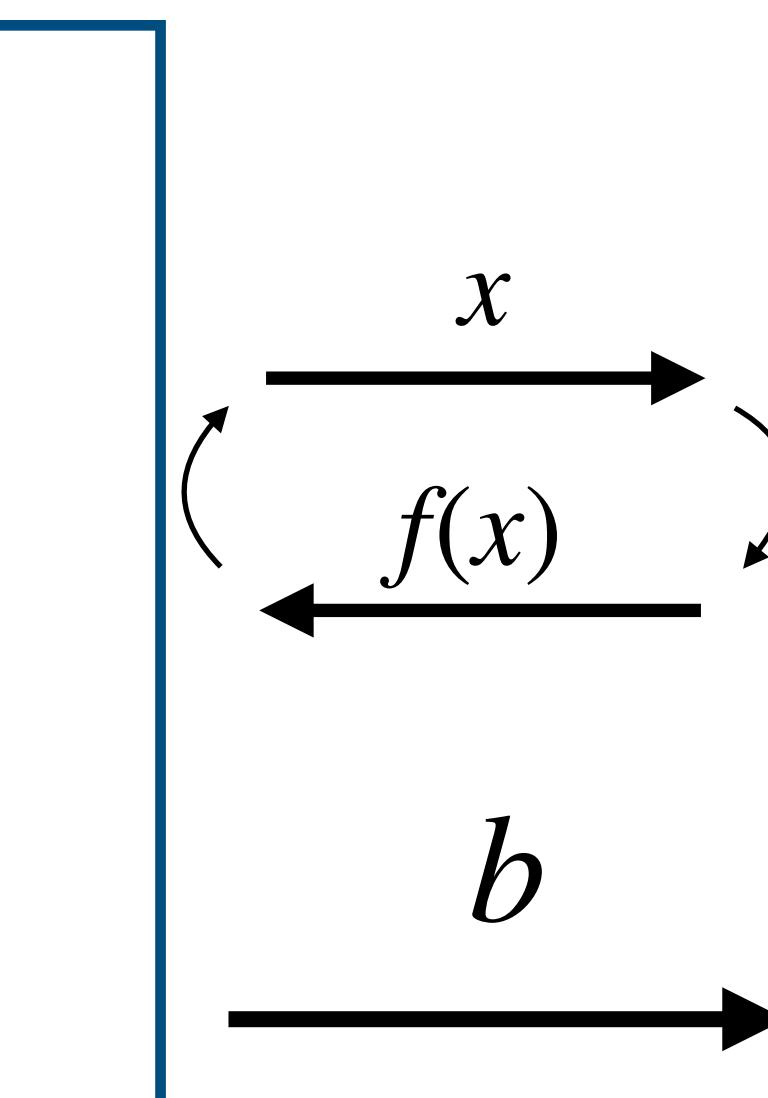
Distinguisher  $D$



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

Random World

Distinguisher  $D$



$$f \leftarrow \mathcal{F}_n$$

$\approx$

$$\left| \Pr_{k \leftarrow \{0,1\}^n} [D^{F_k}(1^n) = 1] - \Pr_{f \leftarrow \mathcal{F}_n} [D^f(1^n) = 1] \right| \leq \epsilon(n)$$

# Security Games for PRF vs PRG

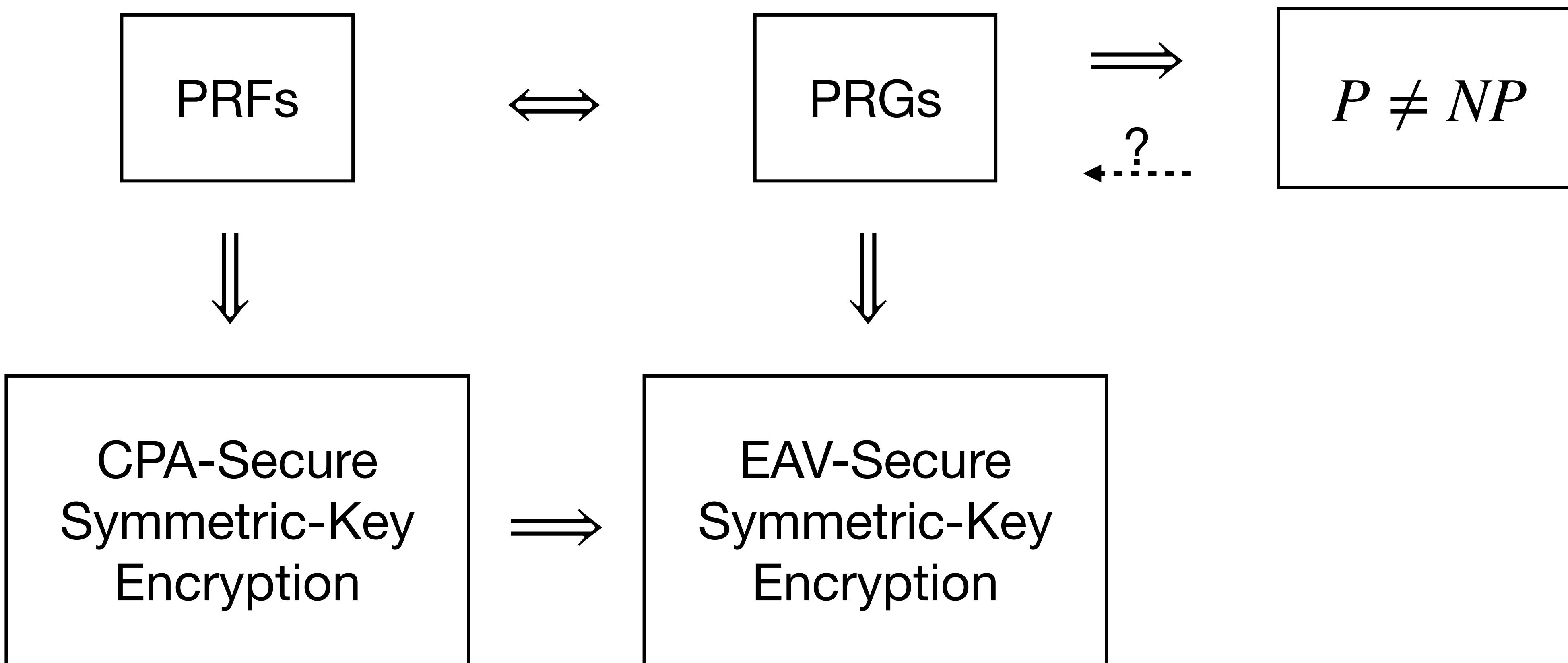
## PRF:

- $k$  is chosen uniformly at random (not known to  $D$ )
- $D$  chooses (and knows) points the function will be evaluated on  $(x_1, x_2, \dots)$ 
  - $D$  can see up to polynomially many  $(x_1, f(x_1)), (x_2, f(x_2)), \dots$
- Security of the PRF says that  $D$  behaves almost the same as when  $f = F_k$  and when  $f =$  truly random

## PRG:

- $D$  gets a single string (“one shot”)
  - $D$  does not get to choose the PRG seed or see multiple evaluations
- Security of the PRG says  $D$  behaves almost the same when:
  - $s$  is chosen at random (not known to  $D$ ) and  $G(s)$  is given to  $D$
  - $D$  is given a truly random string

# The World of Crypto Primitives We've Seen So Far



# CPA-Secure Encryption from PRFs

# CPA-Secure Encryption from PRFs

Let  $F : \{0,1\}^n \times \{0,1\}^{\ell_{\text{in}}} \rightarrow \{0,1\}^{\ell_{\text{out}}}$  be a PRF

We define  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  as follows:

- $\text{Gen}(1^n)$ : Sample  $k \leftarrow \{0,1\}^n$
- $\text{Enc}(k, m)$ : On input  $m \in \{0,1\}^{\ell_{\text{in}}}$ , sample  $r \leftarrow \{0,1\}^{\ell_{\text{in}}}$  and output
$$c = (r, F_k(r) \oplus m)$$
- $\text{Dec}(k, c)$ : On input  $c = (c_1, c_2)$ , output  $F_k(c_1) \oplus c_2$

**Theorem:** If  $F$  is a PRF, then  $\Pi$  is CPA-secure

# Proof Idea

Let  $F : \{0,1\}^n \times \{0,1\}^{\ell_{\text{in}}} \rightarrow \{0,1\}^{\ell_{\text{out}}}$  be a PRF

We define  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  as follows:

- $\text{Gen}(1^n)$ : Sample  $k \leftarrow \{0,1\}^n$
- $\text{Enc}(k, m)$ : On input  $m \in \{0,1\}^{\ell_{\text{in}}}$ , sample  $r \leftarrow \{0,1\}^{\ell_{\text{in}}}$  and output
$$c = (r, F_k(r) \oplus m)$$
- $\text{Dec}(k, c)$ : On input  $c = (c_1, c_2)$ , output  $F_k(c_1) \oplus c_2$

**Theorem:** If  $F$  is a PRF, then  $\Pi$  is CPA-secure

Consider a version  $\hat{\Pi}$  where we use a truly random function instead of a PRF. Split the proof into two parts:

- **Part 1:** The schemes  $\Pi$  and  $\hat{\Pi}$  are computationally indistinguishable.  
(intuitively: no PPT  $A$  playing in the CPA game can tell whether it's playing with  $\Pi$  or  $\hat{\Pi}$ )

- **Part 2:** The scheme  $\hat{\Pi}$  is CPA-secure.  
(intuitively: no PPT  $A$  can win the CPA game with probability better than  $1/2 + \text{negl}(n)$ )

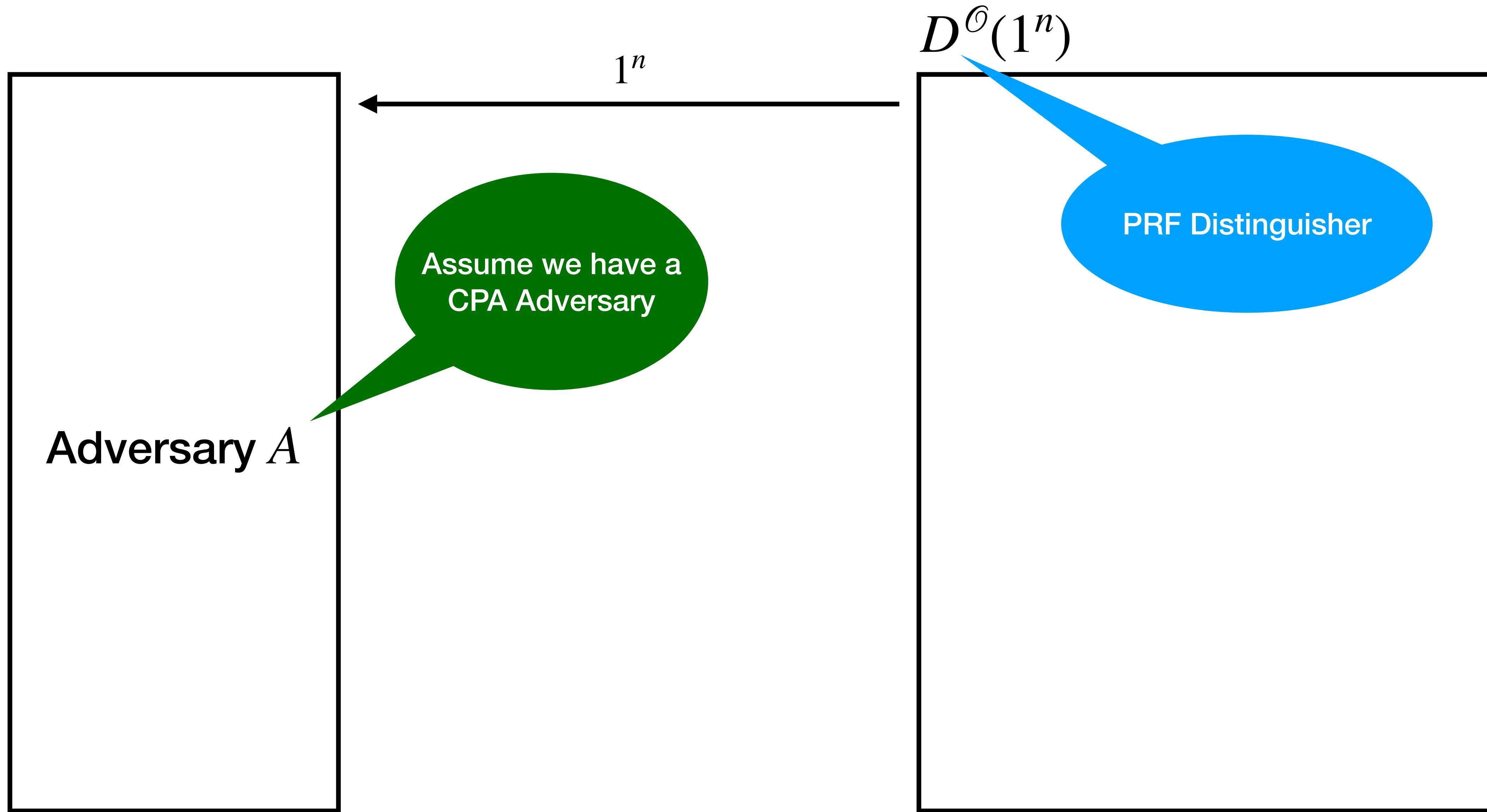
# Proving Lemma 1

**Lemma 1:** For all PPT  $A$ , there exists a negligible function  $\epsilon_1(\cdot)$  s.t.

$$|\Pr[\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1] - \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1]| \leq \epsilon_1(n)$$

**Proof:** Let  $A$  be any PPT CPA adversary. We will construct a distinguisher  $D$  that uses  $A$  to try to break the PRF security of  $F$  (i.e., distinguish  $F$  from random)

