COMS BC3262: Introduction to Cryptography

Lecture 2: Perfect Secrecy and Computational Security

January 26, 2026

Logistics Recap

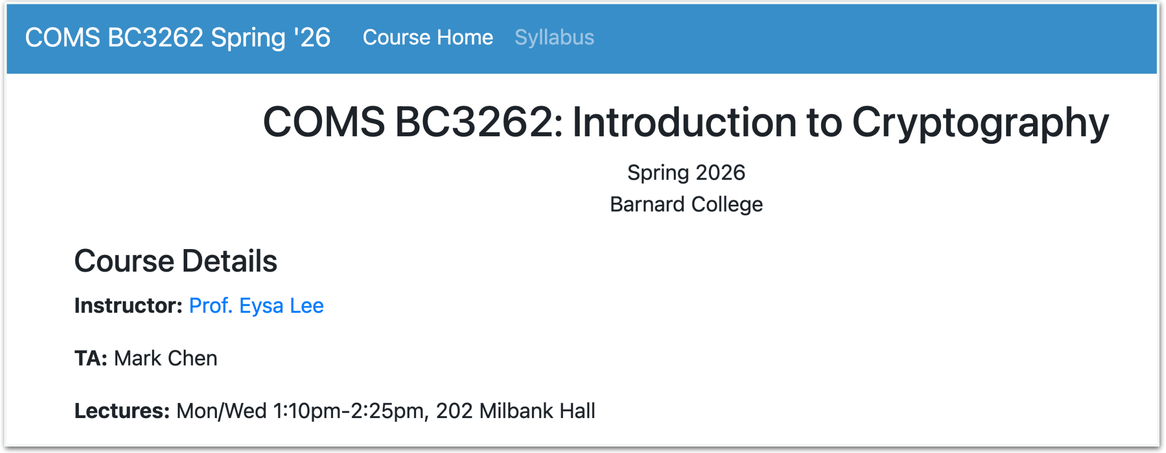
Course website: https://www.eysalee.com/courses/s26/bc3262.html

PDF of slides will be posted under Lecture Schedule around the start of class (or shortly after)

Attendance is expected but not enforced

You're encouraged to work on HWs with other students in the course.
No external resources (including AI) are allowed.
I take cheating personally :(

Please make use of the EdStem for questions about the material!



COMS BC3262 Spring '26 Course Home Syllabus

COMS BC3262: Introduction to Cryptography

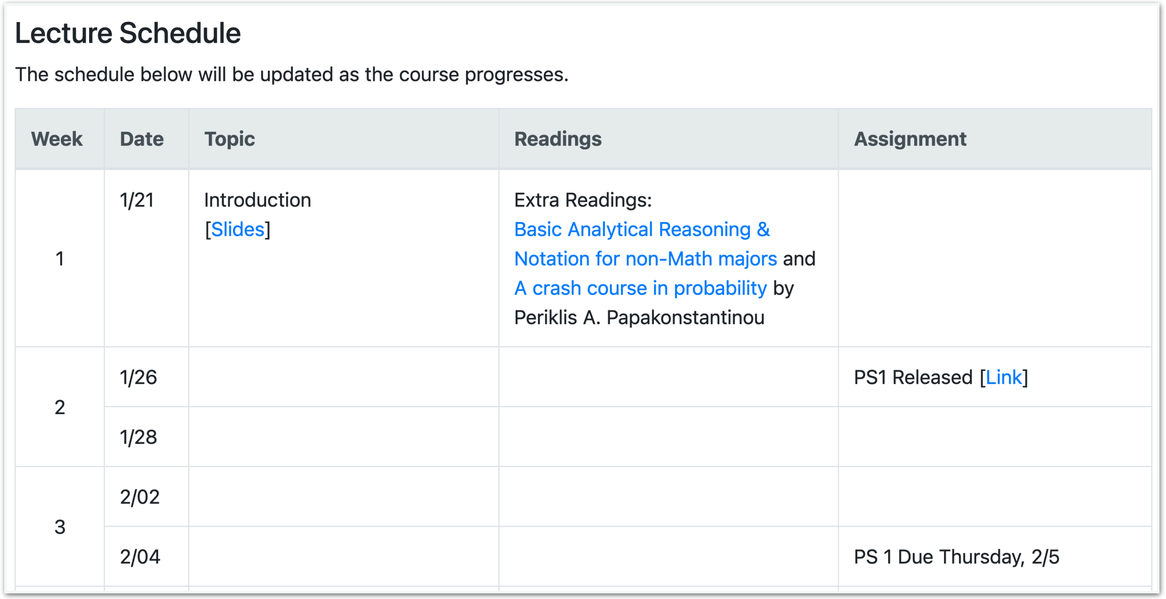
Spring 2026
Barnard College

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	Readings	Assignment
1	1/21	Introduction [Slides]	Extra Readings: Basic Analytical Reasoning & Notation for non-Math majors and A crash course in probability by Periklis A. Papakonstantinou	
2	1/26			PS1 Released [Link]
	1/28			
3	2/02			
	2/04			PS 1 Due Thursday, 2/5

More Logistics

My office hours are Mondays 3-5 in Milstein 512

Written exams (midterm and final). If needed, contact CARDS for accommodations

Similar to other theory classes, grades will be curved. You will not be curved down. (90 will always be at least a A-, 80 will always be at least a B-, …)

Regrade window is 2 weeks after you receive your grade. Please only ask for regrades if something was legitimately graded incorrectly. I reserve the right to lower your grade if the original grade was too generous.

