CIS 5560

Cryptography
Lecture 1

Course Staff

Instructor: Pratyush Mishra

prat@upenn.edu

TAs:

Alireza Shirzad (alrshir@upenn.edu)

Anubhav Baweja (abaweja@upenn.edu)

Bharath Namboothiry (namboo@upenn.edu)

Course Format

- **Lecture:** Tues/Thurs 1:45-3:15PM Fagin Hall 114
- **Grading:**
  - Participation: 5%
  - HW: 20%
  - Project: 20%
  - Midterm 1: 27.5%
  - Midterm 2: 27.5%
- **Important dates:**
  - Midterm 1: TBD
  - Midterm 2: TBD

Homeworks

- Usually, 1 per week
- Released on Wednesdays
- Due Friday 5PM
- Submitted on Gradescope
- Drop 2 lowest scores
- Mostly proof-based, with perhaps one programming oriented homework

Programming Project

- Programming-based project that will require you to provide an end-to-end implementation of a secure system.
- Will be released in second half of class.
- Individual project - no partners.
- Divided into stages (design doc → implementation)
- Will release more info soon!

Collaboration/LLM usage Policy

- Please collaborate with others!
- However *always* write up your own solution
- Write down your collaborators names.

- OK to use LLM tools to help understand lecture material and homework and project problems
- Strongly advised to *not* use it to solve homeworks/projects; you will harm your own learning
- Acknowledge use of LLM-based tools

What is Cryptography?

What is Cryptography Good For?

- Secure communication (TLS, Signal)
- Verifying identity (logins)
- Verifying authenticity/integrity of data (hashes, commit IDs)
- Securely processing data
- Securely sharing data
- … and more!

Cryptography helps us do all of these things today.

But how can we design systems to achieve these tasks?

How do we get there?

The three steps in cryptography:

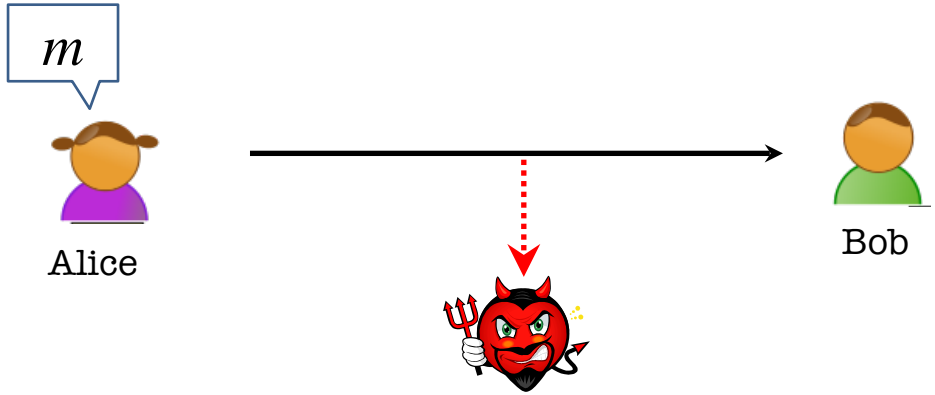
- Precisely specify problem, goal, and threat model
- Propose a construction/protocol
- Prove that breaking construction under threat model will solve an underlying hard problem

Step 1: Figuring out goal and model

Figure out

- What we want guarantees we want to provide
- What adversary can do

Secure Communication



Alice wants to send a message m “securely” to Bob.
What kind of guarantees could Alice and Bob want?

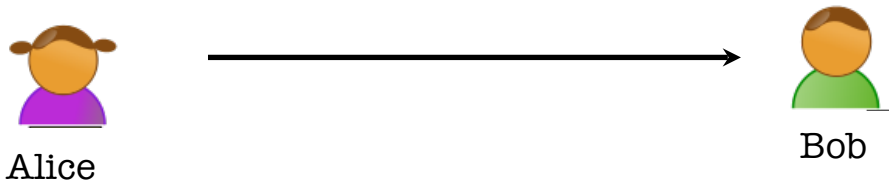
- Nobody should learn m
- Bob should receive m and not some m'
- Bob should be convinced m is from Alice and not somebody else

Identity verification

Alice wants to convince Bob that she's logging in, and not somebody else.

- If login is successful, Bob should be convinced it's Alice
- Even if attacker can try many times
- Even if attacker can observe successful login

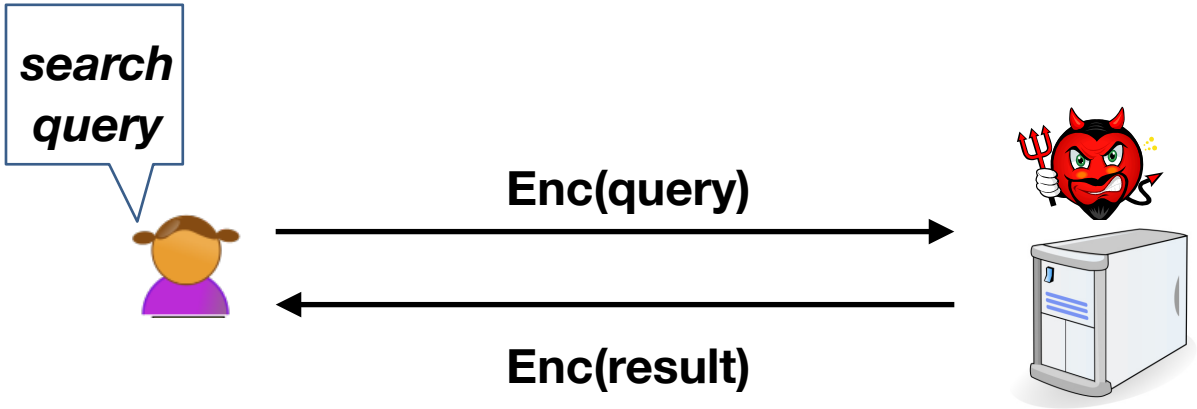
Verifying Integrity/Authenticity of Data



Alice publishes some data. Later, Bob downloads it.
How to convince Bob that he's getting correct data?

- Bob should retrieve the published data.
- Bob should get guarantee that *Alice* published the data.
- Attacker can't stop Bob from getting data?

Secure Data Processing



Alice sends Bob some “hidden” data, and wants Bob to run a program on it and send back (hidden) result.

- Bob should learn nothing about data and result.
- If Bob follows program, Alice should get correct result.
- If Bob uses wrong program, Alice should detect it

How do we get there? Not magic, but science!

The three steps in cryptography:

- Precisely specify problem, goal, and threat model
- Propose a construction/protocol
- Prove that breaking construction under threat model will solve an underlying hard problem

Step 2: Designing a protocol

Once we know our goals, as well as what the attacker can do, then we have to design our protocol.

- Figure out what tools we have
  - Secure communication: block ciphers, MACs
  - Identity: signatures, MACs
  - Data integrity: hashes, signatures
- Figure out how to connect them securely
  - Secure communication: Signal protocol, TLS
  - Identity: European Digital Identity

In this class, we will do all of these things!

- How to reason about security goals and threat model?
- How to design a protocol that meets these goals?
    - What tools can we use?
    - How can we compose them?
- How to formally show that our protocol meets our goals?
    - Formalizing goals as definitions.
    - Formalizing proofs

Things to remember

Cryptography is:

- A tremendous tool
- The basis for many security mechanisms

Cryptography is not:

- The solution to all security problems
- Reliable unless implemented and used properly
- Something you should try to invent yourself
    - many many examples of broken ad-hoc designs