COMS BC3262: Introduction to Cryptography

Lecture 1: Introduction

January 21, 2026

What is Cryptography?

- Oxford Dictionary: “the art of writing or solving codes”
  - Greek: *κρυπτός γράφειν* *(kryptós graphein)*
  - English: *hidden writing*

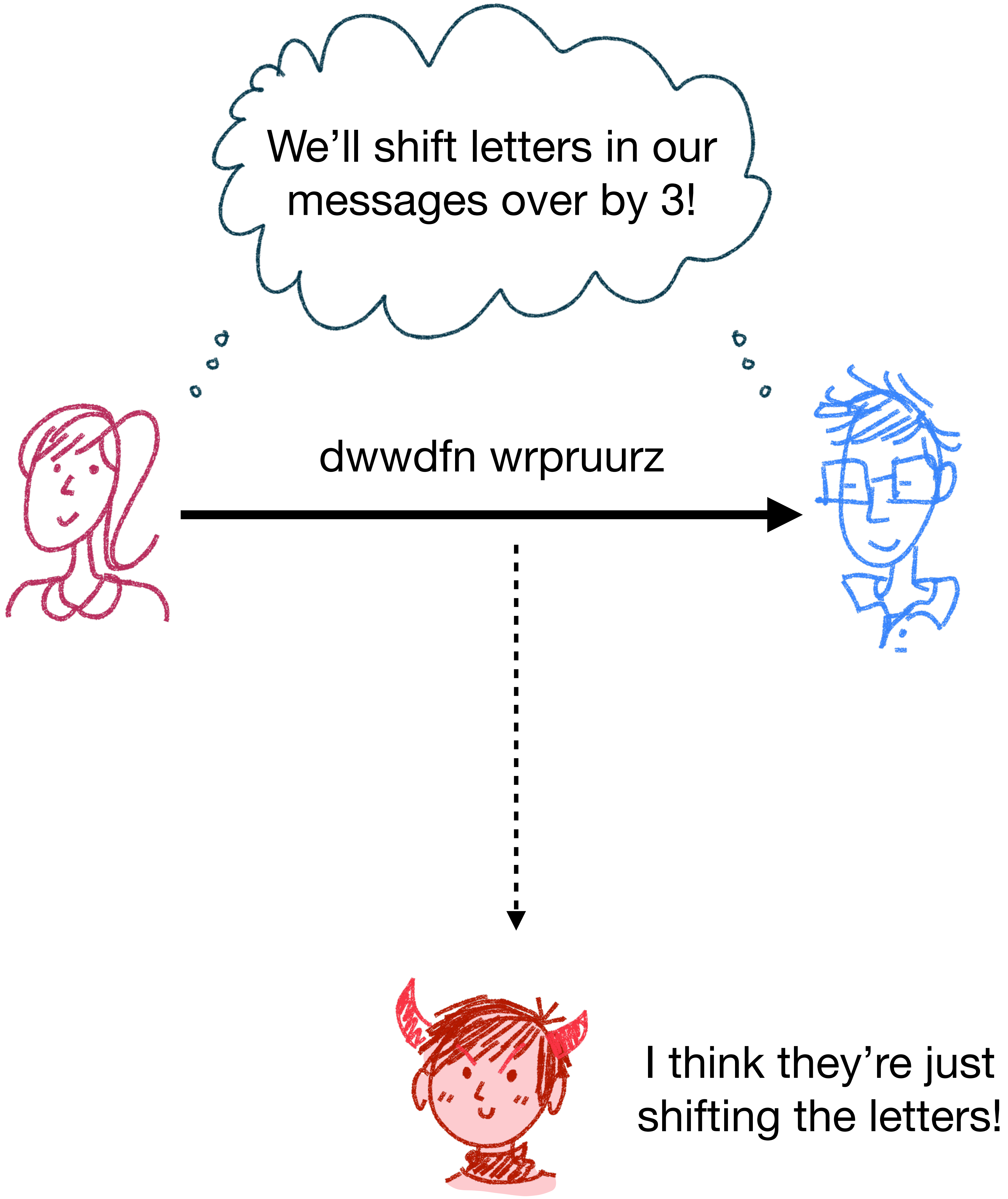
What is Cryptography?

- Oxford Dictionary: “the art of writing or solving codes”
    - Greek: *κρυπτός γράφειν* *(kryptós graphein)*
    - English: *hidden writing*
- For many centuries, focused exclusively on secure communication
    - Consumers were military and intelligence organizations
- Constructions relied on **creativity** and **personal skill**

We want to coordinate an attack, but what if Eve intercepts?

Historical Cryptography

- People used their intuition to come up with very clever codes that seem to be secure
    - Shift cipher (Caesar's cipher): shift each letter by a predetermined amount
    - Substitution cipher: sample a permutation and apply the permutation to each letter
- … until someone more clever broke them
    - Shift cipher: Only 26 possible solutions. Try each one!
    - Substitution cipher: Statistical patterns in the English language narrows down likely permutations



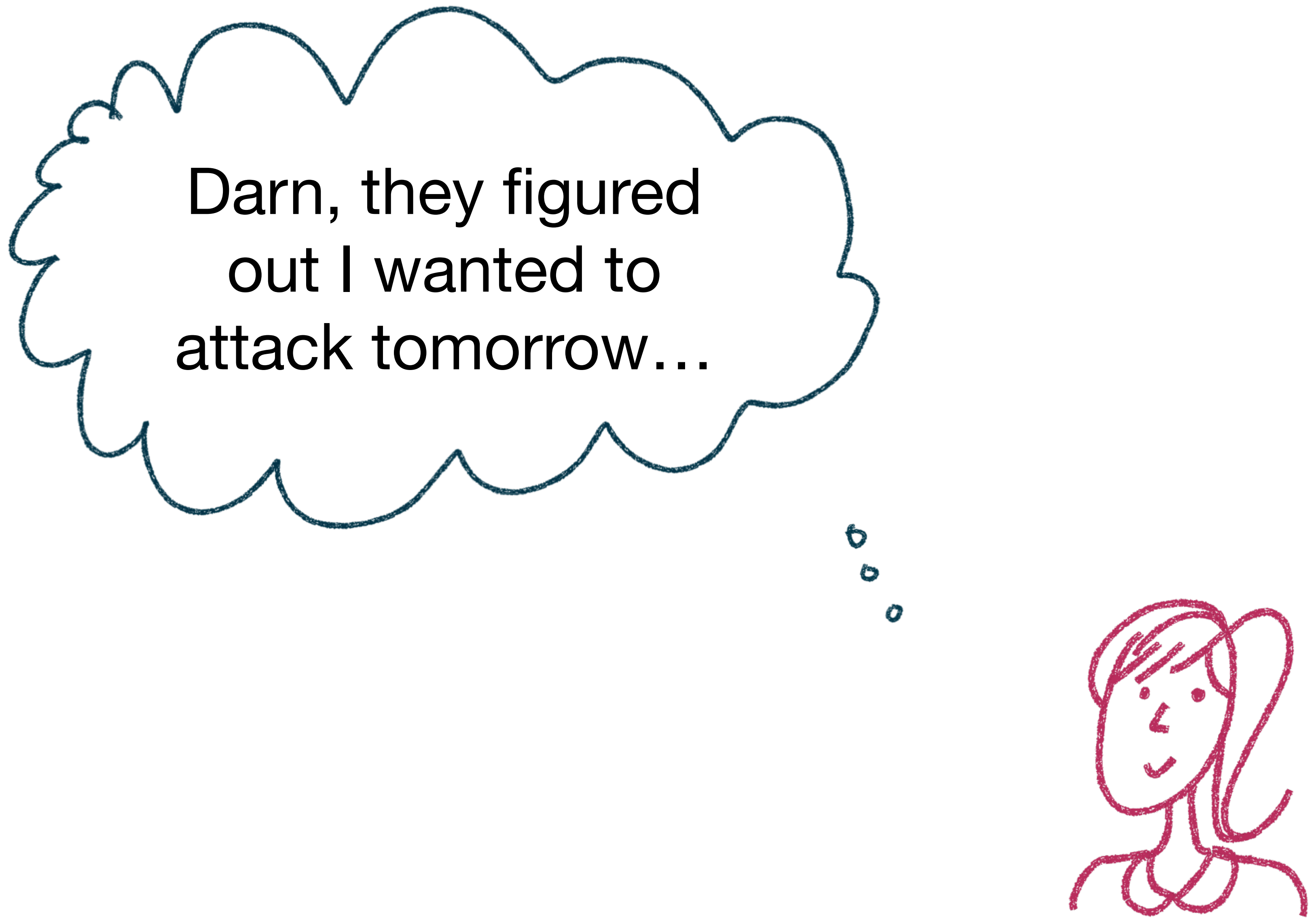
We'll shift letters in our messages over by 3!

dwwdfn wrpruurz

I think they're just shifting the letters!

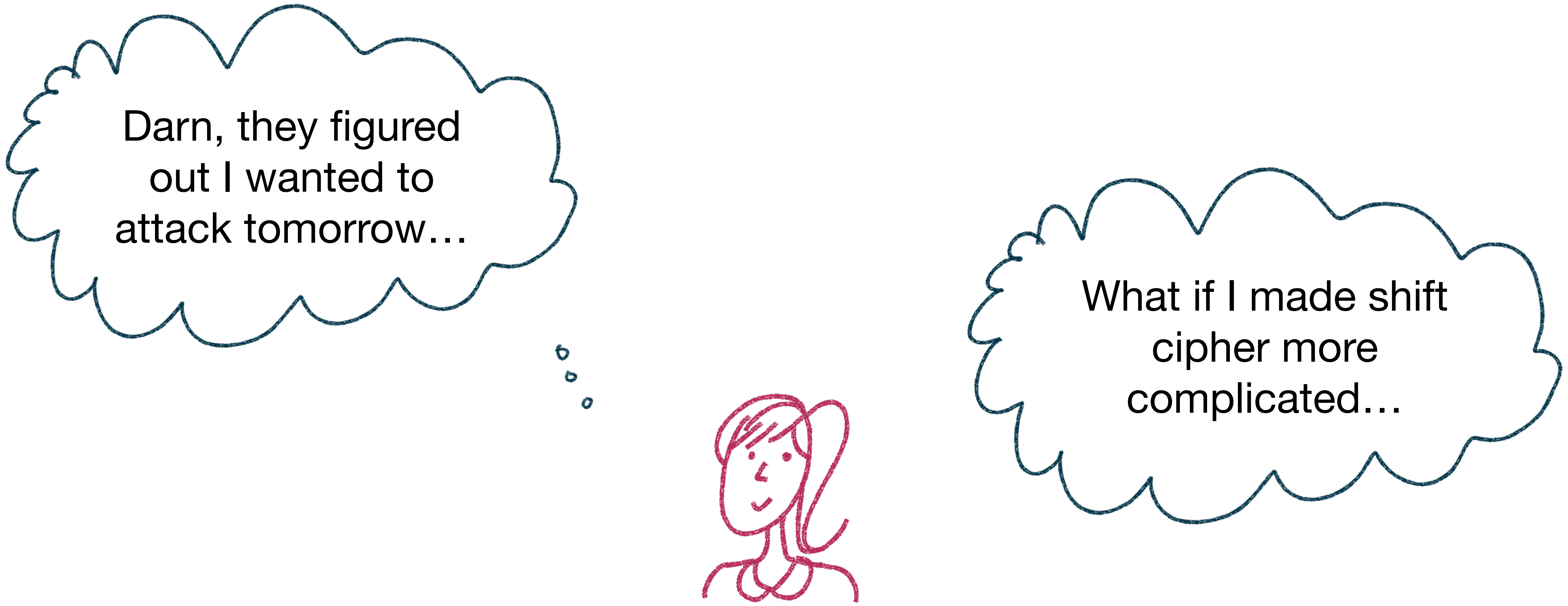
Historical Cryptography

- What constitutes a good code?
  - *The enemy general doesn’t found out when your army will attack*
- What does it mean if your code is broken?
  - *You weren’t clever enough…*



