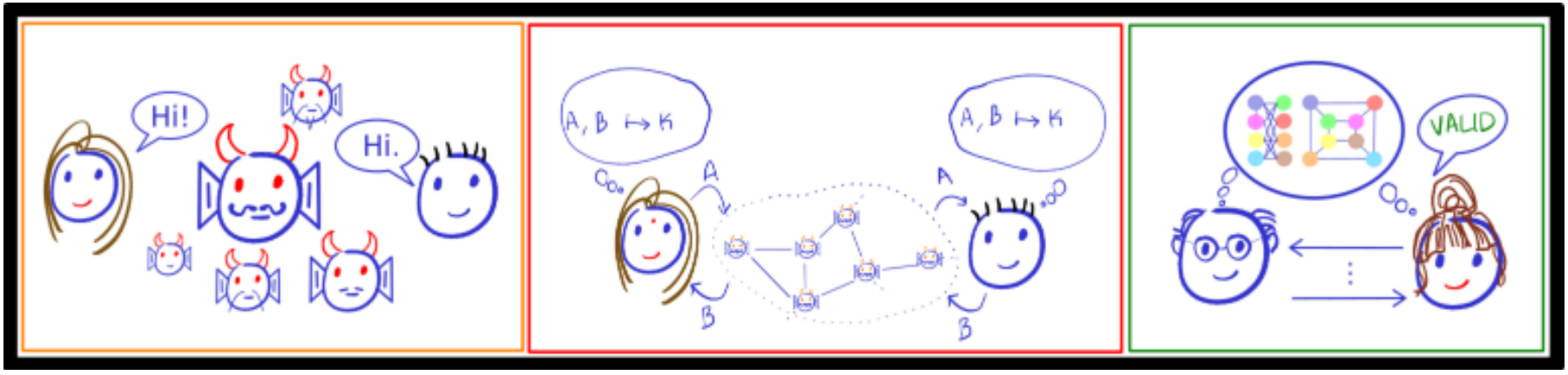


CS409m: Introduction to Cryptography

Lecture 01 (30/Jul/25)

Instructor: Chethan Kamath

Administrivia...

- When and where: Slot 5 in CC101
- Contact hours: after lectures, or appointment by e-mail
- Teaching assistants:
  - Nilabha Saha (210260037) and Priyanshu Singh (24M2101)

- Weekly TA help session:
  - Poll: 19:00-20:30 on Tuesdays *or* Fridays?
- Any volunteer for class rep.?

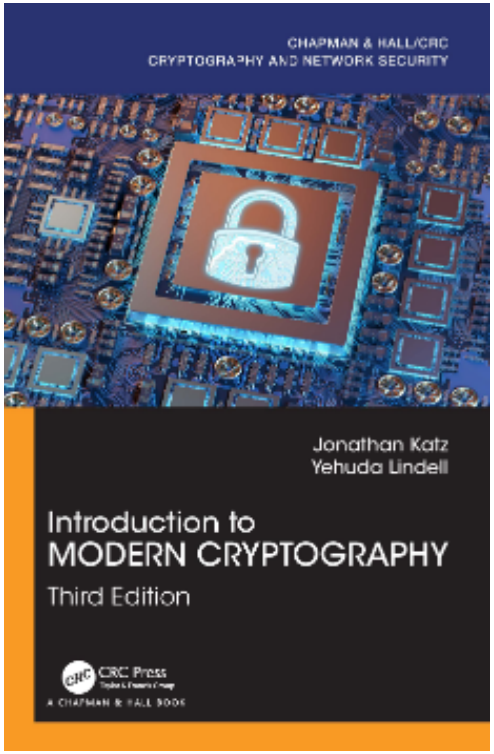
Administrivia...

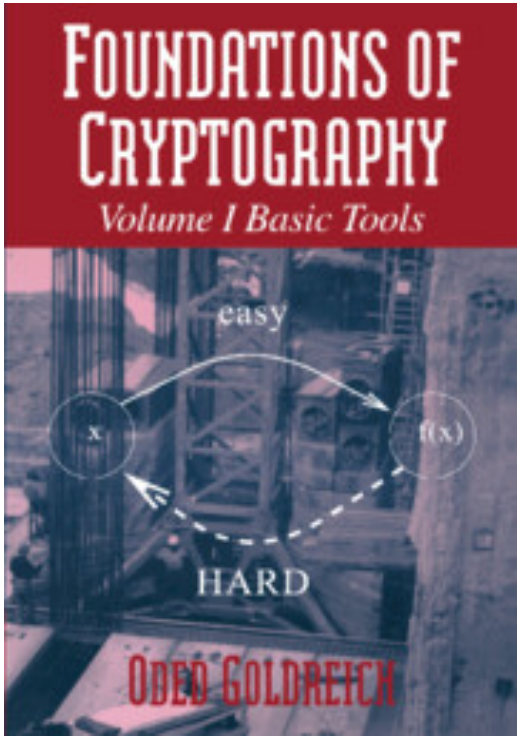
- Grading Scheme
  - Six ungraded assignments to help with quizzes and exams

Weightage	Towards
35%	End-sem
30%	Mid-sem
20%	Two (out of three) quizzes
10%	Hands-on exercises
5%	Class participation, pop-quizzes

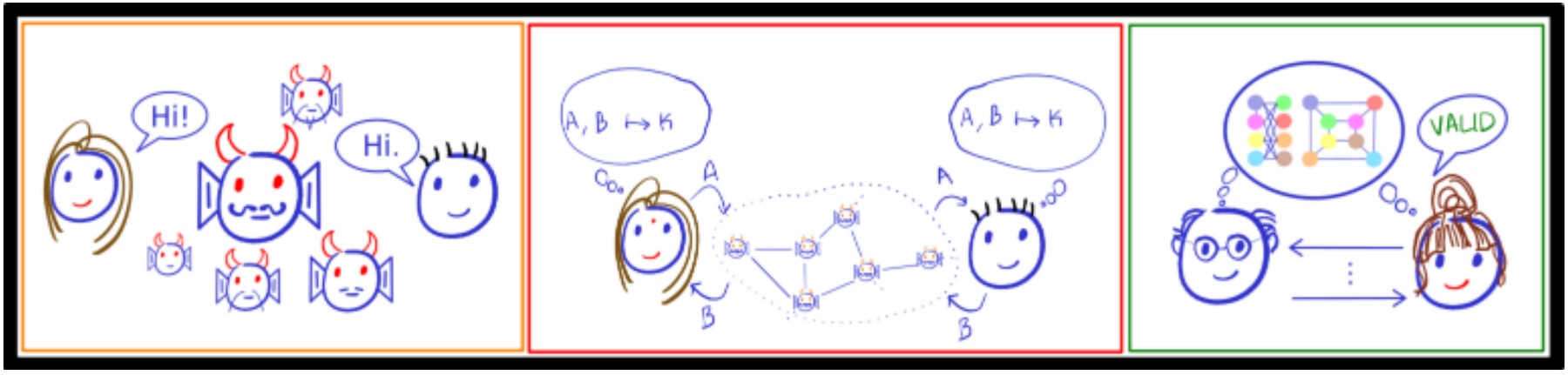
- Attendance is not mandatory (but encouraged)
- Important dates on course website (soon)
  - Hands-on Exercise 0: today!
  - Assignment 1: 01/Aug
  - Quiz 1: 22/Aug

Administrivia...





