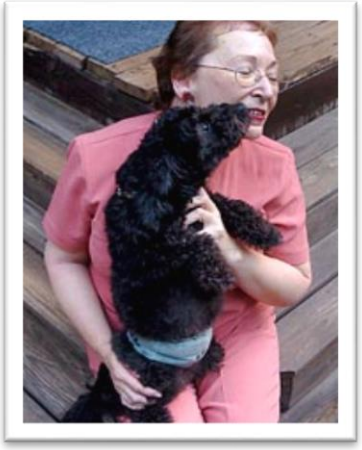
DJoin: Differential Private Join Queries over Distributed Databases

- Written by Arjun Narayan & Andreas Haeberlen
- Presented by Suyash Rathi

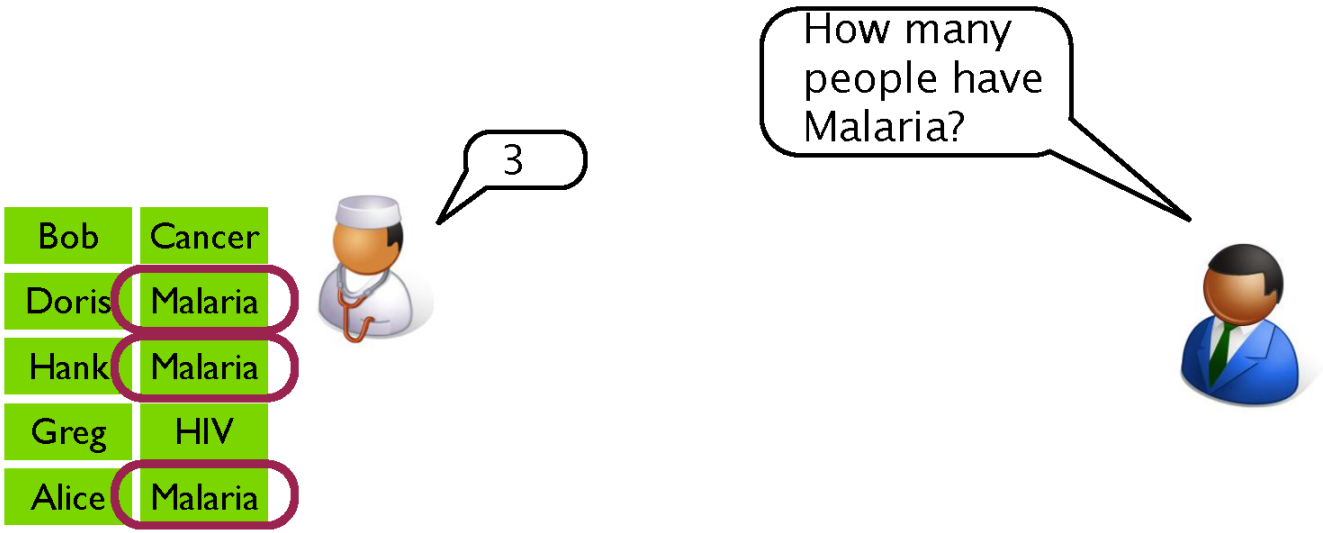
AOL Searcher No. 4417749

- AOL released 20 million web search queries – Research purposes
- Identity is removed and is replaced by a searcher number
- Searches by Searcher No. 4417749
  - “numb fingers”
  - “60 single men”
  - “dog that urinates on everything”
  - “landscapers in Lilburn, Ga,”
  - Search queries for several people with last name “Arnold”



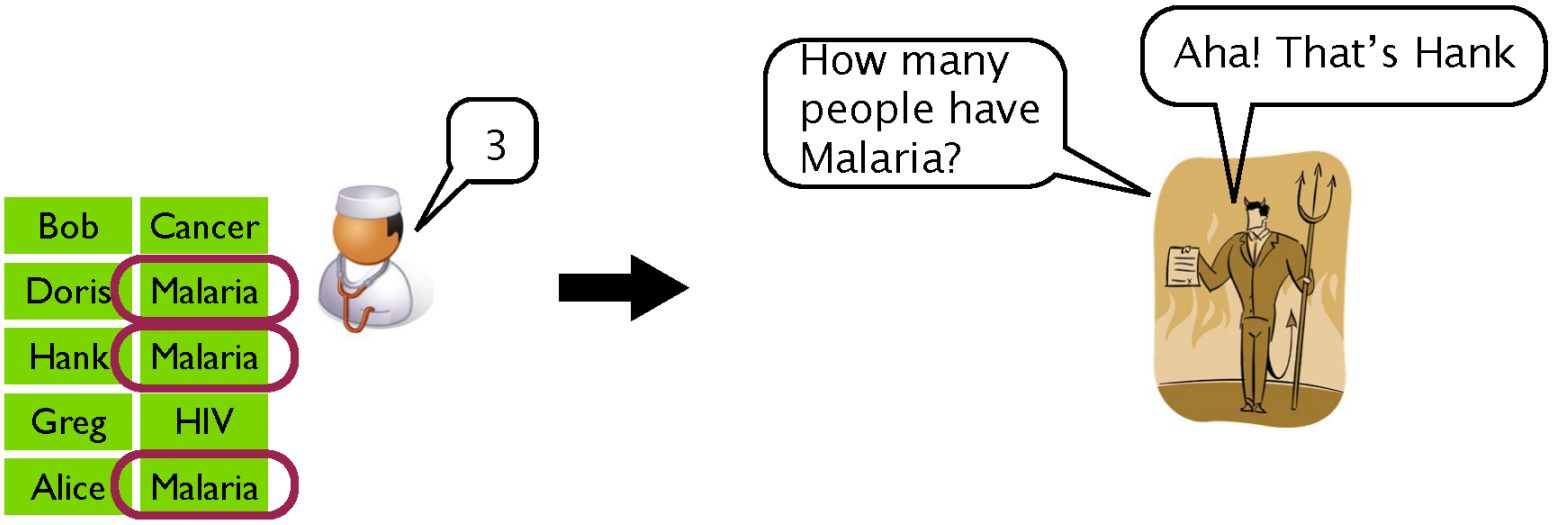
- It was easy to trail these searches to find Thelma Arnold.
- Thelma Arnold's identity was betrayed by AOL records of her Web searches.
- In this case even her poor dog Dudley’s problem was revealed.

