DJoin: Differentially Private Join Queries over Distributed Databases

Arjun Narayan Andreas Haeberlen

University of Pennsylvania

Motivation

Researcher

"Is there a Malaria epidemic in Elbonia?"

Airlines

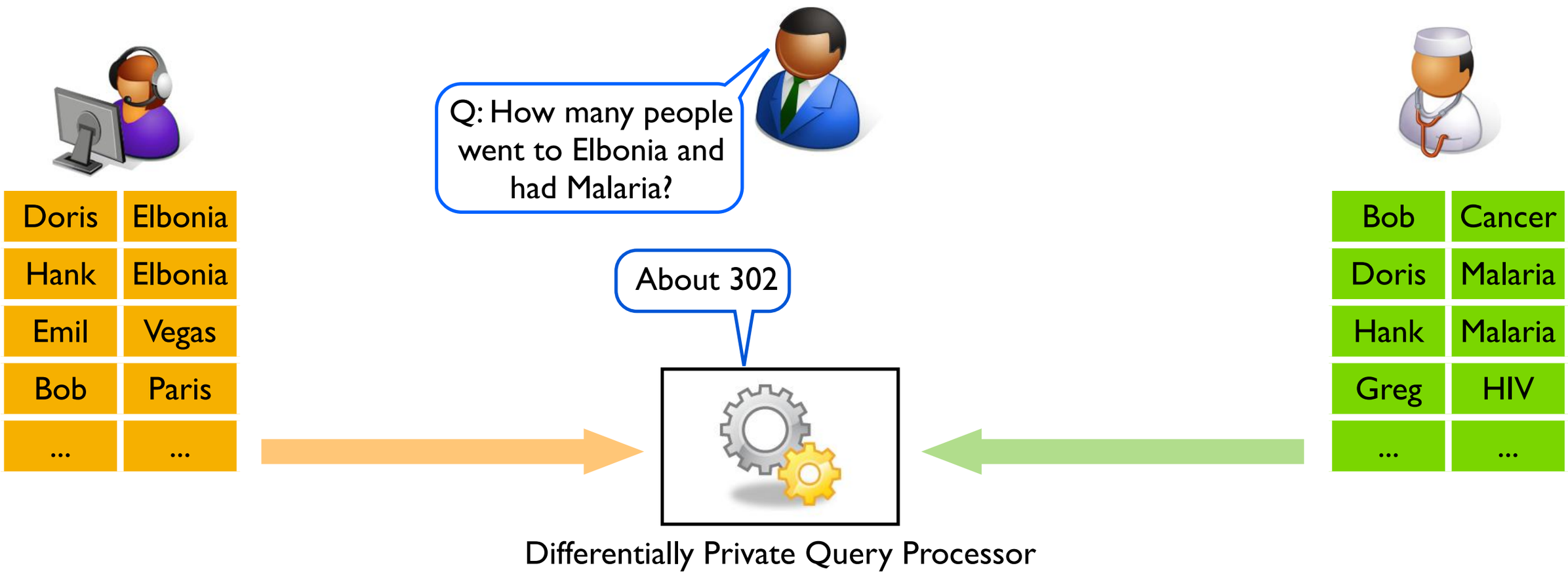
Doris	Elbonia
Hank	Elbonia
Emil	Vegas
Bob	Paris
...	...

Doctors

Bob	Cancer
Doris	Malaria
Hank	Malaria
Greg	HIV
...	...

What could we do with all this data?

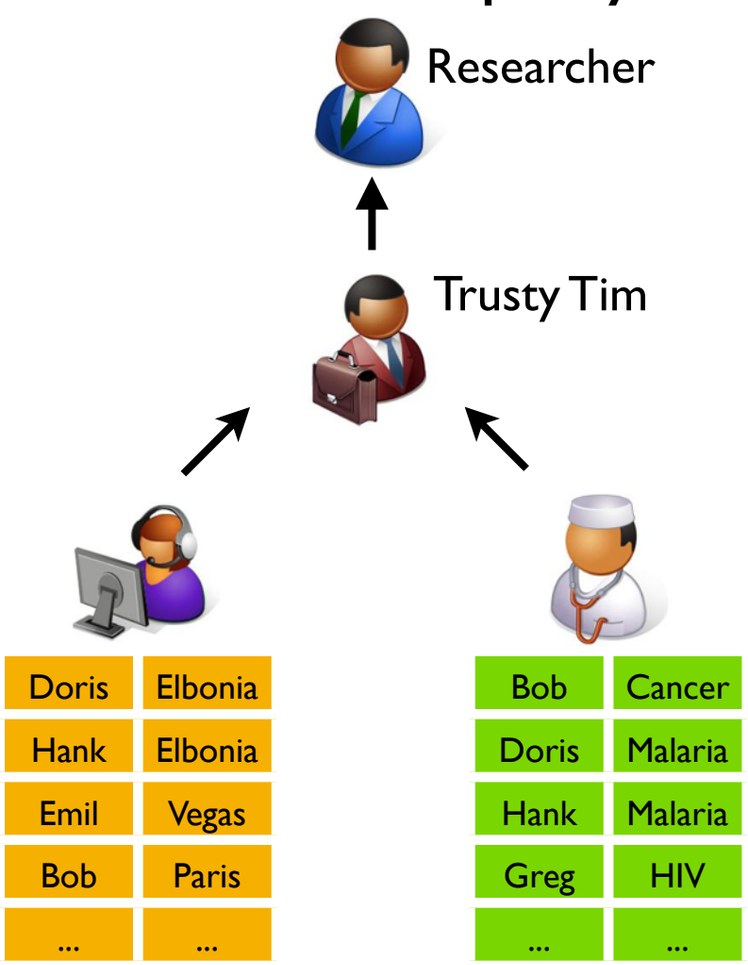
Differential Privacy



- Offers strong, provable privacy guarantees:
  - By giving an upper bound on what an adversary can learn
  - While still allowing us to answer queries safely
- There are existing implementations:
  - PINQ [Sigmod 2009], Airavat [NSDI 2010], ...

