Trends in databases ∩ crypto/security

RALUCA ADA POPA
UC BERKELEY

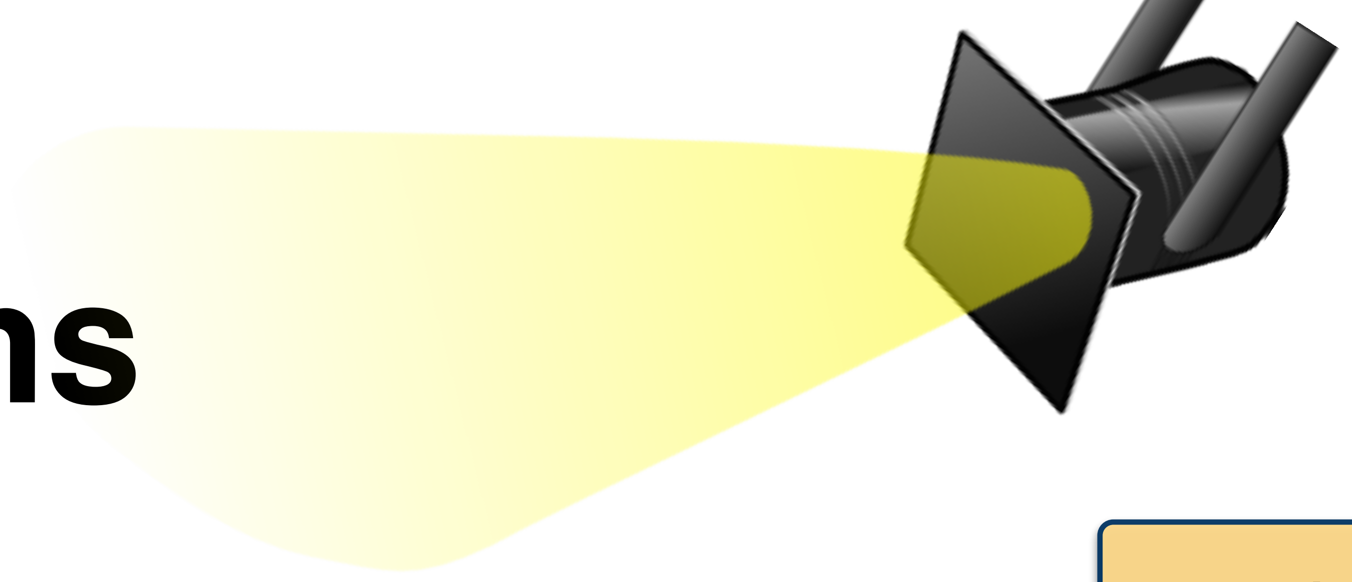
riselab
UC Berkeley

This talk…

1. Trends: Key database-related problems that the crypto/security community is currently working on / needs help with

2. Suggestions: My call for action

An exciting time for crypto+systems



- The world has started building systems using **advanced cryptography**
- ~$300 billion rely on advanced crypto (in cryptocurrencies)
- Enables building much more interesting database systems

> - authenticated data structures (e.g., blockchains, Merkle trees)
> - zero-knowledge proofs

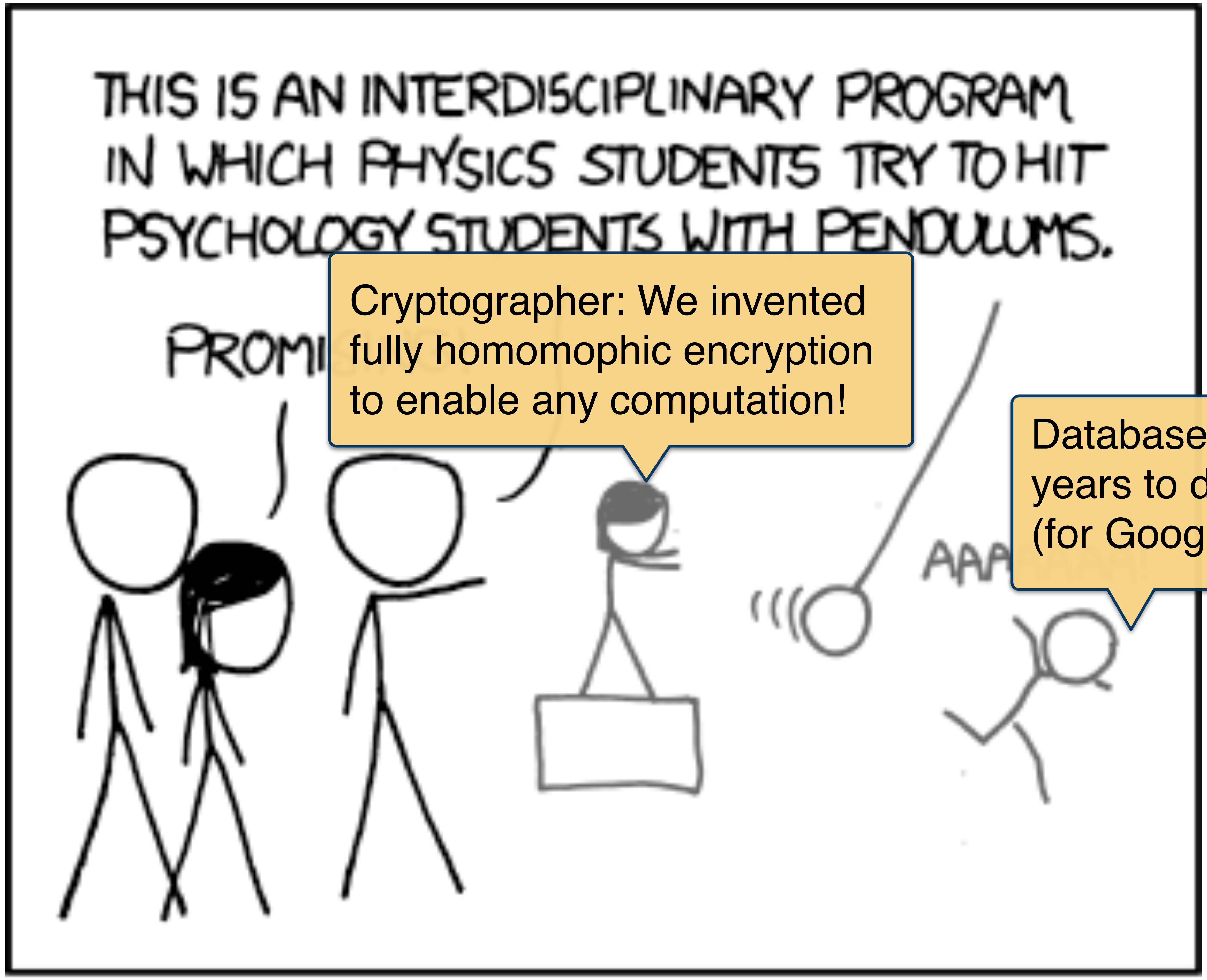
Key problems and trends in a nutshell…

1. Secure federated analytics
2. Decentralized security
   - Scalable consensus
   - Scalable authenticated data structures
3. Side-channel prevention