Background: Differential Privacy



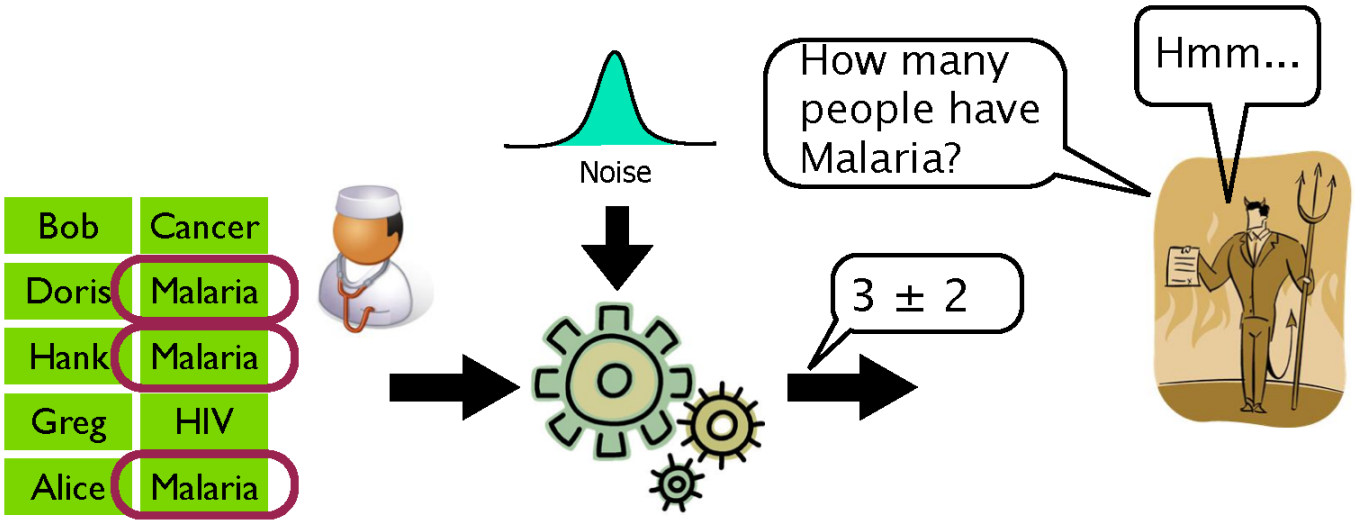
- Typically answers queries about aggregates.
- But to protect privacy, we need more...

Background: Differential Privacy



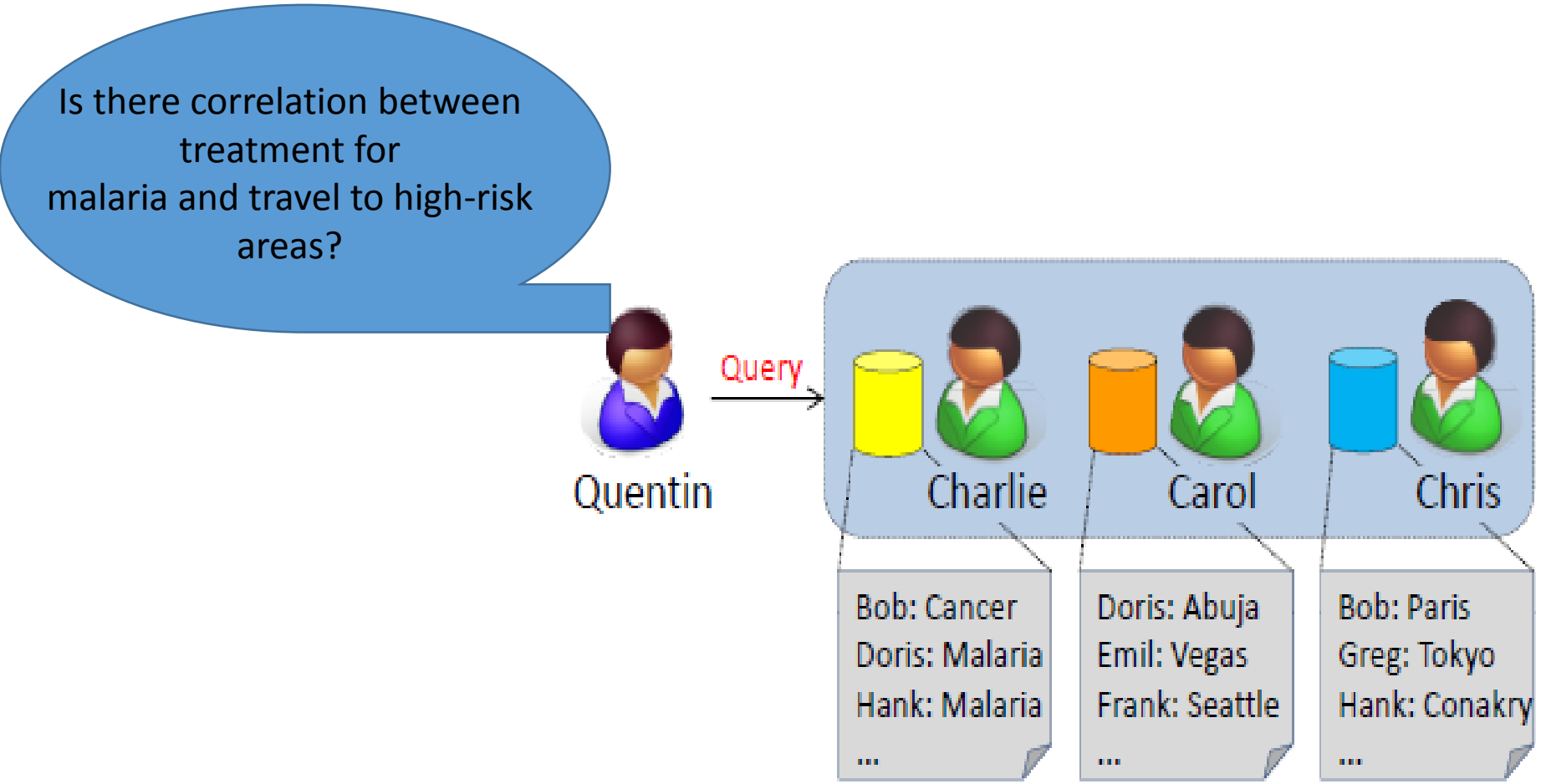
- Suppose our researcher's credentials have been stolen.
  - And the thief has certain outside information.
- We need guarantees even when the querier has outside information!
  - "I know that 2 other people have Malaria, but what about Hank?"