- Resources
  - Slides and other resources will be posted on course website
    - cse.iitb.ac.in/~ckamath/courses/2025a/CS409m.html
  - Announcements/online discussion on Moodle:
    - moodle.iitb.ac.in/course/view.php?id=7460



CS409m: Introduction to Cryptography

Lecture 01 (30/Jul/25)

Instructor: Chethan Kamath

You Use Cryptography all the Time!

Secure communication

Online transactions

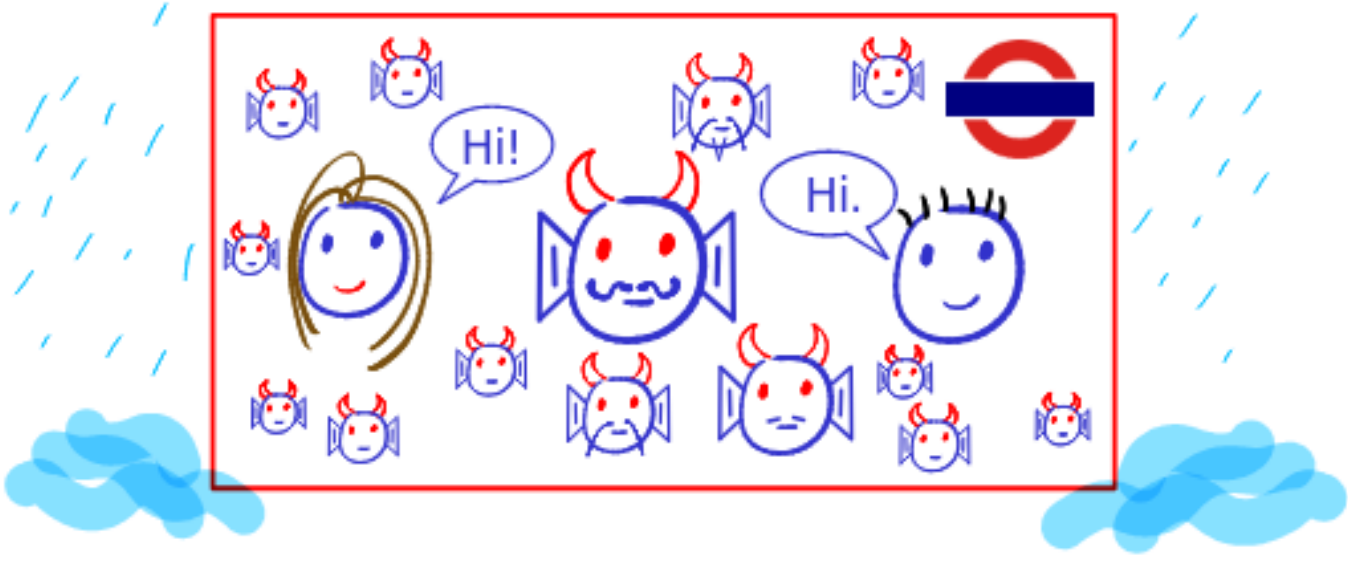
Using laptop/phone

Using internet

Credit for images: Wikipedia

What is Cryptography?

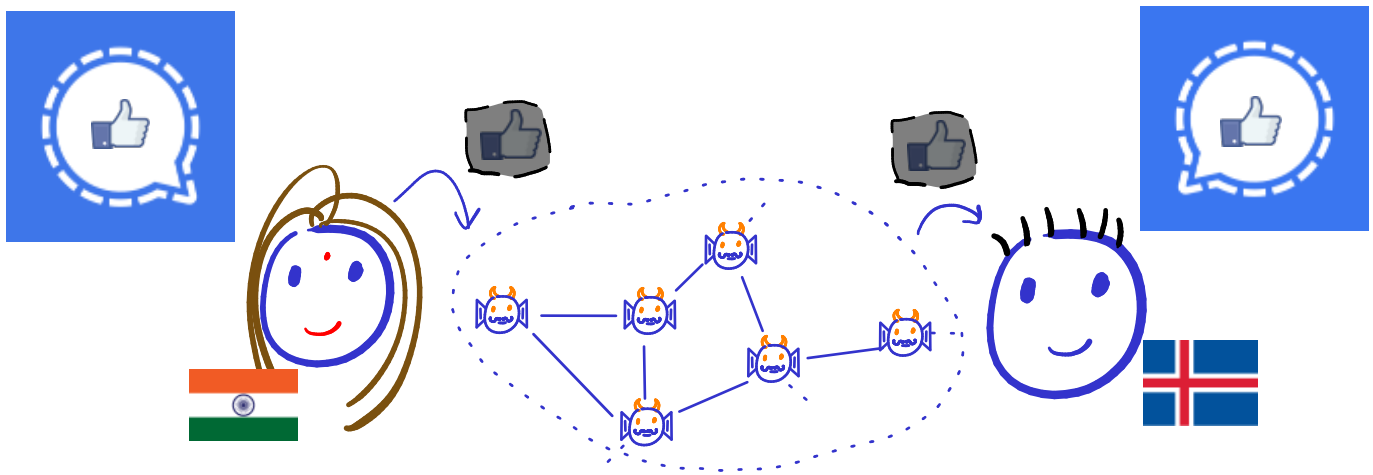
- Science of carrying out *tasks* securely in an adversarial setting
- A loose analogy: gossip



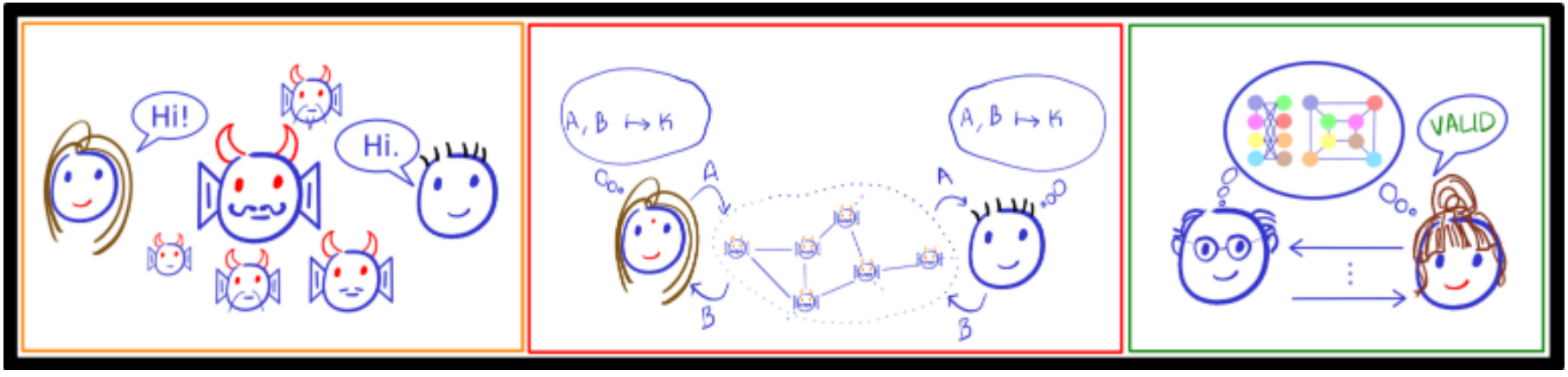
- Security goal: conversation remains secret
- Adversarial setting: eavesdroppers in
  - Bengaluru metro (understand Kannada, English and Hindi)
  - Mumbai local (understand Marathi, English and Hindi)

What is Cryptography?...