Feedback on assignments may be limited for the sake of timely grading. You may visit my or Mark's office hours for detailed feedback.

I’m sorry this is logistics but longer

Problem Sets due on Thursdays will be discussed the following Monday.

Please speak up if I’m going too fast or if something is unclear!
This is my first time teaching this class, and it’s helpful for everyone (including future students) to know if we should spend more time on certain topics

If you notice something wrong on the course schedule (e.g., university holiday is not indicated), please tell me! I’m surprisingly bad at academic calendars.

Similarly, if something conflicts with a religious holiday you observe, let me know as soon as possible! I’m *really* bad at calendars in general.

If you send me an important email* and don’t receive a response within a week, feel free to send a follow-up. It’s really easy to lose track of emails, and I don’t mind.

*important meaning this is not something that the TA could possibly help you with. If this is something the TA can answer, please post on EdStem

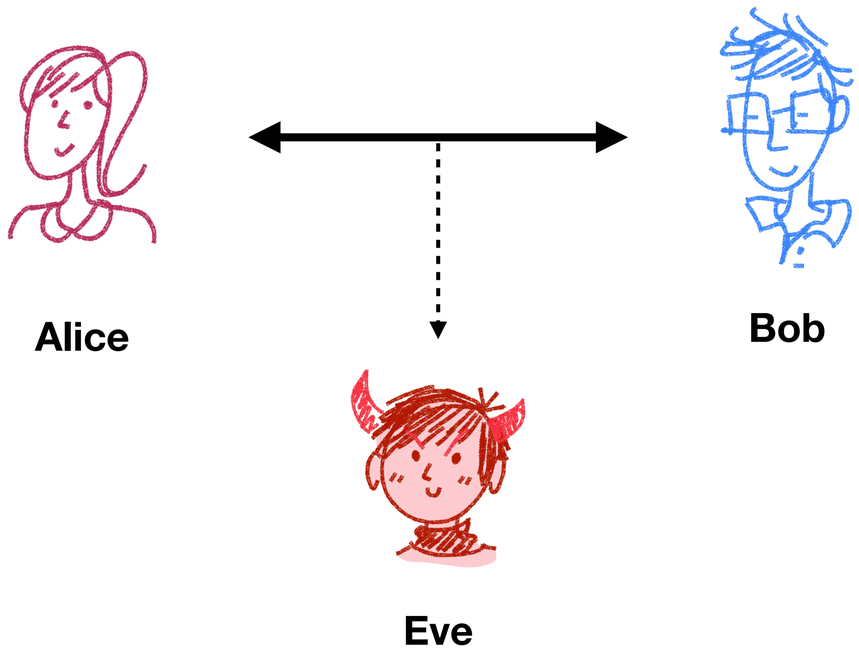
Symmetric-Key Encryption

Last time: Secure Communication

- Alice and Bob wish to communicate secretly, but Eve is able to observe the communication

- How do we formalize this?

  - What parts are public to the adversary? Should be kept secret?



Kerchoff’s Principle

- **Definition:** The only thing that should be assumed private is the key; everything else should be assumed to be public
  - Why?
- Immediate consequence is that there must be randomness involved somewhere in the algorithms!
  - Why?

Symmetric Key Encryption

Syntax: Three algorithms (Gen, Enc, Dec)

- **Key generation**: $k \leftarrow \text{Gen}()$
- **Encryption**: $c \leftarrow \text{Enc}(k, m)$
- **Decryption**: $m = \text{Dec}(k, c)$

$\mathcal{K}$ - Key space
$\mathcal{M}$ - Message space
$\mathcal{C}$ - Ciphertext space

$\leftarrow$ (possibly) random assignment

$=$ deterministic assignment

m

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

c

$m = \text{Dec}(k, c)$

Symmetric Key Encryption

Syntax: Three algorithms (Gen, Enc, Dec)

- **Key generation**: $k \leftarrow \text{Gen}()$
- **Encryption**: $c \leftarrow \text{Enc}(k, m)$
- **Decryption**: $m = \text{Dec}(k, c)$

$\mathscr{K}$ - Key space
$\mathscr{M}$ - Message space
$\mathscr{C}$ - Ciphertext space

$\leftarrow$ (possibly) random assignment

$=$ deterministic assignment

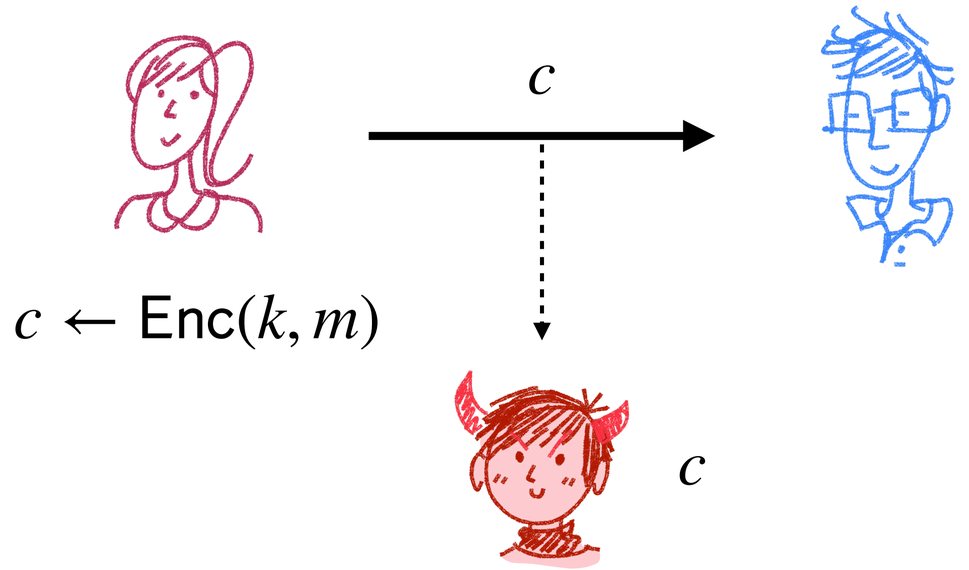
Correctness: $\forall k \in \mathscr{K}, m \in \mathscr{M}$
$\text{Dec}(k, \text{Enc}(k, m)) = m$