Background: Differential Privacy



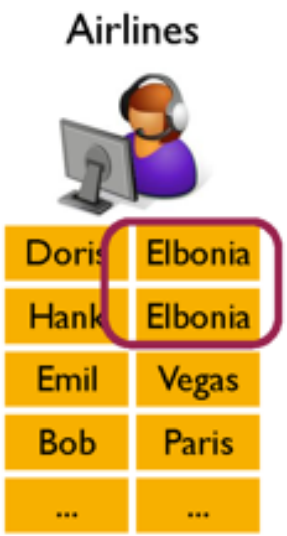
- We need guarantees even when the querier has outside information.
  - "I know that 2 other people have Malaria, but what about Hank?"
- Solution: Differential Privacy adds noise to the answer.
  - Effect: Bounds how much more certain the adversary can be.
- Lots of mathematical detail omitted.
  - Dwork [ICALP 2006]

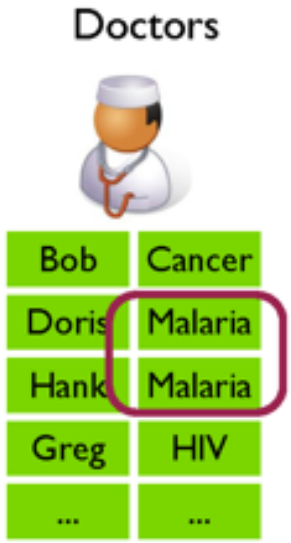
Motivation Scenario



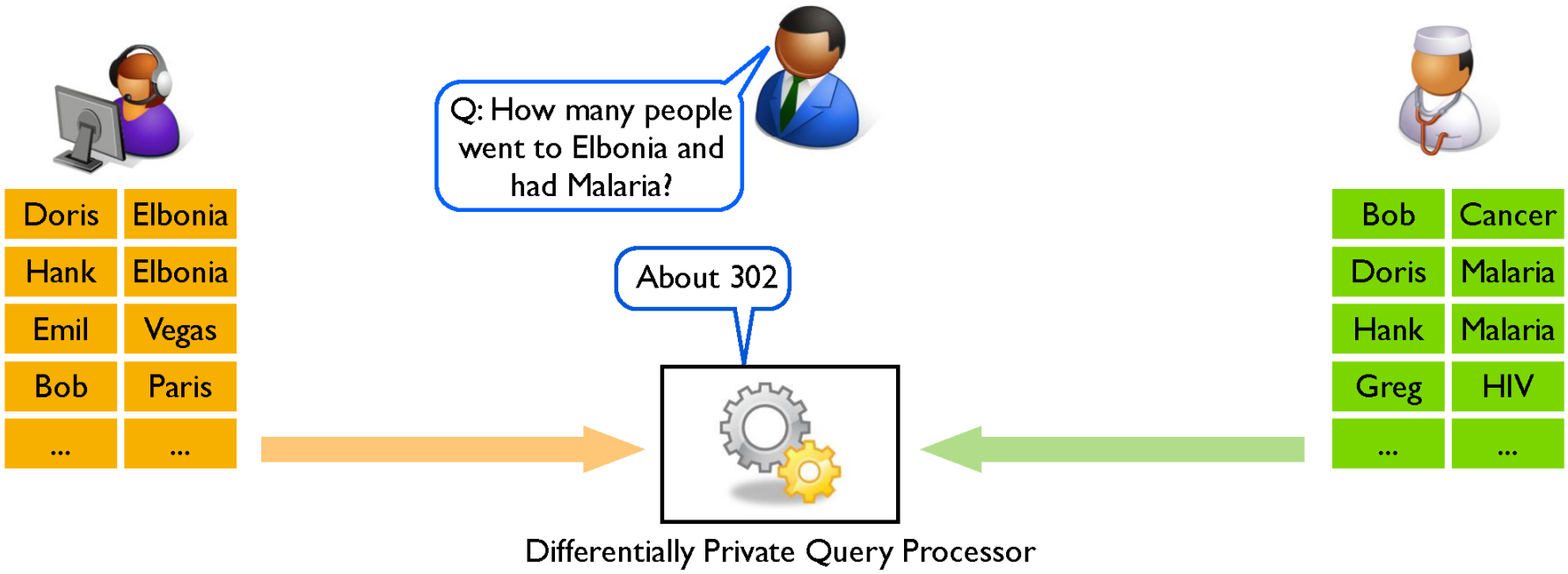
Motivation

"Is there a Malaria epidemic in Elbonia?"