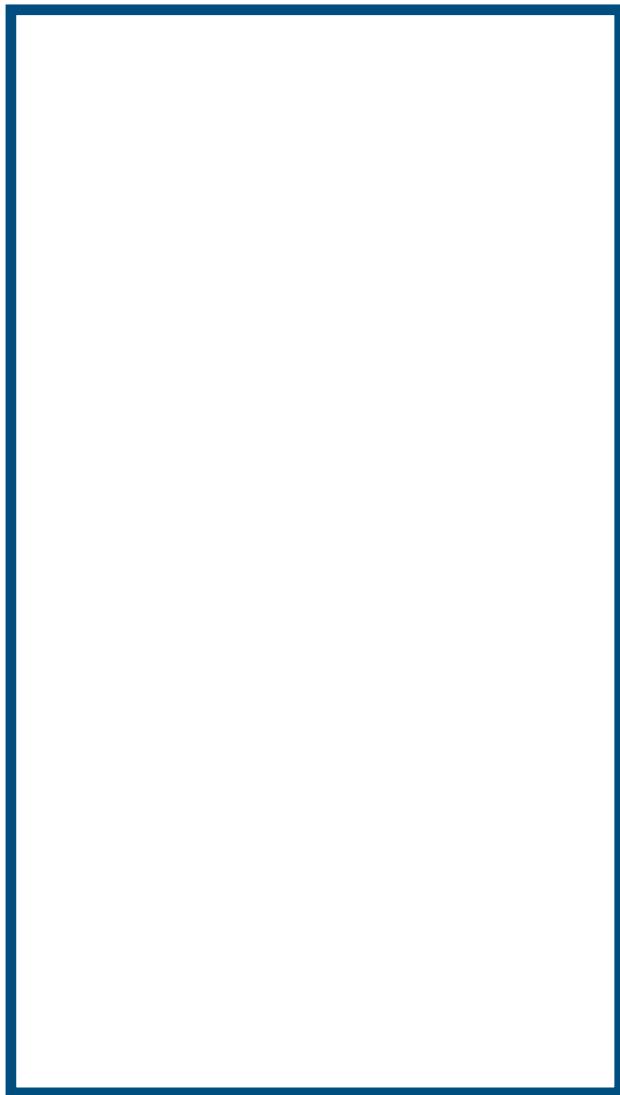
# Reduction for Lemma 1



# Recall: PRF Security Game

PRF World

Distinguisher  $D$



Oracle  $\mathcal{O}$

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

$$y = F_k(x)$$

$\approx$

Random World

Distinguisher  $D$

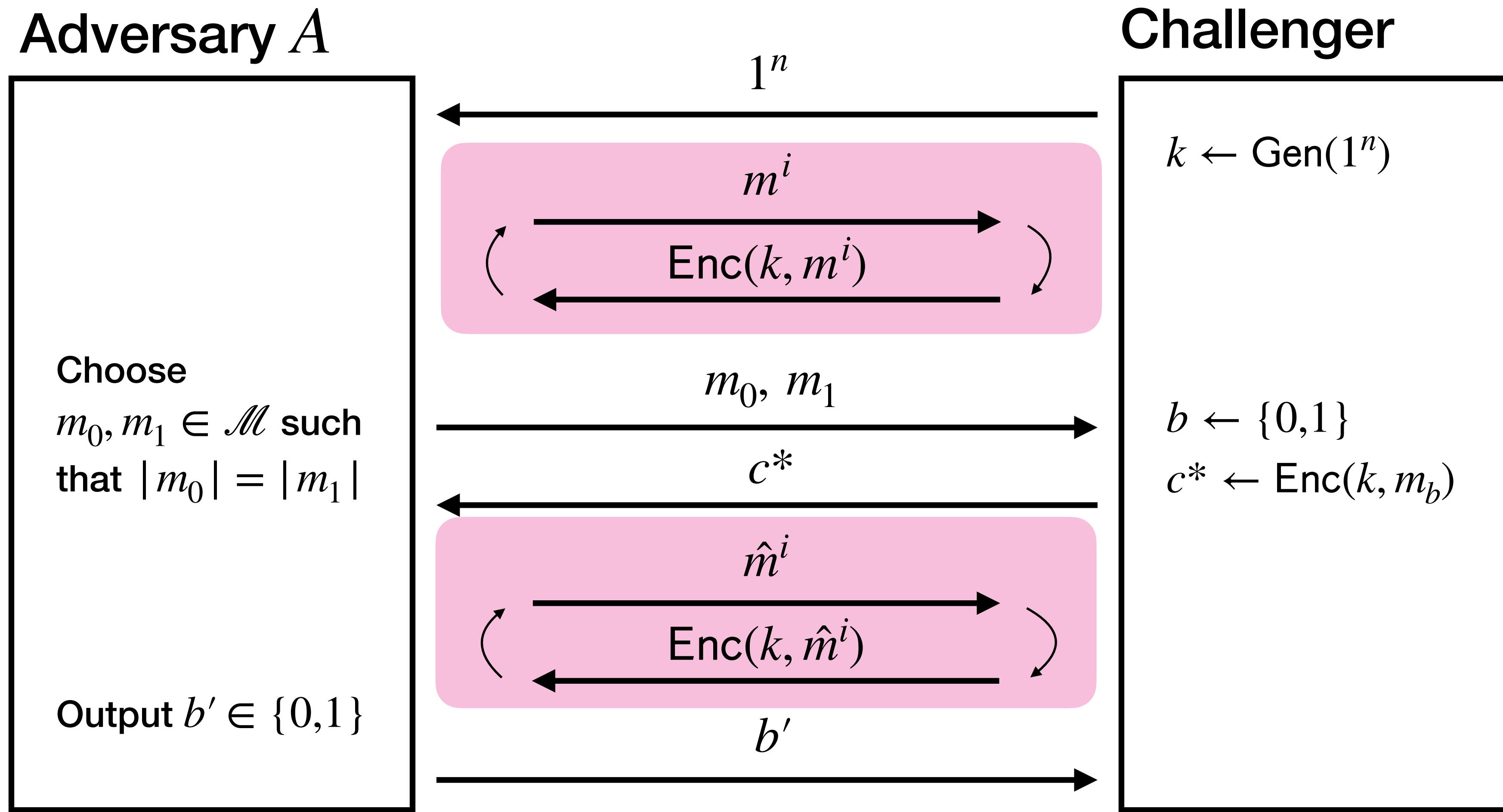


Oracle  $\mathcal{O}$

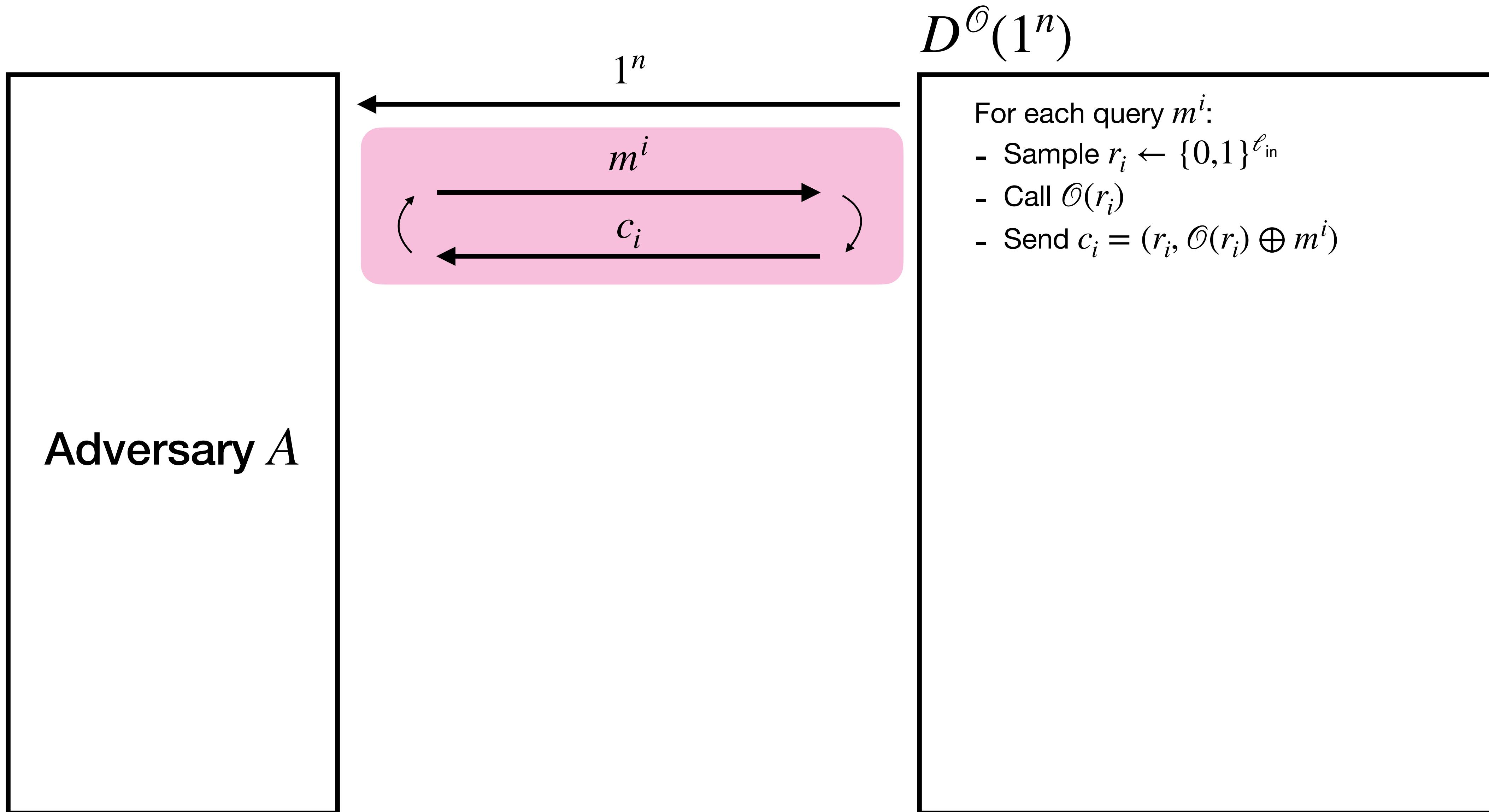
$$f \leftarrow \mathcal{F}_n$$

$$y = f(x)$$

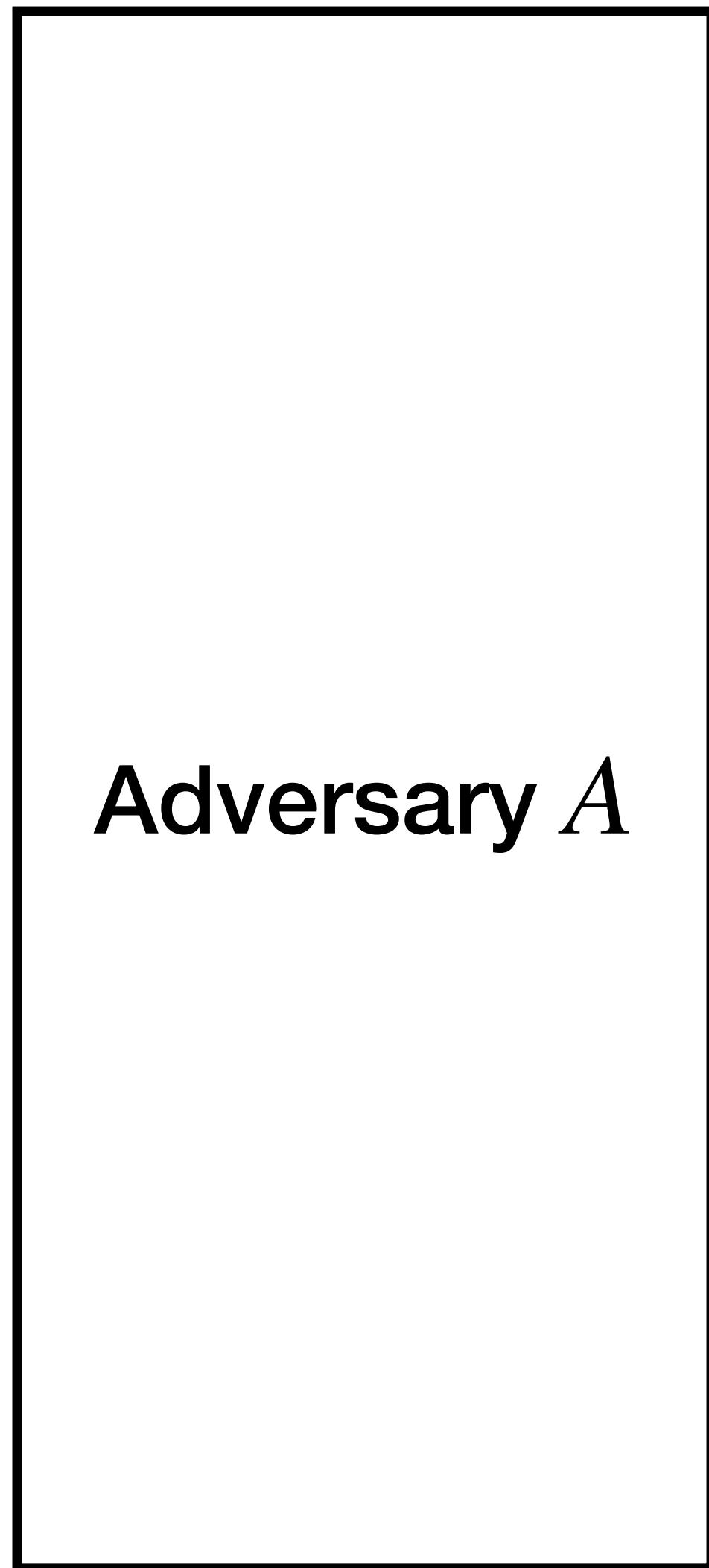
# Recall: Chosen-Plaintext Attack (CPA)



# Reduction for Lemma 1



# Reduction for Lemma 1



$D^{\mathcal{O}}(1^n)$

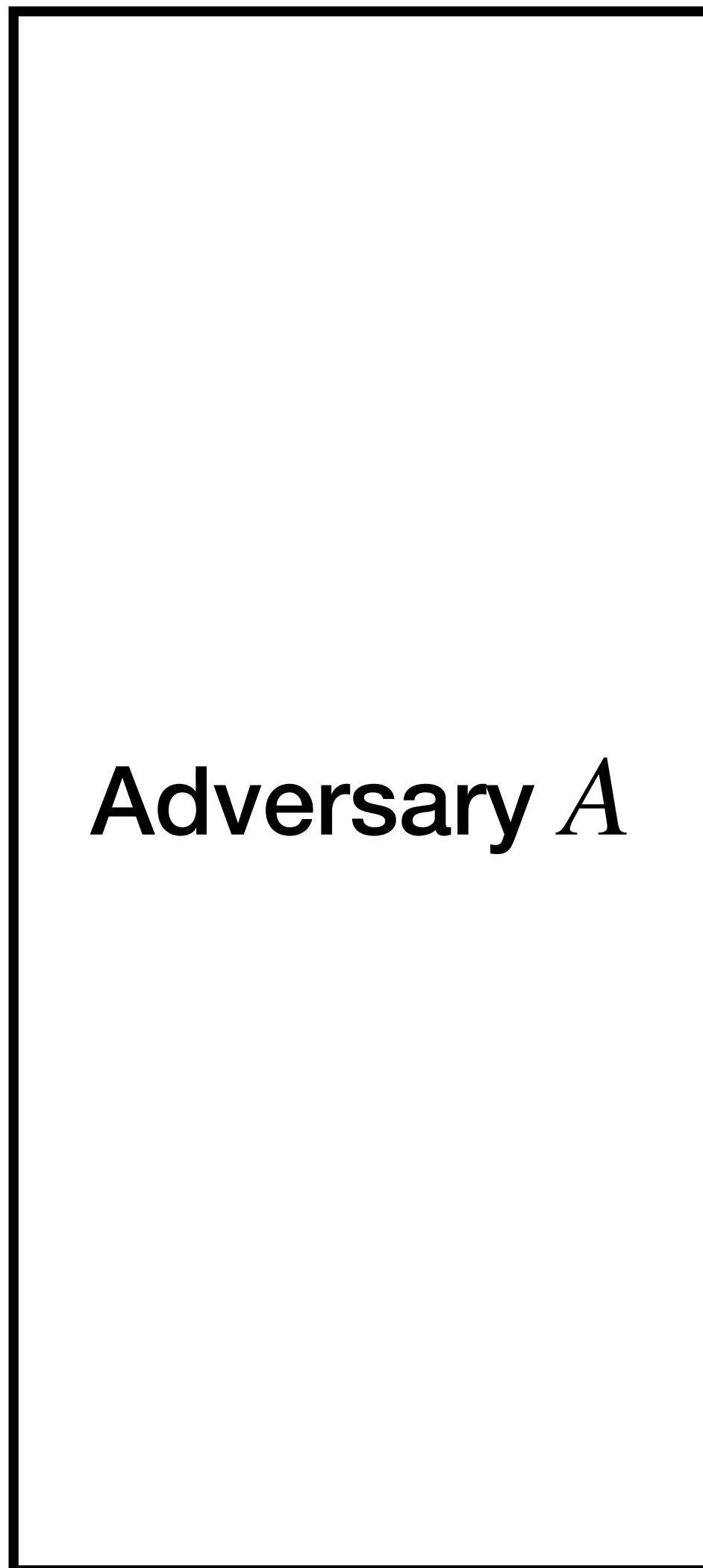
For each query  $m^i$ :