Historical Cryptography

- What constitutes a good code?
  - *The enemy general doesn’t found out when your army will attack*
- What does it mean if your code is broken?
  - *You weren’t clever enough… and now you need another code*


Darn, they figured out I wanted to attack tomorrow…
What if I made shift cipher more complicated…

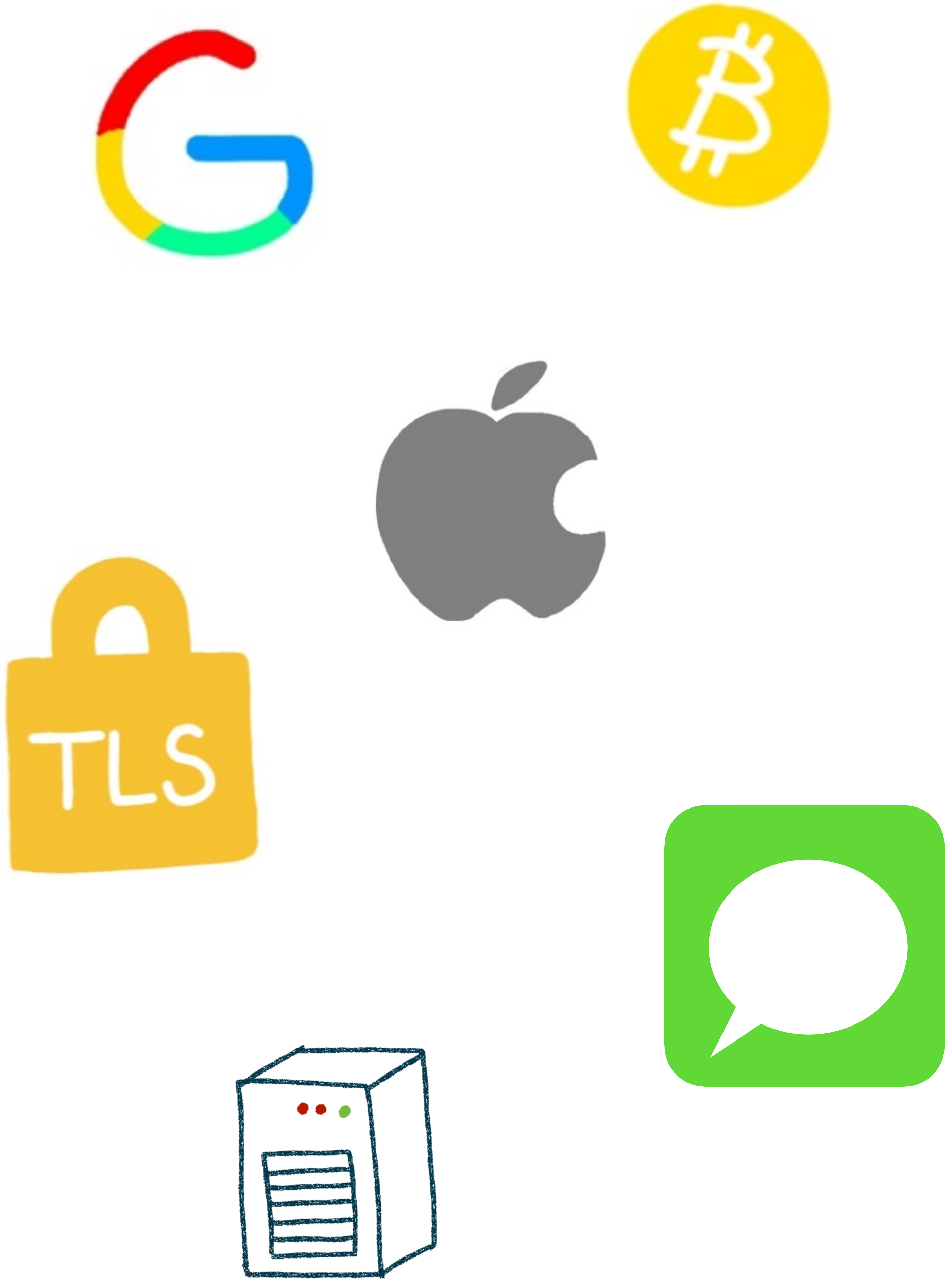
Historical Cryptography

- Cryptography and ciphers have a fascinating history
  - Interesting and creative ideas (which are almost all broken by now)
  - Influenced world history (e.g., cryptanalysis of the German Enigma in WWII)
- Much of cryptography up until the 20th century was a **heuristic process**
- What does it mean for something to be “secure”?
  - Is this notion of “secure” useful and realistic?
  - Can we convince ourselves and others that a scheme meets this notion?

By Alessandro Nassiri - Museo della Scienza e della Tecnologia "Leonardo da Vinci", CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=47910919

Modern Cryptography

- Late 20th century we started viewing cryptography as a science
    - Formalized notions of security
    - Rigorous threat models, firm foundations, and proofs!
- Much much more than secure communication
    - Authentication, anonymous communication + authentication, verifying computations efficiently, secure elections, consensus, computing on secret data without revealing it (*Homomorphic Encryption*), private computation with people you don't trust (*Multiparty Computation*), currency without a central authority (*Blockchain/E-Cash*), and more!
- Used everywhere by everyone (including you!)



Modern Cryptography

Analyzing the security of a cryptographic system consists of:

1. Formalizing a precise definition of security

   If you don’t understand what you want to achieve, how can you possible know when (or if) you have achieved it?

2. Stating the underlying assumptions

   Others will attempt to validate (or invalidate) your assumptions

3. Proving that the definition is satisfied given the assumptions

   Can schemes still get “broken”?

   Yes! If the definition does not capture real-world attacks or the assumptions turn out invalid

Goals of this Course

- Understand the theoretical basis for the real world crypto systems all around you (now, and in the near future)

- Understand computational assumptions and the implications in cryptography

- Recognize the limitations of what is possible to achieve

- Develop skills to prove security or assess validity of claimed security of cryptographic constructions according to established definitions

Prerequisites

- Required: COMS W3203 (Discrete math) or equivalent
  - You are expected to be comfortable understanding formal definitions and writing proofs
- Recommended: COMS W3261 (CS Theory)
  - Basic number theory and probability will be covered as needed

This class in its current form almost completely overlaps with COMS W4261.

It is not recommended you take this class if you've taken COMS W4261!

Materials

Recommended textbooks:

- “Introduction to Modern Cryptography” by Jonathan Katz and Yehuda Lindell

  (available via the Columbia Library)

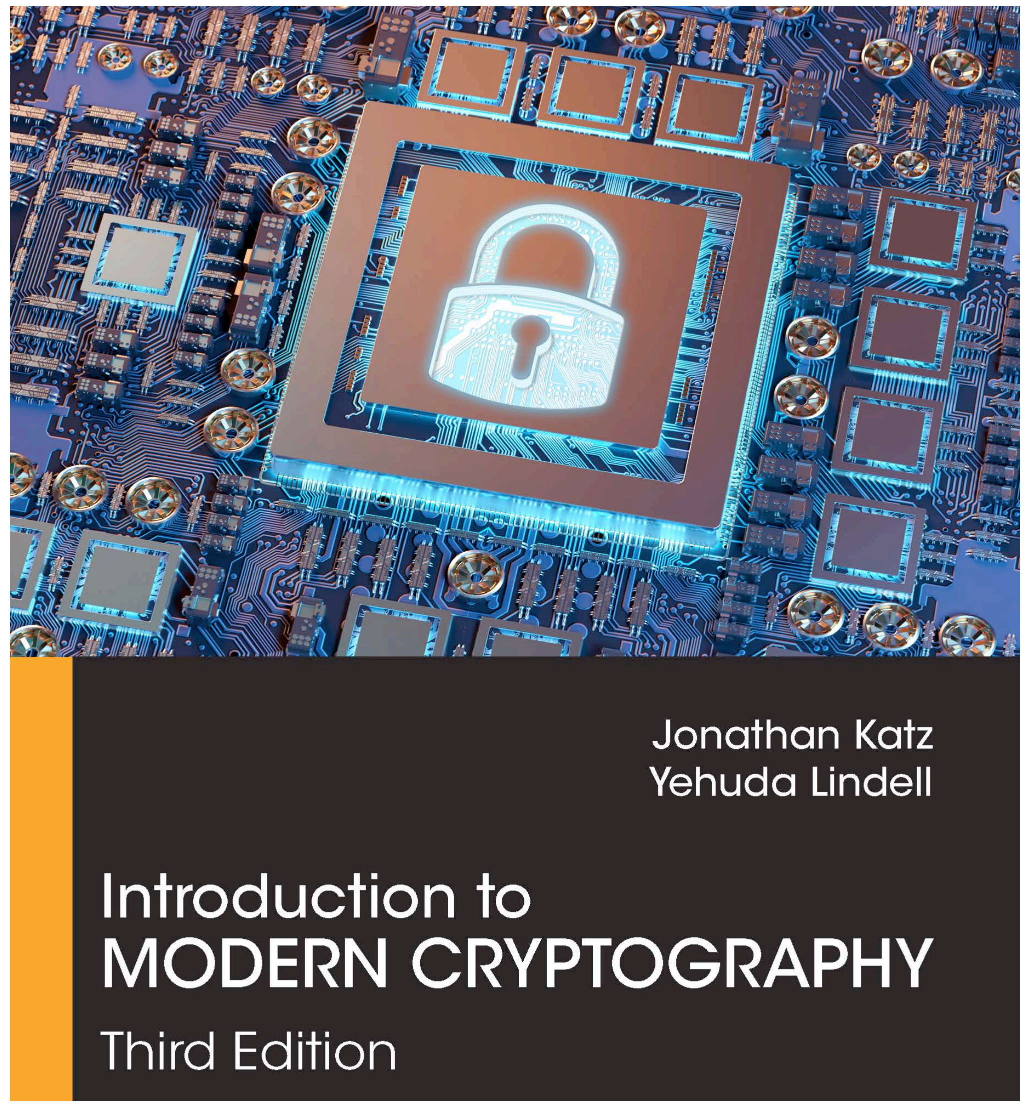
- “A Course in Cryptography” by Rafael Pass and abhi shelat

  (free online: https://www.cs.cornell.edu/courses/cs4830/2010fa/lecnotes.pdf)