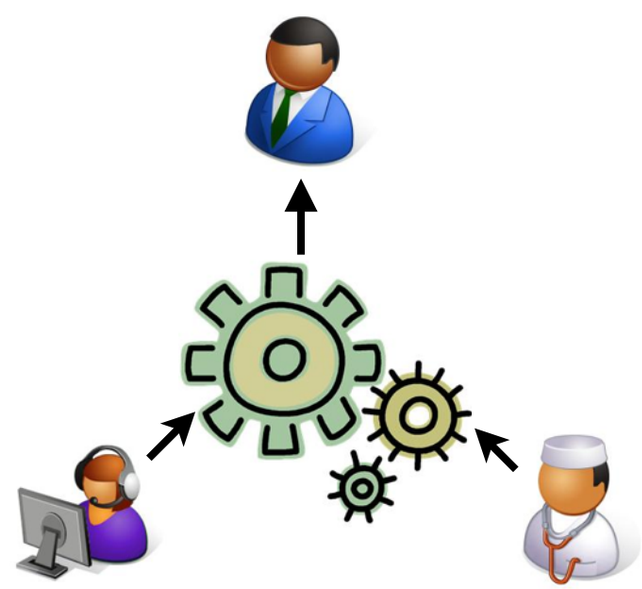
What doesn't work

Idea 1: Give all the data to a trusted party



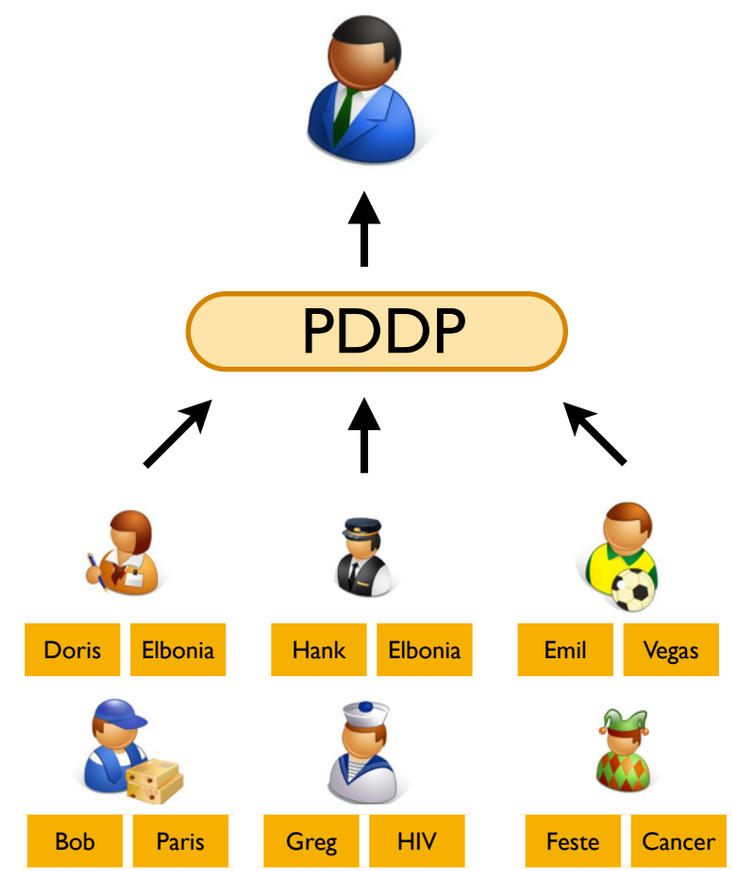
What if we don't have a trusted party?

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

$\cap$

Doris
Hank
Alice

Doris
Hank

2

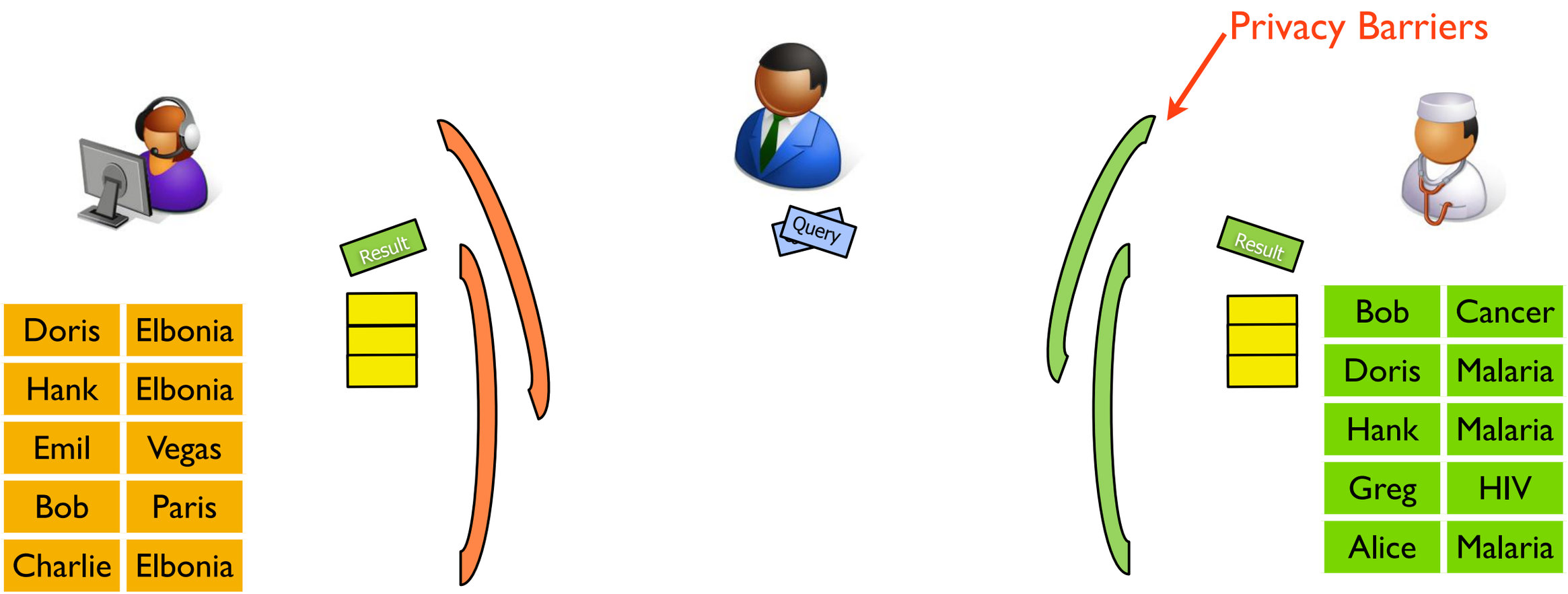
Doris
Hank
Alice

Bob	Cancer
Doris	Malaria
Hank	Malaria
Greg	HIV
Alice	Malaria

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.
- There are partial solutions, e.g. (Freedman et al, Eurocrypt 2004)
  - Their algorithm is not differentially private, but we can extend it.