- Science of carrying out *tasks* securely in an adversarial setting
- More realistically: texting



- Security goal: messages remains secret
- Adversarial setting:
  - Your TAs
  - The service provider (e.g., designs software, has access to server)
  - State actors (e.g., can tamper with phone, inject malware)



CS409m: Introduction to Modern Cryptography

Lecture 01 (30/Jul/25)

Instructor: Chethan Kamath

Classical *vs.* Modern Cryptography

Guiding principles:

- Code design
- Code breaking

- Formally define security goal and adversarial setting
- Rely on precise, well-studied assumptions
- Rigorous mathmematical security proof

Users:

- Monarchs, military...

- Everyone! (e.g., HTTPs)

E.g.:

- Classical ciphers
- Steganography

- Diffie-Hellman key-exchange, RSA encryption...

Credit for images: Wikipedia

About this Course: What to Expect?

- Undergraduate-level cryptography course
  - Closely follows Sruthi Sekar's CS409m from Fall'24

- Goal: *formally* study how to carry out certain tasks securely in an adversarial setting
- We will follow the following *guiding template*:
  1. Identify the task
  2. Come up with precise threat model M (a.k.a security model)
     - Adversary/Attack: What are the adversary's capabilities?
     - Security Goal: What does it mean to be secure?
  3. Construct a scheme Π
  4. Formally prove that Π in secure in model M

- No prerequisites, but the following is a plus

  - Basic probability, algebra and number theory
  - Knowledge of Python

This Lecture: An Overview of the Modules...

1. Module I: Secure Communication in Shared-Key Setting
2. Module II: Secure Communication in Public-Key Setting
3. Module III: Some Advanced Topics

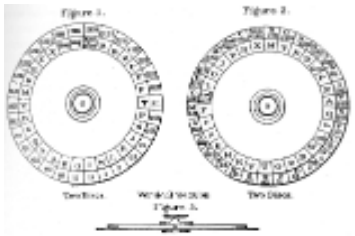
An Overview of the Modules

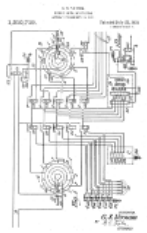
1. Module I: Secure Communication in Shared-Key Setting
2. Module II: Secure Communication in Public-Key Setting
3. Module III: Some Advanced Topics

The Universal Need for Private Communication

MODULE I
(Shared keys)

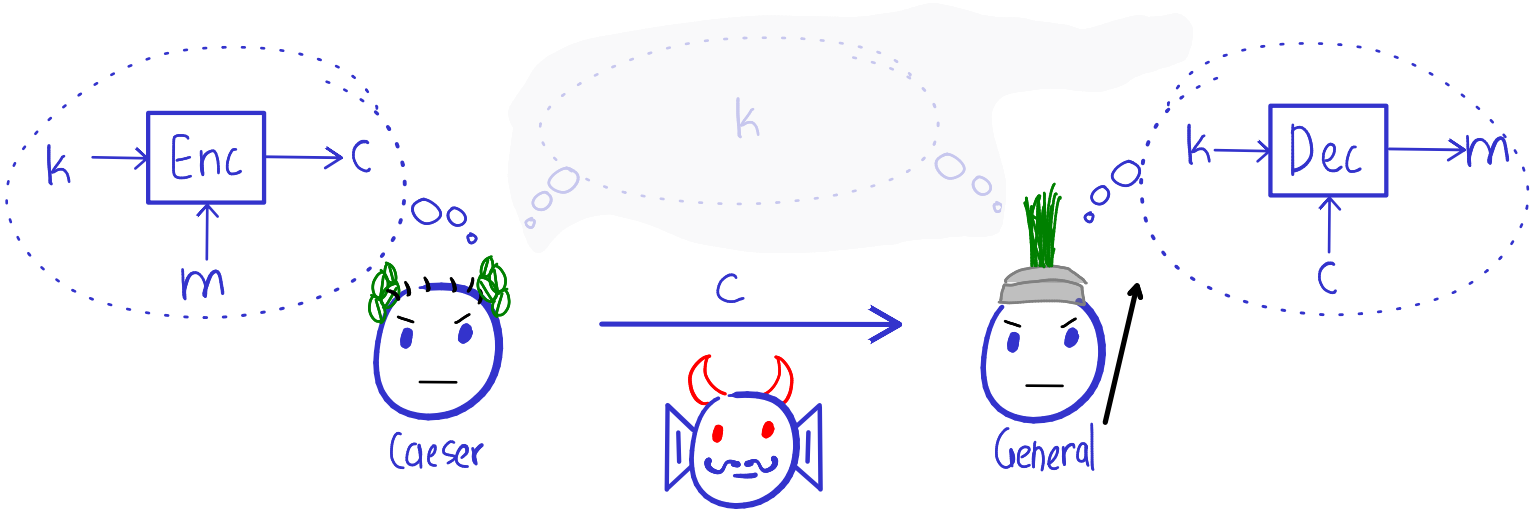
For a large part of history





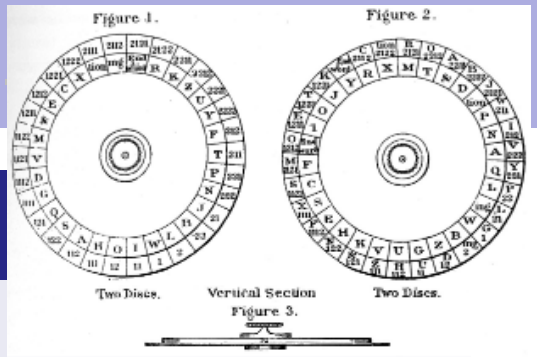
Credit for images: Wikipedia

Task: Secret Communication with Shared Keys...

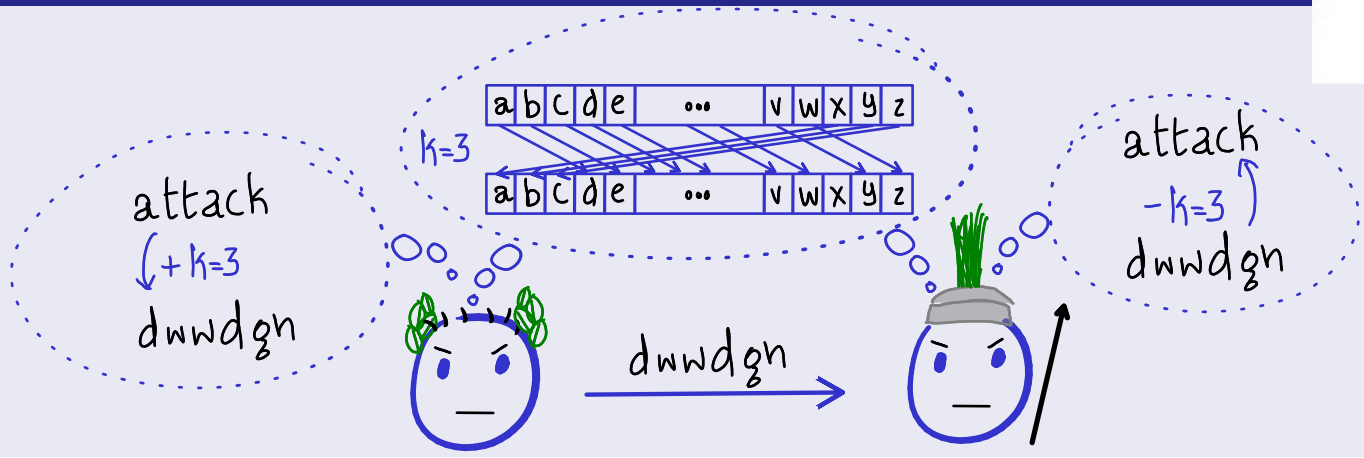


- Setting: Caeser and his General (somehow) share a key k and want to communicate m
  - k from keyspace $\mathcal{K}$
  - m: message from some message space (set) $\mathcal{M}$
  - c: ciphertext (hidden message) lies in ciphertext space $\mathcal{C}$
- *Eve* is listening!

