COMS BC3262: Introduction to Cryptography

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

February 4, 2026

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

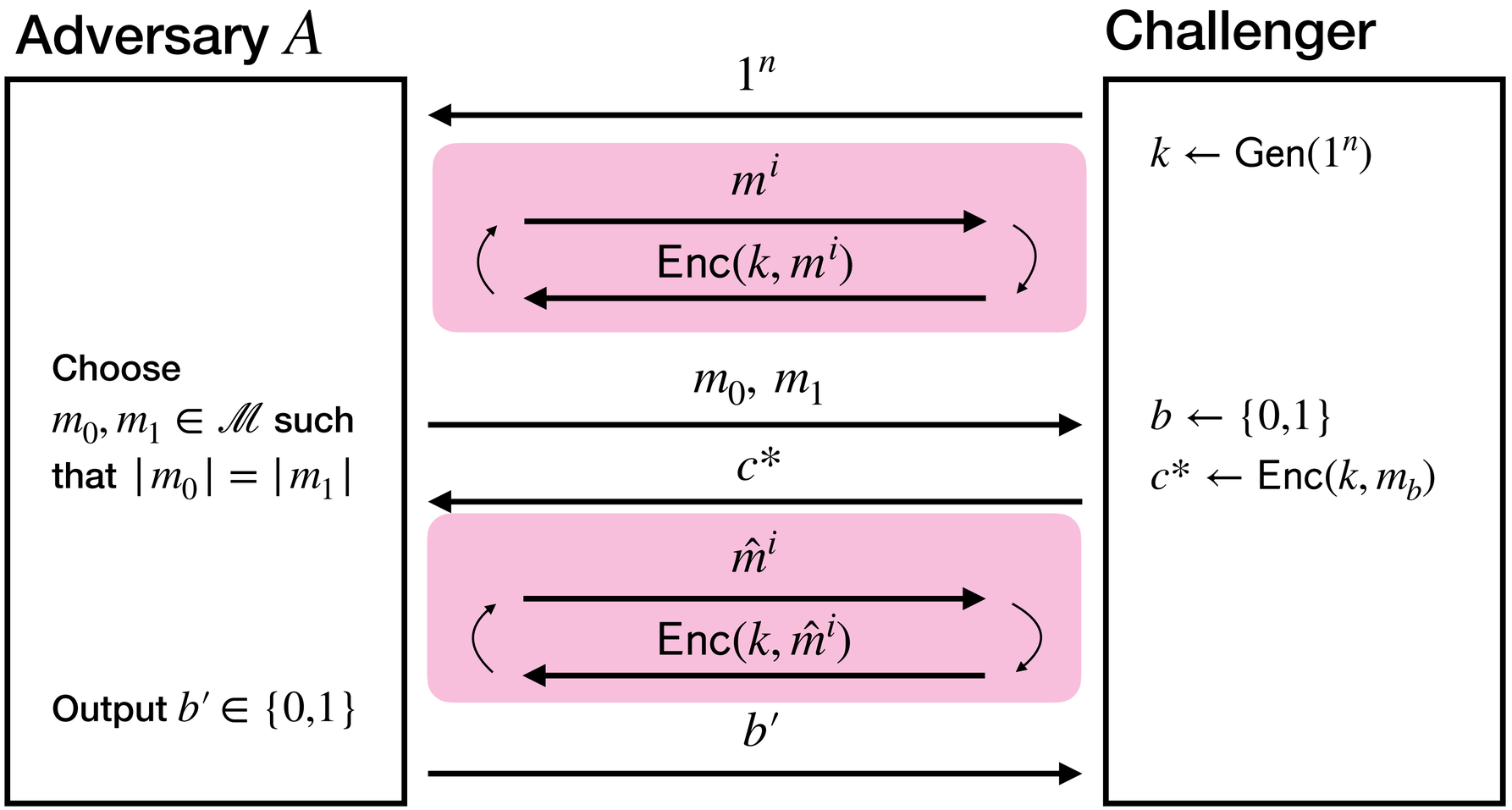
Please see EdStem for clarifications on some of the questions (mostly notation)

Today's Lecture

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

Chosen-Plaintext Attack (CPA)

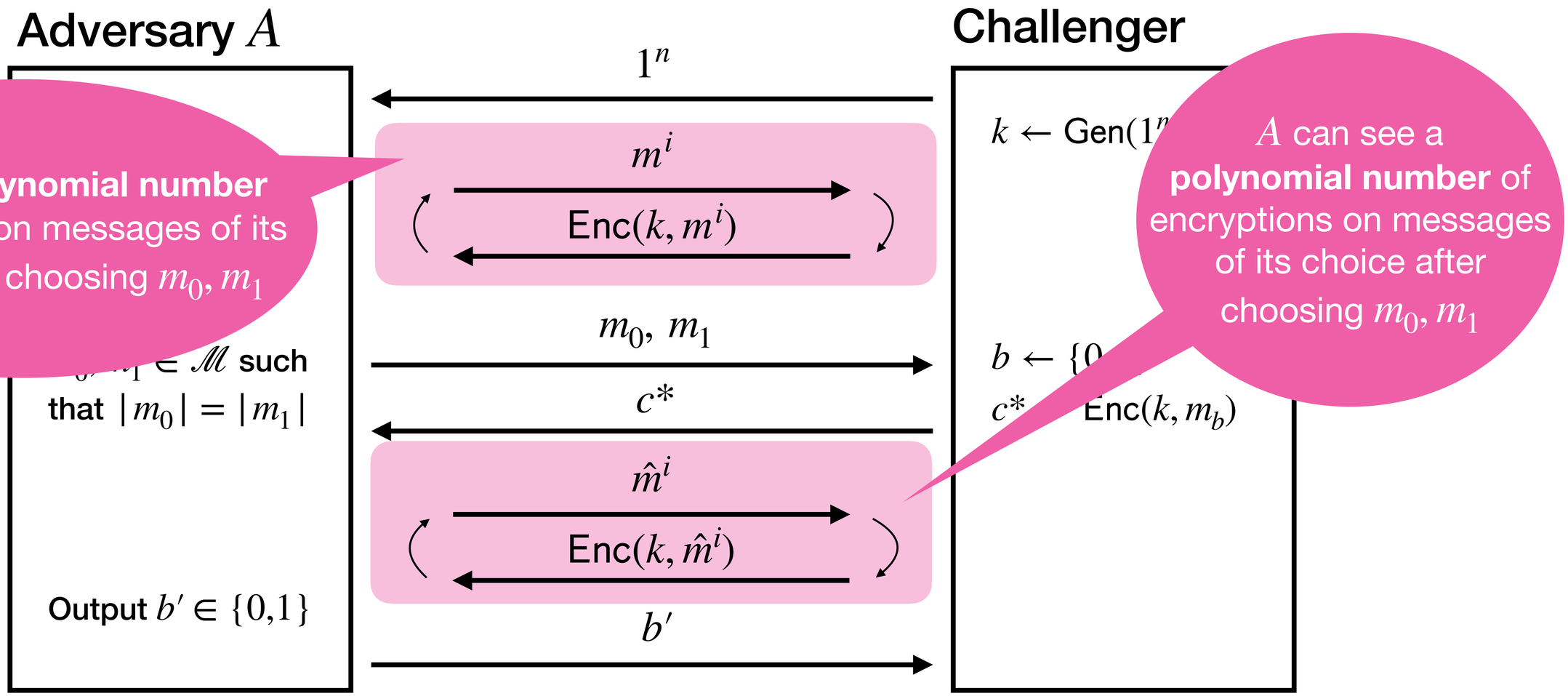
Chosen-Plaintext Attack (CPA)



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

and 0 otherwise

Chosen-Plaintext Attack (CPA)



A can see **polynomial number** of encryptions on messages of its choice before choosing m_0, m_1

A can see a **polynomial number** of encryptions on messages of its choice after choosing m_0, m_1

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

and 0 otherwise

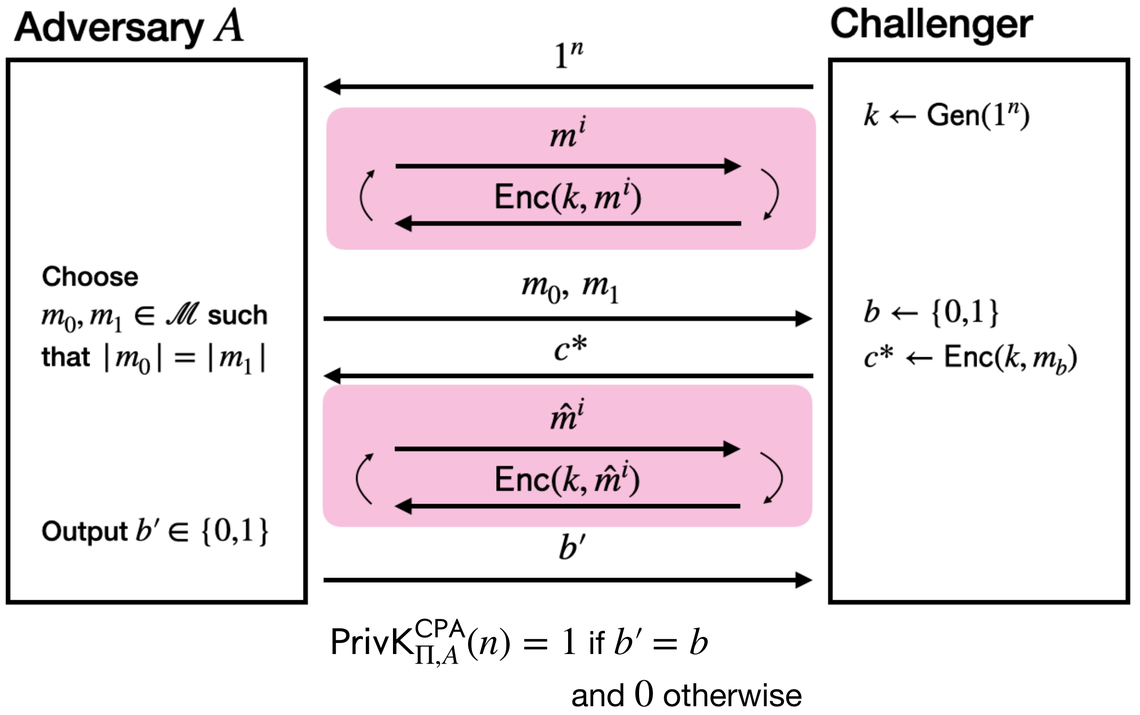
We typically refer to these types of queries as having “oracle access”

Chosen-Plaintext Attack (CPA)

Definition:

Π 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

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



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 poly-length key and a truly random function

- This is the idea behind a pseudorandom function (PRF)

Pseudorandom Functions (PRFs)

Definition:

Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ 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

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

Pseudorandom Functions (PRFs)

Definition:

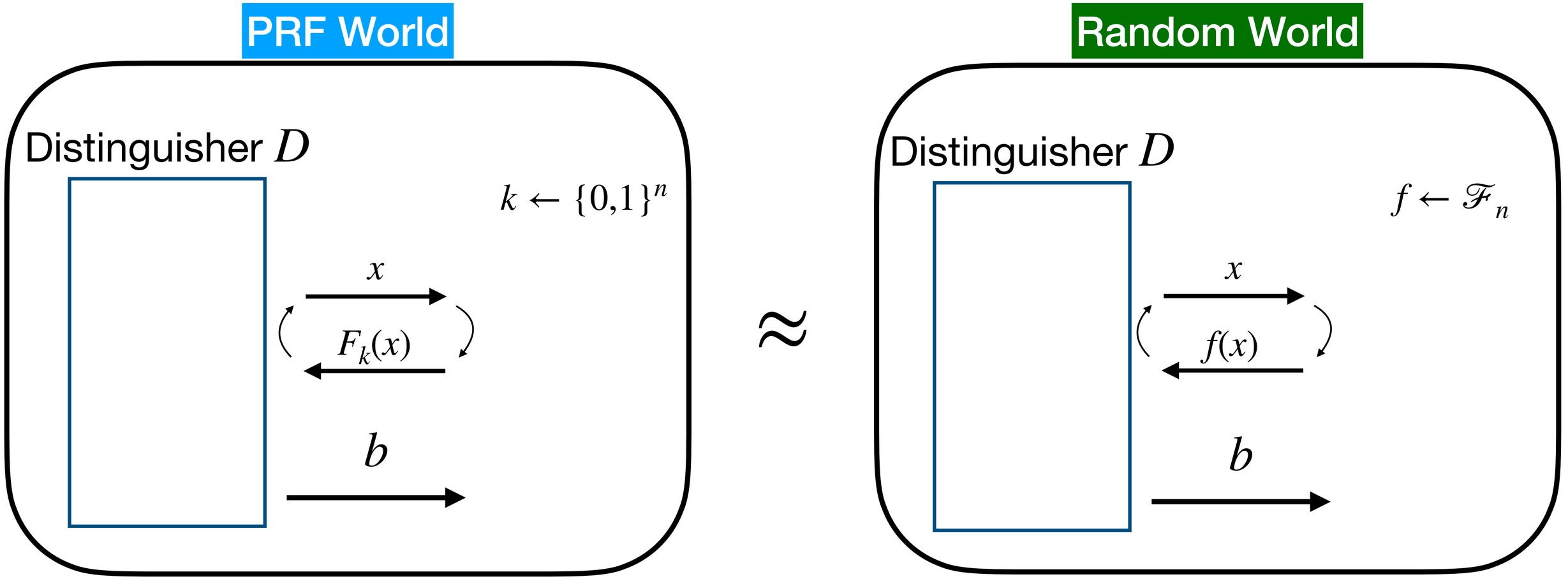
Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed function. We say that F is a pseudorandom function (PRF) if for all PPT D there exists a negligible function $\epsilon(\cdot)$ such that

