Jana: Secure Computation with Differential Privacy, and Applications

Rebecca Wright
Director, DIMACS
Professor, Computer Science Dept., Rutgers University
www.cs.rutgers.edu/~rebecca.wright

DIMACS/Northeast Big Data Hub Workshop
on Overcoming Barriers to Data Sharing
October 23-24, 2017
JANA: PRACTICAL PRIVATE DATA-AS-A-SERVICE

“BENE VIXIT, BENE QUI LATUIT.” - OVID

Carried out as part of DARPA’s Brandeis program.

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) and Space and Naval Warfare Systems Center, Pacific (SSC Pacific) under Contract No. N66001-15-C-4070. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or SSC Pacific.
Talk Outline

- Overview of Jana
- Specific directions in secure multiparty computation (MPC), order-revealing encryption, and differential privacy
- Application scenarios
- Conclusions
Private Data as a Service

- Data as a service has proved very popular and useful.
  - Easy to use
  - Familiar interfaces
  - Fast
  - Reliable (ACID properties)
  - Privacy and security models can include encryption for data in transit, and in some cases for data at rest, some also allow computation on encrypted data (e.g., via order-revealing encryption).

- We explore the use and advancement of state-of-the-art privacy tools and methods to develop a private-data-as-a-service platform with stronger, more flexible privacy.

- Coupled with implementation and practical use cases, this lets us explore engineering issues and practical tradeoffs, and drive new research.
The Jana Platform for Private Data as a Service
Jana Capabilities

- Functionality
  - Generous subset of SQL
  - RDBMS ACID properties

- Privacy
  - Data-in-transit: public key cryptography
  - Data-at-rest: deterministic, random, searchable
  - Computation: MPC, or in RDBMS using deterministic & searchable encryption
  - Results: differential privacy applied (if needed) while in MPC

- Performance
  - 10Ks of records moving to 100Ks, queries in seconds to hours

- Deployment
  - Web service with RESTful API
  - Docker appliance
Currently Implemented Subset of SQL

- SELECT, PROTECT, JOIN, UNION, INTERSECT, EXCEPT
- Integer, String, Boolean, Enum, Fixed-Point, Date
- Nested query support

```sql
SELECT person.person_id, lastname, firstname, diseasestate, gender, birthdate
FROM person
  JOIN community ON community.community_id = person.residence
  JOIN person2diseasestate ON person2diseasestate.person_id = person.person_id
  JOIN policyauthority2community ON policyauthority2community.community_id = community.community_id
  JOIN policyauthority ON policyauthority.authority_id = policyauthority2community.authority_id
WHERE person2diseasestate.transitiondate < '04-20-2017'
  AND person2diseasestate.diseasestate IN ('I')
  AND policyauthority.authority = 'CebuCityCommunityPA'
AND person.person_id NOT IN
  (SELECT person.person_id
   FROM person
     JOIN community ON community.community_id = person.residence
     JOIN person2diseasestate ON person2diseasestate.person_id = person.person_id
     JOIN policyauthority2community ON policyauthority2community.community_id = community.community_id
     JOIN policyauthority ON policyauthority.authority_id = policyauthority2community.authority_id
     WHERE person2diseasestate.transitiondate < '04-20-2017'
     AND person2diseasestate.diseasestate IN ('R', 'D')
     AND policyauthority.authority = 'CebuCityCommunityPA');
```
Underlying Primitives/Mechanisms

- SPDZ for secure multiparty computation [DPSZ12, DKLPS13]

- possibility of using order-revealing encryption or other deterministic encryption to make some kinds of queries much faster [AKS04, BCLO09]

- distributed generation of geometric noise for differential privacy, similar to [DKMMN06]
Some Research and Integration
Issues and Results

- Problem: We want symmetric encryption that can be efficiently computed “inside” the MPC.
  - Results: MPC-friendly symmetric encryption [GRRSS16]

- Problem: Want to better understand the privacy implications of using order-preserving encryption.
  - Results: How (in)secure is order-revealing encryption? [DDC16]
  - Ongoing work to try to fully characterize tradeoffs and develop best-possible solutions.