Shift Cipher (Caeser Cipher)



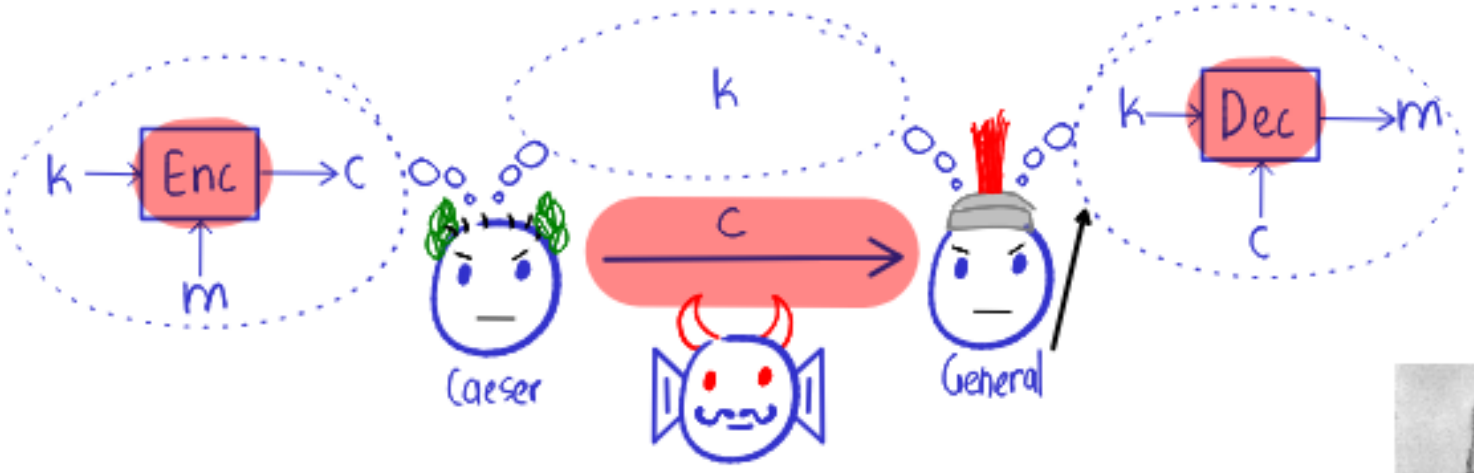
Construction 1 (for message space $\{a, \cdots, z\}^\ell$)



Exercise 1

1. What is the key-space? What is the ciphertext-space?
2. What is the probability that $k = 10$? What is Enc(10, `attack`)?
   Assume that Caeser only sends either `attack` or `defend`.
3. What is the probability that the ciphertext is `kddkmu`, (resp. `kddkmw`)?
4. If ciphertext is `kddkmu`, is it possible that message is `defend`?

First Let's Try to Model our Eavesdropper Eve

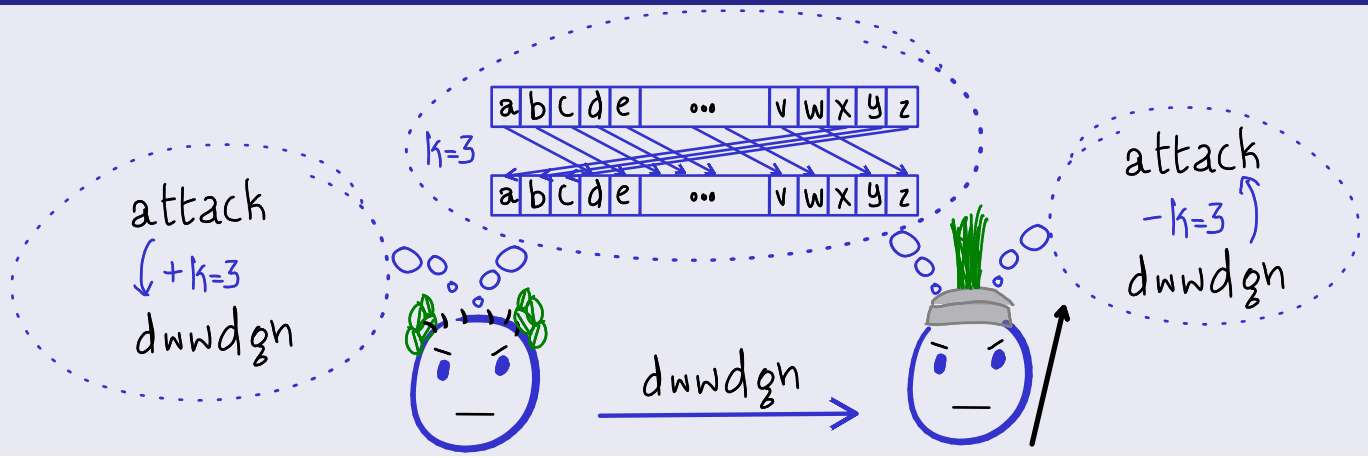


- Can be modelled as an algorithm
- ? What does Eve have access to?
  - Description of the algorithms? Yes, Kerckhoffs' principle:

    'One ought to design systems under the assumption that the enemy will immediately gain full familiarity with them.'
  - What about the key? No, then everything is open
  - Randomness used to encrypt?

Shift Cipher (Caeser Cipher)...

Construction 1 (for message space $\{a, \cdots, z\}^{\ell}$)



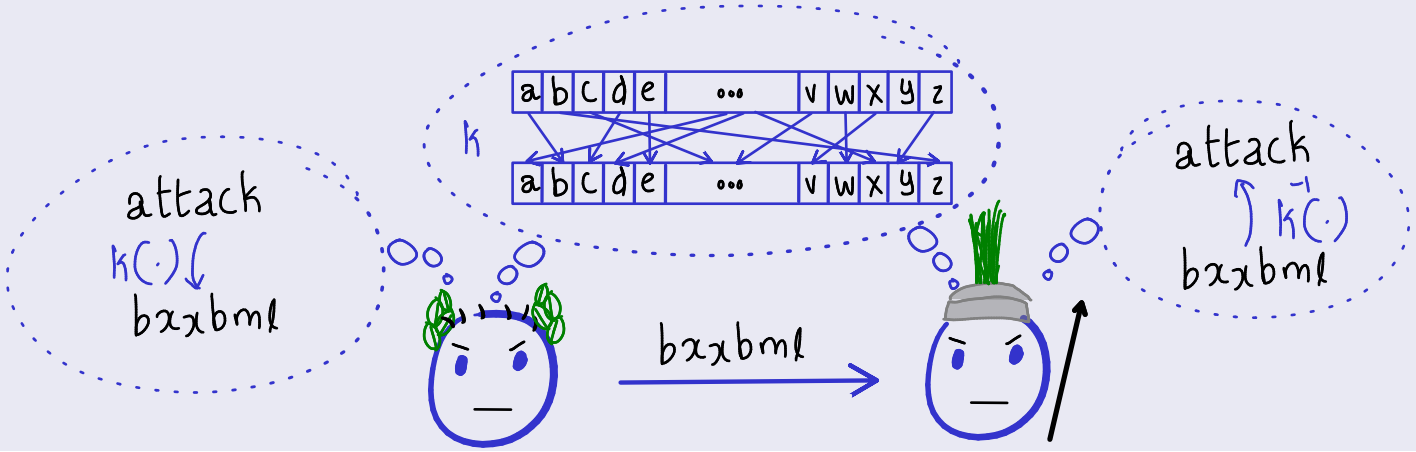
- What can Eve learn?
  - Whole message, by exhaustive key search (brute force)
- What have *we* learnt?
  - *Large-enough* key-space is necessary to thwart *brute force*

Exercise 2

That about what happens if the length of the message is $\ell = 1$

Substitution Cipher...

