COMS BC3262: Introduction to Cryptography

Lecture 3: Pseudorandom Generators

January 28, 2026

Logistics Recap

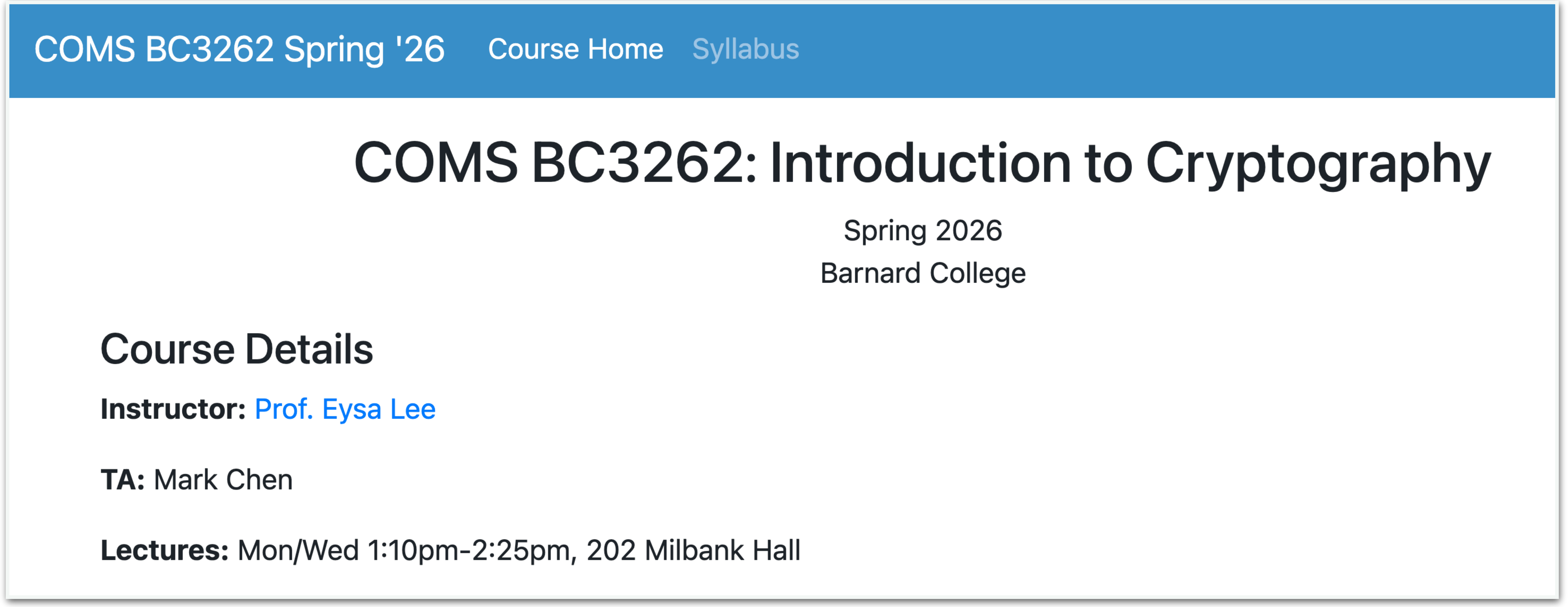
Please see course website for slides, homework, course expectations:

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

I've also reduced the number of total problem sets in the course to 5.

PS1 is due next week Thursday, PS2 is released next week Wednesday

Lowest PS grade is dropped



COMS BC3262: Introduction to Cryptography

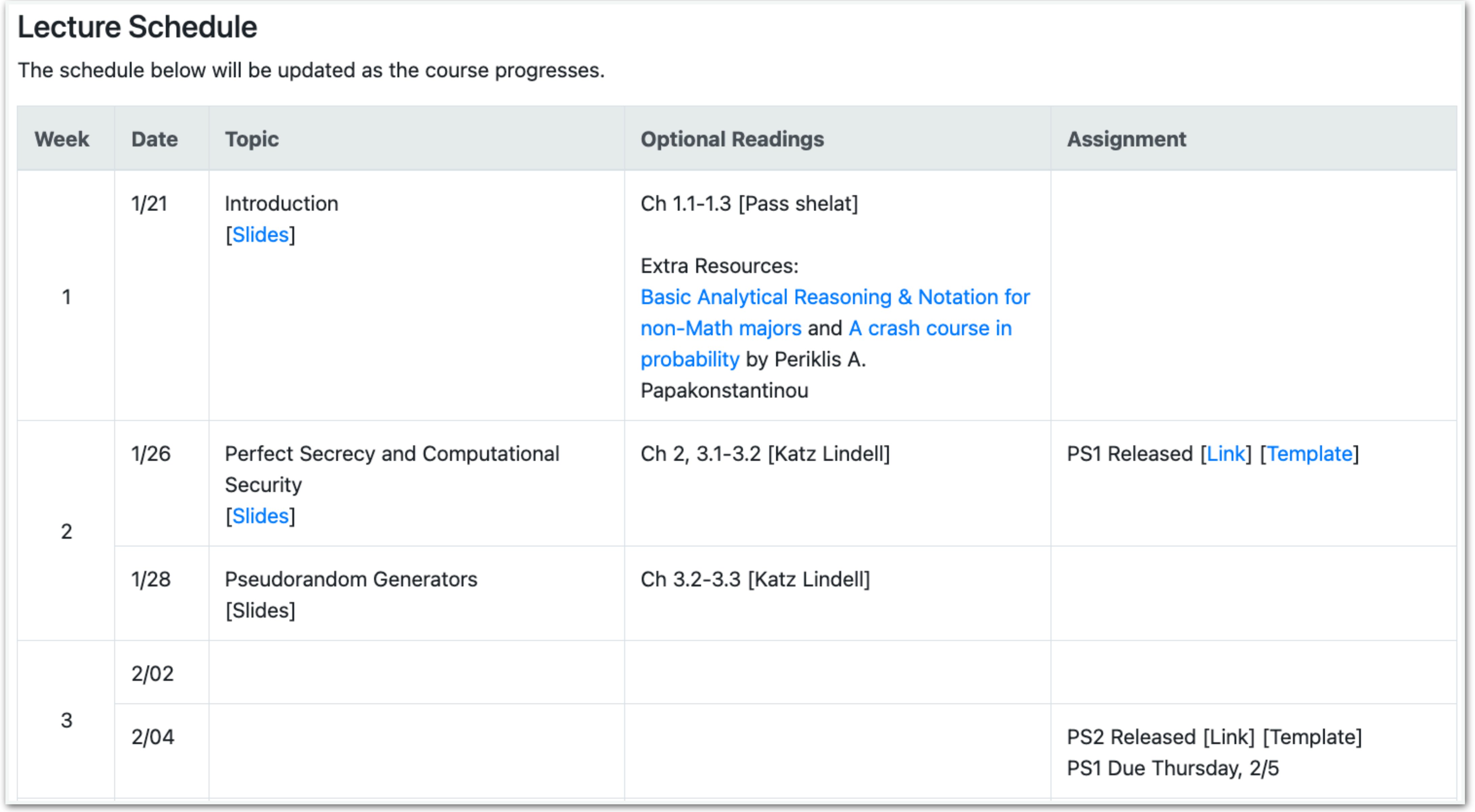
Spring 2026
Barnard College

Course Details

Instructor: Prof. Eysa Lee

TA: Mark Chen

Lectures: Mon/Wed 1:10pm-2:25pm, 202 Milbank Hall



Lecture Schedule

The schedule below will be updated as the course progresses.

Week	Date	Topic	Optional Readings	Assignment
1	1/21	Introduction [Slides]	Ch 1.1-1.3 [Pass shelat] Extra Resources: Basic Analytical Reasoning & Notation for non-Math majors and A crash course in probability by Periklis A. Papakonstantinou	
2	1/26	Perfect Secrecy and Computational Security [Slides]	Ch 2, 3.1-3.2 [Katz Lindell]	PS1 Released [Link] [Template]
	1/28	Pseudorandom Generators [Slides]	Ch 3.2-3.3 [Katz Lindell]	
3	2/02			
	2/04			PS2 Released [Link] [Template] PS1 Due Thursday, 2/5

Last Time: Perfect Secrecy

Perfect Secrecy

Definition: A symmetric-key encryption scheme is **perfectly secret** if for every distribution M over $\mathscr{M}$, for every $m \in \mathscr{M}$, and for every $c \in \mathscr{C}$ with $\Pr[C = c] > 0$ it holds that

$$\Pr[M = m \,|\, C = c] = \Pr[M = m]$$

Recall that:

- Eve may know an a priori distribution M
- K and M define a distribution $C = \mathsf{Enc}(K, M)$

then perfect secrecy means that the distributions M and C are **independent**

c does not give Eve any extra information about m!

Another Definition of Perfect Secrecy

Alternative Definition: A symmetric-key encryption scheme is **perfectly secret** if for every $m_0, m_1 \in \mathscr{M}$ and for every $c \in \mathscr{C}$ it holds that

$$\Pr[\mathsf{Enc}(K, m_0) = c] = \Pr[\mathsf{Enc}(K, m_1) = c]$$

This is the notation on the homework.

The probabilities are over the choice of K and the encryption scheme

One-Time Pad

Keys, messages, and ciphertexts are all the same length
$\mathscr{K} = \mathscr{M} = \mathscr{C} = \{0,1\}^\ell$

- Gen uniformly samples $k \leftarrow \{0,1\}^\ell$
- $\mathsf{Enc}(k, m) = m \oplus k$
- $\mathsf{Dec}(k, c) = c \oplus k$

Correctness: $\forall k \in \mathscr{K}, m \in \mathscr{M}$

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

Theorem: One-time pad is perfectly secret for any plaintext of any length ℓ

One-Time Pad Limitations

- Key can only be used once:

  Given $c = \mathsf{Enc}(k, m)$ and $c' = \mathsf{Enc}(k, m')$, you can learn $c \oplus c' = m \oplus m'$

- Keys are as long as the plaintext!
  - This unfortunately is not specific to one-time pad…

Theorem:

Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a symmetric-key encryption scheme with key space $\mathscr{K}$ and message space $\mathscr{M}$.

If Π is perfectly secret, then $|\mathscr{K}| \geq |\mathscr{M}|$

Limitations of Perfect Secrecy

Perfect secrecy is really great, but…

- The key must be at least as long as the message
- Definition of perfect secrecy only considers a single message

For extremely important communication I may be willing, but for everyday?

Big question: Can we guarantee “security” while avoiding these limitations?

