Privacy Changes Everything

JENNIE ROGERS
NORTHWESTERN UNIVERSITY

Joint work with Xi He, Ashwin Machanavajjhala, and Xiao Wang
The elephant in the room

• Web companies are stewards to an amazing amount of peoples’ private data

• This isn’t going well.

• Governments are stepping in and regulating how they store, share, and use data – e.g., GDPR, CCPA

If we don’t start managing our data with privacy-preserving systems, the targeted advertising party may be over!
Companies know about this issue...
Security and Privacy (S&P) Settings

- Column-at-a-time access control policy
- Systems behave exactly like a standard DBMS from client’s perspective
Why can’t I just add a password to my DBMS? And use encrypted storage?

• Your attack surface is the entire stack

• Your data may reside on untrusted cloud servers

• *Privacy changes everything* – we need to re-architect our systems with it as a first principle
Naïve DBMS deployment on untrusted cloud

What could go wrong?

- **Storage**: National Security Letter compels service provider to decrypt data

- **Query processing**: insider threat sees data-dependent query traces and result sizes

- **Client side**: rogue user systematically queries DB to deduce its private contents

Regulatory compliance ≠ meaningful S&P guarantees!
What about existing work from the security and privacy community?

- Existing S&P solutions are piecemeal – they address specific steps in the DBMS workflow
- Usually require multiple PhD-level experts to deploy them
- When deployed, their apps are almost always hard-coded
- Composing these techniques is non-trivial
Where do we come in?

• To date we’ve mostly focused on making the DBMS fast and scalable – to great success!

• S&P is usually an afterthought

• We have a lot to offer in this emerging space of making privacy-preserving analytics practical and usable
Trustworthy Database Management Systems

- **End-to-end S&P guarantees**: Covers from when a client submits a query to when they receive their results.
- **Efficient**: Offers performance comparable to that of current DBMSs.
- **Robust**: Supports ad-hoc query workloads and diverse user needs.
- **Usable**: UI like that of a standard DBMS for low barrier to entry. Understandable S&P guarantees.
Query Optimizer Goals

• Rigorous, explainable guarantees of:
  • Privacy of storage and computation over data
  • Privacy of data itself, esp. under repeated querying

• Maximize:
  • Query result accuracy
  • The efficiency of query execution

S&P guarantees are not boolean, this leads to interesting trade-offs
Secure multiparty computation (MPC) protects data during query evaluation with:

- Data encrypted in flight
- Oblivious, data-independent execution transcript
- Useful for query processing in untrusted cloud
- MPC is really slow! 1,000X+ slower than running in the clear
Building Block: Differential Privacy

- Differential privacy injects carefully controlled levels of noise into query results
- Protects private tuples from reconstruction attacks
- Computational differential privacy blurs transcript of SQL-over-MPC – substantial speedups over full-oblivious!
Why can’t we naively integrate these building blocks into existing systems?

Untrusted cloud, noisy query results
What about access control?

• Need to check permissions for ad-hoc SQL queries with complex, composable access control policies

• Need to account for privacy budget availability and utility of query results
But wait, there’s more!

- All of this gets harder when we start to think about transactions and recovery

- Trustworthy DBMSs will want to provide proofs that their query results are correct and complete

- With distributed DBMS or multiple data owners trust models per node will factor into how we efficiently run queries
VaultDB, a Private Data Federation

• A data sharing platform for people who don’t trust each other

• Querying the union of tuples from 2+ private datastores

• SQL-over-MPC with differentially private results

• Oblivious queries reveal no information except that which can be deduced from query results
How it is done now...

“How many patients suffer from rare disease X?”

```
SELECT COUNT(DISTINCT patient_ID)
FROM diagnosis
WHERE diag=X;
```
The VaultDB Approach

“How many patients suffer from rare disease X?”

SELECT COUNT(DISTINCT patient_ID) FROM diagnosis WHERE diag=X;
VaultDB Architecture

How many patients suffer from X?

1. Crypto protocols to run query securely
2. Secure computation over records of all data providers
3. Secret shares of query output (1/data provider)

Client Workstation

VaultDB

N patients
VaultDB

• Investigating practical models of semi-oblivious query processing and query optimization using constraints from table definitions

• Preparing to pilot VaultDB over electronic health records with Chicago-area hospitals (CAPriCORN)

• Diverse team spanning DB, S&P, and medicine

Xi He
Waterloo

Abel Kho
Northwestern Med.

Ashwin Mechanavajjhala
Duke

Xiao Wang
Northwestern

CAPriCORN
Conclusions

• Private data is everywhere, we need to take a principled approach to managing it

• We must reimagine the DBMS with S&P as a first-class citizen in design

• We can’t solve this problem alone, need deep collaborations with colleagues in S&P, domain experts

For more on this, see:
Jennie Rogers, Johes Bater, Xi He, Ashwin Machanavajjhala, Madhav Suresh, and Xiao Wang.
Questions?