COMS BC3262: Introduction to Cryptography

Lecture 4: CPA-Security and PRFs

February 2, 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

Please see course website for slides, homework, course expectations:

https://www.eysalee.com/courses/s26/bc3262.html

PS1 is due Thursday, PS2 is released Wednesday

Lowest PS grade is dropped

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

EdStem

- Class forum for questions and announcements
  - People have already been asking great questions! Thank you!
- Appropriate for public posts: Clarifications on HW or material that may be helpful for the entire class
- Appropriate for private posts: Anything specific to you or your solution to a HW problem
  - When in doubt, you can always post privately at first
  - You are allowed to ask questions about specific parts of your solution

    (e.g, *“Do I need to elaborate/prove how I go between these two steps?”* or *“can I take XYZ as fact”*)
  - We will ignore questions asking if a solution is correct

Today's Lecture

- More on PRGs
- A quick detour to semantic security
- Multiple message security
    - CPA-Security!
- PRFs

Pseudorandom Generators (PRGs)

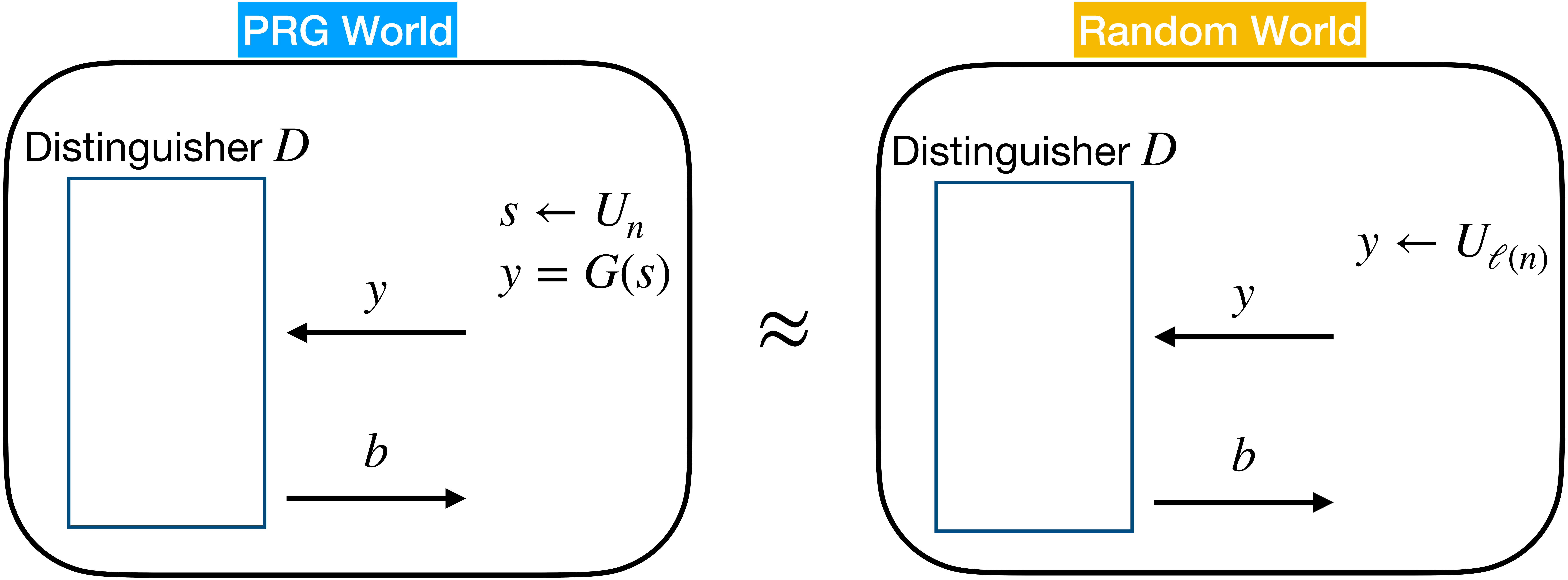
Pseudorandom Generators (PRGs)

Definition: Let G be a deterministic polynomial-time algorithm and $\ell(\cdot)$ be a polynomial s.t. for any input $s \in \{0,1\}^n$ we have $G(s) \in \{0,1\}^{\ell(n)}$. Then G is a **pseudorandom generator** if the following two conditions hold:

- **Expansion**: $\ell(n) > n$
- **Pseudorandomness**: For every PPT "distinguisher" D there exists a negligible function negl$(\cdot)$ s.t.

$$\left| \Pr_{s \leftarrow \{0,1\}^n} \left[D\left(G(s)\right) = 1 \right] - \Pr_{r \leftarrow \{0,1\}^{\ell(n)}} \left[D(r) = 1 \right] \right| \leq \text{negl}(n)$$

PRG Distinguisher



PRG World

Distinguisher D

$s \leftarrow U_n$

$y = G(s)$

y

b

$\approx$

Random World

Distinguisher D

$y \leftarrow U_{\ell(n)}$

y

b

$$\left| \Pr_{s\leftarrow\{0,1\}^n}\left[D\left(G(s)\right) = 1\right] - \Pr_{r\leftarrow\{0,1\}^{\ell(n)}}\left[D(r) = 1\right] \right| \le \mathsf{negl}(n)$$

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$)

Let $G'(s) = \text{reverse}(G(s))$ where $\text{reverse}(\text{x})$ reverses the bits of x

Is $G'(s)$ a PRG?

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$

Let $G'(s) = \text{reverse}(G(s))$ where $\text{reverse}(\text{x})$ reverses the bits of x

Is $G'(s)$ a PRG?

Yes!

- Intuition: If s is random, then $G(s)$ "looks random". Reversing its bits also also looks random

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$

Let $G'(s) = \text{reverse}(G(s))$ where $\text{reverse}(\text{x})$ reverses the bits of x

We're going to prove $G'(s)$ is PRG via reduction:

- Given a distinguisher D' that breaks G', we'll use it to construct a distinguisher D that breaks G.
- If D' is PPT and succeeds with non-negligible probability, then D is also PPT and succeeds with non-negligible probability.
- But if we assume G is secure, then such a D can't exist. Therefore D' also can't exist, so G' is a secure PRG.

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$

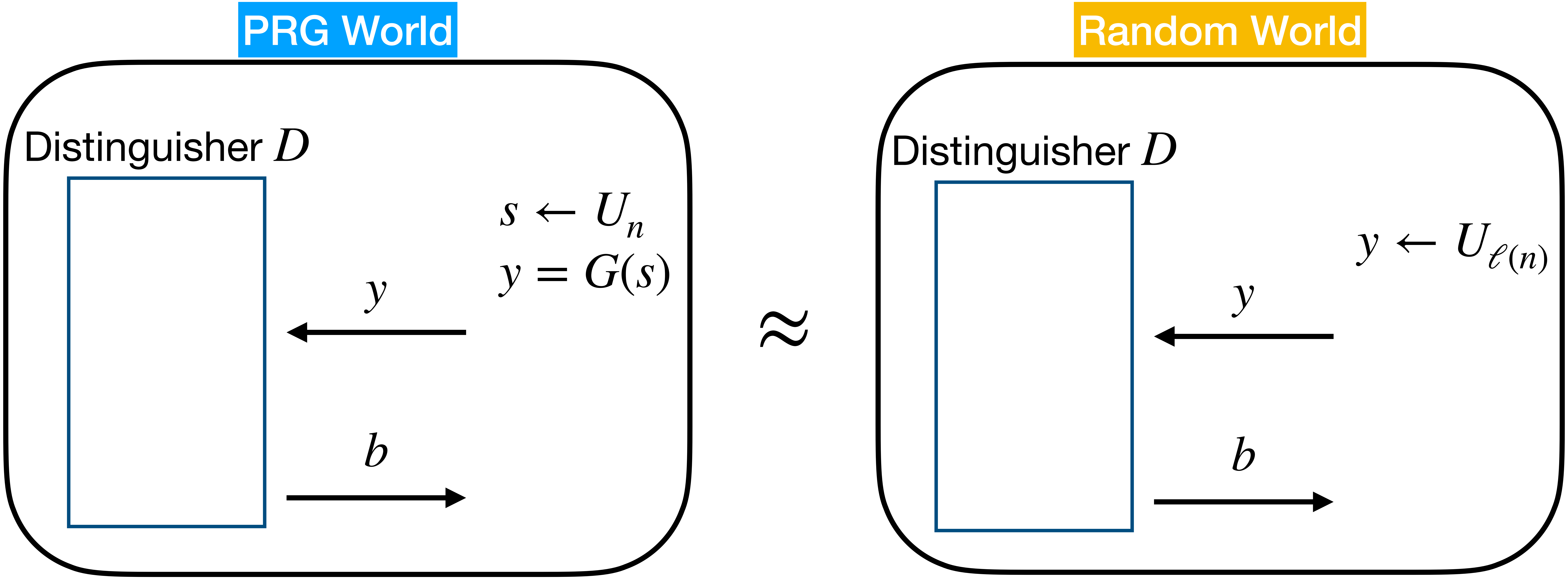
Let $G'(s) = \text{reverse}(G(s))$ where reverse(x) reverses the bits of x

We're going to prove $G'(s)$ is PRG via reduction:

- Given a distinguisher D' that breaks G', w distinguisher D that breaks G.
- If D' is PPT and succeeds with non-negligib PPT and succeeds with non-negligible prob y.
- But if we assume G is secure, then suc a D can't exist. Therefore D' also can't exist, so G' is a secure PRG.

Don't forget to state somewhere what you're trying to prove :)

Recall: PRG Distinguisher



$$\left| \Pr_{s \leftarrow \{0,1\}^n}\left[D\left(G(s)\right) = 1\right] - \Pr_{r \leftarrow \{0,1\}^{\ell(n)}}\left[D(r) = 1\right] \right| \leq \mathsf{negl}(n)$$

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$

Let $G'(s) =$ reverse($G(s)$) where reverse(x) reverses the bits of x

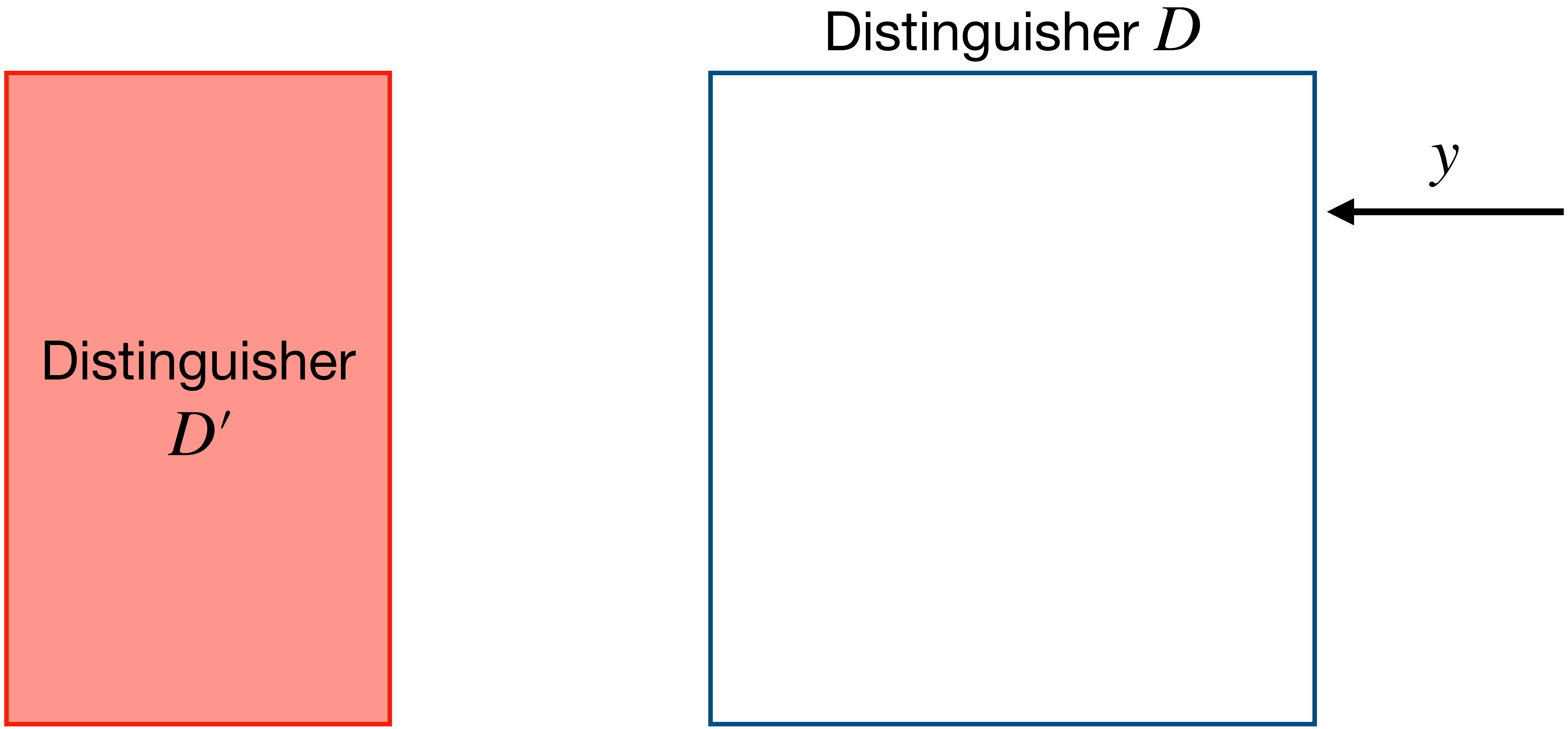
Distinguisher D'

Distinguisher D

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$

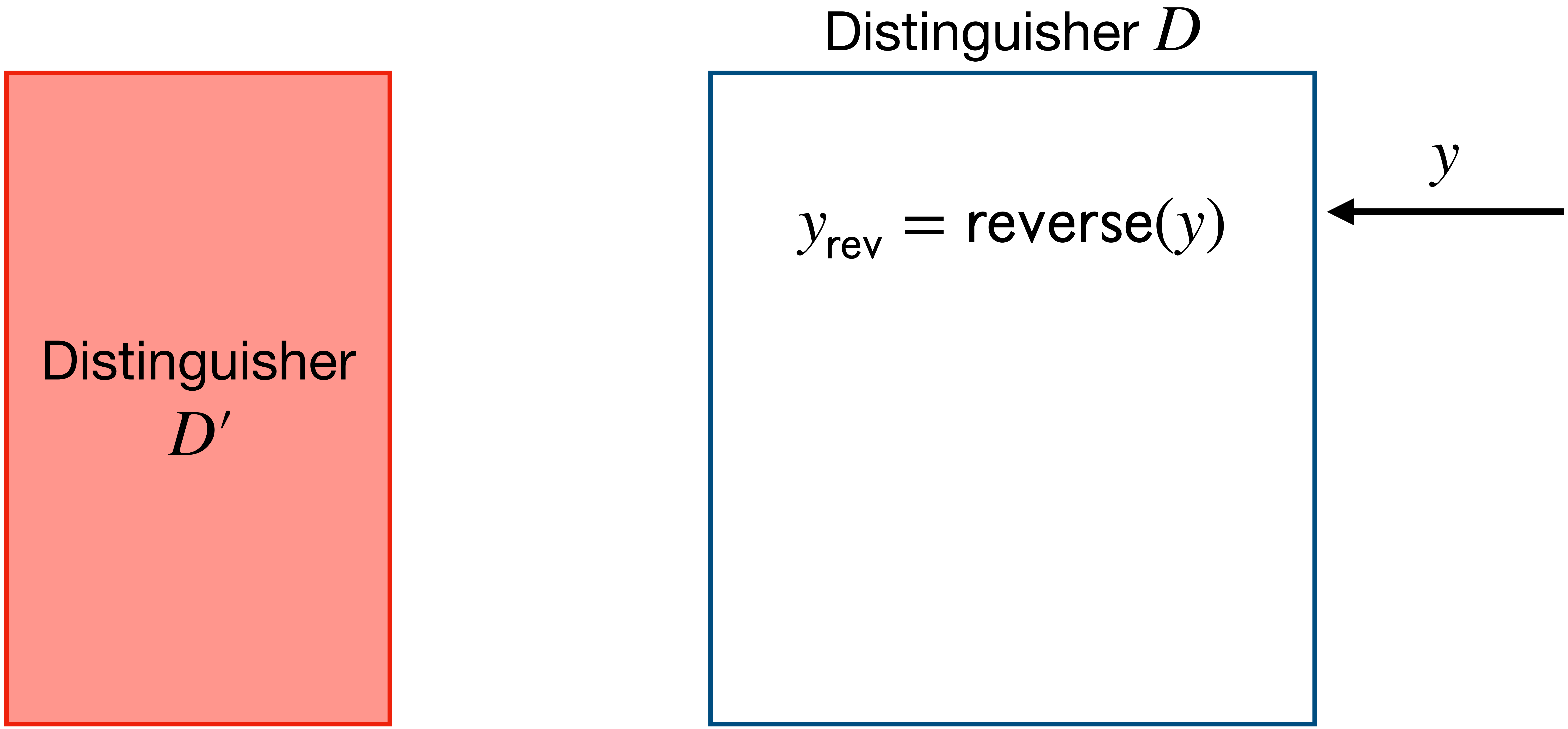
Let $G'(s) = \text{reverse}(G(s))$ where $\text{reverse}(\text{x})$ reverses the bits of x



PRG Example

Assume G is a PRG ($G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$

Let $G'(s) = \mathsf{reverse}(G(s))$ where $\mathsf{reverse}(\mathsf{x})$ reverses the bits of x



Distinguisher D

$y_{\text{rev}} = \mathsf{reverse}(y)$

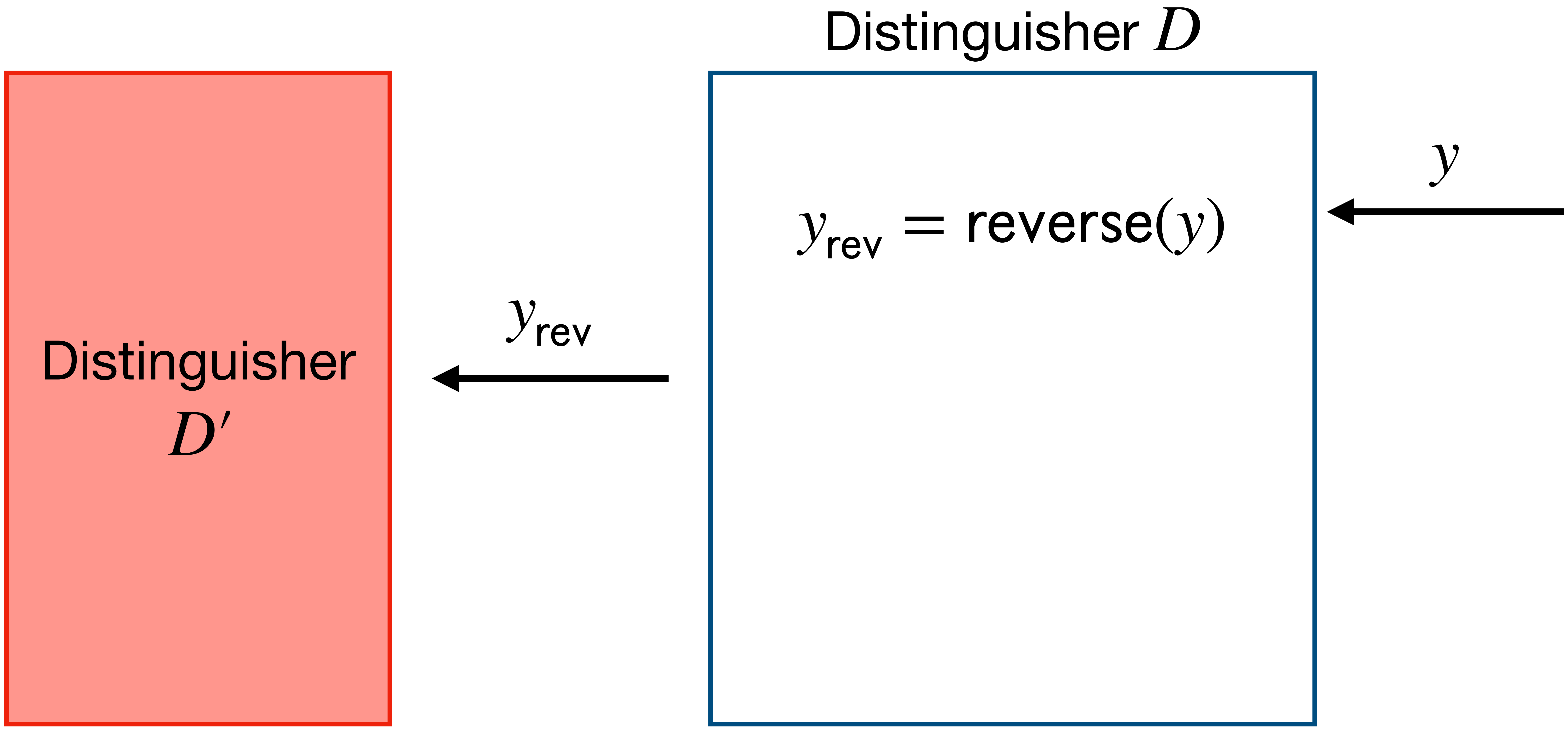
y

Distinguisher D'

