Why Privacy? (+ course policies)

Emma Dauterman

CS 3505

Fall 2025

Agenda:

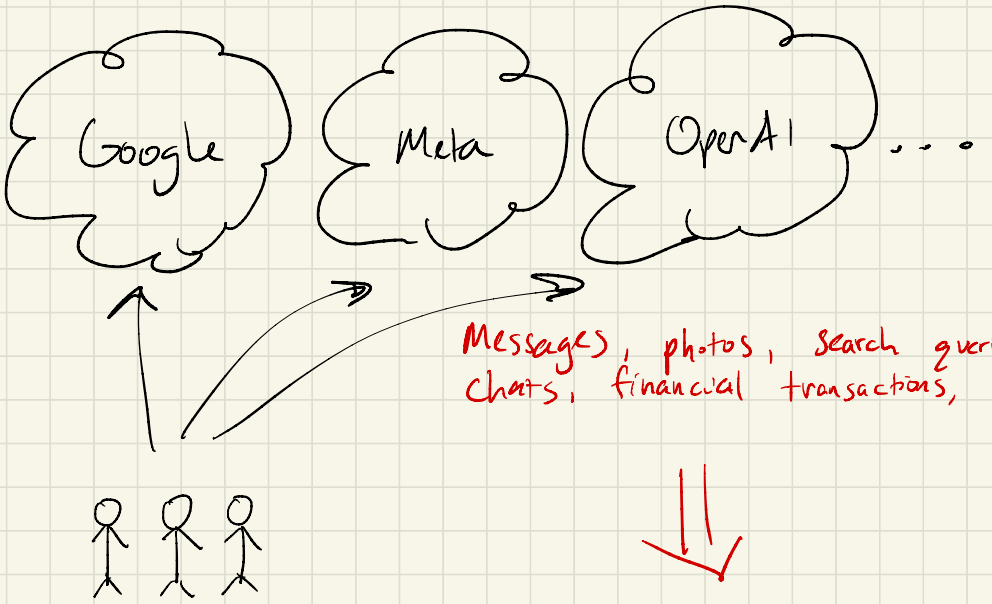
1. Why build privacy-preserving systems?

   What is a privacy-preserving system? (for this class)

2. Course Logistics

3. Cryptographic foundations (time permitting)

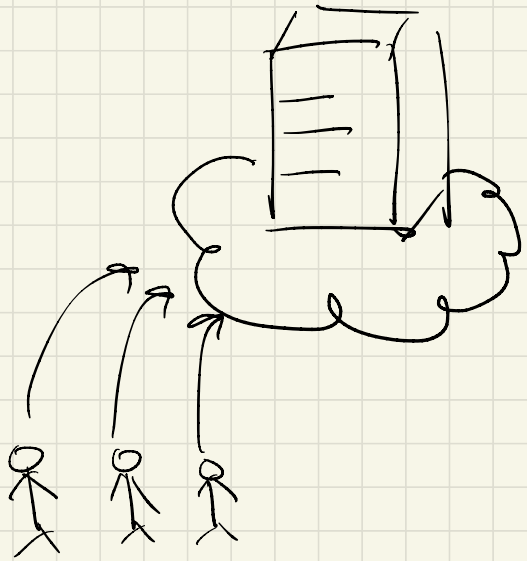
Today: Companies store massive amounts of personal data



Messages, photos, search queries, chats, financial transactions,

Can reveal sensitive information:

- What we care about
- Location
- Political opinions
- Medical information
- Financial information
- ... and more



Massive amounts of data in one place creates potential for

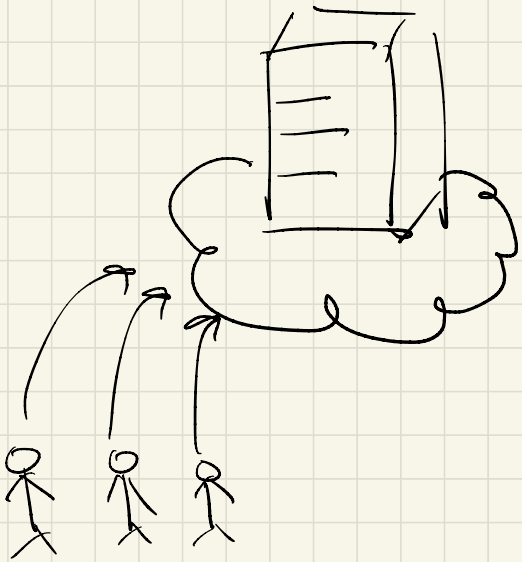
abuse, misuse, or theft of data.