Call for action in a nutshell…

- Address the 3 problems requires **close collaboration** between database systems and cryptography experts
- It is not enough for database systems to use cryptography as as **black box**

Example: database folks want to hide the contents of a database and the query execution from a DB server

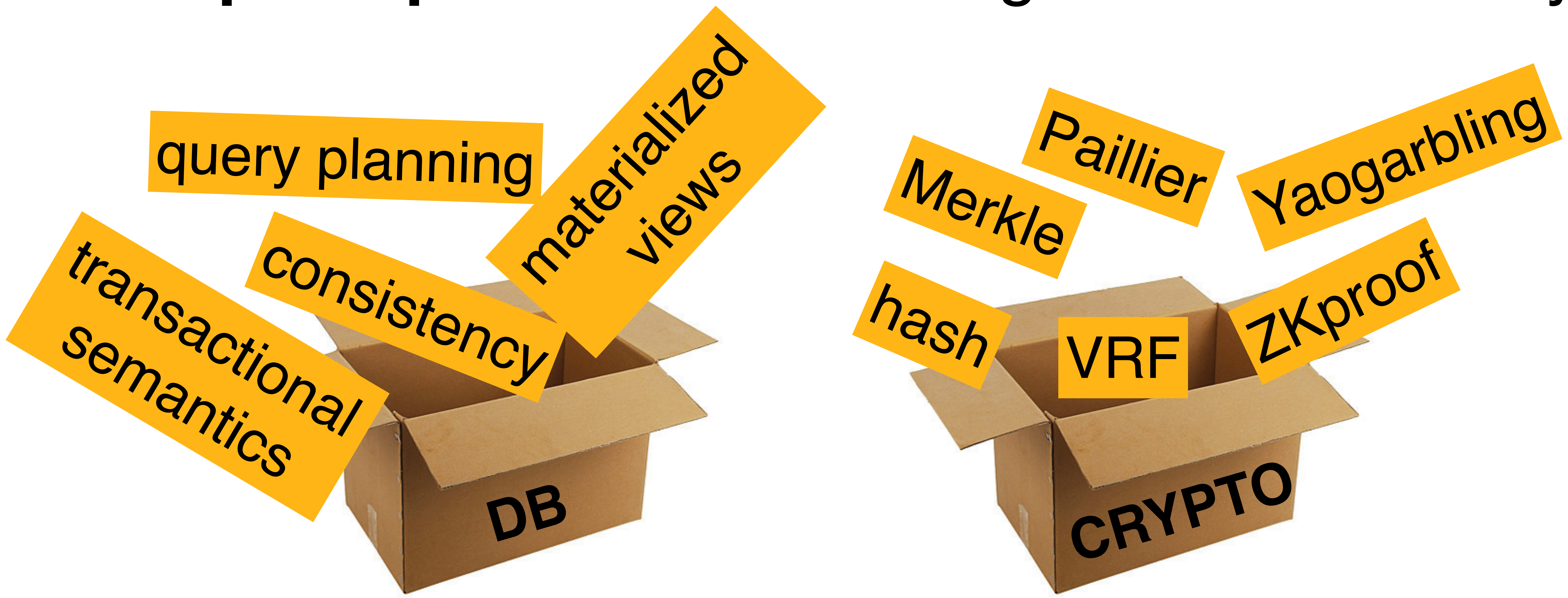


Cryptographer: We invented fully homomophic encryption to enable any computation!

Databaser: It takes a million years to do a Google search (for Google's database size).

Call for action in a nutshell…

- We need to **open up both boxes** and get our hands dirty



1. understand your database workload
2. open up the crypto box, understand the tools it has, their pros and cons, or design new ones
3. use database principles or tools to compose them (e.g. query planning)

tailored crypto is much faster!

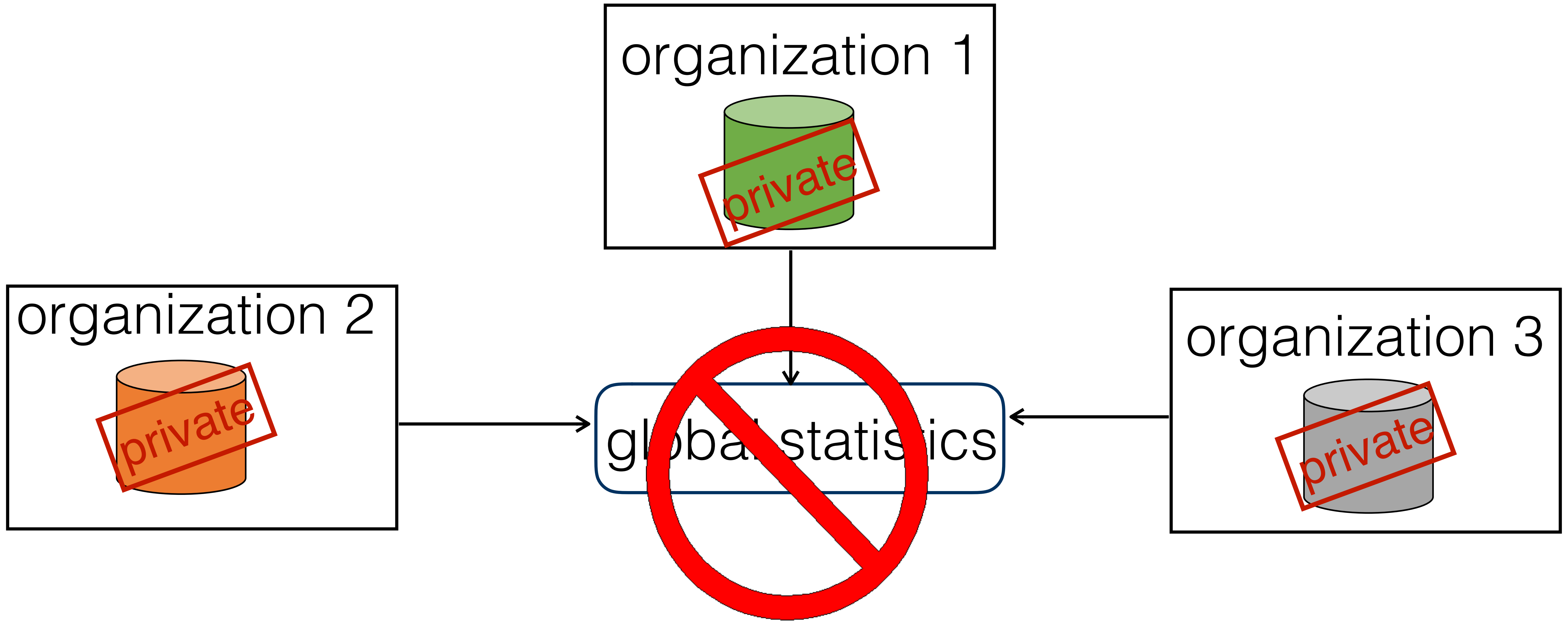
the DB person has to learn some advanced crypto