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

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



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

Security Games for PRF vs PRG

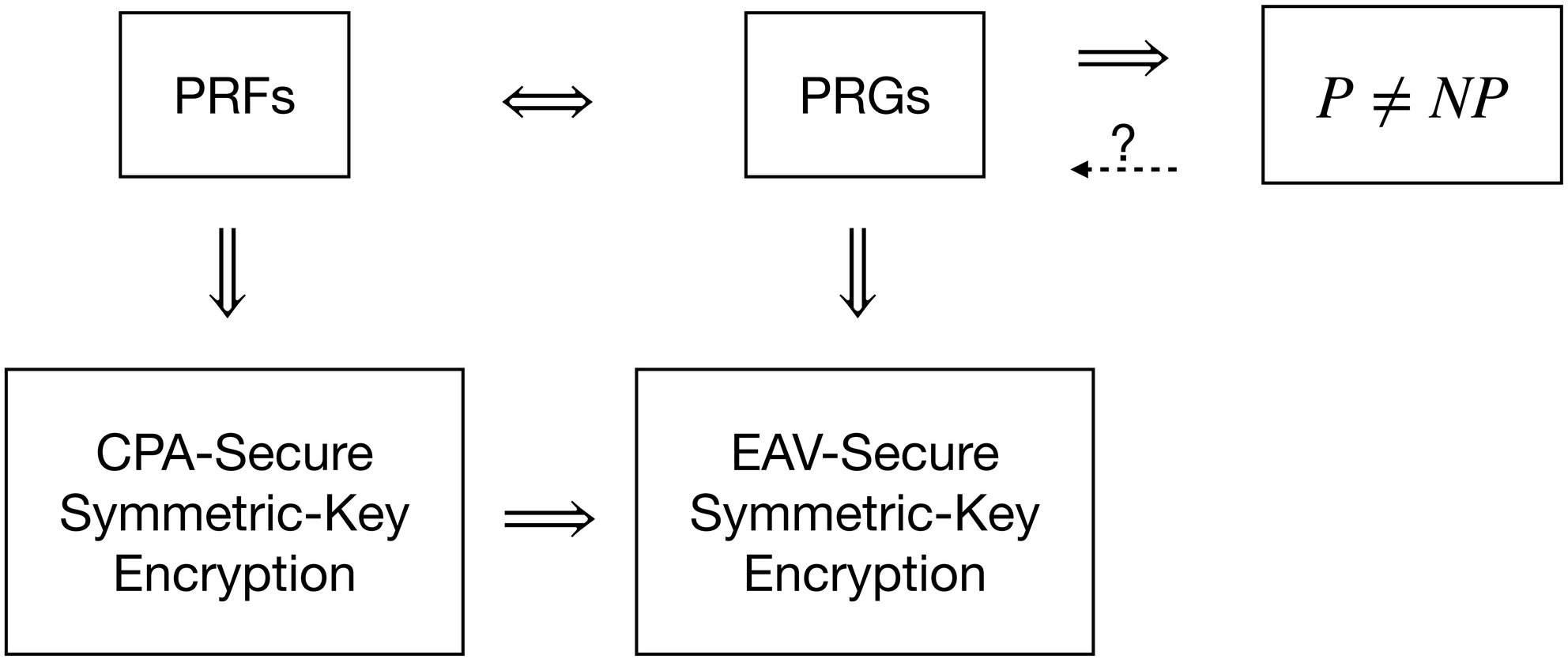
PRF:

- k is chosen uniformly at random (not known to D)
- D chooses (and knows) points the function will be evaluated on $(x_1, x_2, \ldots)$
  - D can see up to polynomially many $(x_1, f(x_1)), (x_2, f(x_2)), \ldots$
- 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 $F : \{0,1\}^n \times \{0,1\}^{\ell_{\text{in}}} \to \{0,1\}^{\ell_{\text{out}}}$ 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 Π is CPA-secure

Proof Idea

Let $F : \{0,1\}^n \times \{0,1\}^{\ell_{\text{in}}} \to \{0,1\}^{\ell_{\text{out}}}$ 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 Π 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 Π and $\hat{\Pi}$ are computationally indistinguishable.
  (intuitively: no PPT A playing in the CPA game can tell whether it's playing with Π or $\hat{\Pi}$)

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

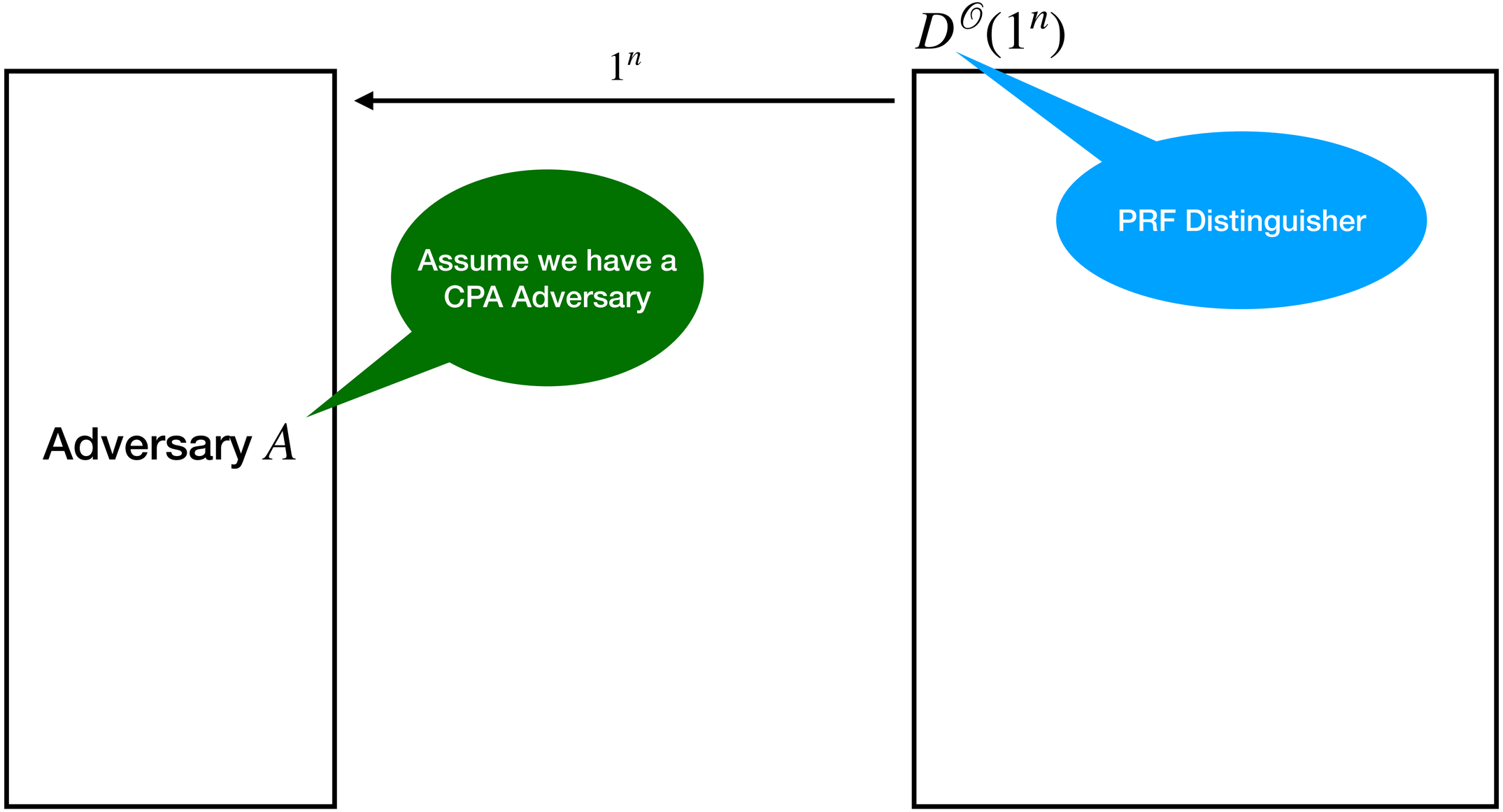
Proving Lemma 1

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

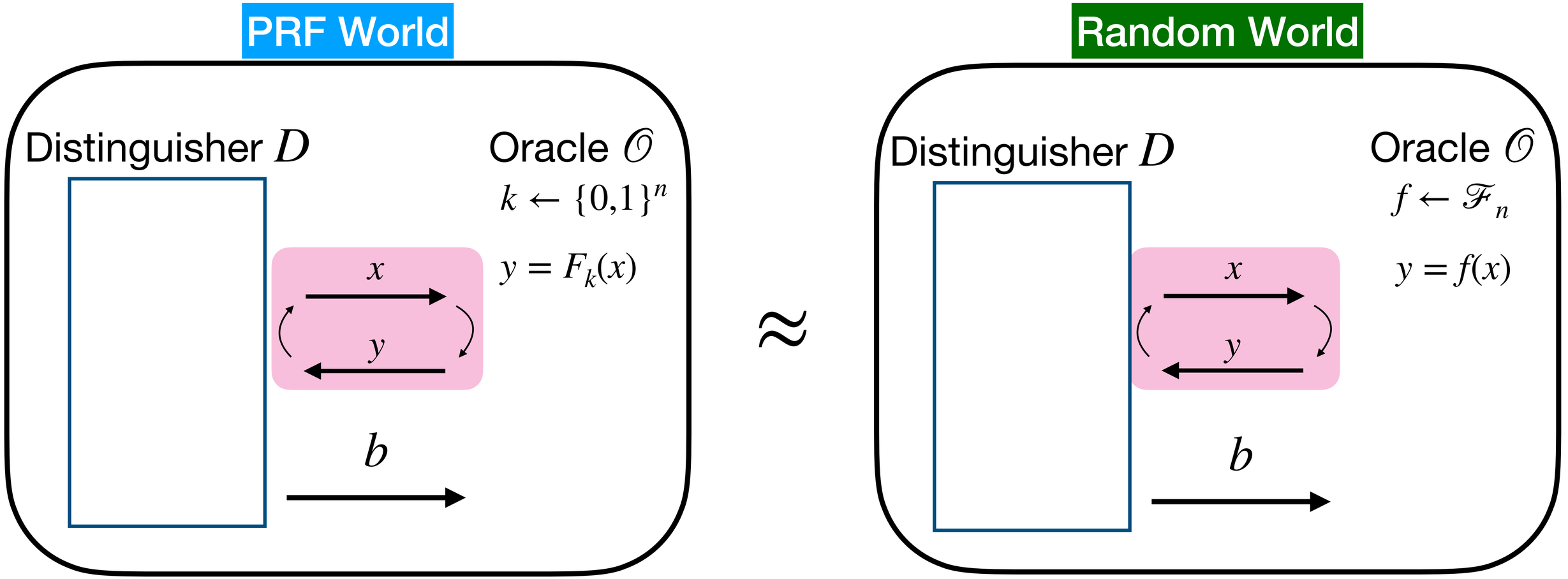
$$|\Pr[\text{PrivK}^{\text{CPA}}_{\Pi,A}(n) = 1] - \Pr[\text{PrivK}^{\text{CPA}}_{\hat{\Pi},A}(n) = 1]| \leq \epsilon_1(n)$$

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)