Overview: DJoin

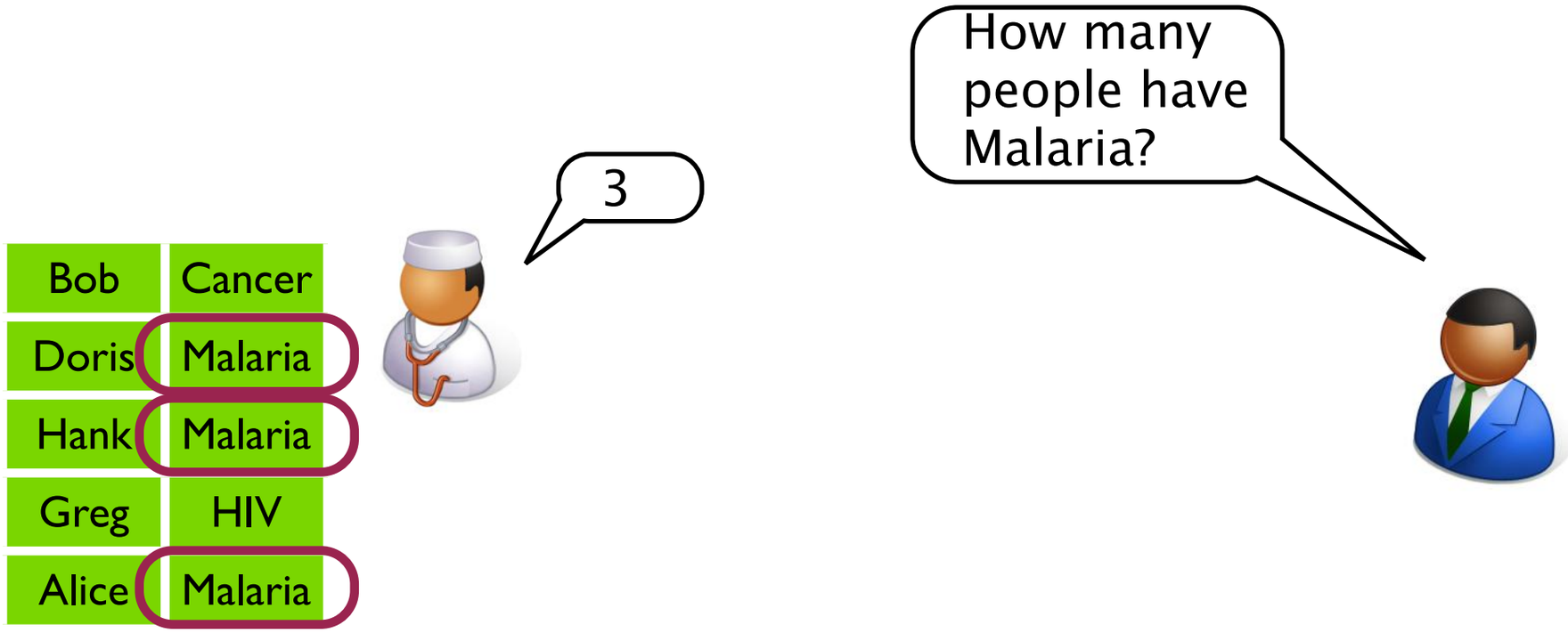


- Rewrites JOIN queries into Set Intersections where possible.
- Executes a distributed private set intersection protocol.
- Gives us differential privacy guarantees for each party:
  - With respect to all other participants.
- We can now make use of our data!

Outline

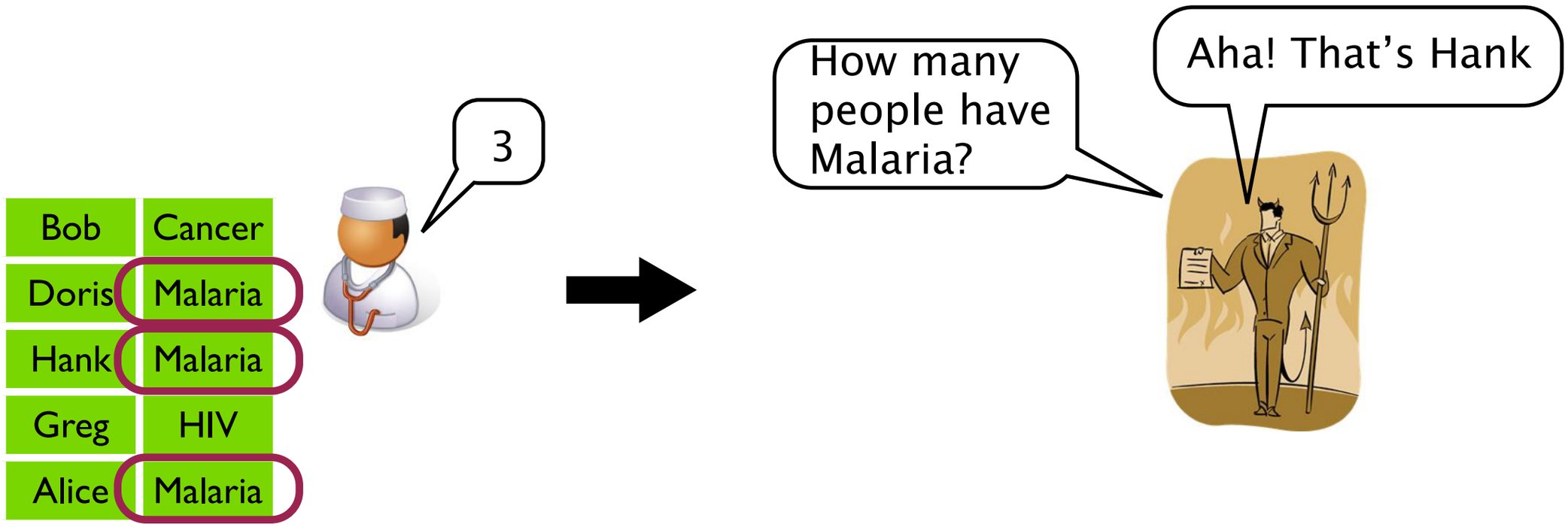
- Motivation: Distributed queries over private data
- Background
  - Differential privacy
  - Private Set Intersection Cardinality (PSI-CA) algorithm
- The DJoin System
  - Making the BN-PSI-CA algorithm differentially private
  - Denoise-Combine-Renoise
  - Query Rewriting
- Evaluation
- Conclusion