- Sample  $r_i \leftarrow \{0,1\}^{\ell_{\text{in}}}$
- Call  $\mathcal{O}(r_i)$
- Send  $c_i = (r_i, \mathcal{O}(r_i) \oplus m^i)$

When given  $m_0, m_1$ :

- Sample  $b \leftarrow \{0,1\}$
- Sample  $r^* \leftarrow \{0,1\}^{\ell_{\text{in}}}$
- Send  $c^* = (r^*, \mathcal{O}(r^*) \oplus m_b)$

# Reduction for Lemma 1



$D^{\mathcal{O}}(1^n)$

For each query  $m^i$ :

- Sample  $r_i \leftarrow \{0,1\}^{\ell_{\text{in}}}$
- Call  $\mathcal{O}(r_i)$
- Send  $c_i = (r_i, \mathcal{O}(r_i) \oplus m^i)$

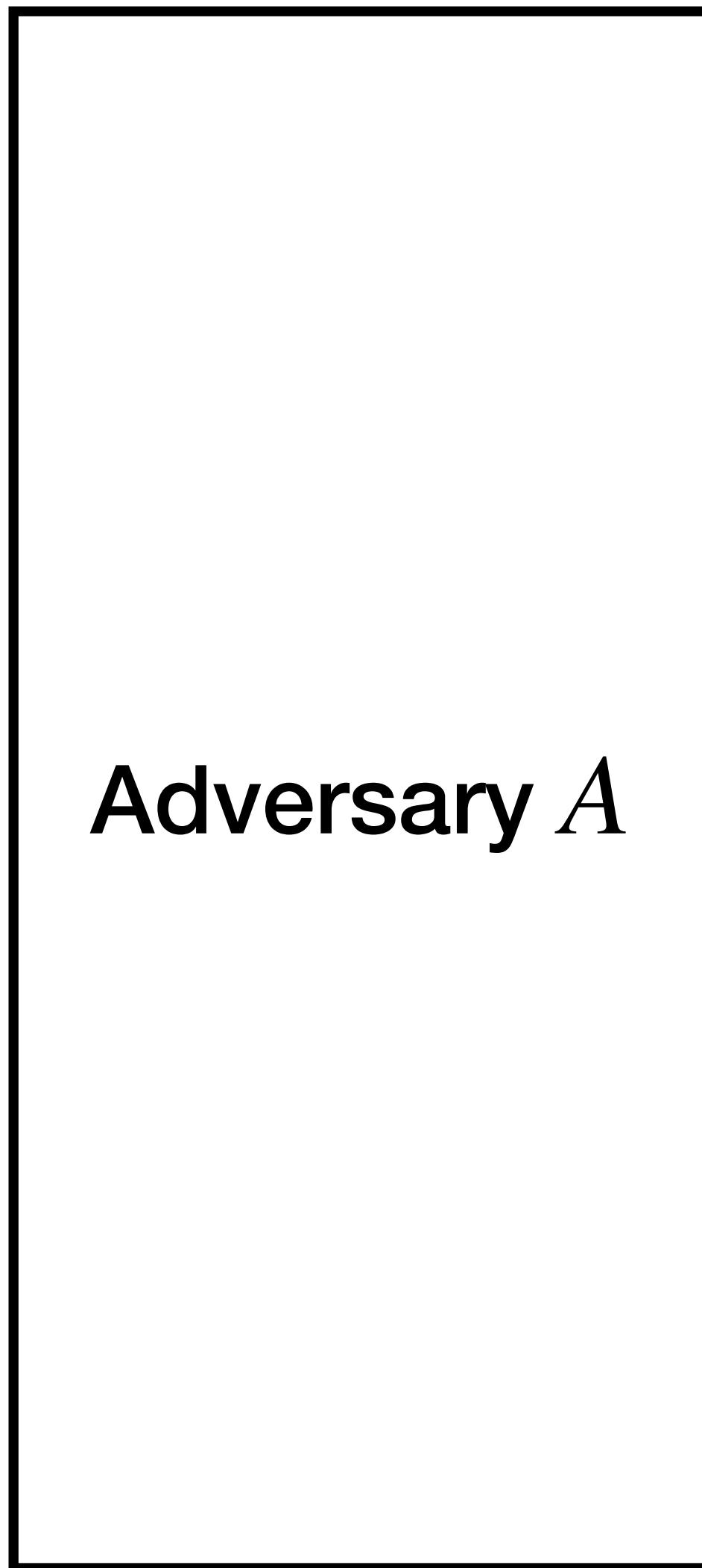
When given  $m_0, m_1$ :

- Sample  $b \leftarrow \{0,1\}$
- Sample  $r^* \leftarrow \{0,1\}^{\ell_{\text{in}}}$
- Send  $c^* = (r^*, \mathcal{O}(r^*) \oplus m_b)$

For each query  $\hat{m}^i$ :

- Sample  $r_i \leftarrow \{0,1\}^{\ell_{\text{in}}}$
- Send  $c_i = (r_i, \mathcal{O}(r_i) \oplus \hat{m}^i)$

# Reduction for Lemma 1



$D^{\mathcal{O}}(1^n)$

For each query  $m^i$ :

- Sample  $r_i \leftarrow \{0,1\}^{\ell_{\text{in}}}$
- Call  $\mathcal{O}(r_i)$
- Send  $c_i = (r_i, \mathcal{O}(r_i) \oplus m^i)$

When given  $m_0, m_1$ :

- Sample  $b \leftarrow \{0,1\}$
- Sample  $r^* \leftarrow \{0,1\}^{\ell_{\text{in}}}$
- Send  $c^* = (r^*, \mathcal{O}(r^*) \oplus m_b)$

For each query  $\hat{m}^i$ :

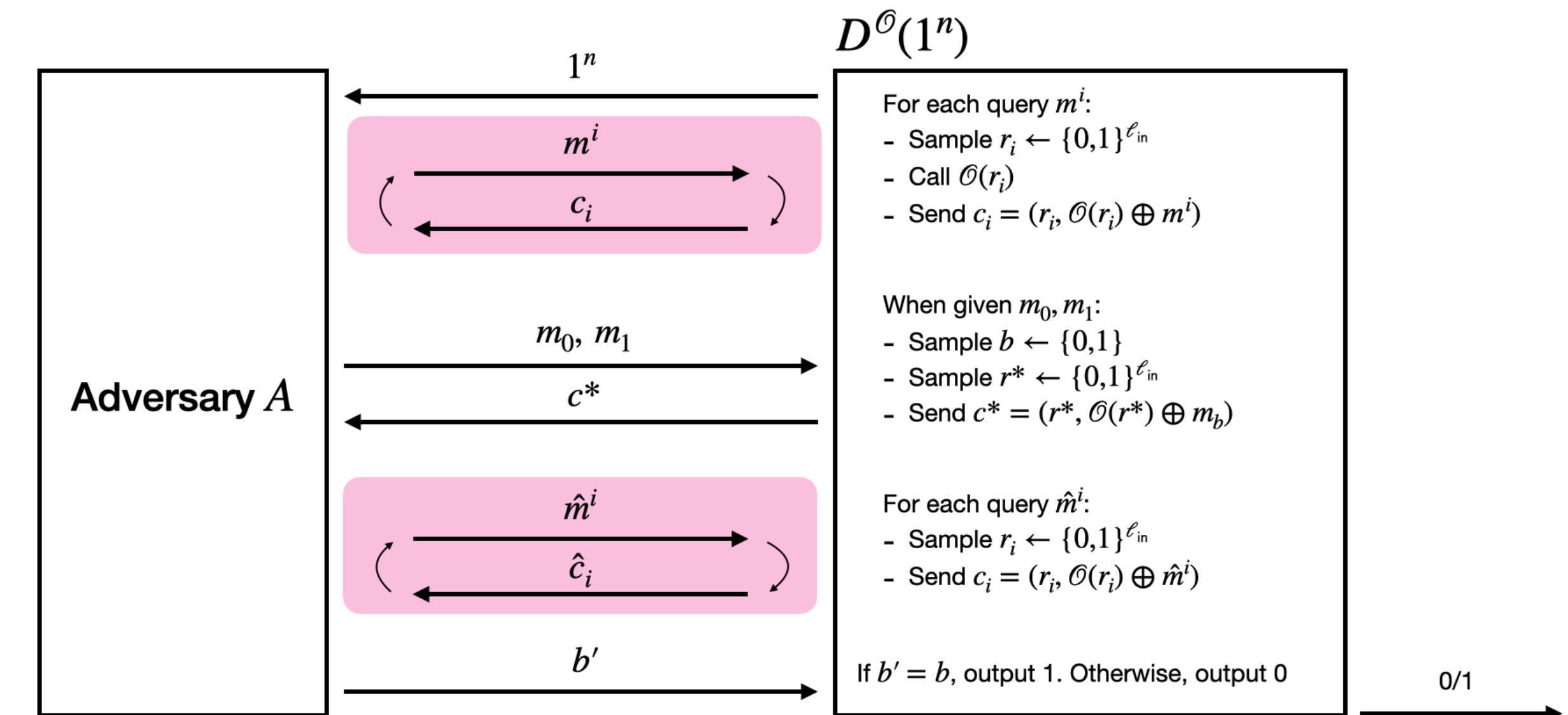
- Sample  $r_i \leftarrow \{0,1\}^{\ell_{\text{in}}}$
- Send  $c_i = (r_i, \mathcal{O}(r_i) \oplus \hat{m}^i)$

If  $b' = b$ , output 1. Otherwise, output 0

0/1

# Reduction for Lemma 1

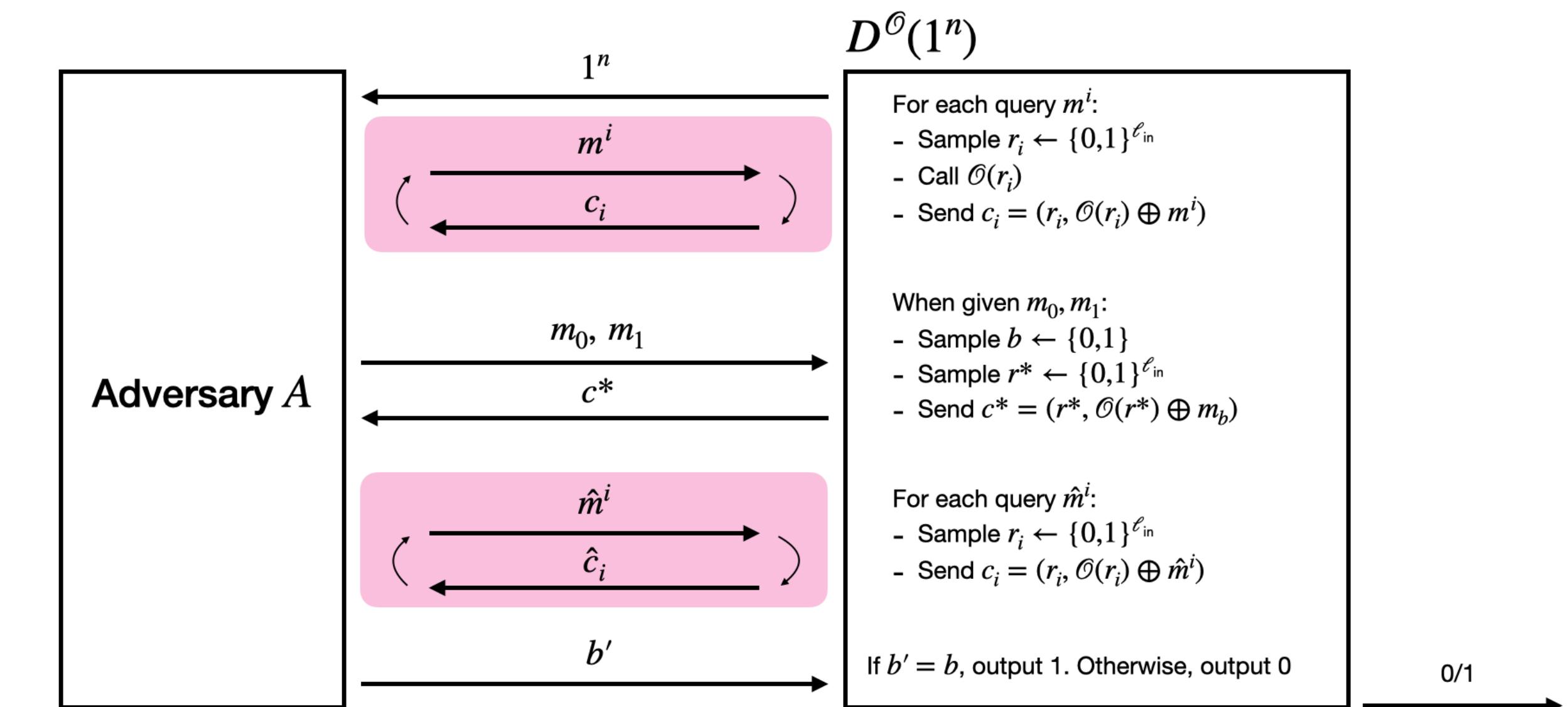
**Case 1 (PRF world):**  $\mathcal{O} \leftarrow \{F_k\}_{k \in \{0,1\}^n}$



# Reduction for Lemma 1

**Case 1 (PRF world):**  $\mathcal{O} \leftarrow \{F_k\}_{k \in \{0,1\}^n}$

$$\Pr_{k \leftarrow \{0,1\}^n} [D^{F_k}(1^n) = 1] = \Pr [\text{PrivK}_{\Pi, A}^{\text{CPA}}(n) = 1]$$