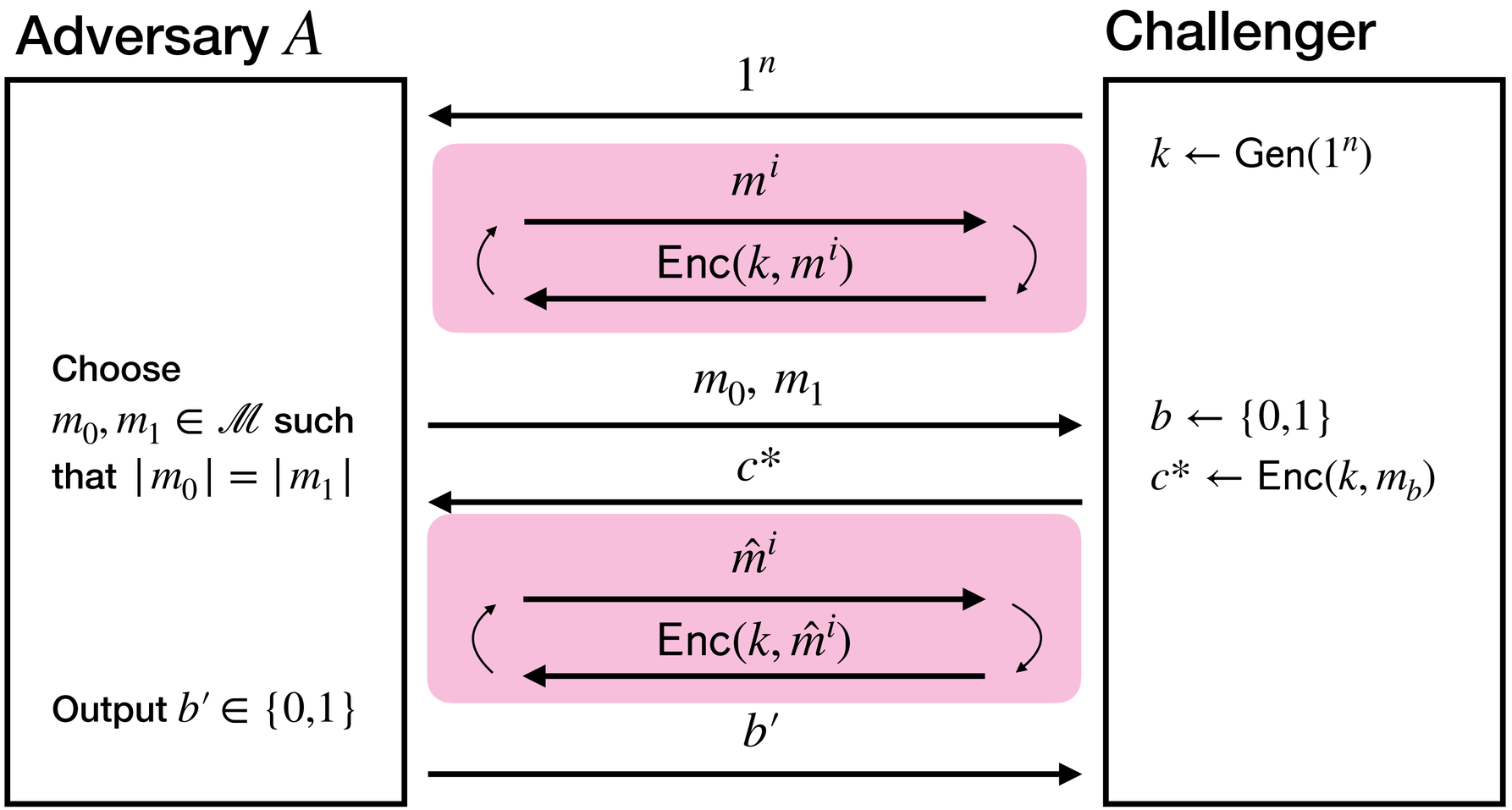
Reduction for Lemma 1



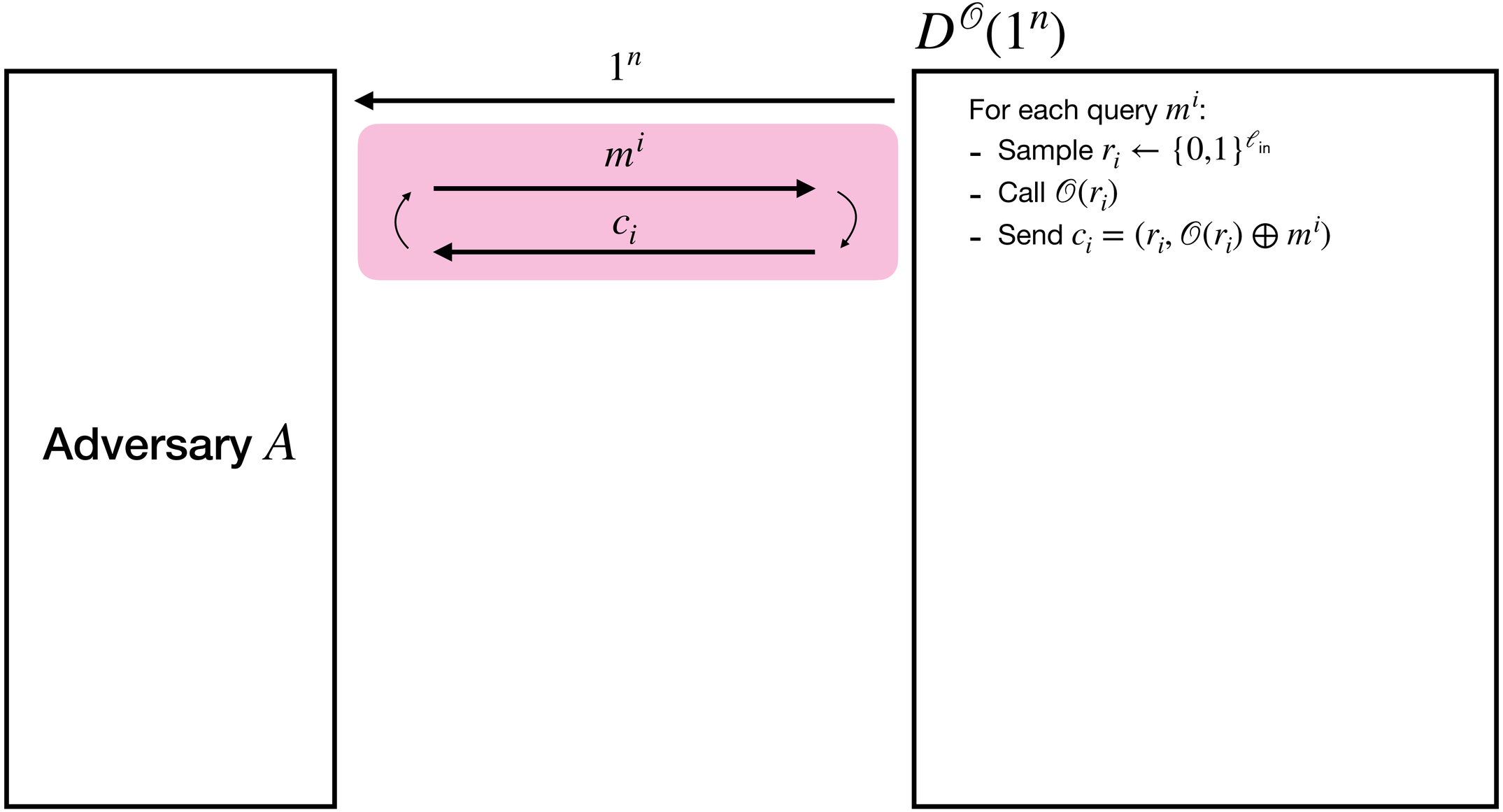
Recall: PRF Security Game



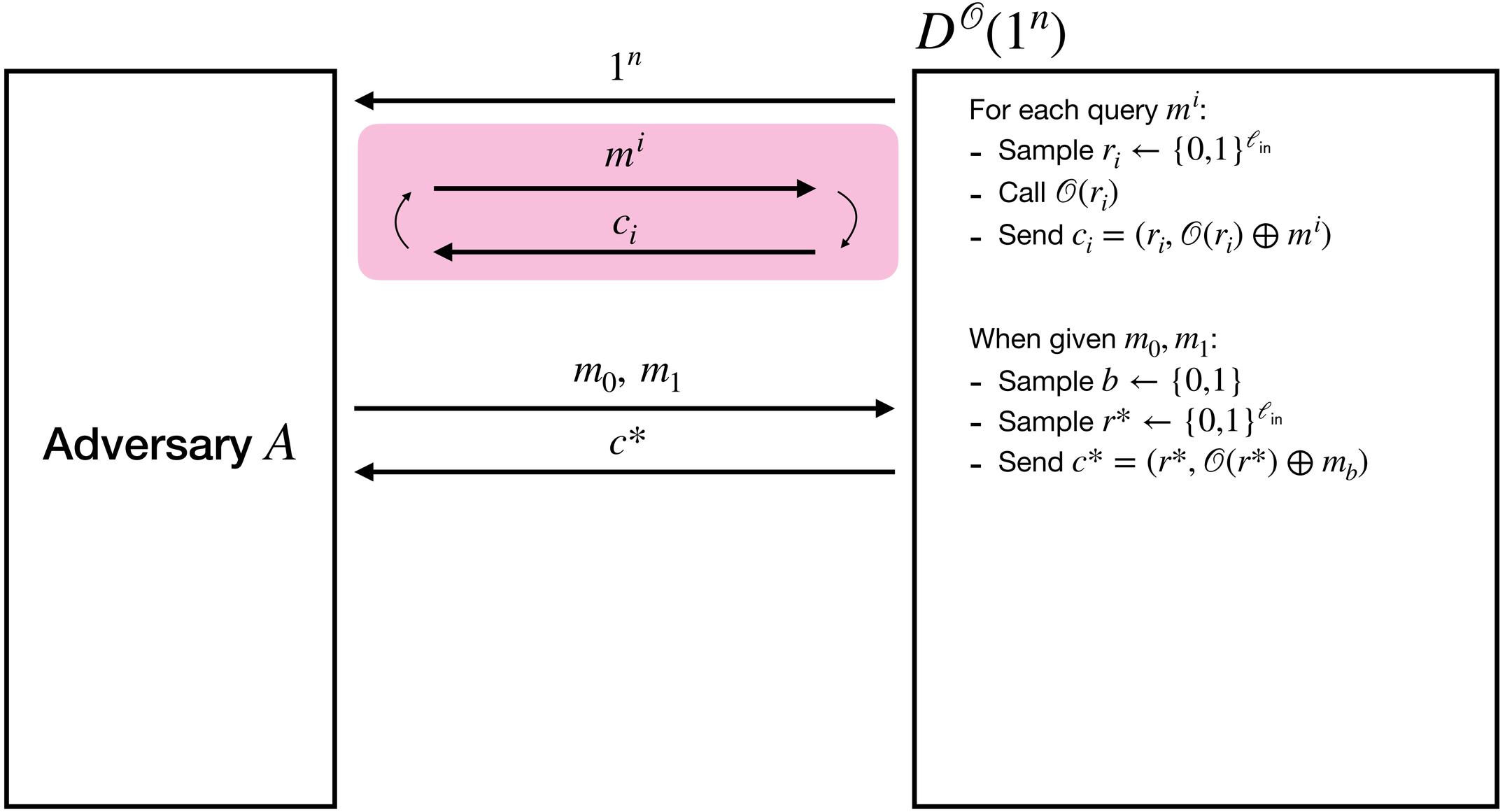
Recall: Chosen-Plaintext Attack (CPA)