Background: Differential Privacy



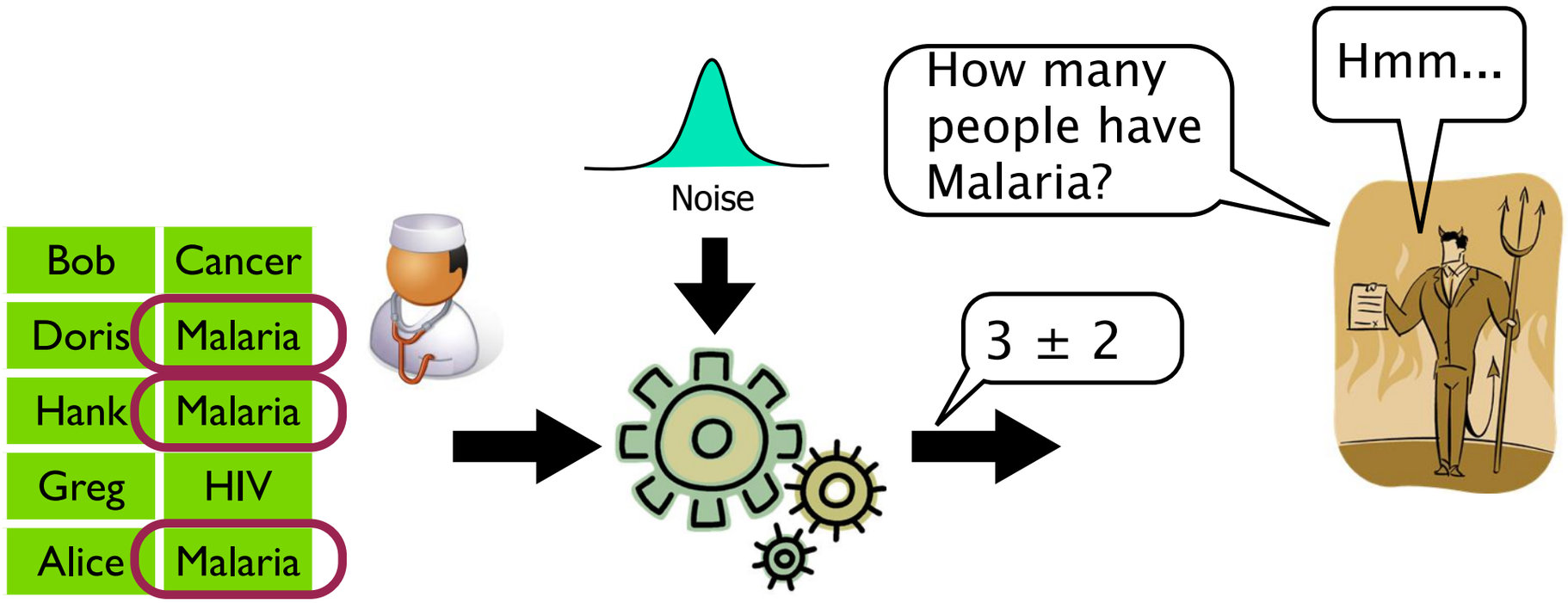
- Typically answers queries about aggregates.
- But to protect privacy, we need more...

Background: Differential Privacy



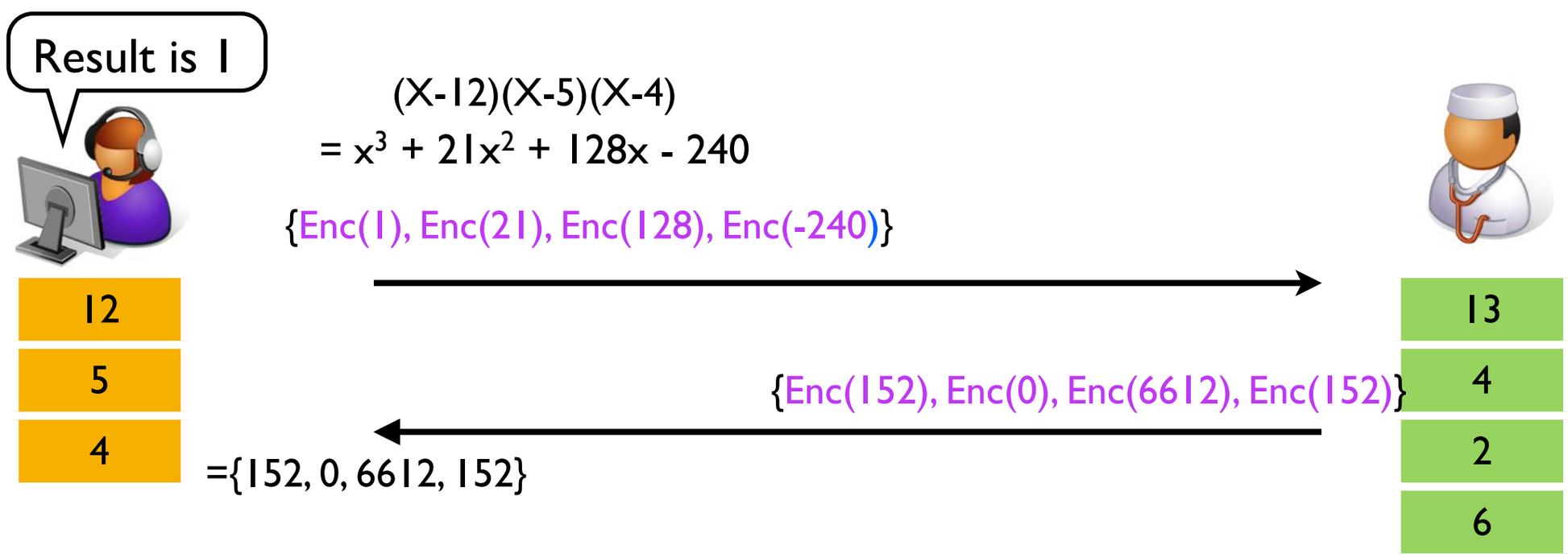
- Suppose our researcher's credentials have been stolen.
  - And the thief has certain outside information.
- We need guarantees even when the querier has outside information!
  - "I know that 2 other people have Malaria, but what about Hank?"

Background: Differential Privacy



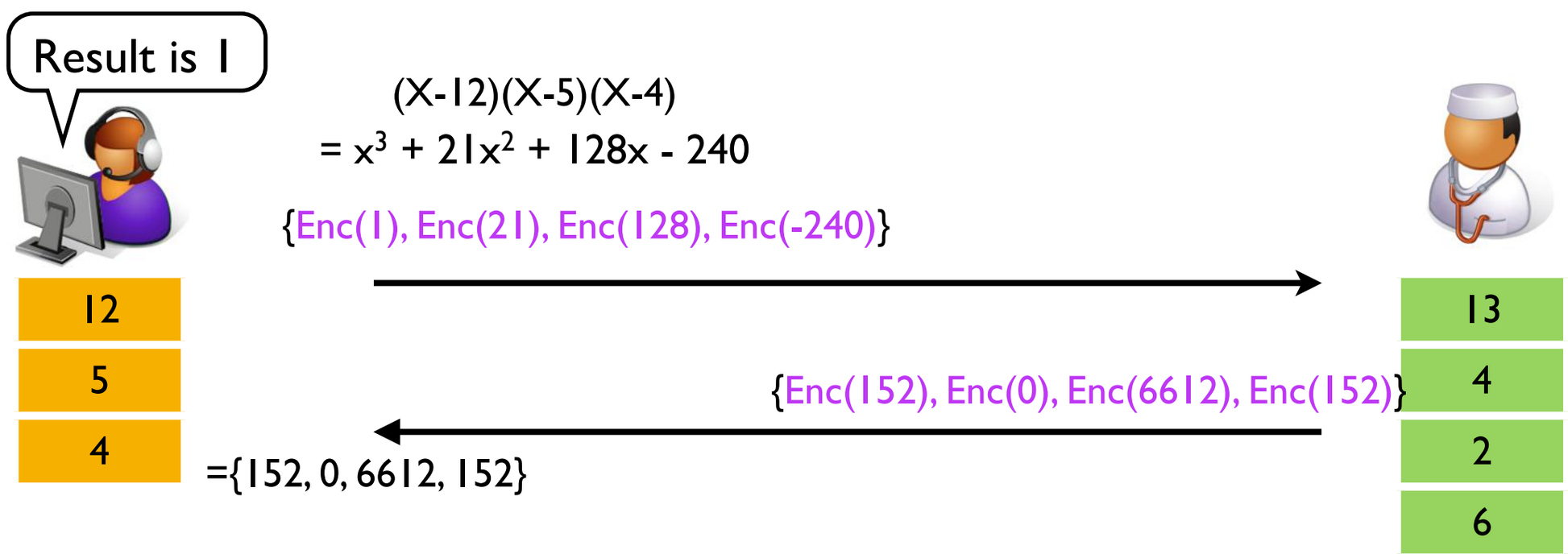
- We need guarantees even when the querier has outside information.
  - "I know that 2 other people have Malaria, but what about Hank?"
- Solution: Differential Privacy adds noise to the answer.
  - Effect: Bounds how much more certain the adversary can be.
- Lots of mathematical detail omitted.
  -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

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.