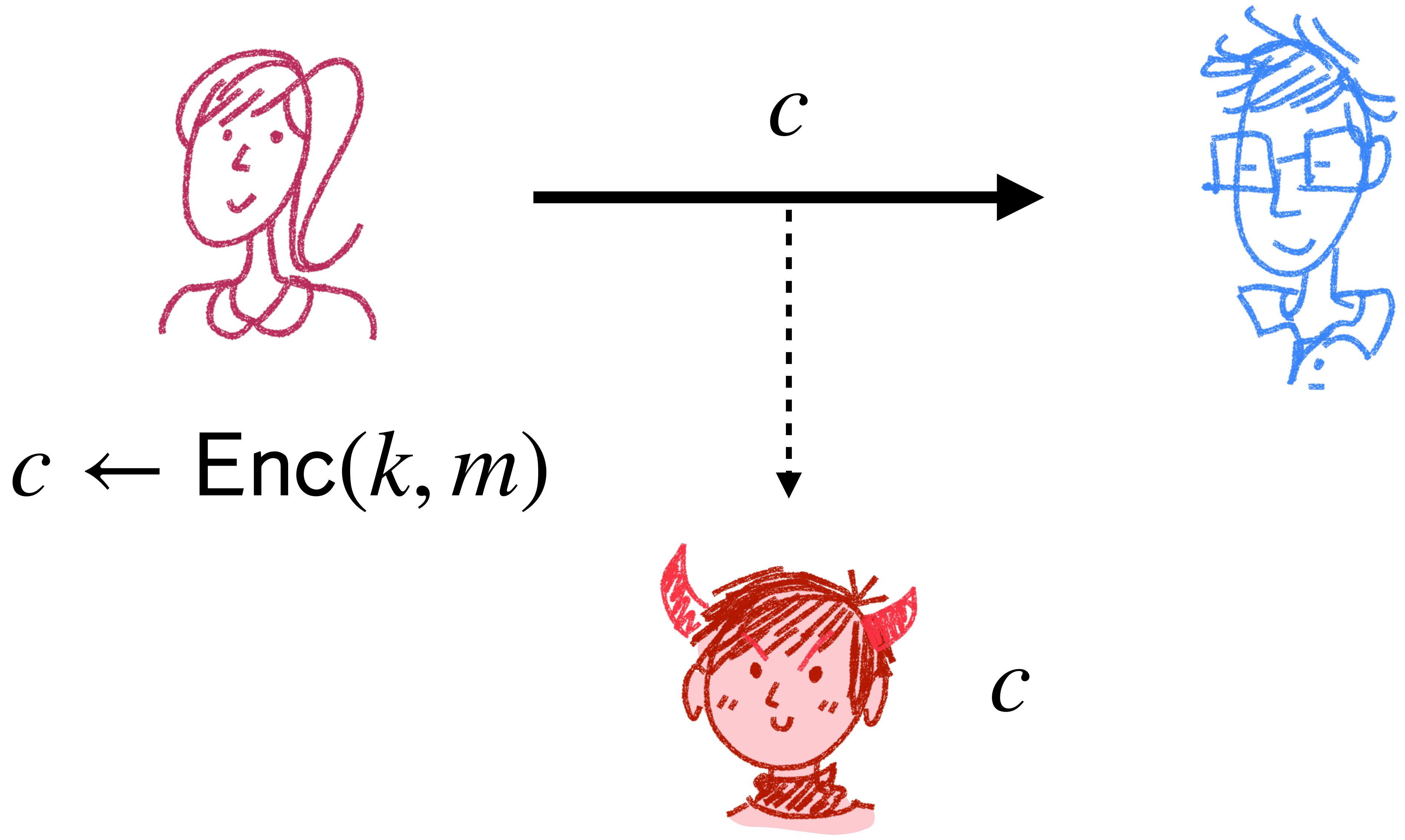
Last Time: Computational Security

Relaxing Our Setting

In order for us to get keys smaller than the message, we're going to relax our setting.

What are we trying to capture?

- Eve sees a ciphertext, but she shouldn't be able to tell if it's an encryption of this message or that message

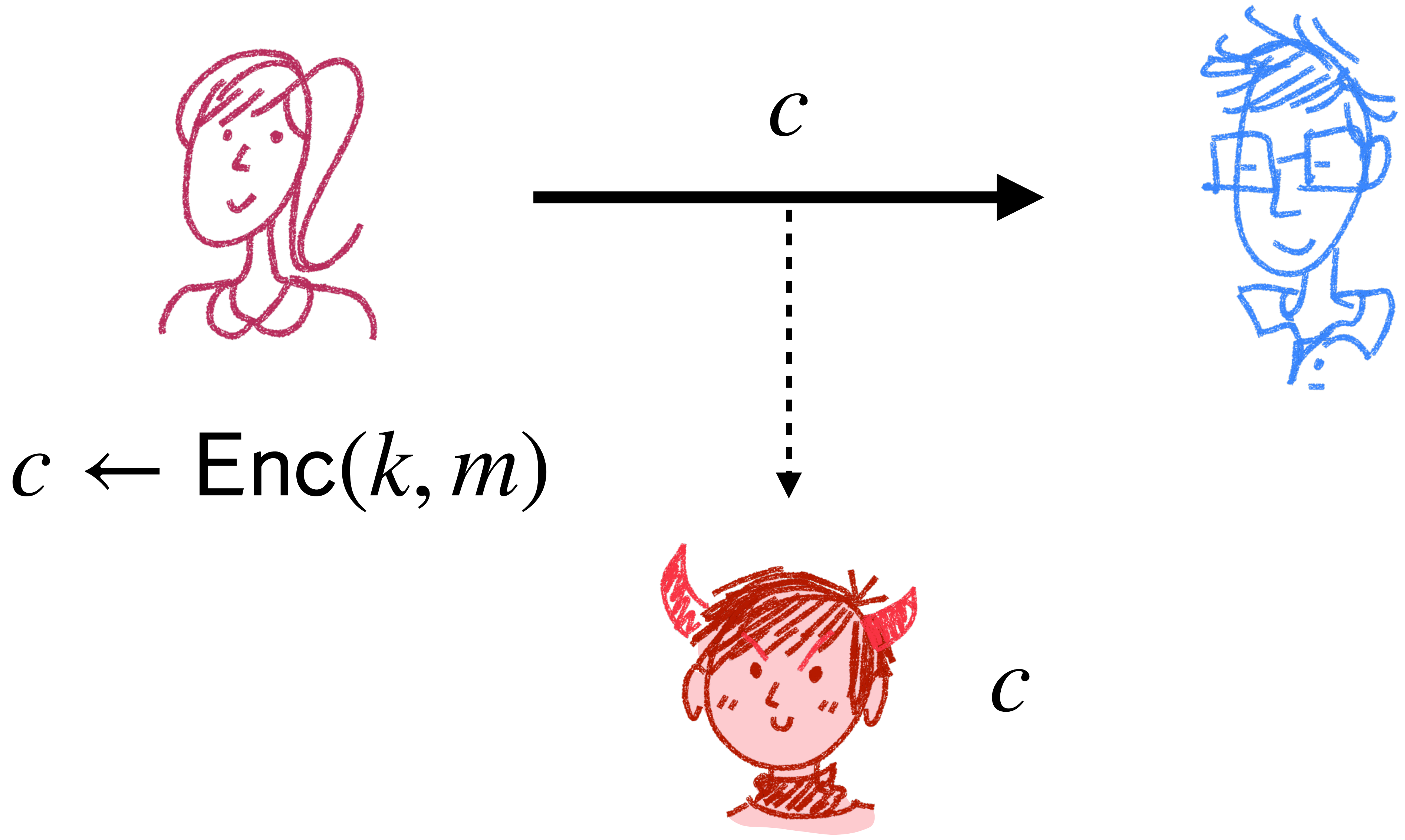


Relaxing Our Setting

In order for us to get keys smaller than the message, we're going to relax our setting.

What are we trying to capture?

- Eve sees a ciphertext, but she shouldn't be able to tell **in polynomial time** if it's an encryption of this message or that message

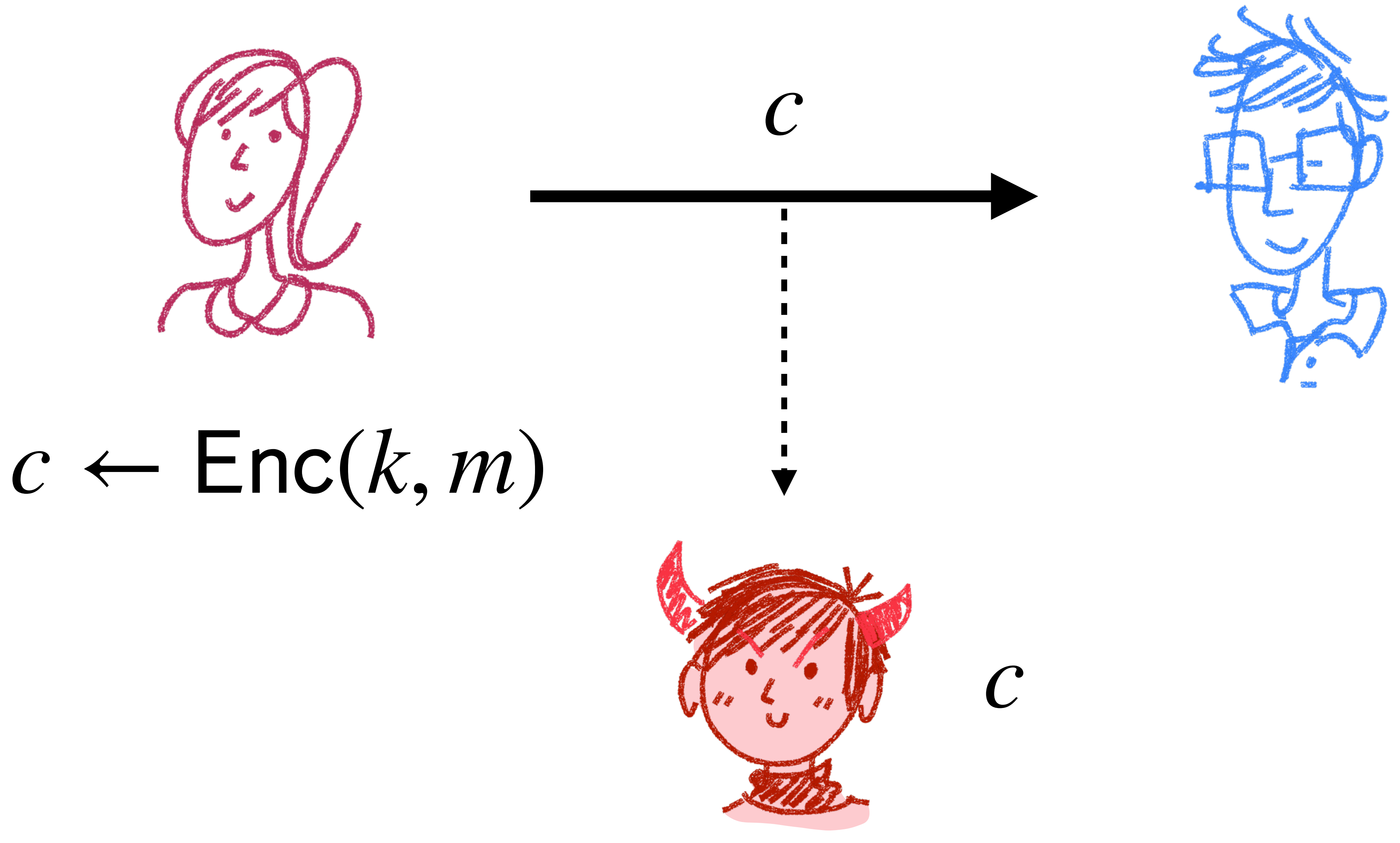


Relaxing Our Setting

In order for us to get keys smaller than the message, we're going to relax our setting.