Construction 2 (Message space $\{a, \cdots, z\}^\ell$)



- Key is a *permutation* of $\{a, \cdots, z\}$.

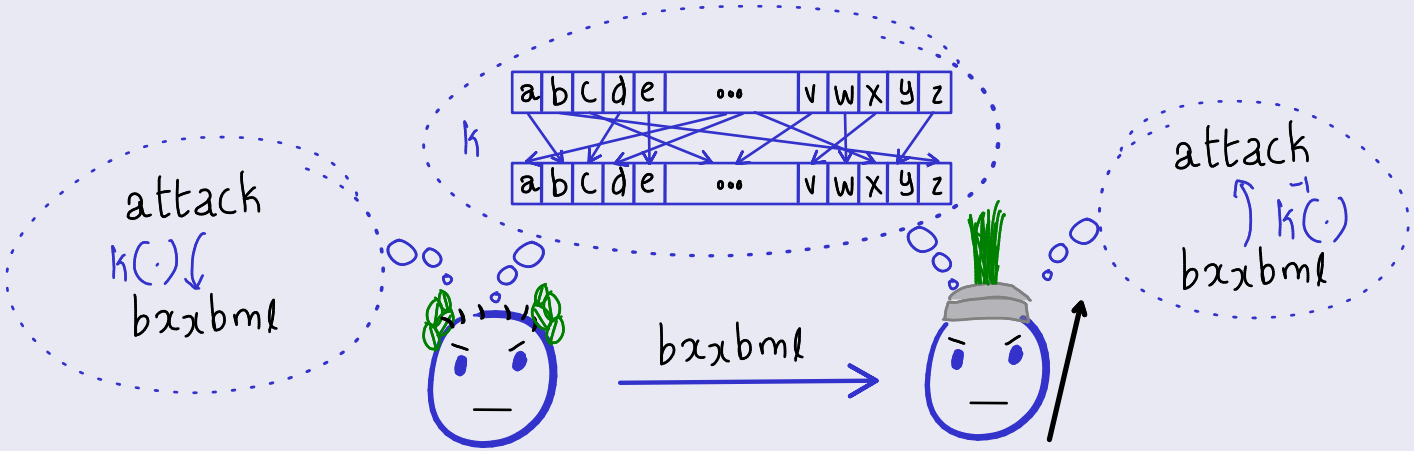
(?) What is the key-space? How large is it?

Exercise 3 (Decrypt the following)

Xibkgltizksb rh gsv hxrvmxv lu hvxfivob xziibrmt lfg gzhph (v.t., hvxivg xlnnfmrxzgrlm) rm zm zwevihzirzo hvggrmt.

Substitution Cipher...

Construction 2 (Message space $\{a, \cdots, z\}^\ell$)

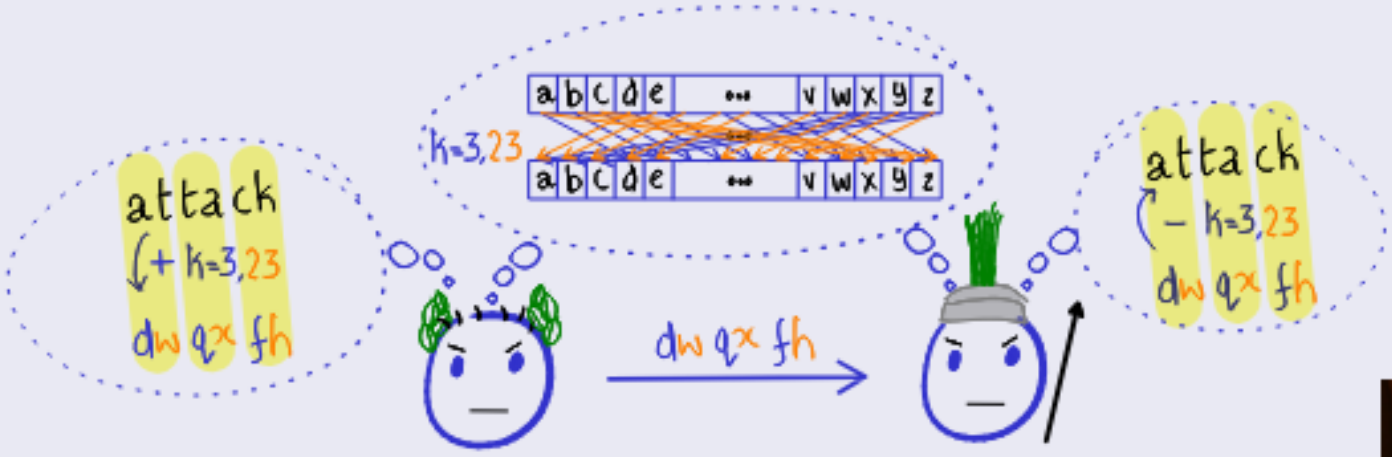


- What can Eve learn?
  - Can easily *distinguish* certain messages. Which?
  - Can recover key with a bit more effort (frequency analysis)
- What have *we* learnt?
  - Large key-space maybe necessary, but is not *sufficient*
  - Must *hide* simple *statistical properties* of the plaintext
    - Should not map a plaintext character to same ciphertext character

Polyalphabetic Ciphers...

- Let's map a plaintext character to different ciphertext characters

Construction 3 (*Polyalphabetic* shift cipher (Vignère cipher))

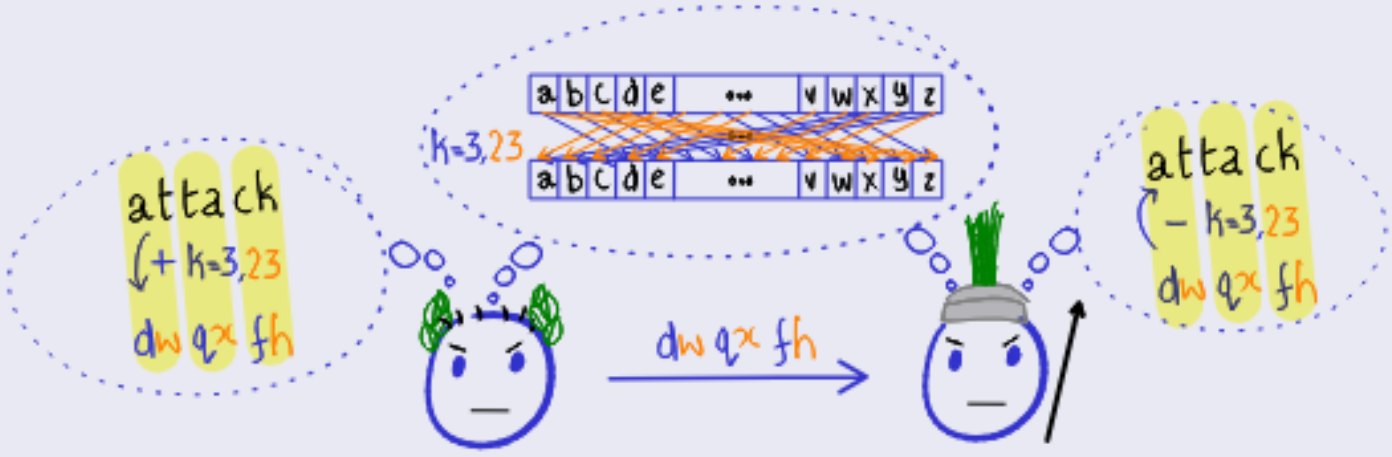


Exercise 4

1. Write down the pseudocode for polyalphabetic shift cipher.
2. Work out the details of *polyalphabetic* substitution cipher.