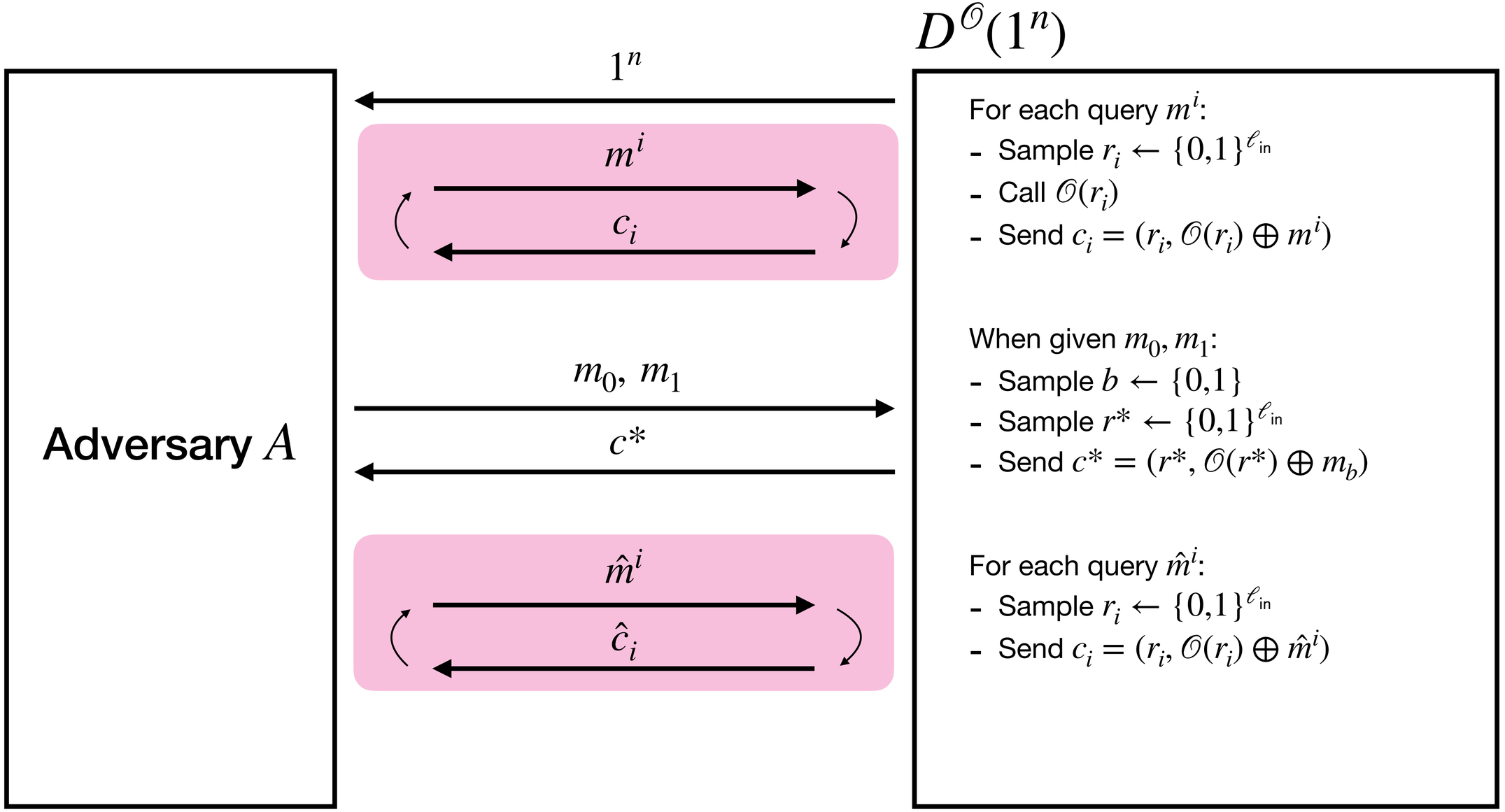
Reduction for Lemma 1



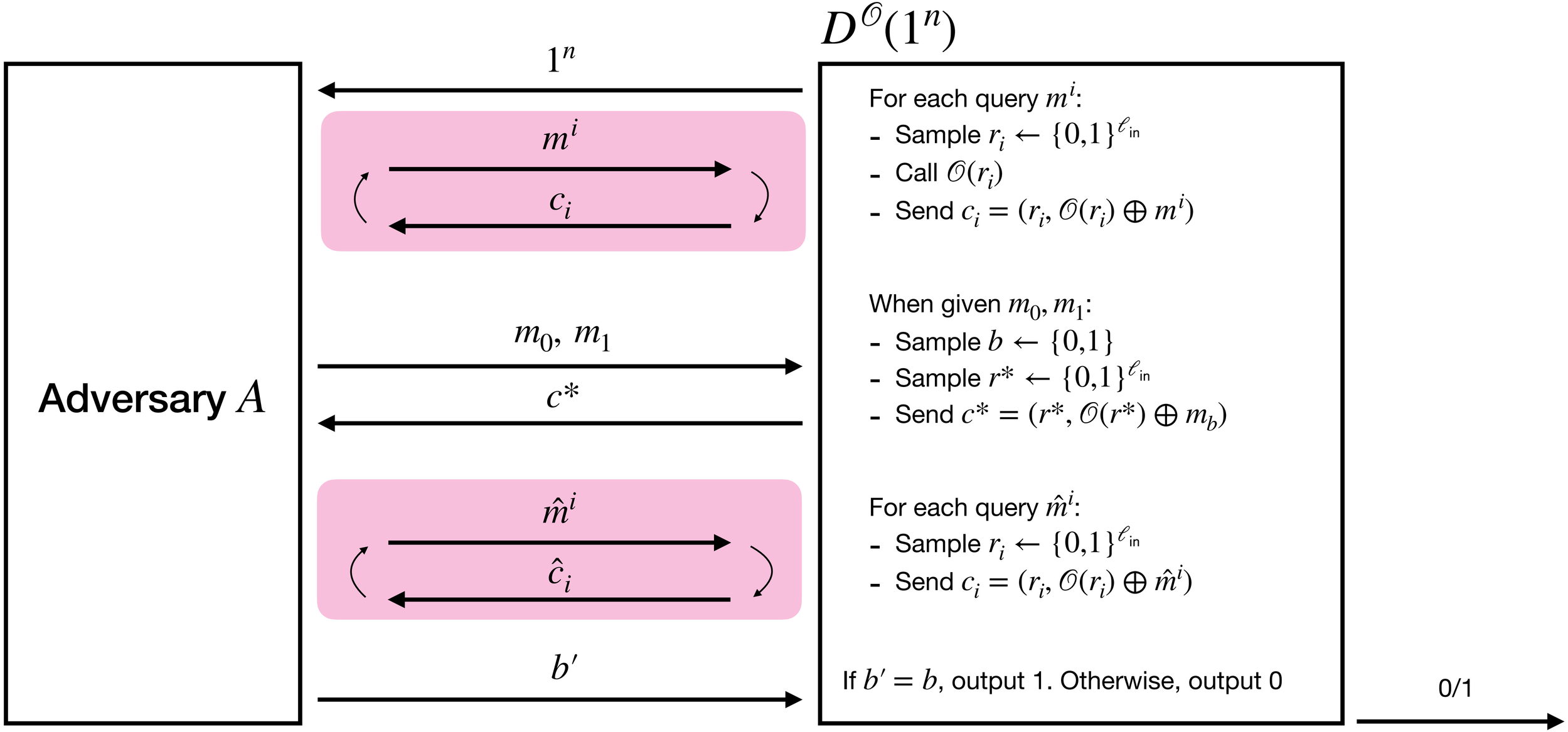
Reduction for Lemma 1



Reduction for Lemma 1

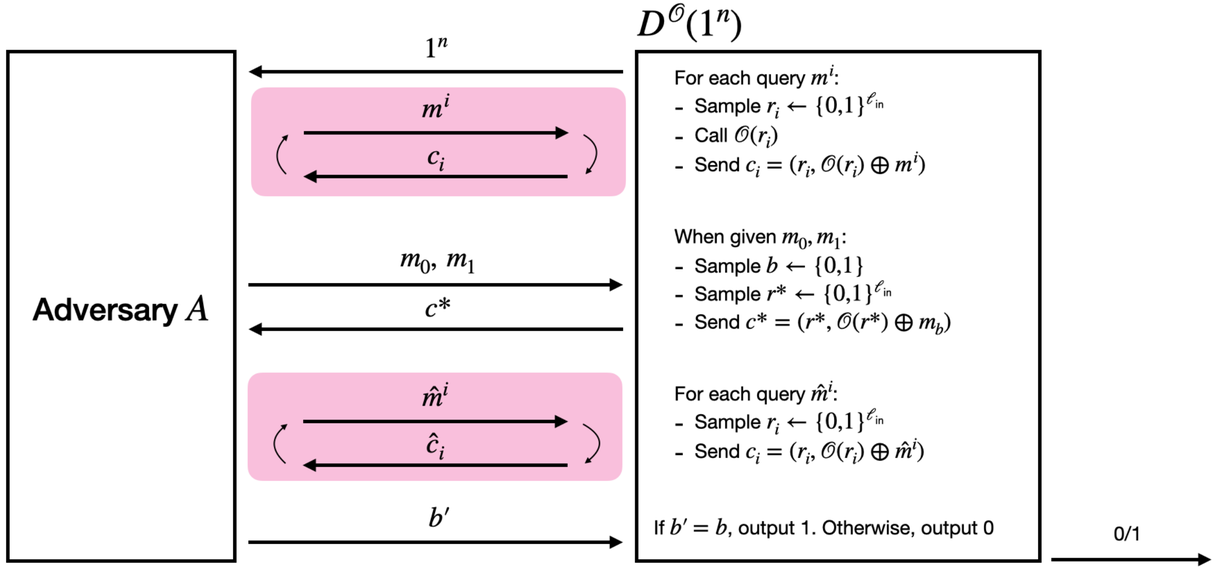


Reduction for Lemma 1



Reduction for Lemma 1

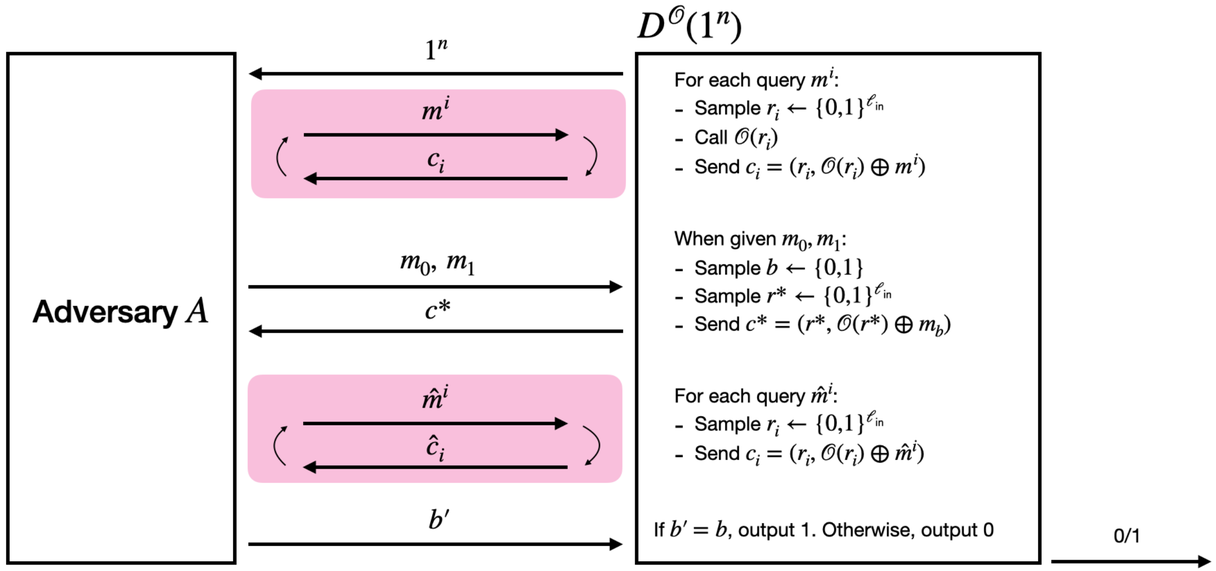
Case 1 (PRF world): $\mathcal{O} \leftarrow \{F_k\}_{k \in \{0,1\}^n}$