[CryptDB, Verena, BlindBox, Embark, PrivStats, VPriv, …]

Secure federated analytics

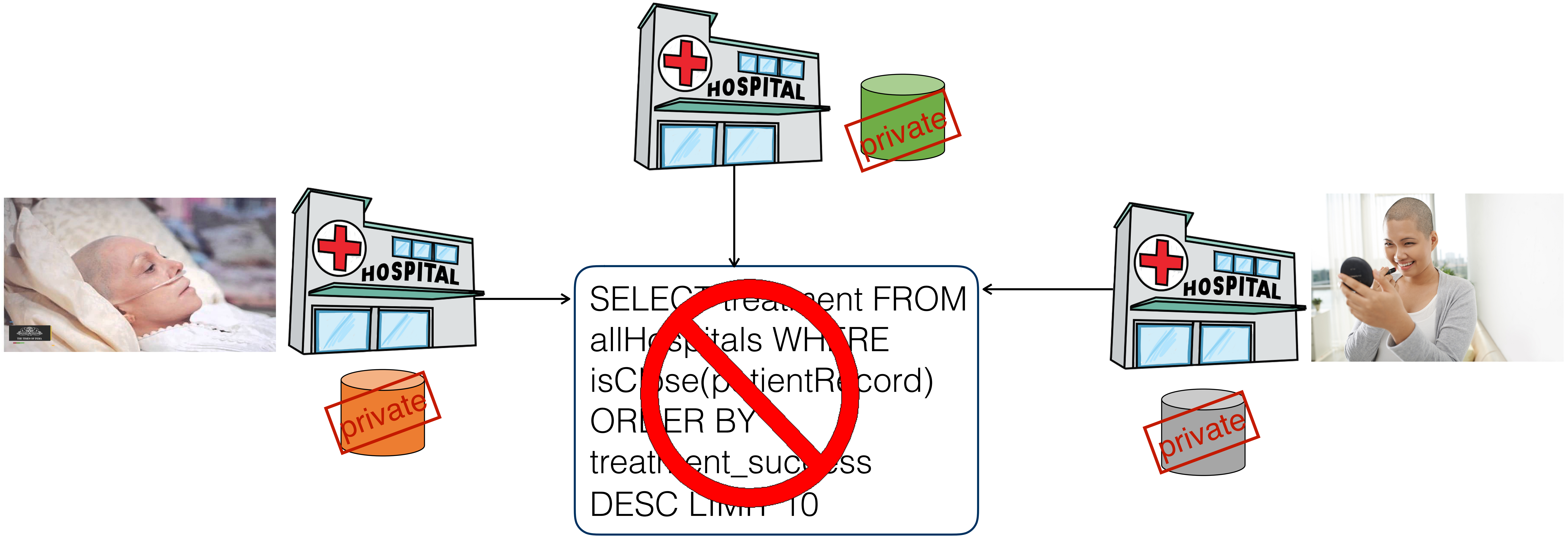
Problem

- N organizations have sensitive databases they cannot share
- They want to run **joint** data analytics