Security: Hm…

Defining Security

Consider the following experiment:

- Choose a key $k \leftarrow \text{Gen}()$
- Encrypt a message $c \leftarrow \text{Enc}(k, m)$
- Give the ciphertext c to the adversary

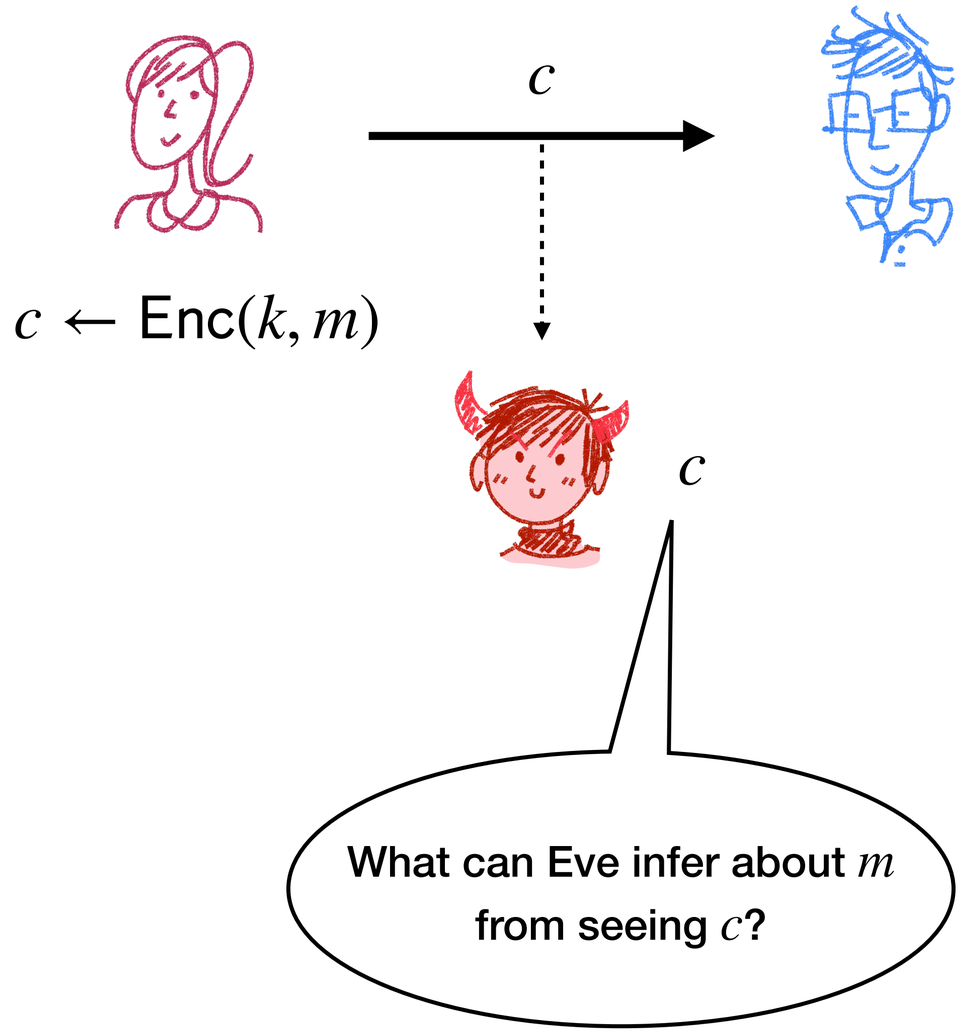


Defining Security

Consider the following experiment:

- Choose a key $k \leftarrow \text{Gen}()$
- Encrypt a message $c \leftarrow \text{Enc}(k, m)$
- Give the ciphertext c to the adversary