Reduction for Lemma 1

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

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

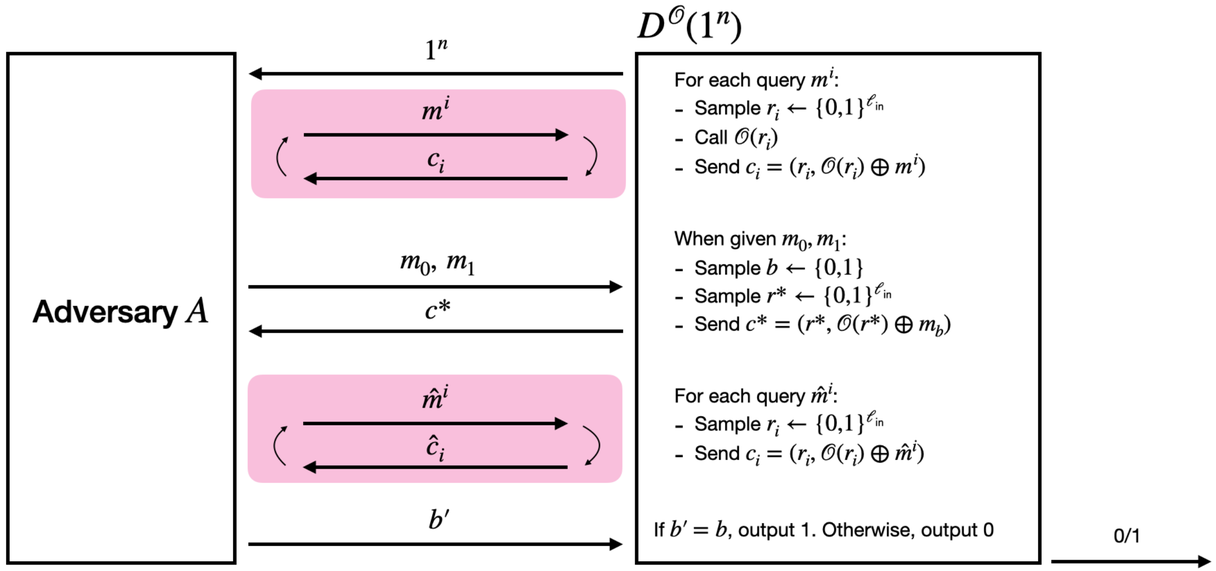


Reduction for Lemma 1

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

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

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



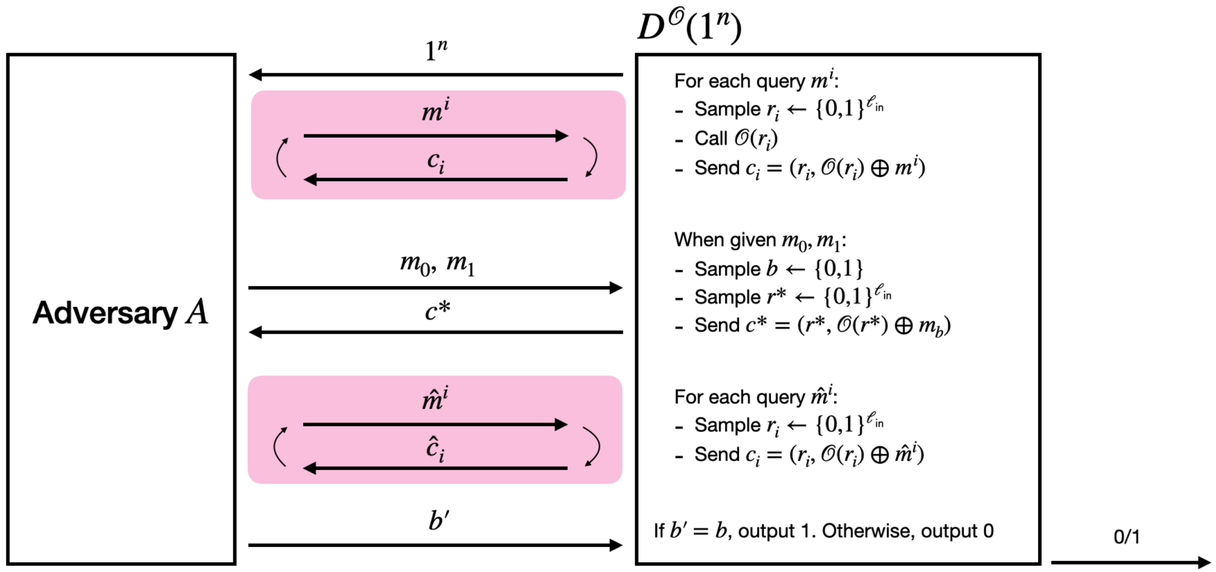
Reduction for Lemma 1

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

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

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

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



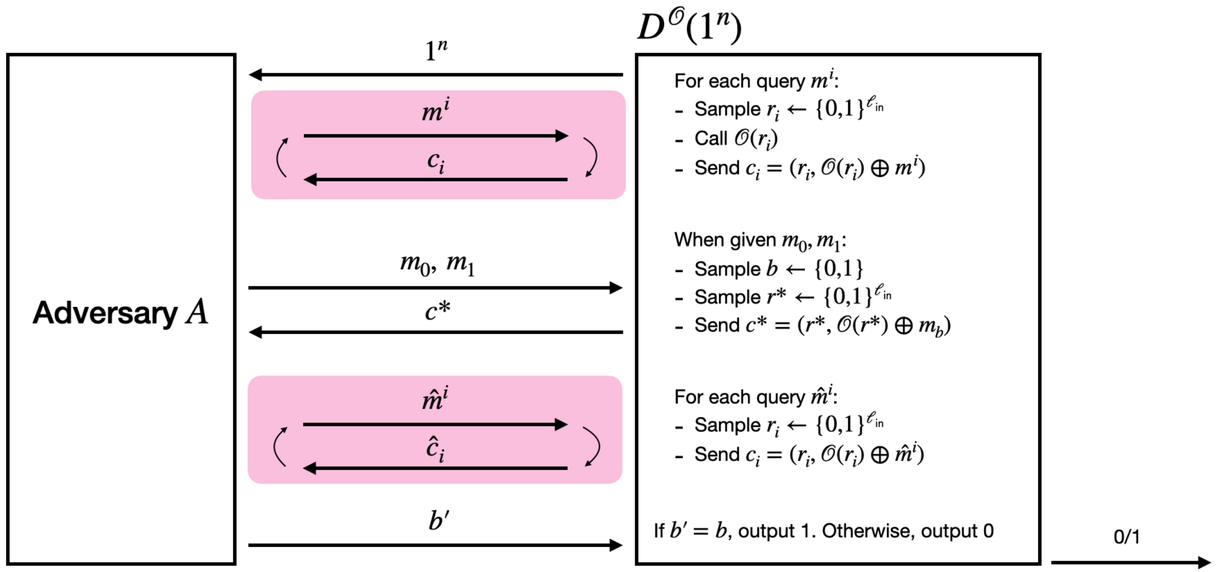
Reduction for Lemma 1

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

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

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

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



Putting this together:

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

Proving Lemma 1

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

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

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

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

Proving Lemma 2

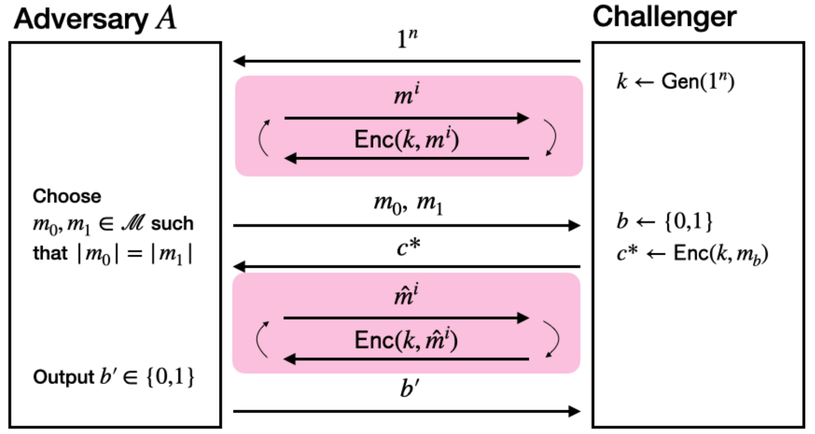
Lemma 2: For all PPT A, there exists a negligible function $\epsilon_2(\cdot)$ s.t.

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

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 $\{0,1\}^n$

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

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 $\{0,1\}^n$

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

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 $\{0,1\}^n$

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

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

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

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. $r^* = r_i$)

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

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. $r^* = r_i$)

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

This term is negligible

Lemma 1 + Lemma 2

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

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

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

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

Putting them together we have for all PPT A,

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

$$\leq \epsilon_1(n) + \frac{1}{2} + \epsilon_2(n)$$

Because $\epsilon_1(n) + \epsilon_2(n)$ is negligible, we have that Π 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

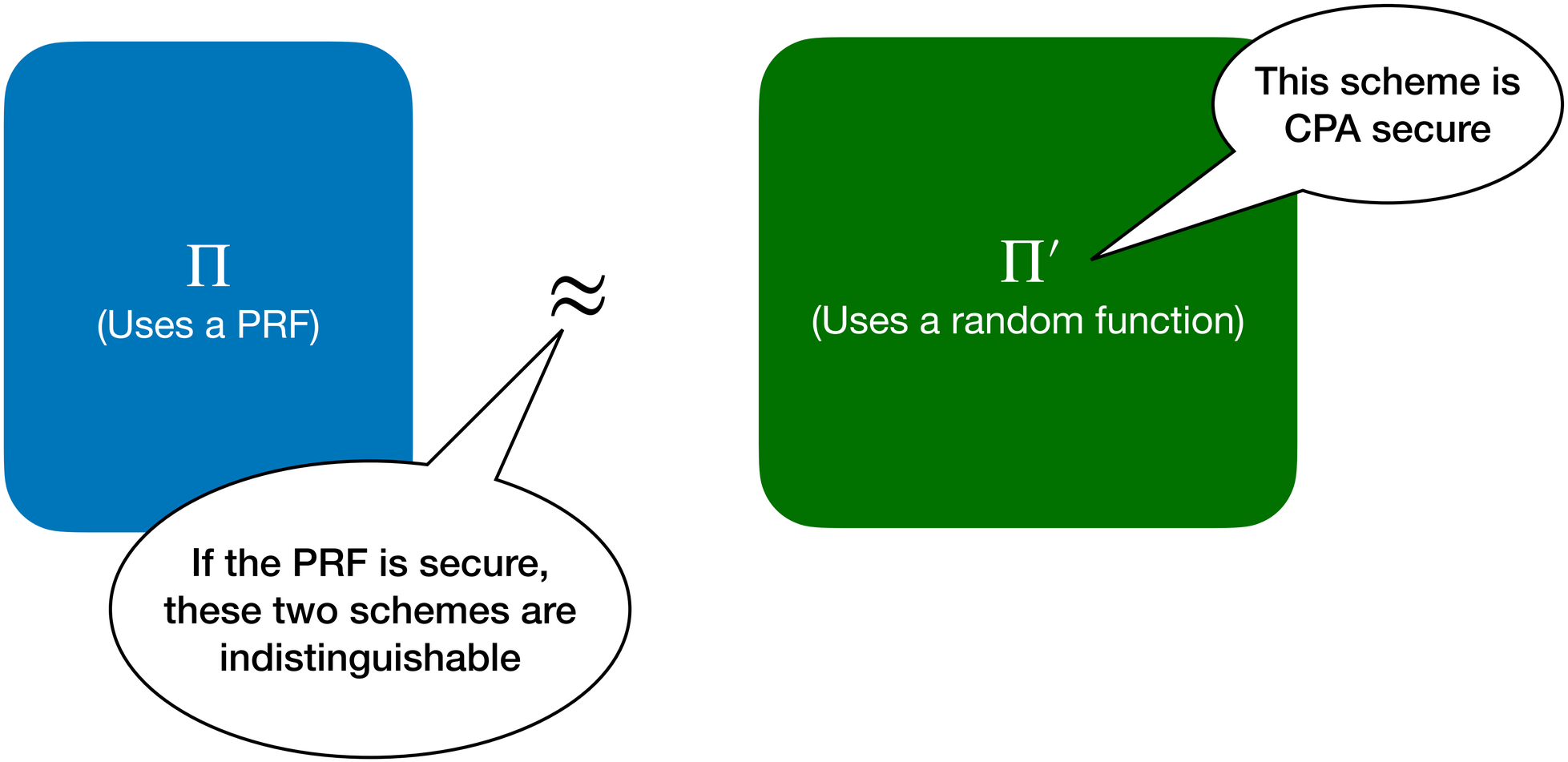
2. The version with the random function is secure

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?

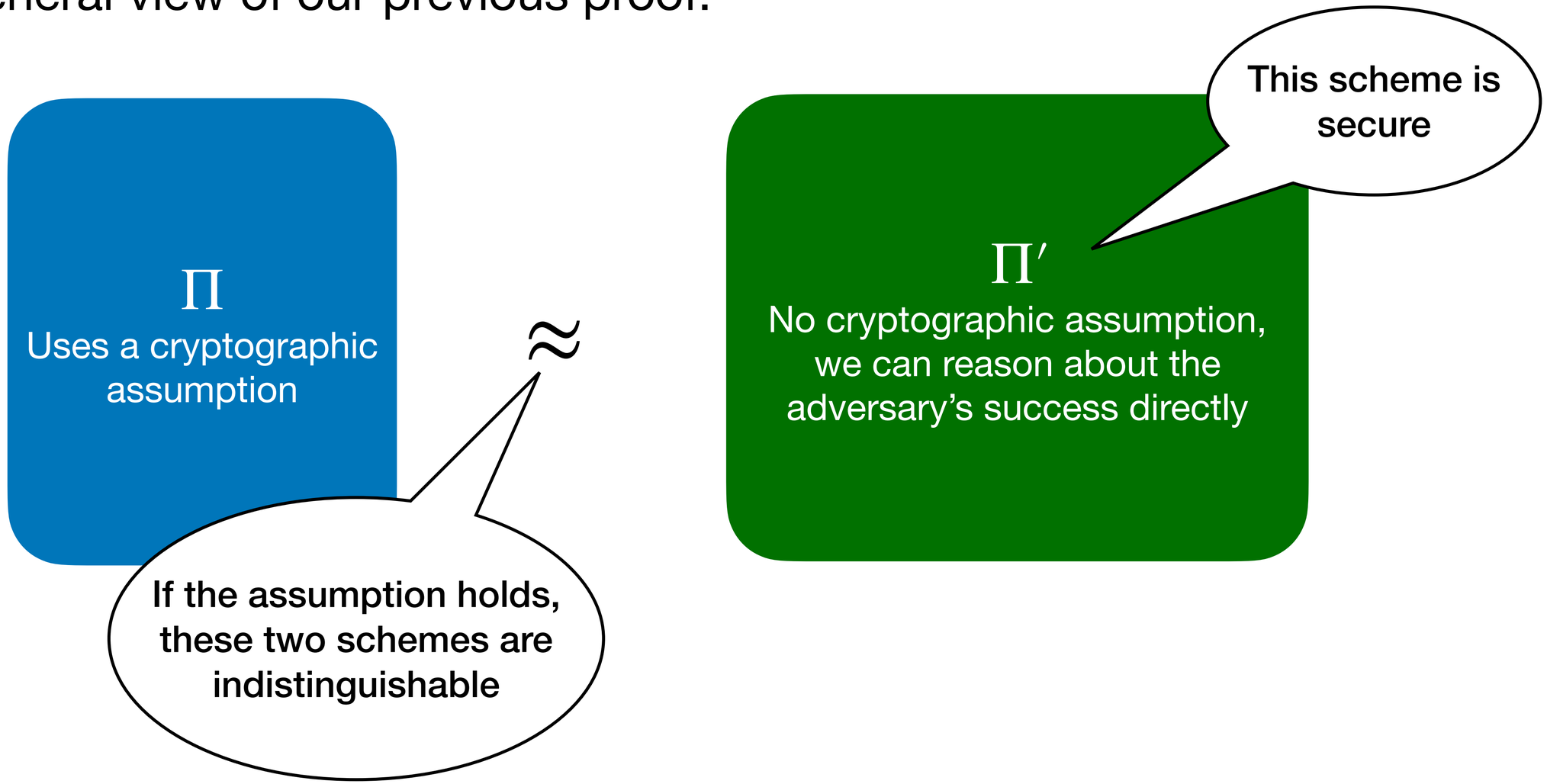
How to Prove More Involved Constructions?