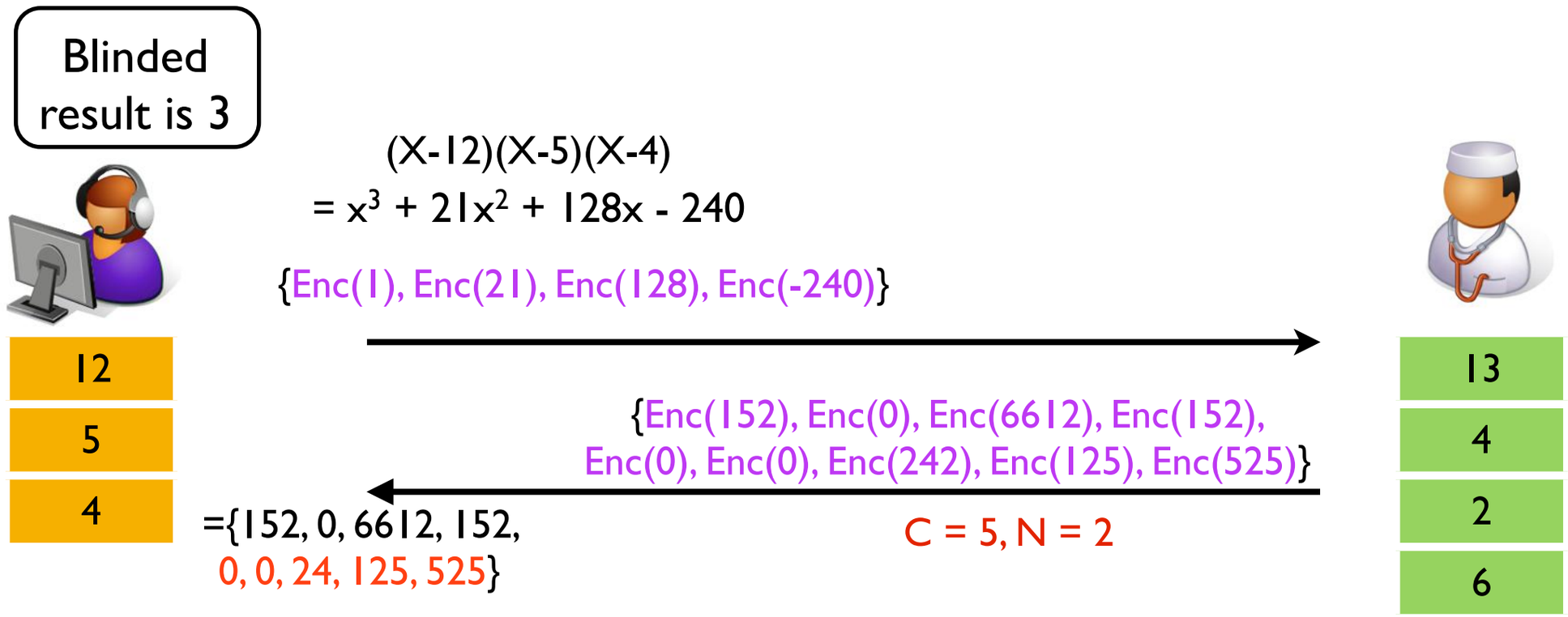
PSI-CA without Differential Privacy



- Protocol from Freedman et al [Eurocrypt 2004]

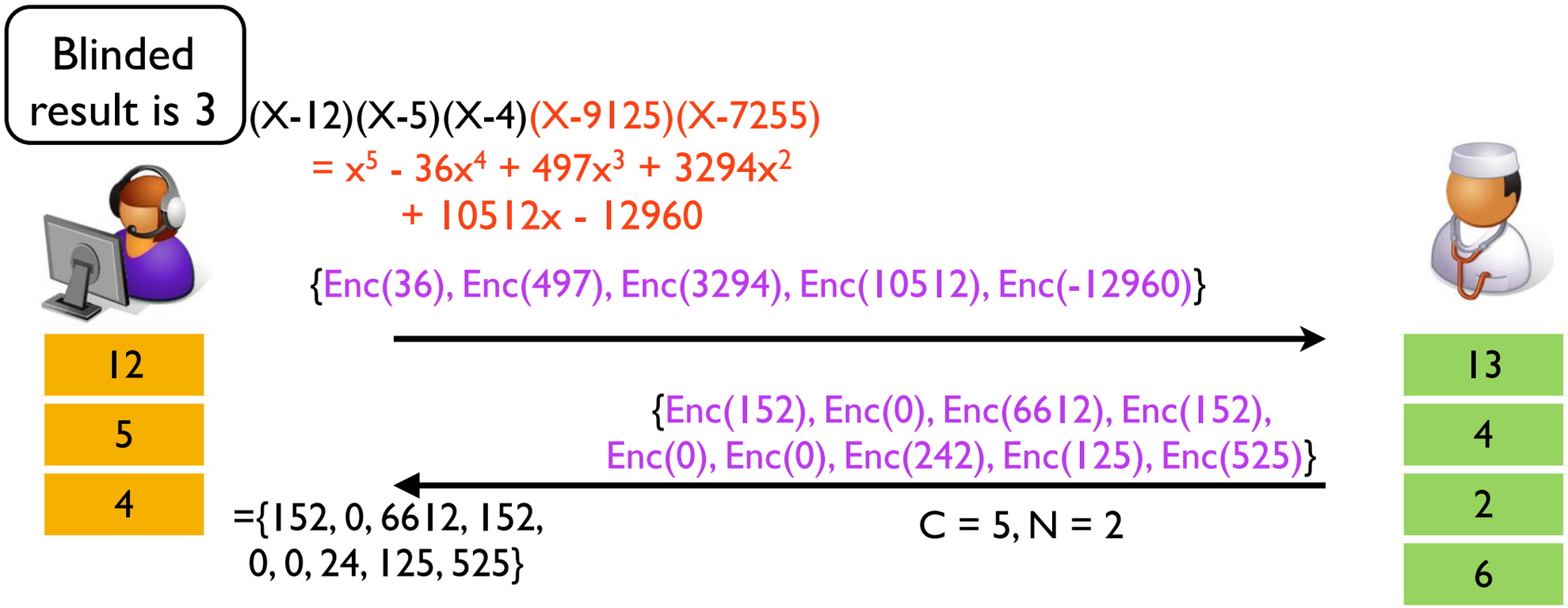
- This protocol is **not differentially private** because:
  - 1. The first party learns the exact size of the intersection.
  - 2. Both parties learn the exact size of the other database.