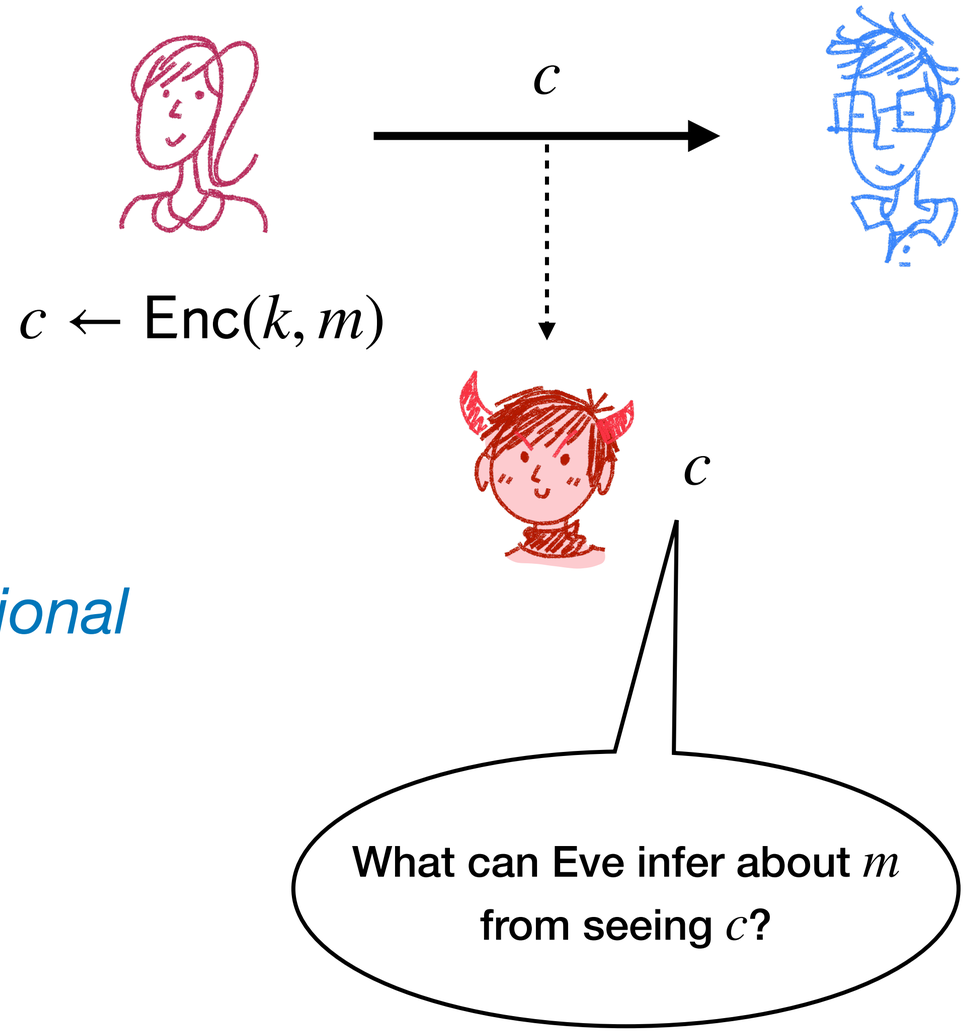
Intuitively, what should security capture?



Defining Security

Consider the following experiment:

- Choose a key $k \leftarrow \text{Gen}()$
- Encrypt a message $c \leftarrow \text{Enc}(k, m)$
- Give the ciphertext c to the adversary



Intuitively, what should security capture?

Adversary should not be able to learn any *additional* information about m from the ciphertext

Perfect Secrecy

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

$$\Pr[M = m \mid C = c] = \Pr[M = m]$$

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**

Another Definition of Perfect Secrecy

For any pair of messages $m_0, m_1 \in \mathcal{M}$, Eve cannot tell if c is an encryption of m_0 or m_1

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

$$\Pr[\text{Enc}(k, m_0) = c : k \leftarrow \text{Gen}] = \Pr[\text{Enc}(k, m_1) = c : k \leftarrow \text{Gen}]$$

This notation is slightly different from last lecture, but it means the same thing

Theorem: These two definitions of perfect secrecy are equivalent

Exclusive OR (XOR)

The **XOR** of 2 bits $a, b \in \{0,1\}$ is defined as follows:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Suppose a, b are sampled at random independently of each other.

Are a and $a \oplus b$ independent from each other?

Exclusive OR (XOR)

For longer strings, XOR is defined bit-wise
e.g., $101 \oplus 011 = 110$

The **XOR** of 2 bits $a, b \in \{0,1\}$ is defined as follows:

a	b	$a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Suppose a, b are sampled at random independently of each other.

Are a and $a \oplus b$ independent from each other?

Exclusive OR (XOR)

The **XOR** of 2 bits $a, b \in \{0,1\}$ is defined as follows:

For longer strings, XOR is defined bit-wise
e.g., $101 \oplus 011 = 110$

m	k	$c = m \oplus k$
0	0	0
0	1	1
1	0	1
1	1	0

Suppose a, b are sampled at random independently of each other.

Are a and $a \oplus b$ independent from each other?

One-Time Pad

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

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

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

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

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

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

> **Recall**: A symmetric-key encryption scheme is **perfectly secret** if for every $m_0, m_1 \in \mathcal{M}$ and for every $c \in \mathcal{C}$ it holds that
>
> $$\Pr[\text{Enc}(k, m_0) = c : k \leftarrow \text{Gen}] = \Pr[\text{Enc}(k, m_1) = c : k \leftarrow \text{Gen}]$$

Proof:

$\Pr[K = k] = 2^{-\ell}$ for every $k \in \{0,1\}^\ell$

For any $m, c \in \{0,1\}^\ell$ it holds that

$$\Pr[\text{Enc}(k, m) = c : k \leftarrow \text{Gen}] = \Pr[m \oplus k = c : k \leftarrow \{0,1\}^\ell] = 2^{-\ell}$$

Therefore, for any $m_0, m_1 \in \mathcal{M}$ and for every $c \in \mathcal{C}$ it holds that

$$\Pr[\text{Enc}(k, m_0) = c : k \leftarrow \text{Gen}] = \Pr[\text{Enc}(k, m_1) = c : k \leftarrow \text{Gen}] = 2^{-\ell}$$

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 $|\mathscr{K}| \geq |\mathscr{M}|$

Proof intuition

Theorem: Let Π be a symmetric-key encryption scheme with key space $\mathcal{K}$ and message space $\mathcal{M}$. If Π is perfectly secret, then $|\mathcal{K}| \geq |\mathcal{M}|$

Proof idea:

Suppose there were fewer keys than messages.