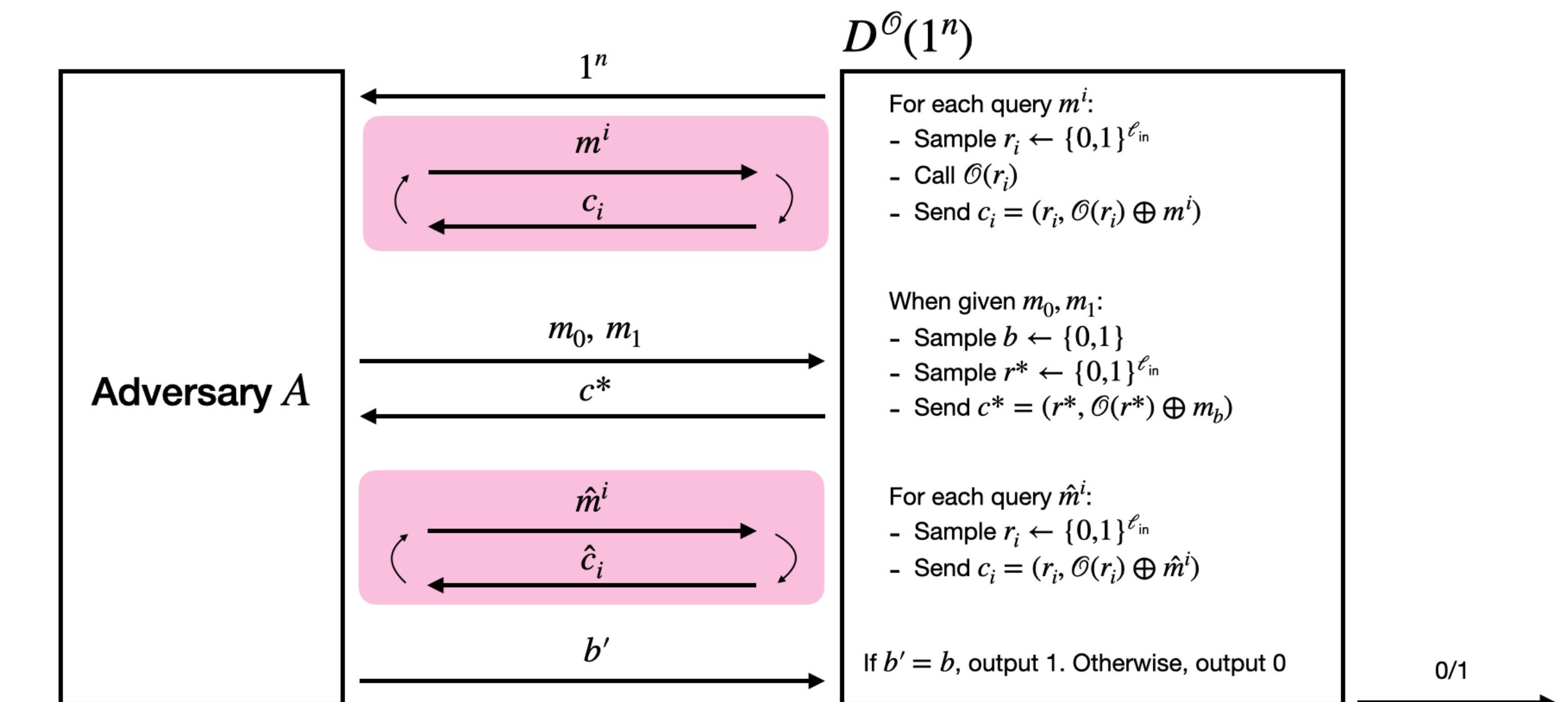


# Reduction for Lemma 1

**Case 1 (PRF world):**  $\mathcal{O} \leftarrow \{F_k\}_{k \in \{0,1\}^n}$

$$\Pr_{k \leftarrow \{0,1\}^n} [D^{F_k}(1^n) = 1] = \Pr [\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1]$$

**Case 2 (random world):**  $\mathcal{O} \leftarrow \mathcal{F}$



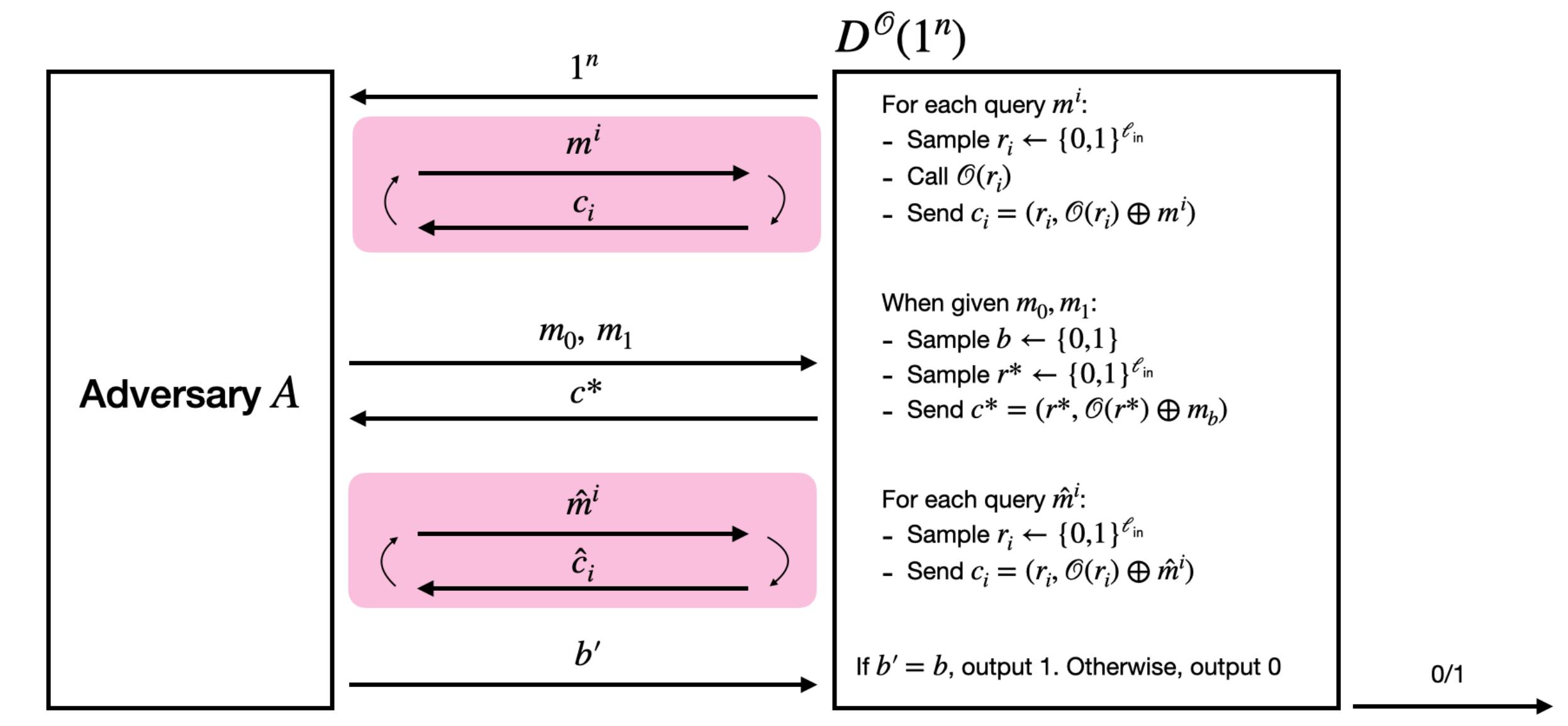
# Reduction for Lemma 1

**Case 1 (PRF world):**  $\mathcal{O} \leftarrow \{F_k\}_{k \in \{0,1\}^n}$

$$\Pr_{k \leftarrow \{0,1\}^n} [D^{F_k}(1^n) = 1] = \Pr [\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1]$$

**Case 2 (random world):**  $\mathcal{O} \leftarrow \mathcal{F}$

$$\Pr_{f \leftarrow \mathcal{F}_n} [D^f(1^n) = 1] = \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1]$$



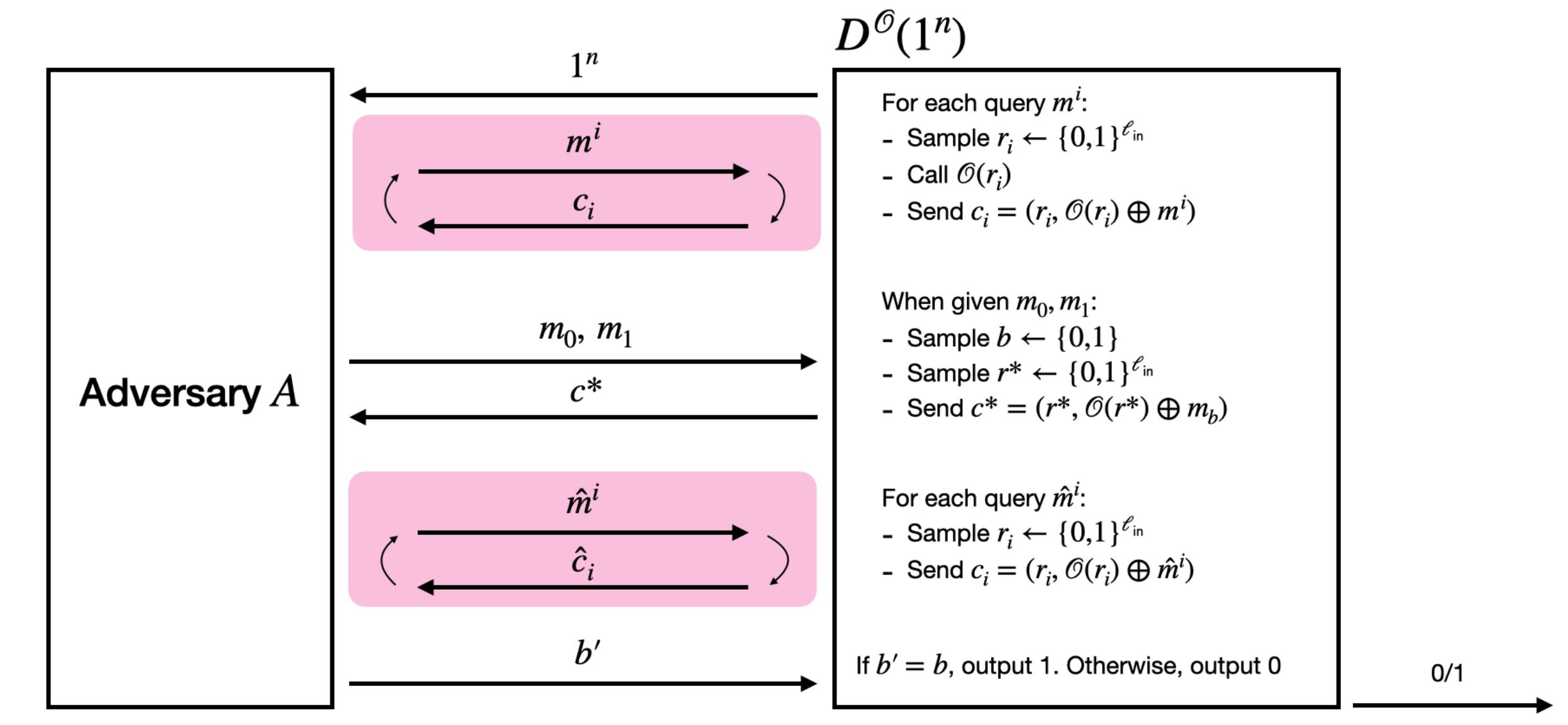
# Reduction for Lemma 1

**Case 1 (PRF world):**  $\mathcal{O} \leftarrow \{F_k\}_{k \in \{0,1\}^n}$

$$\Pr_{k \leftarrow \{0,1\}^n} [D^{F_k}(1^n) = 1] = \Pr [\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1]$$

**Case 2 (random world):**  $\mathcal{O} \leftarrow \mathcal{F}$

$$\Pr_{f \leftarrow \mathcal{F}_n} [D^f(1^n) = 1] = \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1]$$



**Putting this together:**

$$\left| \Pr_{k \leftarrow \{0,1\}^n} [D^{F_k}(1^n) = 1] - \Pr_{f \leftarrow \mathcal{F}_n} [D^f(1^n) = 1] \right| = \left| \Pr [\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1] - \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1] \right|$$

# Proving Lemma 1

**Lemma 1:** For all PPT  $A$ , there exists a negligible function  $\epsilon_1(\cdot)$  s.t.

$$|\Pr[\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1] - \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1]| \leq \epsilon_1(n)$$

**Proof:** Let  $A$  be any PPT CPA adversary. We will construct a distinguisher  $D$  that uses  $A$  to try to break the PRF security of  $F$  (i.e., distinguish  $F$  from random)

Since  $F$  is a secure PRF and  $D$  a PPT algorithm, then there exists a negligible function  $\epsilon_1(\cdot)$  such that

$$|\Pr[\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1] - \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1]| \leq \epsilon_1(n)$$

# Proving Lemma 2

**Lemma 2:** For all PPT  $A$ , there exists a negligible function  $\epsilon_2(\cdot)$  s.t.

$$\Pr[\text{PrivK}_{\hat{\Pi}, A}^{\text{CPA}}(n) = 1] \leq \frac{1}{2} + \epsilon_2(n)$$