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$

Let $G'(s) = \text{reverse}(G(s))$ where reverse(x) reverses the bits of x



Distinguisher D

y

$y_{\text{rev}} = \text{reverse}(y)$

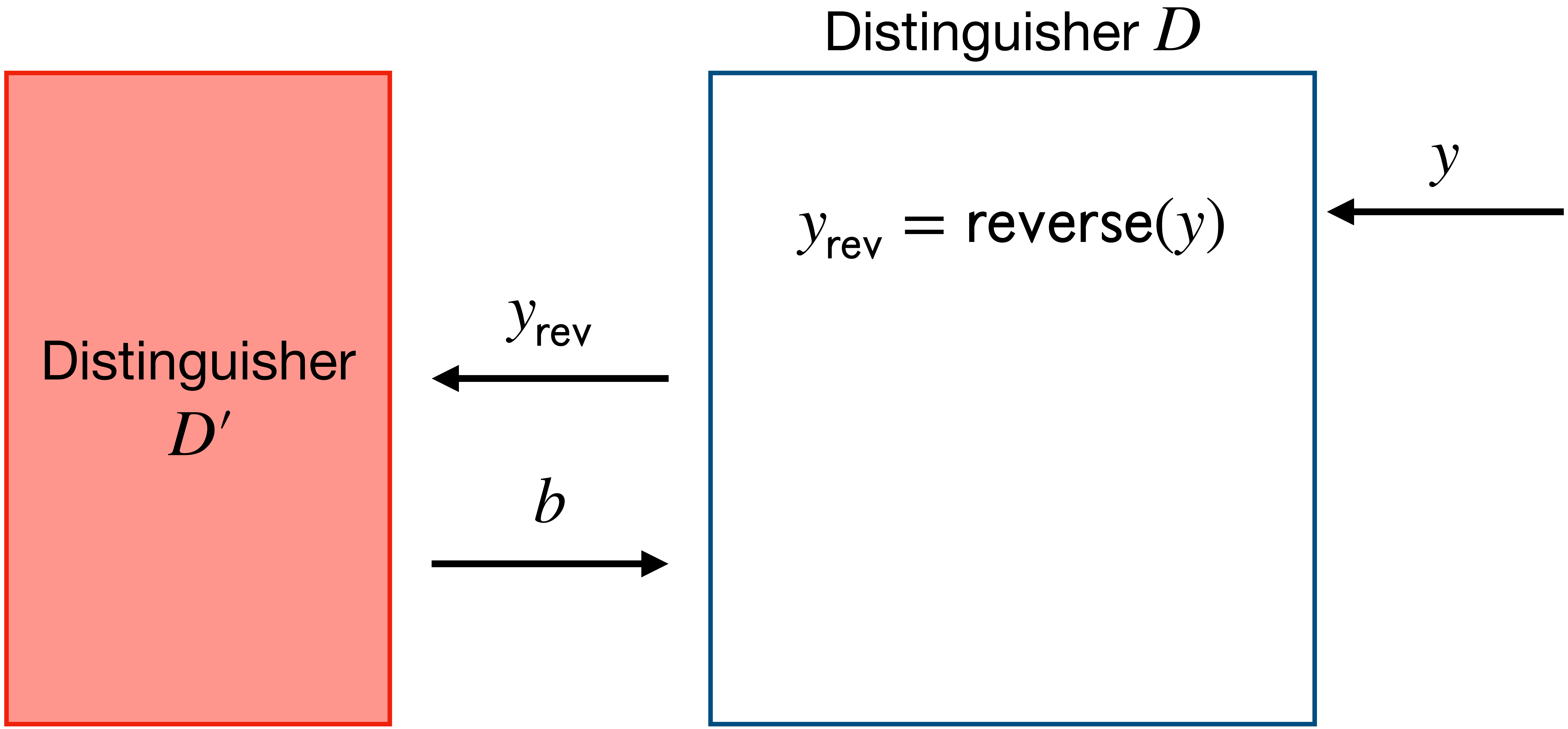
y_{rev}

Distinguisher D'

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$

Let $G'(s) = \text{reverse}(G(s))$ where $\text{reverse}(\text{x})$ reverses the bits of x



Distinguisher D

$y_{\text{rev}} = \text{reverse}(y)$

y

y_{rev}

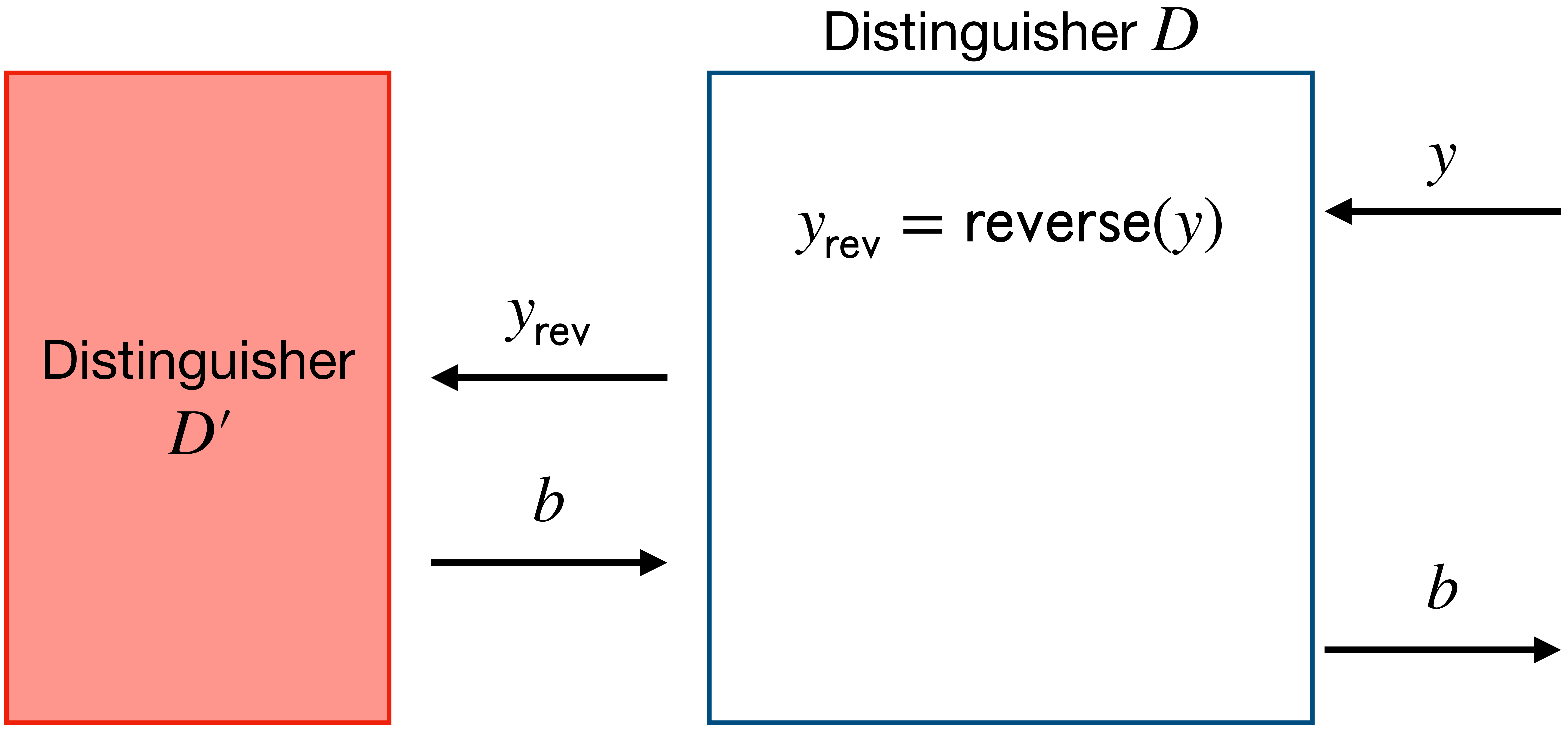
b

Distinguisher D'

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$

Let $G'(s) = \text{reverse}(G(s))$ where $\text{reverse}(\text{x})$ reverses the bits of x



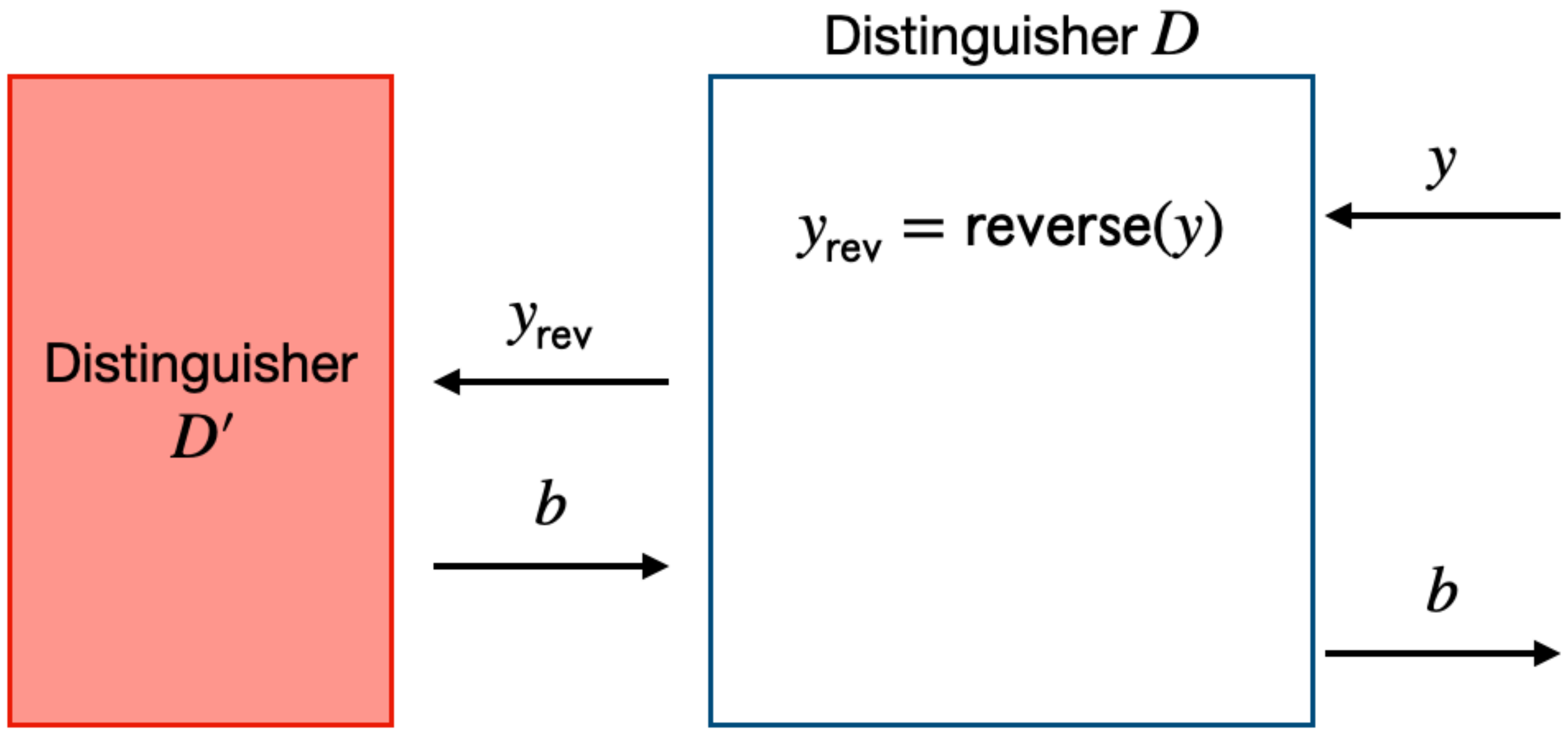
PRG Example

Assume G is a PRG ($G : \{0,1\}^n \rightarrow \{0,1\}^{\ell(n)}$)

Let $G'(s) = \text{reverse}(G(s))$ where reverse(x) reverses the bits of x

Case 1: $y \leftarrow \{0,1\}^{\ell(n)}$

$$\begin{aligned}\Pr_{r\leftarrow U_{\ell(n)}}[D(r) = 1] &= \Pr_{r\leftarrow U_{\ell(n)}}[D'(\text{reverse}(r)) = 1]\\ &= \Pr_{r\leftarrow U_{\ell(n)}}[D'(r) = 1]\end{aligned}$$



Distinguisher D'

y_{rev}

b

Distinguisher D

$y_{\text{rev}} = \text{reverse}(y)$

y

b

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$

Let $G'(s) = \text{reverse}(G(s))$ where reverse(x) reverses the bits of x

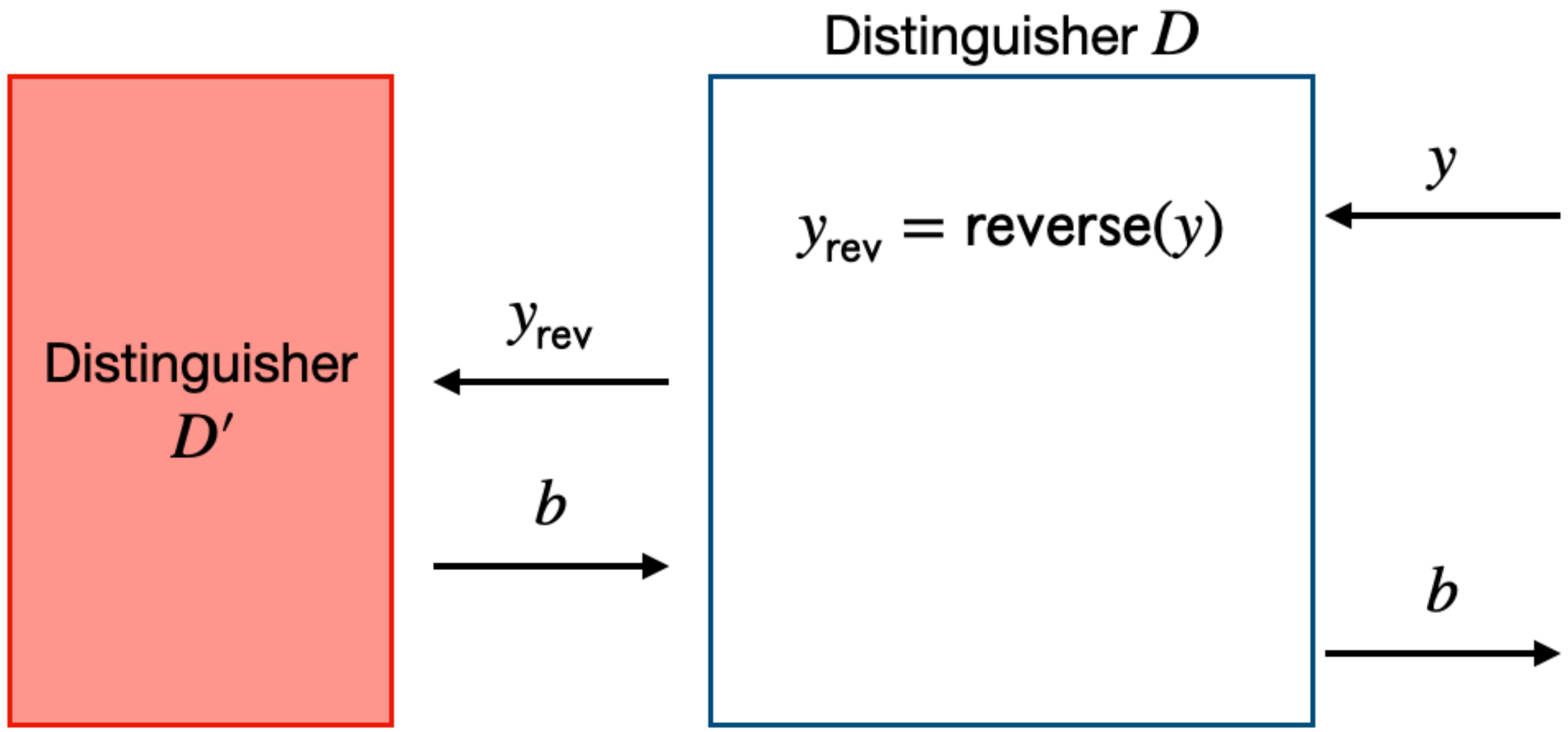
Case 1: $y \leftarrow \{0,1\}^{\ell(n)}$

$$\Pr_{r \leftarrow U_{\ell(n)}}[D(r) = 1] = \Pr_{r \leftarrow U_{\ell(n)}}[D'(\text{reverse}(r)) = 1]$$

$$= \Pr_{r \leftarrow U_{\ell(n)}}[D'(r) = 1]$$

Case 2: $y = G(s)$ where $s \leftarrow \{0,1\}^n$

$$\Pr_{s \leftarrow U_n}[D(G(s)) = 1] = \Pr_{s \leftarrow U_n}[D'(G'(s)) = 1]$$



Distinguisher D

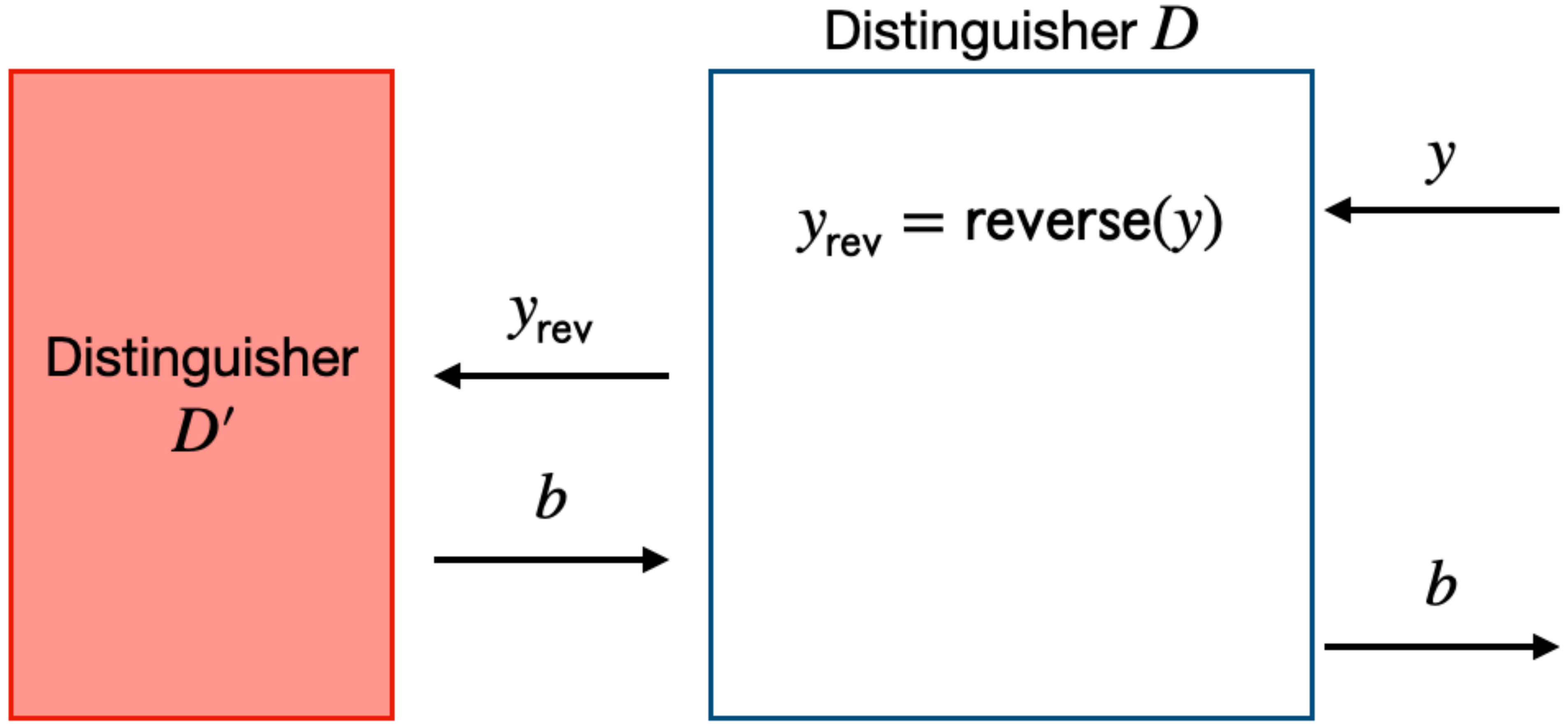
$y_{\text{rev}} = \text{reverse}(y)$

Distinguisher D'

PRG Example

Assume G is a PRG ($G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$

Let $G'(s) = \text{reverse}(G(s))$ where reverse(x) reverses the bits of x



Distinguisher D

$y_{\text{rev}} = \text{reverse}(y)$

y

b

y_{rev}

b

Distinguisher D'

Case 1: $y \leftarrow \{0,1\}^{\ell(n)}$

$$\Pr_{r \leftarrow U_{\ell(n)}}[D(r) = 1] = \Pr_{r \leftarrow U_{\ell(n)}}[D'(\text{reverse}(r)) = 1]$$

$$= \Pr_{r \leftarrow U_{\ell(n)}}[D'(r) = 1]$$

Case 2: $y = G(s)$ where $s \leftarrow \{0,1\}^n$

$$\Pr_{s \leftarrow U_n}[D(G(s)) = 1] = \Pr_{s \leftarrow U_n}[D'(G'(s)) = 1]$$

If $\left| \Pr_{s \leftarrow \{0,1\}^n}\left[D'\left(G'(s)\right) = 1\right] - \Pr_{r \leftarrow \{0,1\}^{\ell(n)}}\left[D'(r) = 1\right] \right| \geq 1/n^c$ for some c, then the difference between Case 1 and 2 would also be non-negligible. But since we assume G is secure, then no such distinguisher can exist. Therefore G' is secure if G is secure