What are we trying to capture?

- Eve sees a ciphertext, but she shouldn't be able to tell **in polynomial time** if it's an encryption of this message or that message **except with negligible probability**



c

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

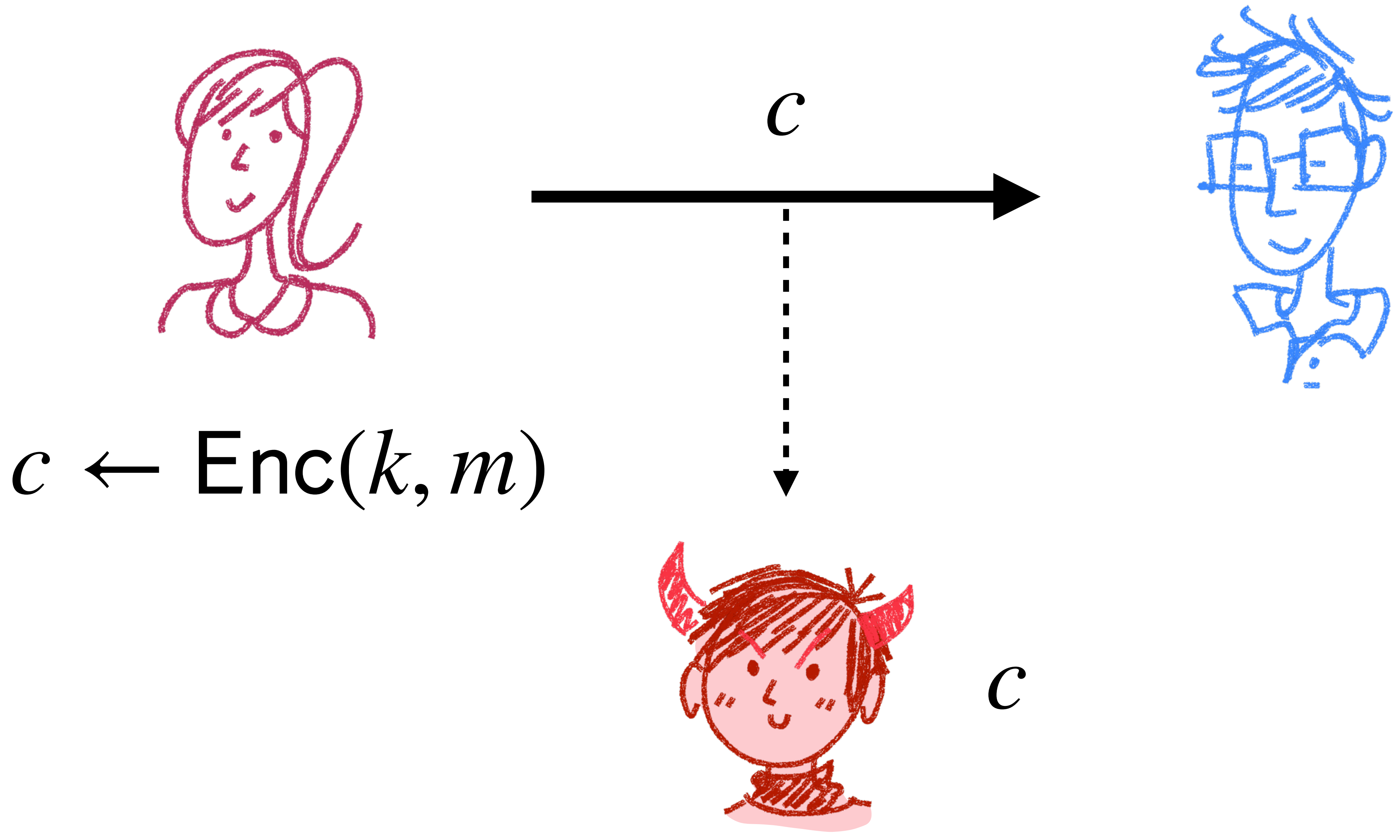
c

Relaxing Our Setting

In order for us to get keys smaller than the message, we're going to relax our setting.

What are we trying to capture?

- Eve sees a ciphertext, but she shouldn't be able to tell **in polynomial time** if it's an encryption of this message or that message **except with negligible probability**



This is the idea behind **Indistinguishable Encryptions**

Correction to last time: Definition of Negligible Functions

- A function is negligible if it approaches 0 faster than any inverse polynomial
- **Definition**: A function $f : \mathbb{N} \to \mathbb{R}_+$ is a negligible function if for every positive polynomial $p(\cdot)$ there exists N such that for all $n > N$ it holds that

$$f(n) < \frac{1}{p(n)}$$

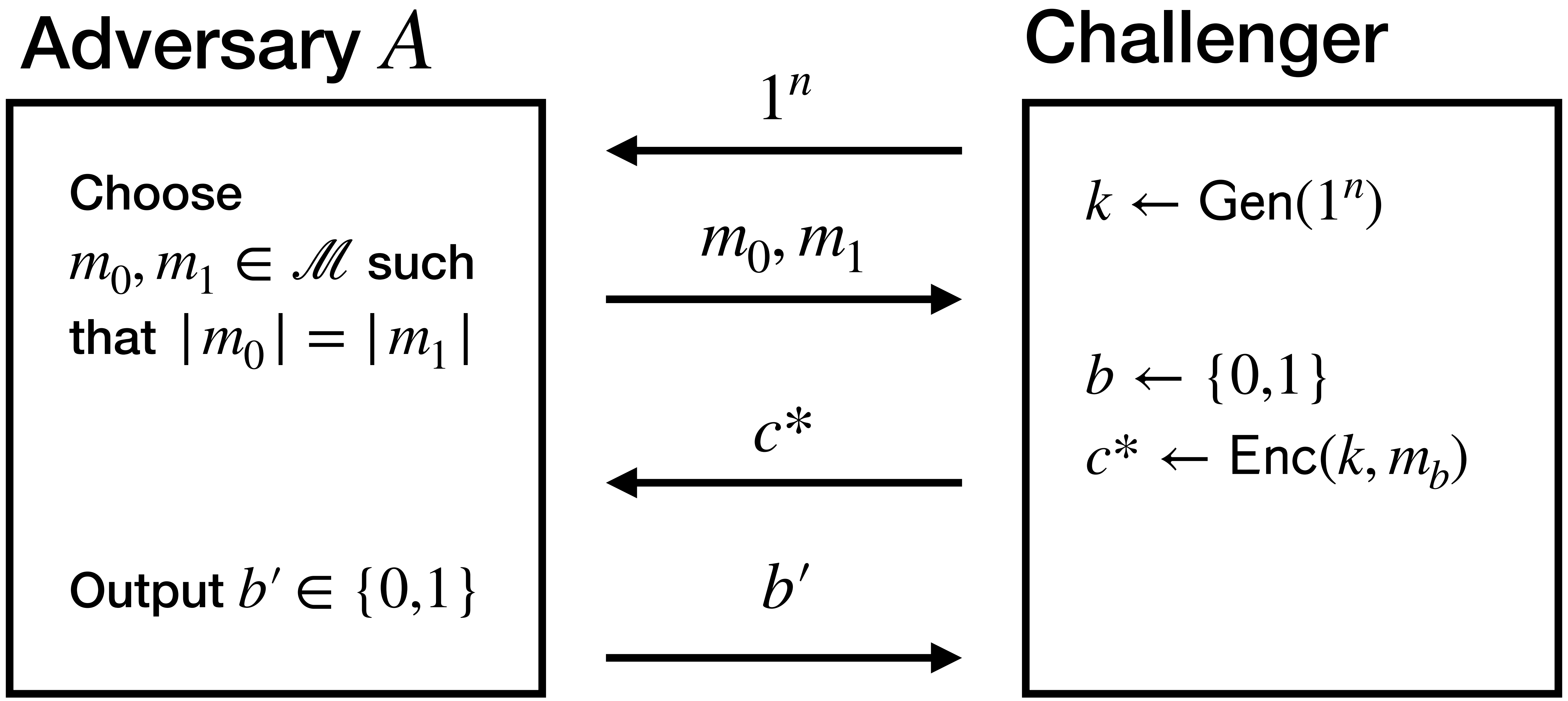
Last time had a typo. This should be *non-negative* reals

- Examples:
  - 2^{-n}, $2^{-\sqrt{n}}$, and $2^{-\log^2(n)}$ are negligible functions
  - $1/2$, $1/\log^2(n)$, and $1/n^5$ are non-negligible functions
- We denote by **negl**(n) an arbitrary negligible function

n is going to be our security parameter

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

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

Challenger

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

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

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

A wins if $b' = b$

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

$\mathsf{PrivK}^{\mathsf{eav}}_{\Pi,A}(n) = 0$ otherwise

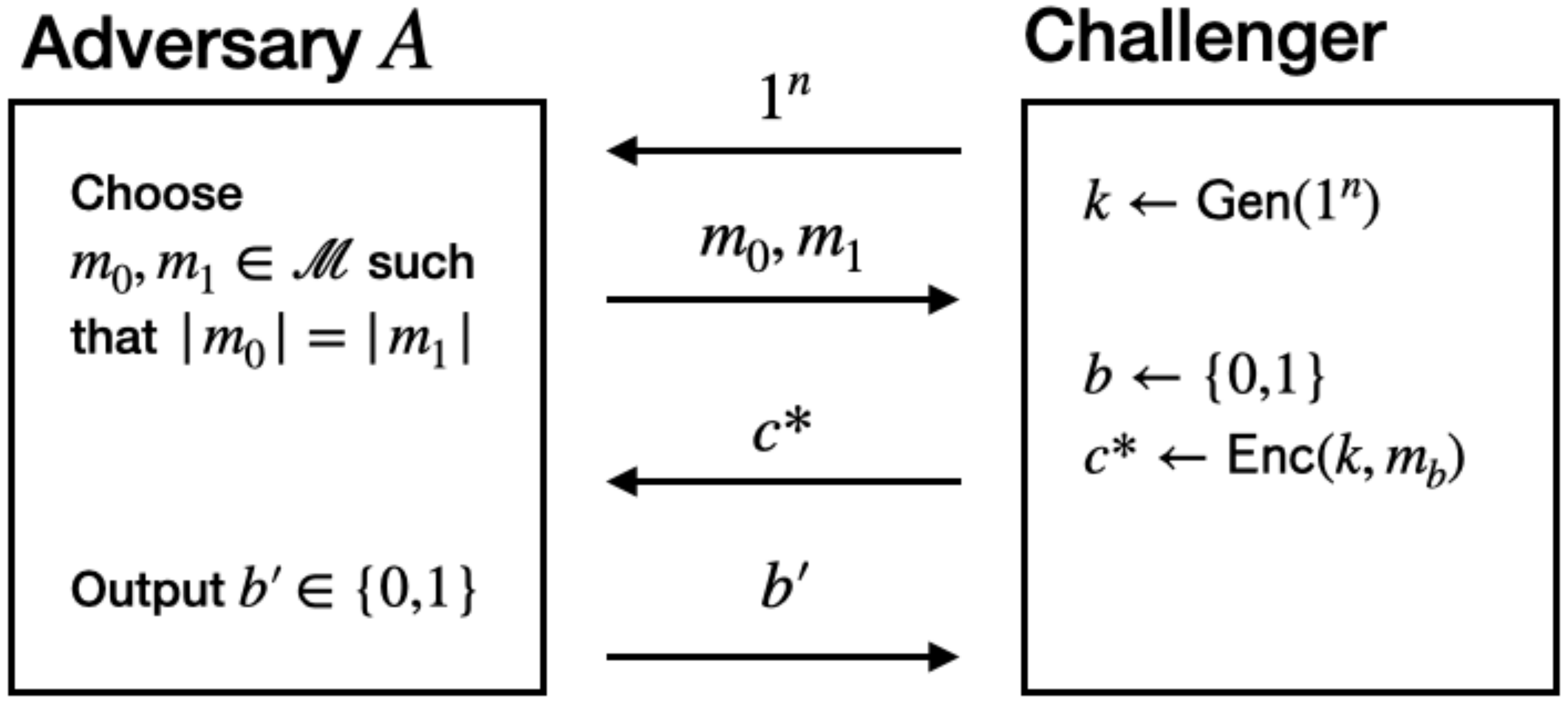
Indistinguishable Encryptions

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

$\mathsf{PrivK}^{\mathsf{eav}}_{\Pi,A}(n)$:

- A receives 1^n from the Challenger and replies with m_0, m_1 (s.t. $|m_0| = |m_1|$)
- Challenger runs $k \leftarrow \mathsf{Gen}(1^n)$, samples a bit $b \leftarrow \{0,1\}$, and computes $c* \leftarrow \mathsf{Enc}(k, m_b)$.
- Challenger sends $c*$ to A.
- A outputs $b' \in \{0,1\}$

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


Adversary A
Challenger
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'
$k \leftarrow \mathsf{Gen}(1^n)$
$b \leftarrow \{0,1\}$
$c* \leftarrow \mathsf{Enc}(k, m_b)$

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

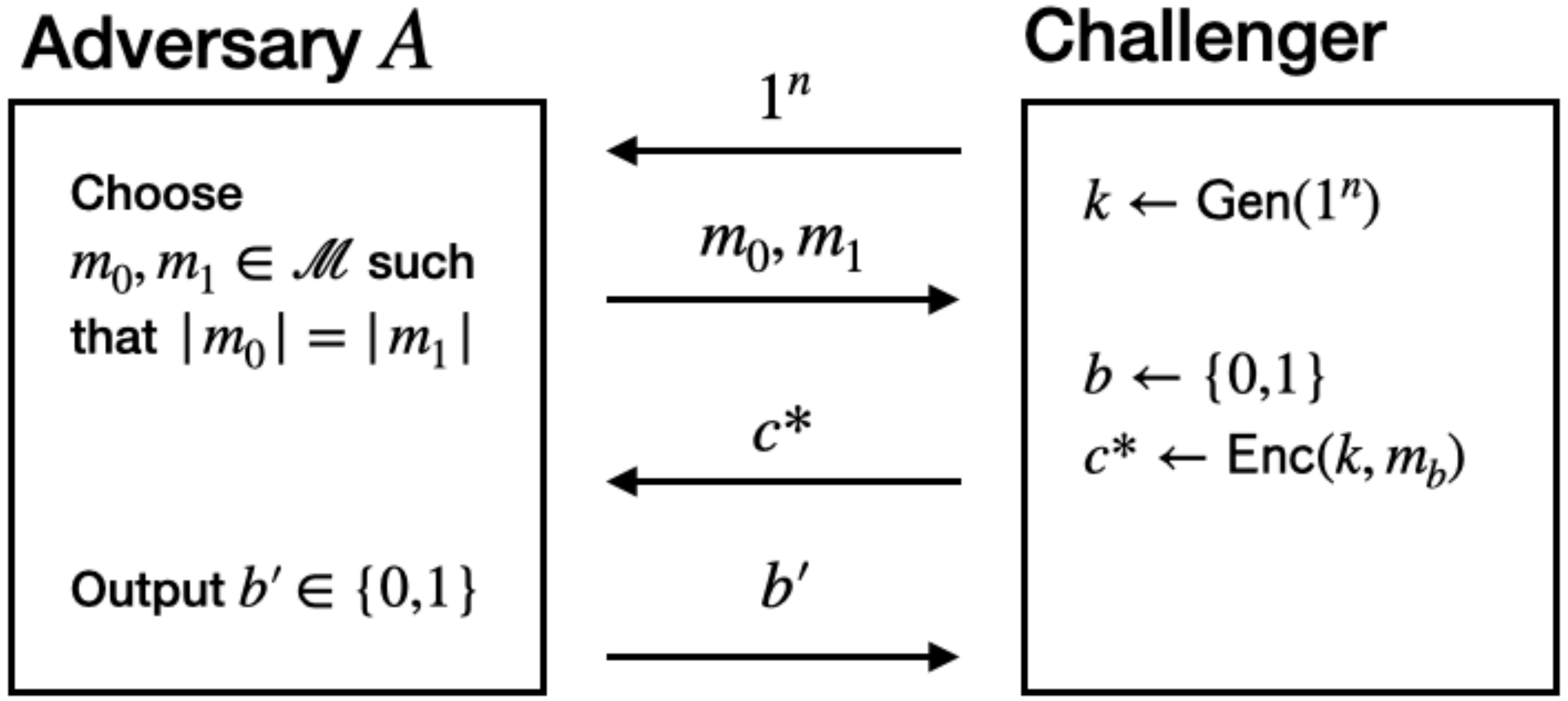
Indistinguishable Encryptions

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

Definition:

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

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



Adversary A

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

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

Challenger

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

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

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

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

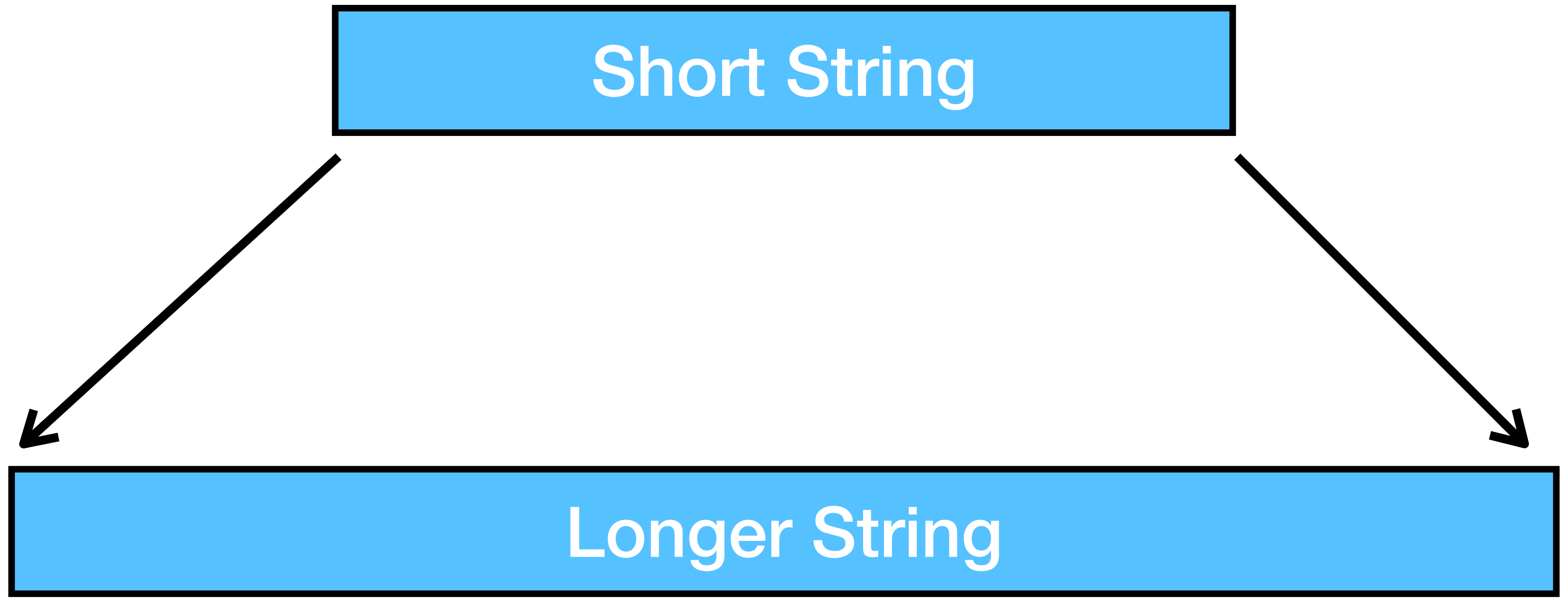
$\mathsf{PrivK}^{\mathsf{eav}}_{\Pi,A}(n) = 0$ otherwise

Constructing an EAV-Secure Scheme?

Can we come up with an EAV-Secure scheme that has keys that are smaller than the message size?

Idea: Deterministic algorithm expanding a shorter string into a longer one

- We could take our smaller key, expand it, and do one-time pad
- Our expanded key needs to "looks random" to any efficient algorithm



"indistinguishable"

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.

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

Pseudorandom Generators (PRGs)

et G … ynomial-time algorithm and $\ell(\cdot)$ be
s. … n we have $G(s) \in \{0,1\}^{\ell(n)}$. Then
pseudorando … f the following two conditions hold:

- **Ex ansion**: $\ell(n) > n$
- **Pse dorandomness**: or e … ls … negl gible function neg $(\cdot)$ s.t.

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

s is drawn uniformly from $\{0,1\}^n$

D gets the pseudorandom $G(s)$ as input

r is drawn uniformly from $\{0,1\}^{\ell(n)}$

D gets a truly random value as input

Do PRGs Exist?

If so, how difficult are they to construct?

Let’s gain some intuition by looking at some candidate PRGs.

Recall the two properties of PRGs:

- **Expansion**: $|G(s)| > |s|$
- **Pseudorandomness**: For every PPT D there exists a negligible function $\mathsf{negl}(\cdot)$ s.t.

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

Candidate PRGs

Consider the following candidates that expand a seed $s = s_1 \ldots s_n \in \{0,1\}^n$

- $G_1(s) = 0s_1 \ldots s_n$ (appends 0 to the front of s)

- $G_2(s) = \bar{s}_1 \ldots \bar{s}_n$ (flips the bits of s)

- $G_3(s) = s_1 \ldots s_n s_1$ (appends the first bit to the end)

- $G_4(s) = s_1 \ldots s_n z$ where $z = s_1 \oplus \ldots \oplus s_n$ (appends the XOR of all the bits of s)

Candidate PRGs

Consider the following candidates that expand a seed $s = s_1 \ldots s_n \in \{0,1\}^n$

- $G_1(s) = 0s_1 \ldots s_n$ (appends 0 to the front of s)

How might we build a distinguisher for candidate G_1?