- Problem: The noise for differential privacy, as well as many functions we might want to compute make use of non-finite-field operations.
  - Goal: MPC-friendly differential privacy
  - For noise, currently using variant of [DKMMN06].
MPC-friendly symmetric encryption [GRRSS16]

- Goal: design pseudo-random functions (PRFs) that are suitable for use in a secret-sharing based MPC system.
  - I.e., in which data is shared as elements of a finite field $F_p$, of large prime characteristic.
  - Enables efficient protocols to compute relatively complex functions such as integer comparison, fixed point arithmetic, and linear programming.
  - In contrast, byte/word-oriented operations such as those in AES are hard to represent.

- Results: GRRSS consider three different candidate PRFs: the Naor-Reingold PRF [NR97], a PRF based on the Legendre symbol [DHI03], and a specialized block cipher design called MiMC [AGRRT16]. No one of them dominates in all situations, but MiMC performed best for throughput, has lowest pre-processing requirements, and is best for encrypting/decrypting data into or out of the MPC.

- Outcome for Jana:
  - We have now included MiMC in the Jana codebase.
MPC-Friendly PRFs and Modes

Encryption time

Take-away:
- Several good options
- Choice for latency depends on number of blocks
MPC-Friendly PRFs and Modes

Take-away:
Throughput favors MiMC-based PRF and OTR.
Some Research and Integration
Issues and Results

- Problem: We want symmetric encryption that can be efficiently computed “inside” the MPC.
  - Results: MPC-friendly symmetric encryption [GRRSS16]

- Problem: Want to better understand the privacy implications of using order-preserving encryption.
  - Results: How (in)secure is order-revealing encryption? [DDC16]
  - Ongoing work to try to fully characterize tradeoffs and develop best-possible solutions.

- Problem: The noise for differential privacy, as well as many functions we might want to compute make use of non-finite-field operations.
  - Goal: MPC-friendly differential privacy
  - For noise, currently using variant of [DKMMN06].
Order-Revealing Encryption (ORE) [AKSX’04, BCLO’09]

Order-Preserving Encryption (OPE): A symmetric encryption scheme that is deterministic and strictly increasing.

- Order-Revealing Encryption is a generalized form of OPE. Both enable efficient computation of range queries on encrypted data.
- ORE/OPE are inherently less secure than standard encryption, subject to chosen-plaintext attacks.
- Research approach: Construct ORE schemes with best-possible security against passive attackers who only capture ciphertexts.
### DDC16: New Security Issues with ORE

#### Attacks on ORE with Correlated Columns

<table>
<thead>
<tr>
<th>Zip</th>
<th>First Name</th>
<th>Last Name</th>
<th>Zip</th>
<th>D.O.B</th>
</tr>
</thead>
<tbody>
<tr>
<td>686065</td>
<td>6d9737</td>
<td>a22844</td>
<td>686065</td>
<td>5ad287</td>
</tr>
<tr>
<td>48eb42</td>
<td>9d8ea6</td>
<td>753996</td>
<td>48eb42</td>
<td>abd94c</td>
</tr>
<tr>
<td>26861e</td>
<td>10eca7</td>
<td>b6b59c</td>
<td>26861e</td>
<td>405702</td>
</tr>
<tr>
<td>01c36e</td>
<td>d99ff8</td>
<td>a2e2a0</td>
<td>01c36e</td>
<td>0abd94</td>
</tr>
</tbody>
</table>

prior work: attacks single column

**DDC work:** attacks multiple columns

- Possible to attack multiple columns even when individual columns are not individually amenable to attack.
DDC16: New Security Issues with ORE

**Attacks on ORE with Correlated Columns**

**Attacks on ORE with Non-Uniform Data**

- First analysis of practical ORE when data are not uniform.
- Some practical ORE constructions reveal far more information on real data than on random data.
DDC16: New Security Issues with ORE

Attacks on ORE with Correlated Columns

Attacks on ORE with Non-Uniform Data

Experiments on geolocation and time stamps.
DDC16: New Security Issues with ORE

**Attacks on ORE with Correlated Columns**

**Attacks on ORE with Non-Uniform Data**

Experiments on geolocation and time stamps.