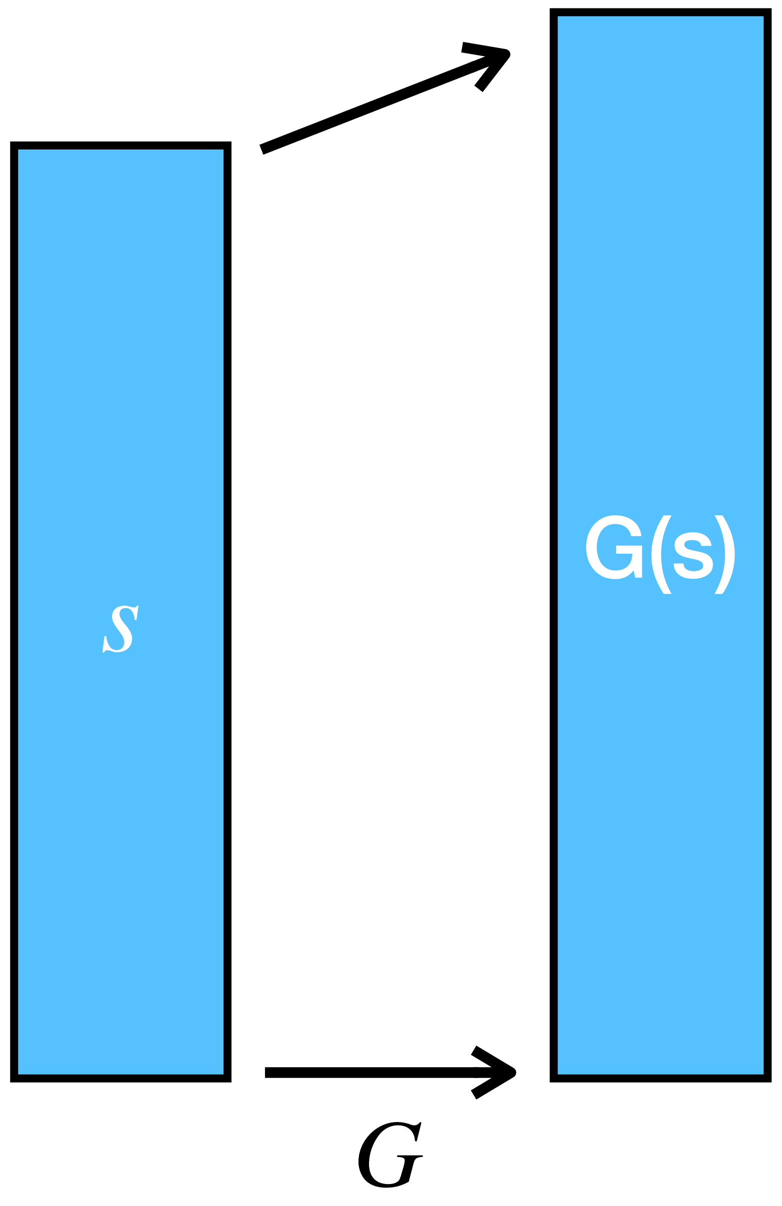
Increasing the Stretch of a PRG

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

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

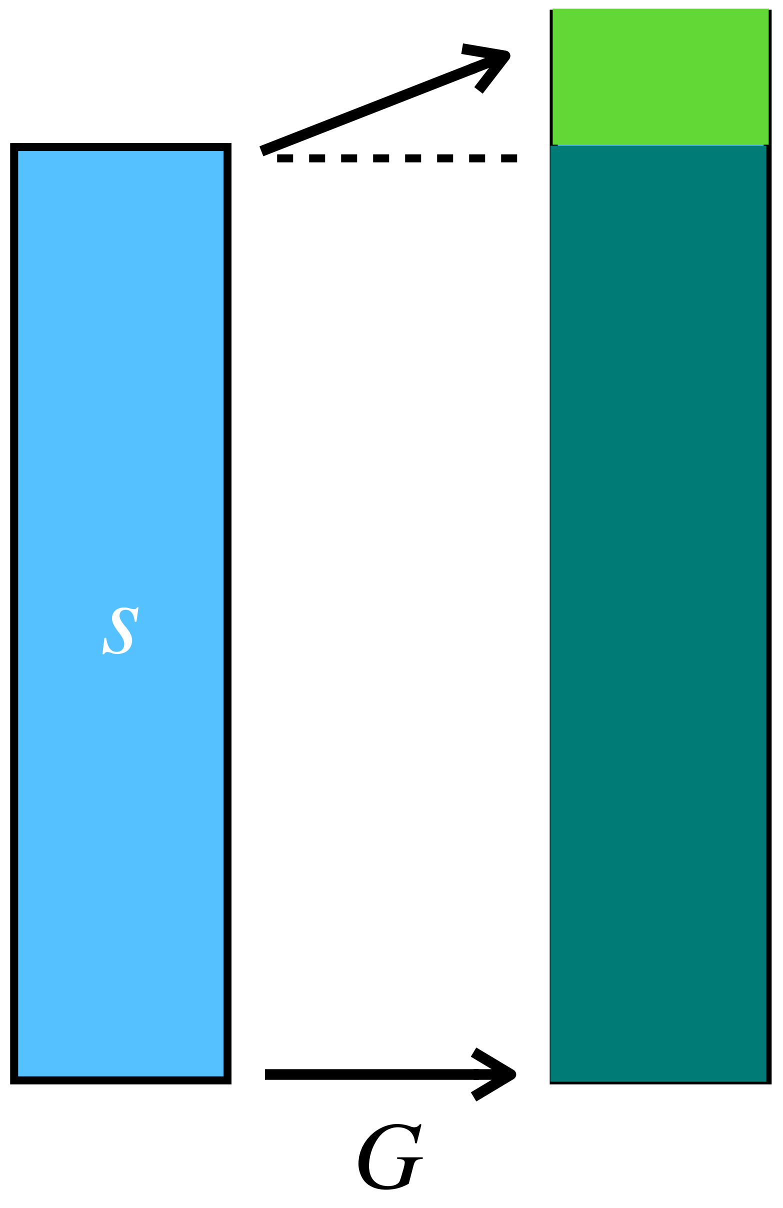
Proof sketch:



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

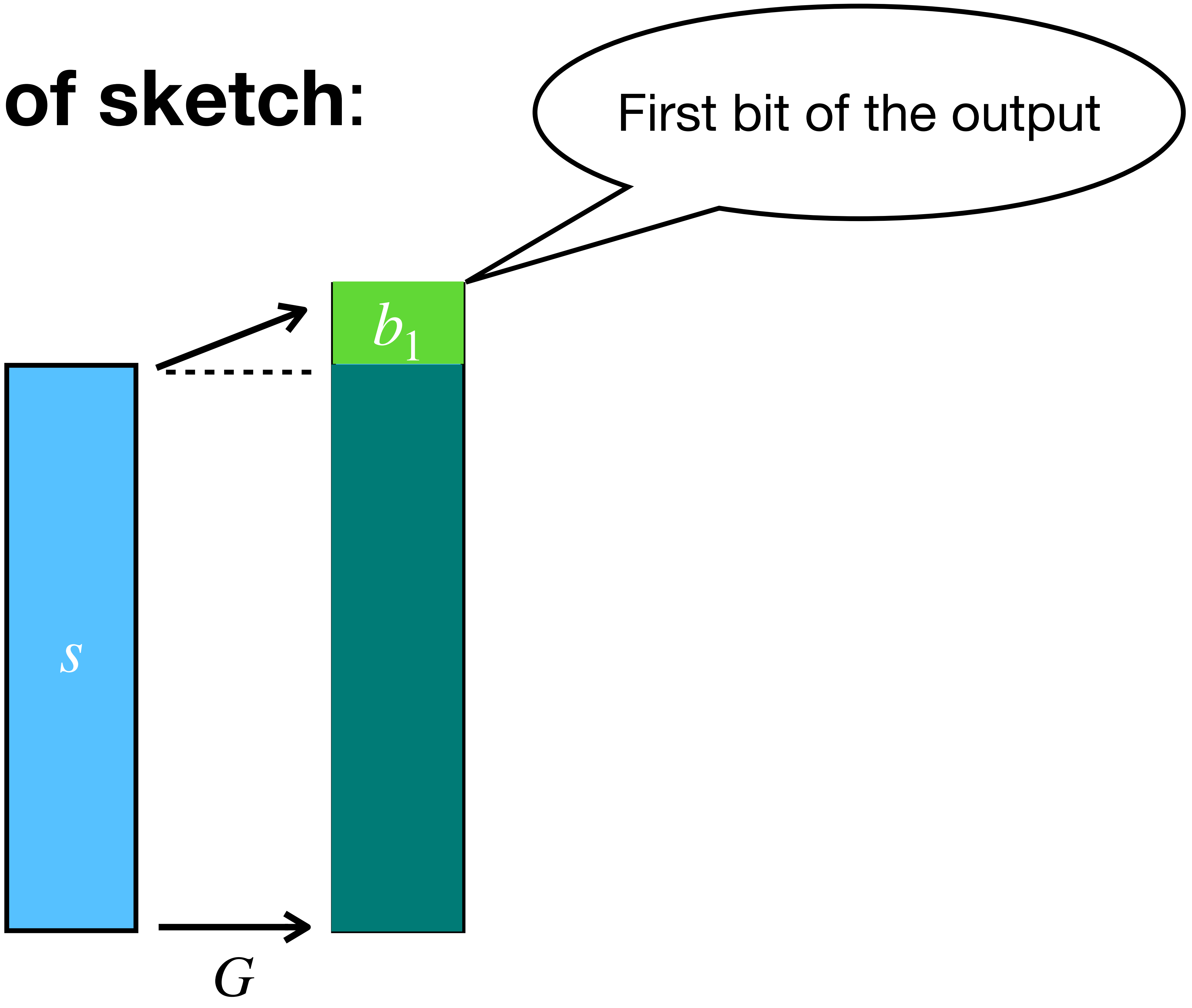
Proof sketch:



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

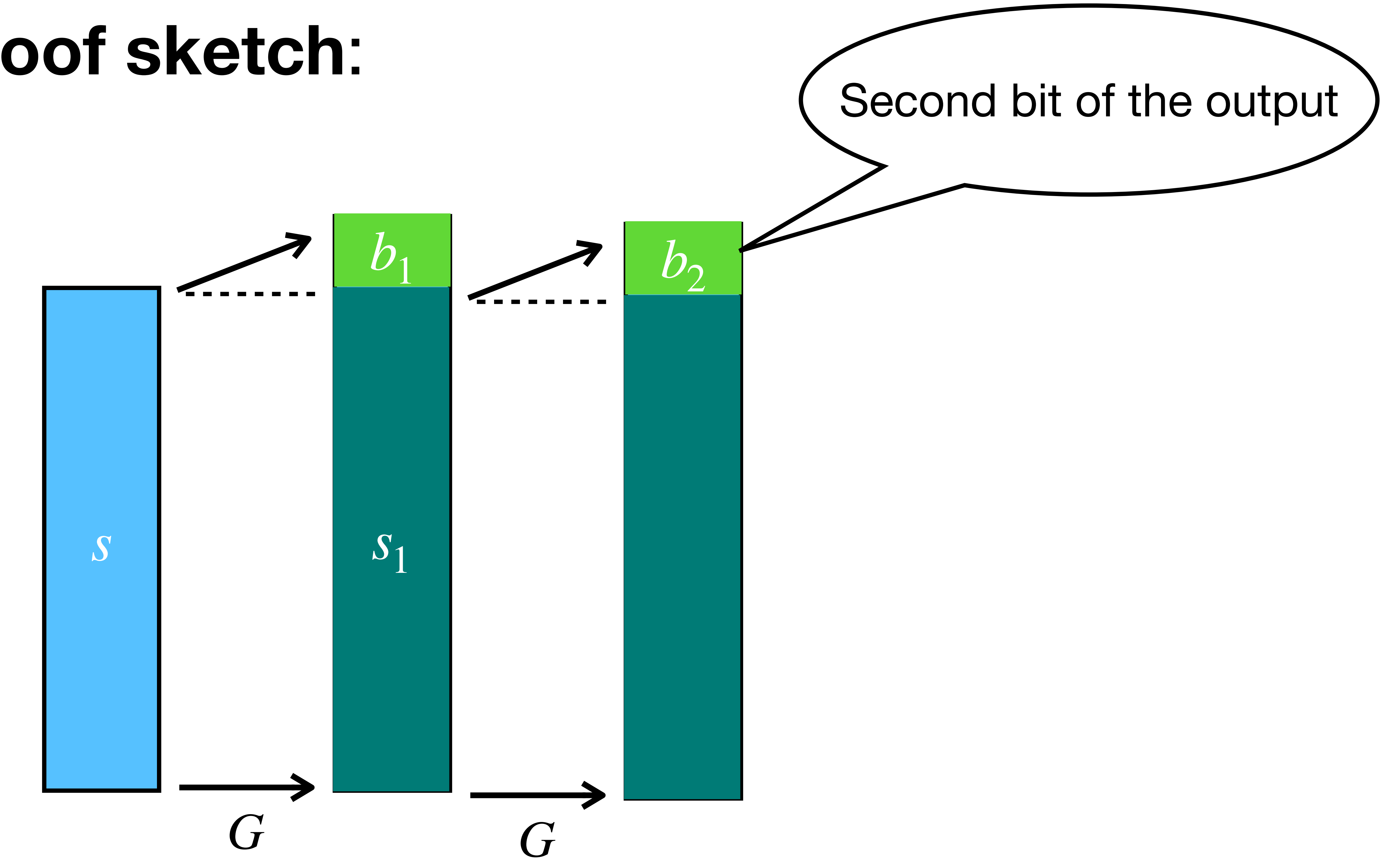
Proof sketch:



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

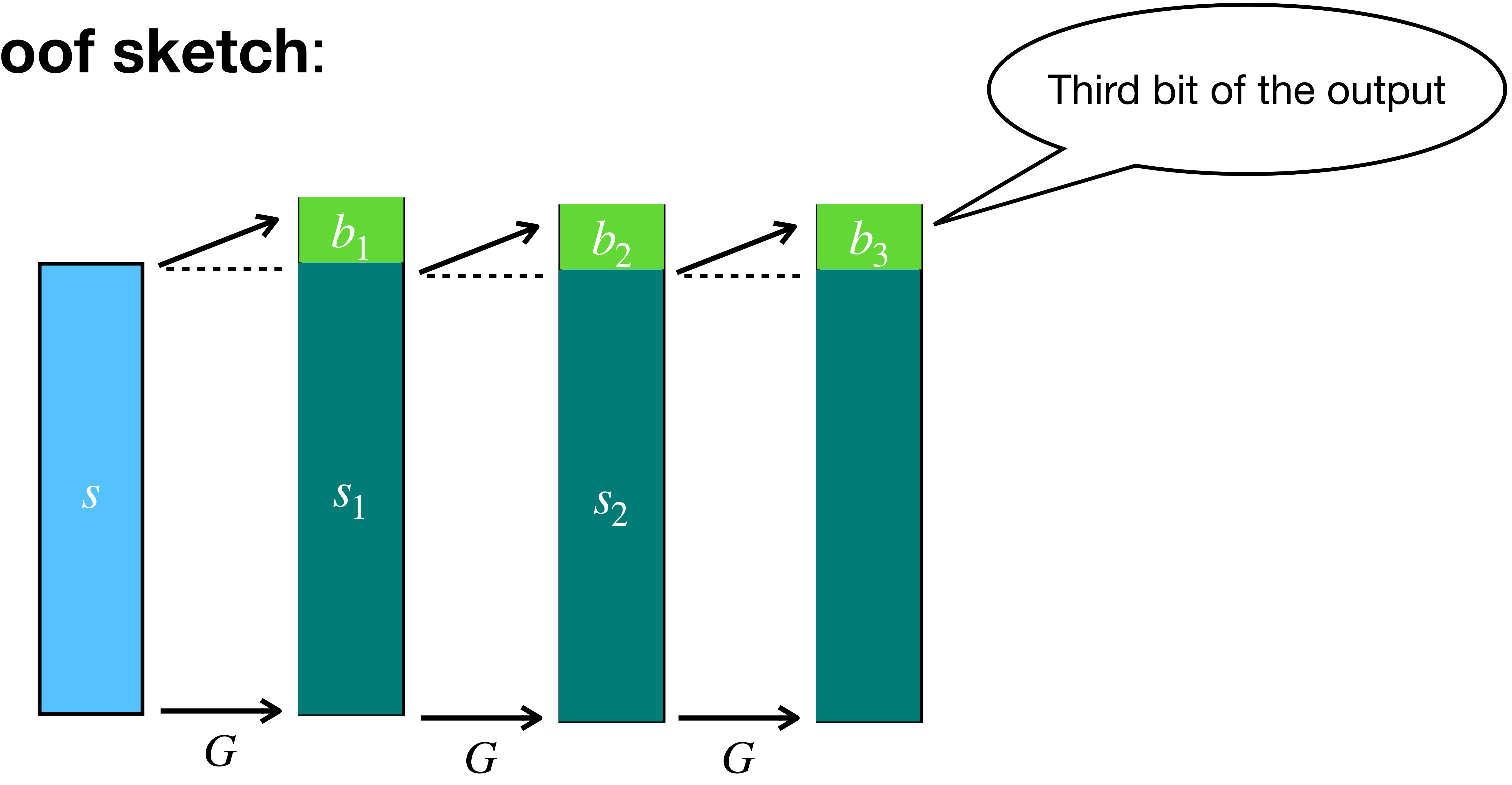
Proof sketch:



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

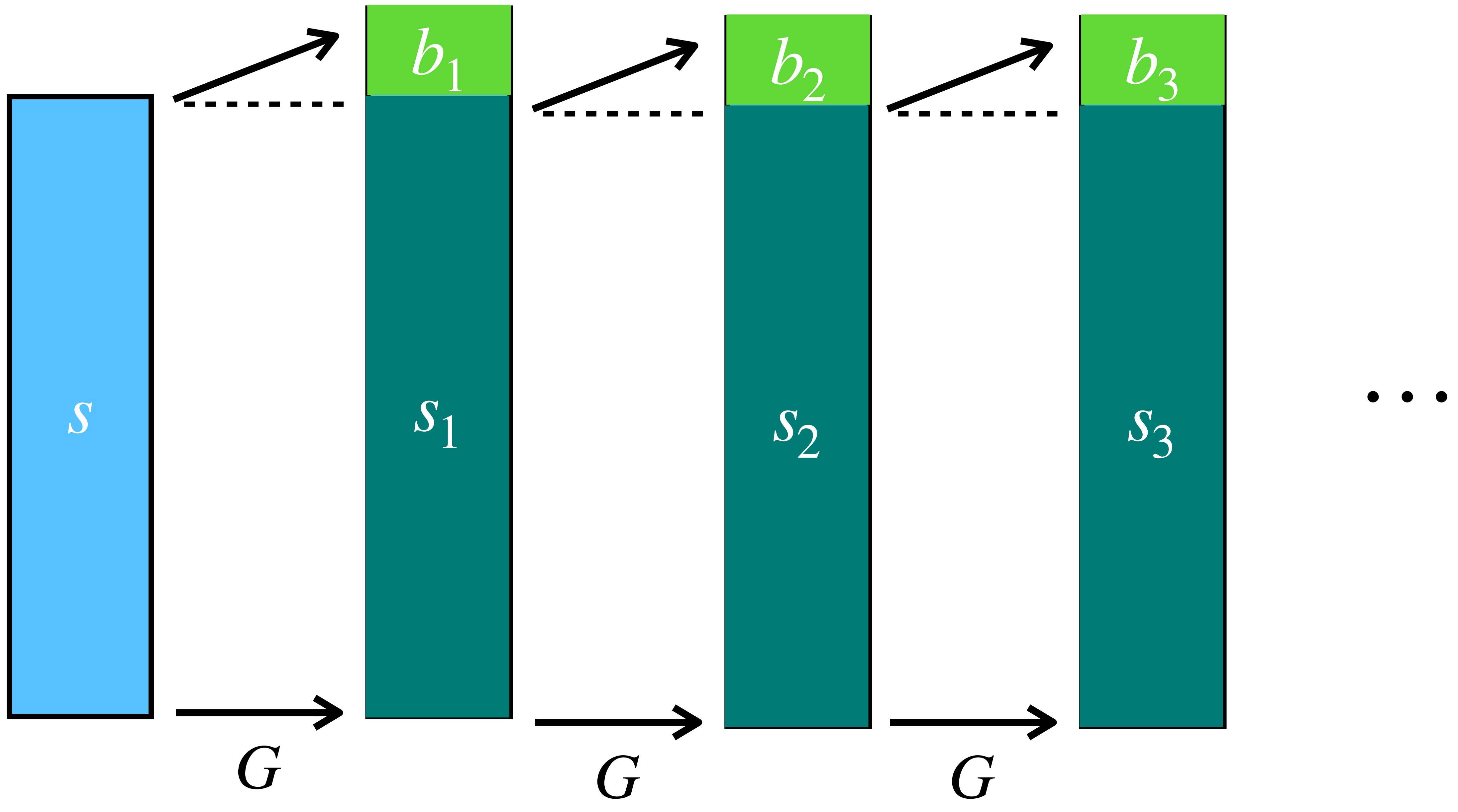
Proof sketch:



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:

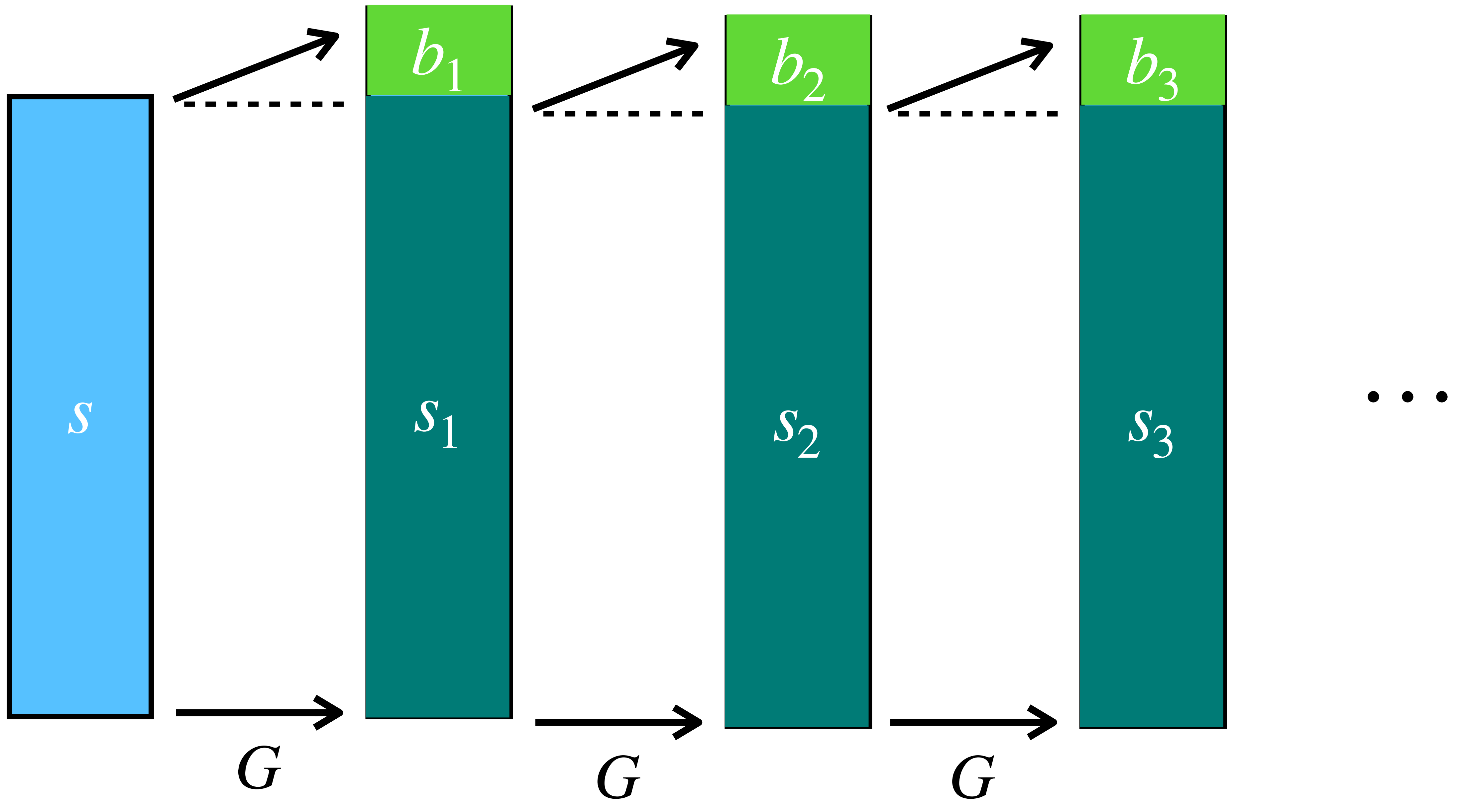


We can continue this polynomially many times

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:



Features:

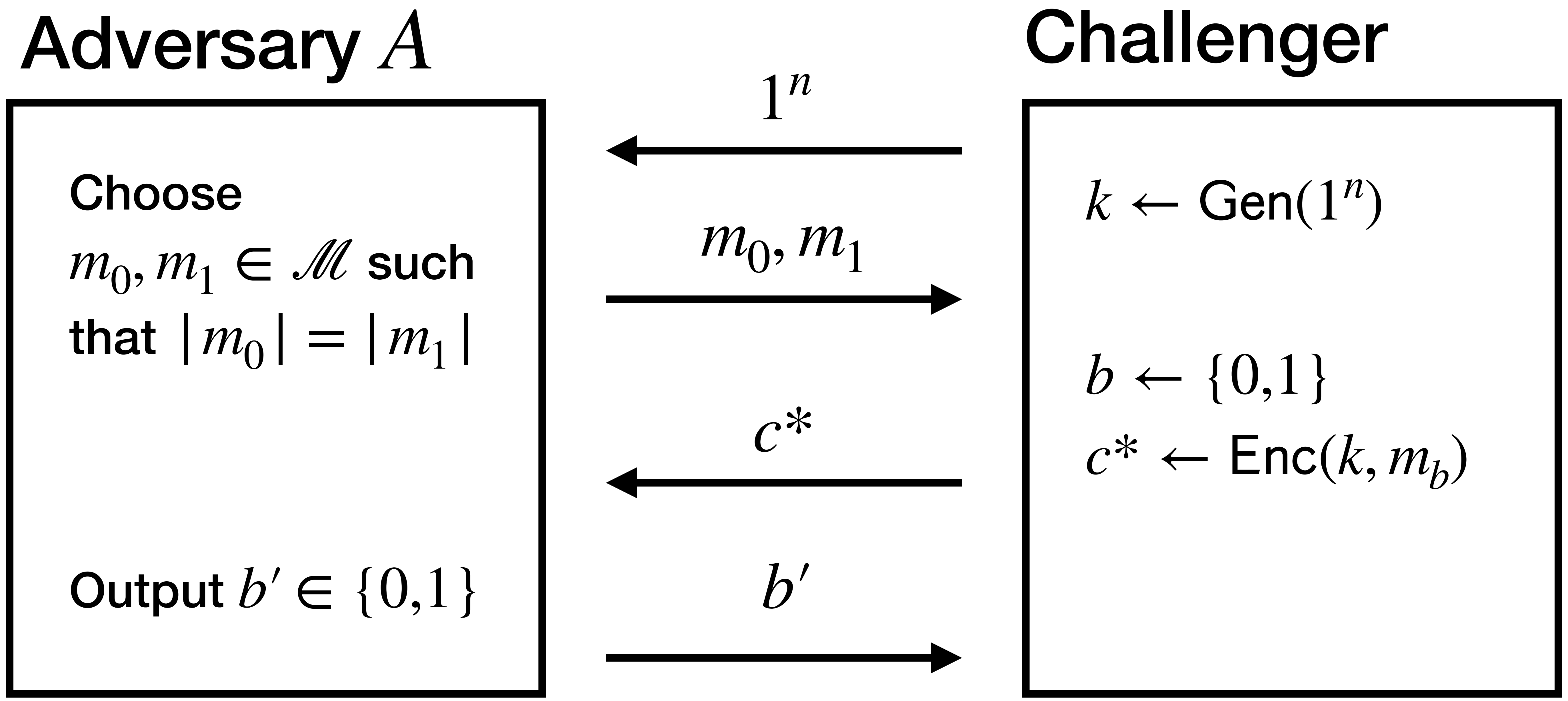
- Can keep outputting bits without knowing ℓ ahead of time (stream cipher!)
- At any point can output s_ℓ

Security via a reduction to the security of G (hybrid argument)

Detour: Semantic Security

Recall: Indistinguishable Encryptions

Given $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ and an adversary A, consider the experiment $\text{PrivK}^{\text{eav}}_{\Pi,A}(n)$:



Adversary A

Choose $m_0, m_1 \in \mathscr{M}$ such that $|m_0| = |m_1|$

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

Challenger

$k \leftarrow \text{Gen}(1^n)$

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

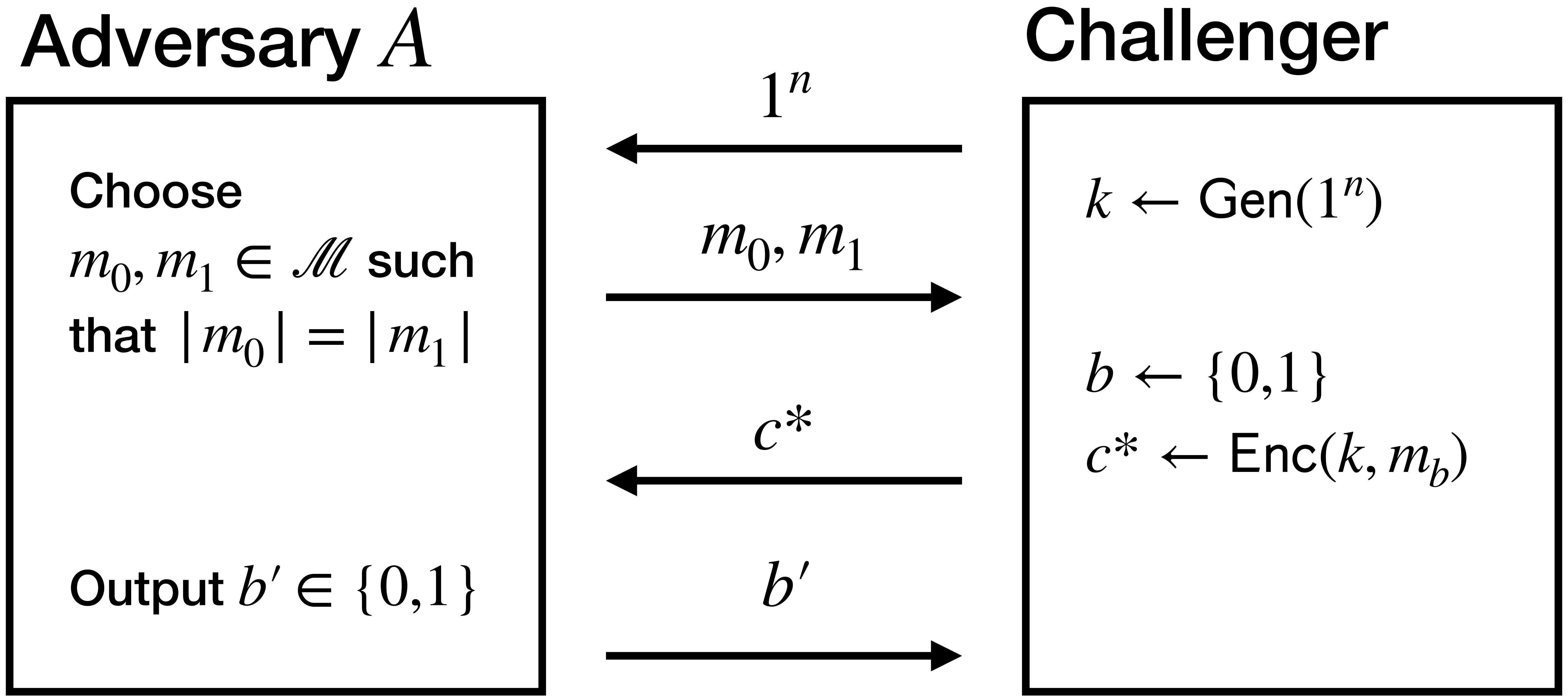
$c* \leftarrow \text{Enc}(k, m_b)$

A wins if $b' = b$

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

Recall: Indistinguishable Encryptions

Given $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and an adversary A, consider the experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\Pi,A}(n)$:



Adversary A

Choose
$m_0, m_1 \in \mathcal{M}$ such
that $|m_0| = |m_1|$

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

1^n

m_0, m_1

c^*

b'

Challenger

$k \leftarrow \mathsf{Gen}(1^n)$

$b \leftarrow \{0,1\}$
$c^* \leftarrow \mathsf{Enc}(k, m_b)$

Intuition: Adversary can't tell if a ciphertext corresponds to m_0 or m_1

A wins if $b' = b$

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

Semantic Security

Semantic security [Goldwasser-Micali ’82]:

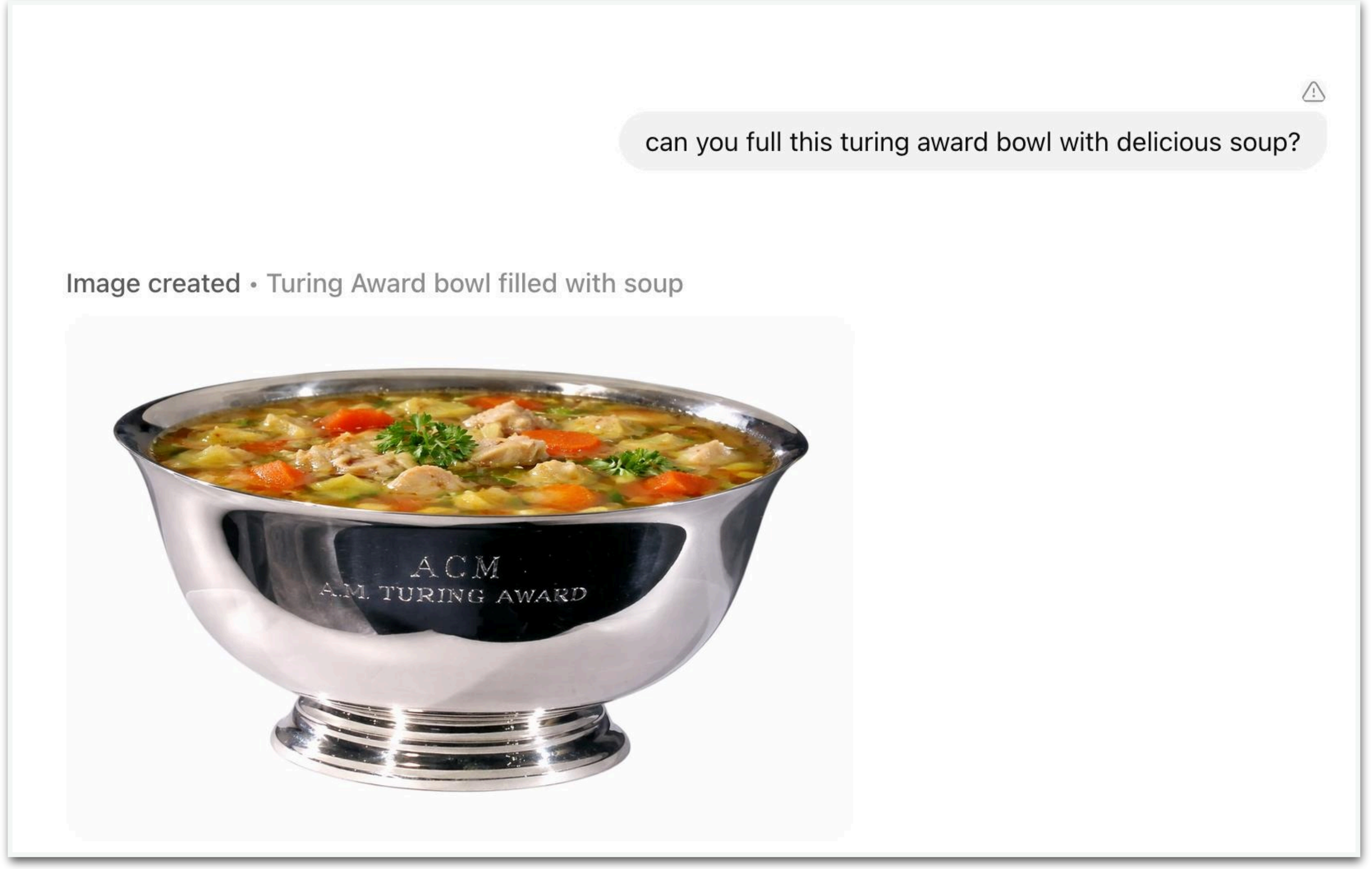
“Whatever” can be computed efficiently given the ciphertext can essentially be computed efficiently without the ciphertext

Theorem: Π is **semantically secure** if and only if it has **indistinguishable encryptions**.

Why do we need both notions?

- Semantic security explains “what security means”
- Indistinguishability of encryptions is “easier to work with” when we’re trying to prove security

Shafi Goldwasser and Silvio Micali won the Turing award in 2012 for pioneering provable security



can you full this turing award bowl with delicious soup?

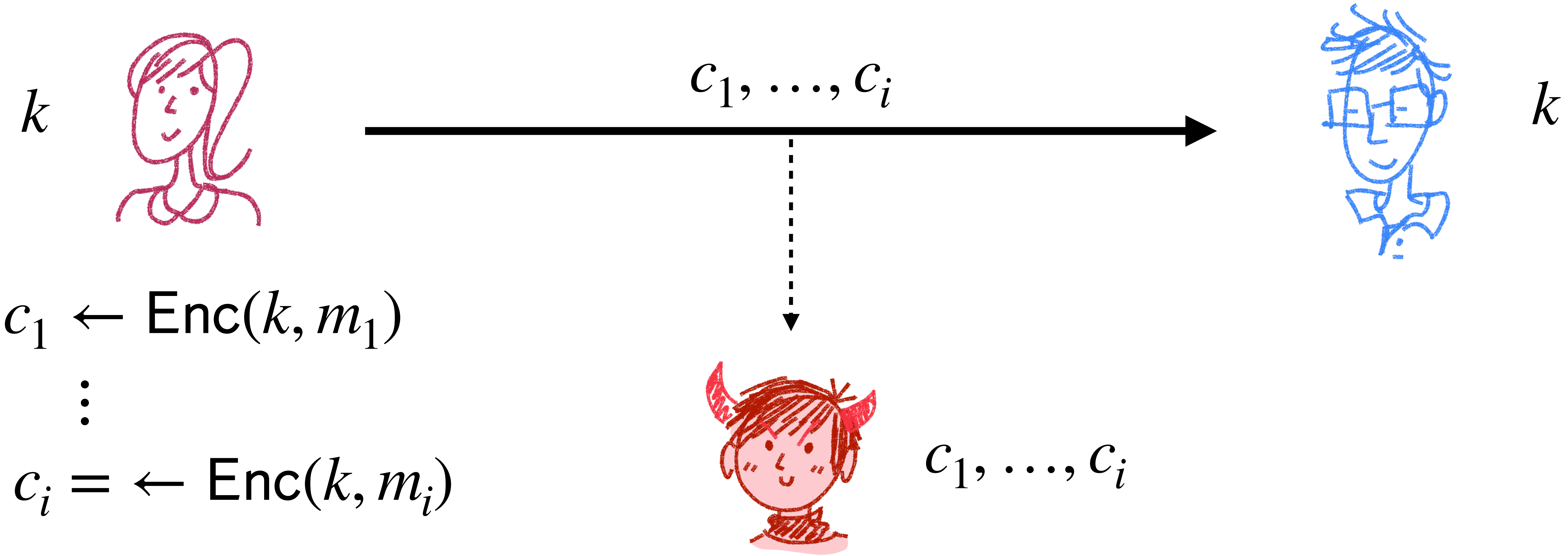
Image created • Turing Award bowl filled with soup

My friend informed me via this screenshot that the physical Turing award is a bowl

Security for Multiple Messages

Security for Multiple Messages

Often you want to be able to encrypt many messages with the same key.



k

$c_1, \ldots, c_i$

k

$c_1 \leftarrow \mathsf{Enc}(k, m_1)$

$\vdots$

$c_i = \leftarrow \mathsf{Enc}(k, m_i)$

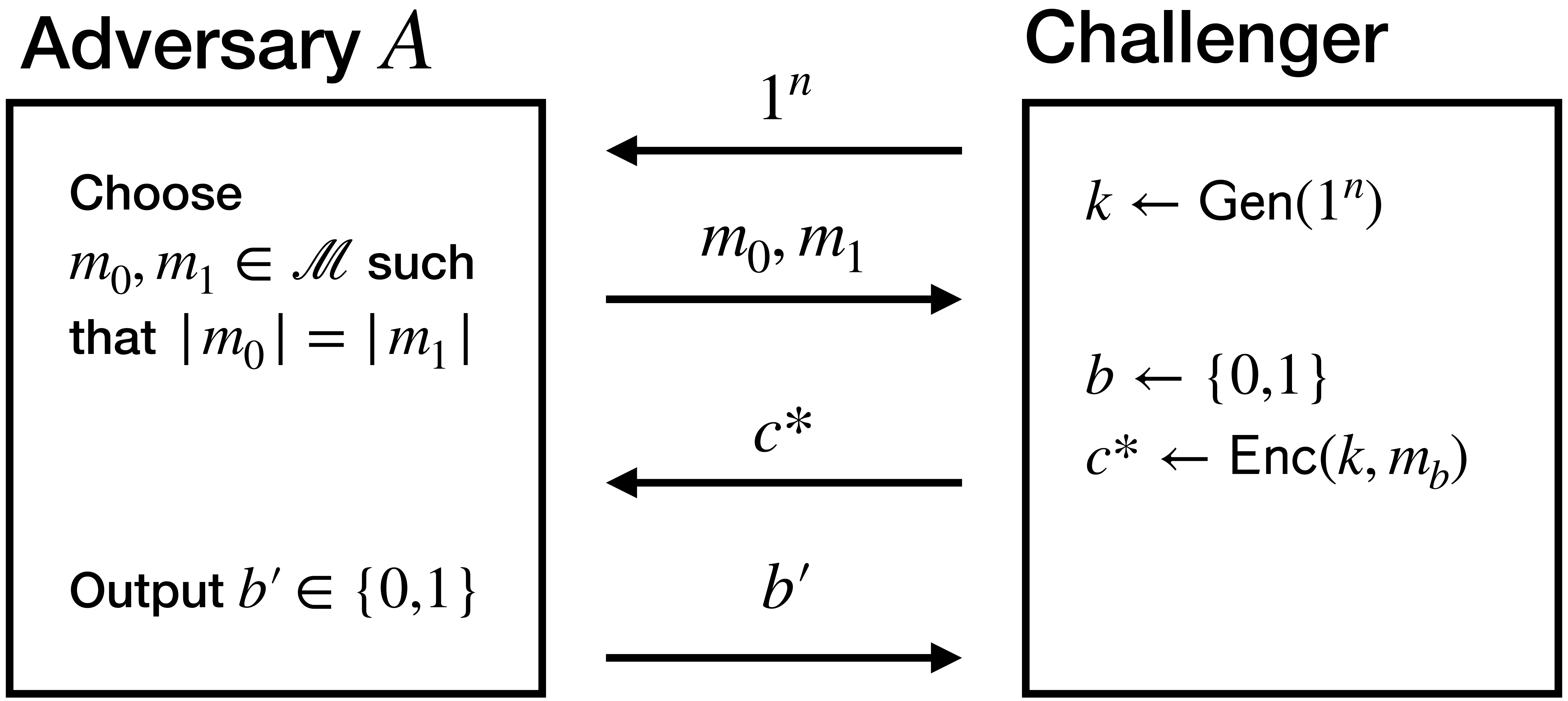
$c_1, \ldots, c_i$

Security for Multiple Messages

- Can we encrypt multiple messages?
  - Note that our OTP using a PRG from before completely breaks down
- One option: Keep state between encryptions
  - E.g., in a TLS connection we know how many bits were encrypted
  - However, keeping state is often very problematic
  - Undesirable to base security on state maintenance (e.g., multiple TLS connections)
- Can we use a stateless encryption?
- How might we define security for multiple messages?