- $G_2(s) = s_1 \ldots s_n$ (flips the bits of s)
- $G_3(s) = s_1 \ldots s_n s_1$ (appends the first bit to the end)
- $G_4(s) = s_1 \ldots s_n z$ where $z = s_1 \oplus \ldots \oplus s_n$ (appends the XOR of all the bits of s)

Candidate PRGs

Consider the following candidates that expand a seed $s = s_1 \ldots s_n \in \{0,1\}^n$

- $G_1(s) = 0s_1 \ldots s_n$ (appends 0 to the front of s)
- $G_2(s) =$

How might we build a distinguisher for candidate G_1?

$D(y)$: On input $y \in \{0,1\}^{n+1}$, check if y ends in 0.

If so, output 1. Else, output 0.

- $G_3(s) =$
- $G_4(s) = s_1 \ldots s_n z$ where $z = s_1 \oplus \ldots \oplus s_n$ (appends the XOR of all the bits of s)

Candidate PRGs

Consider the following candidates that expand a seed $s = s_1 \ldots s_n \in \{0,1\}^n$

- $G_1(s) = 0s_1 \ldots s_n$ (appends 0 to the front of s)

How might we build a distinguisher for candidate G_1?

- $G_2(s) =$

$D(y)$: On input $y \in \{0,1\}^{n+1}$, check if y ends in 0.

If so, output 1. Else, output 0.

- $G_3(s) =$

- $\Pr_{s \leftarrow U_n}\left[D\left(G_1(s)\right) = 1\right] = 1$ $\oplus \ldots \oplus s_n$ (appe $\Pr_{r \leftarrow U_{n+1}}\left[D\left(r\right) = 1\right] = 1/2$

Notation: $s \leftarrow U_n$ means drawing s uniformly from $\{0,1\}^n$

Candidate PRGs

Consider the following candidates that expand a seed $s = s_1 \dots s_n \in \{0,1\}^n$

- $G_1(s) = 0s_1 \dots s_n$ (appends 0 to the front of s)
- $G_2(s) = \bar{s}_1 \dots \bar{s}_n$ (flips the bits of s)
- $G_3(s) = s_1 \dots s_n s_1$ (appends the first bit to the end)
- $G_4(s) = s_1 \dots s_n z$ where $z = s_1 \oplus \dots \oplus s_n$ (appends the XOR of all the bits of s)

Candidate PRGs

Consider the following candidates that expand a seed $s = s_1 \ldots s_n \in \{0,1\}^n$

- $G_1(s) = 0s_1 \ldots s_n$ (appends 0 to the front of s)
- $G_2(s) = \bar{s}_1 \ldots \bar{s}_n$ (flips the bits of s)
- $G_3(s) = s_1 \ldots s_n s_1$ (appends the first bit to the end)
- $G_4(s) = s_1 \ldots s_n z$ where $z = s_1 \oplus \ldots \oplus s_n$ (appends the XOR of all the bits of s)

None of these are PRGs!

Do PRGs Exist?

Claim: For any deterministic, expanding $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$, the following distinguisher D, which runs in exponential time, distinguishes the output of G from random with high probability.

$D(y)$: On input $y \in \{0,1\}^{\ell(n)}$:

For all $s \in \{0,1\}^n$, check if $G(s) = y$. If so, output 1.

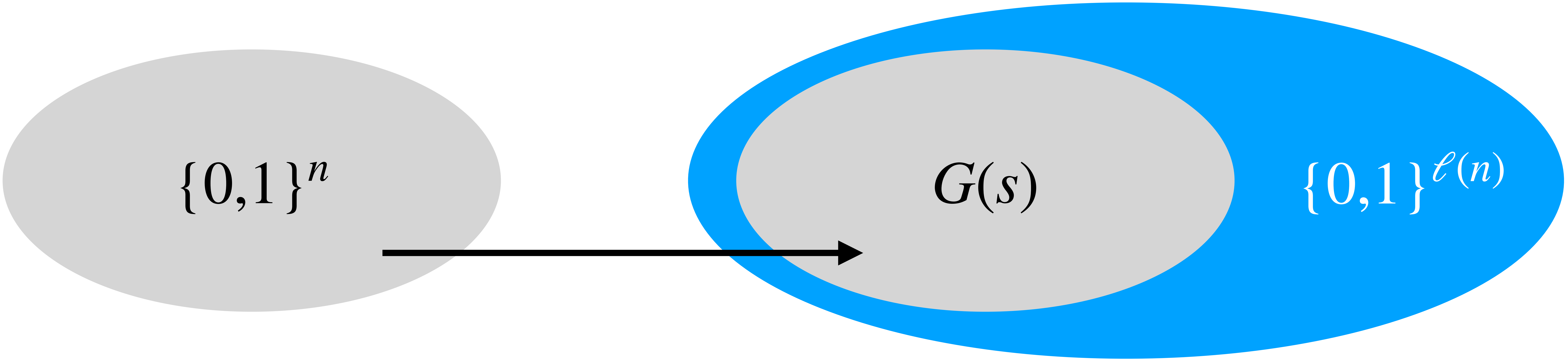
If there are no $s \in \{0,1\}^n$ where $G(s) = y$, then output 0.

Do PRGs Exist?

Proof idea: Probability of outputting 1 when given $G(s)$ is 1.

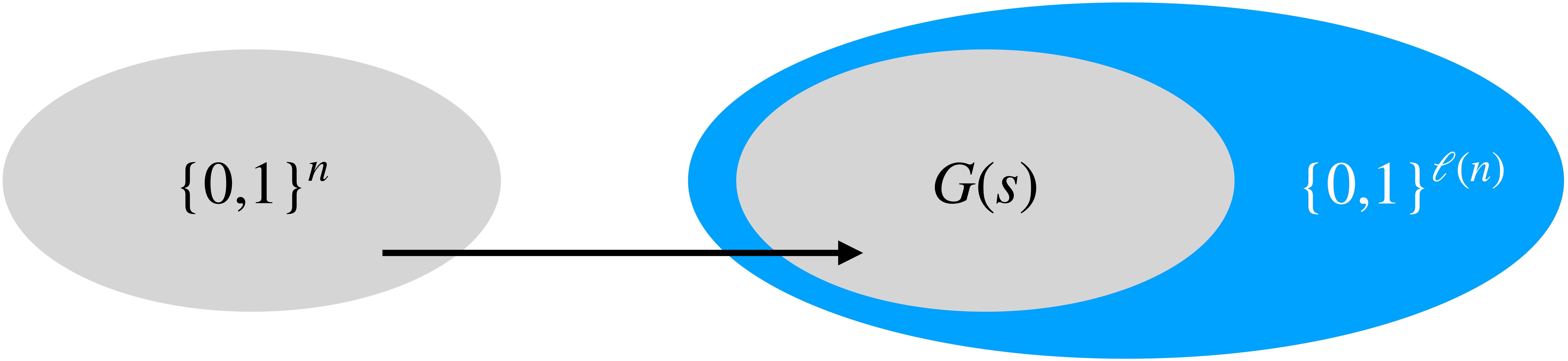
Do PRGs Exist?

Proof idea: Probability of outputting 1 when given $G(s)$ is 1.



Do PRGs Exist?

Proof idea: Probability of outputting 1 when given $G(s)$ is 1.



The probability $r \leftarrow U_{\ell(n)}$ is in the range of G is $\leq \dfrac{2^n}{2^{\ell(n)}}$.

Since $\ell(n) > n$, we have probability of outputting 1 when given r is $\leq 1/2$.
Therefore the difference is is non-negligible.

Do PRGs Exist?

Notice that the distinguisher just relies on checking if $y \in \{0,1\}^{\ell(n)}$ is a possible output of G

- If there were an **efficient** way of checking this, then G cannot be a PRG

Corollary: If $P = NP$, then PRGs do not exist.

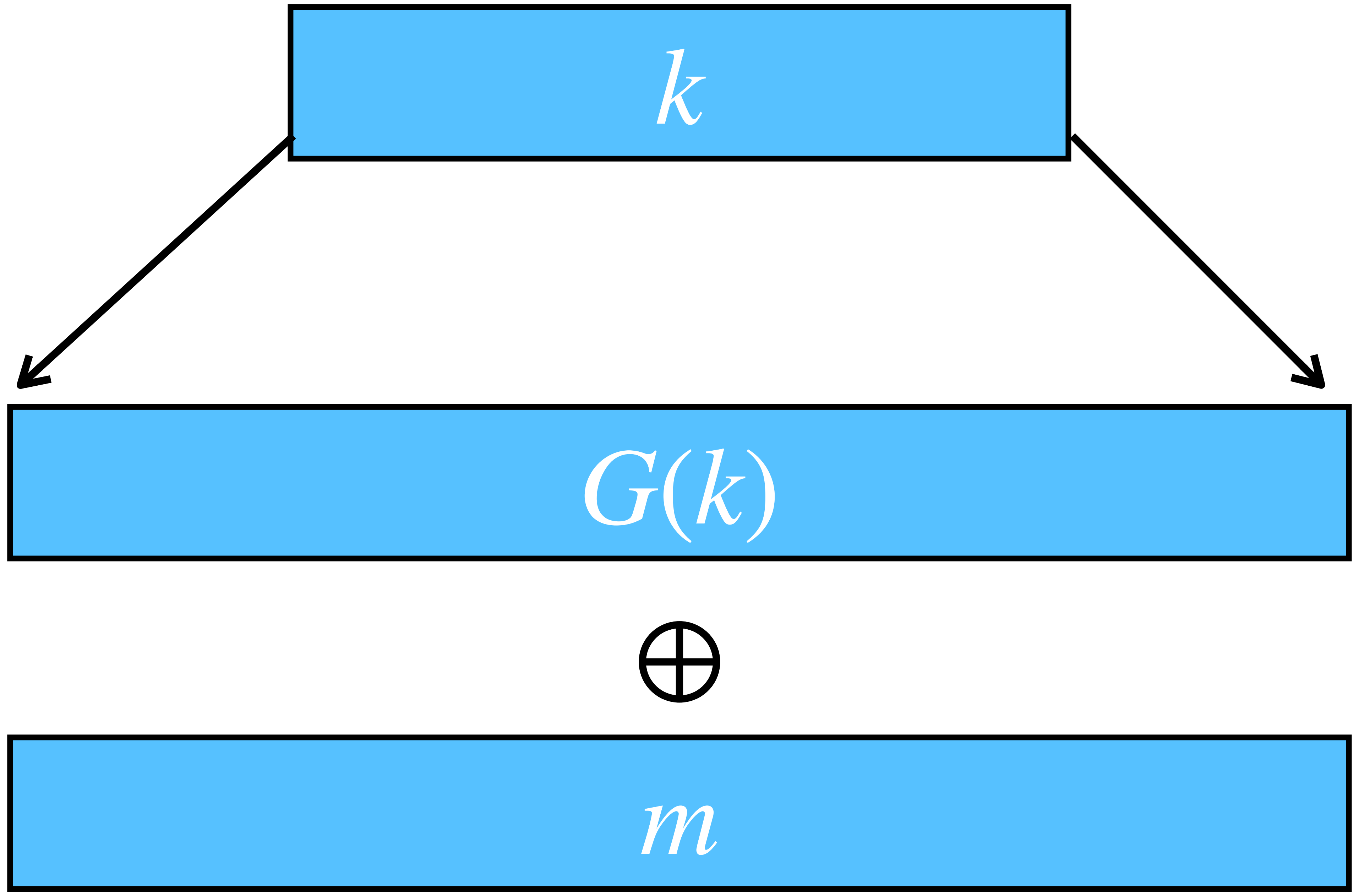
Proof idea: Similar as before but use $P = NP$ to check in poly-time if the input is in the range of G

But also, if PRGs exist, then $P \neq NP$!