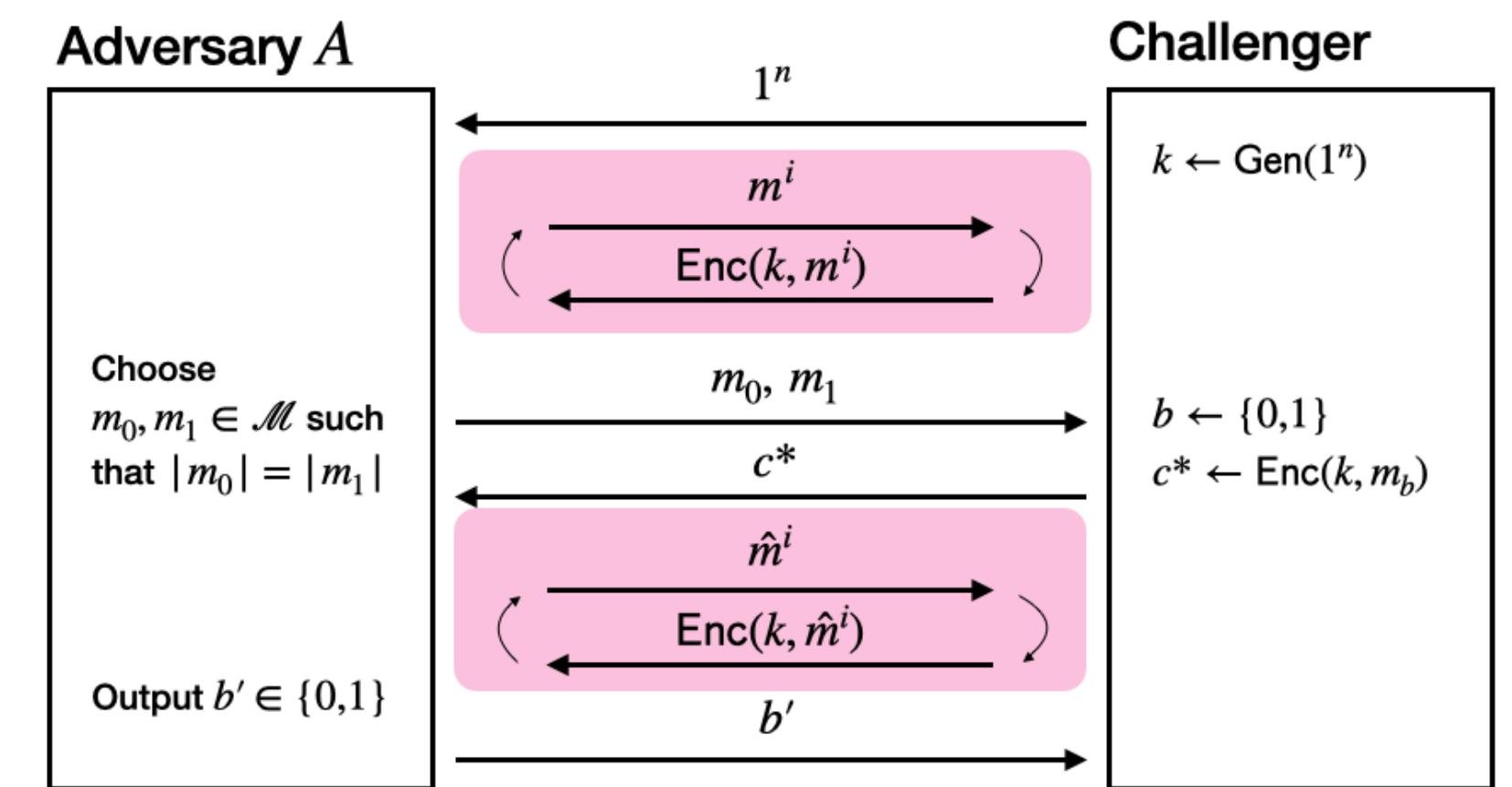
**Proof sketch:** Recall the CPA security game.

In  $\hat{\Pi}$  each encryption query is answered with

$(r_i, f(r_i) \oplus m^i)$  where  $f$  is a random function and  $r_i \leftarrow \{0,1\}^n$ .

As long as the  $r^*$  used for  $c^*$  was *not* used in any

of the oracle queries, then  $f(r^*)$  is uniform and independent of  $A$ 's view.



# Proving Lemma 2

## Proof continued:

Let  $q(n)$  be the bound on the number of queries made by  $A$  to the encryption oracle.

Let **Repeat** be the event in which  $r^*$  was used at least once by the encryption oracle (i.e., exists some  $i$  s.t.  $r^* = r_i$ ). Note that each  $r_i$  is chosen uniformly at random from  $\{0,1\}^n$

$$\begin{aligned}\Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1] &= \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \overline{\text{Repeat}}] \cdot \Pr[\overline{\text{Repeat}}] \\ &\quad + \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \text{Repeat}] \cdot \Pr[\text{Repeat}]\end{aligned}$$

# Proving Lemma 2

## Proof continued:

Let  $q(n)$  be the bound on the number of queries made by  $A$  to the encryption oracle.

Let **Repeat** be the event in which  $r^*$  was used at least once by the encryption oracle (i.e., exists some  $i$  s.t.  $r^* = r_i$ ). Note that each  $r_i$  is chosen uniformly at random from  $\{0,1\}^n$

$$\begin{aligned}\Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1] &= \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \overline{\text{Repeat}}] \cdot \Pr[\overline{\text{Repeat}}] \\ &\quad + \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \text{Repeat}] \cdot \Pr[\text{Repeat}] \\ &\leq \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \overline{\text{Repeat}}] + \Pr[\text{Repeat}]\end{aligned}$$

By the law of total probability:

$$P[A] = \Pr[A \mid \overline{B}] \cdot \Pr[\overline{B}] + \Pr[A \mid B] \cdot \Pr[B] \leq \Pr[A \mid \overline{B}] + \Pr[B]$$

# Proving Lemma 2

## Proof continued:

Let  $q(n)$  be the bound on the number of queries made by  $A$  to the encryption oracle.

Let Repeat be the event in which  $r^*$  was used at least once by the encryption oracle (i.e., exists some  $i$  s.t.  $r^* = r_i$ ). Note that each  $r_i$  is chosen uniformly at random from  $\{0,1\}^n$

$$\begin{aligned}\Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1] &= \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \overline{\text{Repeat}}] \cdot \Pr[\overline{\text{Repeat}}] \\ &\quad + \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \text{Repeat}] \cdot \Pr[\text{Repeat}]\end{aligned}$$

$$\leq \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \overline{\text{Repeat}}] + \Pr[\text{Repeat}]$$

What's the probability  $\hat{\Pi}$  wins when  $r^*$  is not used by the encryption oracle?

What's the probability we sample the same  $r^*$  that's used in an oracle query?

# Proving Lemma 2

## Proof continued:

Let  $q(n)$  be the bound on the number of queries made by  $A$  to the encryption oracle.

Let **Repeat** be the event in which  $r^*$  was used at least once by the encryption oracle  
(i.e., exists some  $i$  s.t.  $r^* = r_i$ )

$$\begin{aligned}\Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1] &= \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \overline{\text{Repeat}}] \cdot \Pr[\overline{\text{Repeat}}] \\ &\quad + \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \text{Repeat}] \cdot \Pr[\text{Repeat}] \\ &\leq \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \overline{\text{Repeat}}] + \Pr[\text{Repeat}] \\ &\leq \frac{1}{2} + \frac{q(n)}{2^n}\end{aligned}$$

# Proving Lemma 2

## Proof continued:

Let  $q(n)$  be the bound on the number of queries made by  $A$  to the encryption oracle.

Let Repeat be the event in which  $r^*$  was used at least once by the encryption oracle  
(i.e., exists some  $i$  s.t.  $r^* = r_i$ )

$$\begin{aligned}\Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1] &= \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \overline{\text{Repeat}}] \cdot \Pr[\overline{\text{Repeat}}] \\ &\quad + \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \text{Repeat}] \cdot \Pr[\text{Repeat}] \\ &\leq \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1 \mid \overline{\text{Repeat}}] + \Pr[\text{Repeat}] \\ &\leq \frac{1}{2} + \frac{q(n)}{2^n}\end{aligned}$$

This term is negligible

# Lemma 1 + Lemma 2

**Lemma 1:** For all PPT  $A$ , there exists a negligible function  $\epsilon_1(\cdot)$  s.t.

$$|\Pr[\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1] - \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1]| \leq \epsilon_1(n)$$

**Lemma 2:** For all PPT  $A$ , there exists a negligible function  $\epsilon_2(\cdot)$  s.t.

$$\Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1] \leq \frac{1}{2} + \epsilon_2(n)$$

Putting them together we have for all PPT  $A$ ,

$$\begin{aligned} \Pr[\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1] &\leq |\Pr[\text{PrivK}_{\Pi,A}^{\text{CPA}}(n) = 1] - \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1]| + \Pr[\text{PrivK}_{\hat{\Pi},A}^{\text{CPA}}(n) = 1] \\ &\leq \epsilon_1(n) + \frac{1}{2} + \epsilon_2(n) \end{aligned}$$

Because  $\epsilon_1(n) + \epsilon_2(n)$  is negligible, we have that  $\Pi$  is CPA-secure.

# Hybrid Arguments

# How to Prove More Involved Constructions?

Our CPA-secure encryption was argued in two parts:

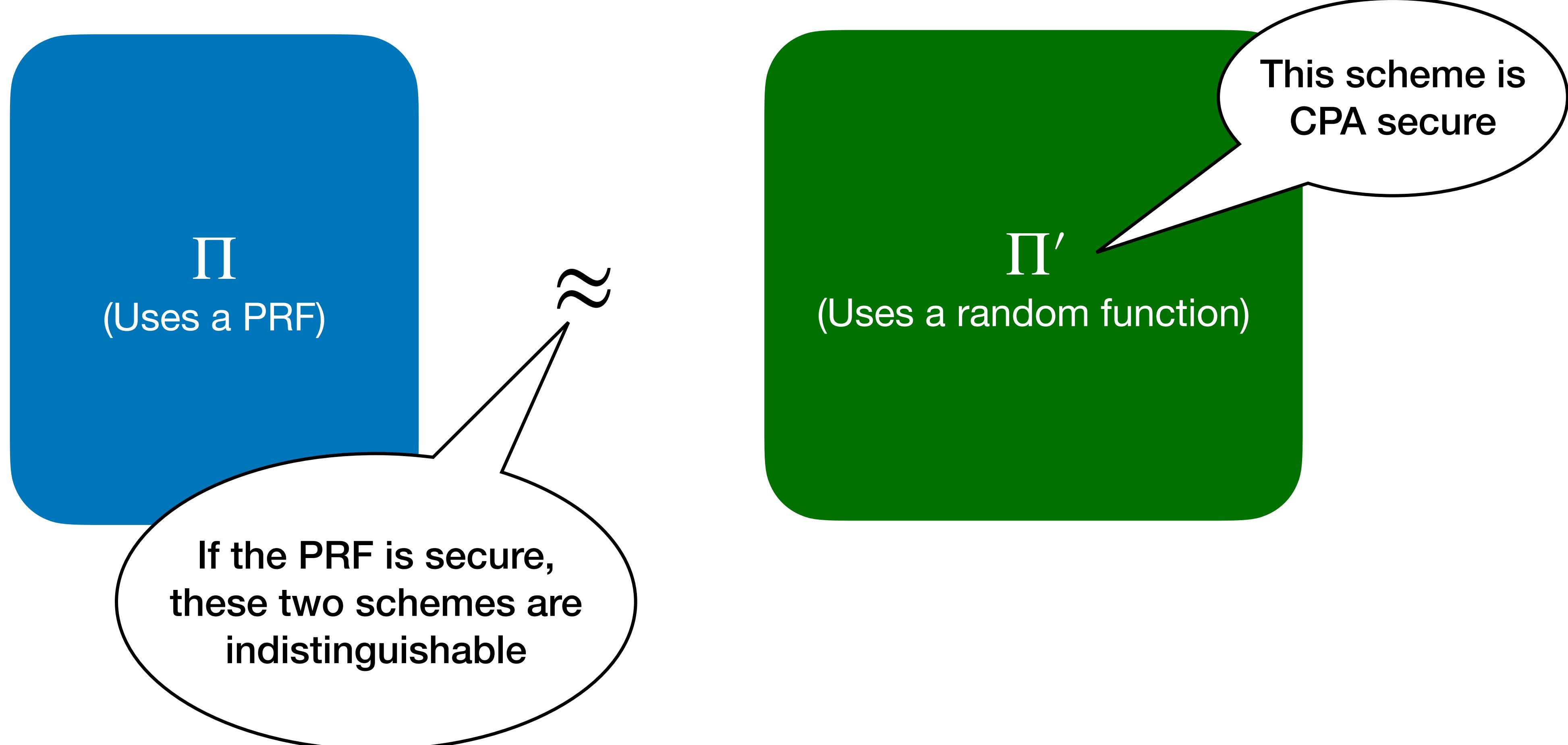
1. The version with the PRF and the version with the random function are indistinguishable
2. The version with the random function is secure

What if our scheme used more than one PRF? Or used PRFs *and* PRGs?

More generally, how do we prove something with lots of moving parts?

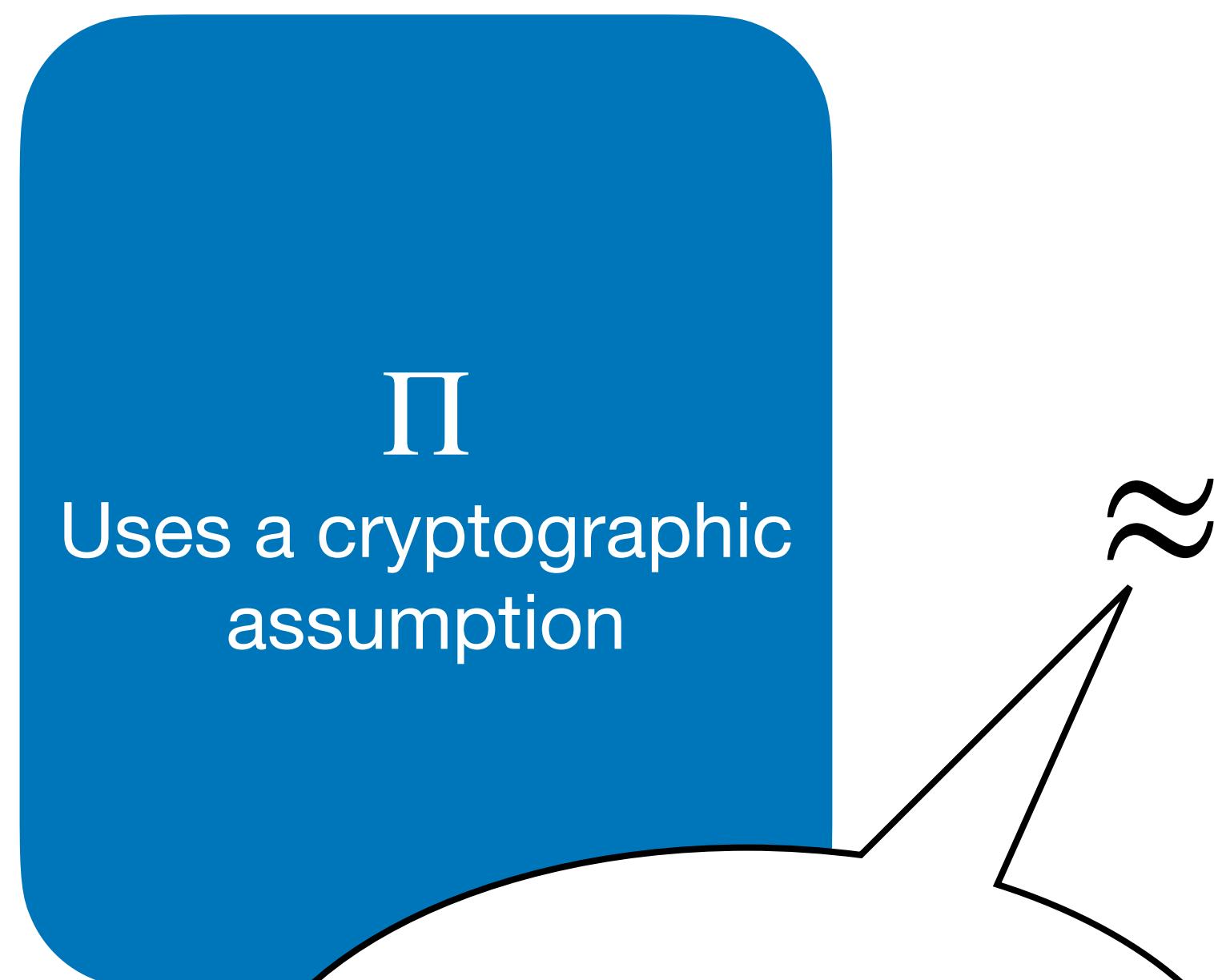
# How to Prove More Involved Constructions?