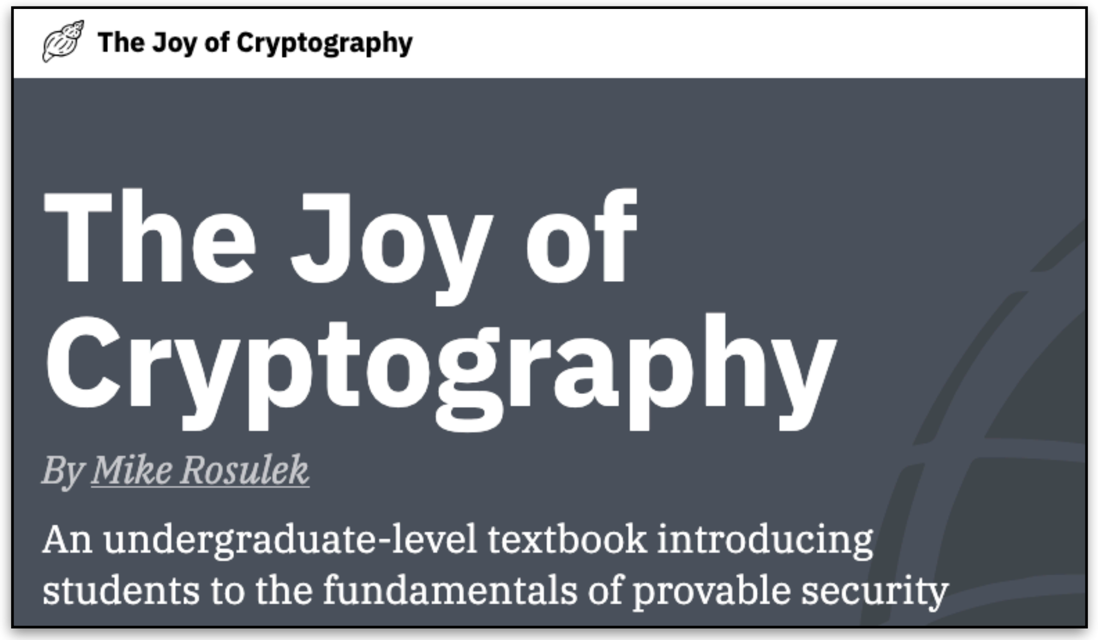
- “The Joy of Cryptography” by Mike Rosulek

  (typically free, but paywalled until July 2026… https://joyofcryptography.com/)

There will not be any assignments from the books but they may be useful to assist in understanding the material



A COURSE IN
CRYPTOGRAPHY



The Joy of Cryptography

The Joy of Cryptography

By Mike Rosulek

An undergraduate-level textbook introducing students to the fundamentals of provable security

Coursework

- **Problem Sets (40%):** 5-6 homeworks throughout the semester
  - Each one you'll get roughly 2 weeks to work on it
  - Can be solved collaboratively (you are encouraged to work in small groups!)
- **Midterm (25%):** In-class written exam Wednesday, March 11
  - No notes except a single letter-sized reference sheet
- **Final Exam (35%):** Written final exam during finals week (tentatively May 11)
  - No notes except a single letter-sized reference sheet

Contact CARDS if you need accommodations!

Problem Sets

- LaTeX is recommended, but anything legible is fine
  - If you handwrite and I can’t read it, you will not get credit!
- You’re allowed up to 3 late days per assignment at a 10% penalty per day
  - The lowest problem set grade will be dropped at the end of the semester
- The use of generative AI for anything related to assignments is prohibited
  - Please do not make me have to switch us to oral assignments/exams. Please.
- If you don’t know the answer, you can write “**I don’t know**” to receive 20% of the credit for that question*

*This does not apply to extra credit problems

Syllabus (tentative)

Part 1: Foundational Primitives

- One-way Functions (OWF)
- Pseudorandom Generators (PRGs)
- Pseudorandom Functions (PRFs)
- Hash Functions
- Symmetric Encryption

Part 2: Public Key Cryptography

- Public Key Encryption
- Digital Signatures
- Identification Schemes
- RSA and DSA

Any Extra Time: Advanced Cryptography

- Zero Knowledge Proofs
- Two-Party/Multiparty Computation
- Quantum Cryptography

We will not talk about:

- Details of specific protocols and standards currently being used
- Cryptoanalysis (historical or modern)
- “Mathematical” Crypto (e.g., Elliptic Curves, Class Groups, Isogenies, heavy number theory)
- Implementations (there are no programming assignments)
- Blockchains and cryptocurrencies
- Secure or private AI/ML
- Social and legal issues

These are all interesting but outside the scope of this class.

There are many other courses that may touch on these topics!

About Me

- I’m a new professor at Barnard :)
- My research is in cryptography
  - Multiparty computation, threshold signatures, and digital authentication
  - I’m considered “practical” since I focus on things that are efficient enough to be deployed today
- If this material is interesting to you, there are quite a lot of people at CU/BC working on cryptography!

That's me!

Eysa Lee
eylee@barnard.edu

About your TA

- 4th year undergrad at Columbia!
- Currently doing theory CS and crypto research with tons of people at CU
- Can help you with your homework or general questions about the material :)
    - Office hours times TBD

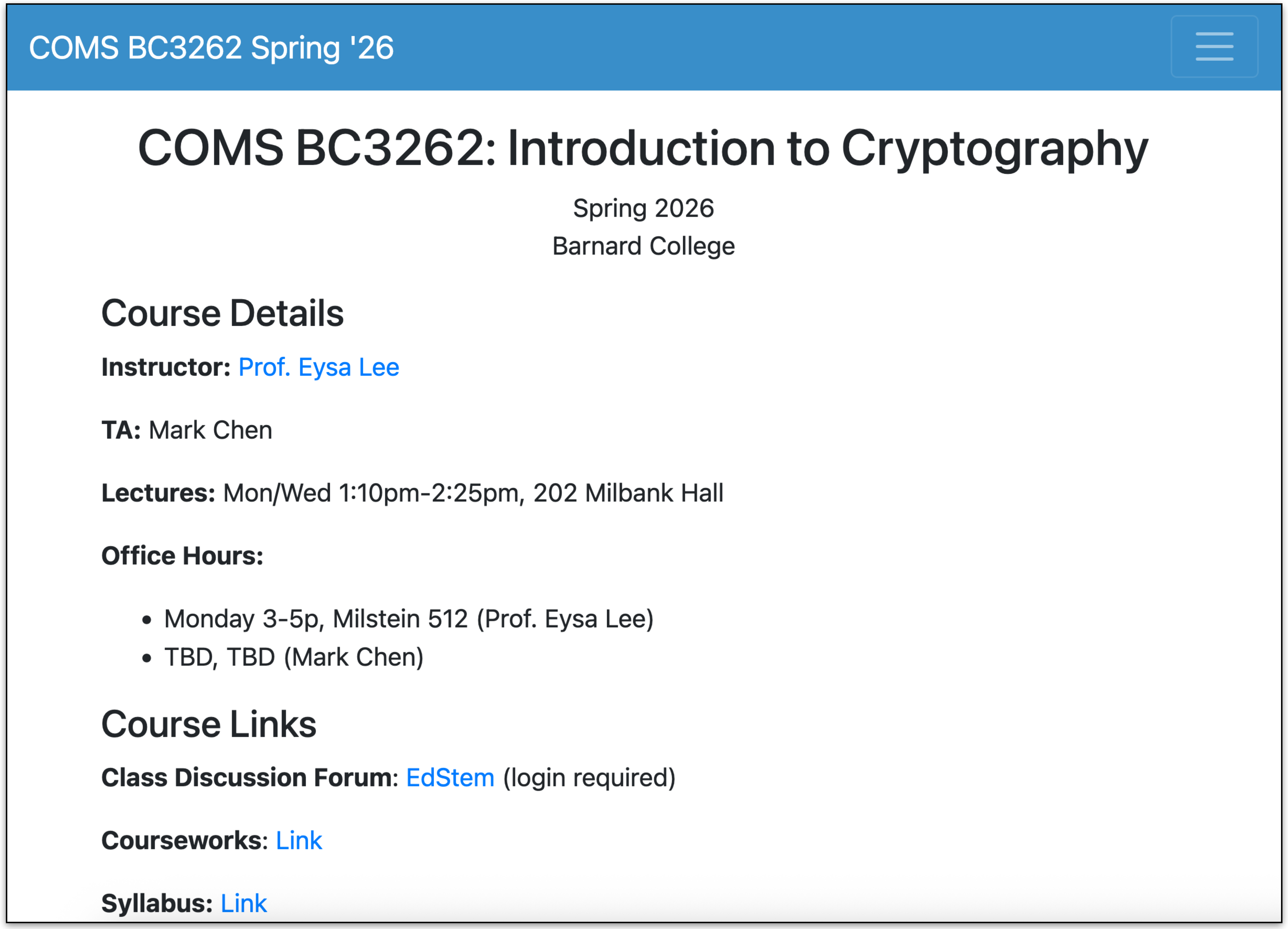
Side note: Don’t go to him for regrades.
He isn’t the one grading your assignments

Mark Chen
yc3879@columbia.edu

Questions?

More information can be found on the course website:

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



COMS BC3262 Spring '26

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

Office Hours:

- Monday 3-5p, Milstein 512 (Prof. Eysa Lee)
- TBD, TBD (Mark Chen)

Course Links

Class Discussion Forum: EdStem (login required)

Courseworks: Link

Syllabus: Link

Otherwise, we can get started!