Polyalphabetic Ciphers...

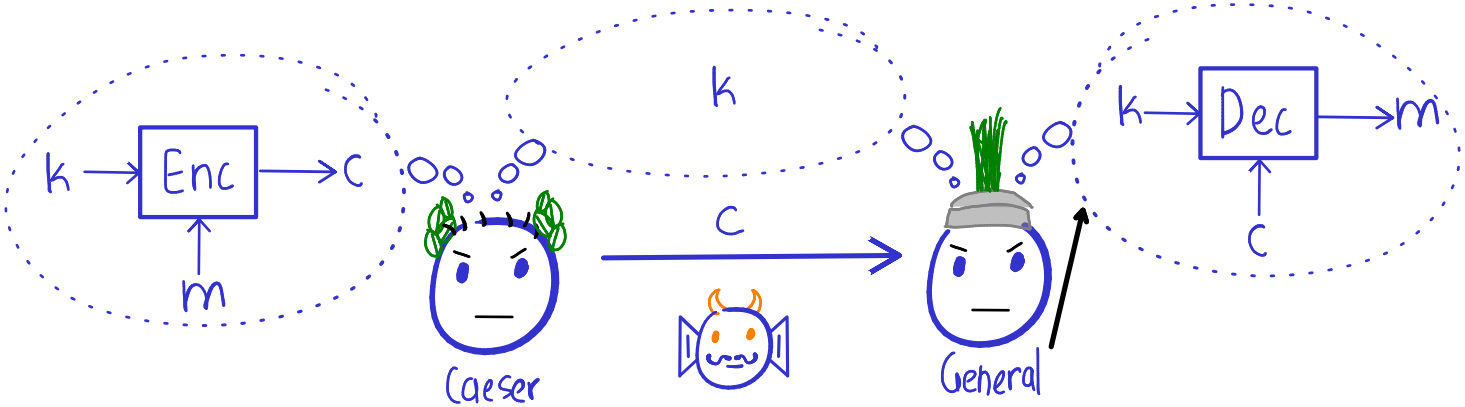
Construction 3 (*Polyalphabetic* shift cipher (Vignère cipher))



- What can Eve learn?
  - Can still *distinguish* certain messages. Any guesses?
  - Can still recover key (more complicated frequency analysis)
- What have we learnt?
  - Must hide *all* statistical patterns of the plaintext
  - Equivalently: Eve must *learn no information* about the plaintext
    - Perfect secrecy!

Task: Secret Communication with Shared Keys...

- What we will learn in Module I:
  - One-time pad (OTP), and why it is perfectly secret
  - Shannon's impossibility: for perfect secrecy, $|k| \geq |M|$

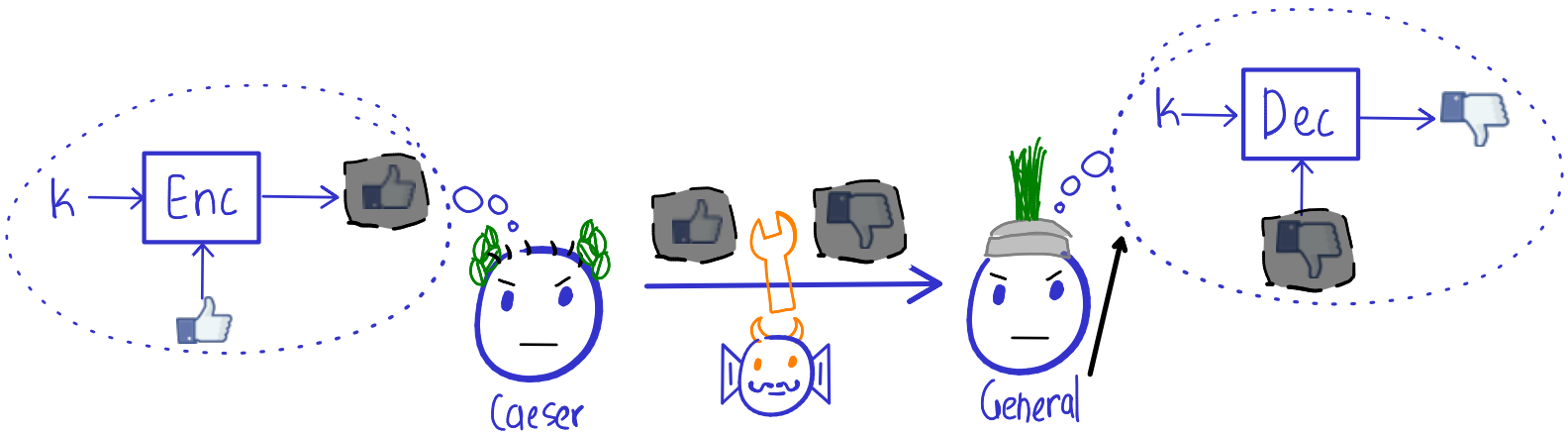


  - How to overcome Shannon's impossibility?
  - Restrict/bound the adversary's computational capabilities
    - How to model computationally-bounded adversaries?
    - Hardness assumptions: e.g., pseudo-random generator (PRG)
    - Secret communication with $|M| > |k|$ assuming PRG

What Else? Dealing with More Resourceful Adversaries

- What if Eve *also* has control over the messages?
- What we will learn: *chosen-plaintext attack* (CPA) and CPA-secure scheme from pseudo-random functions