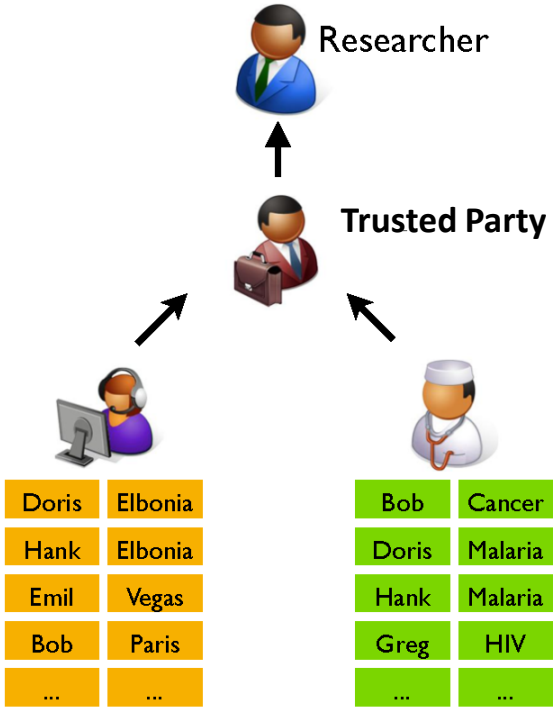
Differential Privacy



- Offers strong, provable privacy guarantees:
  - By giving an upper bound on what an adversary can learn
  - While still allowing us to answer queries safely

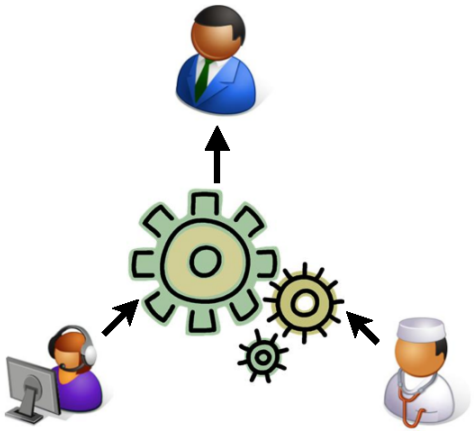
Possible Solutions

Idea 1: Give all the data to a trusted party



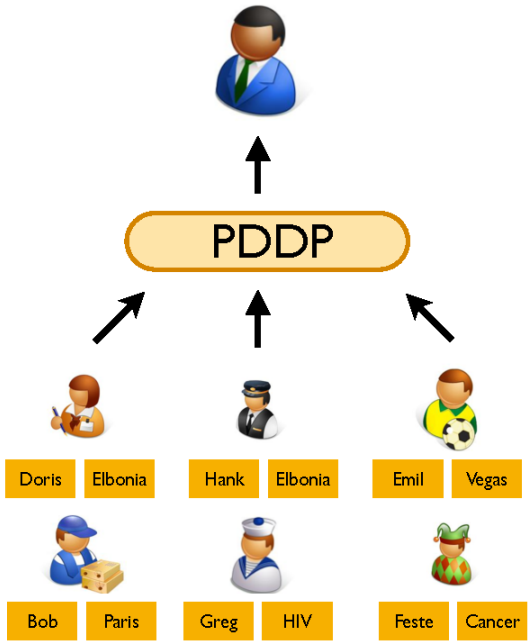
What if we don't have a trusted party?

Idea 2: Use Secure Multiparty Computation (SMC)



It will take years.

Idea 3: Use PDDP [NSDI 2012]



Handles only certain types of queries, not including JOINs

Queries with Joins

SELECT COUNT(X) FROM HOSPITAL JOIN AIRLINE
WHERE Destination= "Elbonia" AND Diagnosis = "Malaria"

Doris	Elbonia
Hank	Elbonia
Emil	Vegas
Bob	Paris
Charlie	Elbonia

Who went to Elbonia?