- Current constructions rely on various computational assumptions that we'll see later

One-Time Pad Using a PRG

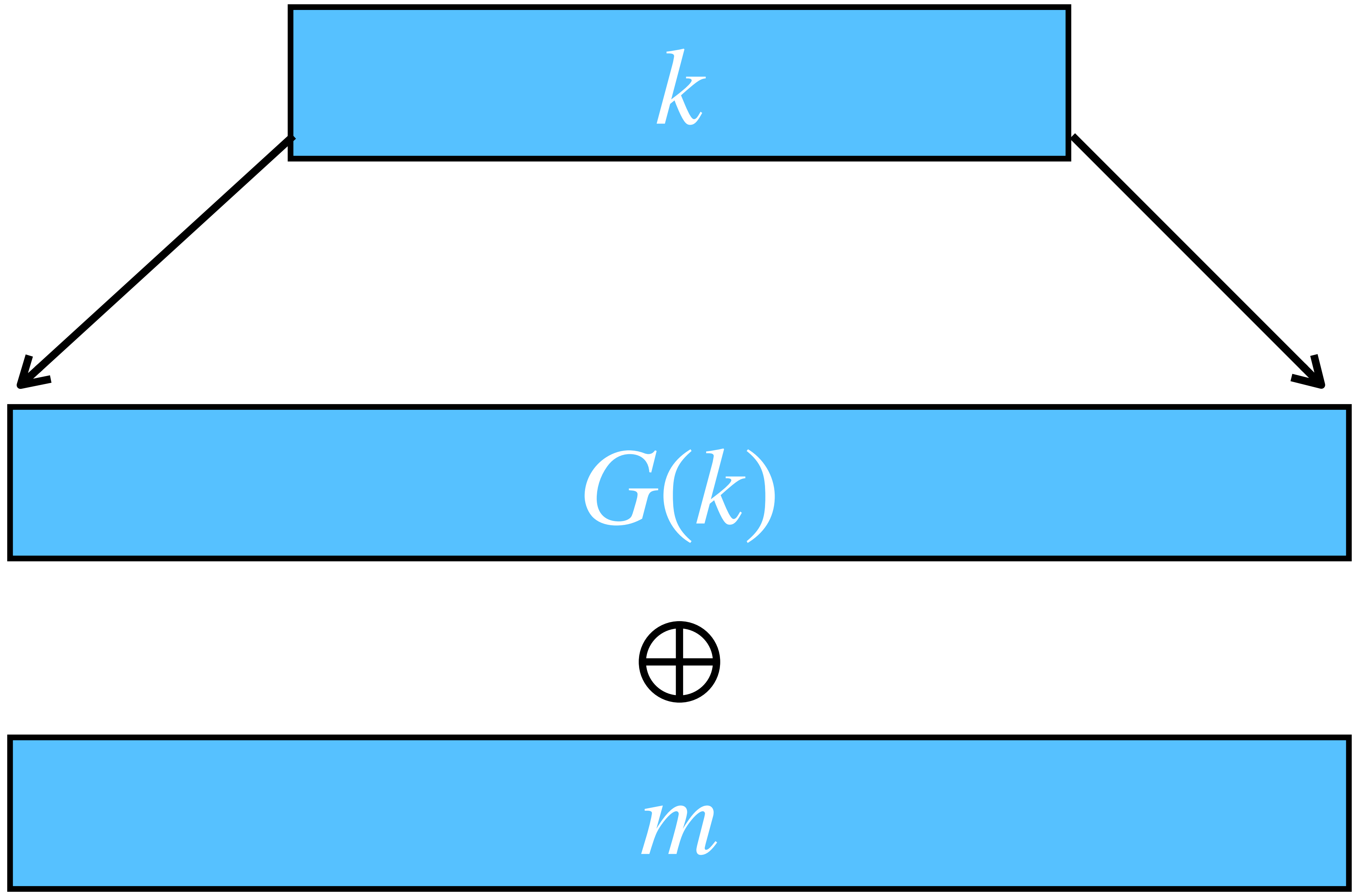
- Let G be a PRG with expansion $\ell(n)$ (i.e., $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$)
- $\mathscr{K} = \{0,1\}^n$, and $\mathscr{M} = \mathscr{C} = \{0,1\}^{\ell(n)}$
- Gen(1^n) samples $k \leftarrow \{0,1\}^n$
- Enc$(k, m) = m \oplus G(k)$
- Dec$(k, c) = c \oplus G(k)$



One-Time Pad Using a PRG

- Let G be a PRG with expansion $\ell(n)$ (i.e., $G : \{0,1\}^n \to \{0,1\}^{\ell(n)}$)
- $\mathscr{K} = \{0,1\}^n$, and $\mathscr{M} = \mathscr{C} = \{0,1\}^{\ell(n)}$
- Gen(1^n) samples $k \leftarrow \{0,1\}^n$
- Enc$(k, m) = m \oplus G(k)$
- Dec$(k, c) = c \oplus G(k)$

Theorem: If G is a PRG, then this scheme is EAV-secure



One-Time Pad Using a PRG

Theorem: If G is a PRG, then this scheme is EAV-secure

"If there were such an A that could break EAV-Security of our scheme, we could use it to break the security of G. But since we assume G is secure, then such an A can't exist"

Paradigm: Proof by Reduction

- Given an adversary A for the encryption scheme, construct a distinguisher D for the PRG
- D uses A to break the security of the PRG
- D's efficiency and advantage are polynomially related to A's

One-Time Pad Using a PRG

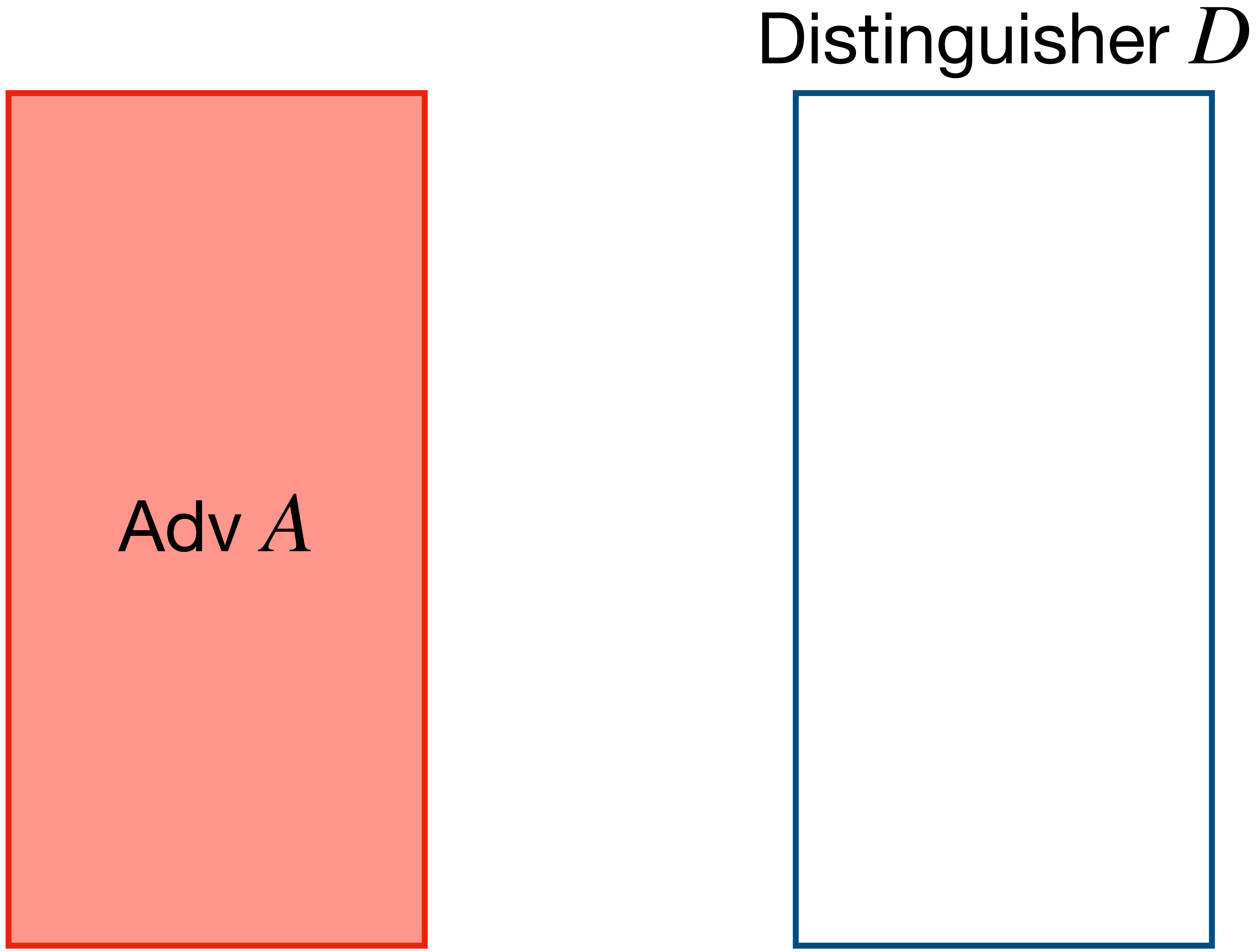
Theorem: If G is a PRG, then this scheme is EAV-secure

Proof: Assume towards contradiction that there exists a PPT adversary A and a polynomial $p(n)$ s.t.

$$\Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\Pi,A}(n) = 1] \geq 1/2 + 1/p(n)$$

We will show that there exists a PPT distinguisher D and a polynomial $q(n)$ s.t.

$$\left| \Pr_{s \leftarrow U_n}[D(G(s)) = 1] - \Pr_{r \leftarrow U_{\ell(n)}}[D(r) = 1] \right| \geq 1/q(n)$$