Symmetric-Key Encryption

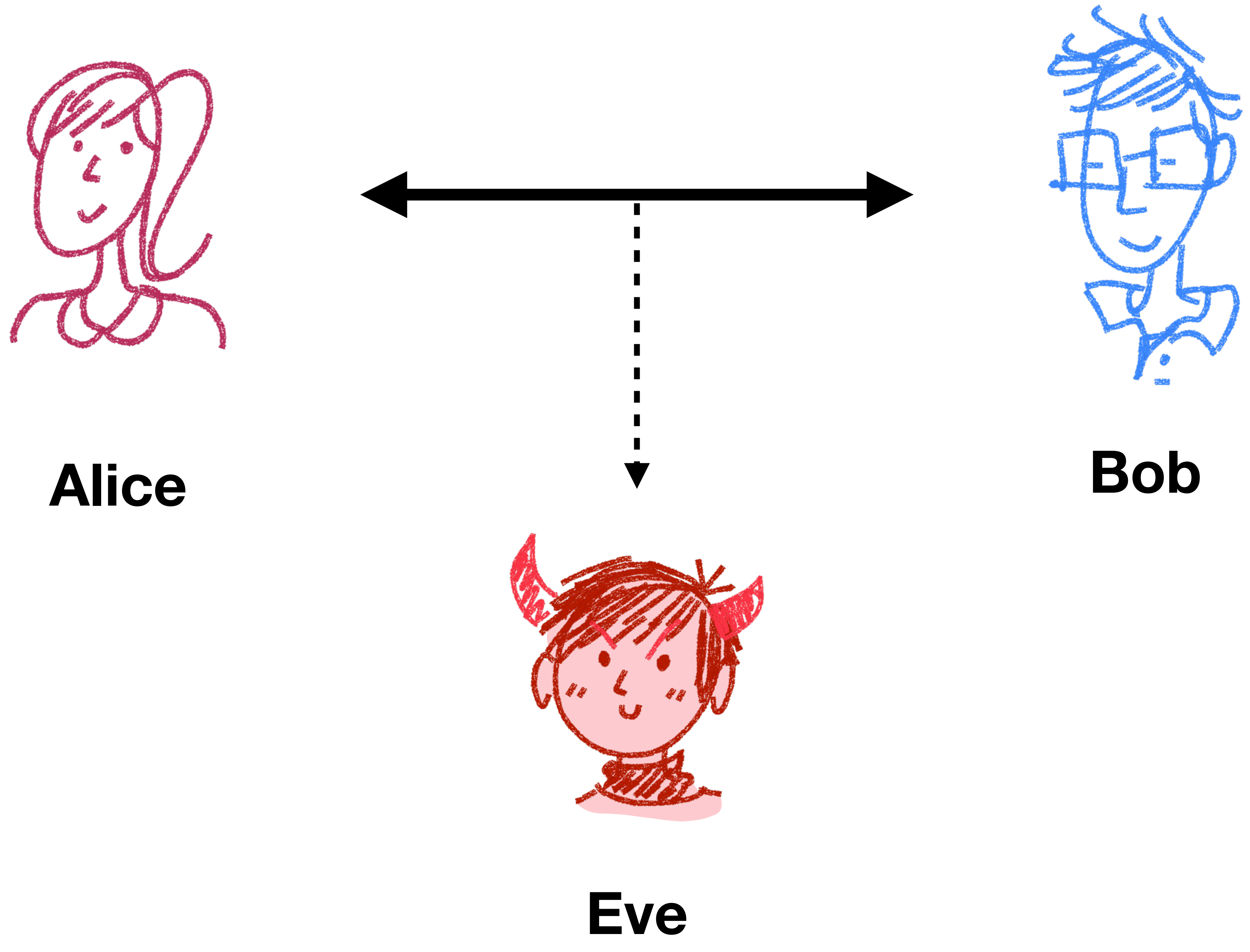
Secure Communication

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

- **A possible solution**: Before communicating, Alice and Bob agree on a “secret code”

  - The “secret code” will consist of a *key*, an algorithm for “scrambling” messages, and an algorithm for “unscrambling” messages

  - How do we formalize this?



How to formalize this secret code?

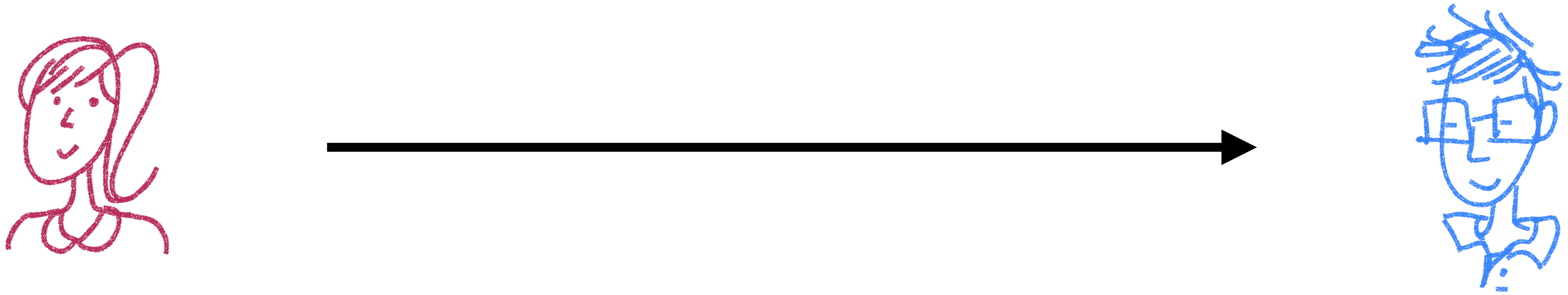
- What information should be *public* (known to everyone) vs *private* (kept secret)?
- Historically, everything (keys and algorithms) was kept private
    - Security by obscurity :)
    - What could go wrong?
- **Kerchoff's Principle (1884):** The only thing that should be assumed private is the key; everything else should be assumed to be public
    - Immediate consequence is that there must be randomness involved somewhere in the algorithms!

Symmetric Key Encryption

Syntax: Three algorithms (Gen, Enc, Dec)

- **Key generation** algorithm Gen outputs a key $k \in \mathscr{K}$
- **Encryption** algorithm Enc takes a key $k \in \mathscr{K}$ and a plaintext $m \in \mathscr{M}$ and outputs a ciphertext $c \in \mathscr{C}$
- **Decryption** algorithm Dec takes a key $k \in \mathscr{K}$ and a ciphertext $c \in \mathscr{C}$ and outputs a plaintext $m \in \mathscr{M}$

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



Symmetric Key Encryption

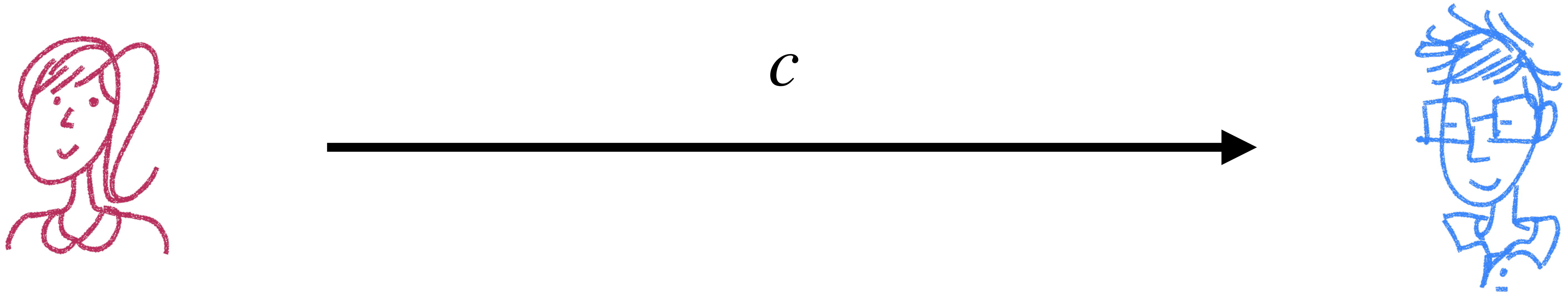
Syntax: Three algorithms (Gen, Enc, Dec)

- **Key generation** algorithm Gen outputs a key $k \in \mathscr{K}$
- **Encryption** algorithm Enc takes a key $k \in \mathscr{K}$ and a plaintext $m \in \mathscr{M}$ and outputs a ciphertext $c \in \mathscr{C}$
- **Decryption** algorithm Dec takes a key $k \in \mathscr{K}$ and a ciphertext $c \in \mathscr{C}$ and outputs a plaintext $m \in \mathscr{M}$

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

$k \leftarrow$ Gen()
$c \leftarrow$ Enc(k, m)
$m =$ Dec(k, c)

m
$c \leftarrow$ Enc(k, m)



c

$m =$ Dec(k, c)

Symmetric Key Encryption

Syntax: Three algorithms (Gen, Enc, Dec)

- **Key generation** algorithm Gen outputs a key $k \in \mathscr{K}$
- **Encryption** algorithm Enc takes a key $k \in \mathscr{K}$ and a plaintext $m \in \mathscr{M}$ and outputs a ciphertext $c \in \mathscr{C}$
- **Decryption** algorithm Dec takes a key $k \in \mathscr{K}$ and a ciphertext $c \in \mathscr{C}$ and outputs a plaintext $m \in \mathscr{M}$

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

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

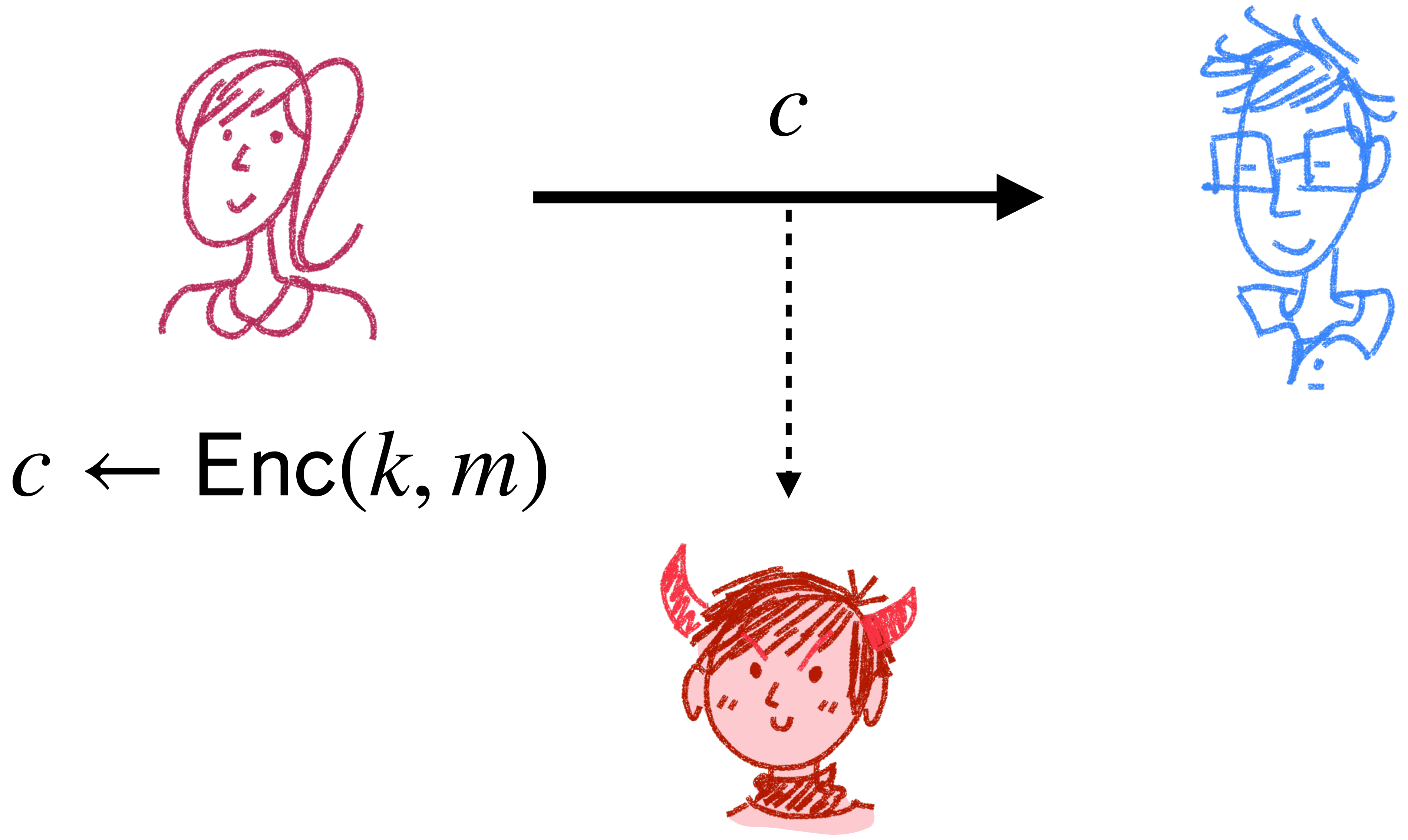
Correctness: $\forall k \in \mathscr{K}, m \in \mathscr{M}$
$\mathsf{Dec}(k, \mathsf{Enc}(k, m)) = m$

