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

Is there correlation between treatment for malaria and travel to high-risk areas?
Motivation

“Is there a Malaria epidemic in Elbonia?”

Airlines
- Doris
- Hank
- Emil
- Bob
- ...

Elbonia
Elbonia
Vegas
Paris
...

STOP
PRIVATE

Doctors
- Bob
- Doris
- Hank
- Greg
- ...

Cancer
Malaria
Malaria
HIV
...

Researcher
Differential Privacy

Q: How many people went to Elbonia and had Malaria?

About 302

Differentially Private Query Processor

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

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

Who went to Elbonia?

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

\[(X-12)(X-5)(X-4)\]
\[= x^3 + 21x^2 + 128x - 240\]

\[\{\text{Enc(1), Enc(21), Enc(128), Enc(-240)}\}\]

Result is 1

\[\{\text{Enc(152), Enc(0), Enc(6612), Enc(152)}\}\]

\[\{152, 0, 6612, 152\}\]

- **Protocol from Freedman et al [Eurocrypt 2004]**
- The airline have two sets A and B and want to jointly compute |A ∩ 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

\[(X-12)(X-5)(X-4) = x^3 + 21x^2 + 128x - 240\]

\{Enc(1), Enc(21), Enc(128), Enc(-240)\}

\{Enc(152), Enc(0), Enc(6612), Enc(152)\}

\{152, 0, 6612, 152\}

- 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

\[(X-12)(X-5)(X-4) = x^3 + 21x^2 + 128x - 240\]
\[
\{\text{Enc}(1), \text{Enc}(21), \text{Enc}(128), \text{Enc}(-240)\}
\]

12
5
4

13
4
2
6

\{\text{Enc}(152), \text{Enc}(0), \text{Enc}(6612), \text{Enc}(152), \text{Enc}(0), \text{Enc}(0), \text{Enc}(242), \text{Enc}(125), \text{Enc}(525)\}

- 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

1. **Blinded result is 3**
   \[(X-12)(X-5)(X-4)(X-9125)(X-7255)\]
   \[= x^5 - 36x^4 + 497x^3 + 3294x^2 + 10512x - 12960\]
   \[=\{\text{Enc}(36), \text{Enc}(497), \text{Enc}(3294), \text{Enc}(10512), \text{Enc}(-12960)\}\]

2. **12**
   \[=\{\text{Enc}(152), \text{Enc}(0), \text{Enc}(6612), \text{Enc}(152), \text{Enc}(0), \text{Enc}(242), \text{Enc}(125), \text{Enc}(525)\}\]
   \[C = 5, N = 2\]

3. **Challenge 2**: Both parties learn the exact size of the other database.

4. **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|

\[
|X.a \cap Y.a| + \quad + \quad |X.b \cap Y.b| + \quad - \quad |X.ab \cap Y.ab| + \quad
\]

= |X.a \cap Y.a| + |X.b \cap Y.b| - |X.ab \cap Y.ab| +

\[
\text{Done in Secure Multiparty Computation}
\]

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

<table>
<thead>
<tr>
<th>Query</th>
<th>BN-PSI-CAAs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SELECT NOISY COUNT(A.x) FROM A,B WHERE A.x=B.y</td>
<td>1</td>
</tr>
<tr>
<td>2. SELECT NOISY COUNT(A.x) FROM A,B WHERE A.x=B.x AND (A.y!=B.y)</td>
<td>2</td>
</tr>
<tr>
<td>3. SELECT NOISY COUNT(A.x) FROM A,B WHERE A.x=B.y AND (A.z=&quot;x&quot; OR B.p=&quot;y&quot;)</td>
<td>2</td>
</tr>
<tr>
<td>4. SELECT NOISY COUNT(A.x) FROM A,B WHERE A.x=B.x OR A.y=B.y</td>
<td>3</td>
</tr>
<tr>
<td>5. SELECT NOISY COUNT(A.x) FROM A,B WHERE A.x LIKE &quot;%xyz%&quot; AND A.w=B.w AND (B.y+B.z&gt;10) AND (A.y&gt;B.y)</td>
<td>8</td>
</tr>
</tbody>
</table>

- 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


Thank You!