September 2017: Equifax data breach revealing information of ~147M people
- driver's license numbers
- social security numbers
- date of birth
- etc.

Jan 2018: Strava location sharing reveals locations of secret US army bases

October 2020: Therapy records of tens of thousands of patients in Finland stolen, some leaked online

March 2023: Hackers steal password backup data from LastPass (possible to recover passwords if master password is "weak")



Server storing user data becomes

Central point of attack

Attacker can exploit

one vulnerability

to steal

many users' data.

Almost impossible to defend against every possible vulnerability

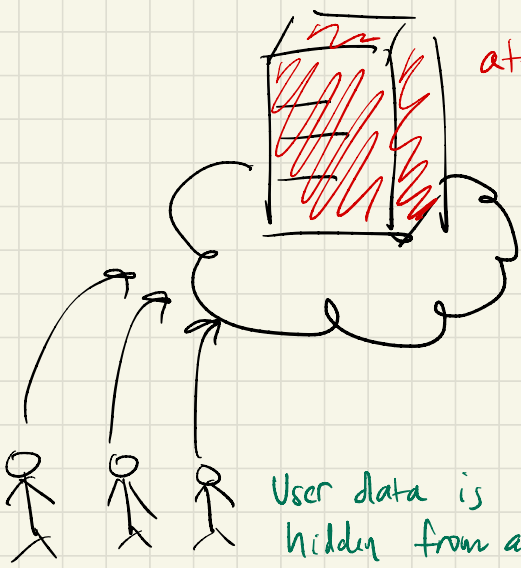
⇒ attacker only needs to find one mistake

Limited alternatives to surrendering our personal data

Expect to be able to:
- Search over the internet
- Send messages to friends
- View recommended content
- etc.

This class: building systems that protect user data in the presence of compromise

Put another way: how to achieve the functionality that users expect without the privacy risk?



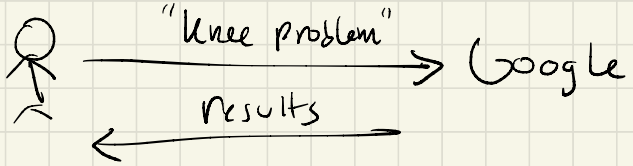
Idea: attacker "learns nothing" about user data

Use cryptography: extracting user data is as hard as solving a cryptographic problem

Some problems that we'll explore in this class:

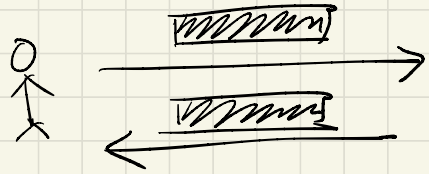
(1.) Private Search

Today:



Problem: Search queries reveal sensitive information.

Instead:

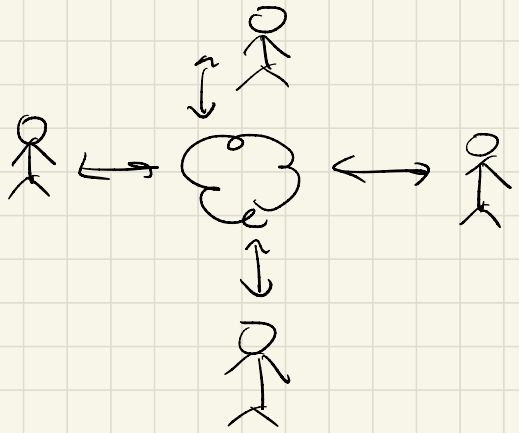


Client sends encrypted query and receives encrypted results

Problem of private information retrieval (Tiptoe)

Some problems that we'll explore in this class:

(2.) Multi-party computation

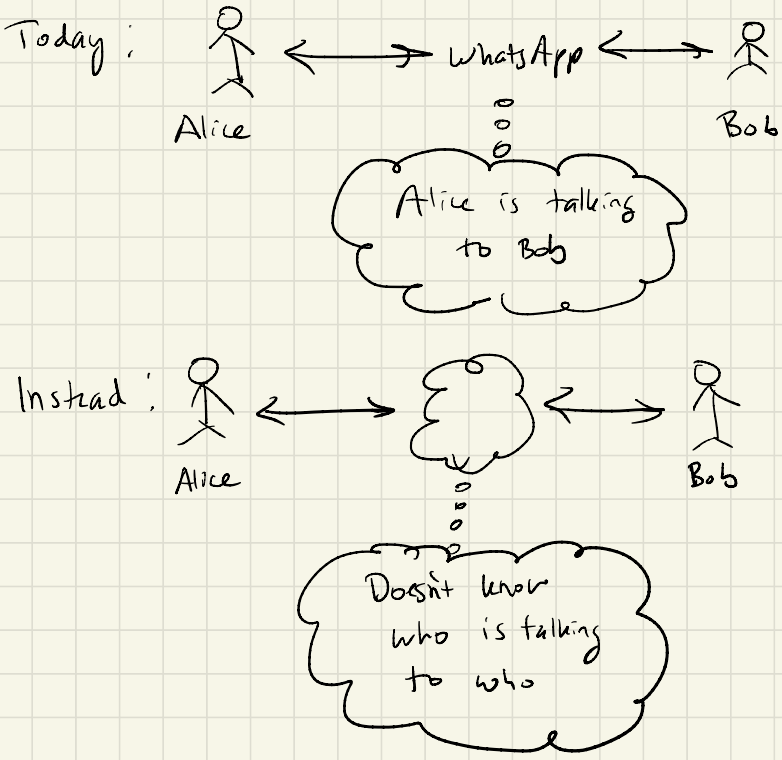


Want to run a computation where different parties inputs' are hidden, and parties only learn the output of the computation

Applications:
- analysis of health data across hospitals
- detecting money-laundering across banks
- training a ML model from users' data

Some problems that we'll explore in this class:

(3.) Private messaging



⇒ Hide communication patterns across users (not just <u>content</u> of communication)

Course Logistics

Course for advanced undergrads + grad students

Recommended prereq: CS 155 or equivalent

Website: cs350s.stanford.edu

Sign up for gradescope + Ed discussion

Office hours: Tuesday 4:30 - 5:30 in CoDa W340
(or by appointment)

Course components:

1. Active participation in class (attendance recorded)

2. Read paper + complete reading response for each class.
   - ↳ due 3PM on gradescope

3. Paper presentation in a small group
   - ↳ give a short presentation on a paper to the class
   - ↳ See signup on Ed after class

4. Final project: building a privacy-preserving system (small-scale research project) in groups of 1-3:

- Project proposal (10/16)
- Progress report (11/6)
- Final project report (12/2)
- Final presentation (12/2, 12/4)

Choosing an idea:

- Read through papers in syllabus or security conferences (IEEE Security & Privacy, USENIX Security & Privacy) and ask if it's possible to achieve
  - better (or different) privacy?
  - better performance?

  or
  - extend the techniques for more functionality, or to a new application domain?

- Come to office hours and talk about project ideas! Helps if you have a general area of interest

In groups of 1-3, you'll need to:

- Conduct a literature review
- Design a system
- Specify security & privacy properties
- Implement + evaluate your system

Course policies:

- no late reading responses accepted
- We'll drop the two lowest reading response grades
- Students can miss two lectures without it affecting their grade.
- We'll have several guest lectures during the quarter - please arrive on time and give our guests your full attention!
- LLM usage is permitted with proper citation (how prompted, any text used in quotes)

  ... but your grade may be affected if you use the LLM for the "challenging" part of the assignment

  Example of permitted LLM usage:
  - asking a LLM about background concepts (other papers or cryptography books may be better resources!)