Security: Hm…

How to define 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



c

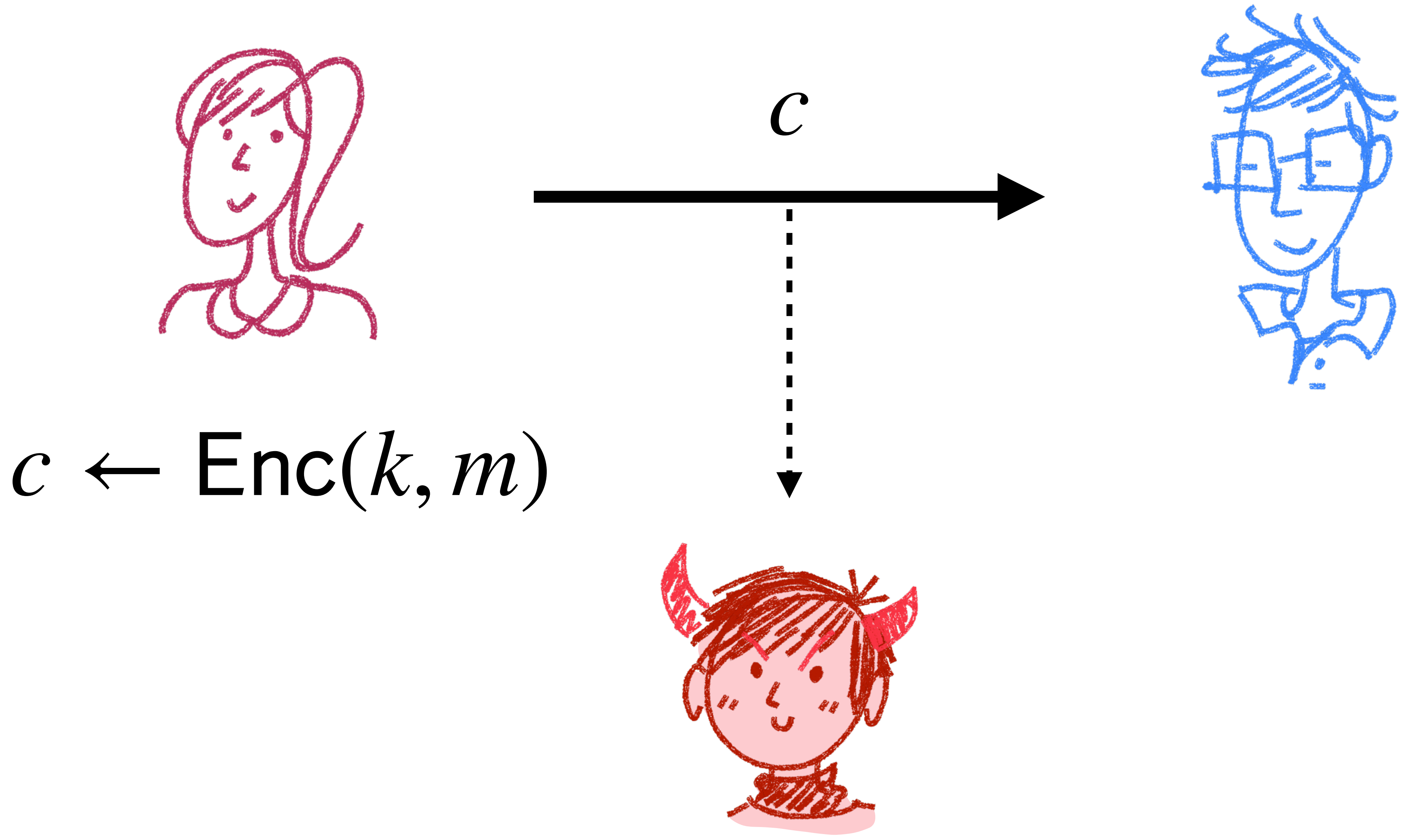
$c \leftarrow \text{Enc}(k, m)$

How to define security?

Consider the following experiment:

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

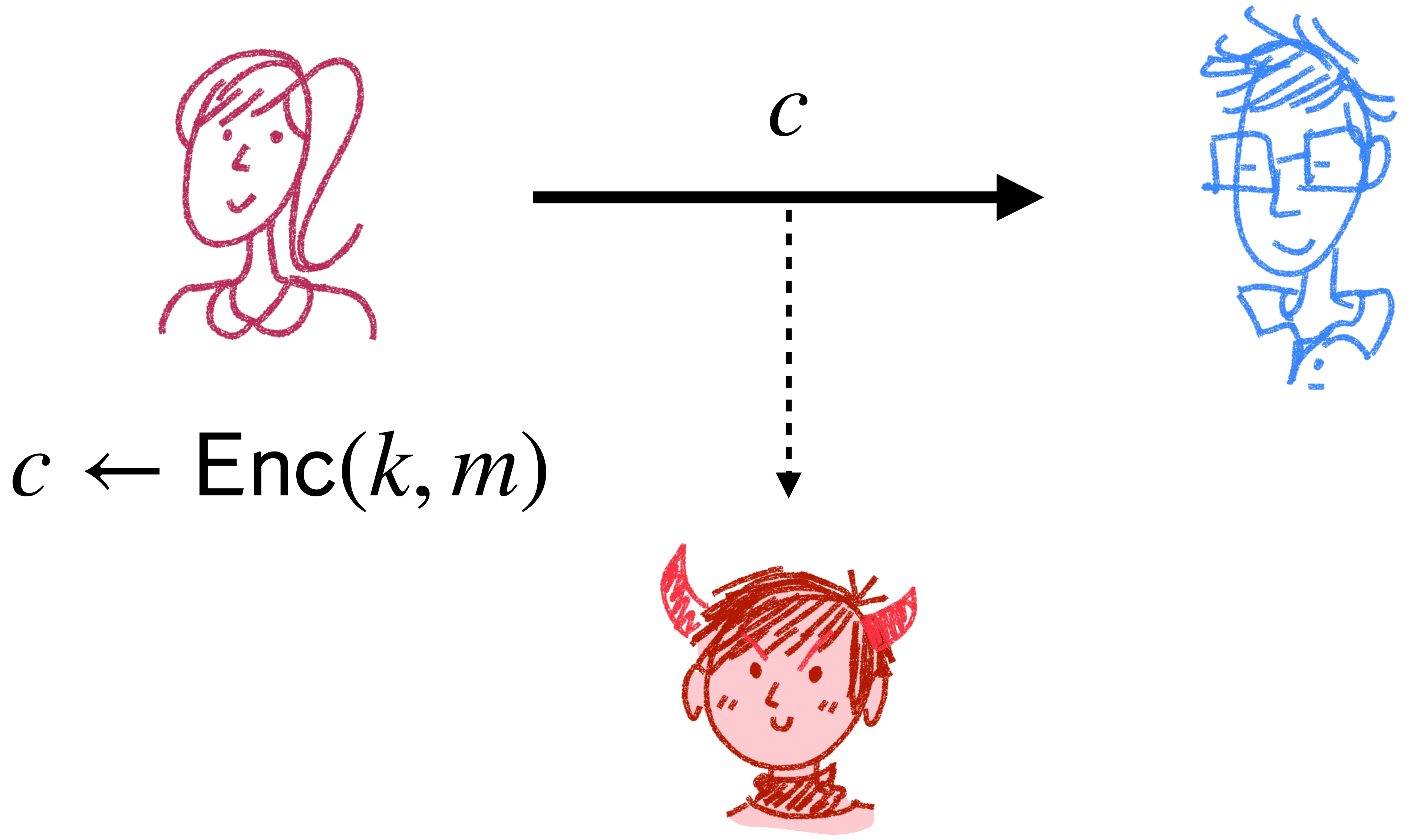
Idea 1: Adversary should not be able to learn k



How to define security?

Consider the following experiment:

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



c

$c \leftarrow$ Enc(k, m)

Idea 1: Adversary should not be able to learn k

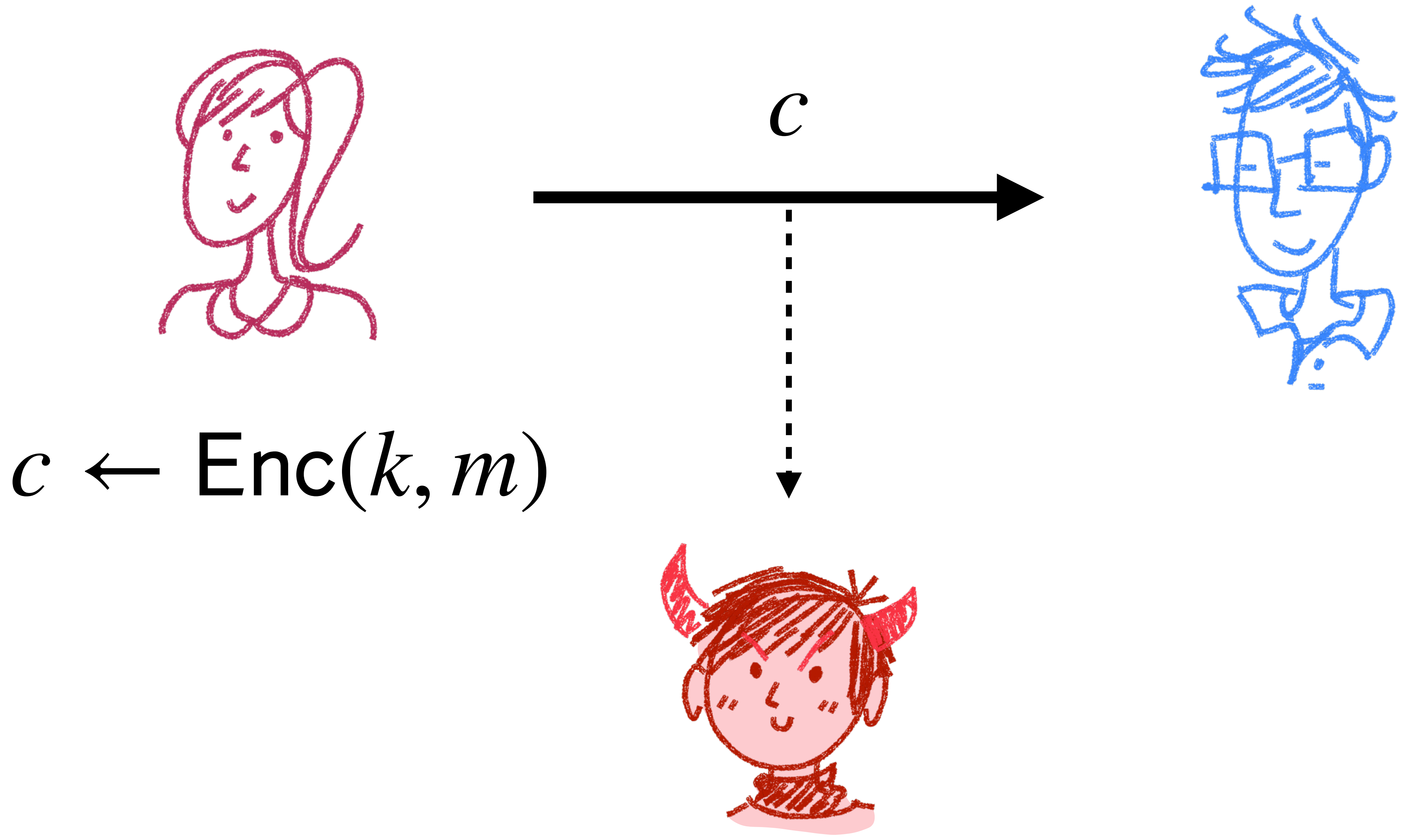
Problem: The identity function satisfies this requirement Enc$(k, m) = m$

How to define security?