Let's look at how the CPA-security proof for our PRF-based scheme worked:

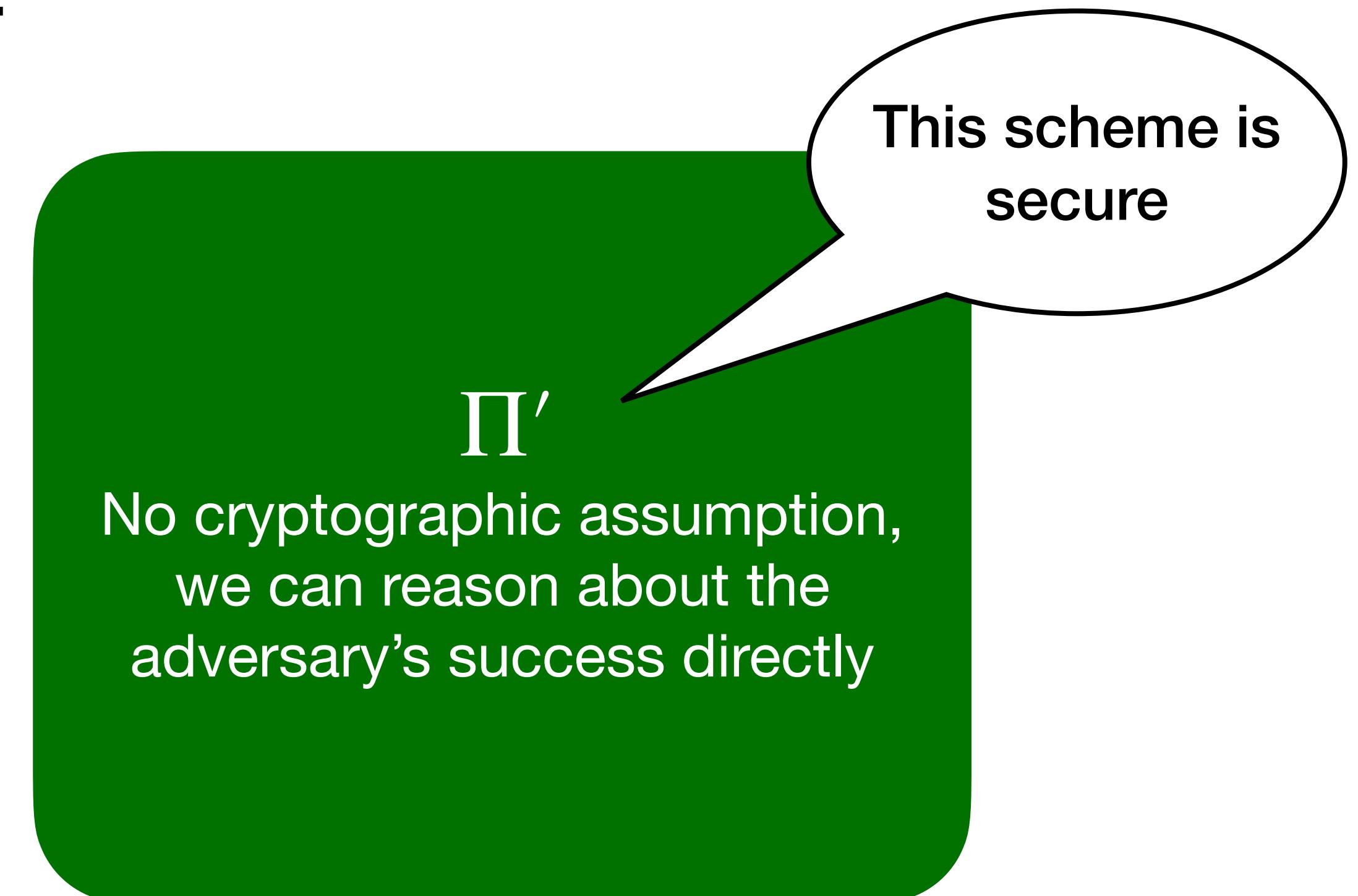


# How to Prove More Involved Constructions?

A more general view of our previous proof:



If the assumption holds,  
these two schemes are  
indistinguishable



# How to Prove More Involved Constructions?

Protocols that uses cryptographic primitives in more than one place

$\Pi$

Uses cryptographic assumptions

?

$\Pi'$

No cryptographic assumptions

# How to Prove More Involved Constructions?

Protocols that uses cryptographic primitives in more than one place

$\Pi$

Uses cryptographic assumptions

$\Pi''$

Version of  $\Pi$  with one of the cryptographic components replaced

$\Pi'$

No cryptographic assumptions

# How to Prove More Involved Constructions?

Protocols that uses cryptographic primitives in more than one place

$\Pi$

Uses cryptographic assumptions

$\Pi''$

Version of  $\Pi$  with one of the cryptographic components replaced

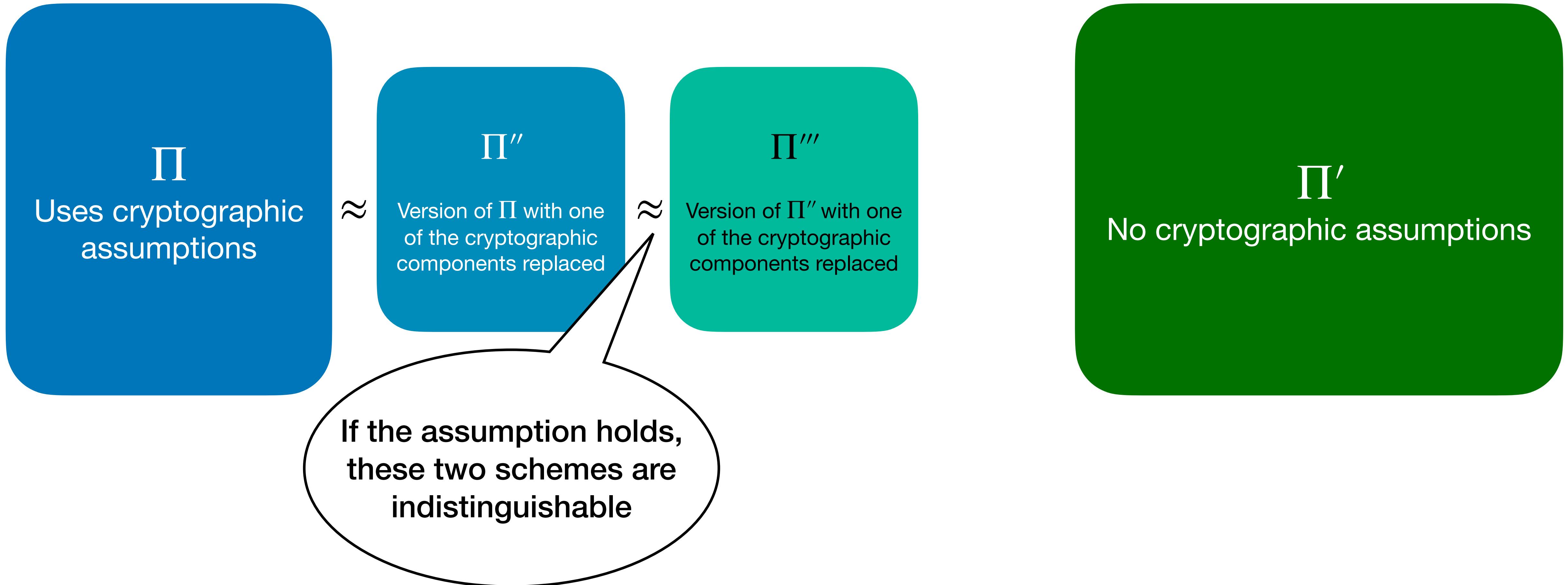
If the assumption holds,  
these two schemes are  
indistinguishable

$\Pi'$

No cryptographic assumptions

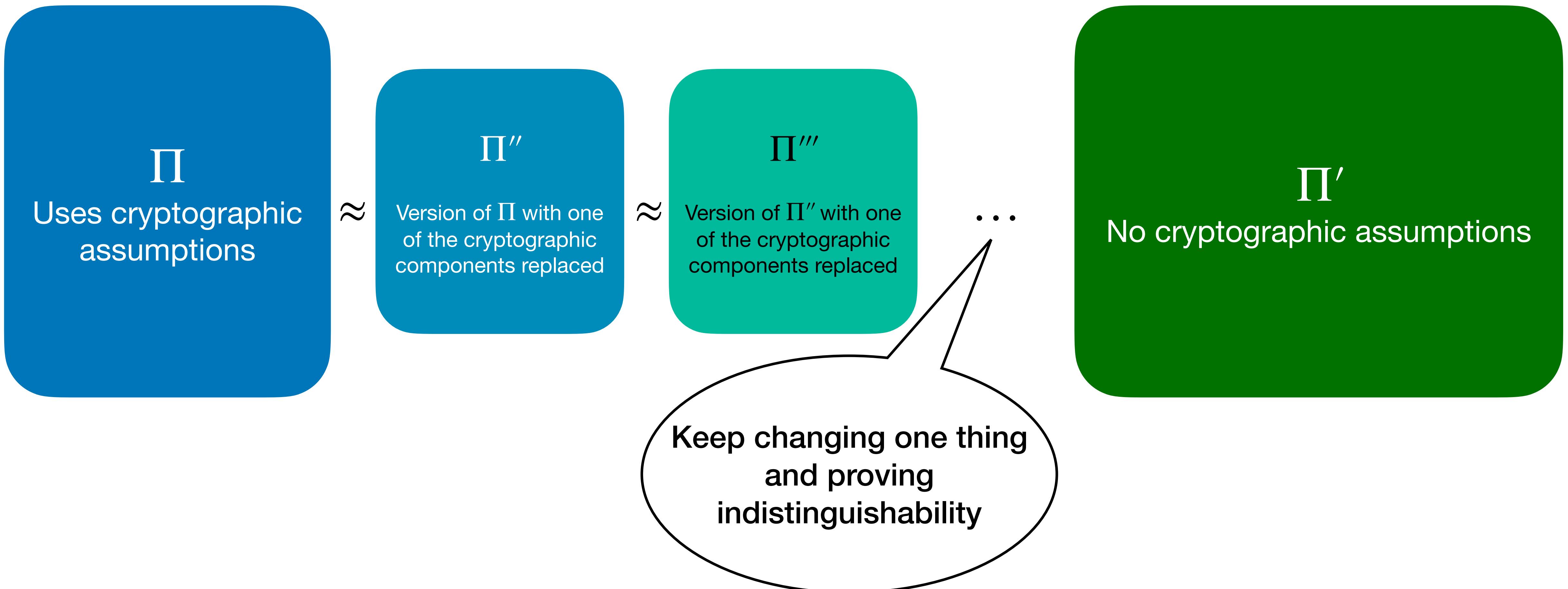
# How to Prove More Involved Constructions?

Protocols that uses cryptographic primitives in more than one place



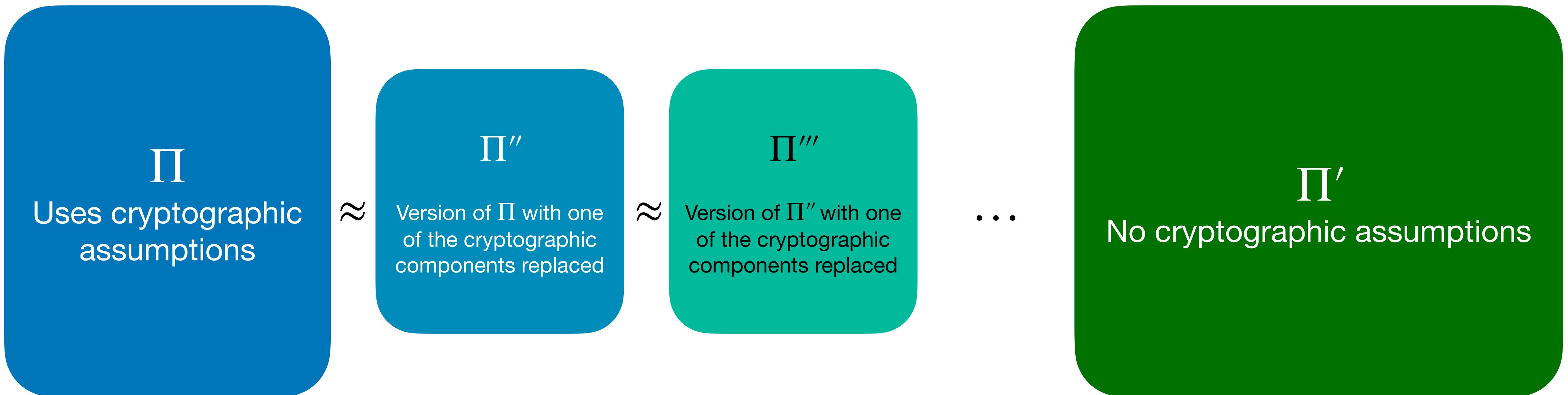
# How to Prove More Involved Constructions?