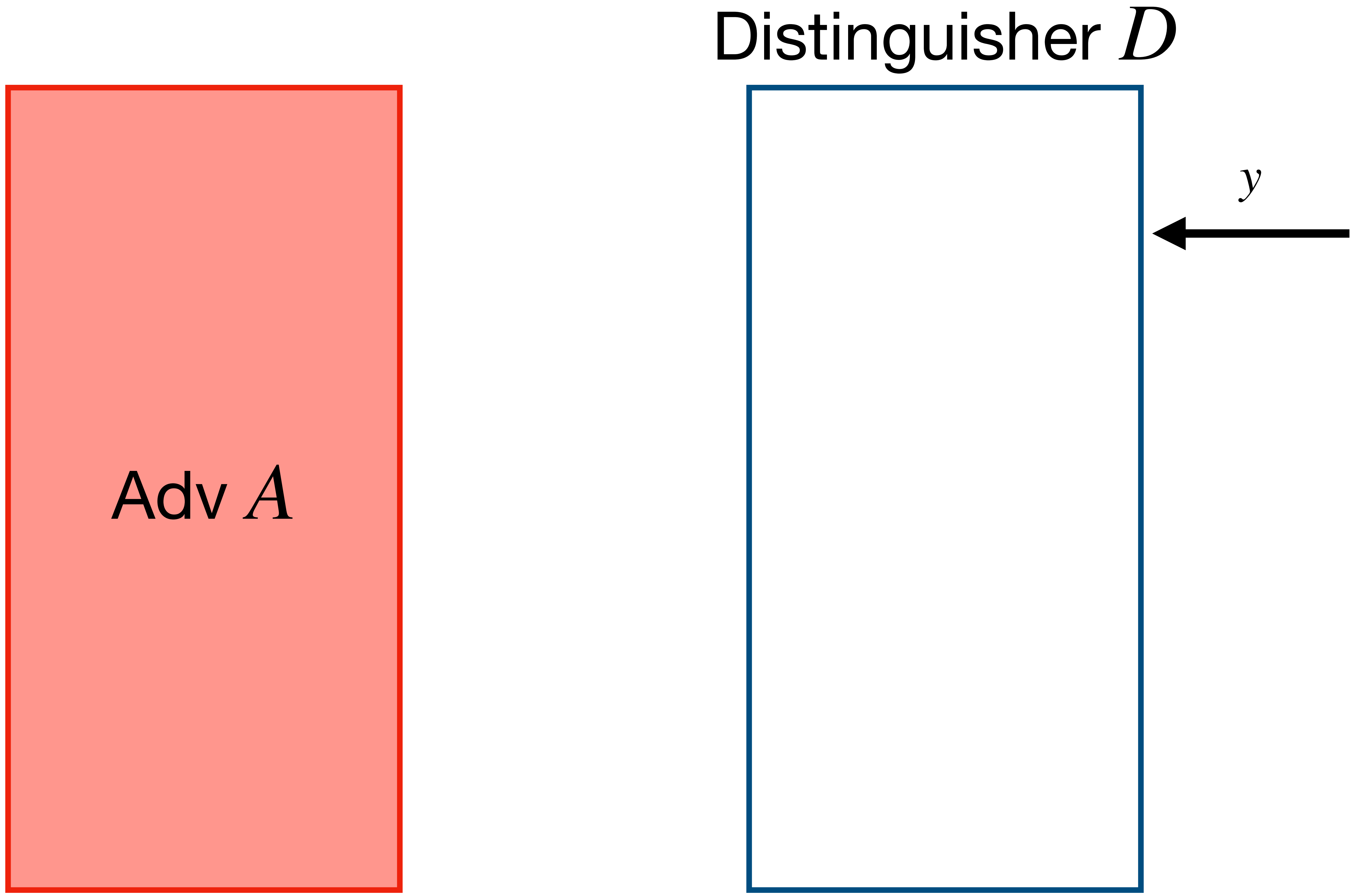


One-Time Pad Using a PRG

Theorem: If G is a PRG, then this scheme is EAV-secure

Proof: Assume towards contradiction that there exists a PPT adversary A and a polynomial $p(n)$ s.t.
$\Pr[\text{PrivK}^{\text{eav}_{\Pi,A}}(n) = 1] \geq 1/2 + 1/p(n)$

We will show that there exists a PPT distinguisher D and a polynomial $q(n)$ s.t.
$|\Pr_{s \leftarrow U_n}[D(G(s)) = 1] - \Pr_{r \leftarrow U_{\ell(n)}}[D(r) = 1]| \geq 1/q(n)$



Distinguisher D

y

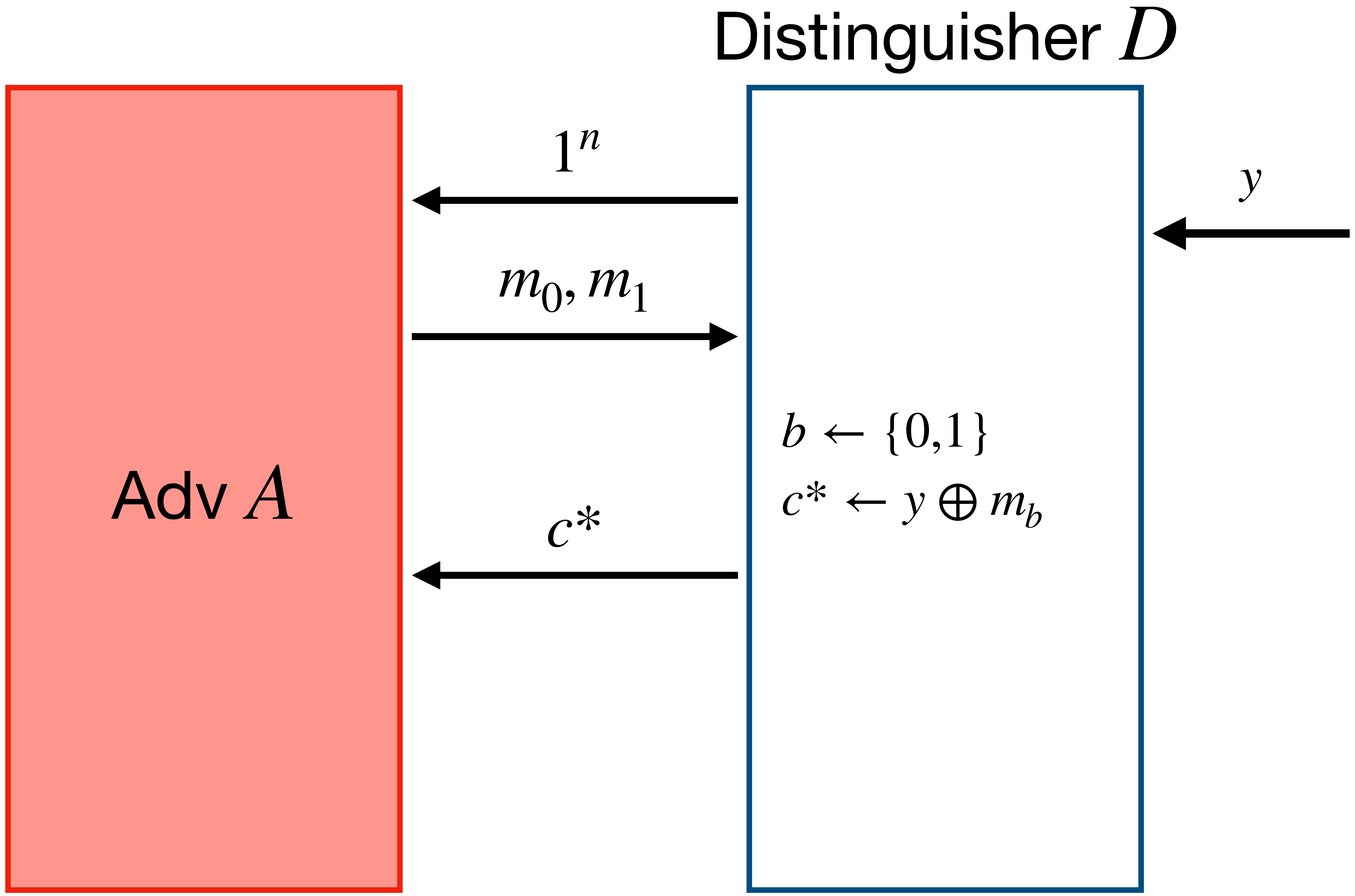
Adv A

One-Time Pad Using a PRG

Theorem: If G is a PRG, then this scheme is EAV-secure

Proof: Assume towards contradiction that there exists a PPT adversary A and a polynomial $p(n)$ s.t.
$\Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\Pi,A}(n) = 1] \geq 1/2 + 1/p(n)$

We will show that there exists a PPT distinguisher D and a polynomial $q(n)$ s.t.
$| \Pr_{s \leftarrow U_n}[D(G(s)) = 1] - \Pr_{r \leftarrow U_{\ell(n)}}[D(r) = 1] | \geq 1/q(n)$

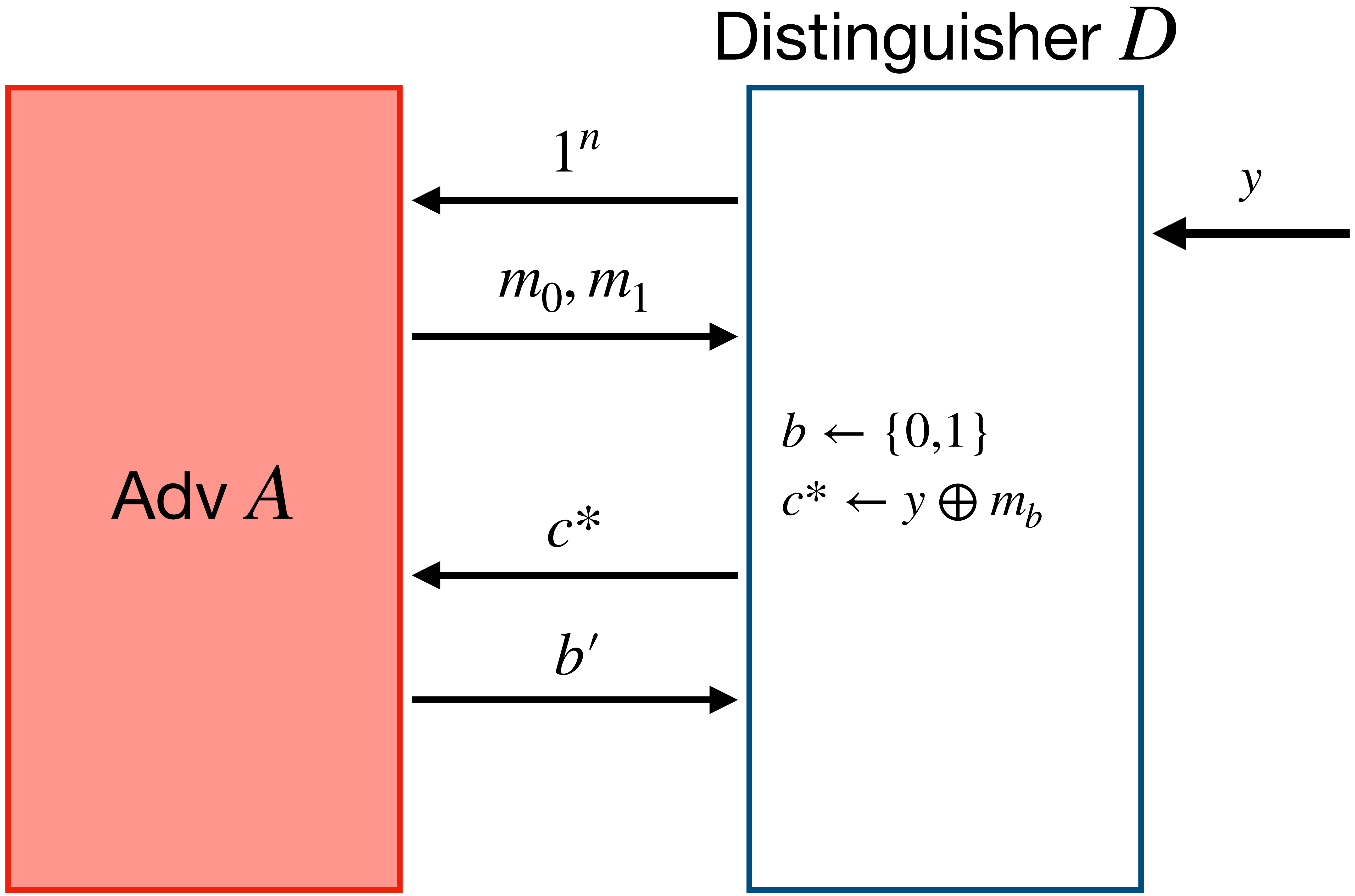


One-Time Pad Using a PRG

Theorem: If G is a PRG, then this scheme is EAV-secure

Proof: Assume towards contradiction that there exists a PPT adversary A and a polynomial $p(n)$ s.t.
$\Pr[\mathsf{PrivK}^{\mathsf{eav}}_{\Pi,A}(n) = 1] \geq 1/2 + 1/p(n)$