The number of messages that can map to any given ciphertext is bounded by the number of keys.

Therefore, for any given ciphertext, there must exist a message that could *not* have possibly generated this ciphertext.

One-Time Pad Limitations

Proof:

Assume $|\mathcal{K}| < |\mathcal{M}|$. We will show such a scheme cannot be perfect secret.

Let M be the uniform distribution over $\mathcal{M}$, and fix some $m \in \mathcal{M}$.

Fix some $c \in \mathcal{C}$, which is a possible encryption of m.

Let $\mathcal{M}(c)$ be defined as the set of messages $\hat{m}$ such that $\hat{m} = \mathsf{Dec}(\hat{k}, c)$ for some key $\hat{k} \in \mathcal{K}$. Clearly, $|\mathcal{M}(c)| \leq |\mathcal{K}|$.

Thus, the assumption $|\mathcal{K}| < |\mathcal{M}|$ implies $|\mathcal{M}(c)| < |\mathcal{M}|$.

In particular, this means there exists some $m^* \in \mathcal{M}$ such that $m^* \notin \mathcal{M}(c)$.

This implies that $\Pr[M = m^* \,|\, C = c] = 0 \neq 1/|\mathcal{M}| = \Pr[M = m^*]$, which means the scheme is not perfectly secret.

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?

Computational Security

What is “computational” security?

- The information is all there: $\mathrm{Enc}(k, m)$ may completely determine k and m
- It should be **computationally infeasible** to retrieve any useful information

Review: Polynomial vs Exponential

- A polynomial function (over the integers) is of the form

$$f(n) = \sum_{i=0}^{d} a_i n^i = a_0 + a_1 n \ldots a_{d-1} n^{d-1} + a_d n_d$$

  where d is a constant and $a_0, \ldots, a_d$ are integers

- A function f is dominated by a polynomial function if there exists a constant d such that for sufficiently large n it holds that $f(n) < n^d$.
  - Formally, there exists N such that for all $n > N$ if holds that $f(n) < n^d$
  - By abuse of notation, sometimes we also call such f a polynomial, e.g., $n^5 + \log(n)$

- A function f is (dominated by) an exponential function if for sufficient large n it holds that $f(n) < c^{p(n)}$ for a constant c and a polynomial $p(\cdot)$

Inverse Polynomial vs Inverse Exponential

- 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

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

- 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 $\mathsf{negl}(n)$ an arbitrary negligible function

Computational Security

What is “computational” security?

- The information is all there: $\mathrm{Enc}(k, m)$ may completely determine k and m
- It should be **computationally infeasible** to retrieve any useful information

Two realistic relaxations compared to perfect secrecy:

- Security is only preserved against **computationally bounded** adversaries
- Adversaries are allowed to succeed with some **small probability**

Computational Security

Whenever $|\mathcal{K}| < |\mathcal{M}|$, the following attacks always work:

- **Exhaustive search**: Given a ciphertext c, try decrypting with all possible keys (This takes time $|\mathcal{K}|$)
- **Guess the key**: Guess a key $k \in \mathcal{K}$ and use it to decrypt c (This succeeds with probability $1/|\mathcal{K}|$)

Both of these attacks you learn something new about the message, which is not allowed in perfect secrecy

Defining Security: The Concrete Approach

"A scheme is (t, ϵ)-secure if every adversary running for time at most t succeeds in breaking the scheme with probability at most ϵ"

For example:

- $t = 2^{60}$ (order of the number of seconds since the big bang)
- $\epsilon = 2^{-30}$ (expected to occur once every 100 years)
- $\epsilon = 2^{-60}$ (expected to occur once every 100 billion years)

- Very important in practice, may be tailored to specific technology
- In general, hard to analyze
- Not always clear. What we can say if adversary runs for time $2t$ or $t/2$

Defining Security: The Asymptotic Approach

"A scheme is secure if every **probabilistic polynomial-time** (PPT) adversary succeeds in breaking the scheme with only negligible probability"

Definition:

An algorithm A runs in probabilistic polynomial-time if there exists a polynomial $p(\cdot)$ such that, for any input $x \in \{0,1\}^*$ and random tape $r \in \{0,1\}^*$, the computation of $A(x; r)$ terminates within $p(|x|)$ steps.

Defining Security: The Asymptotic Approach

Definition:

An algorithm A runs in probabilistic polynomial-time if there exists a polynomial $p(\cdot)$ such that, for any input $x \in \{0,1\}^*$ and random tape $r \in \{0,1\}^*$, the computation of $A(x; r)$ terminates within $p(|x|)$ steps.

The security parameter:

- Gen takes as input the security parameter 1^n and outputs $k \in \mathcal{K}_n$
- Keys produced by $\mathsf{Gen}(1^n)$ should provide security against adversaries whose running time is polynomial in n (increasing n provides better security)
- $\mathcal{K} = \bigcup_{n \in \mathbb{N}} \mathcal{K}_n, \mathcal{M} = \bigcup_{n \in \mathbb{N}} \mathcal{M}_n, \mathcal{C} = \bigcup_{n \in \mathbb{N}} \mathcal{C}_n$

Why These Choices?