Protocols that uses cryptographic primitives in more than one place



# How to Prove More Involved Constructions?

Protocols that uses cryptographic primitives in more than one place

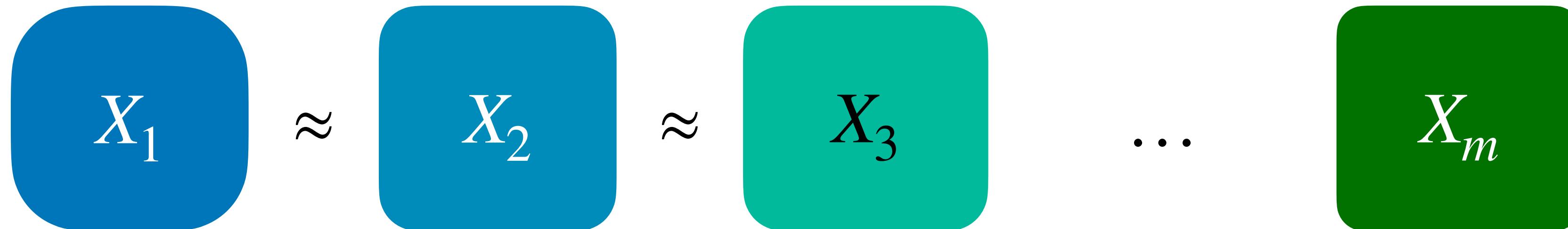


This is the idea behind a **hybrid argument**

# Hybrid Arguments

Computational indistinguishability is transitive (to up polynomially many distributions)

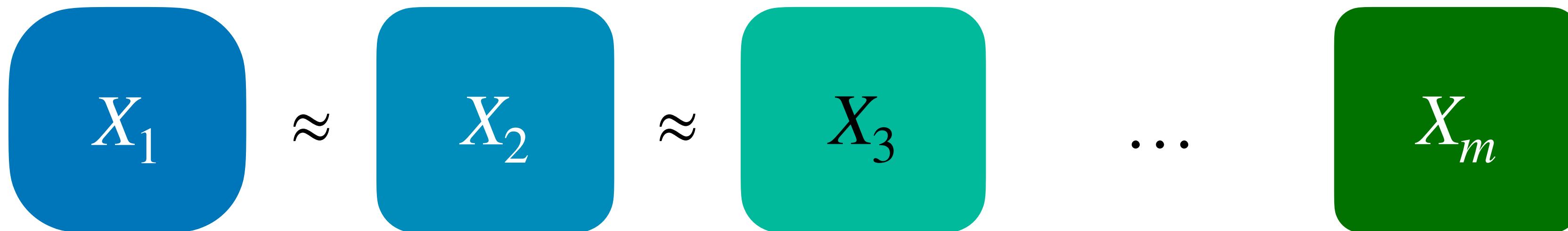
- If we want to prove that two distributions are indistinguishable, we can insert (polynomially many) **hybrid distributions** between them and prove each pair is indistinguishable



# Hybrid Arguments

Computational indistinguishability is transitive (to up polynomially many distributions)

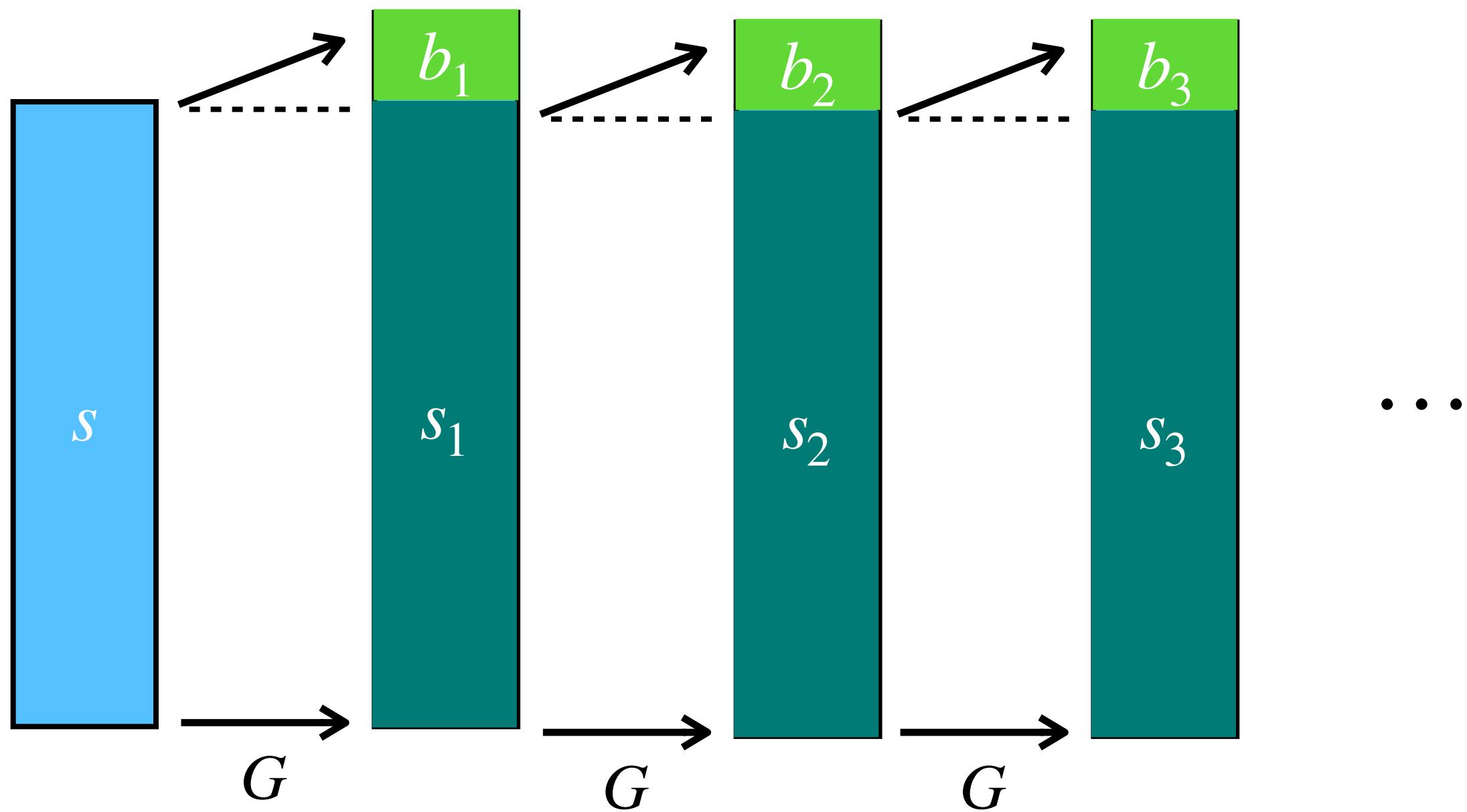
- If we want to prove that two distributions are indistinguishable, we can insert (polynomially many) **hybrid distributions** between them and prove each pair is indistinguishable
- If an adversary can distinguish  $X_i$  from  $X_{i+1}$  with at most  $\epsilon$  probability, then an adversary can distinguish  $X_1$  from  $X_m$  with at most  $m \cdot \epsilon$  probability
  - If  $\epsilon$  is negligible and  $m$  is a polynomial, this product is negligible



# Recall: Increasing the Stretch of a PRG

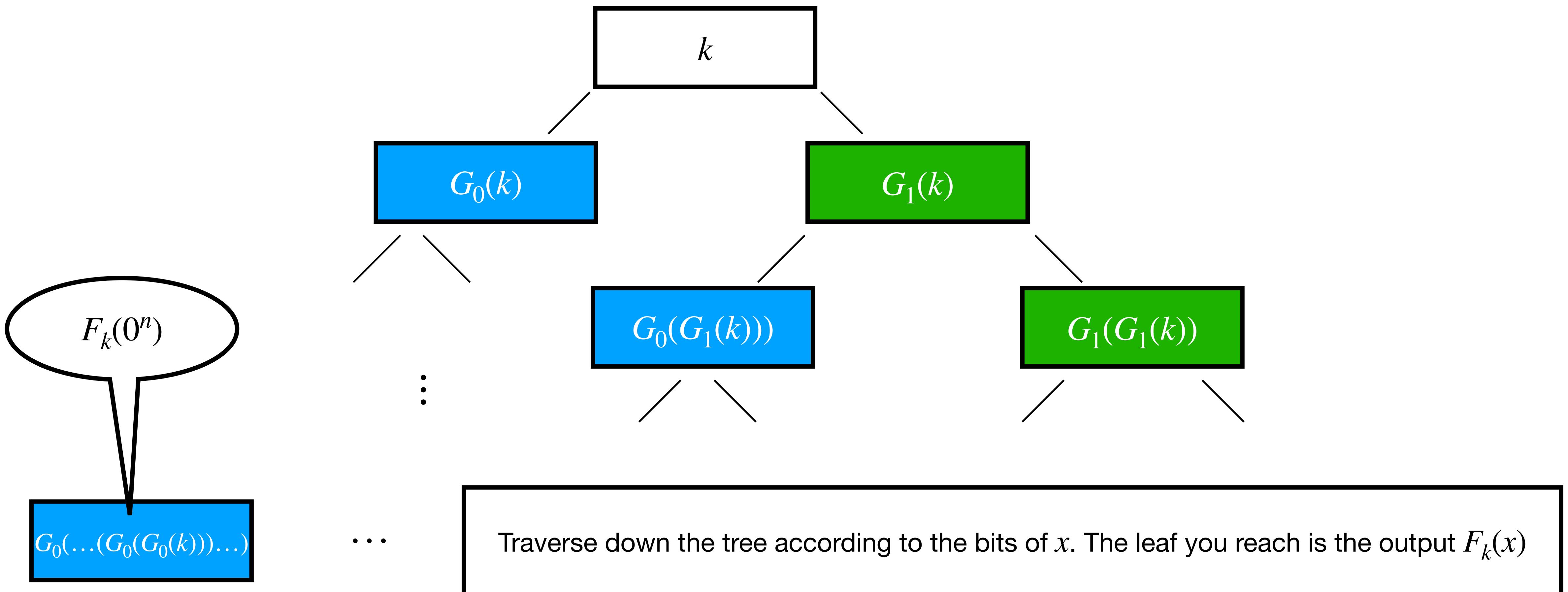
**Theorem:** If there exists a PRG with 1-bit expansion (i.e.,  $\ell(n) = n + 1$ ), then for any polynomial  $p(n)$ , there exists a PRG with  $p(n)$ -bit expansion

**Proof sketch:**



# Recall: GGM Construction

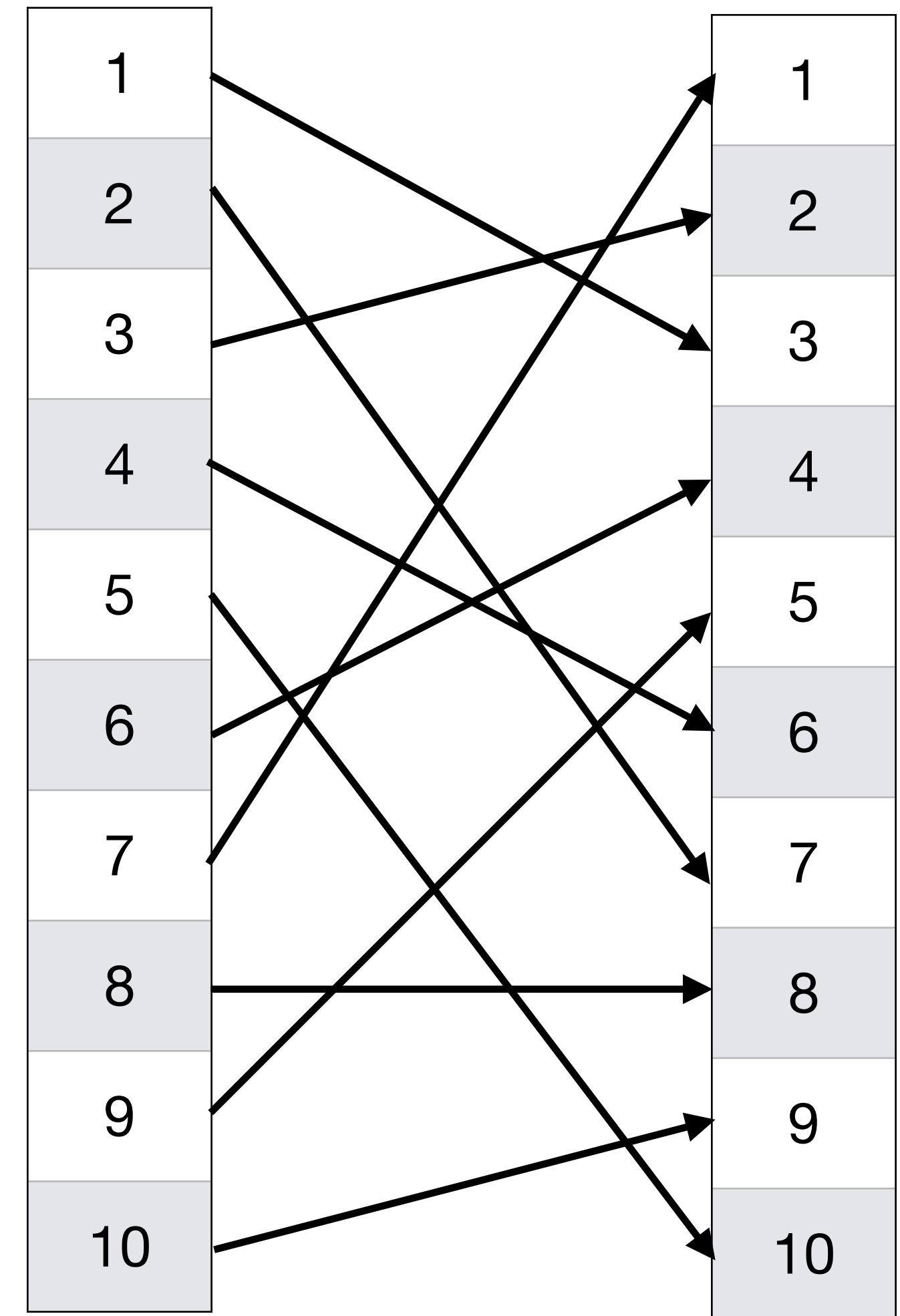
$$F_k(x) = G_{x_n}(\dots(G_{x_2}(G_{x_1}(k)))\dots)$$