**Meta-Conclusion:** Need to cryptanalyze definitions/models for secure-but-leaky ORE in practice.
Case Study: California Road Intersections

Data: Latitude/longitude of 21,000 road intersections, each encoded in 27 bits.

If bounding box is known: Can guess 30% of points to within 50km
Inferring More Bits from MSDB Leakage

Most significant differing bit leakage on California dataset:

01x010100110110111xxxxxxxxxxxx
01x000100101001x10xxxxxxxxxxxx
01x0011000011001xxxxxxxxxxxx
10x0011010x00111xxxxxxxxxxxx
01x001010101111xxxx0xxxxxxxx
10x010110001010x10xxxxxxxxxx
01x0100110001x1xxxxxxxxxxxxx
...

**Guessing algorithm:**
1) For each x, try replacing with 0/1
2) Take guess that minimizes total pairwise distance between points.

Visualized with “x ⟷ 0.5”: 

![Map visualization](image)
Results From Inference Algorithm

- ran the attack on dataset sizes 200 and 2000.
- attack guesses more than 50% of points to within 0.5km
- even though explicit MSDB leakage did not reveal any point to within 400km
Order-revealing Encryption Conclusions

1. Correlation causes information leakage, even for ideal ORE.
2. Leaky ORE may be much leakier than previously thought.
3. We should consider other primitives and different approaches for database protection (and cryptanalyze them).
Some Research and Integration Issues and Results

- Problem: We want symmetric encryption that can be efficiently computed “inside” the MPC.
  - Results: MPC-friendly symmetric encryption [GRRSS16]

- Problem: Want to better understand the privacy implications of using order-preserving encryption.
  - Results: How (in)secure is order-revealing encryption? [DDC16]
  - Ongoing work to try to fully characterize tradeoffs and develop best-possible solutions.

- Problem: The noise for differential privacy, as well as many functions we might want to compute make use of non-finite-field operations.
  - Goal: MPC-friendly differential privacy
  - For noise, currently using variant of [DKMMN06].
Differential Privacy in SPDZ

- Support for typical aggregates: count, sum, average
  - Computed in SPDZ in order to maintain privacy
  - We need MPC-friendly DP mechanisms.
  - We currently are using a geometric distribution to generate noise in SPDZ (approximating Laplace noise), similar to [DKMMN06]

- Extended query language to support
  - SELECT … DP_COUNT(<w>, <column>) … FROM … WHERE …
    - …and DP_SUM, DP_AVERAGE too
  - Interface allows a querier to specify required accuracy.
    - Then applies as much noise (privacy) as possible to aggregate <column> values within <w> of the actual answer with 95% confidence.
Privacy vs. Performance

The graph shows the runtime (in seconds) for different values of $w$. The y-axis represents the runtime, while the x-axis represents the values of $w$ ranging from 1 to 1000.

Two lines are plotted:
- Counting (*), represented by blue points.
- DP_Count($w$, *), represented by cyan points.

As $w$ increases, both Count(*) and DP_Count($w$, *) show an increase in runtime, with DP_Count($w$, *) having a steeper slope than Count(*).
For now, the Jana implementation simply tracks how much privacy budget has been expended, and can return this information on request.

We envision support for more complex modes of operation, including discarding data (for privacy reasons, or other reasons but with beneficial privacy implications).

As far as the question of “what values of epsilon are safe”, this is application-dependent, as well as dependent on risk tolerance of involved stakeholders. But developing general guidelines is likely a community effort (akin to recommending key sizes in cryptography).
Differential Privacy Conclusions

- Generating appropriately distributed noise is expensive in secret-sharing-based MPC, even for straightforward additive noise mechanisms.

