EECS 498/598: Encrypted Systems
Winter 2022
Lecture 4: Encrypted Data Management
(EDM)

Paul Grubbs

paulgrub@umich.edu

Beyster 4709

Agenda for this lecture

- Announcements
- The problem
- A non-solution
- Key ideas in EDM research
- Take-home questions

Agenda for this lecture

- Announcements
- The problem
- A non-solution
- Key ideas in EDM research
- Take-home questions

Announcements

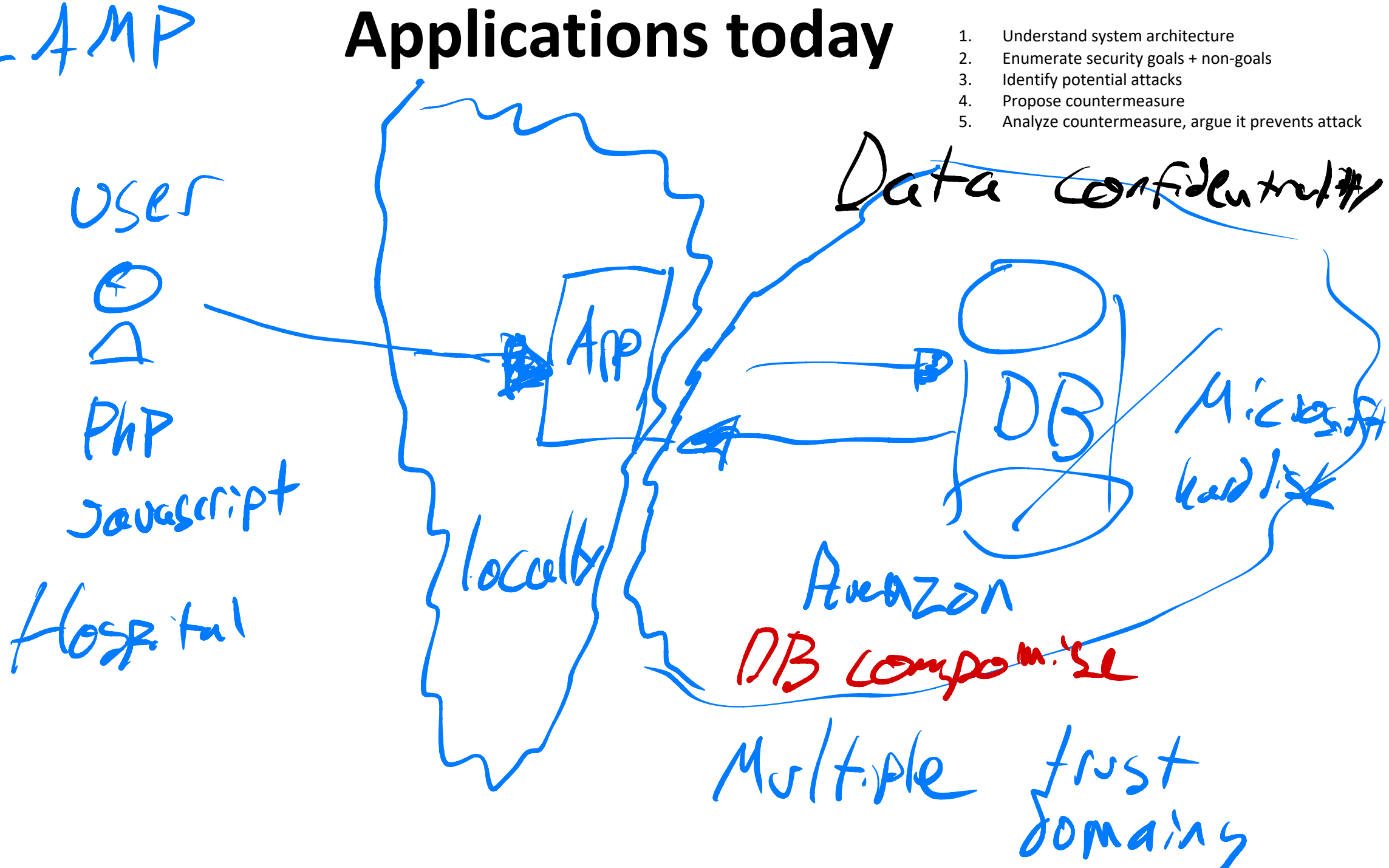
- Small mix-up with preferences – please submit again
- Course Slack is up and running. Post and say hello!