# Pseudorandom Permutations (PRPs)

# Permutations

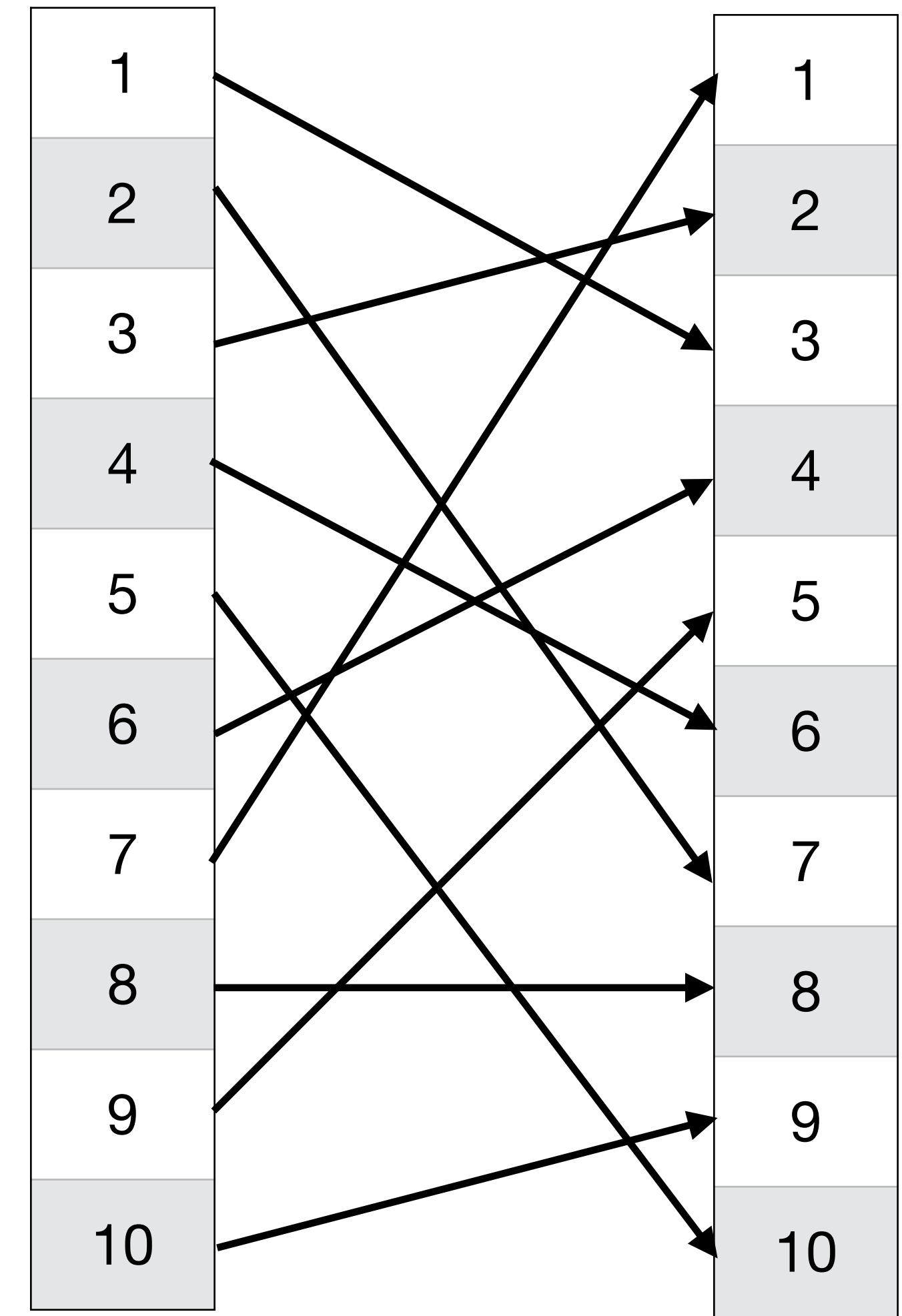
- A **permutation** is a function  $f$  from a finite set to itself that is 1-1 and onto (**bijection**)
  - The inverse  $f^{-1}$  is well defined
- We will consider permutations over  $\{0,1\}^\ell$



Please pretend these numbers are from  $\{0,1\}^\ell$

# Permutations

- A **permutation** is a function  $f$  from a finite set to itself that is 1-1 and onto (**bijection**)
  - The inverse  $f^{-1}$  is well defined
- Recall  $\mathcal{F}_n$  is the set of all possible functions from  $\{0,1\}^n \rightarrow \{0,1\}^n$ 
  - We will denote  $\mathcal{P}_n$  as the set of all possible permutations from  $\{0,1\}^n \rightarrow \{0,1\}^n$



Please pretend these numbers are from  $\{0,1\}^n$

# Pseudorandom Permutations (PRPs)

**Definition (PRP):**

$F$  is a pseudorandom permutation (PRP) if  $F$  is a PRF and

for all  $k \in \{0,1\}^n$ ,  $F_k : \{0,1\}^{\ell(n)} \rightarrow \{0,1\}^{\ell(n)}$  is a bijection (i.e.,  $F_k^{-1}$  exists),

and  $F_k^{-1}$  is efficiently computable given  $k$

(i.e., there is a polynomial-time algorithm that given  $k, y$  computes  $F_k^{-1}(y)$ )

# Strong Pseudorandom Permutations (PRPs)

**Definition (*Strong* PRP):**

$F$  is a *strong pseudorandom permutation* if  $F$  is a PRP and for all PPT  $D$  there exists a negligible function  $\epsilon(\cdot)$  such that

$$\left| \Pr_{k \leftarrow \{0,1\}^n} [D^{F_k, F_k^{-1}}(1^n) = 1] - \Pr_{f \leftarrow \mathcal{P}_n} [D^{f, f^{-1}}(1^n) = 1] \right| \leq \epsilon(n)$$

# Strong Pseudorandom Permutations (PRPs)

**Definition (*Strong* PRP):**

$F$  is a *strong pseudorandom permutation* if  $F$  is a PRP and for all PPT  $D$  there exists a negligible function  $\epsilon(\cdot)$  such that

$$\left| \Pr_{k \leftarrow \{0,1\}^n} [D^{F_k, F_k^{-1}}(1^n) = 1] - \Pr_{f \leftarrow \mathcal{P}_n} [D^{f, f^{-1}}(1^n) = 1] \right| \leq \epsilon(n)$$

Pseudorandomness holds even if the distinguisher has access to  $F_k$  and  $F_k^{-1}$

# Practical Heuristics

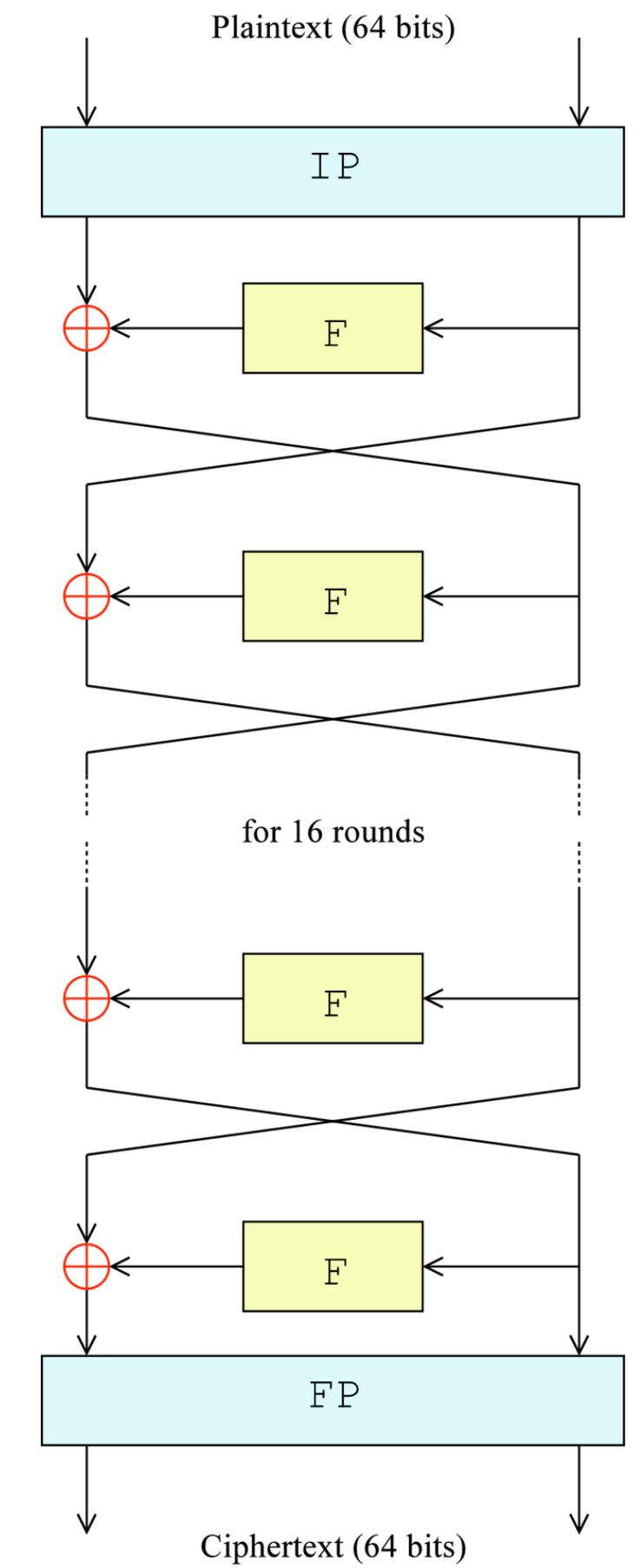
# Practical Heuristics: Block Ciphers

- Block ciphers typically refers to practical constructions of strong PRPs
  - Described in terms of *concrete* values rather than asymptotics (i.e., input and output length are fixed rather than a function of the security parameter)
  - In practice, PRFs/PRPs in protocols are instantiated with block ciphers

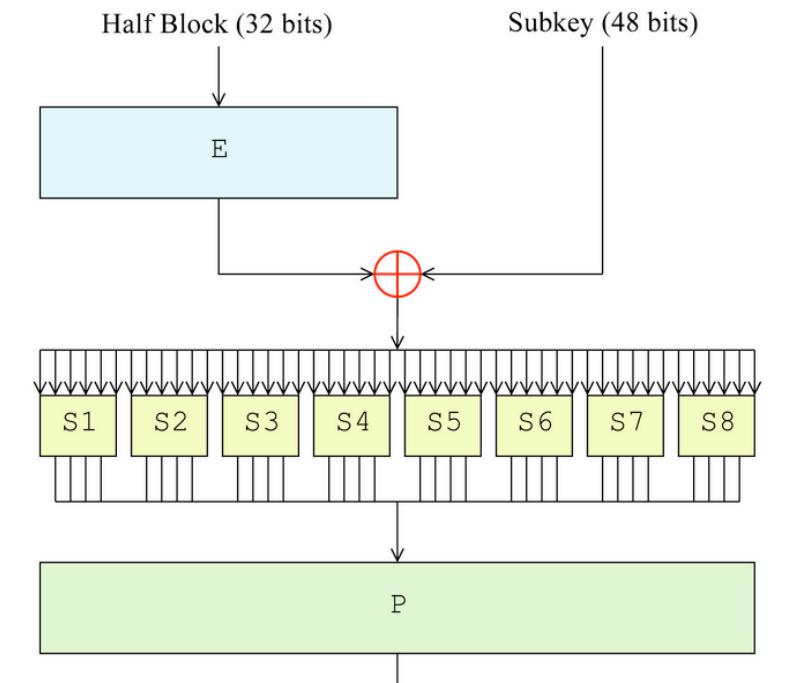
We'll look at a few well-known block ciphers but won't go into too much detail

# Block Ciphers: DES

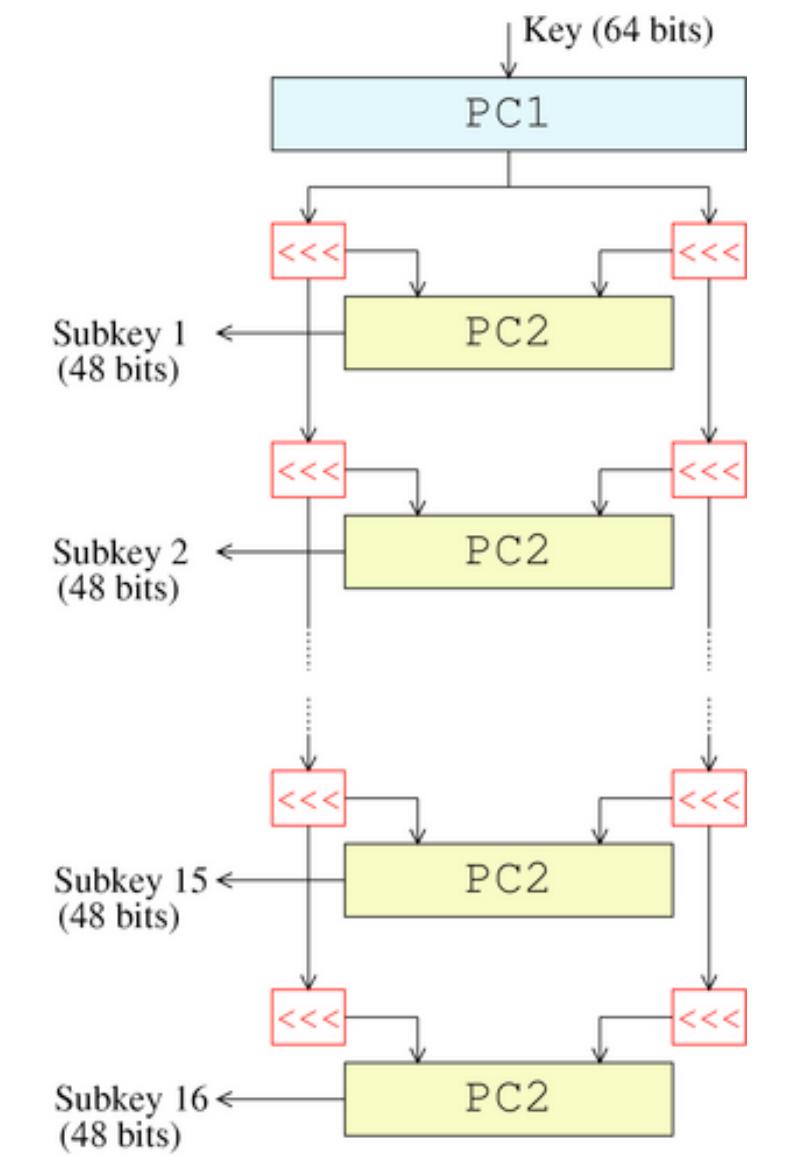
- The Data Encryption Standard (DES) was developed in the 1970s by IBM (with help from the NSA)
  - Adopted in 1977 as a Federal Information Processing Standard (FIPS)
  - Key length is 56-bits and block length (length of its input and output) is 64-bits



Overall structure of DES



Function F used in DES



Generating the subkeys for F

# Block Ciphers: DES

- The Data Encryption Standard (DES) was developed in the 1970s by IBM (with help from the NSA)
  - Adopted in 1977 as a Federal Information Processing Standard (FIPS)
  - Key length is 56-bits and block length (length of its input and output) is 64-bits
  - Best known attack in practice is essentially a brute-force key search ( $\approx 2^{56}$ )
    - However, no longer considered secure due to its short key length
    - Brute force attacks in running time  $2^{56}$  are feasible today

# Block Ciphers: 3DES

- Triple-DES (3DES) was standardized by NIST and was widely used in practice after DES

$$3\text{DES}_{k_1, k_2, k_3}(x) = \text{DES}_{k_1}(\text{DES}_{k_2}^{-1}(\text{DES}_{k_3}(x)))$$

- The main drawbacks were that the keys are  $(3 \cdot 56)$ -bits but can be broken in time  $2^{2 \cdot 56}$  and it is slower than DES
- 3DES stopped being recommended by NIST in 2024
- Use in practice has mostly been replaced by AES

# Block Ciphers: AES

- In 1997 NIST published a call for candidate block ciphers to replace DES
  - The winner would be called the **Advanced Encryption Standard (AES)**
  - 15 candidates were proposed by different teams from around the world
    - Each candidate was extensively analyzed by the public and other teams
  - The winner (“Rijndael”, named after its designers Vincent Rijmen and Joan Daemen) was announced in late 2000
    - NIST conceded that any of the 5 finalists would have been choices, as no serious security flaws were found in any of them. Rijndael was chosen based on properties like efficiency, performance in hardware, flexibility.
  - Key length is 128/192/256 bits and block length is 128 bits
  - To date, no known practical attacks better than brute-force key search

# Reminders

- PS 1 due on Thursday!
  - Any legible format is acceptable (typed or handwritten)
- PS 2 was released today
  - Due in 2 weeks on Thursday
- Mark's office hours are today in Milstein 503 from 4:30-6:30