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.

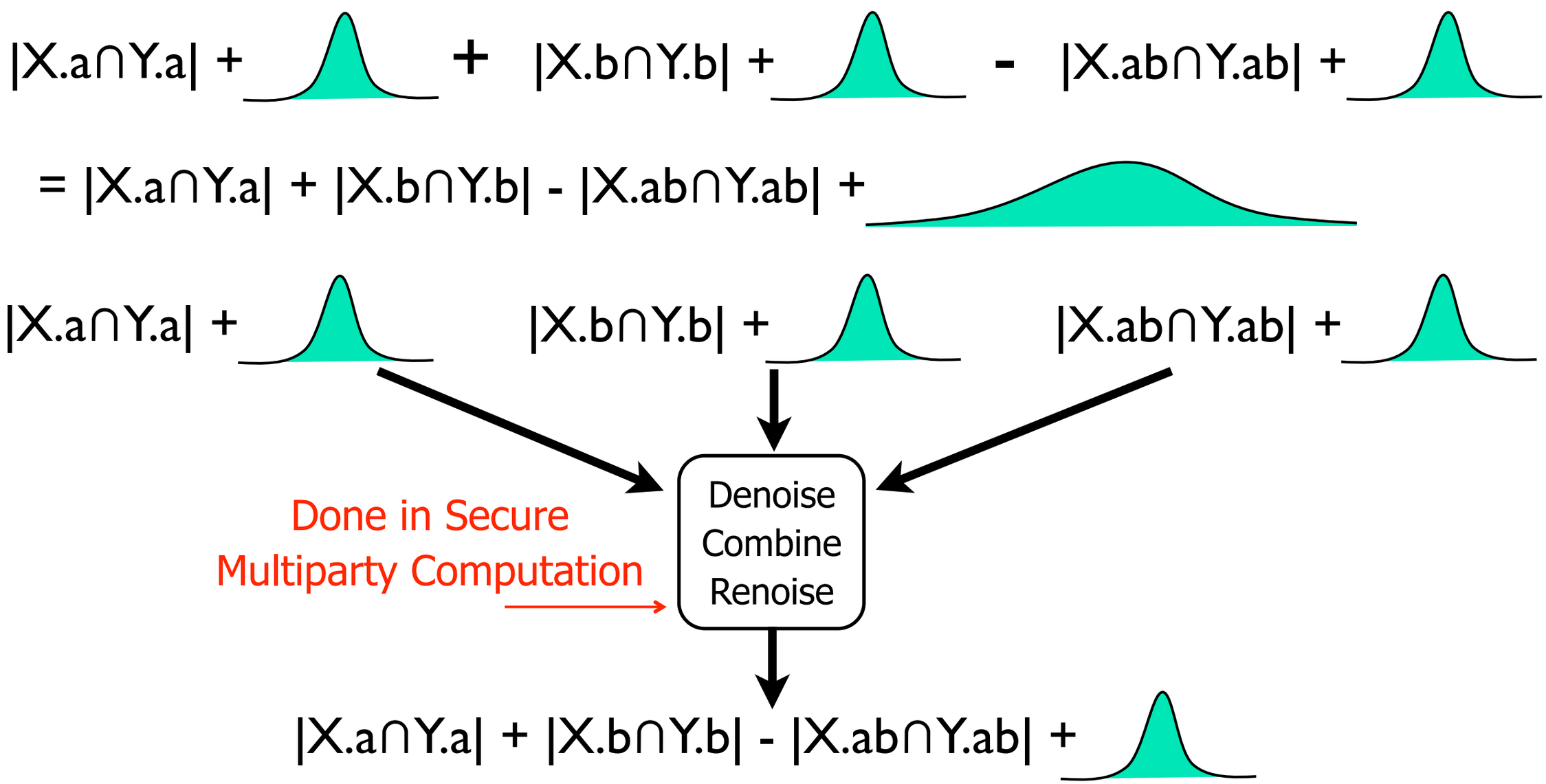
Denoise-Combine-Renoise

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



The DJoin System

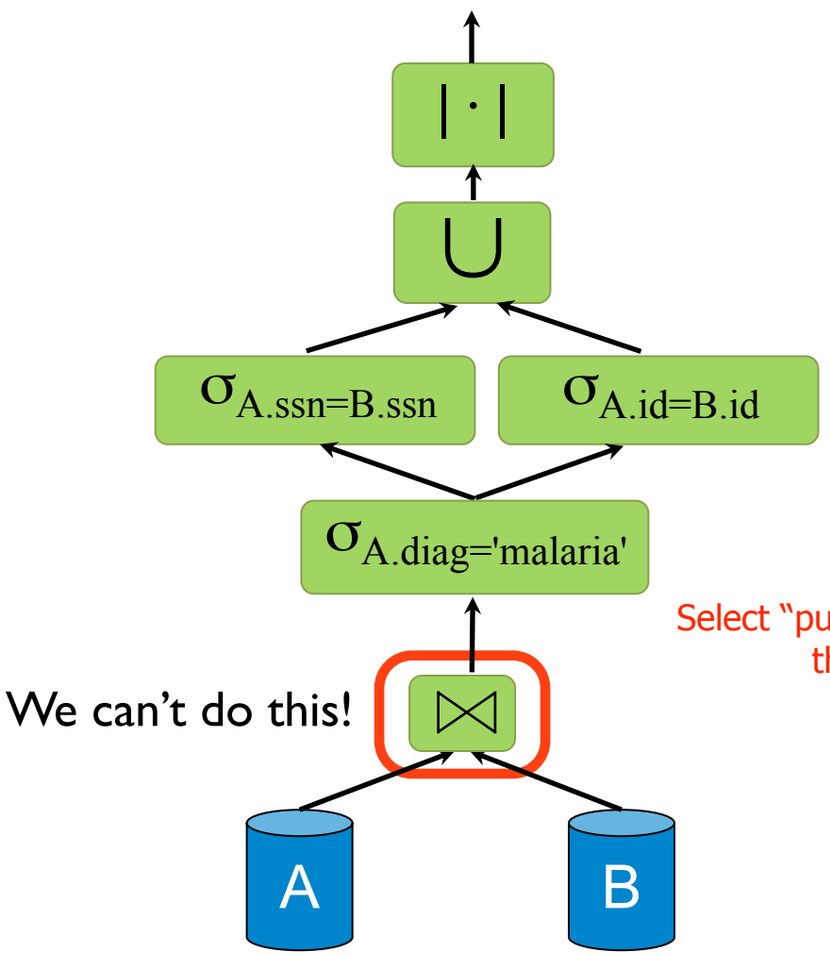
- We built a system called DJoin that uses BN-PSI-CA and DCR to answer queries.
- DJoin solves several other challenges:

- Query Rewriting ← This talk
  Convert SQL-like queries into set intersections.
- Privacy Budget:
  Prevents queriers from issuing too many queries.
- Sensivitity Analysis:
  Determines how much noise is necessary for a given query.
- Encoding:
  Translate JOINs that are not equivalent to a single set intersection.
- Database Management:
  Manages the local information on each node.
- Secure Execution:
  Protects against malicious queriers.