Starting Point: Indistinguishable Encryptions

Given $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and an adversary A, consider the experiment $\mathsf{PrivK}^{\mathsf{eav}}_{\Pi,A}(n)$:



Adversary A

Choose $m_0, m_1 \in \mathcal{M}$ such that $|m_0| = |m_1|$

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

1^n

m_0, m_1

$c*$

b'

Challenger

$k \leftarrow \mathsf{Gen}(1^n)$

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

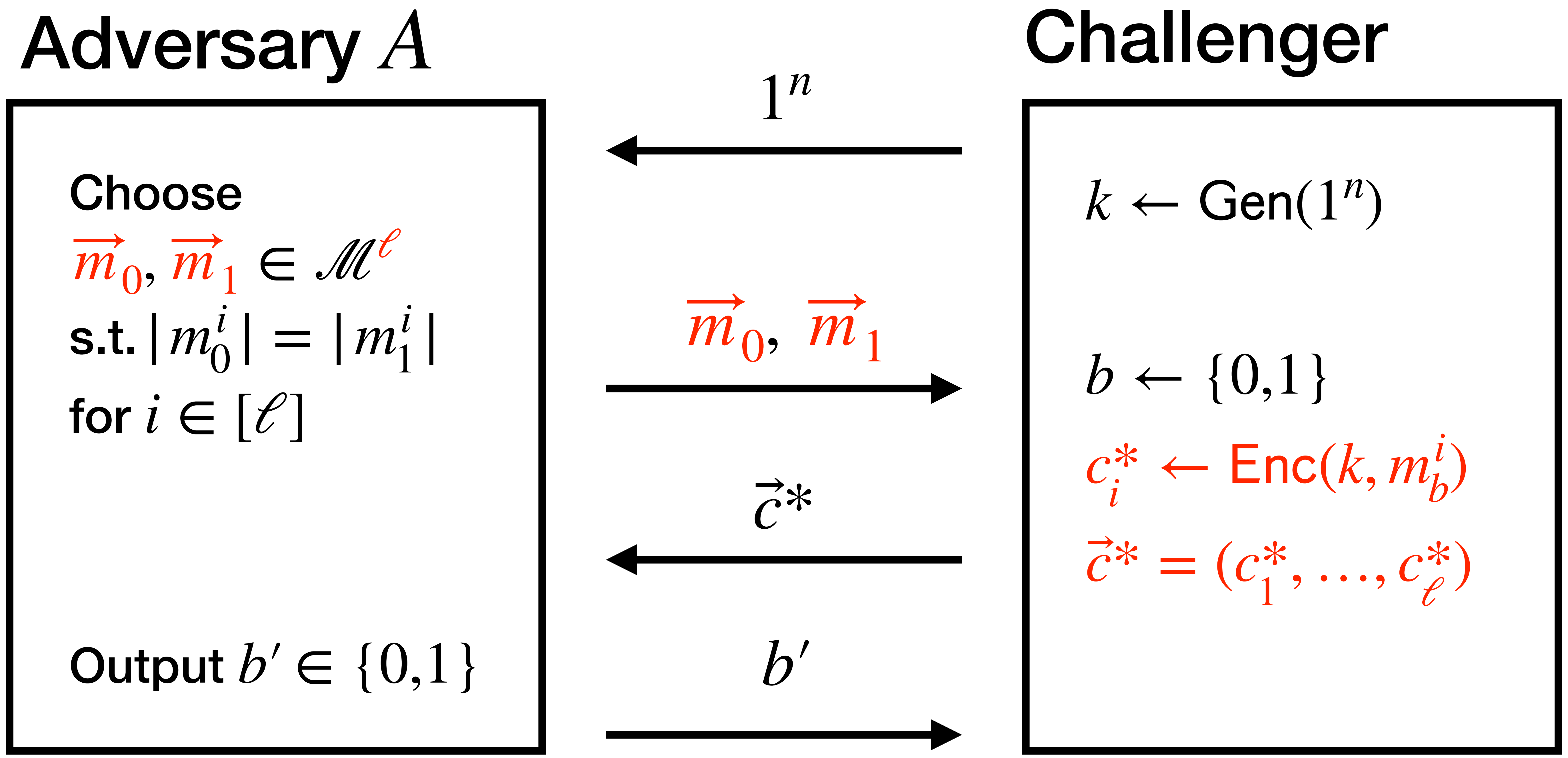
$c* \leftarrow \mathsf{Enc}(k, m_b)$

A wins if $b' = b$

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

Encrypting Multiple Messages

Given $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and an adversary A, consider the experiment $\mathsf{PrivK}^{\mathsf{mult}}_{\Pi,A}(n)$:



Adversary A

Choose
$\vec{m}_0, \vec{m}_1 \in \mathcal{M}^{\ell}$
s.t. $|m_0^i| = |m_1^i|$
for $i \in [\ell]$

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

1^n

$\vec{m}_0, \ \vec{m}_1$

$\vec{c}*$

b'

Challenger

$k \leftarrow \mathsf{Gen}(1^n)$

$b \leftarrow \{0,1\}$
$c_i^* \leftarrow \mathsf{Enc}(k, m_b^i)$
$\vec{c}* = (c_1^*, \ldots, c_\ell^*)$

A wins if $b' = b$

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

Encrypting Multiple Messages

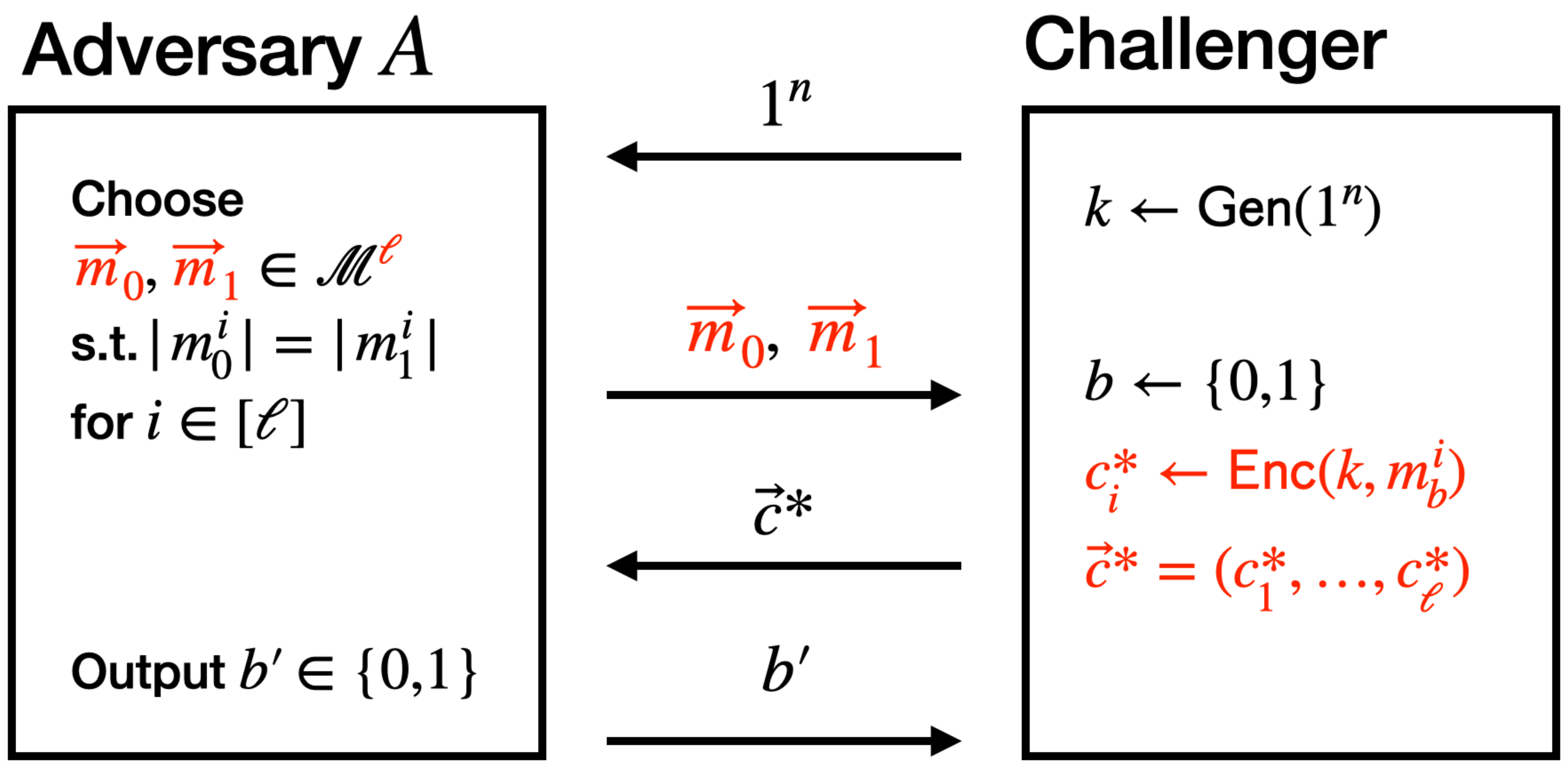
Given $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and an adversary A, consider the experiment $\mathsf{PrivK}^{\mathsf{mult}}_{\Pi,A}(n)$:

Definition:

Π has multiple-messages indistinguishable encryptions in the presence of an eavesdropper if for every PPT adversary A there exists a negligible function $\epsilon(\cdot)$ such that

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

where the probability is taken over the random coins used by A and by the experiment.



Adversary A

Choose
$\vec{m}_0, \vec{m}_1 \in \mathcal{M}^{\ell}$
s.t. $|m_0^i| = |m_1^i|$
for $i \in [\ell]$

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

Challenger

$k \leftarrow \mathsf{Gen}(1^n)$

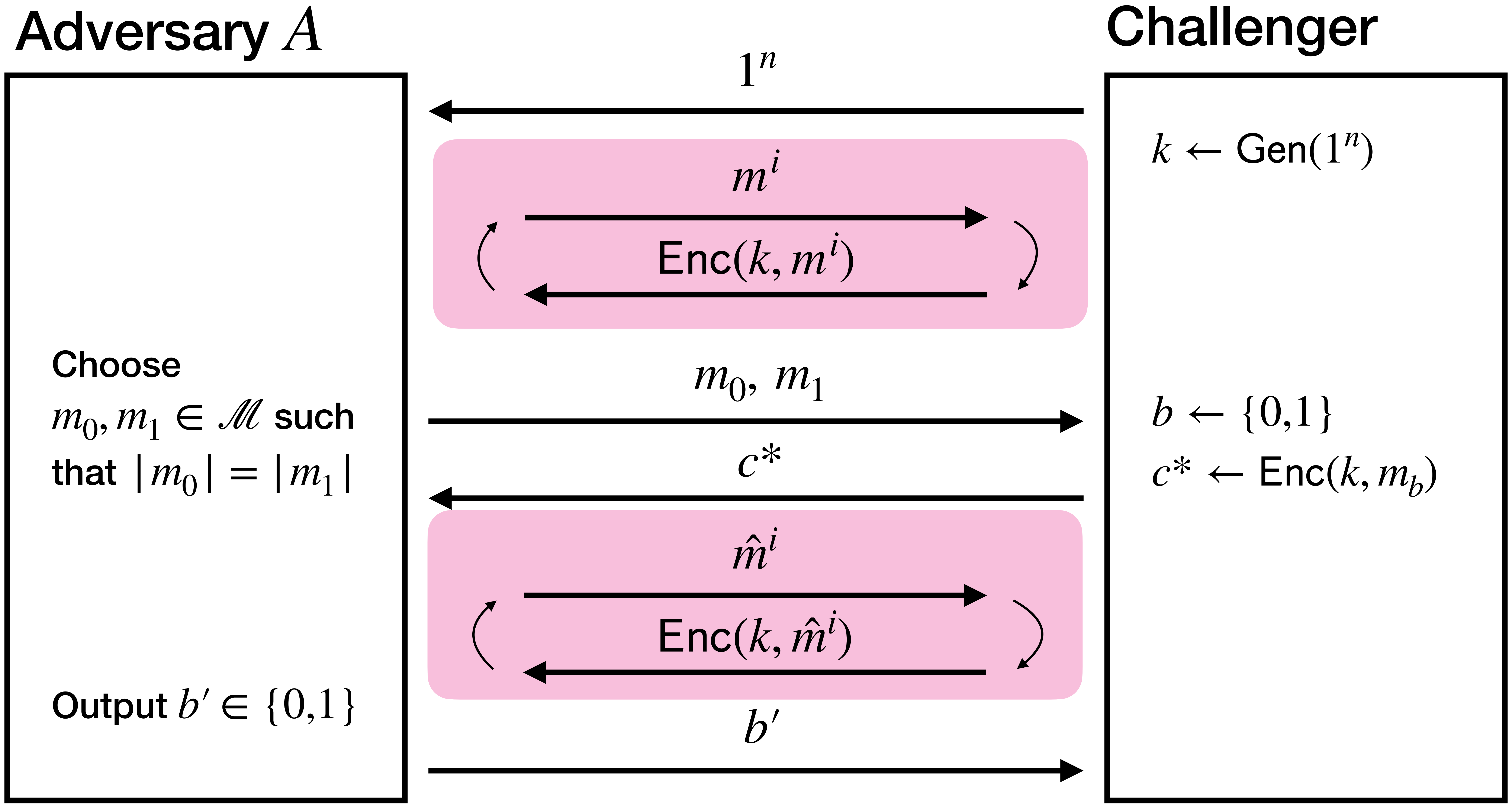
$b \leftarrow \{0,1\}$
$c_i^* \leftarrow \mathsf{Enc}(k, m_b^i)$
$\vec{c}* = (c_1^*, \ldots, c_\ell^*)$

A wins if $b' = b$

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

Chosen-Plaintext Attack (CPA)

Chosen-Plaintext Attack (CPA)



Adversary A

Choose $m_0, m_1 \in \mathscr{M}$ such that $|m_0| = |m_1|$

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

Challenger

$k \leftarrow$ Gen(1^n)

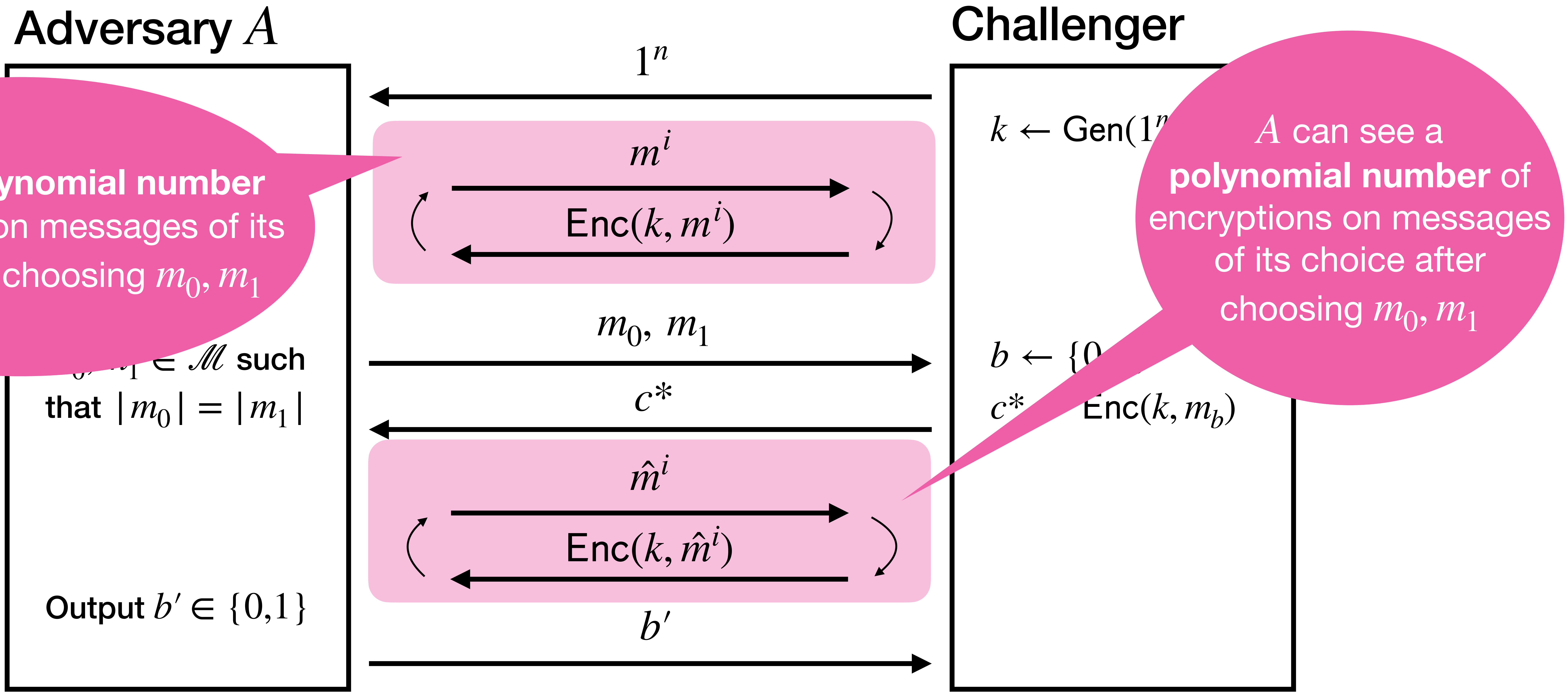
$b \leftarrow \{0,1\}$

$c* \leftarrow$ Enc(k, m_b)

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

and 0 otherwise

Chosen-Plaintext Attack (CPA)



Adversary A

Challenger

1^n

$k \leftarrow$ Gen(1^n

m^i

Enc(k, m^i)

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

$\in \mathscr{M}$ such that $|m_0| = |m_1|$

$m_0,\ m_1$

$b \leftarrow \{0$

$c*$

$c*$ Enc(k, m_b)

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

$\hat{m}^i$

Enc($k, \hat{m}^i$)

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

b'

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

and 0 otherwise

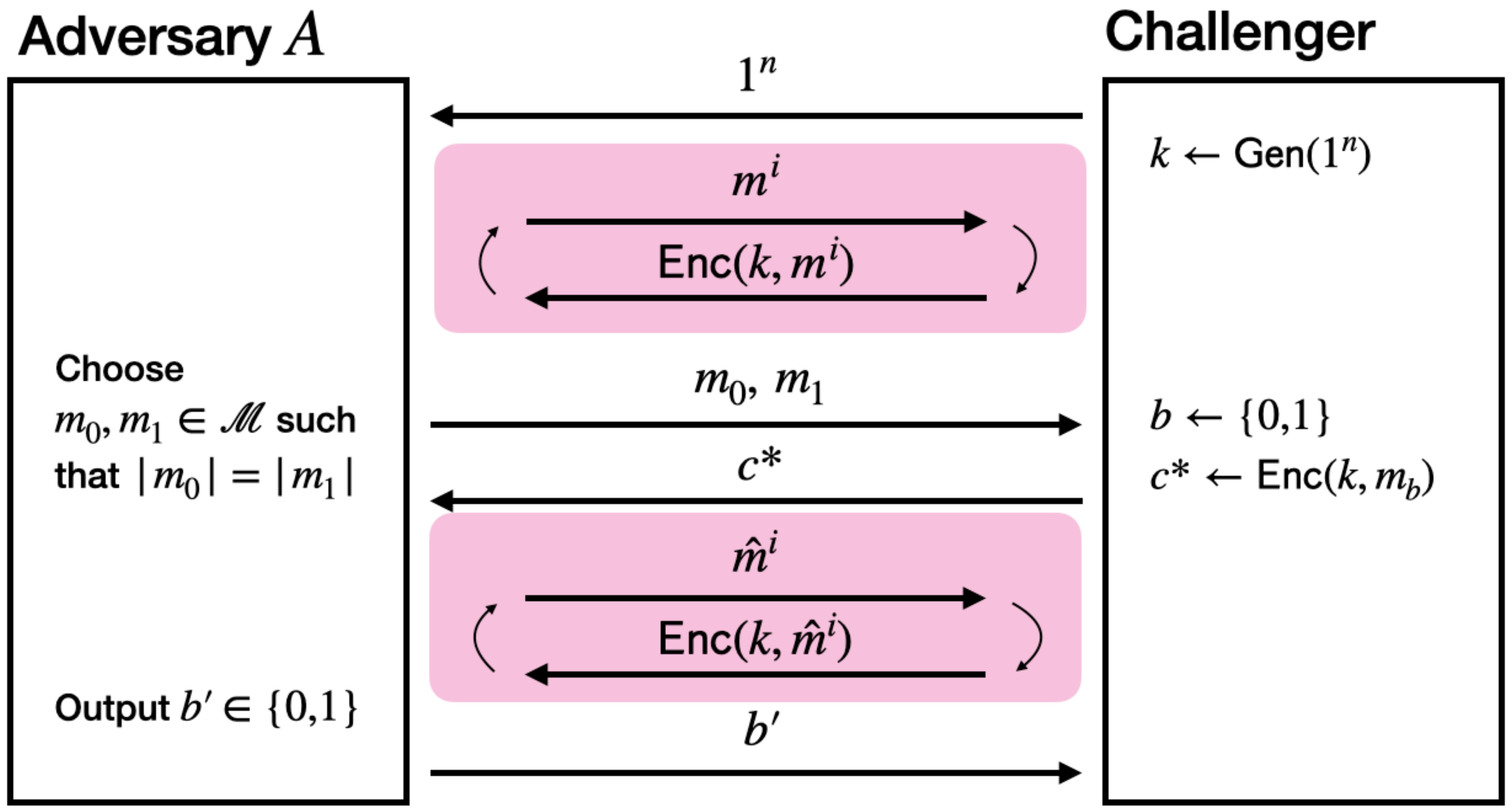
We typically refer to these types of queries as having "oracle access"

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

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



Adversary A

Challenger

$k \leftarrow \mathsf{Gen}(1^n)$

1^n

m^i

$\mathsf{Enc}(k, m^i)$

Choose $m_0, m_1 \in \mathcal{M}$ **such that** $|m_0| = |m_1|$

$m_0,\ m_1$

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

$c^* \leftarrow \mathsf{Enc}(k, m_b)$

c^*

$\hat{m}^i$

$\mathsf{Enc}(k, \hat{m}^i)$

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

b'

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

Notes:

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

Is it realistic to give A oracle access to Enc?

- May 1942: US Navy cryptanalysts discovered that Japan was planning an attack on Midway island in the Central Pacific
  - They had learned this by intercepting a communication message containing a ciphertext fragment “AF” that they believed corresponded to the plaintext “Midway island”
  - They were not able to convince Washington planners that this was the case
- Navy cryptanalysts instructed US forces at Midway to send a plaintext message that their freshwater supplies were low.
  - The Japanese intercepted this message and reported to their superiors that “AF” was low on water

Is it realistic to give A oracle access to Enc?