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



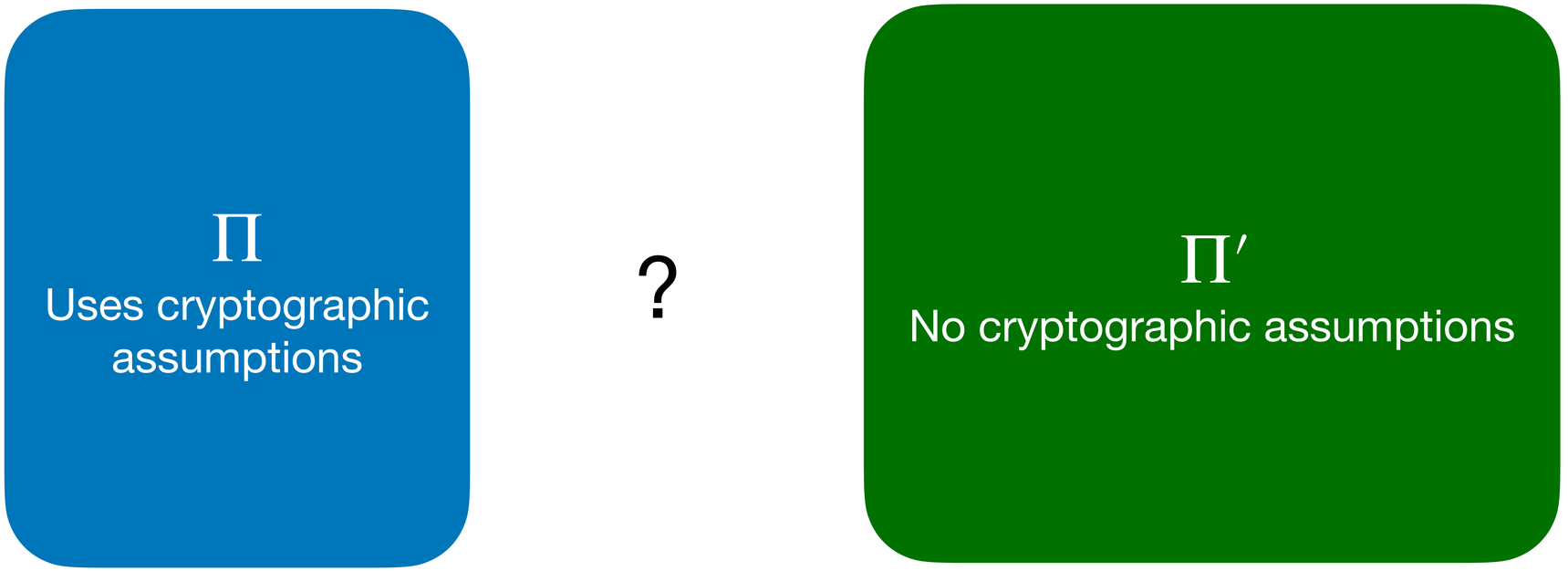
How to Prove More Involved Constructions?

A more general view of our previous proof:



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

Π
Uses cryptographic assumptions

Π''

Version of Π with one of the cryptographic components replaced

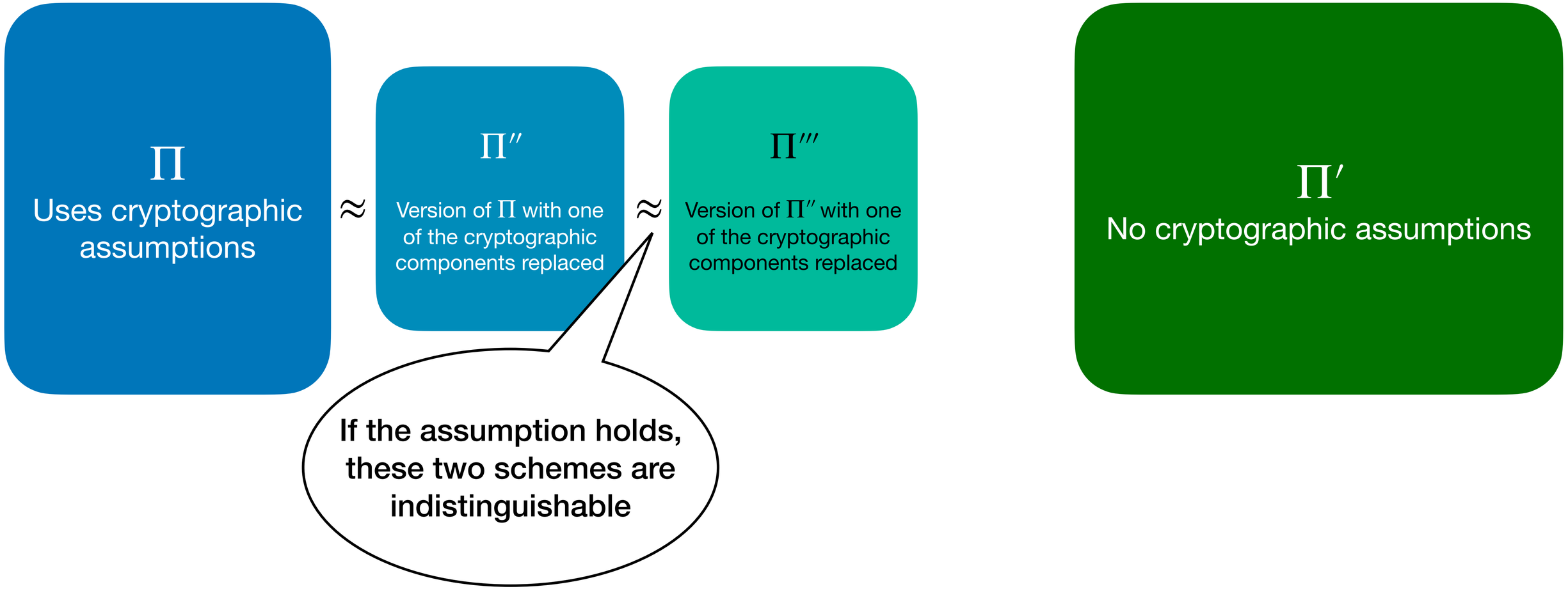
Π'
No cryptographic assumptions

How to Prove More Involved Constructions?

Protocols that uses cryptographic primitives in more than one place

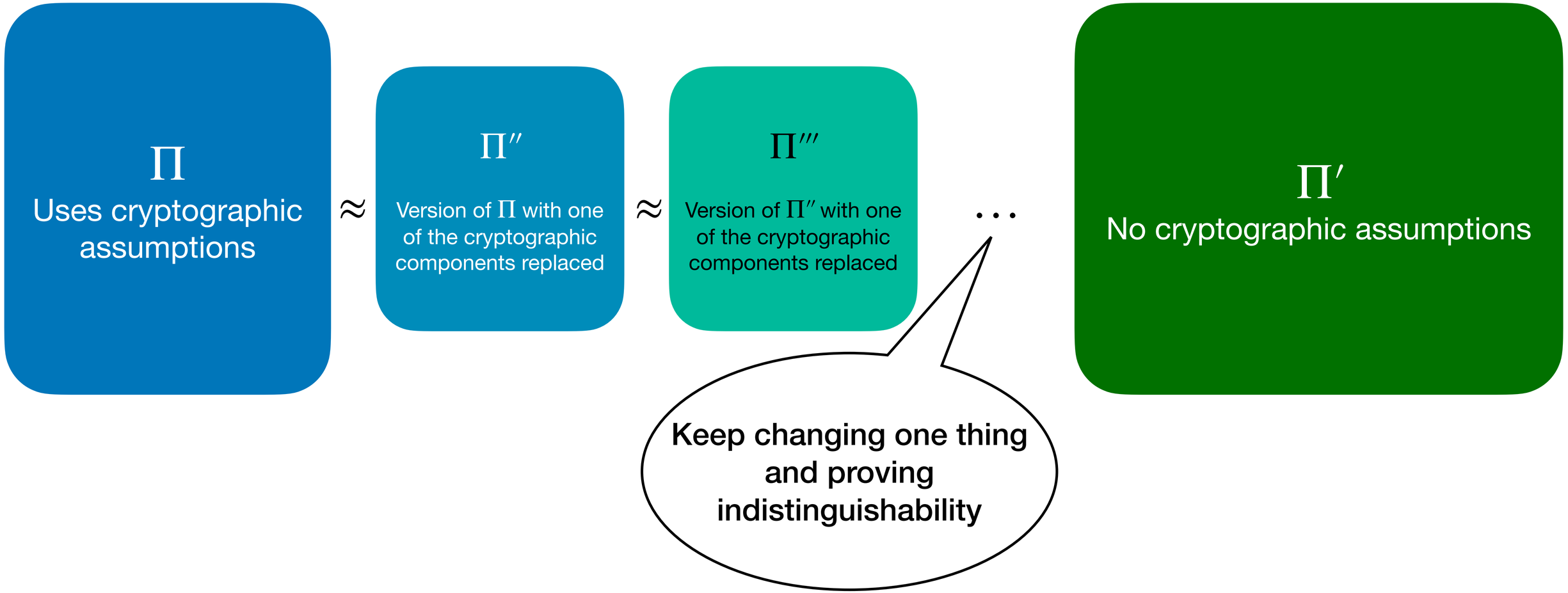
How to Prove More Involved Constructions?

Protocols that uses cryptographic primitives in more than one place



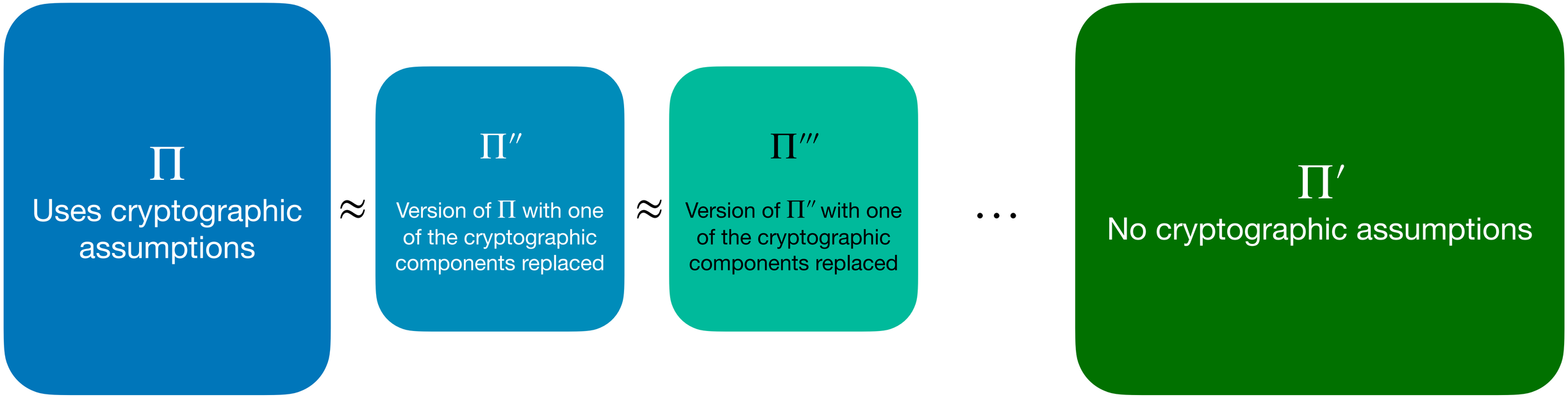
How to Prove More Involved Constructions?

Protocols that uses cryptographic primitives in more than one place



How to Prove More Involved Constructions?