  Example of LLM usage where you will not receive credit:
  - prompting a LLM with the paper and asking it the reading question (or some variant)

  One of the goals of the class is to learn how to engage with academic papers and read them critically! Offloading this work to a LLM won't help you learn.

Class Outline:

- Week 1: Cryptography basics
  (some overlap with CS 255)
- Weeks 2-9: Privacy-preserving systems topics
  1. Cryptographic foundations (as needed)
  2. Systems that use these tools to provide strong privacy
- Week 10: Final presentations

Agenda:

Crash course in cryptography if you haven't seen it before
(And some review if you've taken CS 255)

1. Symmetric-key cryptography
   - Encryption
   - Hash functions
   - Message authentication codes

2. Public-key cryptography
   - Encryption
   - Signatures

We will <u>not</u> cover:
- how to build these
- comprehensive theoretical foundations

⇒ take CS 255 for this!
(can also look at material online)

Cryptography is a powerful tool

... but only when correctly used and implemented

For this class:

- Projects should focus on new privacy-preserving systems

  ⇒ These may require using existing cryptographic tools

  ⇒ If so, make sure that you are using them correctly!

  ⇒ Cryptography can be dangerous to "invent"

    - Use proven tools and existing libraries wherever possible!

Symmetric-key encryption

Goal:
Attacker should not be able to learn any information about the message that Alice sends Bob.

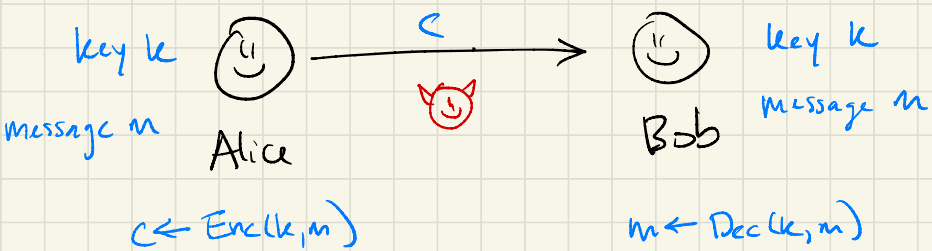
⇒ Confidentiality

Symmetric-key encryption

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

Encryption scheme:

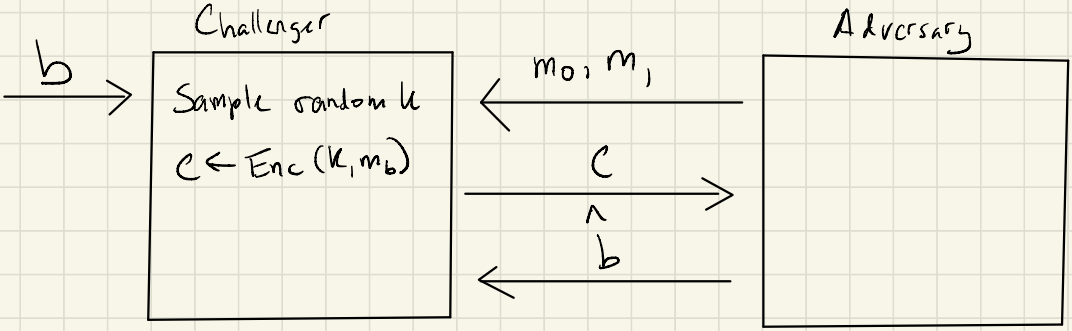
- Encryption algorithm $\quad Enc : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$
- Decryption algorithm $\quad Dec : \mathcal{K} \times \mathcal{C} \to \mathcal{M}$



Confidentiality: Attacker who sees c "learns nothing" about message "m"
(for fixed-size messages)
⇒ how to formally define this?

Semantic Security

Idea: define a game that captures the adversary's advantage in breaking the encryption scheme



An encryption scheme is semantically secure if the adversary's probability of guessing $\hat{b} = b$ is "very close" to $1/2$.