Doris
Hank
Charlie

Doris
Hank
Charlie

∩

Doris
Hank
Alice

Doris
Hank

2

Bob	Cancer
Doris	Malaria
Hank	Malaria
Greg	HIV
Alice	Malaria

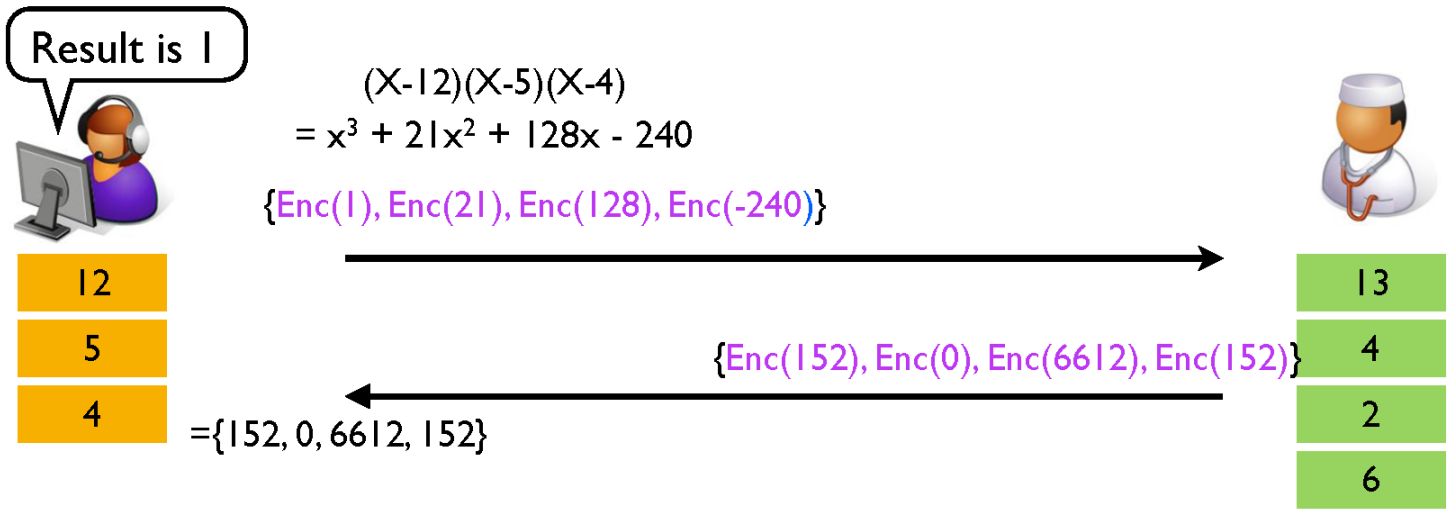
Doris
Hank
Alice

Who had Malaria?

- Challenge: How can we support Joins?
- Key Insight: Not all joins are full cross products.
  - Morally this query is a set intersection.