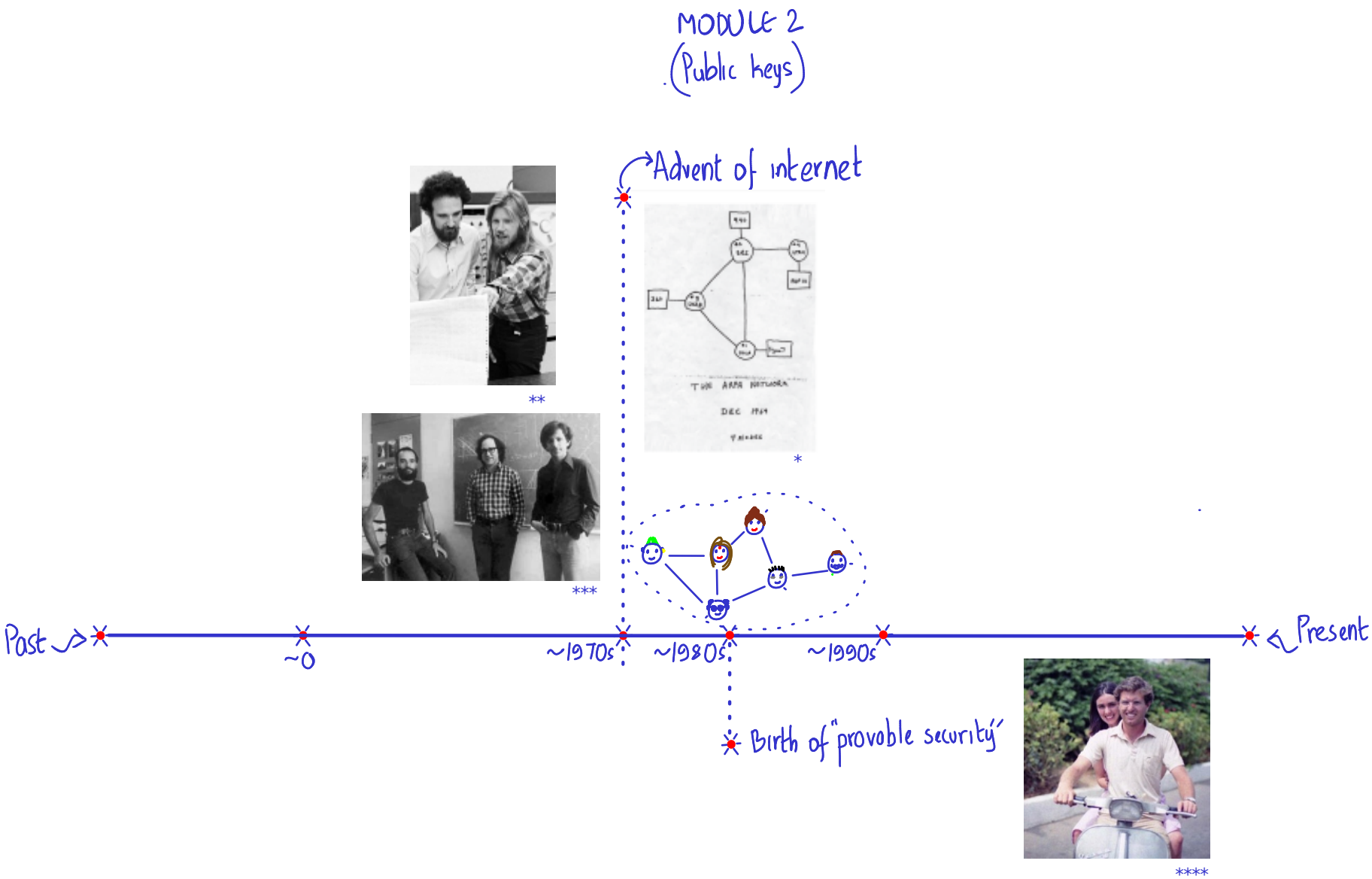
- What if Eve can also *tamper* with the communication?
- What we will learn: message authentication codes



An Overview of the Course

1. Module I: Secure Communication in Shared-Key Setting
2. Module II: Secure Communication in Public-Key Setting
3. Module III: Some Advanced Topics

Advent of Internet and the Scaling Problem

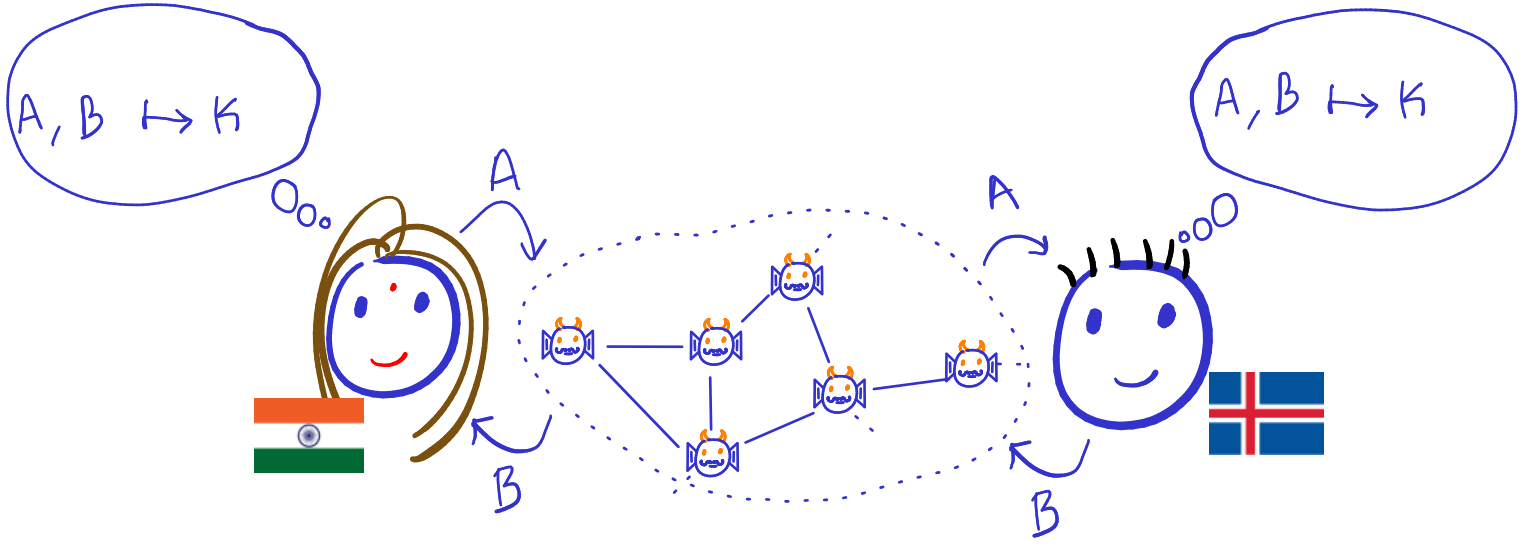


- Limitation of shared-key encryption: requires prior meeting

Credit for images: (*Wikipedia User: DARPA) (**icour.fr) (***cs.miami.edu (Rosenberg)) (****Oded Goldreich)

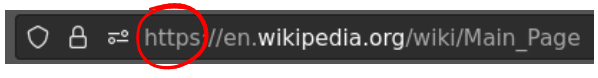
Task 2: Establishing a Shared Key

- Setting: Alice and Bob want to establish a shared key k by communicating *in public* (i.e., exchange a key)



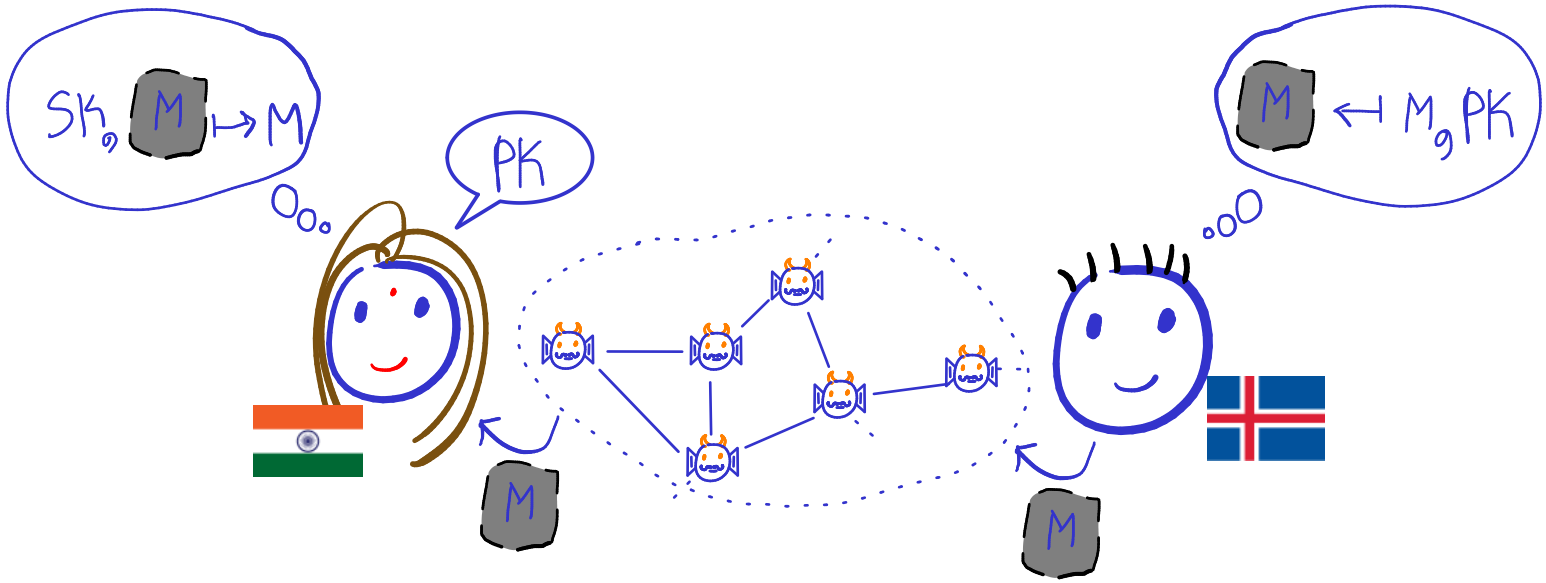
- Threat model
  - Adversary: *Computationally-bounded* eavesdropper Eve
  - Security goal: Eve learns *"no"* information about the shared key