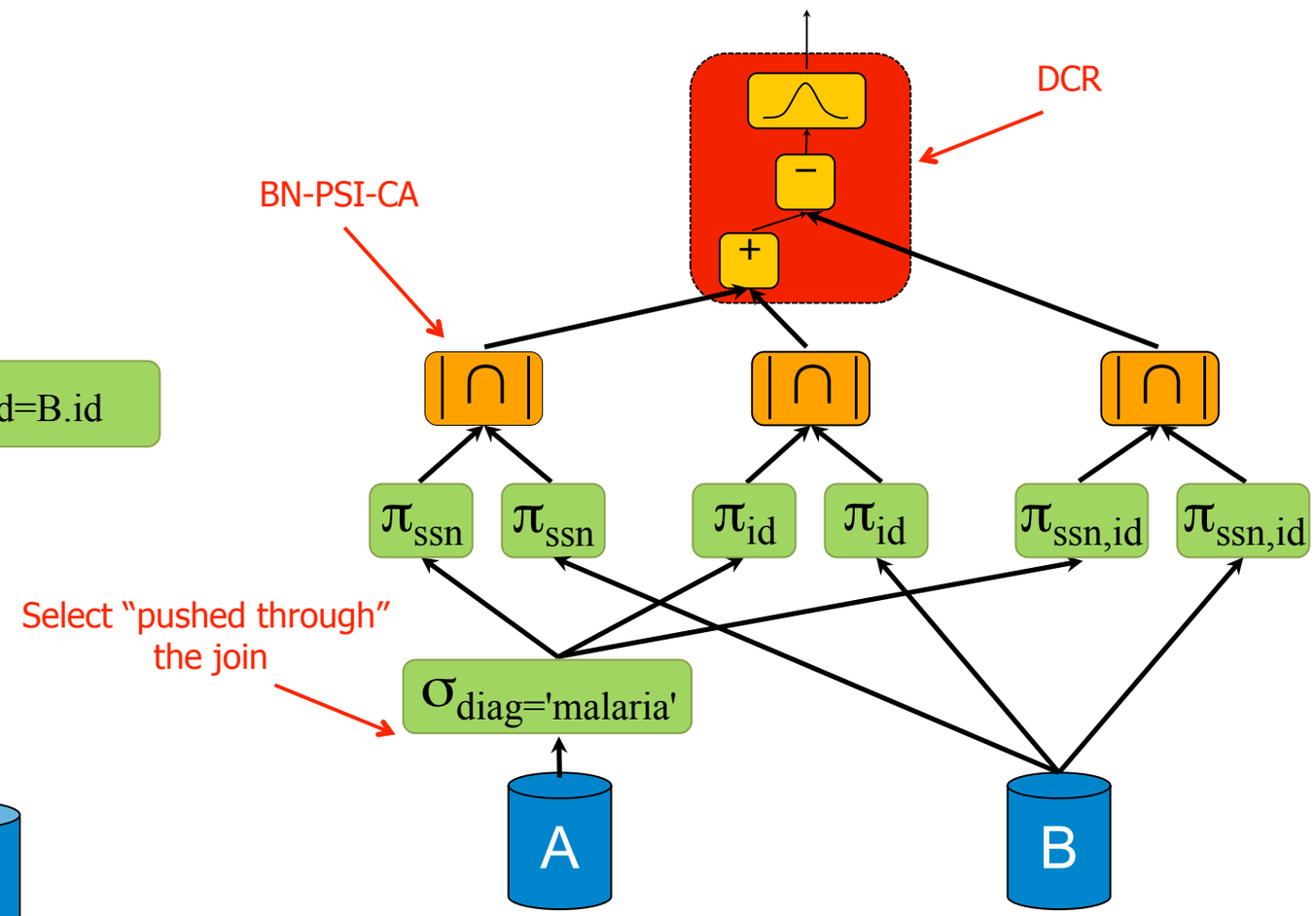
In the paper

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.

When does rewriting work

- ✓ This works if the WHERE clause contains
  - ✓ arbitrary operations on local databases.
  - ✓ conjunctions and disjunctions of equalities across databases.
  - ✓ certain inequalities and numeric comparisons across databases.

- ★ We have all rewrite rules in the paper.

- ◉ Some JOINS cannot be supported:
  - ◉ Some JOINs are not differentially private.
  - ◉ Others because we don't know how to efficiently encode them as set intersections (such as substrings across databases).

Outline

- Motivation: Distributed queries over private data
- Background
  - Differential privacy
  - Private Set Intersection Cardinality (BN-PSI-CA) algorithm
- The DJoin System
  - Making the PSICA algorithm differentially private
  - Denoise-Combine-Renoise
  - Query Rewriting
- Evaluation
- Conclusion