Agenda for this lecture

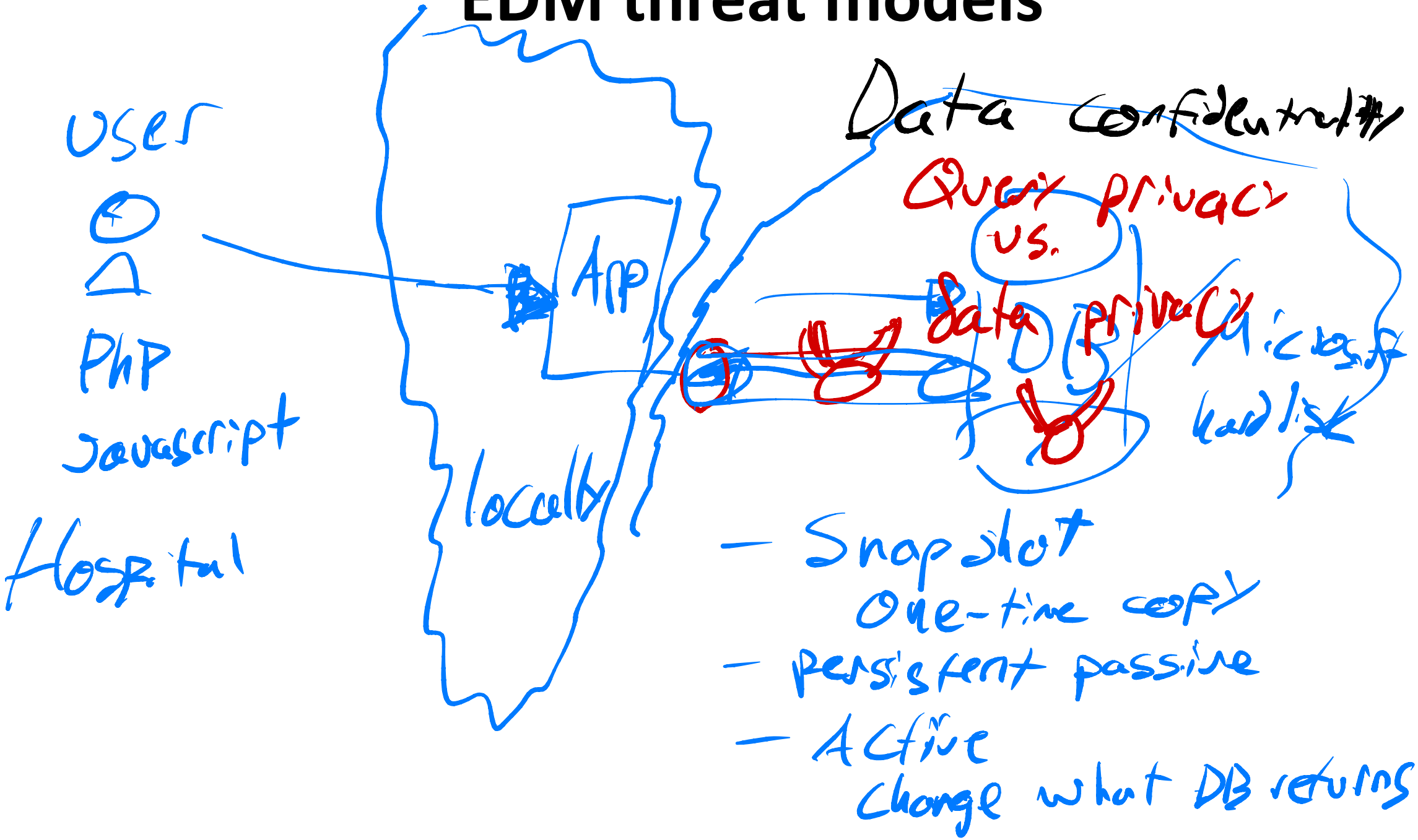
- Announcements
- The problem
- A non-solution
- Key ideas in EDM research
- Take-home questions

Applications today

1. Understand system architecture
2. Enumerate security goals + non-goals
3. Identify potential attacks
4. Propose countermeasure
5. Analyze countermeasure, argue it prevents attack