- “Efficient”: Probabilistic polynomial time (PPT)
- “Negligible”: Smaller than any inverse polynomial

Intuitively well-behaved under composition:

- $\text{poly}(n) \times \text{poly}(n) = \text{poly}(n)$

  Polynomially many invocations of a PPT algorithm is still a PPT algorithm

Why These Choices?

Intuitively well-behaved under composition:

- $\text{negl}(n) + \text{negl}(n) = \text{negl}(n)$

  Proof: Let μ_1 and μ_2 be negligible functions and denote $\mu = \mu_1 + \mu_2$.

  Let $p(\cdot)$ be a positive polynomial and denote $q = 2p$. Then,

  1. There exists N_1 such that for every $n > N_1$ it holds that $\mu_1(n) < 1/q(n)$
  2. There exists N_2 such that for every $n > N_2$ it holds that $\mu_2(n) < 1/q(n)$

  Therefore, for every $N = \max(N_1, N_2)$ it holds that every $n > N$

$$\mu(n) = \mu_1(n) + \mu_2(n) < \frac{1}{q(n)} + \frac{1}{q(n)} = \frac{2}{q(n)} = \frac{2}{2p(n)} = \frac{1}{q(n)}$$

Why These Choices?

Intuitively well-behaved under composition:

- $\text{poly}(n) \times \text{negl}(n) = \text{negl}(n)$

  Polynomial many invocations of a PPT algorithm that succeeds with a negligible probability is an algorithm that succeeds with negligible probability overall

General Comments on Security Definitions

- In our definitions, we constrain the **running time** of the adversary (polynomial) as well as specify **what access** they get (e.g., adversary gets the ciphertext)

- We do not make any assumptions about the **adversarial strategy**

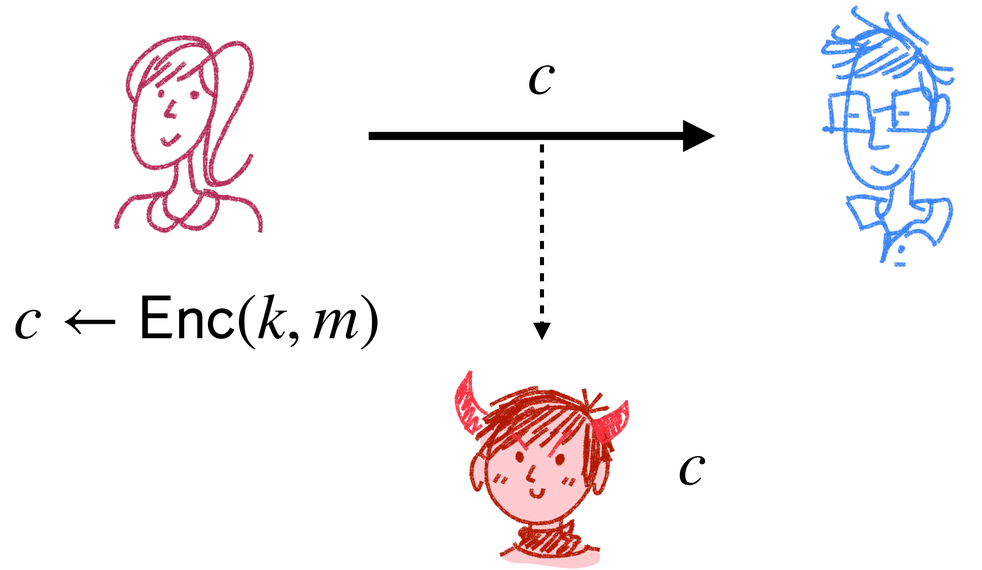
  - If something is secure, it should work for any adversarial strategy (otherwise we can convert it to an attack on some underlying assumption)

  - If something is not secure, we can show an explicit attack

Relaxing Our Setting

What are we trying to capture?

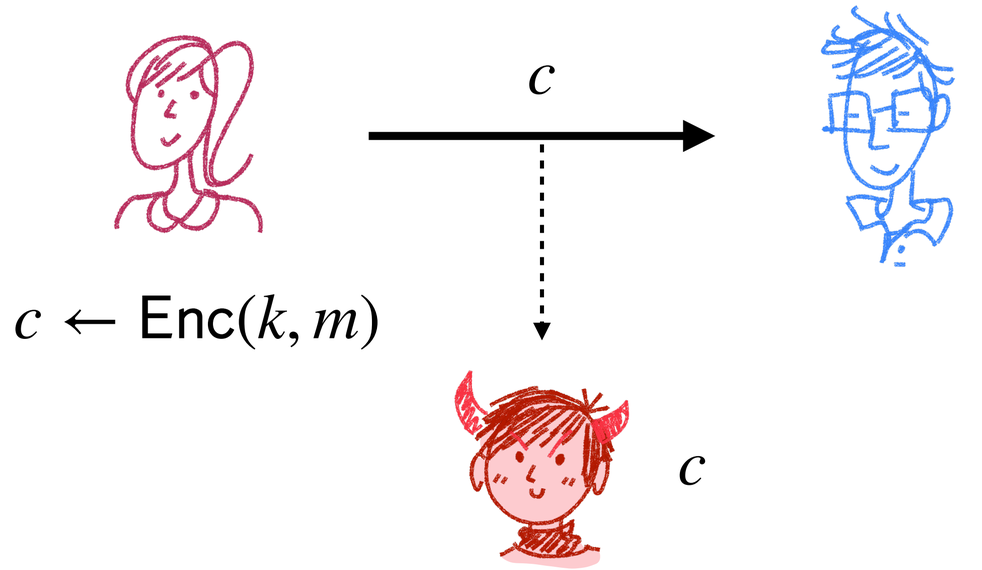
- Eve sees a ciphertext, but she shouldn't be able to tell if it's an encryption of this message or that message



Relaxing Our Setting

What are we trying to capture?