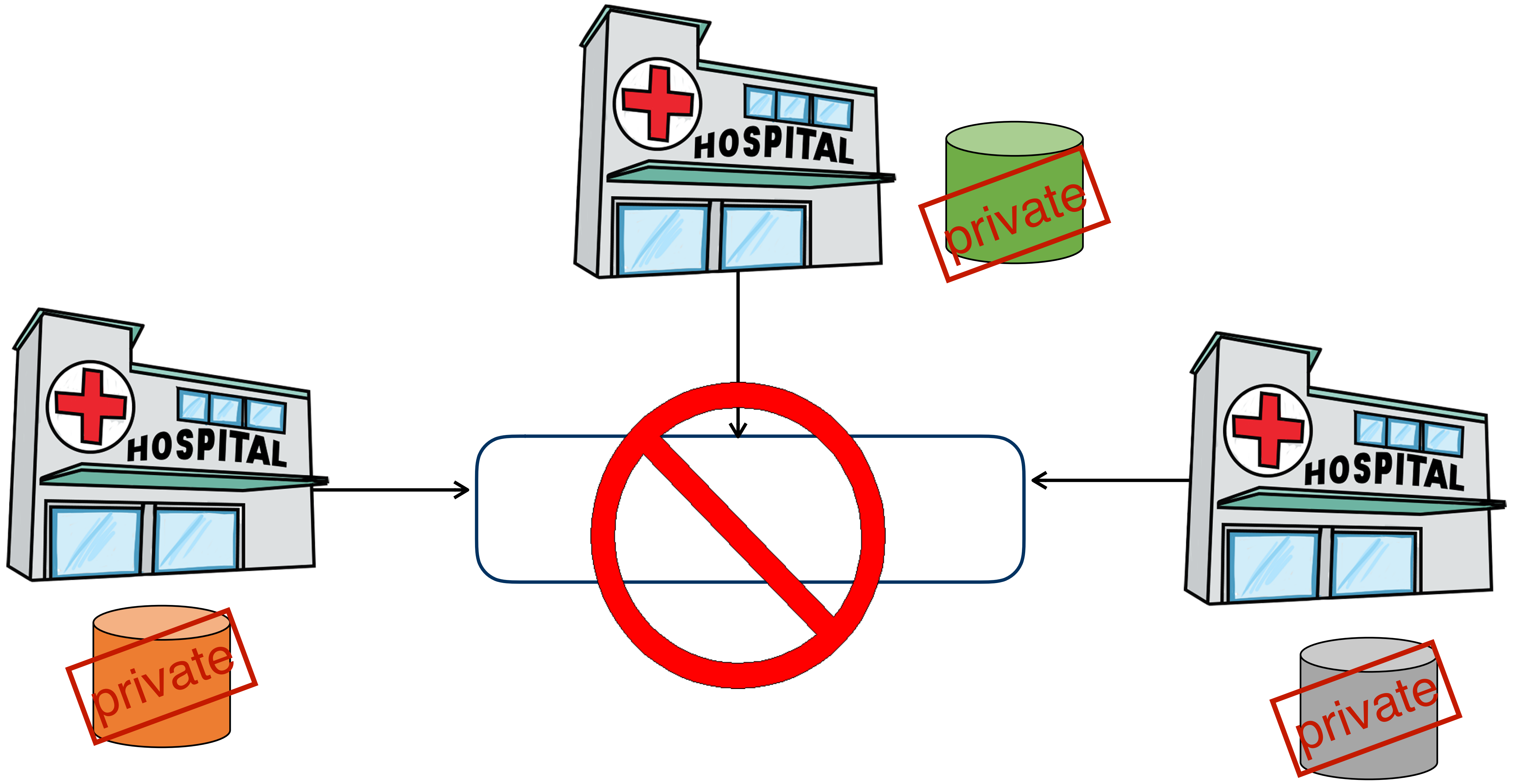
Example 1: Intel Cancer Cloud

- Hospitals have databases of patient data
- Some forms of cancer are rare and it would benefit to find **across hospitals** the patient with the closest form of cancer who was treated successfully
- Hospitals cannot share data due to privacy regulations



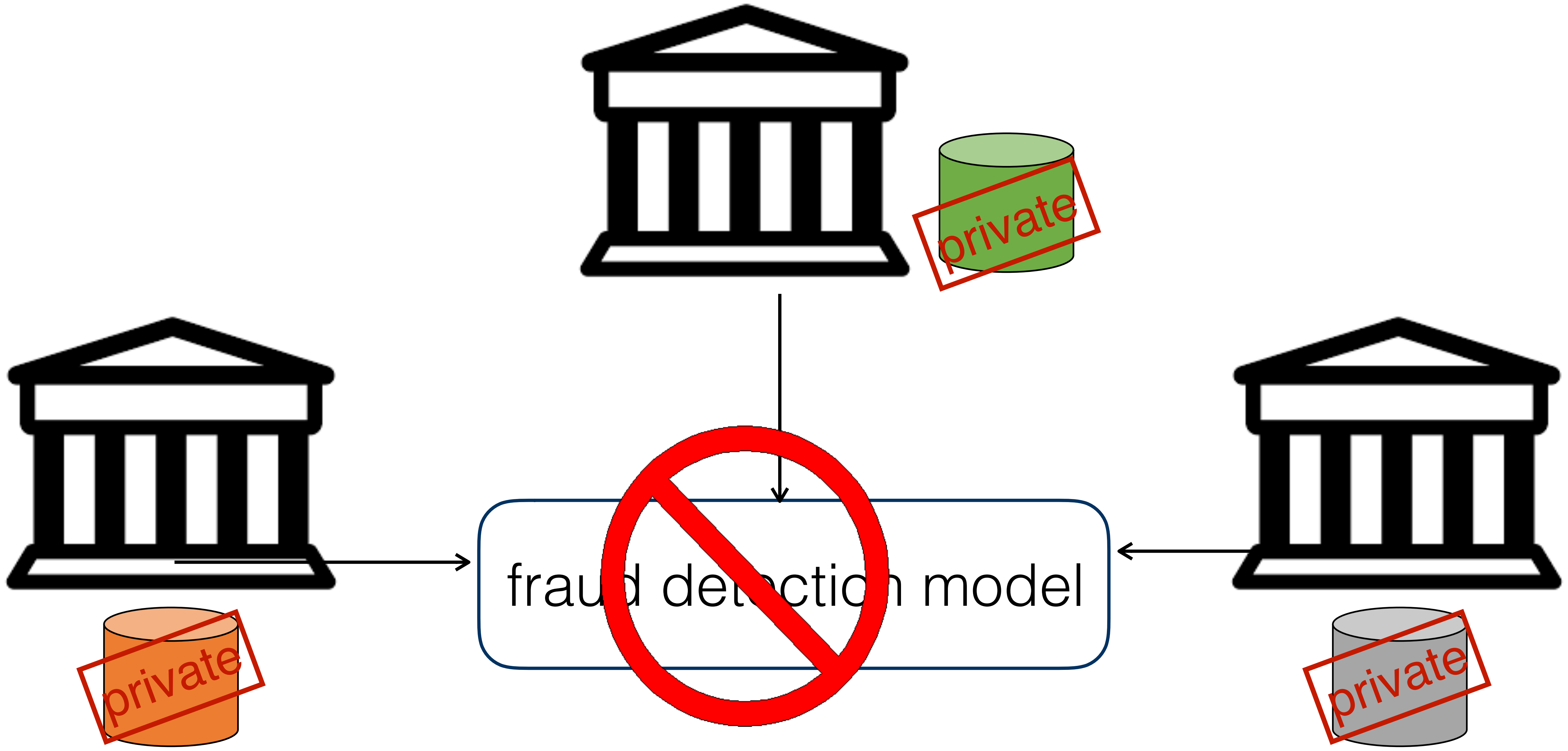
Example 2: Kaiser Permanente

- Kaiser would like to know the average body mass index in each zip code by averaging data from different hospitals
- Hospitals cannot share data due to privacy regulations



Example 2: Banks

- Some banks want to detect fraud or money laundering by analyzing the data from multiple banks because fraud tends to happen cross institutions
- Banks cannot share customer data due to business competition



Solution Insight

1. Organizations share **encrypted** data
2. Compute on the encrypted data, producing an encrypted result
3. **Jointly decrypt** the result and only the result