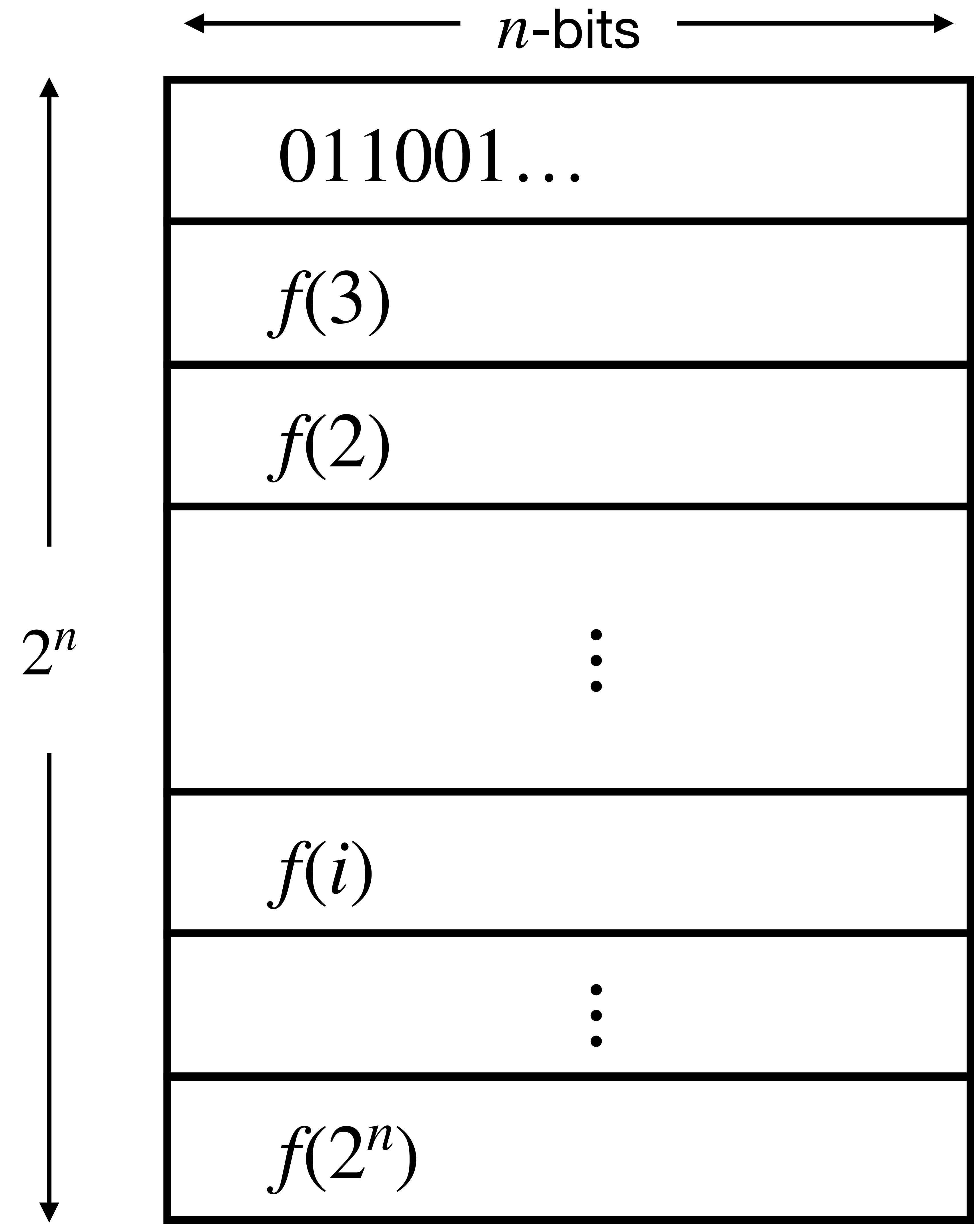
- Cryptanalysts at Bletchley Park would sometimes ask the Royal Air Force to lay mines at specific positions, hoping the Germans would encrypt a “warning” message and an “all clear” message after they were removed

- A daily weather report was transmitted by the Germans at the same time every day, containing the word “Wetter” (German for “weather”) at the same location in every message

Pseudorandom Functions (PRFs)

What if we had a truly random function?

- Suppose our secret key consisted of many random pads
    - Can think of as a truth table of a uniformly random function f chosen from the set of all n-bit to n-bit functions



n-bits

2^n

011001…

$f(3)$

$f(2)$

$\vdots$

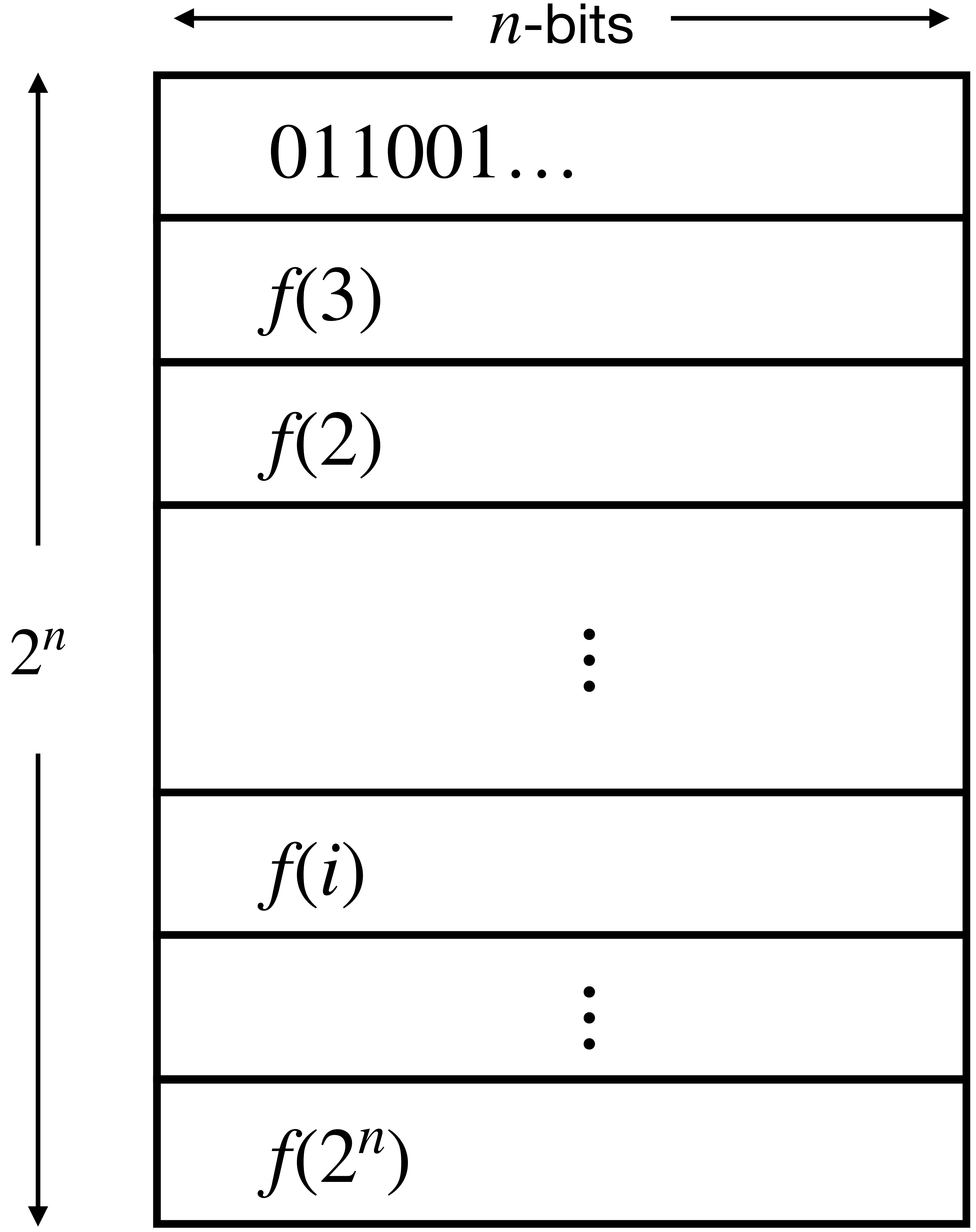
$f(i)$

$\vdots$

$f(2^n)$

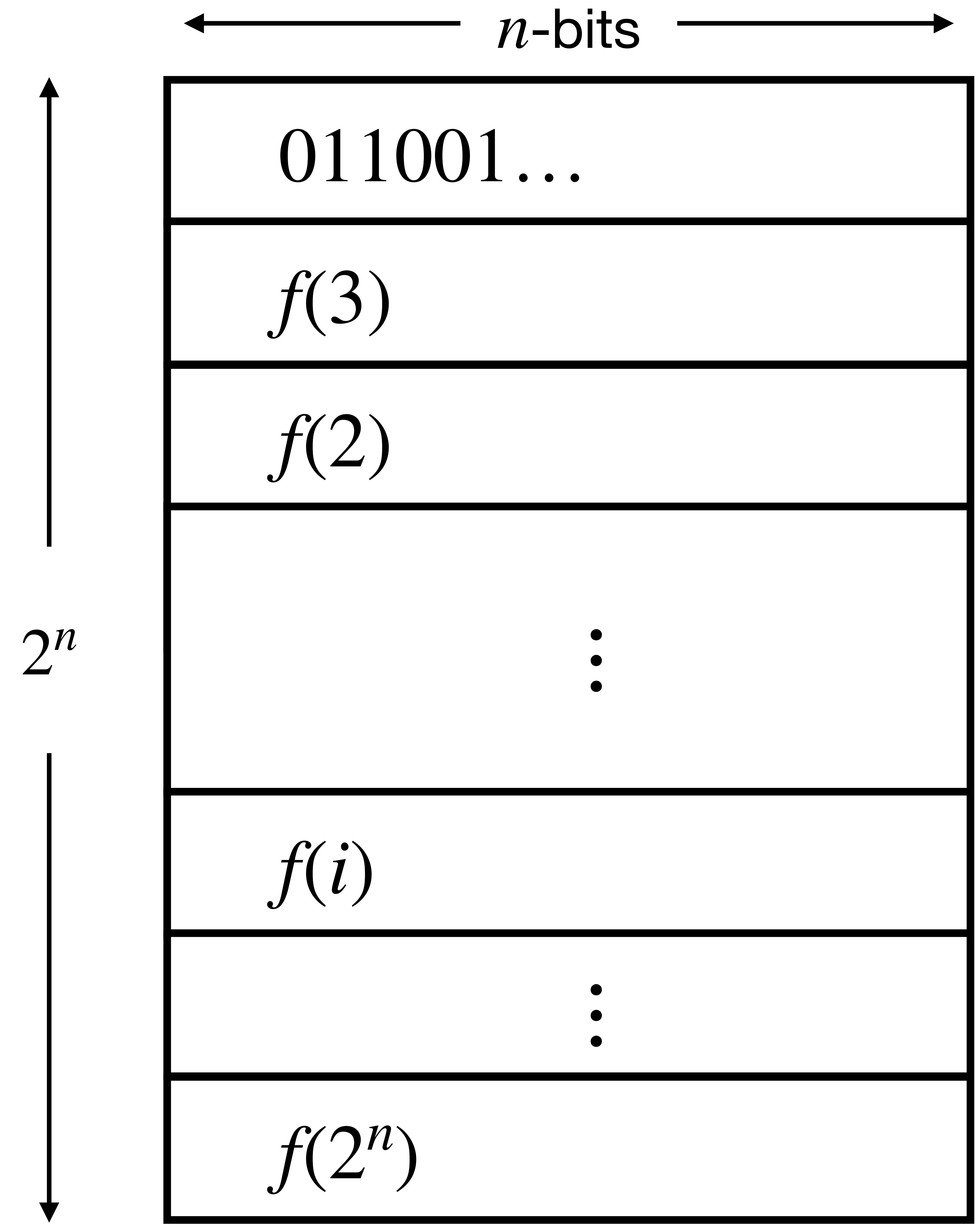
What if we had a truly random function?

- Suppose our secret key consisted of many random pads
    - Can think of as a truth table of a uniformly random function f chosen from the set of all n-bit to n-bit functions
- To encrypt, Alice can choose a random index $i \in \{0,1\}^n$ and send $(i, f(i) \oplus m)$
    - This is like one-time pad, but there's only a negligible probability we reuse the same index
    - Bob has the same truth table/key and can decrypt



What if we had a truly random function?

- Suppose our secret key consisted of many random pads
  - Can think of as a truth table of a uniformly random function f chosen from the set of all n-bit to n-bit functions
- To encrypt, Alice can choose a random index $i \in \{0,1\}^n$ and send $(i, f(i) \oplus m)$
  - This is like one-time pad, but there's only a negligible probability we reuse the same index
  - Bob has the same truth table/key and can decrypt
- Problem: Key is exponentially long!



n-bits

2^n

$011001\ldots$

$f(3)$

$f(2)$

$\vdots$

$f(i)$

$\vdots$

$f(2^n)$

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)

Relevant Notation for PRFs

- A keyed function F has two inputs, where we refer to the first as the key $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$
  - For a key $k \in \{0,1\}^n$, denote $F_k : \{0,1\}^{\ell_{\text{in}}(n)} \to \{0,1\}^{\ell_{\text{out}}(n)}$ to be the function F when we fix the key to k
- F is length-preserving if for all n, $\ell_{\text{in}}(n) = \ell_{\text{out}}(n) = n$
  - That is, for all n and all $k \in \{0,1\}^n$, we have $F_k : \{0,1\}^n \to \{0,1\}^n$
- We say a function F is efficient if there exists a polynomial-time (deterministic) algorithm computing $F(k, x)$

More Relevant Notation

- Sometimes we will use algorithms that can make calls to an oracle (function), which will denote in the superscript
  - $D^{\mathcal{O}}$ denotes an algorithm D with calls to an oracle $\mathcal{O}$
- Let $\mathcal{F}_n$ be the set of all possible functions mapping n-bits to n-bits: $\mathcal{F}_n = \left\{f \mid f : \{0,1\}^n \to \{0,1\}^n\right\}$
  - Sampling a truly random function from this set we will use $h \leftarrow \mathcal{F}_n$

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

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

Pseudorandom Functions (PRFs)

Definition:

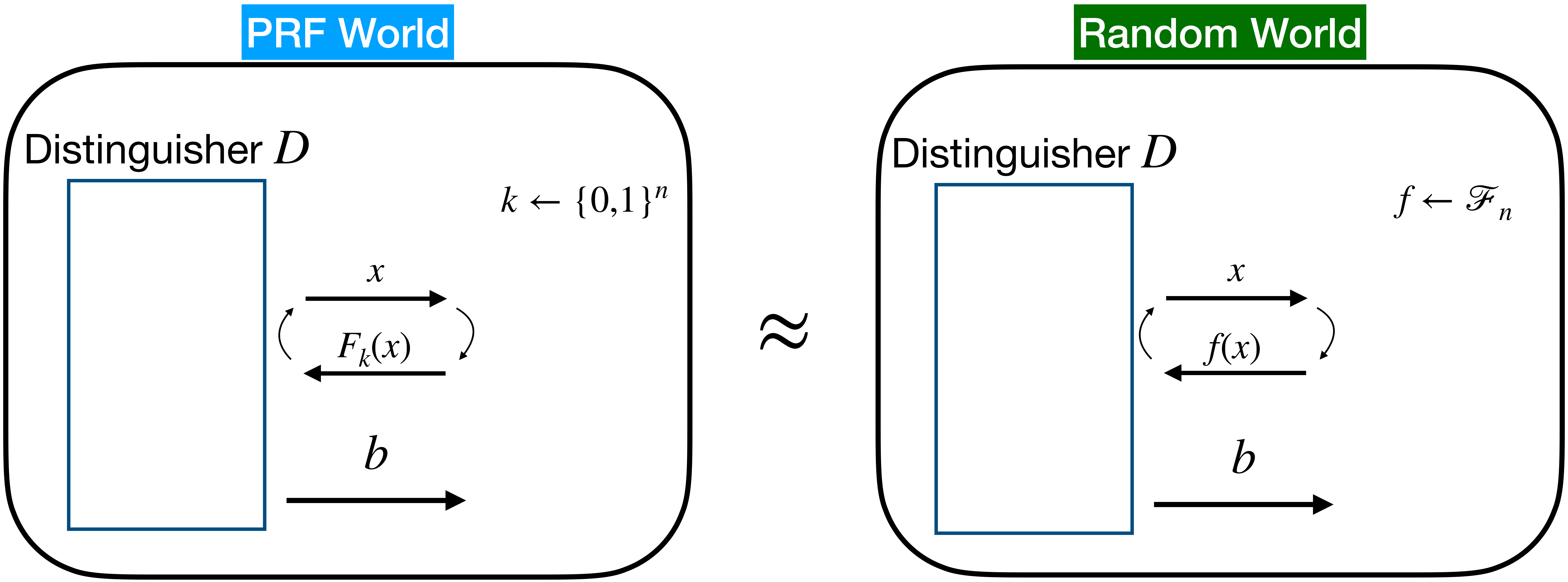
Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed function. We say that F is a pseudorandom function (PRF) if for all PPT D there exists a negligible function $\epsilon(\cdot)$ such that

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

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



PRF World

Distinguisher D

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

x

$F_k(x)$

b

$\approx$

Random World

Distinguisher D

$f \leftarrow \mathscr{F}_n$

x

$f(x)$

b

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

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:

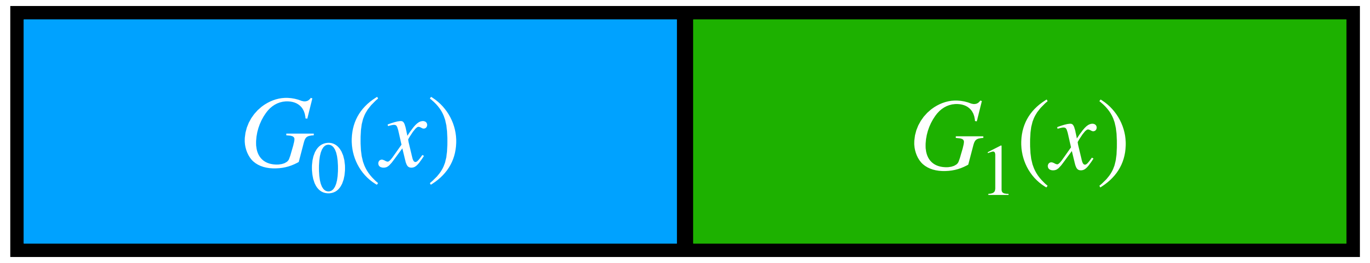
- D gets a single string ("one shot")
  - D does not get to choose the PRG seed or see multiple evaluations
- Security of the PRG says D behaves almost the same when:
  - s is chosen at random (not known to D) and $G(s)$ is given to D
  - D is given a truly random string

PRGs and PRFs

- Theorem: If PRFs exist, then PRGs exist
    - Proof idea: if F is a PRF, then you can construct a PRG as $G(s) = F_s(1) || F_s(2) \ldots F_s(\ell)$ for any ℓ that makes G expanding
- Theorem [GGM]: If PRGs exist, then PRFs exist
    - Construction will be presented without proof
- Put together: PRG$\Longleftrightarrow$PRF

GGM Construction: PRF from PRGs

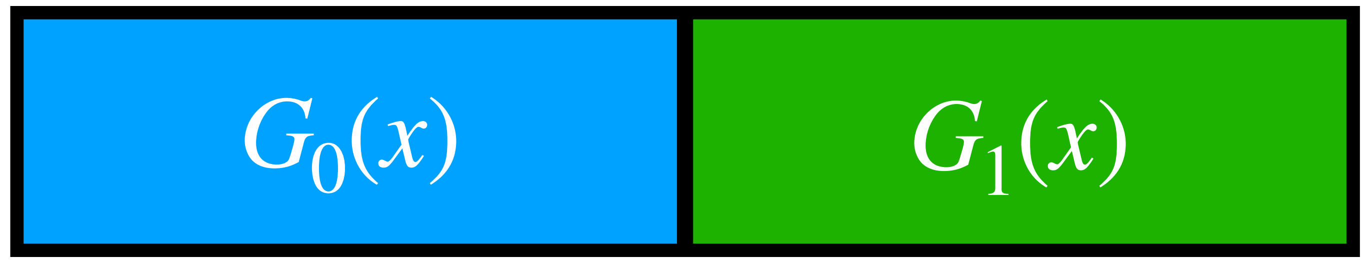
- Suppose G is a length doubling PRG $G : \{0,1\}^n \rightarrow \{0,1\}^{2n}$
- Let $G_0(x)$ denote the first n bits of $G(x)$ and $G_1(x)$ the last n bits of $G(x)$
- $G(x) =$



$G_0(x)$ $G_1(x)$

GGM Construction: PRF from PRGs

- Suppose G is a length doubling PRG $G : \{0,1\}^n \rightarrow \{0,1\}^{2n}$
- Let $G_0(x)$ denote the first n bits of $G(x)$ and $G_1(x)$ the last n bits of $G(x)$
- $G(x) =$



$G_0(x)$ $G_1(x)$

- Define F as follows: For $k \in \{0,1\}^n$ and $x = x_1 \ldots x_n \in \{0,1\}^n$