Consider the following experiment:

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

Idea 2: Adversary should not be able to learn m



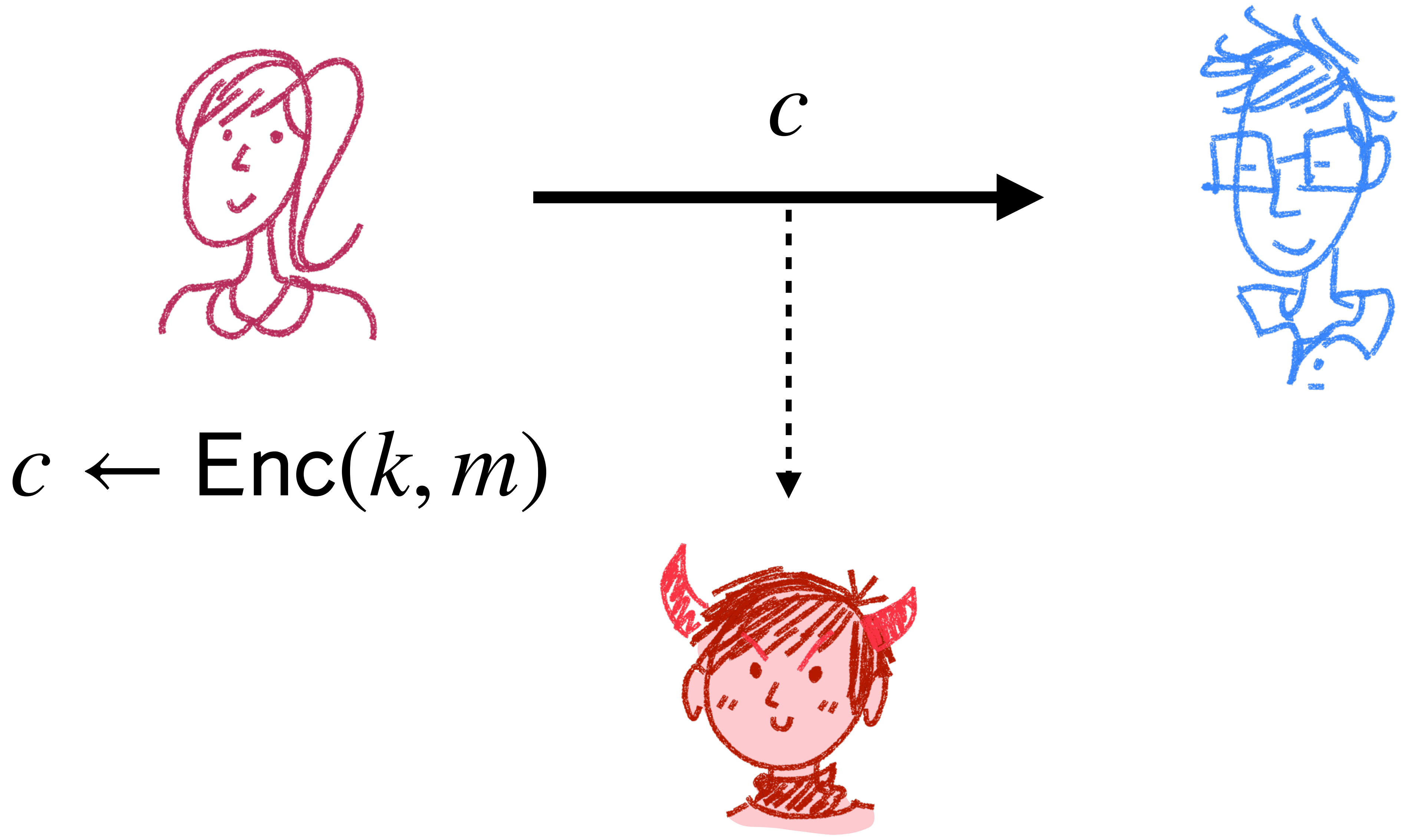
c

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

How to define 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



c

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

Idea 2: Adversary should not be able to learn m

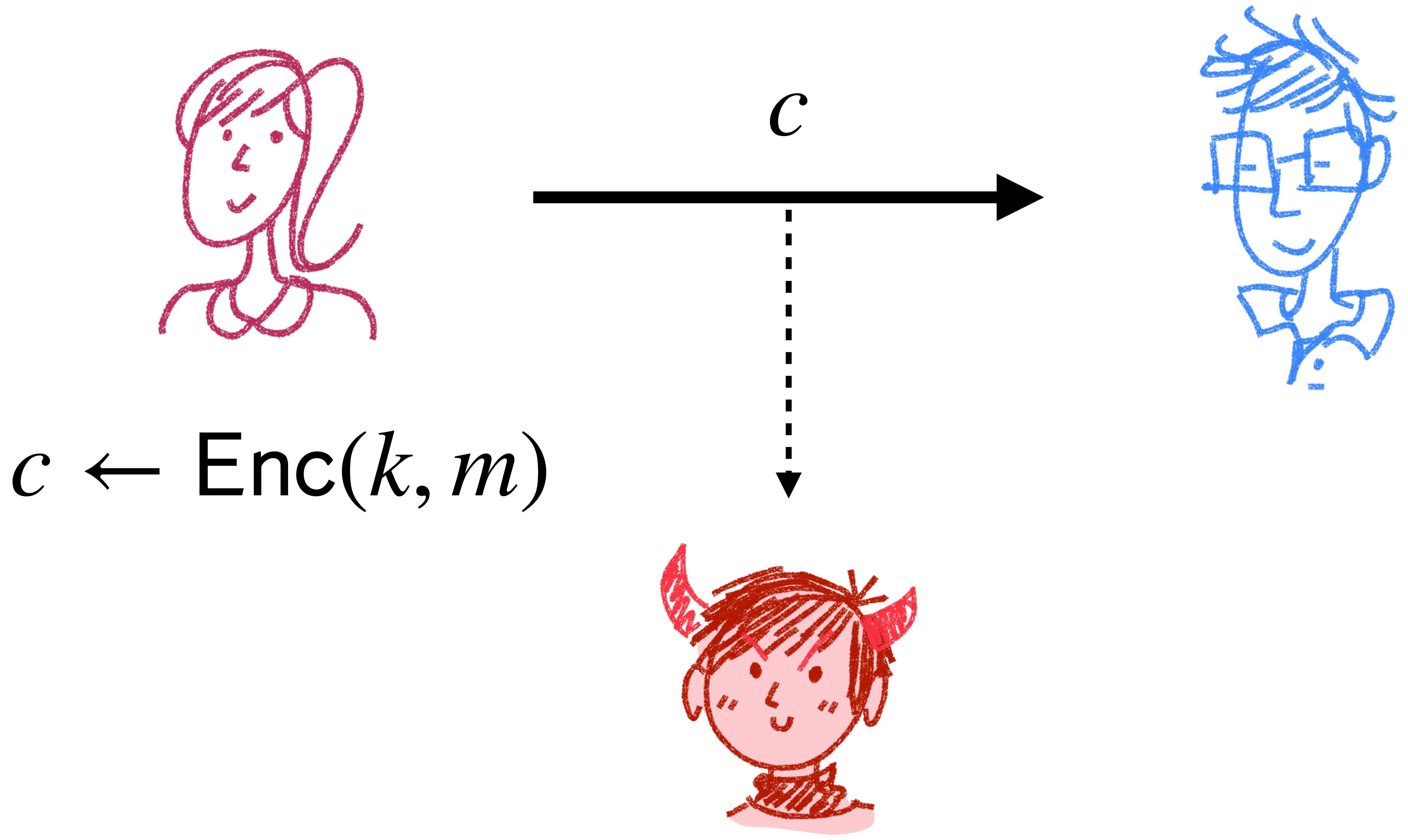
Problem: What if the adversary learns part of m? Say, the first half?

How to define security?

Consider the following experiment:

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

Idea 3: Adversary should not be able to learn any information about m



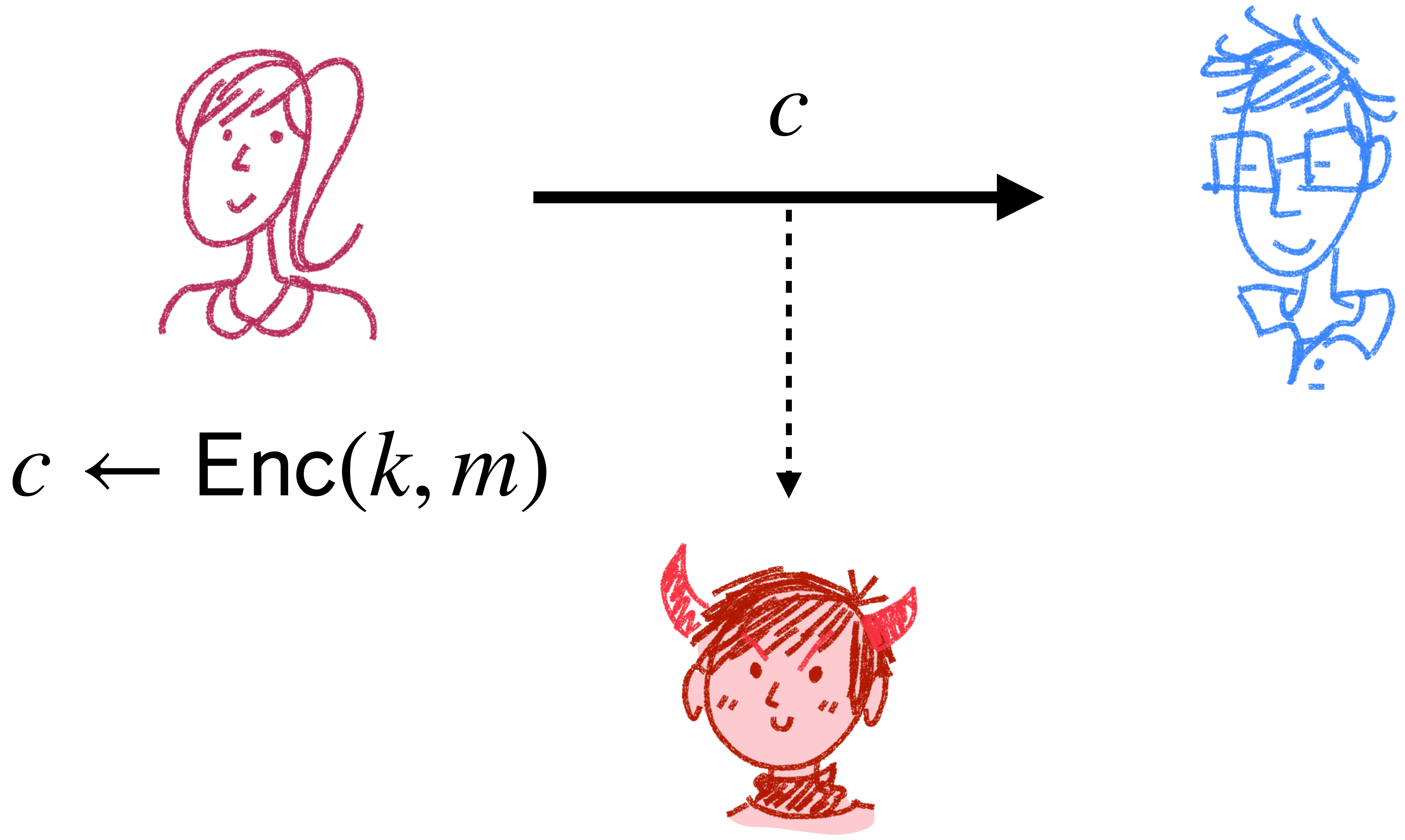
How to define security?