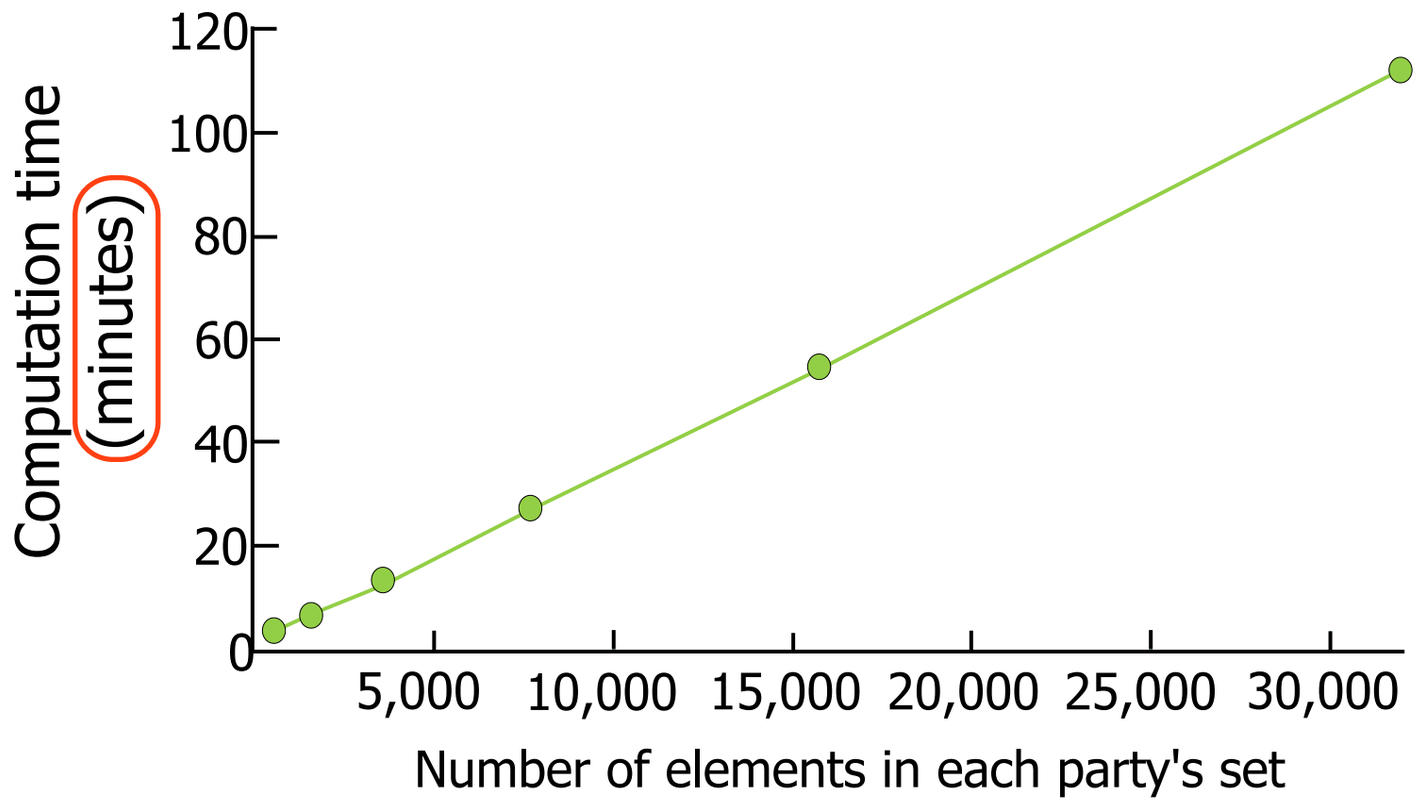
Evaluation

- How expressive are the queries?
- How long do queries take?
- How well does BN-PSI-CA scale with the size of the database?
- How many BN-PSI-CAs per query?
- How expensive is DCR?
- Can you parallelize the system?

Experimental Setup

- We built a DJoin prototype using:
  - Five commodity machines with
  - Xeon E5530 2.4Ghz, 12GB RAM, GBit ethernet
  - mySQL local databases
  - FairplayMP for Secure Multiparty Computation
  - Paillier Cryptosystem
  - BN-PSI-CA based on Freedman et al [Eurocrypt 2004], Kissner/Song [Crypto 2005]
  - Optimizations to speed up runtime
  - 15,000 row databases using synthetic data

Performance of BN-PSI-CA



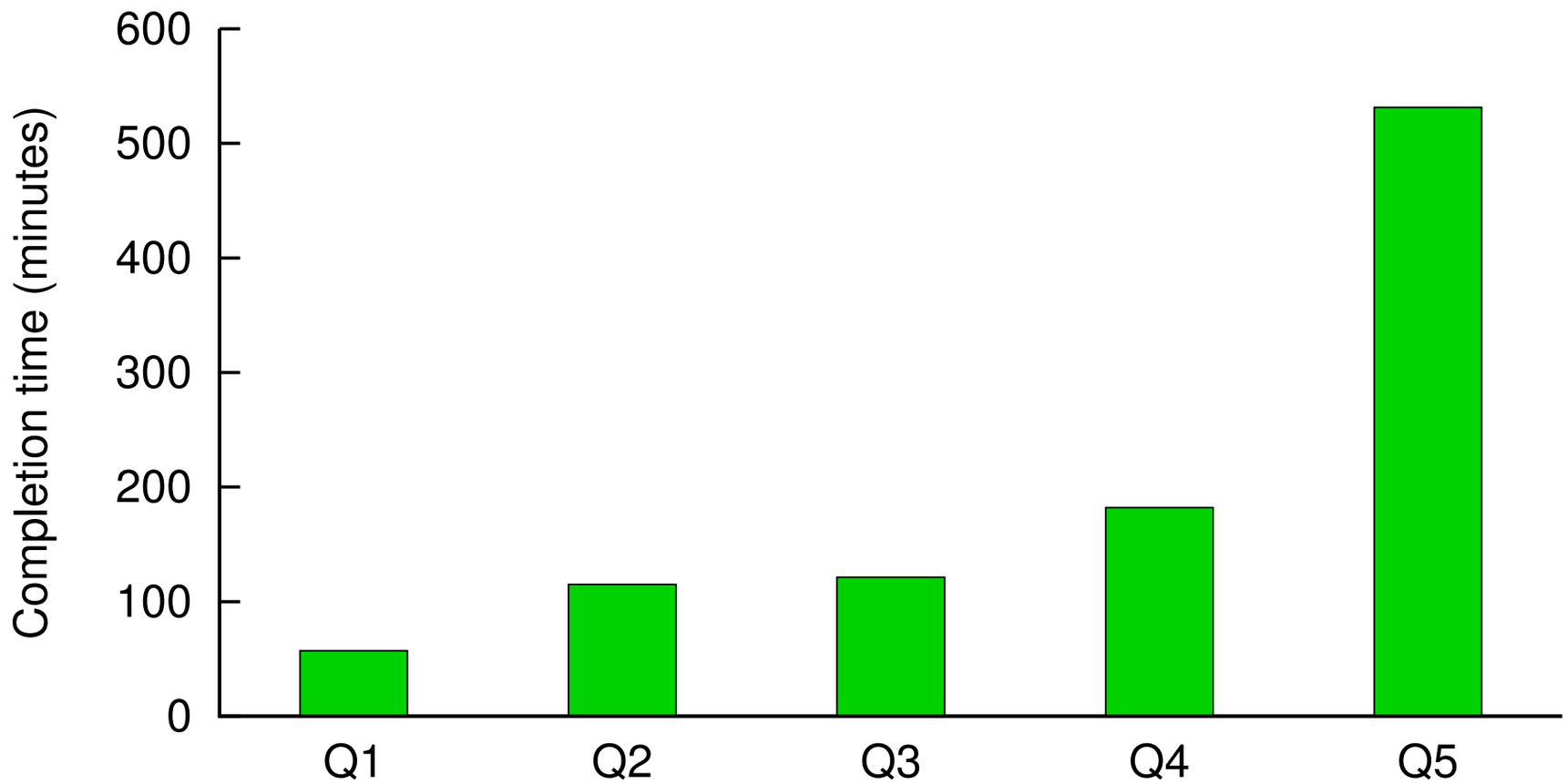
- Almost Linear in database size: $O(|S_1|+|S_2| \ln \ln |S_1|)$.
- Non trivial computational cost: not suitable for interactive use.
- Suitable for offline analysis.
- Parallelizable (3.98x speedup with 4 cores).

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

Example Query Performance



- 15,000 row databases take between 1 and 8 hours.
- Can be parallelized with more cores/machines.
- For comparison: A naive implementation in SMC takes 40 seconds for 8 rows and scales quadratically

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.

Visit our project webpage at http://privacy.cis.upenn.edu!