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

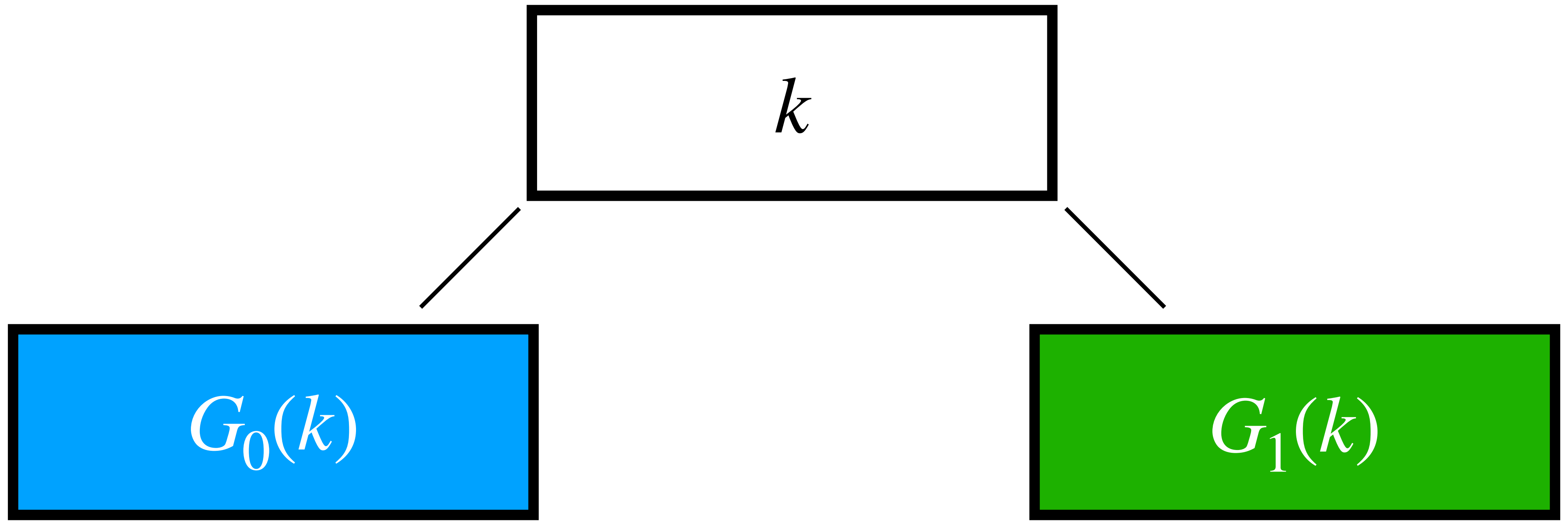
GGM Construction: PRF from PRGs

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

k

GGM Construction: PRF from PRGs

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



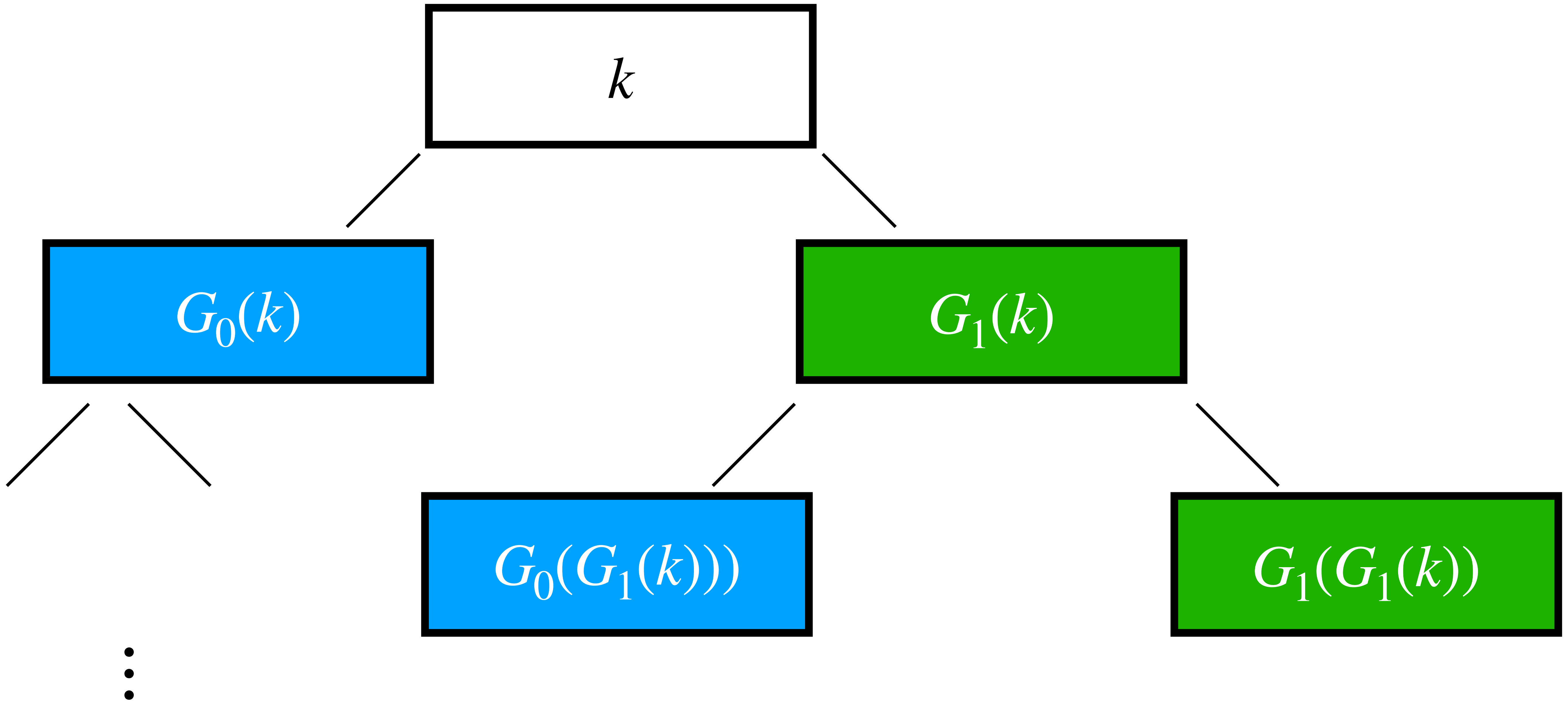
k

$G_0(k)$

$G_1(k)$

GGM Construction: PRF from PRGs

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



k

$G_0(k)$

$G_1(k)$

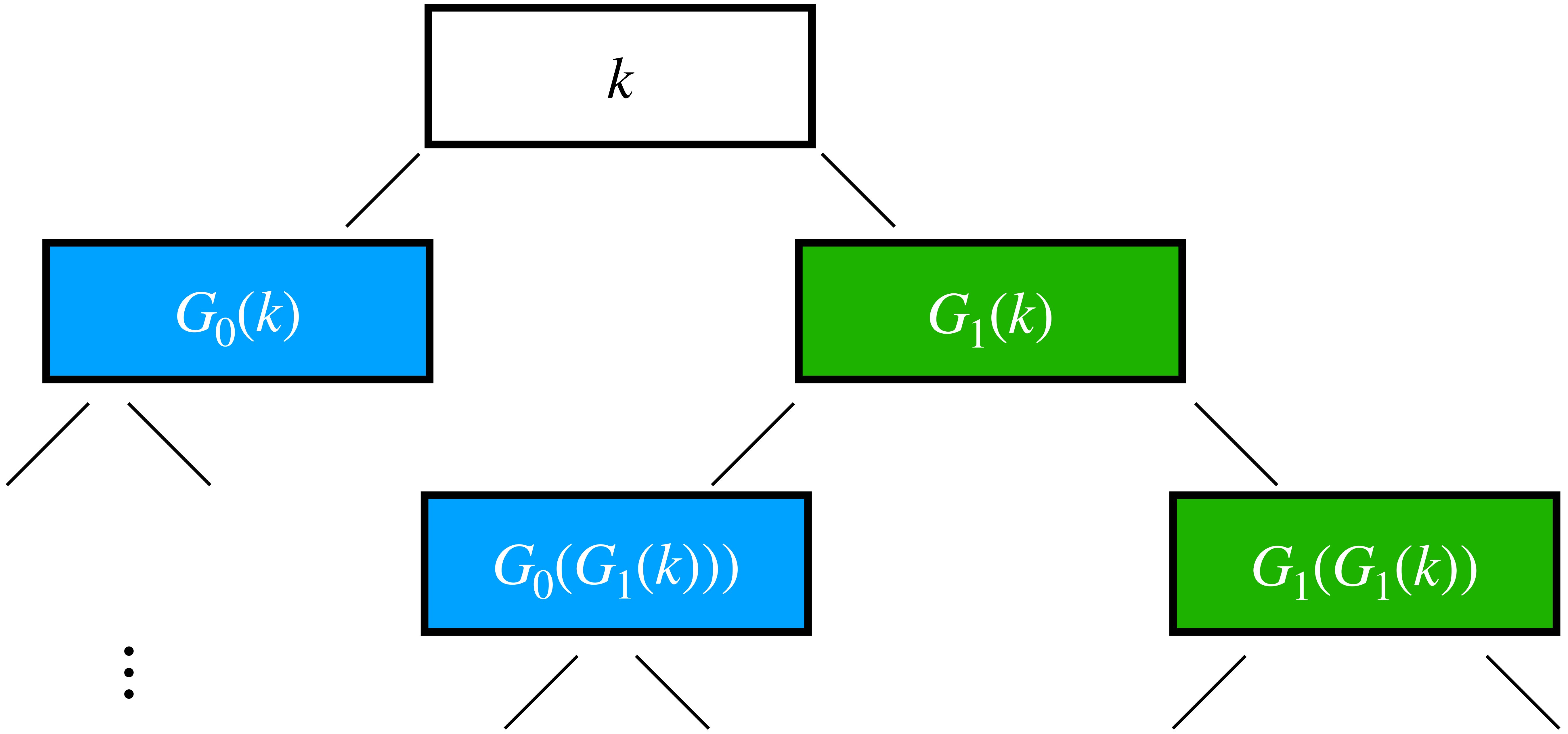
$G_0(G_1(k)))$

$G_1(G_1(k))$

$\vdots$

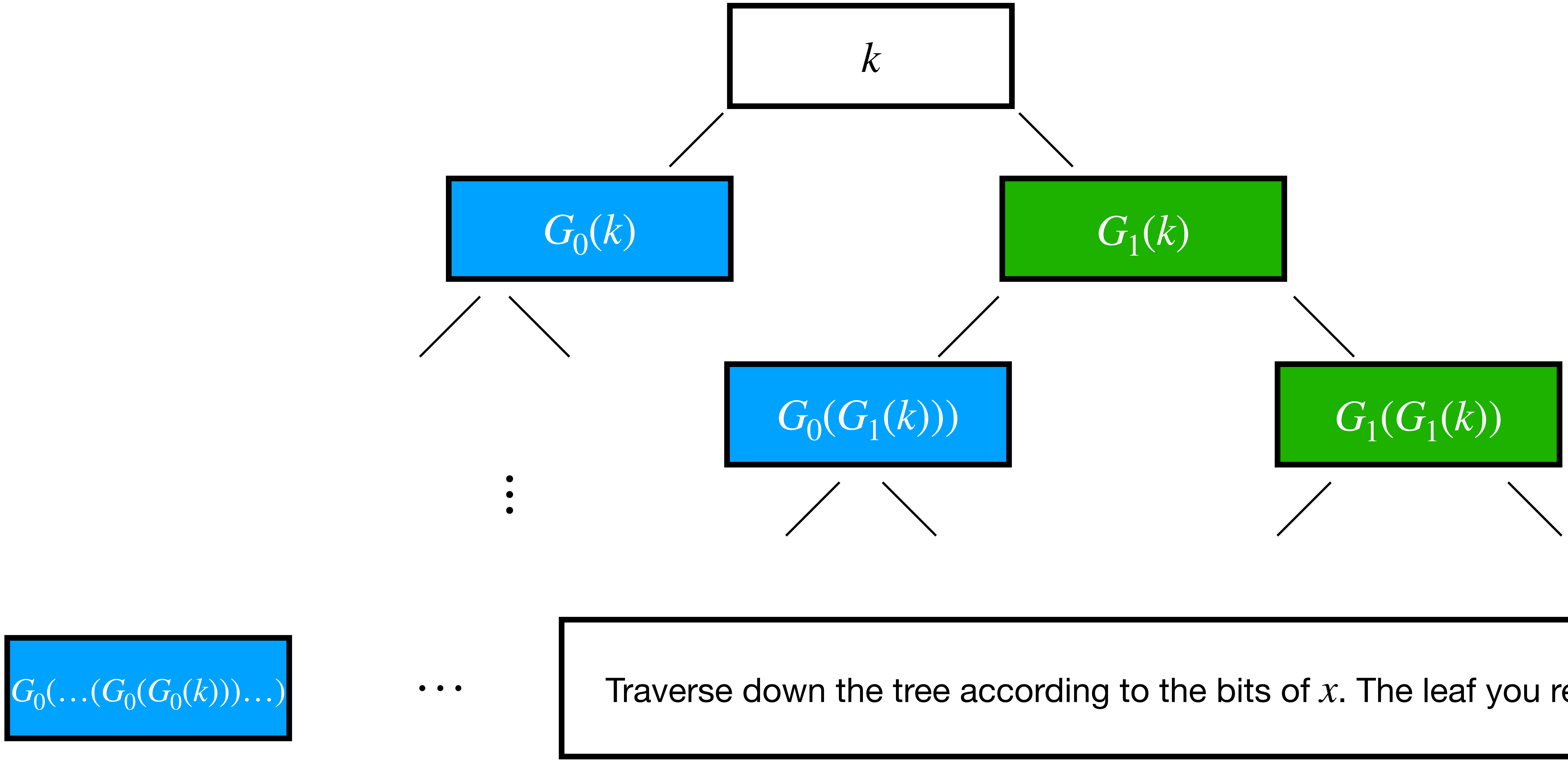
GGM Construction: PRF from PRGs

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



GGM Construction: PRF from PRGs

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



k

$G_0(k)$

$G_1(k)$

$G_0(G_1(k)))$

$G_1(G_1(k))$

$\vdots$

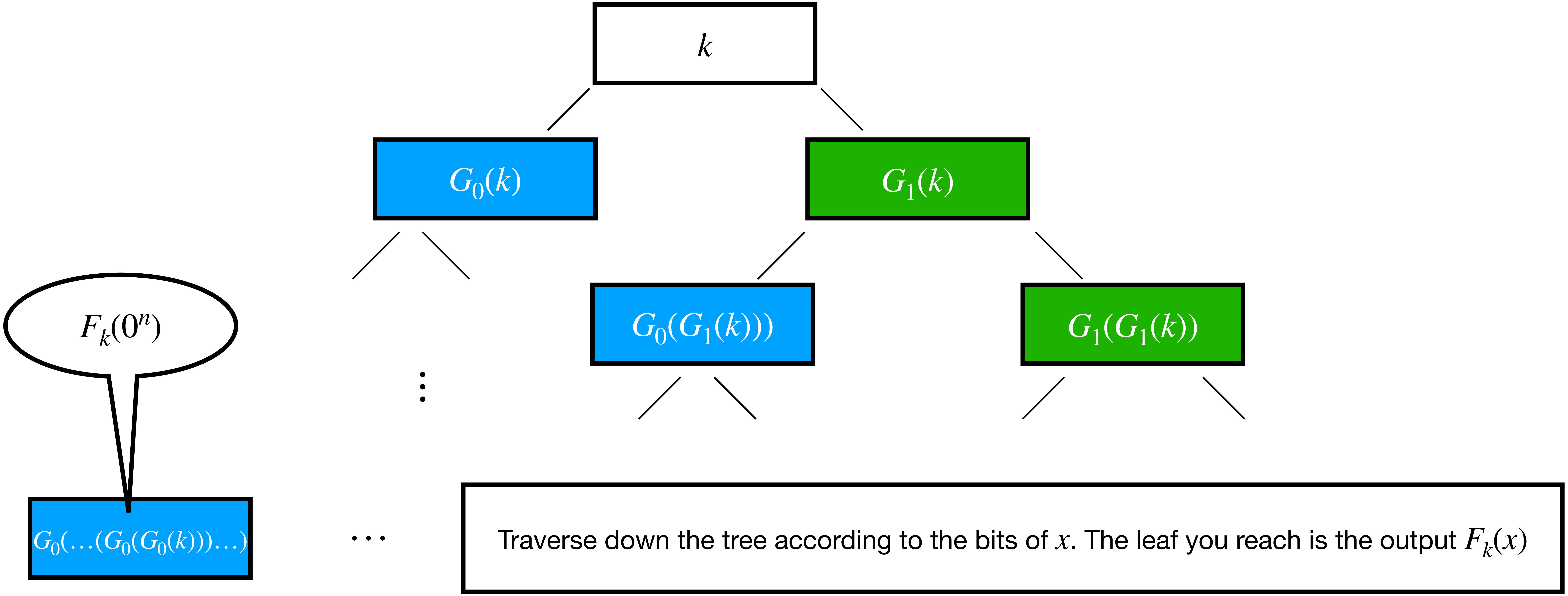
$G_0(\ldots(G_0(G_0(k)))\ldots)$

$\cdots$

Traverse down the tree according to the bits of x. The leaf you reach is the output $F_k(x)$

GGM Construction: PRF from PRGs

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



Traverse down the tree according to the bits of x. The leaf you reach is the output $F_k(x)$

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:

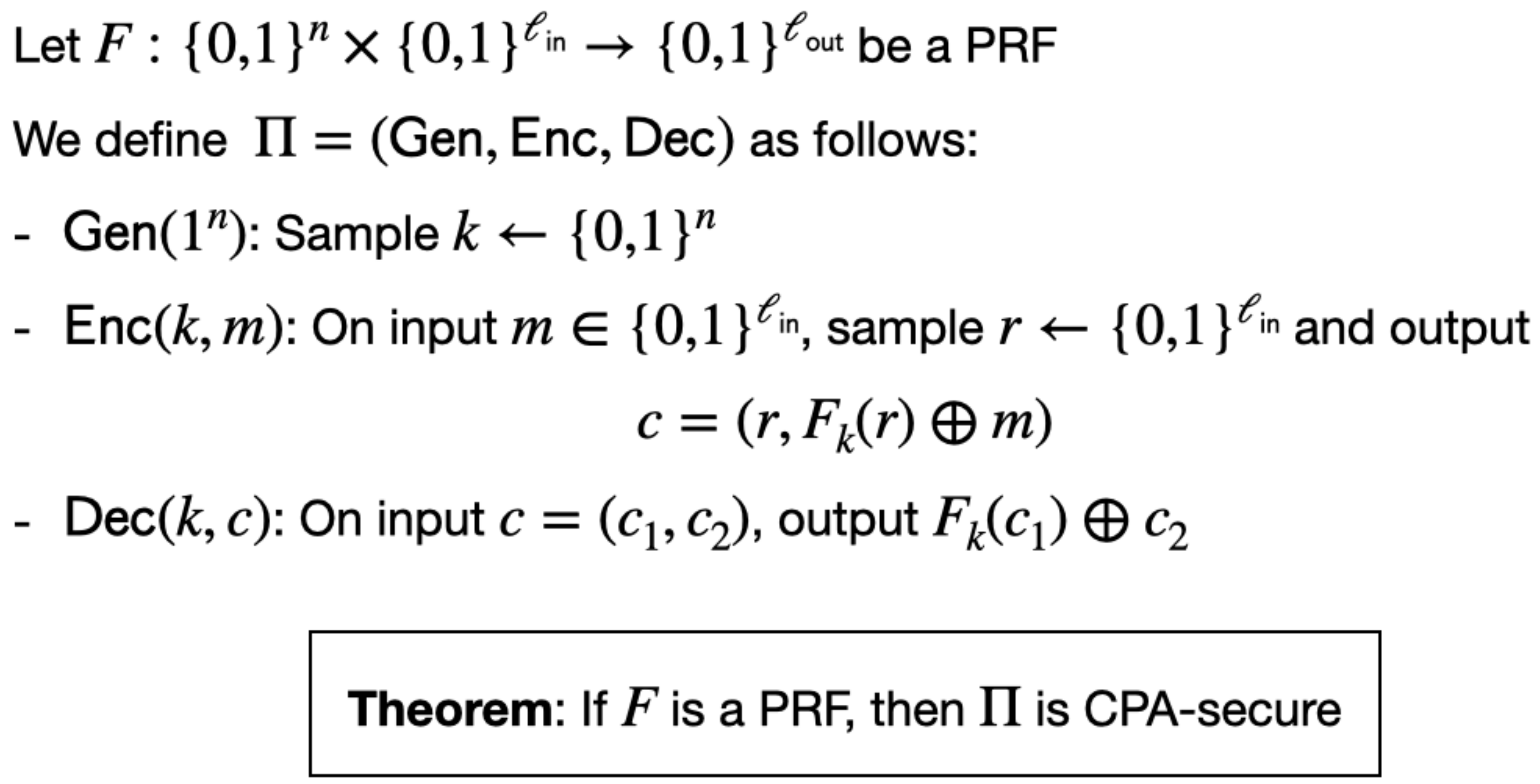
- Gen(1^n): Sample $k \leftarrow \{0,1\}^n$
- Enc(k, m): On input $m \in \{0,1\}^{\ell_{\text{in}}}$, sample $r \leftarrow \{0,1\}^{\ell_{\text{in}}}$ and output

$$c = (r, F_k(r) \oplus m)$$

- 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 $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ as follows:

- $\mathsf{Gen}(1^n)$: Sample $k \leftarrow \{0,1\}^n$
- $\mathsf{Enc}(k, m)$: On input $m \in \{0,1\}^{\ell_{\text{in}}}$, sample $r \leftarrow \{0,1\}^{\ell_{\text{in}}}$ and output

$$c = (r, F_k(r) \oplus m)$$

- $\mathsf{Dec}(k, c)$: 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

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**: Consider a version $\hat{\Pi}$ where we use a truly random function instead of a PRF. The scheme $\hat{\Pi}$ is CPA-secure.

(intuitively: no PPT A can with the CPA game with probability better than $1/2 + \mathsf{negl}(n)$)