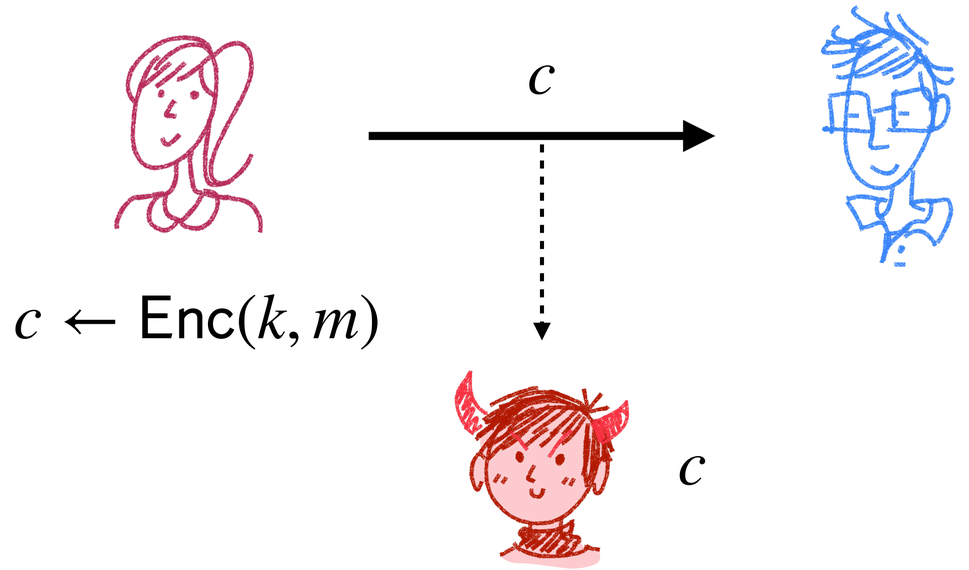
- Eve sees a ciphertext, but she shouldn't be able to tell **in polynomial time** if it's an encryption of this message or that message



Relaxing Our Setting

What are we trying to capture?

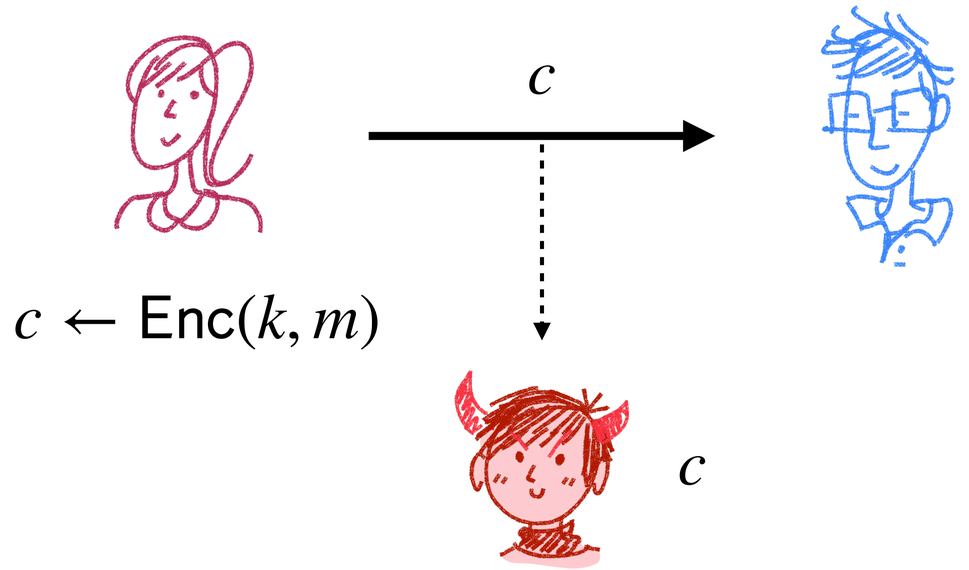
- Eve sees a ciphertext, but she shouldn't be able to tell **in polynomial time** if it's an encryption of this message or that message **except with negligible probability**



Relaxing Our Setting

What are we trying to capture?