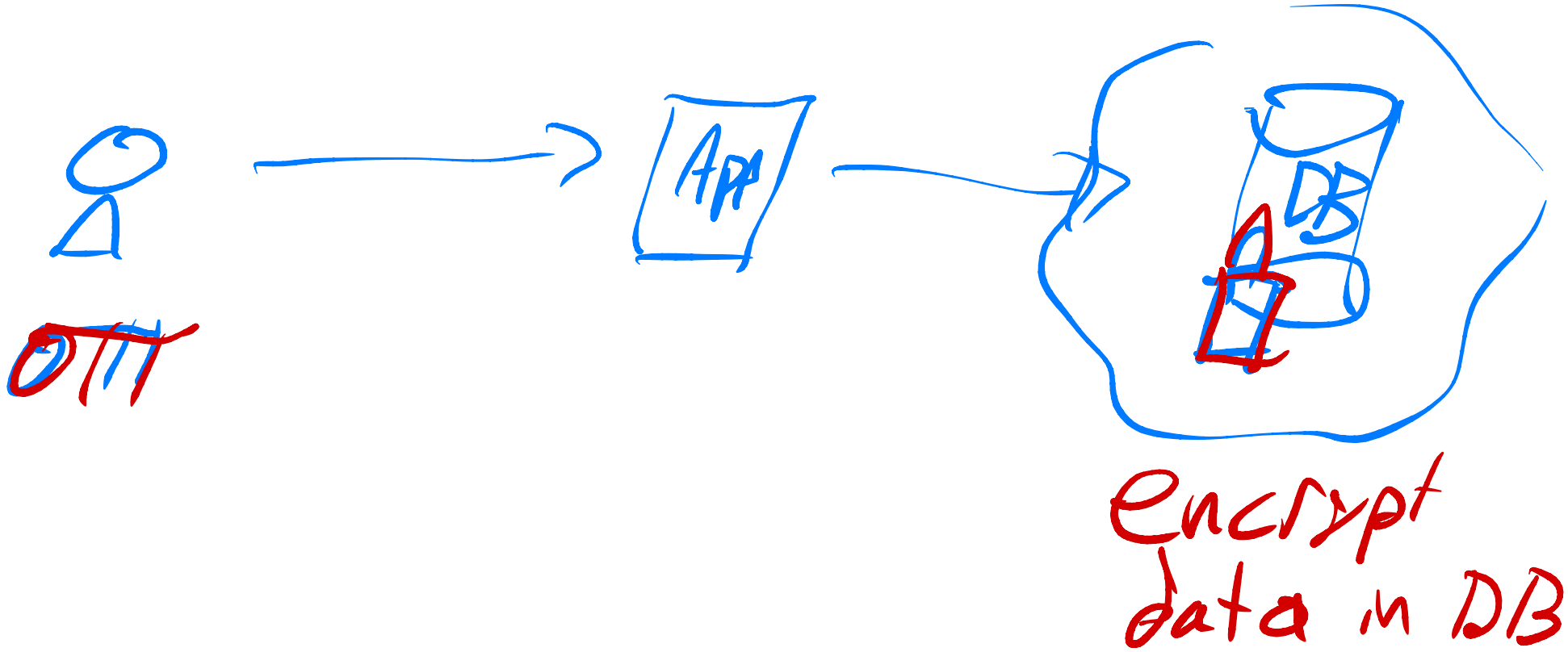
EDM threat models



Agenda for this lecture

- Announcements
- The problem
- A non-solution
- Key ideas in EDM research
- Take-home questions

A non-solution



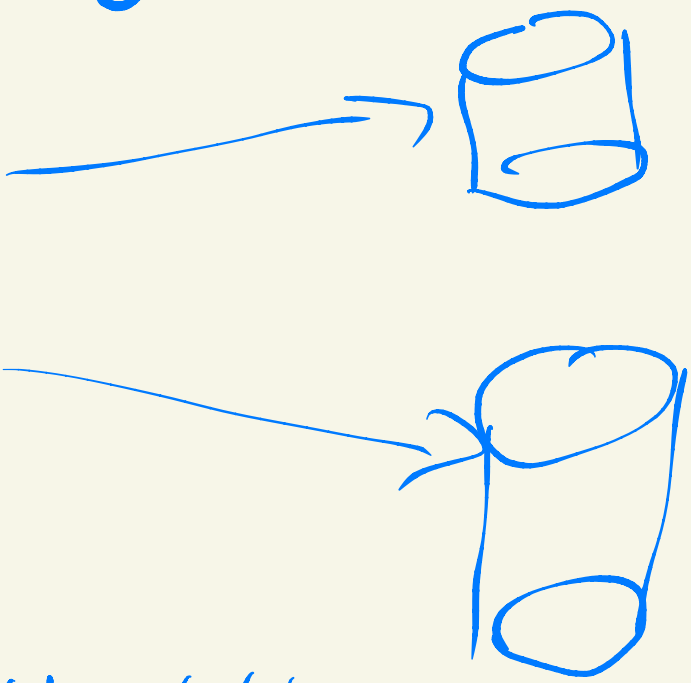
Agenda for this lecture

- Announcements
- The problem
- A non-solution
- Key ideas in EDM research
- Take-home questions

- preprocessing PIR

Each query has two stages:

- "hints" hint1, hint2, ..., hintK
- query



- amortization, batching
- Fully homomorphic encryption

PIR Security

- client query confidential
- NO DB hiding

"Symmetric" PIR:

- privacy for data and query

Key Ideas

Different kinds of privacy
- query vs. data privacy

Different kinds of leakage
- access pattern
- volume leakage
- timing info

client capabilities:
- multiple vs. one client
- storage (size, persistence)
- stateful → PIR: no state!
  → ORAM: stores state

Server capabilities
- PIR: multiple non-colluding servers
- balls-and-bins server computation abilities

Agenda for this lecture

- Announcements
- The problem
- A non-solution
- Key ideas in EDM research
- Take-home questions

Take-home questions

- Would this stop breach XX?
- Threat model?
- Could you add EDM to your application?

Searchable encryption

- more efficiency + more leakage

why do we care? ORAM/PIR have no leakage

- ORAM/PIR very inefficient!
- IS this inherent?

Yes!!

- Known lower bounds

ORAM

size-N

bandwidth overhead $\Omega(\log N)$

Larsen-Nielsen '18

PIR

size-N DB

Server's computation

$O(N)$

amortizing, preprocess.

Efficiency/security tradeoffs inherent!

Constructive:

- relax ORAM/PIR?
- efficient constructions?

Attacks:

- How much leakage is too much?
- Empirical