Protocols that uses cryptographic primitives in more than one place

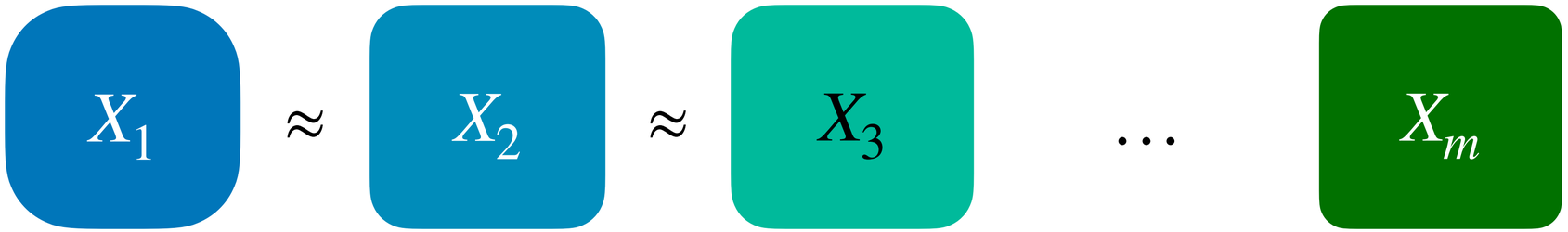


This is the idea behind a **hybrid argument**

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



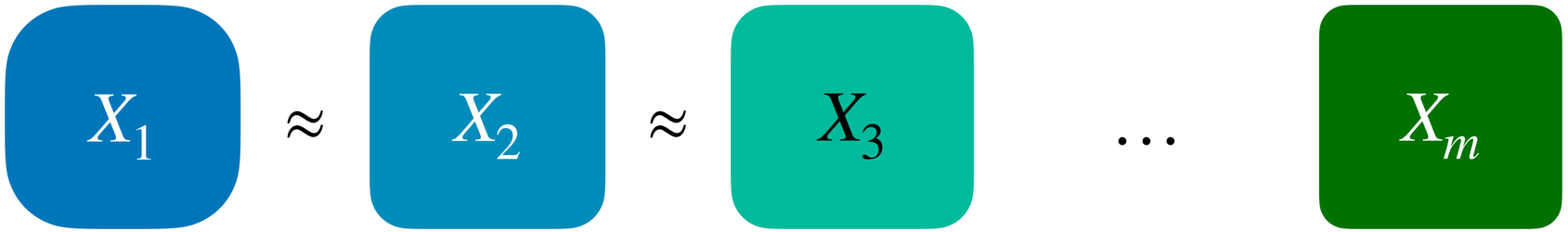
Hybrid Arguments

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 ϵ probability, then an adversary can distinguish X_1 from X_m with at most $m \cdot \epsilon$ probability

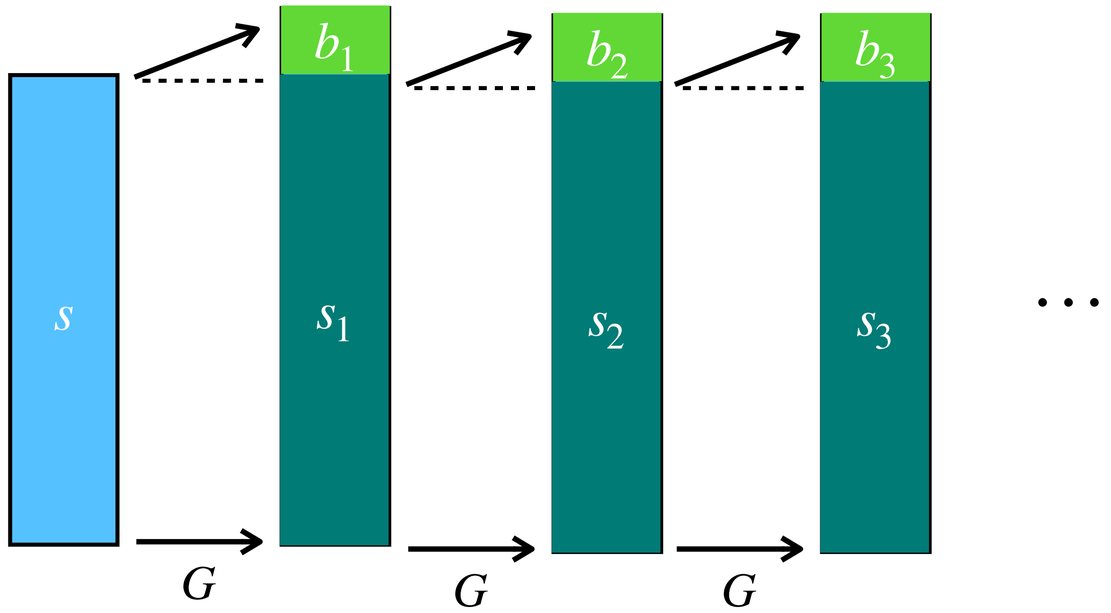
  - If ϵ is negligible and m is a polynomial, this product is negligible



Recall: Increasing the Stretch of a PRG

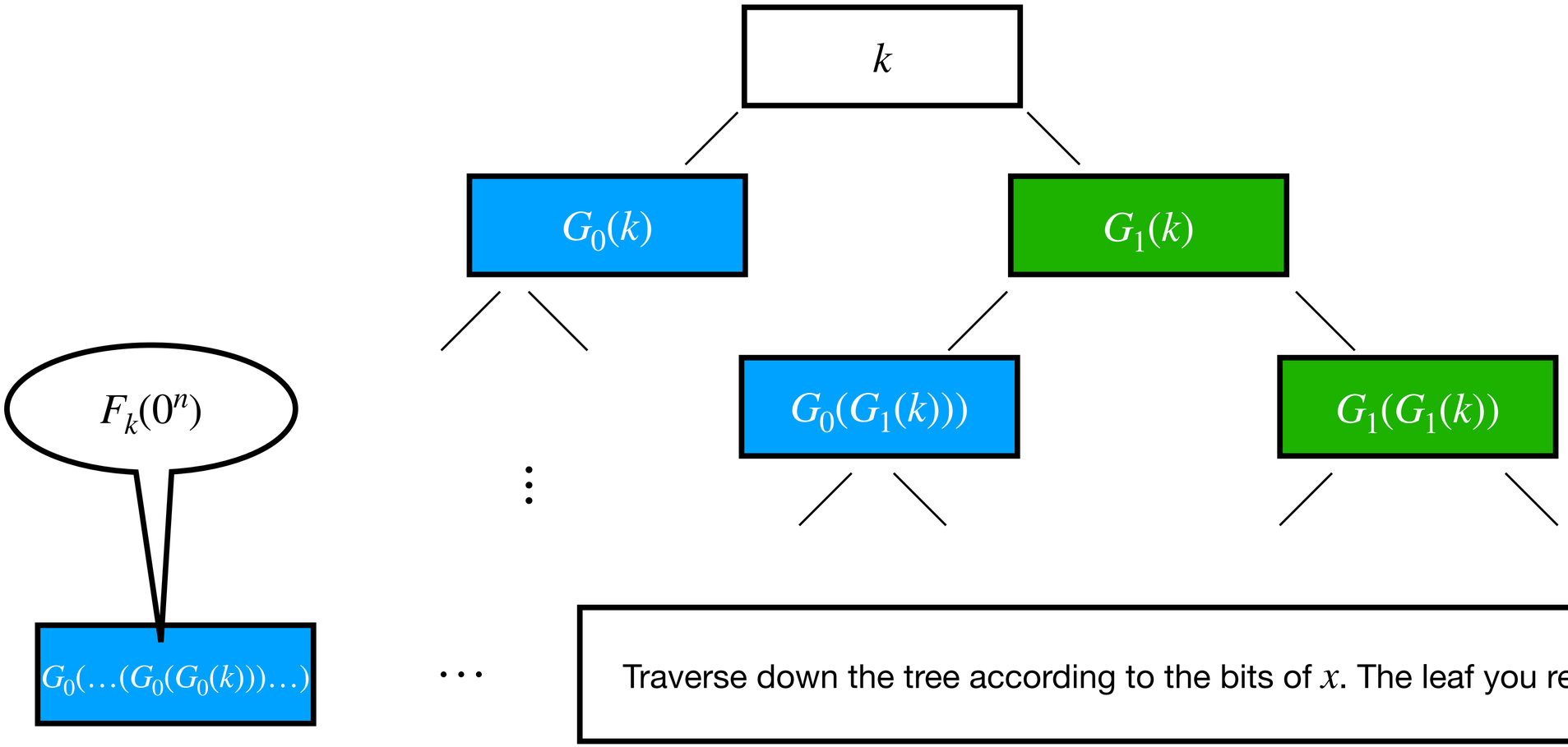
Theorem: If there exists a PRG with 1-bit expansion (i.e., $\ell(n) = n + 1$), then for any polynomial $p(n)$, there exists a PRG with $p(n)$-bit expansion

Proof sketch:



Recall: GGM Construction

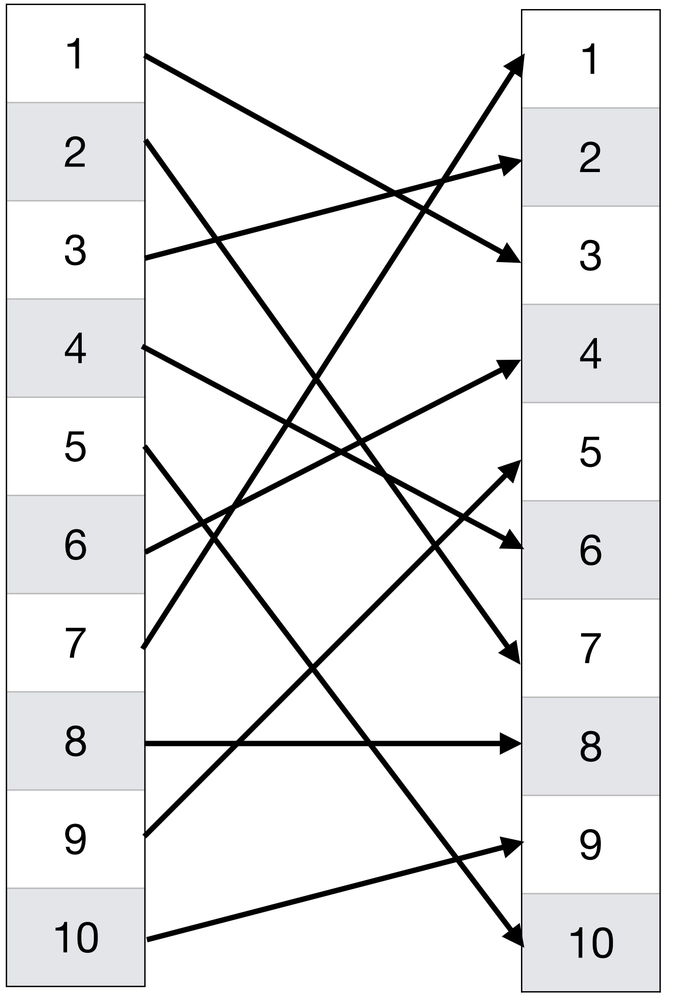
$$F_k(x) = G_{x_n}(\ldots(G_{x_2}(G_{x_1}(k)))\ldots)$$



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

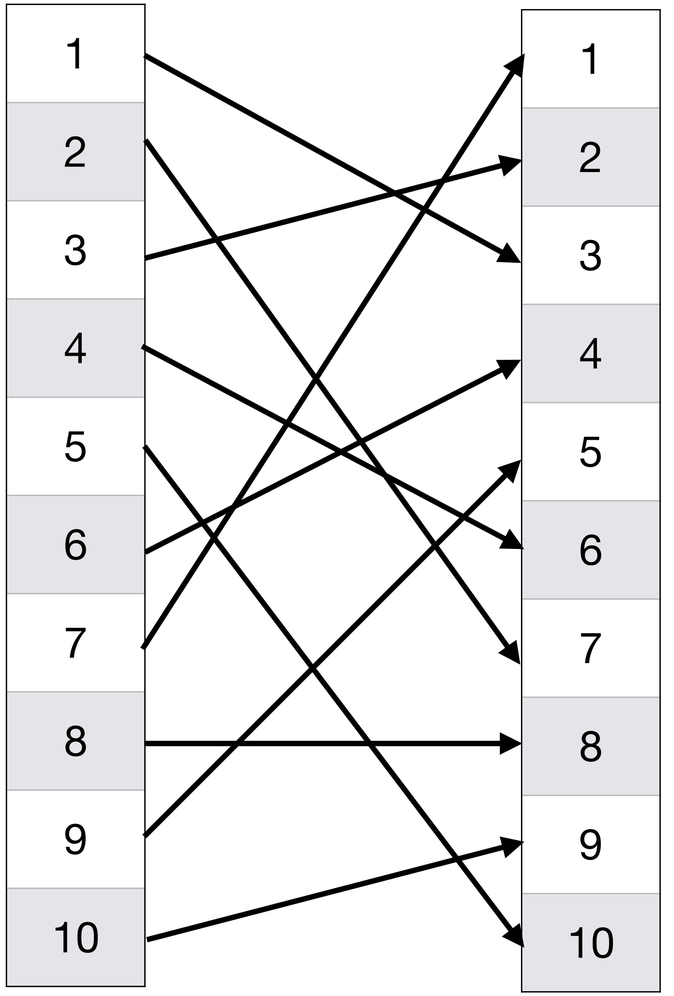


Please pretend these numbers are from $\{0,1\}^{\ell}$

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 $\mathscr{F}_n$ is the set of all possible functions from $\{0,1\}^n \to \{0,1\}^n$
  - We will denote $\mathscr{P}_n$ as the set of all possible permutations from $\{0,1\}^n \to \{0,1\}^n$