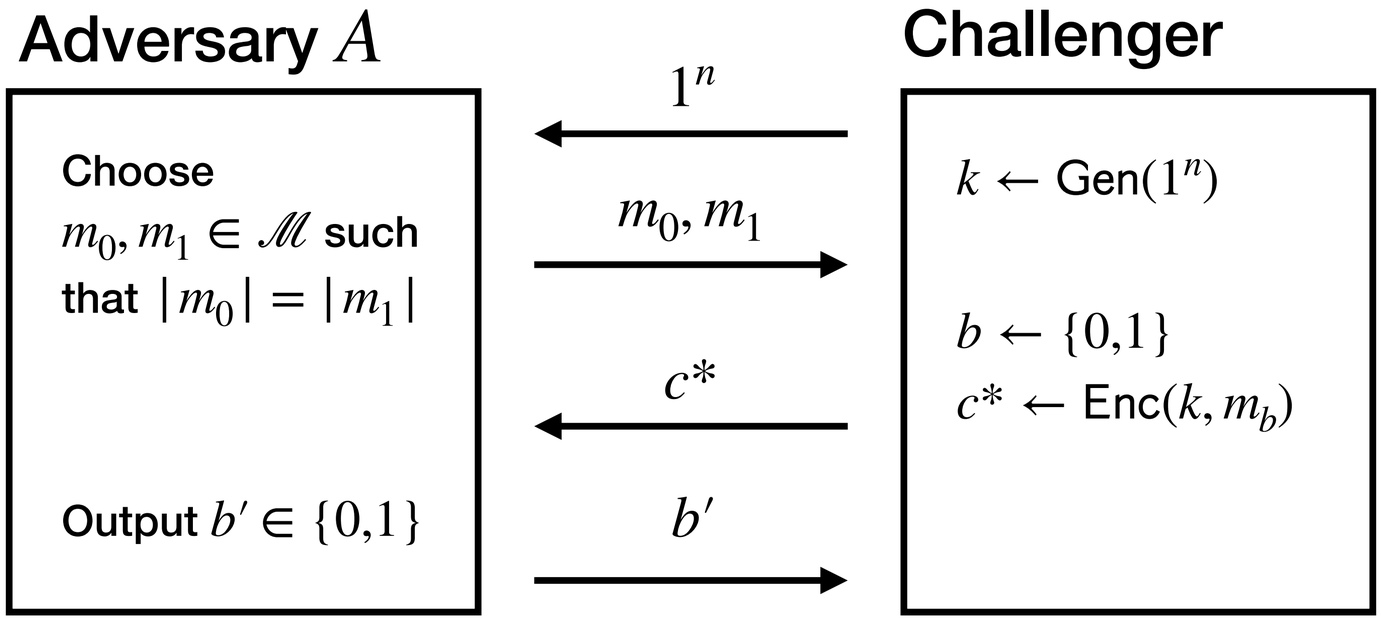
- Eve sees a ciphertext, but she shouldn't be able to tell **in polynomial time** if it's an encryption of this message or that message **except with negligible probability**



This is the idea behind **Indistinguishable Encryptions**

Indistinguishable Encryptions

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



A wins if $b' = b$

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

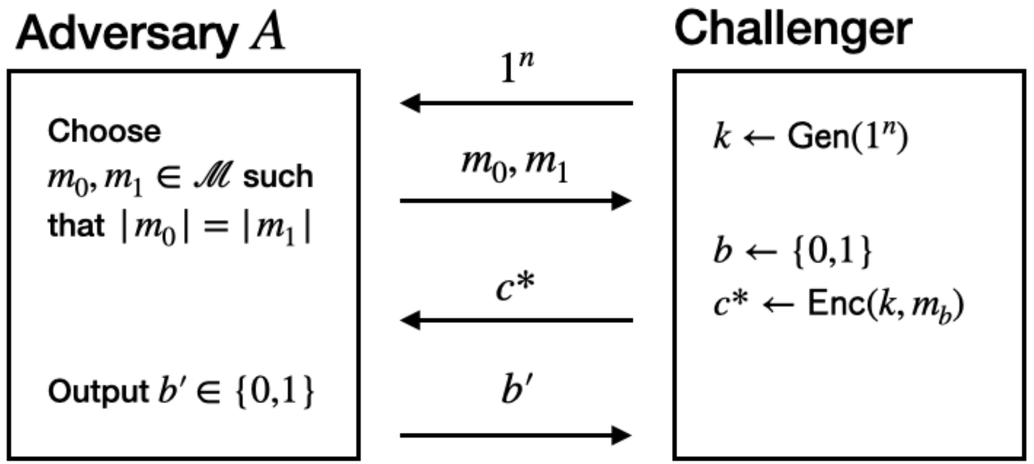
Indistinguishable Encryptions

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

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

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



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

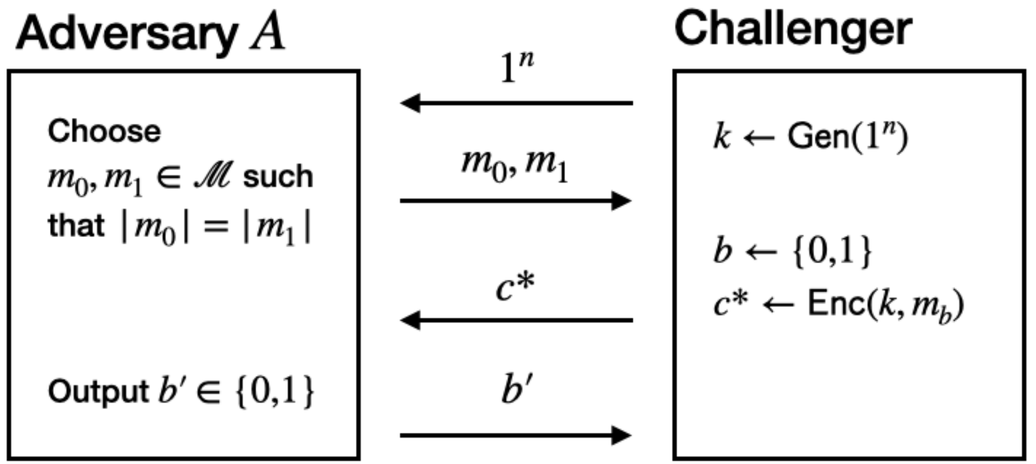
Indistinguishable Encryptions

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

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

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



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

Note: If we replace the "for all PPT adversaries" with "for all algorithms" and set $\epsilon(n) = 0$, this definition becomes equivalent to perfect secrecy

Next Time

- Pseudorandom Number Generators!

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]		PS1 Released [Link] [Template]