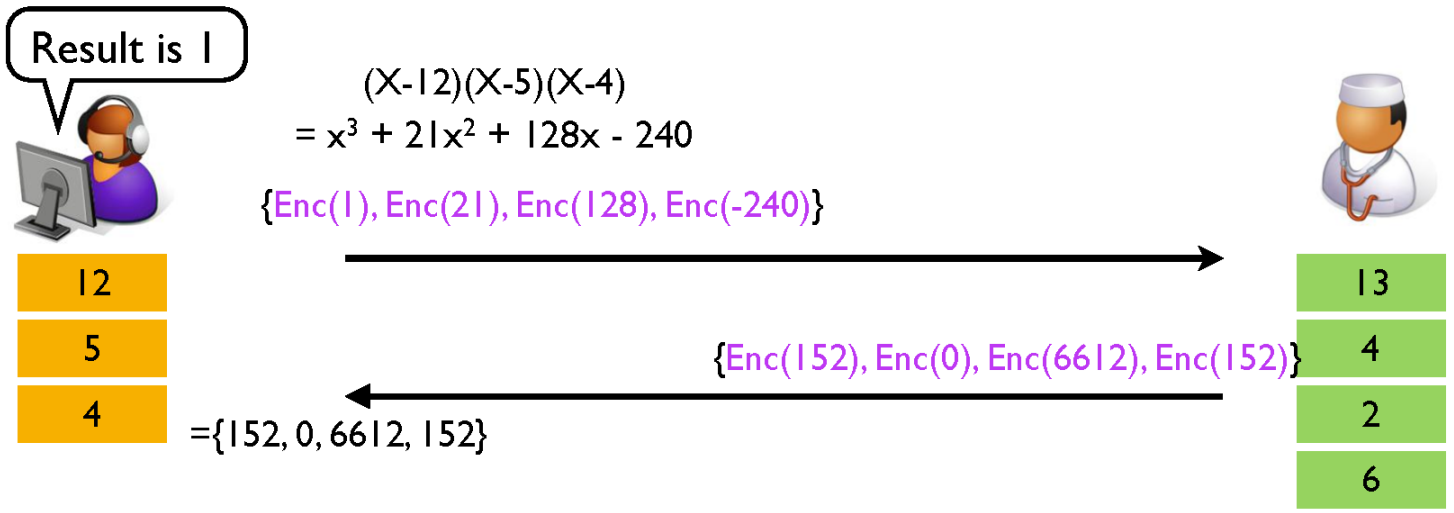
Literature

PSI-CA without Differential Privacy



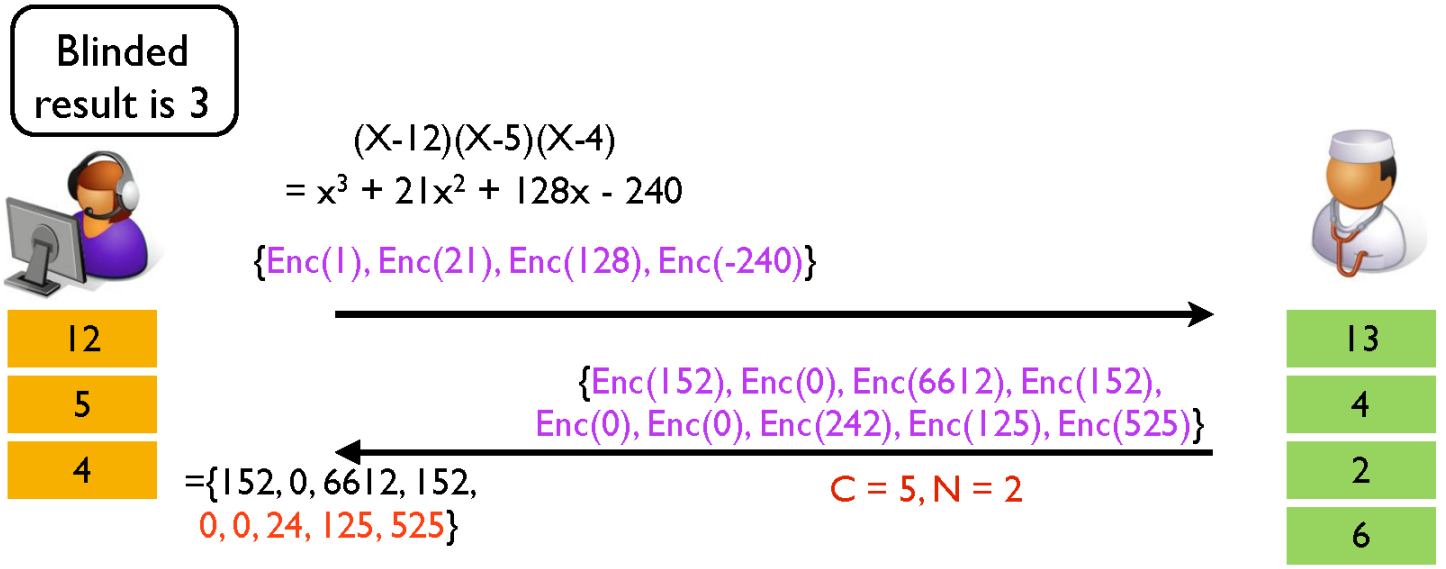
- Protocol from Freedman et al [Eurocrypt 2004]
- The airline have two sets A and B and want to jointly compute $|A \cap B|$.
- The airline makes a polynomial P whose roots are the elements of A.
- The airline encrypts the coefficients of P and sends them to the doctor.
- The doctor evaluates $P(B_i)$ for each element in B.
- The doctor returns the encrypted evaluations to the airline.
- The airline decrypts it and counts the number of zeroes.

PSI-CA without Differential Privacy



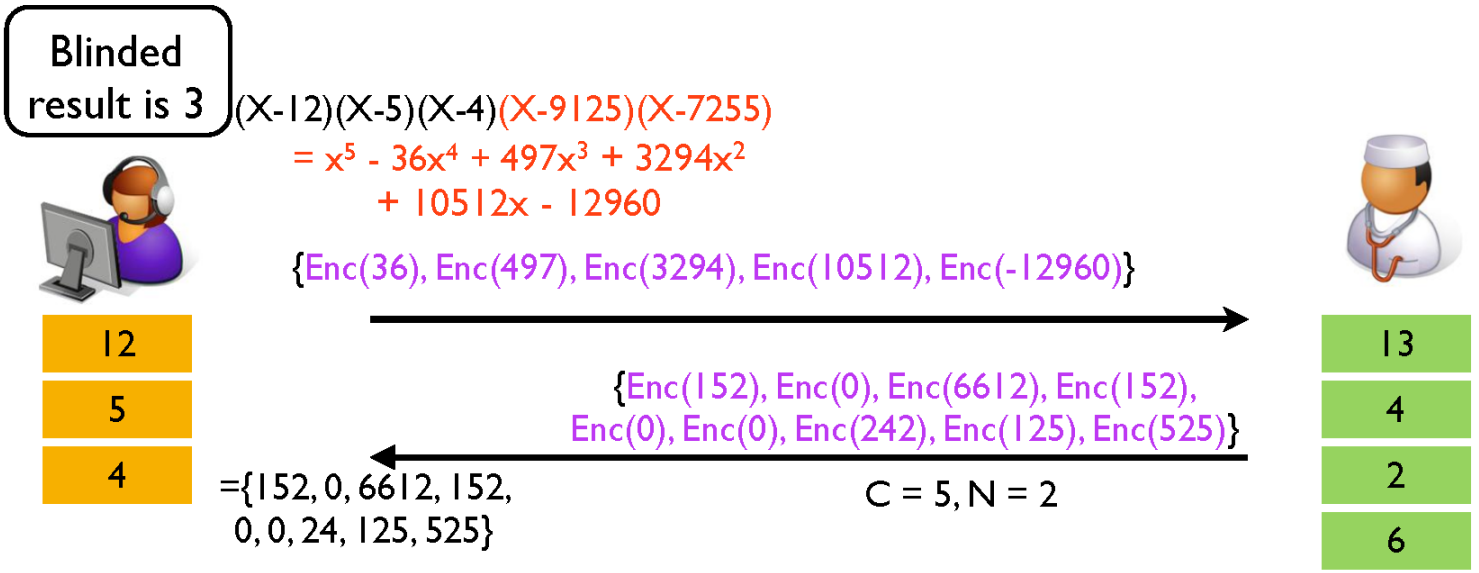
- This protocol is **not differentially private** because:
  - 1. The first party learns the exact size of the intersection.
  - 2. Both parties learn the exact size of the other database.

BN-PSI-CA *with* Differential Privacy



- Challenge 1: The first party learns the exact size of the intersection.
- Idea 1: Party 2 adds or removes some zeros to the result.
  - Problem: We cannot remove zeros because they are encrypted.
    - Remember, differentially private noise is two sided: it could be negative.
  - Solution: First add a fixed block of C zeroes.
    - Now add N noised zeroes, for a total of C-N if N is negative.

BN-PSI-CA *with* Differential Privacy



- Challenge 2: Both parties learn the exact size of the other database.
- Idea 2: Party 1 adds some random elements to the set.
  - This doesn't affect the result.
  - Similar to the solution to Challenge 1.