- What we will learn:
  - Some group theory and number theory
  - Diffie-Hellman key exchange



A Related Task: Secret Communication Using Public Keys

- Setting: Alice has published a public key PK, and Bob wants to send a secret message M to her



- What we will learn:
  - Public-key encryption (PKE) from number-theoretic assumptions
  - Equivalence between PKE and key exchange

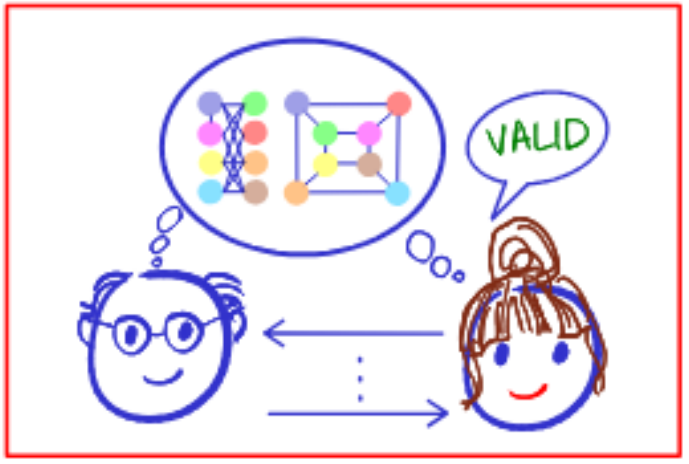
- How to deal with tampering adversary in public-key setting?
- What we will learn: digital signatures

An Overview of the Course

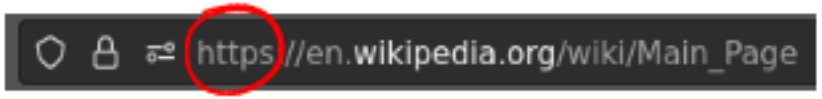
1. Module I: Secure Communication in Shared-Key Setting
2. Module II: Secure Communication in Public-Key Setting
3. **Module III: Some Advanced Topics**

Some Advanced Topics

- Beyond communication?
  - Identification protocols
  - Zero-knowledge proofs
  - ZCash, a cryptocurrency



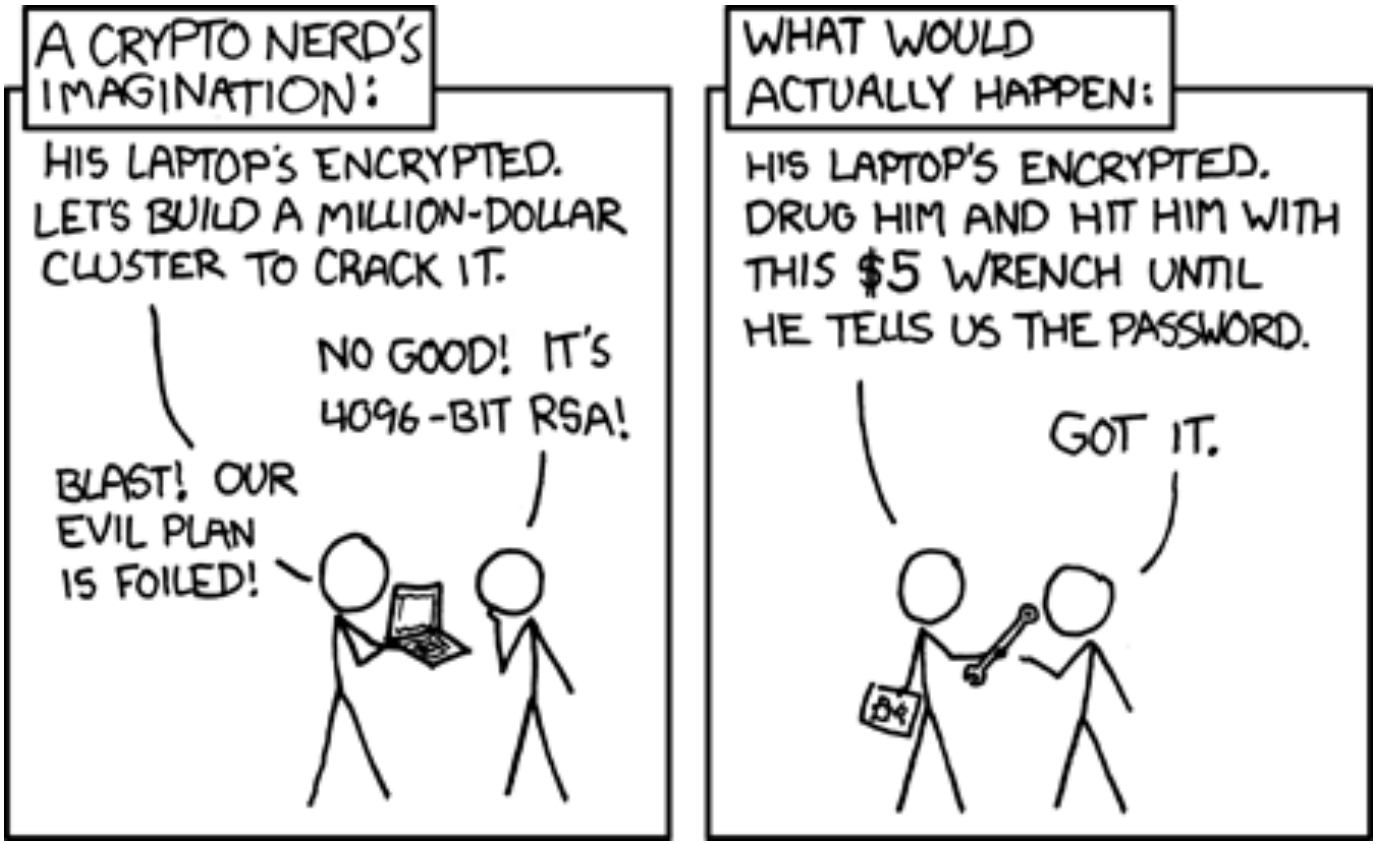
- Combine various primitives!
  - SSL/TLS
  - SSH (if time permits)



- Advanced notions of PKE? (if time permits)
  - Homomorphic encryption

Next Lecture

- Probability toolkit



https://xkcd.com/538/

- More questions?