Consider the following experiment:

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

Idea 3: Adversary should not be able to learn any information about m

Problem: What if the adversary has a priori information about messages?



c

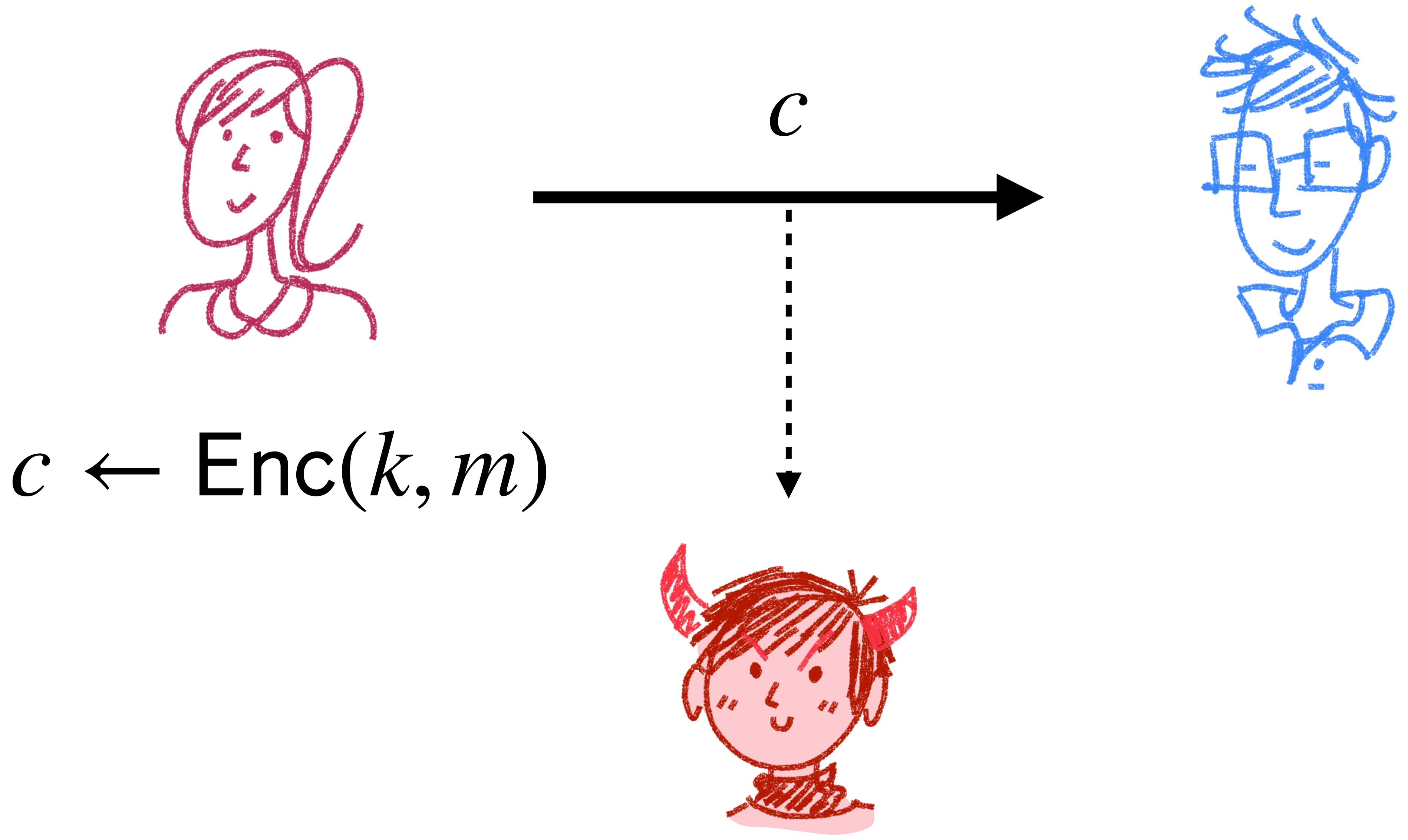
$c \leftarrow \mathsf{Enc}(k, m)$

How to define security?

Consider the following experiment:

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

Idea 4: Adversary should not be able to learn any *additional* information about m



c

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

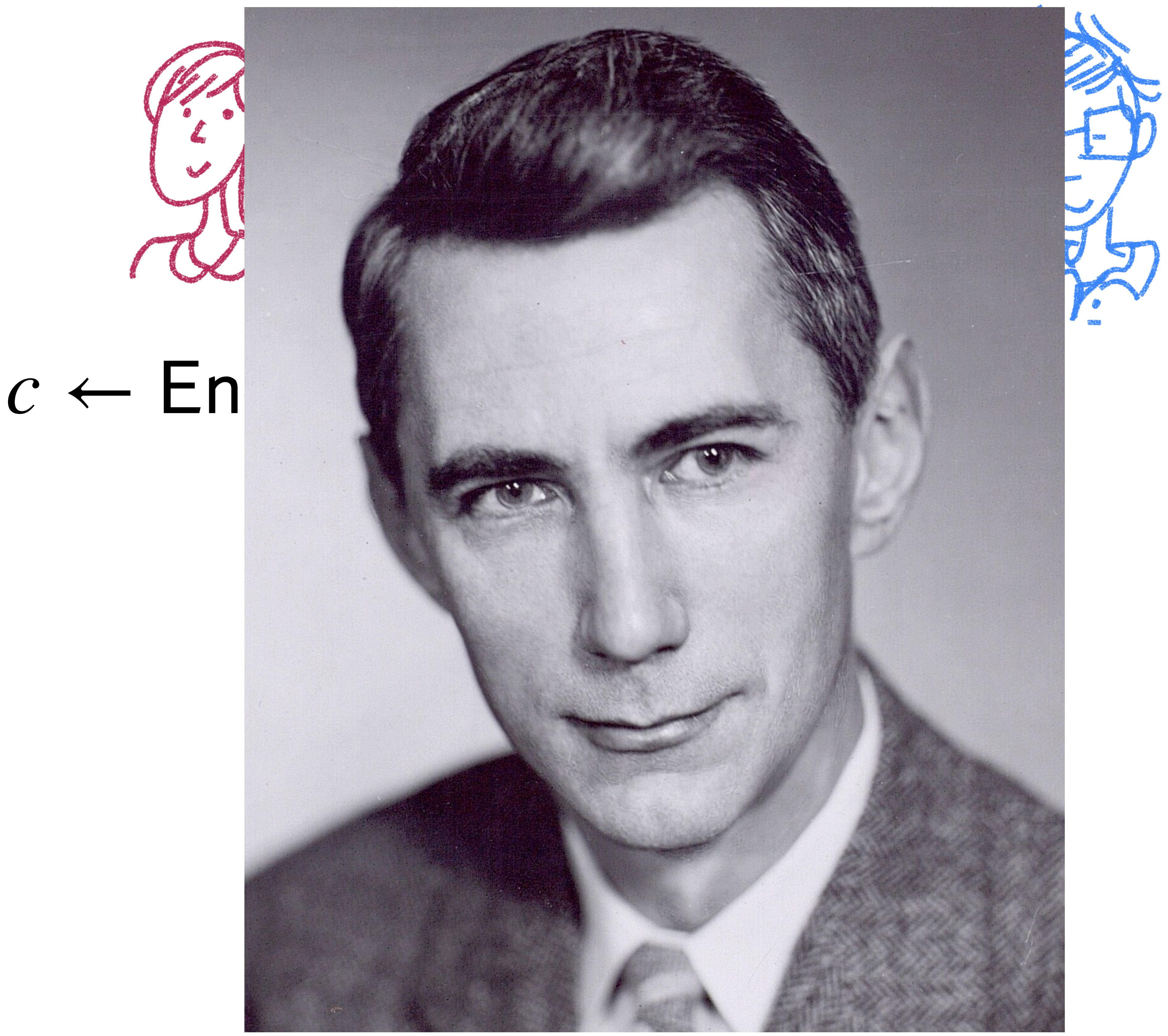
How to define security?

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

Idea 4: Adversary should not be able to learn any *additional* information about m

This is known as **Shannon secrecy**!



By Unknown author - Tekniska Museet of Sweden, Item 43069, via Flickr, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=158532837

Basic Probability

- A **probability space** Ω is a finite (or countable) set and a function $\Pr : \Omega \rightarrow [0,1]$ such that $\sum_{x \in \Omega} \Pr[x] = 1$

- An **event** is a subset of the probability space. The probability of an event $E \subseteq \Omega$ is defined as $\Pr[E] = \sum_{x \in E} \Pr[x]$

- Example: tossing a fair die $\Omega = \{1,2,3,4,5,6\}$ with the function $\Pr[x] = 1/6$

    - The probability of the event $E = \{2,4,6\}$ is $\Pr[E] = 1/2$

    - If it's an unfair die with $\Pr[1] = 1/2$ and all other values $\Pr[x] = 1/10$, then $\Pr[E] = 3/10$

Basic Probability

A **random variable** is a function on the probability space $X : \Omega \to \mathbb{R}$

Induce a distribution $\Pr[X = x] = \sum_{\{u \in \Omega : X(u) = x\}} \Pr[u]$

- Example: Define random variable X as the result of a fair dice
  - $\Pr[X = 3] = 1/6$
  - $\Pr[X < 3] = 1/3$

Basic Probability

A **random variable** is a function on the probability space $X : \Omega \to \mathbb{R}$

Induce a distribution $\Pr[X = x] = \sum_{\{u \in \Omega : X(u) = x\}} \Pr[u]$

- Example: What about a random variable Y that's 0 if the result is even and 1 if it is odd?
  - $\Pr[Y = 3] = 0$
  - $\Pr[Y < 3] = 1$
  - $\Pr[Y = 0] = \Pr[Y = 1] = 1/2$

Basic Probability

Given events A and B we can define

- Union: $A \cup B = \{x \,|\, x \in A \vee x \in B\}$
- Intersection: $A \cap B = \{x \,|\, x \in A \wedge x \in B\}$

When $\Pr[B] > 0$ we can define the **conditional probability** of A given B as

$$\Pr[A \,|\, B] = \frac{\Pr[A \cap B]}{\Pr[B]}$$

Basic Probability

Random variables X and Y are **independent** if for all x, y

$$\Pr[X = x \wedge Y = y] = \Pr[X = x] \cdot \Pr[Y = y]$$

Example: suppose we have the uniform distribution over $\Omega = \{0,1\}^2$ where

$$\Pr[00] = \Pr[01] = \Pr[10] = \Pr[11] = 1/4$$

Define random variables: X is the first bit, Y is the second bit, $Z = X \vee Y$, $W = X \oplus Y$

- Are X, Y independent?