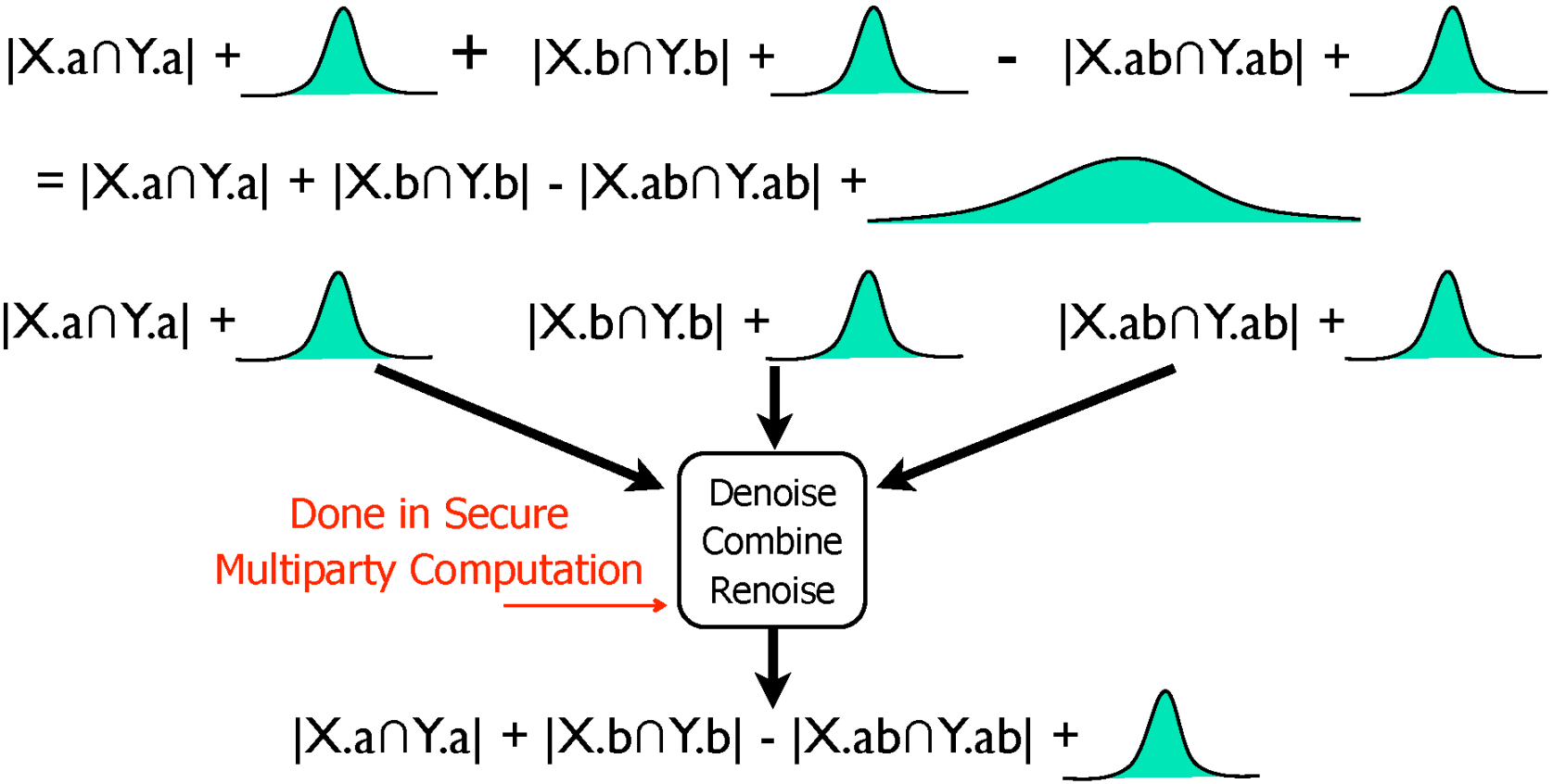
Denoise-Combine-Renoise

Some queries need more than one BN-PSI-CA e.g.,

SELECT |X.a| FROM X,Y WHERE X.a=Y.a OR X.b=Y.b

Need to compute |X.a∩Y.a| + |X.b∩Y.b| - |X.ab∩Y.ab|

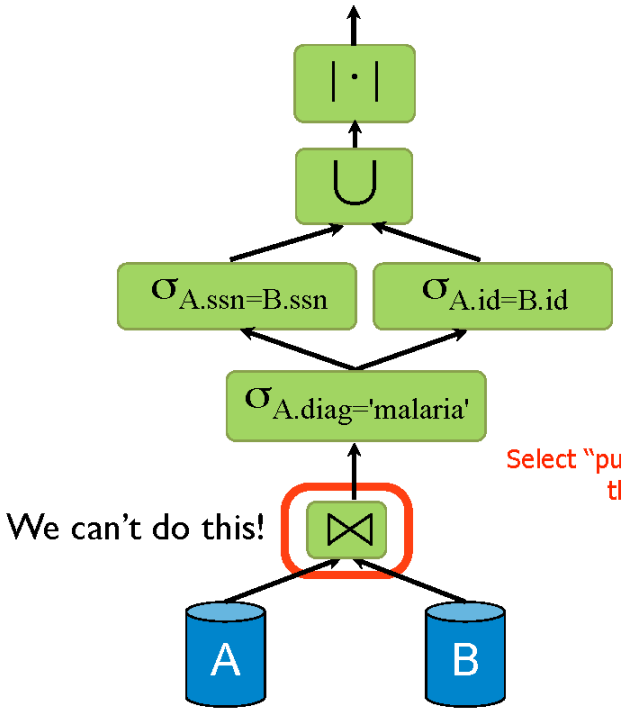
Result of each BN-PSI-CA



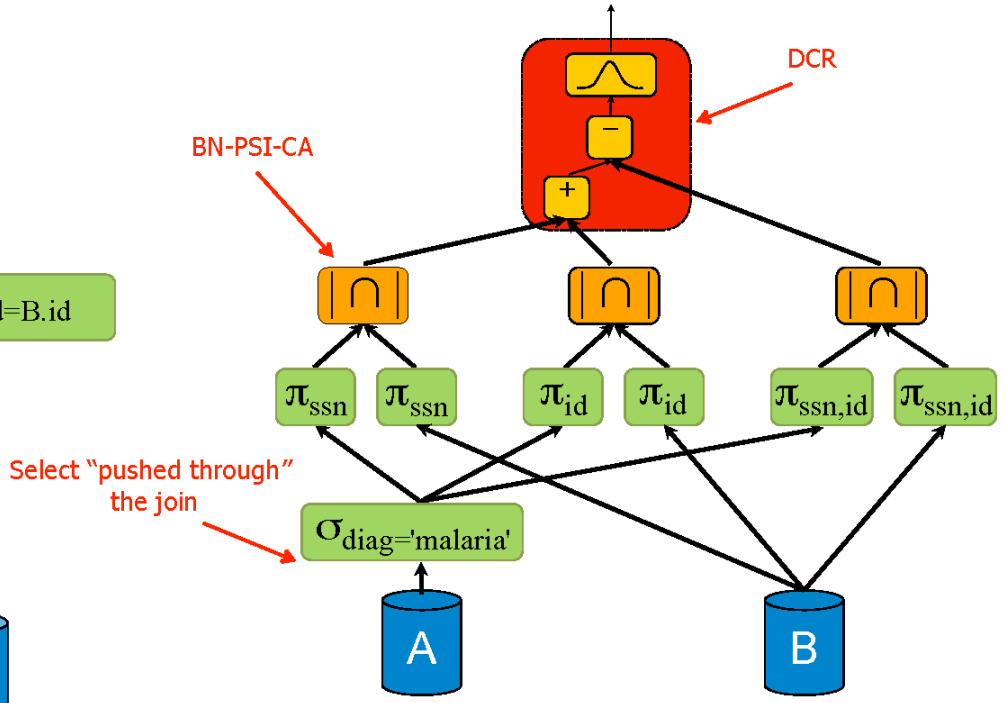
System

Query Rewriting

SELECT NOISY COUNT(A.ssn) FROM A,B WHERE
(A.ssn=B.ssn OR A.id=B.id) AND A.diagnosis= 'malaria'



Query execution with a centralized database.



Differentially private query execution:
with only local operations, set intersections and DCR.

Limitations & Restrictions

- We cannot always re-write queries:
  - One reason could be it does not satisfy differential privacy
  - Another reason could be if there is no optimal way to encode them.

```
SELECT COUNT(A.id) FROM A,B,C
WHERE ((A.x*B.y)<C.z)
```

Also substring queries spreading across multiple data sources would not work.

Privacy Budget

- Each server locally has a privacy budget
- It is the upper bound of information for a user to be revealed
- Each server will have a budget which it can spend.
- So each time a query is processed, its privacy cost is deducted from the budget.

Example Queries

Query	BN-PSI-CAs
1. SELECT NOISY COUNT(A.x) FROM A,B WHERE A.x=B.y	1
2. SELECT NOISY COUNT(A.x) FROM A,B WHERE A.x=B.x AND (A.y!=B.y)	2
3. SELECT NOISY COUNT(A.x) FROM A,B WHERE A.x=B.y AND (A.z="x" OR B.p="y")	2
4. SELECT NOISY COUNT(A.x) FROM A,B WHERE A.x=B.x OR A.y=B.y	3