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 $k \in \{0,1\}^n$, $F_k : \{0,1\}^{\ell(n)} \to \{0,1\}^{\ell(n)}$ 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 $F_k^{-1}(y)$)

Note: This is **not** the same thing that was asked about last class and that I posted about on EdStem

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

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

Note: This is **not** the same thing that was asked about last class and that I posted about on EdStem

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

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

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

Note: This is **not** the same thing that was asked about last class and that I posted about on EdStem

Practical Heuristics

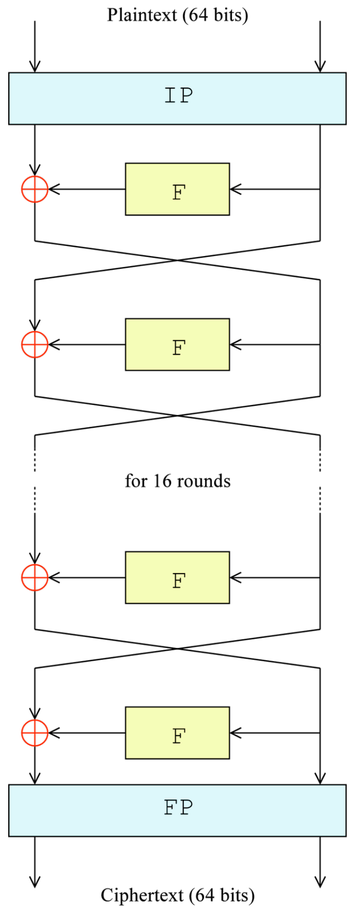
Practical Heuristics: Block Ciphers

- 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

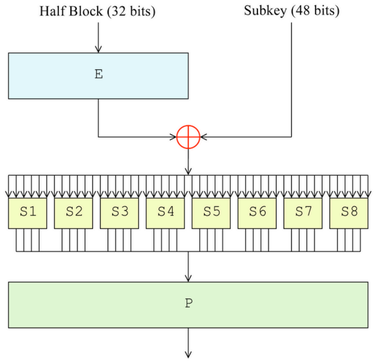
We'll look at a few well-known block ciphers but won't go into too much detail

Block Ciphers: DES

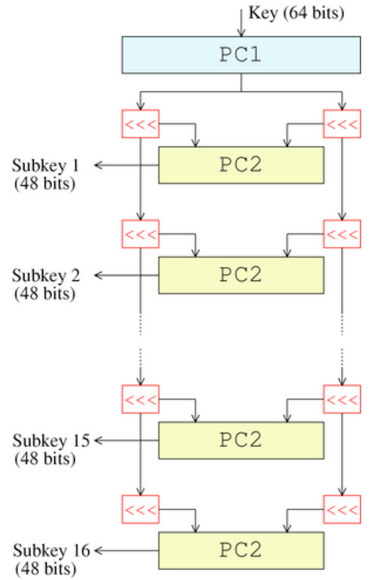
- The Data Encryption Standard (DES) was developed in the 1970s by IBM (with help from the NSA)
  - Adopted in 1977 as a Federal Information Processing Standard (FIPS)
- Key length is 56-bits and block length (length of its input and output) is 64-bits



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

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

- 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

Using CPA-Secure Encryption

- **Recall**: CPA-secure encryption from any PRF

$$\mathsf{Enc}(k, m) = (r, F_k(r) \oplus m)$$

- **In practice**: AES as a PRF enables encrypting a 128-bit message

$$\mathsf{Enc}(k, m) = (r, \mathsf{AES}_k(r) \oplus m)$$

Question: If we view AES as a strong PRP, then AES_k^{-1} is well defined and efficiently computable. Why not use $\mathsf{Enc}(k, m) = \mathsf{AES}_k(m)$ and $\mathsf{Dec}(k, c) = \mathsf{AES}_k^{-1}(c)$?

Using CPA-Secure Encryption

- **How to encrypt long messages**: Partition into blocks

$$\mathsf{Enc}(k, m_1 \ldots m_\ell) = (r_1, F_k(r_1) \oplus m_1), \ldots, (r_\ell, F_k(r_\ell) \oplus m_\ell)$$

- **In practice**: AES as a PRF enables encrypting 128-bit blocks

$$\mathsf{Enc}(k, m_1 \ldots m_\ell) = (r_1, \mathsf{AES}_k(r_1) \oplus m_1), \ldots, (r_\ell, \mathsf{AES}_k(r_\ell) \oplus m_\ell)$$

Drawback: Ciphertext length is double the message length

Using CPA-Secure Encryption

- **How to encrypt long messages**: Partition into blocks

$$\mathsf{Enc}(k, m_1 \ldots m_\ell) = (r_1, F_k(r_1) \oplus m_1), \ldots, (r_\ell, F_k(r_\ell) \oplus m_\ell)$$

- **In practice**: AES as a PRF enables encrypting 128-bit blocks

$$\mathsf{Enc}(k, m_1 \ldots m_\ell) = (r_1, \mathsf{AES}_k(r_1) \oplus m_1), \ldots, (r_\ell, \mathsf{AES}_k(r_\ell) \oplus m_\ell)$$

Drawback: Ciphertext length is double the message length

Can we do better?

Electronic CodeBook (ECB) Mode

- Naive mode of operation

- Just apply the block cipher to each block separately:

$$c = F_k(m_1), F_k(m_2), \ldots, F_k(m_\ell)$$

Electronic CodeBook (ECB) Mode

- Naive mode of operation
- Just apply the block cipher to each block separately:

$$c = F_k(m_1), F_k(m_2), \ldots, F_k(m_\ell)$$

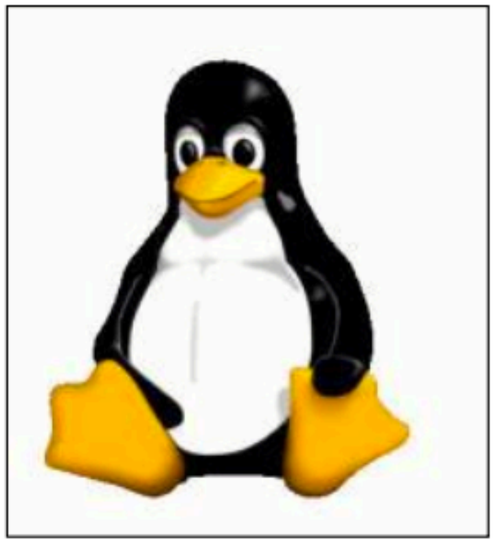
- … It should be very obvious that this is not good

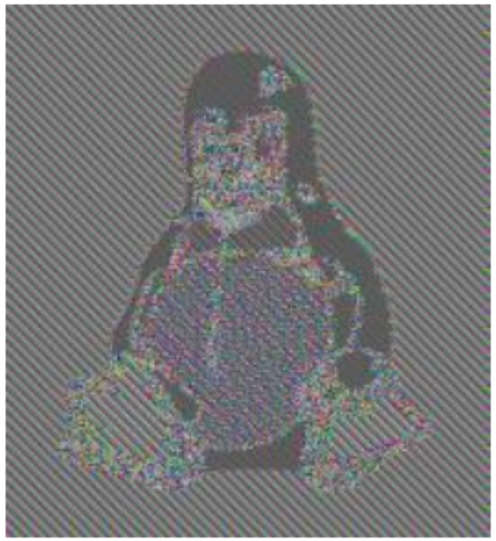
Electronic CodeBook (ECB) Mode

$$c = F_k(m_1), F_k(m_2), \ldots, F_k(m_\ell)$$

- Example by penguin picture:

Original image





ECB mode encryption

https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

Move Fast and Roll Your Own Crypto

A Quick Look at the Confidentiality of Zoom Meetings

By Bill Marczak and John Scott-Railton April 3, 2020

Read our description of Zoom's waiting room vulnerability, as well as frequently asked question about Zoom and encryption issues.

This report examines the encryption that protects meetings in the popular Zoom teleconference app. We find that Zoom has "rolled their own" encryption scheme, which has significant weaknesses. In addition, we identify potential areas of concern in Zoom's infrastructure, including observing the transmission of meeting encryption keys through China.

Key Findings

- Zoom documentation claims that the app uses "AES-256" encryption for meetings where possible. However, we find that in each Zoom meeting, a single AES-128 key is used in ECB mode by all participants to encrypt and decrypt audio and video. The use of ECB mode is not recommended because patterns present in the plaintext are preserved during encryption.
- The AES-128 keys, which we verified are sufficient to decrypt Zoom packets intercepted in Internet traffic, appear to be generated by Zoom servers, and in some cases, are delivered to participants in a Zoom meeting through servers in China, even when all

Cipher Block Chaining (CBC) Mode

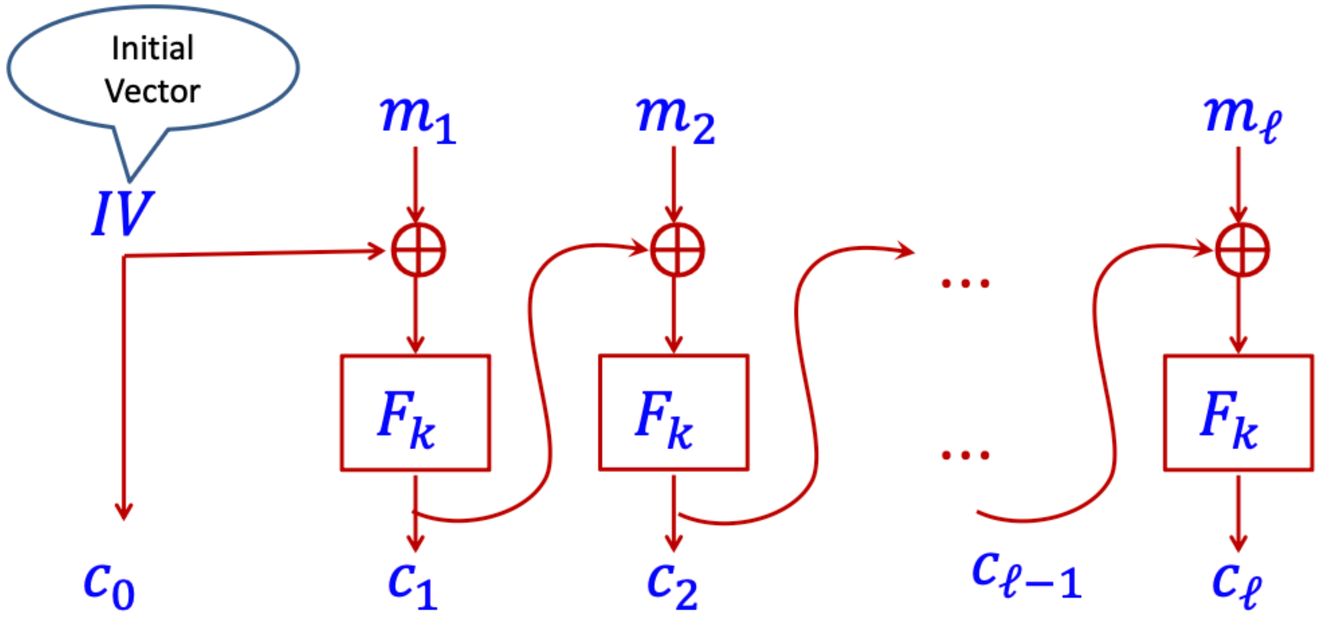
- This mode starts by sampling a random “initialization vector” (IV) and setting that as the first part of the cipher text.

- Each next part of the cipher text is obtained by applying the block cipher to the XOR of the previous cipher text and the message:

$$
\begin{aligned}
c &= (c_0, c_1, c_2, \ldots, c_\ell) \\
&= (IV, F_k(c_0 \oplus m_1), F_k(c_1 \oplus m_2), \ldots, F_k(c_{\ell-1} \oplus m_\ell))
\end{aligned}
$$

Cipher Block Chaining (CBC) Mode

$$c = (c_0, c_1, c_2, \ldots, c_\ell)$$
$$= (IV, F_k(c_0 \oplus m_1), F_k(c_1 \oplus m_2), \ldots, F_k(c_{\ell-1} \oplus m_\ell))$$



CBC Properties

- **Theorem**: If F is a strong PRP, then CBC is CPA-secure
- Expansion is only one block (compared to applying our OTP-inspired technique which doubled the size)
- Encryption is inherently **sequential** and cannot be done in parallel
- Message length is assume to be a multiple of block length, but padding can be added

Other Modes of Operation?

- We’ll talk about next time

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

Acknowledgements: Thank you to Ran Cohen and Tal Malkin for sharing their lecture notes/slides to use as reference!