We will show that there exists a PPT distinguisher D and a polynomial $q(n)$ s.t.
$| \Pr_{s \leftarrow U_n}[D(G(s)) = 1] - \Pr_{r \leftarrow U_{\ell(n)}}[D(r) = 1] | \geq 1/q(n)$

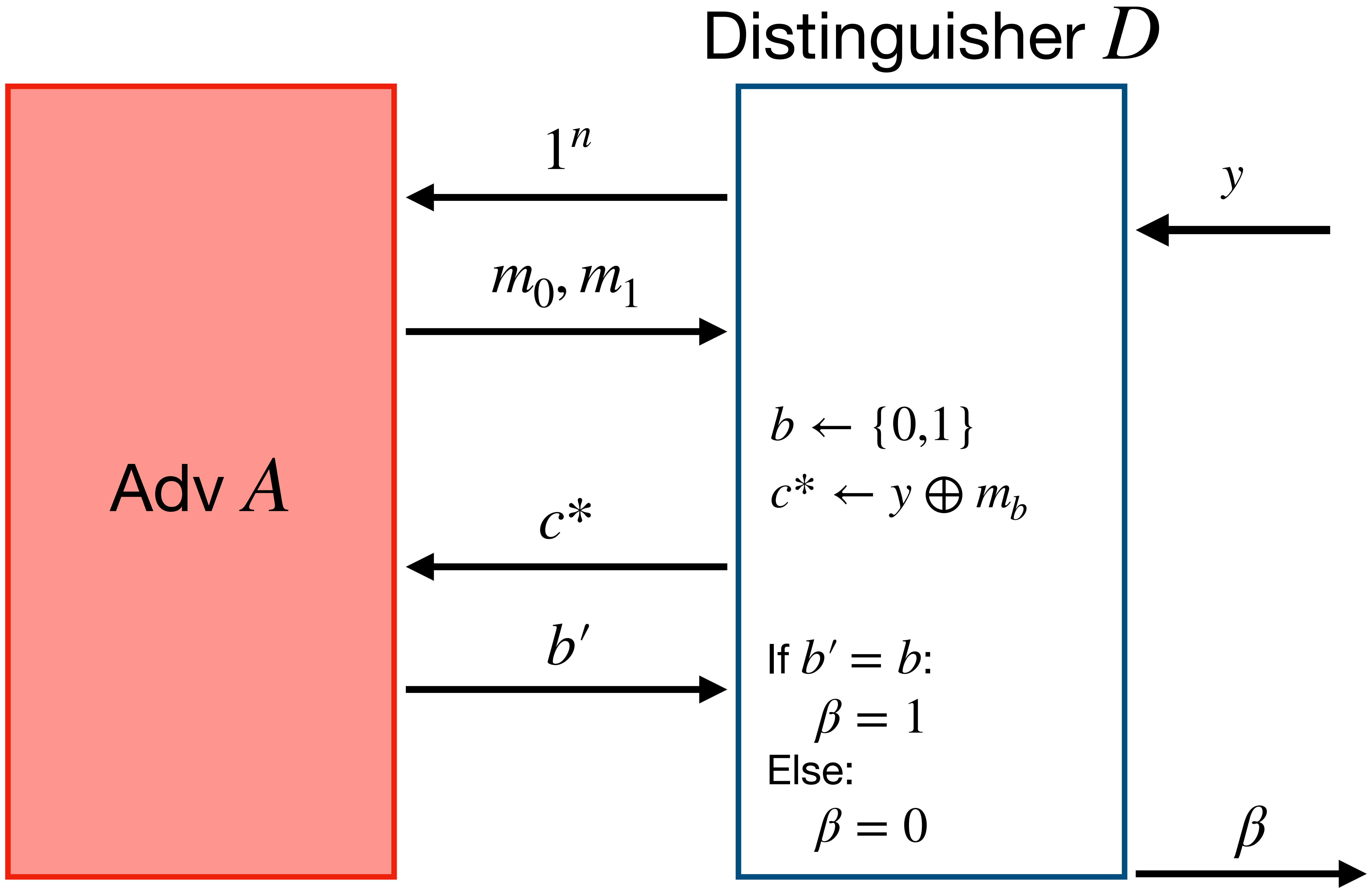


One-Time Pad Using a PRG

Theorem: If G is a PRG, then this scheme is EAV-secure

Proof: Assume towards contradiction that there exists a PPT adversary A and a polynomial $p(n)$ s.t.
$\Pr[\mathsf{PrivK}^{\mathsf{eav}_{\Pi,A}}(n) = 1] \geq 1/2 + 1/p(n)$

We will show that there exists a PPT distinguisher D and a polynomial $q(n)$ s.t.
$| \Pr_{s\leftarrow U_n}[D(G(s)) = 1] - \Pr_{r\leftarrow U_{\ell(n)}}[D(r) = 1] | \geq 1/q(n)$



Distinguisher D

Adv A

1^n

m_0, m_1

c^*

b'

y

$b \leftarrow \{0,1\}$
$c^* \leftarrow y \oplus m_b$

If $b' = b$:
    $\beta = 1$
Else:
    $\beta = 0$

β

One-Time Pad Using a PRG

Theorem: If G is a PRG, then this scheme is EAV-secure

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

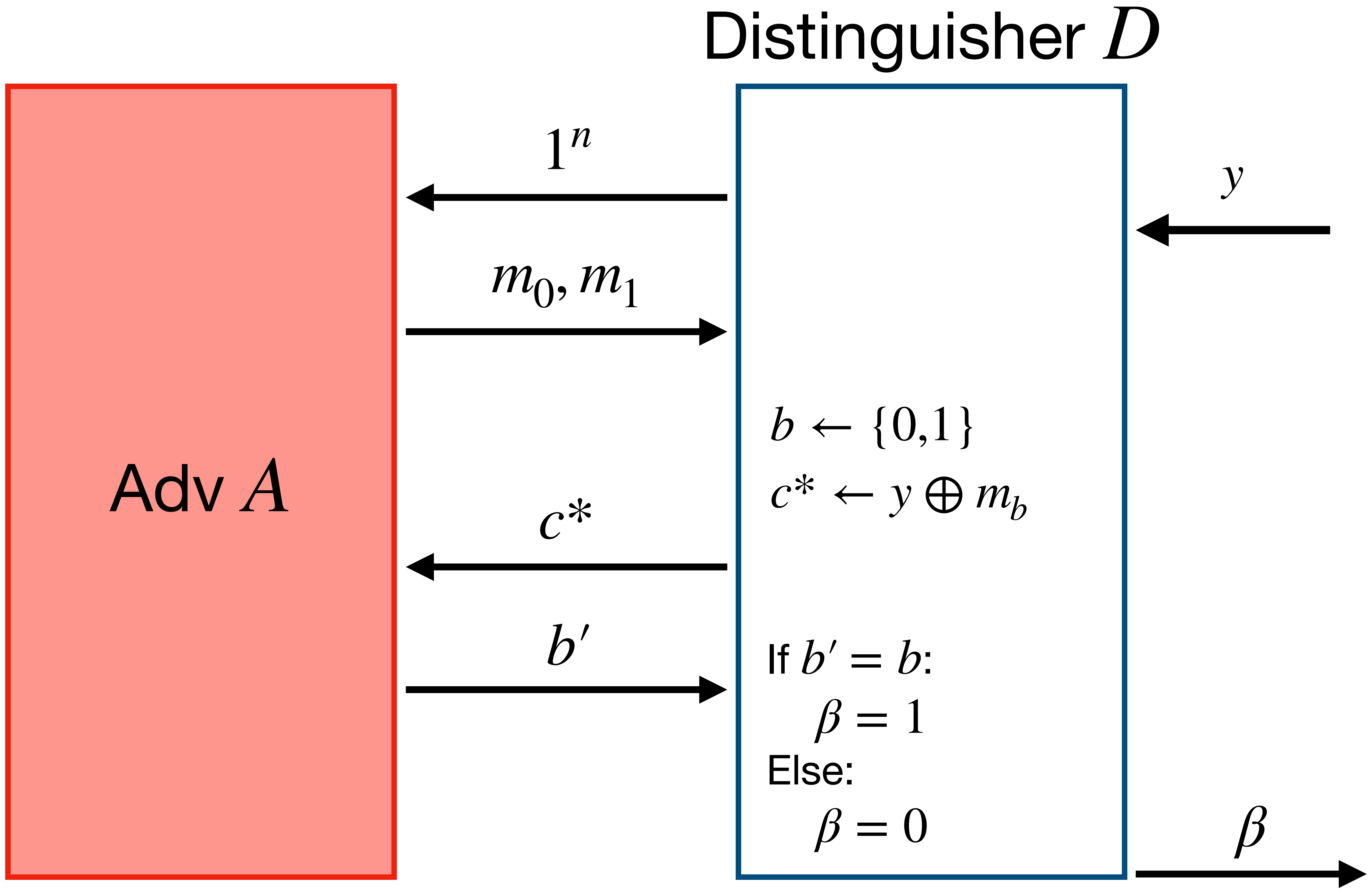
A's view is independent of b

$$\Pr_{r \leftarrow U_{\ell(n)}}[D(r) = 1] = 1/2$$

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

A's view is identical to $\mathsf{PrivK}^{\mathsf{eav}_{\Pi,A}}(n)$

$$\Pr_{s \leftarrow U_n}[D(G(s)) = 1] = \Pr[\mathsf{PrivK}^{\mathsf{eav}_{\Pi,A}}(n) = 1] \geq 1/2 + 1/p(n)$$



Distinguisher D

Adv A

1^n

m_0, m_1

c^*

b'

y

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

$c^* \leftarrow y \oplus m_b$

If $b' = b$:

$\beta = 1$

Else:

$\beta = 0$

β

Next Time

- More on PRGs
    - Also more on reductions?
- Pseudorandom Functions (PRFs)