5. SELECT NOISY COUNT(A.x) FROM A,B WHERE A.x LIKE "%xyz%" AND A.w=B.w AND (B.y+B.z>10) AND (A.y>B.y)	8

- SQL-like syntax
- Full SQL for local operations
- Number of set intersections depends on query complexity
  - Some operations (inequalities) are much more expensive

Summary

- DJoin: A differentially private query processor for distributed databases
- First practical system that supports JOINs (with some restrictions).
- Based on two novel primitives:
  - BN-PSI-CA: Blinded Private Set Intersection Cardinality
  - DCR: Denoise-Combine-Renoise
- Not fast enough for interactive use, but may be sufficient for offline data analysis.

References

- DJoin: Differentially Private Join Queries over Distributed Databases OSDI ’12
- M. Barbaro and T. Zeller. A face is exposed for AOL searcher No. 4417749. *The New York Times*, Aug. 2006. http://nytimes.com/2006/08/09/technology/09aol.html
- M. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Proc. EUROCRYPT*, May 2004.
- C. Dwork. Differential privacy. In *Proc. ICALP*, July 2006.
- F. McSherry. Privacy integrated queries. In *Proc. SIGMOD*, June 2009.
- A. Yao. Protocols for secure computations (extended abstract). In *Proc. FOCS*, Nov. 1982.
- L. Kissner and D. Song. Privacy-preserving set operations. In *Proc. CRYPTO*, Aug. 2005.
- J. Vaidya and C. Clifton. Secure set intersection cardinality with application to association rule mining. *Journal of Computer Security*, 13(4):593–622, Nov. 2005.

Thank You!