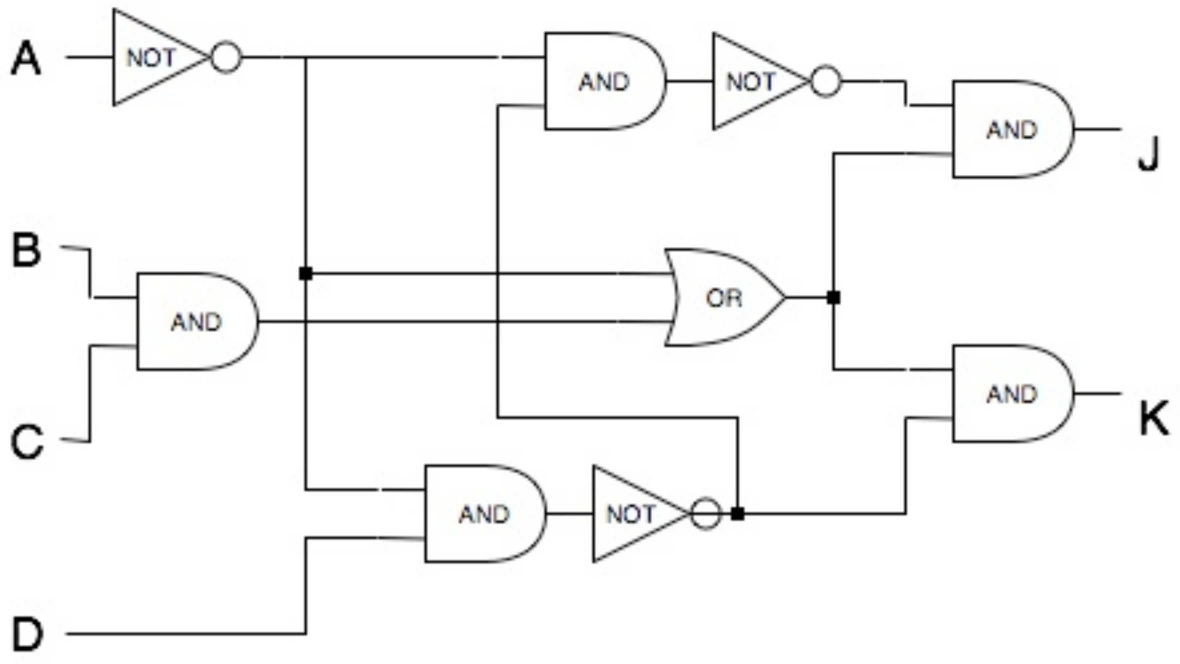
Morale: share the computation result and not the data

Solutions

- Secure multi-party computation (MPC) frameworks [SPDZ, Ag-MPC, Sharemind]
  - Can compute any function securely

SQL query

giant circuit with one input value for every bit of every row in a party's DB



MPC

encrypted circuit into which parties can feed their encrypted databases and outcomes the query result

Solutions

- Secure multi-party computation frameworks [SPDZ, Ag-MPC, Sharemind]
  - Can compute any function securely
- Too few attempts at combining DB techniques with cryptography
  - SMCQL: does locally as much of a query as possible and then invokes MPC

Still inefficient because it still uses generic MPC for the bulk of the computation!

Solutions

- Secure multi-party computation frameworks [SPDZ, Ag-MPC, Sharemind]
  - Can compute any function securely
- Few attempts at combining DB techniques with cryptography
  - SMCQL: does locally as much of a query as possible and then invokes MPC
  - Google's secure aggregation: only for summation

Practical Secure Aggregation for Privacy-Preserving Machine Learning

Keith Bonawitz*, Vladimir Ivanov*, Ben Kreuter*,
Antonio Marcedone†‡, H. Brendan McMahan*, Sarvar Patel*,
Daniel Ramage*, Aaron Segal*, and Karn Seth*
*{bonawitz,vlivan,benkreuter,mcmahan,
sarvar,dramage,asegal,karn}@google.com
Google, Mountain View, CA 94043
‡marcedone@cs.cornell.edu

Google AI Blog

The latest news from Google AI

Federated Learning: Collaborative Machine Learning without Centralized Training Data

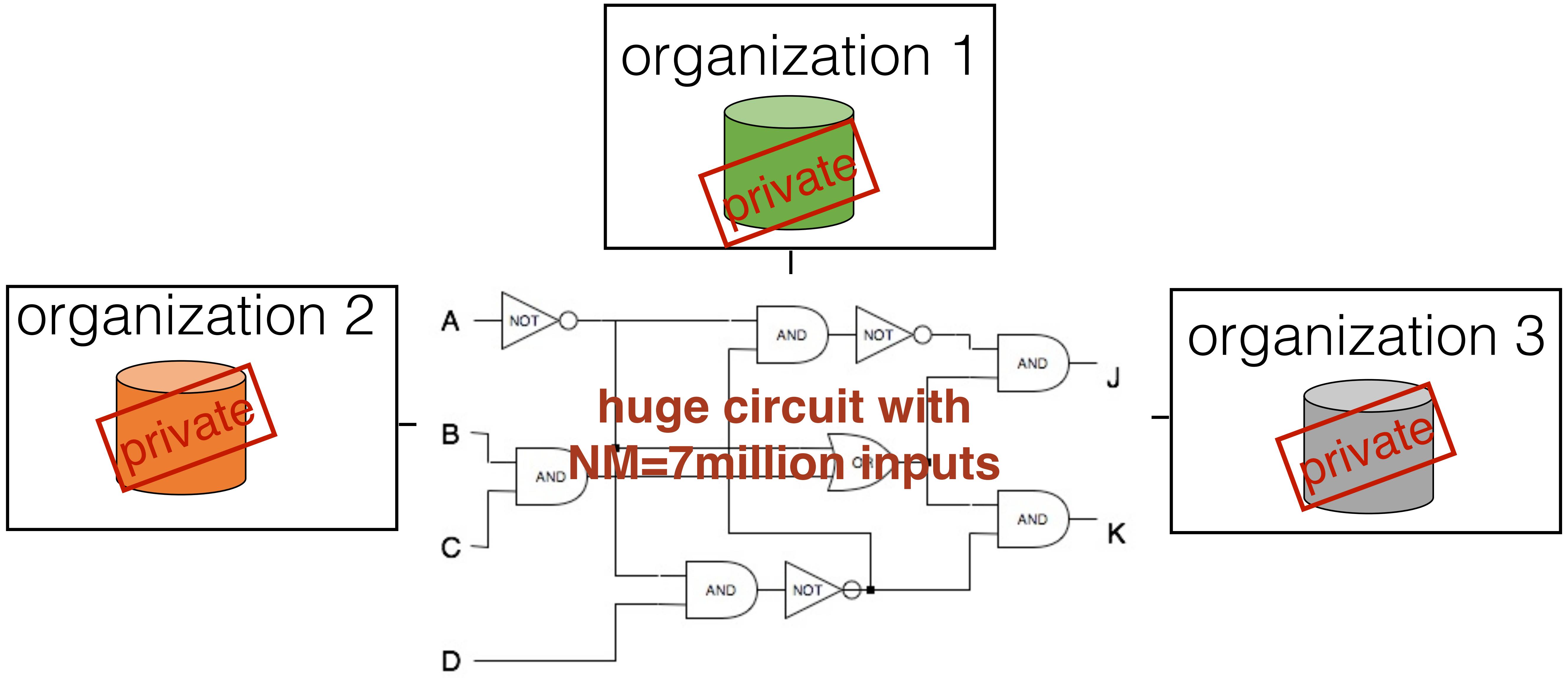
Thursday, April 6, 2017

Posted by Brendan McMahan and Daniel Ramage, Research Scientists

Recall Kaiser example

SELECT sum(BMI) FROM FederatedTable GROUP BY zipcode

- Suppose each of the N=7 organizations has M records, for M = 1 million
- With black-box MPC:

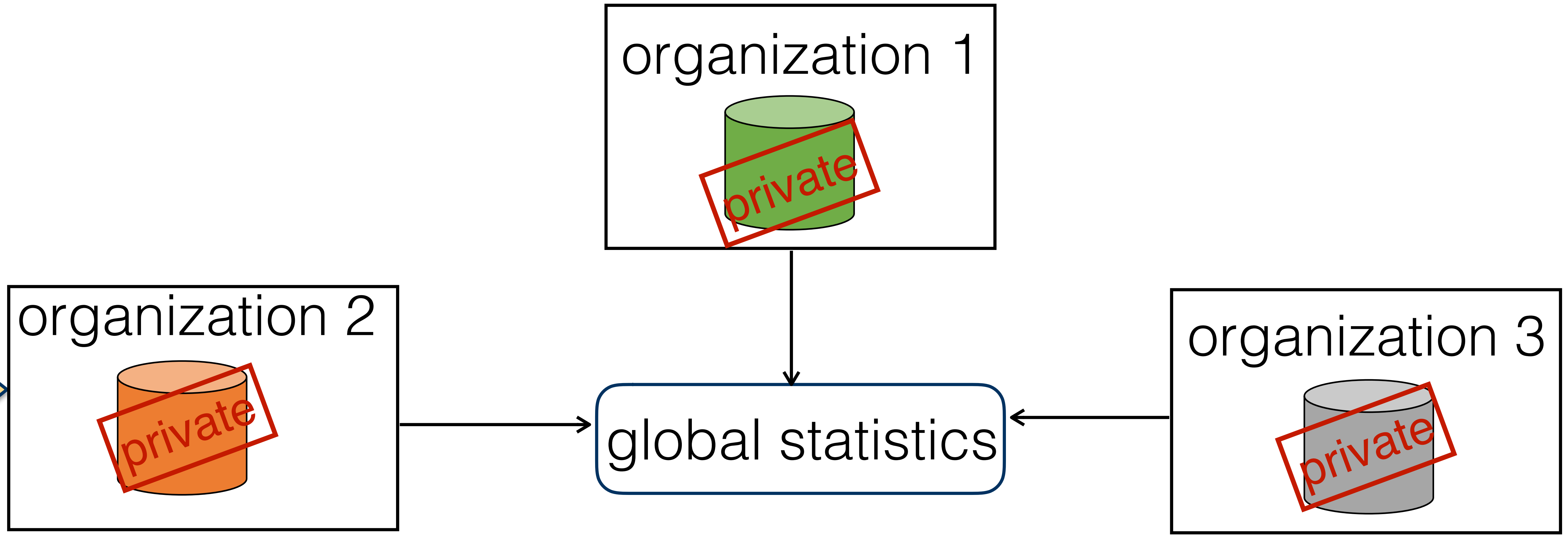


Open-box approach

Each organization runs the query locally and obtains a table of Z values

Z = ~2500 zip codes in California.

Encrypt data with Paillier, a fast encryption scheme enabling summation.



Paillier(sum1)*Paillier(sum2)*Paillier(sum3) = Paillier(sum1+sum2+sum3)

Use black-box MPC to jointly decrypt

2500 resulting encrypted counts inputs to MPC vs. **Fully black-box approach 7million inputs**

Morale: the open-box approach yields much better performance

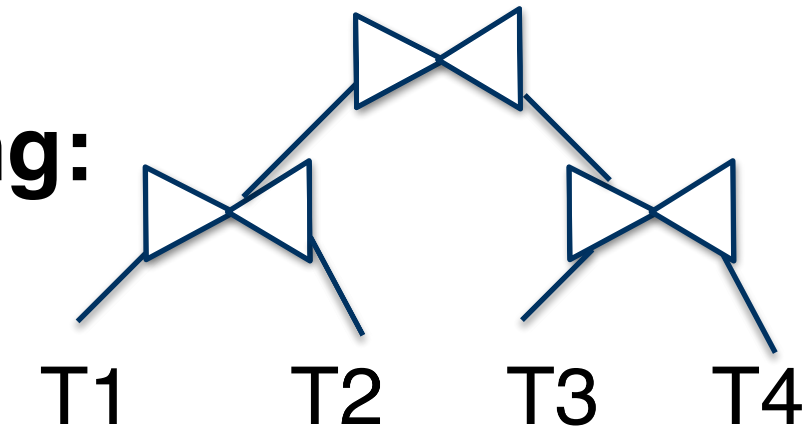
Solutions

- Secure multi-party computation frameworks [SPDZ, Ag-MPC, Sharemind]
  - Can compute any function securely
- Few attempts at combining DB techniques with cryptography
  - SMCQL: does locally as much of a query as possible and then invokes MPC
  - Google's secure aggregation: **only for summation**