- Are X, Z independent?

- Are X, W independent?

How to define security?

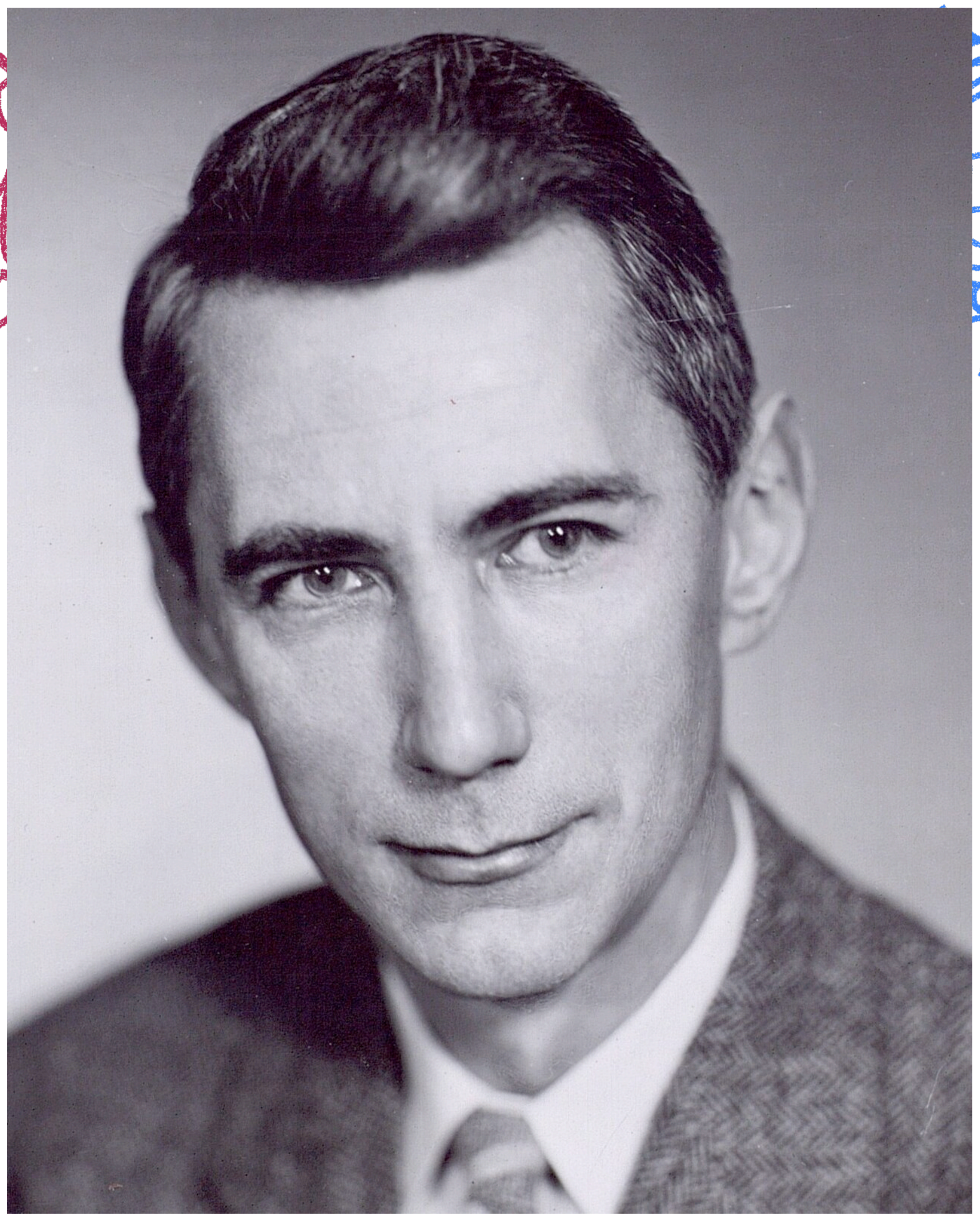
Consider the following experiment:

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

Idea 4: Adversary should not be able to learn any *additional* information about m

This is known as **Shannon secrecy**!

$c \leftarrow$ En



By Unknown author - Tekniska Museet of Sweden, Item 43069, via Flickr, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=158532837

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 knows 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

Perfect Secrecy

Shift cipher:

- $\mathscr{M} = \{a, \ldots, z\}^\ell$ and $\mathscr{C} = \{a, \ldots, z\}^\ell$
- Gen uniformly samples $k \leftarrow \{0,\ldots,25\}$
- Enc shifts each letter k positions forward (wrapping around from z to a)
- Dec shifts backwards k positions

Claim: The shift cipher **is not perfect secure** for plaintexts of length $\ell > 1$

Perfect Secrecy

Proof: To prove the cipher is **not** perfectly secret, we will explicitly define a distribution M, plaintext m, ciphertext c (with $\Pr[C = c] > 0$) such that

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

- Consider M defined by $\Pr[M = ''\text{aa}''] = \Pr[M = ''\text{ab}''] = 1/2$ and the ciphertext $c = ''\text{ab}''$
- Notice that $\Pr[M = ''\text{aa}'' \,|\, C = ''\text{ab}''] = 0$ since a letter in the plaintext must be mapped to the same letter in the ciphertext
- However, by construction $\Pr[M = ''\text{aa}''] = 1/2$
- Therefore we have $\Pr[M = ''\text{aa}'' \,|\, C = ''\text{ab}''] \neq \Pr[M = ''\text{aa}'']$

Another Definition of Perfect Secrecy

For any pair of messages $m_0, m_1 \in \mathscr{M}$, 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 $m_0, m_1 \in \mathscr{M}$ and for every $c \in \mathscr{C}$ it holds that

$$\Pr[\textsf{Enc}(k, m_0) = c] = \Pr[\textsf{Enc}(k, m_1) = c]$$

where $k \leftarrow \textsf{Gen}()$

Theorem: These two definitions of perfect secrecy are equivalent

Perfectly Secure Encryption for $\ell > 1$?

- We’ve seen that Shift Cipher is not perfectly secure
    - Substitution Cipher runs into a similar issue
- Are there any perfectly secret encryptions?
- Notice that shift and substitution ciphers work for messages of length 1
    - What if we adapted these ideas to operate on each element of a message independently?
    - This is the intuition behind one-time pad!

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

- The XOR of 2 strings in $\{0,1\}^{\ell}$ is defined bit-wise

Note that:
- $a \oplus a = 0$
- $a \oplus 0 = a$

One-Time Pad

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

- Gen uniformly samples $k \leftarrow \{0,1\}^\ell$

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

- Enc$(k, m) = m \oplus k$
- Dec$(k, c) = c \oplus k$

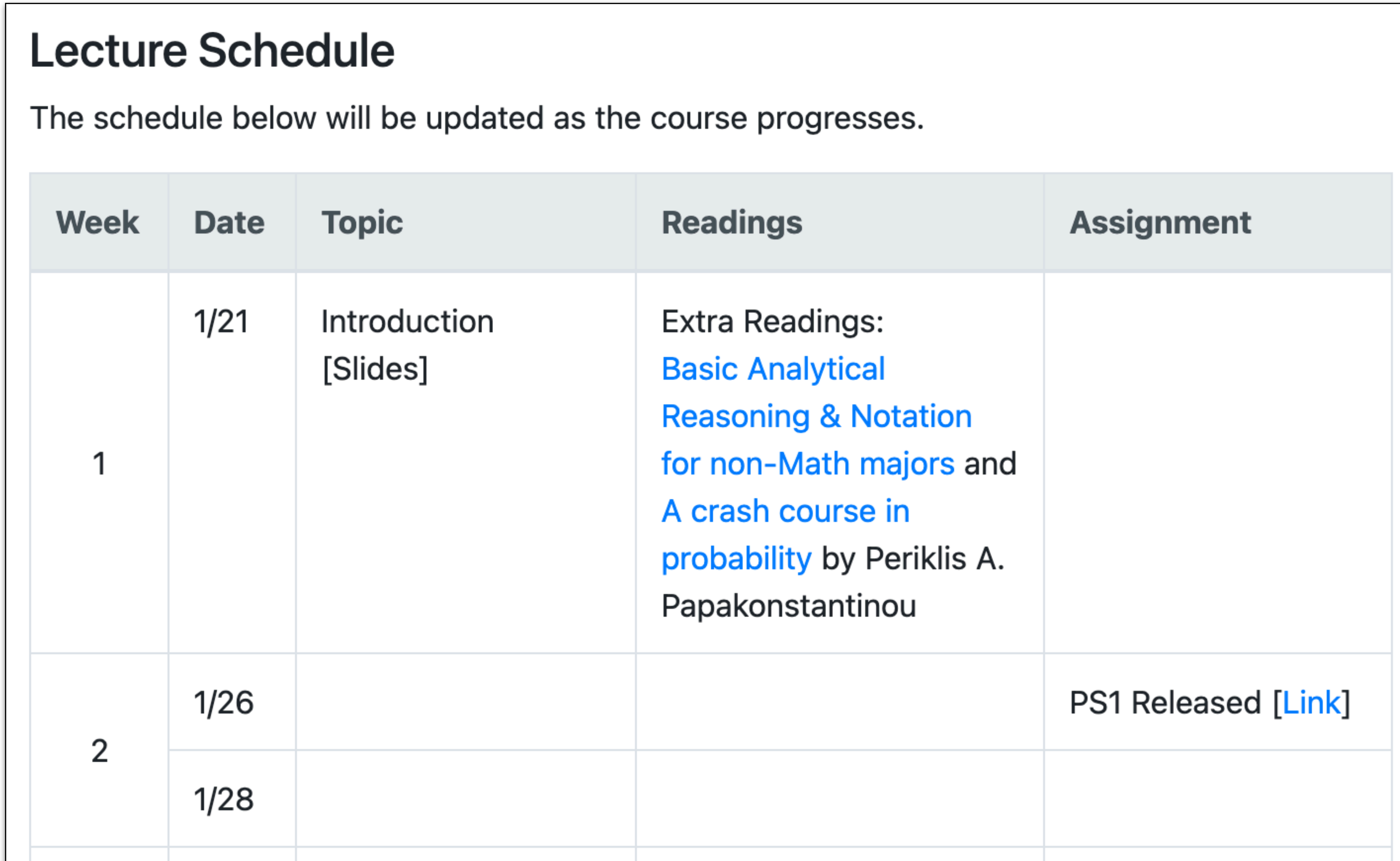
Correctness: $\forall k \in \mathscr{K}, m \in \mathscr{M}$

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

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

Next Time

- Limitations of one time pad
- Computational Security



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			

Acknowledgements: Many slides were inspired by when I TAed for Ran Cohen :)