Towards this goal: One-time pad

A simple encryption scheme:

$Enc(k, m)$: Output $k \oplus m$

$Dec(k, c)$: Output $k \oplus c$

Without the key, attacker cannot distinguish ciphertexts from random

Drawbacks:

↳ key is the same length as the message

↳ Cannot re-use key (<u>one</u>-time pad)

If use one-time pad twice:

$c_1 = k \oplus m_1$

$c_2 = k \oplus m_2$

$c_1 \oplus c_2 = k \oplus m_1 \oplus k \oplus m_2$

$= m_1 \oplus m_2$

$\Rightarrow$ Given two ciphertexts, attacker can learn something about the two messages

Stream Cipher

Idea: Use a short seed to compress one-time pad key

Pseudorandom generator (PRG): "stretch" a short, random seed to a long, random-looking string

Seed space S
Output space L

$PRG: S \to L$

Stream cipher from one-time pad + PRG:

$Enc(k, m) = PRG(k) \oplus m$

$Dec(k, c) = PRG(k) \oplus c$

Still can only encrypt <u>one</u> message

Block cipher

Need a way to encrypt many messages with one key

2 tools

1. Pseudorandom function (PRF)

   Key space k, input space X, output space Y

   $F: k \times X \rightarrow Y$

2. Pseudorandom permutation (PRP)

   Key space K, input/output space X

   $E: K \times X \rightarrow Y$

   - Mapping $E(k, \cdot)$ is one-to-one
   - There is an "efficient" inversion algorithm, $D: k \times X \rightarrow X$

   Also called a "block cipher"

   Example: AES

Any PRP is also a PRF

What does it mean for a function / permutation to be pseudorandom?

⇒ Informally, a PRF / PRP should be indistinguishable from a random function / permutation

Using a block cipher to encrypt multiple messages:

Encryption needs to be randomized.

Idea: Use the same key, but a fresh nonce for each encryption

⇒ a key-nonce pair is never reused

Can use key and nonce together to chain PRF evaluation

⇒ Crypto libraries will do this for you!

⇒ Make sure you're using them correctly!

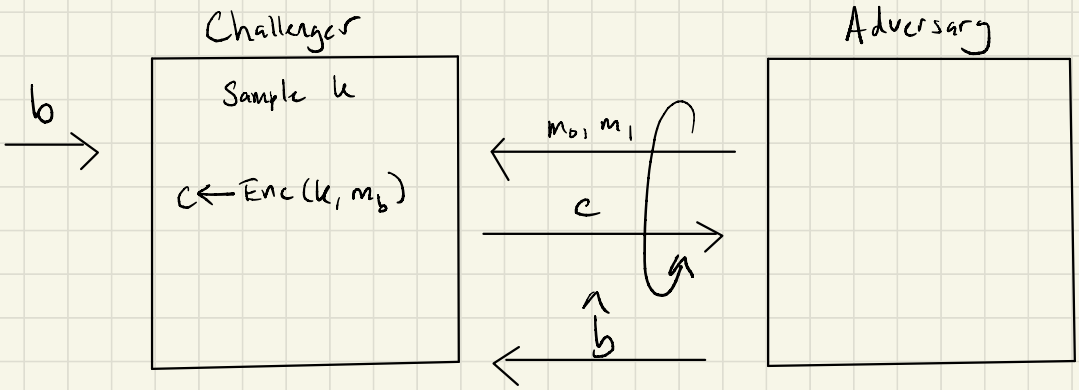
(1) Use a safe "mode" (e.g. CBC, CTR)

(2) Not reusing the same nonce with a key

How to formalize?

Chosen Plaintext Attack (CPA) Security

Expand semantic security to capture multiple messages



An encryption scheme is CPA-secure if the adversary's chance of outputting $\hat{b} = b$ is "very close" to $1/2$.

Note: says nothing about ability to tamper with ciphertext

$\Rightarrow$ Chosen ciphertext attack (CCA) security

For more formal definitions, take CS 255!

Message integrity

Alice and Bob want to ensure that the attacker has not tampered with the message

⇒ Integrity

Confidentiality and integrity go hand in hand

⇒ An attacker that can change a ciphertext can potentially cause application-level damage, depending on the original message

⇒ If the attacker sees this damage, it can learn something about the original message

Takeaway: Use authenticated encryption

(Ex: AES in GCM mode)