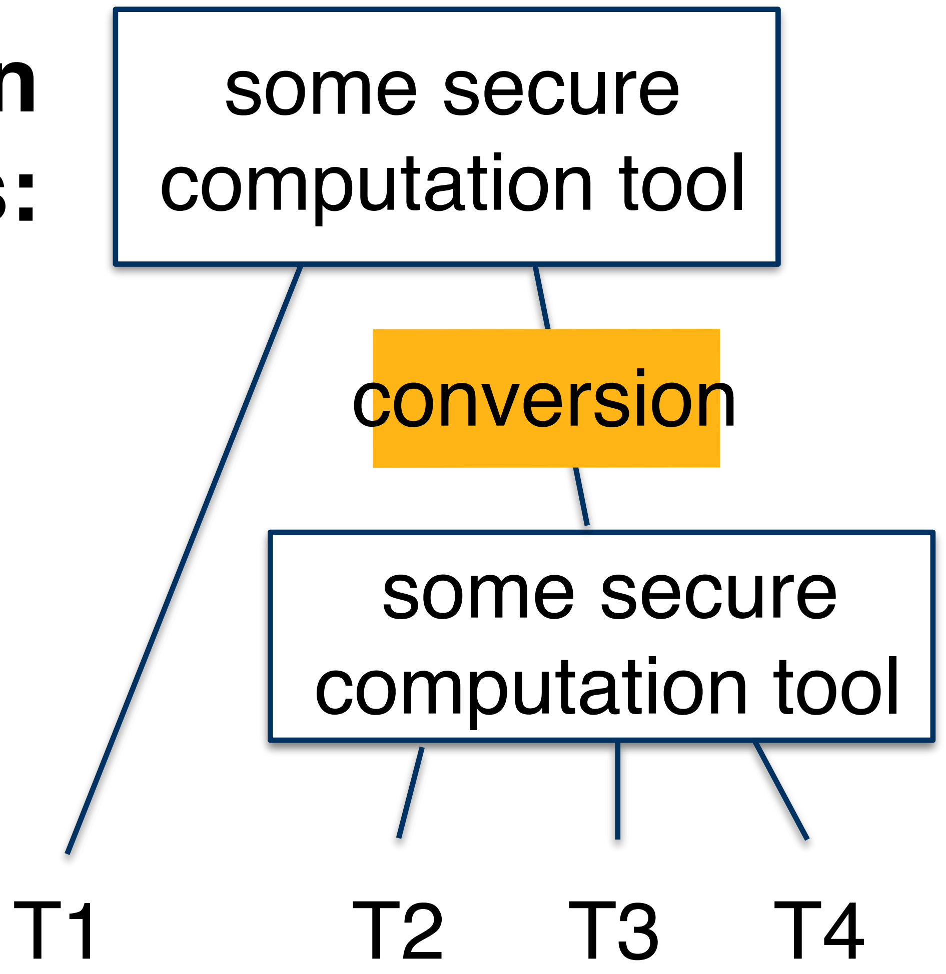
Lots of interesting database questions

Standard query planning and cost modeling no longer work

standard planning:



secure computation possible plans:



exponential possibilities!

of unpredictable costs!

Crypto tools do not have well understood performance (which depends on # of parties, amount of data, network locations)

Different crypto tools do not connect with each other so we need a conversion, whose cost depends also on the pair of tools

A very rewarding problem

- Can have societal impact:
  - better medical research
  - detect fraud easier
  - better user profiling

Security typically is an invisible property that costs; this time it can enable functionality not possible without

Decentralized security: blockchain and ledgers

- Scalable consensus
- Scalable authenticated data structures

1. How to achieve scalable consensus with untrusted and unknown users?

- Bitcoin’s proof of work is far too expensive

Proof of work, or proof of waste?

Bitcoin and the energy usage dilemma

If you’re moderately well versed in the blockchain ecosystem, then you would have by now come across the energy consumption debate that is taking place

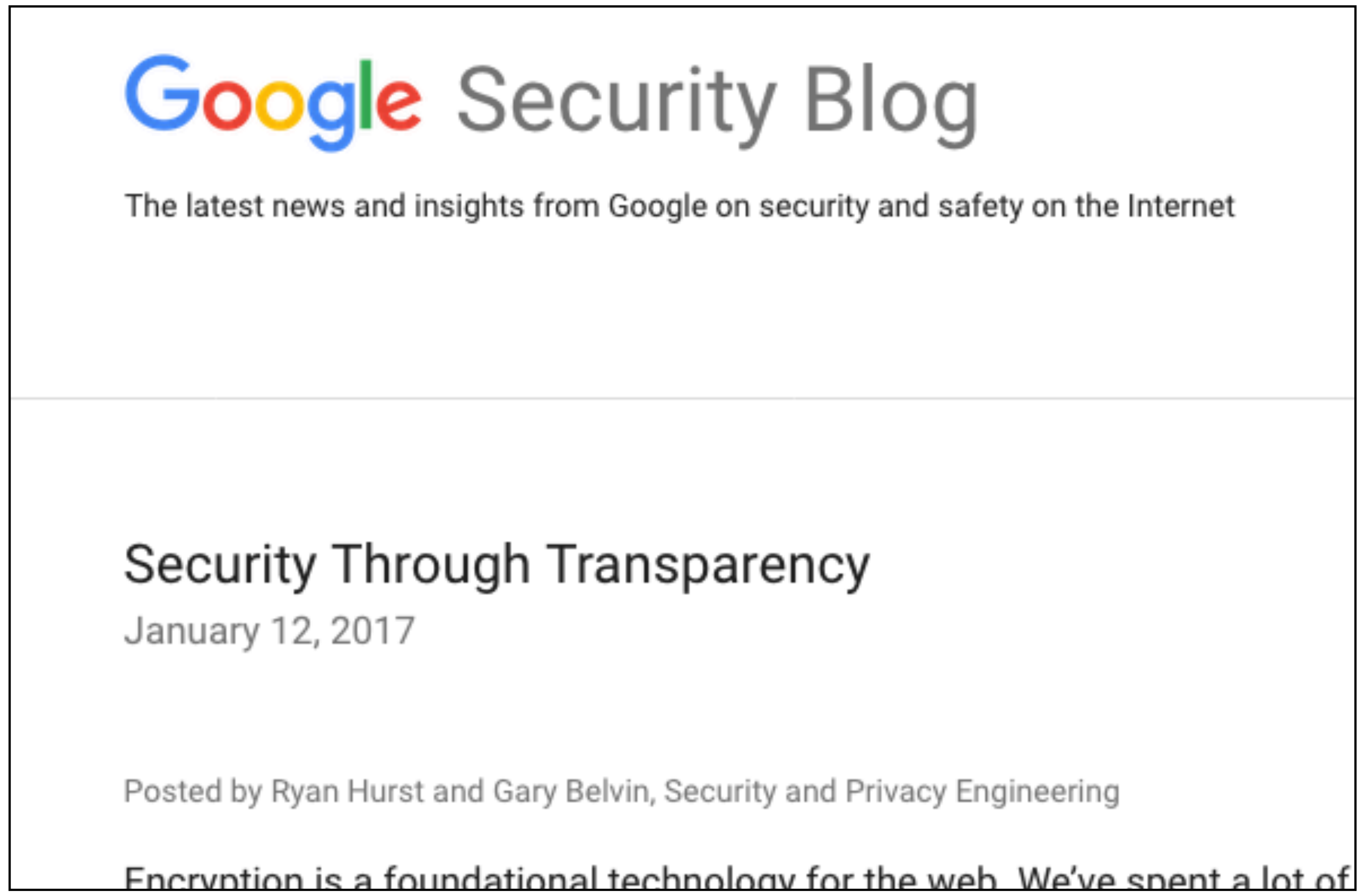
Towards the end of last year, the Bitcoin network was running on enough electricity to power more than 20 European nations. More recently, an article was run with the headline *Bitcoin mining now accounts for almost one percent of the world’s energy consumption*. Under a certain light, this is certainly true.