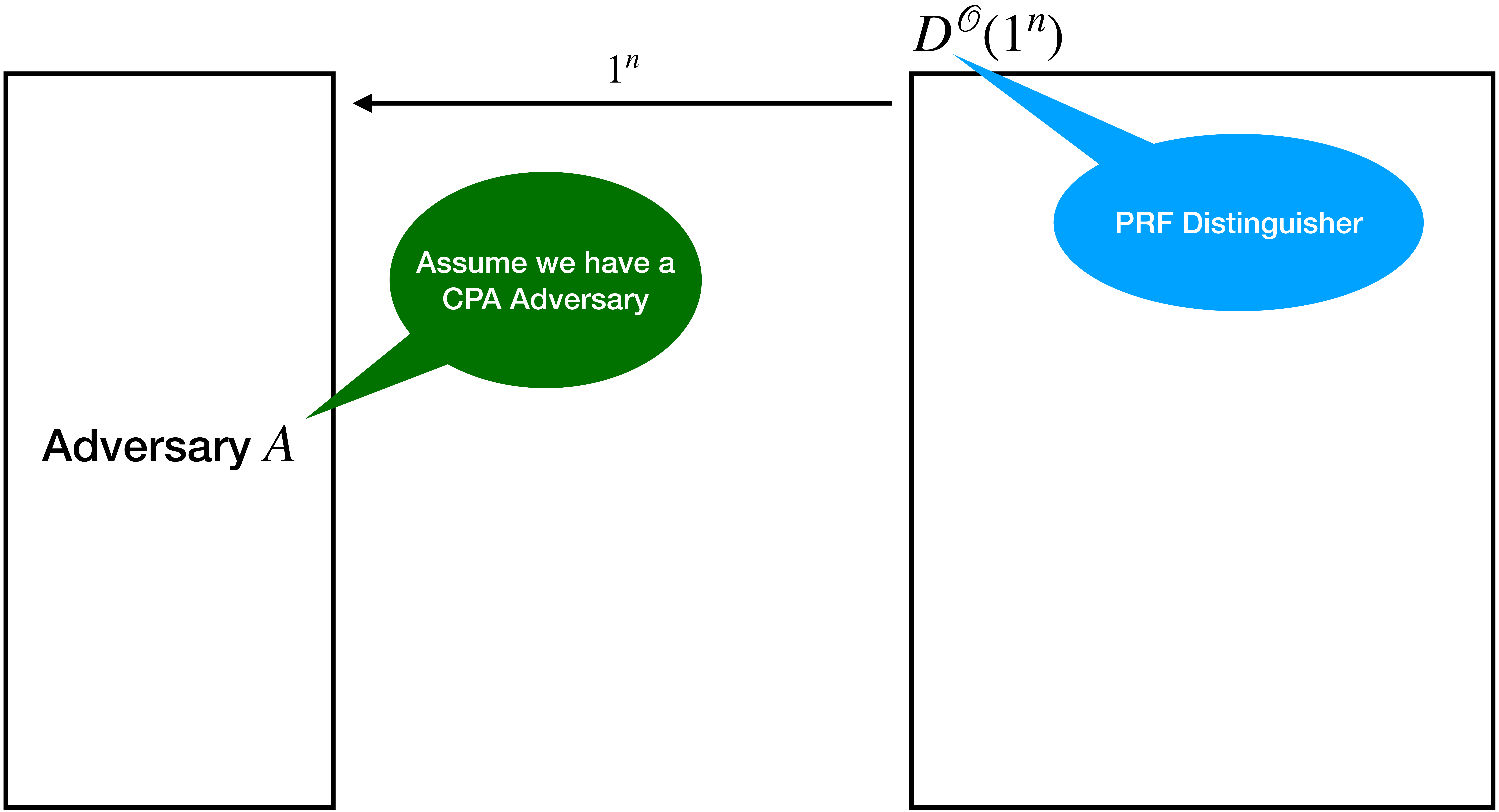
Proving Lemma 1

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

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

Proof sketch: 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



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

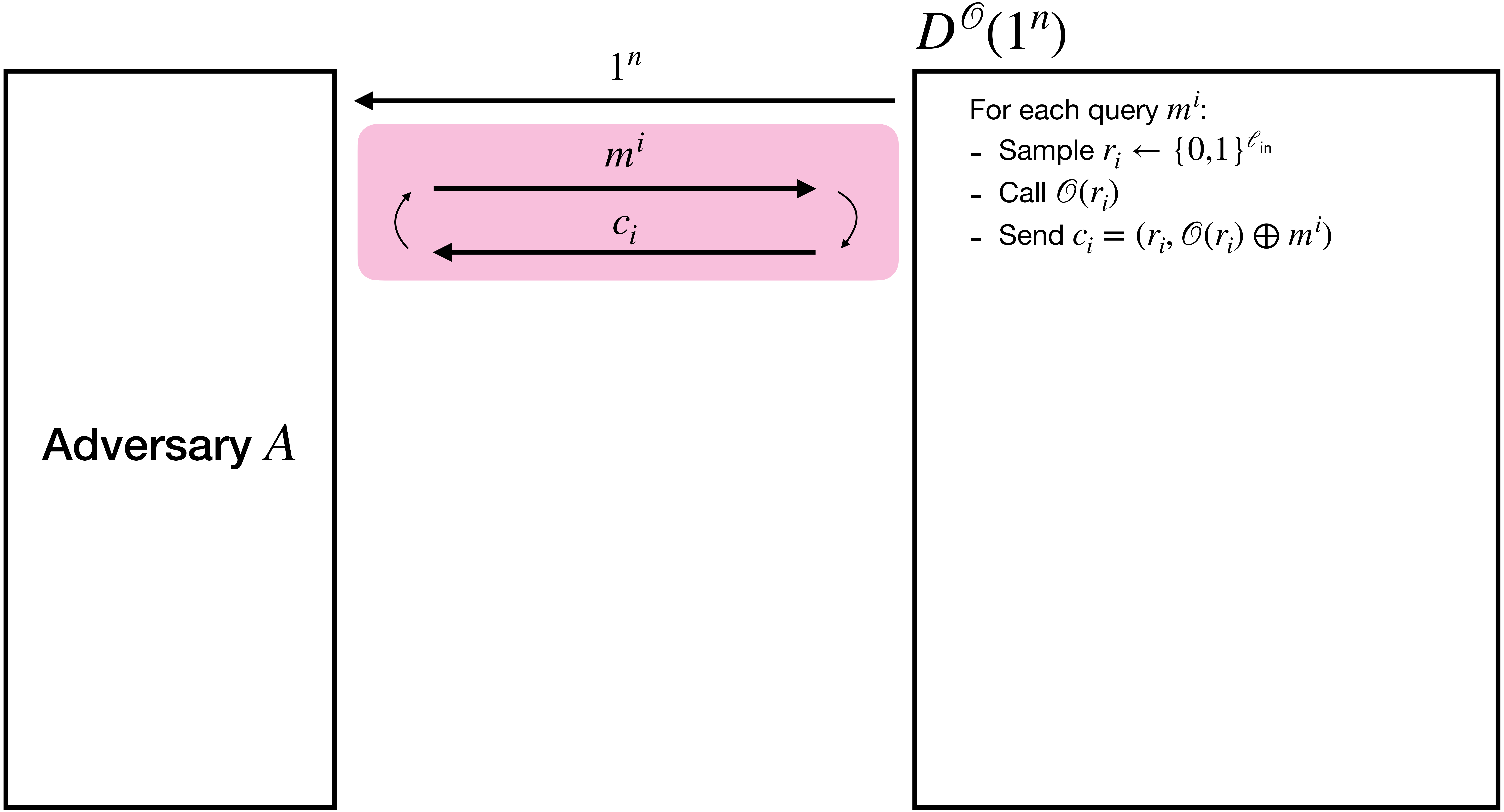
1^n

Assume we have a CPA Adversary

PRF Distinguisher

Adversary A

Reduction for Lemma 1



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

1^n

m^i

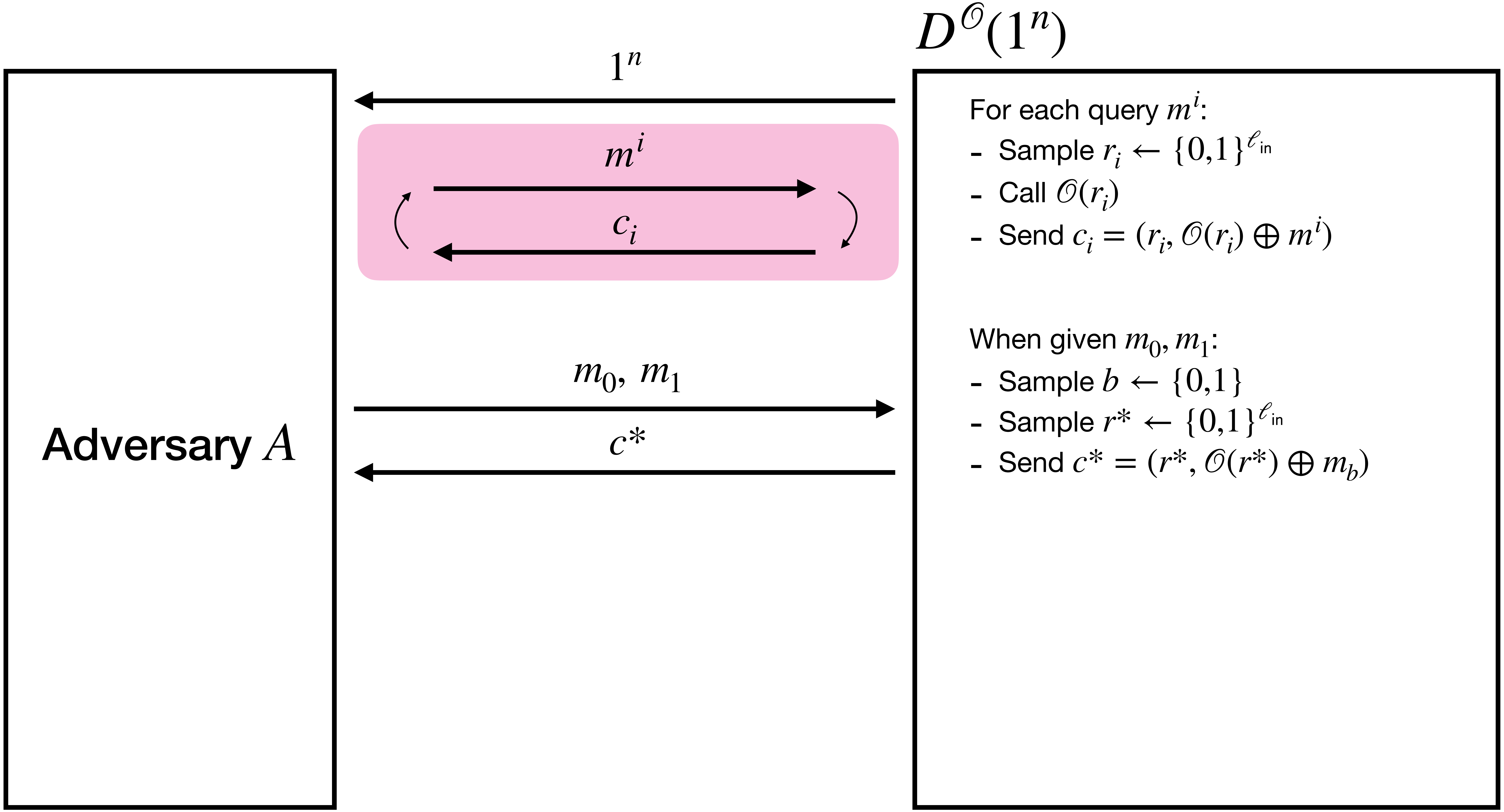
c_i

Adversary A

For each query m^i:

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

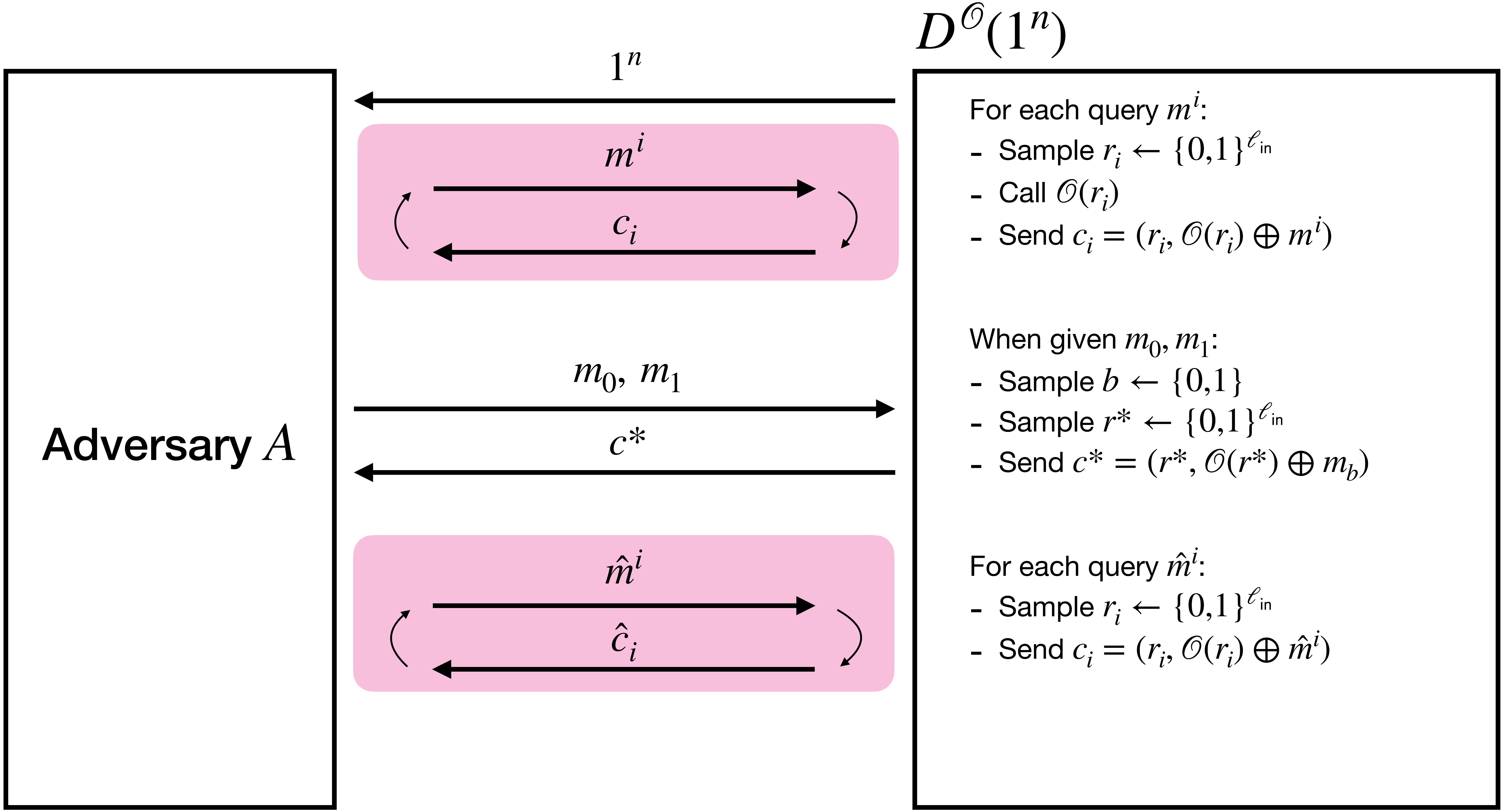
Reduction for Lemma 1



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

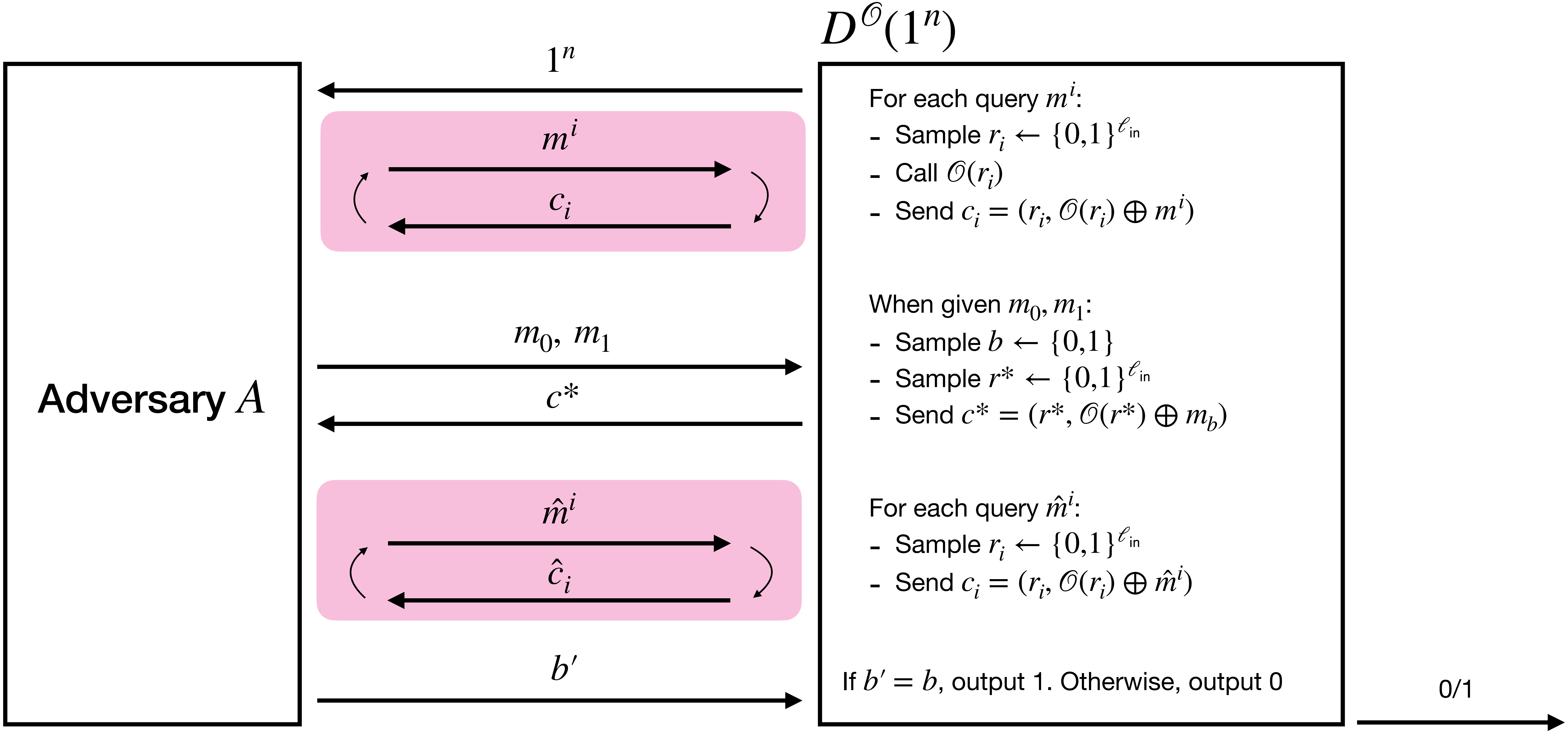


For each query m^i:
- Sample $r_i \leftarrow \{0,1\}^{\ell_{\text{in}}}$
- Call $\mathcal{O}(r_i)$
- Send $c_i = (r_i, \mathcal{O}(r_i) \oplus m^i)$

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



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

Proving Lemma 1

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

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

Proof sketch: 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)

(Informal) D's advantage from the previous slide is the same as A's.

(You can work this out at home)

Proving Lemma 2

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

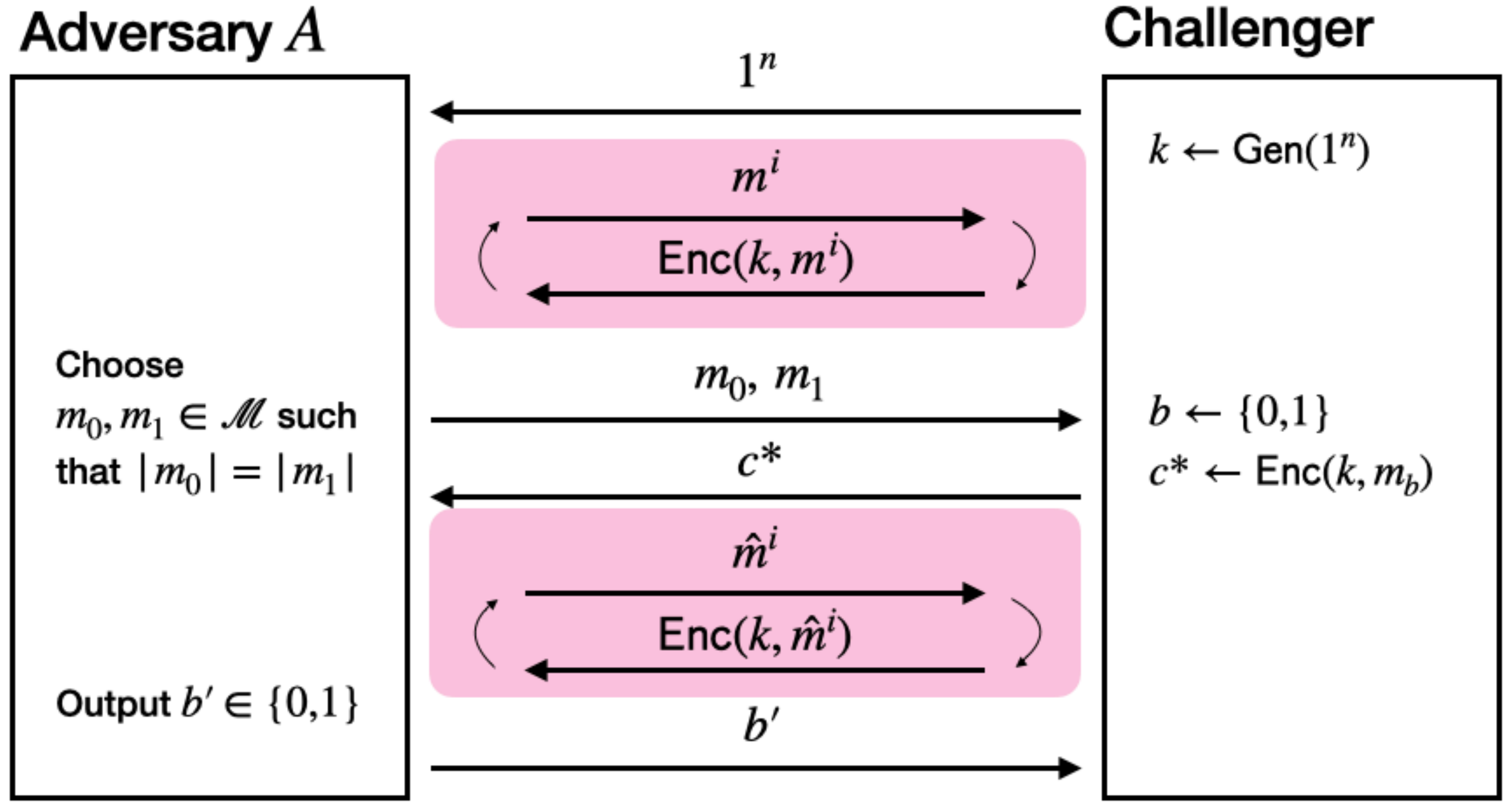
$$\Pr[\mathsf{PrivK}^{\mathsf{CPA}}_{\hat{\Pi},A}(n) = 1] \leq \frac{1}{2} + \epsilon_2(n)$$

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.



Adversary A

Choose $m_0, m_1 \in \mathscr{M}$ such that $|m_0| = |m_1|$

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

Challenger

$k \leftarrow \mathsf{Gen}(1^n)$

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

$c^* \leftarrow \mathsf{Enc}(k, m_b)$

1^n

m^i

$\mathsf{Enc}(k, m^i)$

$m_0,\ m_1$

c^*

$\hat{m}^i$

$\mathsf{Enc}(k, \hat{m}^i)$

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. $r^* = r_i$)

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

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. $r^* = r_i$)

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

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

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

By the law of total probability:

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

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. $r^* = r_i$)

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

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. $r* = r_i$)

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

Proving Lemma 2

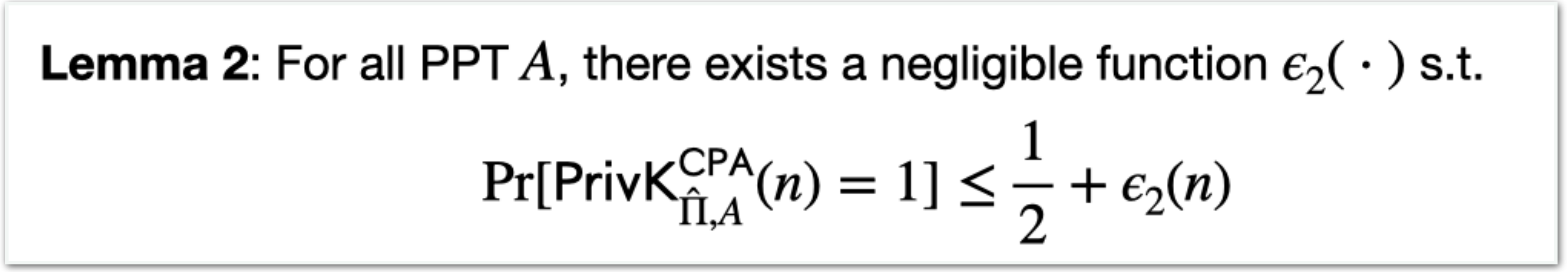
Proof continued:

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. $r* = r_i$)

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

This term is negligible



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

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

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,

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

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

Next Time

- PRPs, Block ciphers, and modes of operation