Alternatives

- Proof of stake (e.g. Algorand) assume coins are distributed among many entities each having a small percentage of wealth, but we know that is not true due to mining pools?
- Can re-use any principles from the line of work on BFT?

2. How to design scalable authenticated data structures?

A new generation of ledgers/blockchains are rising: transparency logs


Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Security Through Transparency

January 12, 2017

Posted by Ryan Hurst and Gary Belvin, Security and Privacy Engineering

Encryption is a foundational technology for the web. We've spent a lot of

- Promising idea: one powerful party (e.g., Google) hosts them, but anyone can verify
- Avoids the resource waste in Bitcoin due to consensus and everyone storing a copy of the blockchain
- Google is committed to giving good performance and scalability, but is not trusted for security

Certificate Transparency

Key Transparency

Transparency logs

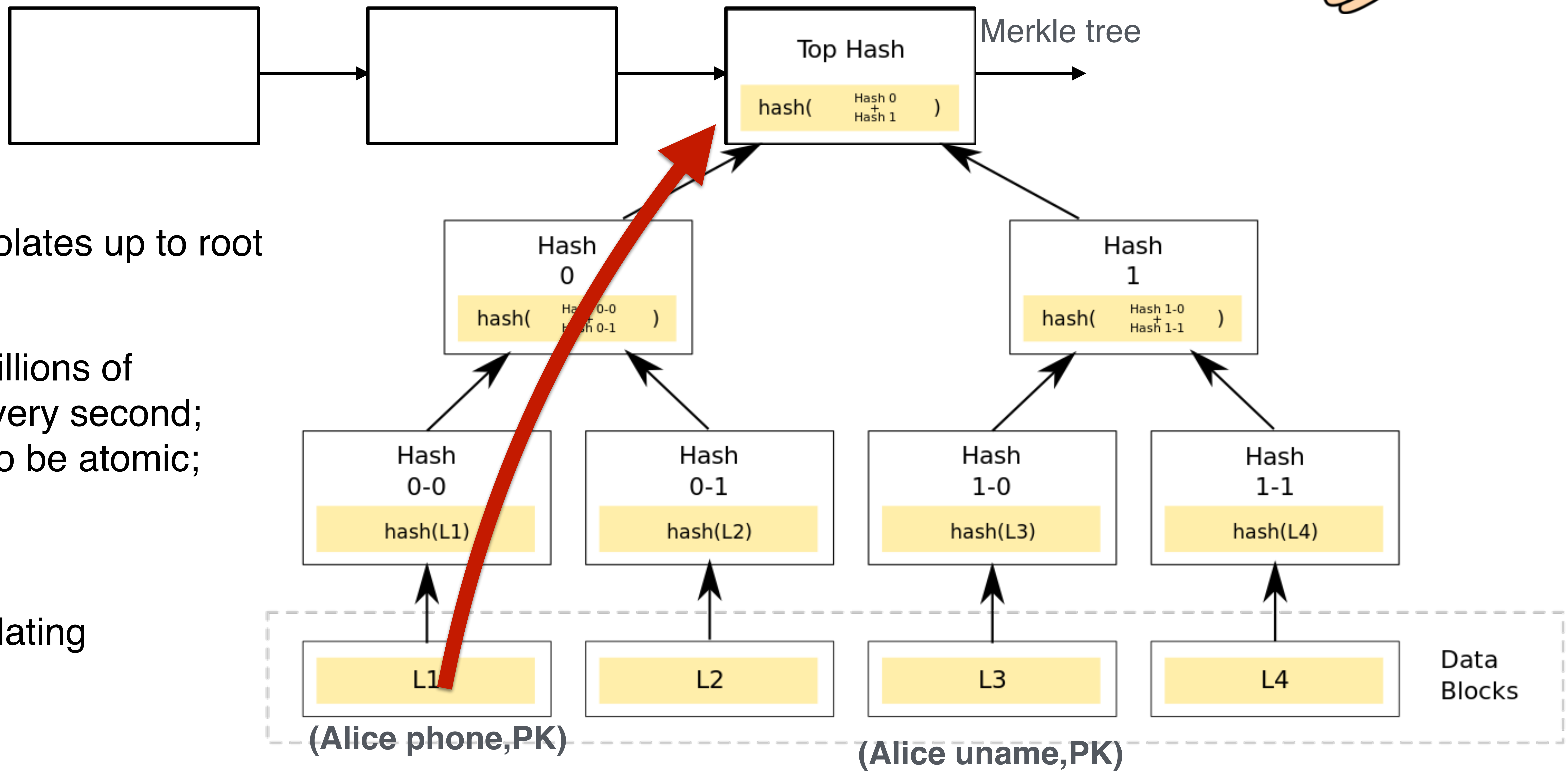
- Based on authenticated data structures
- Ensure that the untrusted server cannot equivocate about the state of a data structure (e.g., log or tree).

What is challenging?

scalable hash chain verification: [DIZK,UsenixSec'2018]

Key transparency

Blockchain



One change in a leaf percolates up to root

Scalability goal: tree has billions of records; recompute tree every second; concurrent updates need to be atomic; need fault tolerance

Cannot lock root when updating

How to design high-throughput transactional semantics for authenticated data structures?

Side channel attacks

A significant side channels leakage in DBs: Access patterns

- An attacker can see the location in the database a read touches
- Examples:
  - DNS privacy, which records you look up
  - Your Google search query
  - Which public keys you look up in KeyTransparency
  - Which records you look up in an encrypted medical database, coupled with frequency attacks

Existing solutions are expensive

- ObliVM 10^6 performance overhead
- Our work on Opaque [NSDI18] and Oblix [Oakland/IEEE S&P 18] reduce overhead to 20x-40x
    - new query planning: oblivious query planning
- Can we protect against side channels more efficiently?

In summary

Very exciting time for databases and crypto

Three key problem areas/trends I see:

1. Secure federated analytics
2. Decentralized security
   - Scalable consensus
   - Scalable authenticated data structures
3. Side-channel prevention

I think we have great prospects of solving them if we collaborate between DB and crypto in an open-box manner

Thanks!