- More work is needed to support users to develop appropriate policies.
Talk Outline

- Overview of Jana
- Specific directions in secure multiparty computation (MPC), order-revealing encryption, and differential privacy
- Application scenarios
- Conclusions
Privacy-preserving Information Mediation for Enterprises (PRIME)

TA3: Enterprise Platform

Karen Myers (PI) [slides used with permission]
Tim Ellis
Tancrède Lepoint

SRI International

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) and Space and Naval Warfare Systems Center, Pacific (SSC Pacific) under Contract No. N66001-15-C-4071. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA or SSC Pacific.
PRIME Enterprise Platform

**Objective:** Enable informed cross-enterprise information sharing that achieves coordination goals while satisfying privacy objectives
Info Sharing for Coalitions in the Pacific
US Pacific Fleet (PACFLT), US Pacific Command (PACOM)

“Information sharing is one of our biggest challenges”
- PACOM Science Advisor

Coalition Composition
- From large multinational groups to limited partnerships
  - Inter-service, inter-agency, international
  - NGOs, OGOs, corporations
- From close allies to adversaries
- Relationships can change abruptly

Data Characteristics
- Distributed storage; access gated by different providers
- Large volumes, possibly streaming
- Much unstructured data
  - text, imagery, PowerPoint

Privacy Tradeoff

Benefits of Sharing

Risks of Sharing
Enterprise Privacy Models

Cross Enterprise
- Independent organizations with no/limited trust; addressing some common goals
- Ad hoc, federated data access model

Within Enterprise
- Trusted partners within a single overarching organization; regulations restrict sharing
- Fixed, federated data access model

Trusted Broker
- Mostly untrusted but with a common trusted party
- Centralized data model, with access controlled by trusted party
Brandeis Enterprise Demo
Humanitarian Assistance/Disaster Relief (HADR)
Operational Threads

Privacy-aware COPs
Display continuously updating AOR info under control of privacy policies. Support basic coordination queries.

**Protect:** ship info (capabilities, tracks, contents), sensor sources

Pandemic
Predict progression of disease and take steps to counter it.

**Protect:** PII, disease spread, disease characteristics

Aid Distribution
Allocate and distribute resources (food, water, medicine) from ships in AOR to areas that require relief.

**Protect:** resource availability, ship capabilities, ship positions
## Jana Pandemic Schema & Query Characteristics

- Private columns (highlighted) in Jana pandemic schema, require encryption & MPC overhead

<table>
<thead>
<tr>
<th>Table</th>
<th>Column</th>
<th>Type</th>
<th>Jana bytes</th>
<th>Pub bytes</th>
<th>SQL bytes</th>
<th>Priv/Pub</th>
<th>Priv Ops</th>
<th>Query 0 COUNT(*)</th>
<th>Query 1 COUNT(*)</th>
<th>Query 2 COUNT(*)</th>
<th>Query 3 Specific Data</th>
<th>Query 4 Outer</th>
<th>Query 4 Inner</th>
</tr>
</thead>
<tbody>
<tr>
<td>community</td>
<td>community_id</td>
<td>int</td>
<td>426</td>
<td>42</td>
<td>4</td>
<td>private</td>
<td>equality</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1/JOIN3</td>
<td>JOIN1/JOIN3</td>
<td>JOIN1/JOIN3</td>
</tr>
<tr>
<td></td>
<td>community_name</td>
<td>string</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>public</td>
<td>SELECT/GROUP1</td>
<td>SELECT/GROUP1</td>
<td>SELECT/GROUP1</td>
<td>SELECT/GROUP1</td>
<td>SELECT/GROUP1</td>
<td>SELECT/GROUP1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>latitude</td>
<td>Lat</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>public</td>
<td>SELECT/GROUP2</td>
<td>SELECT/GROUP2</td>
<td>SELECT/GROUP2</td>
<td>SELECT/GROUP2</td>
<td>SELECT/GROUP2</td>
<td>SELECT/GROUP2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>longitude</td>
<td>Lon</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>public</td>
<td>SELECT/GROUP3</td>
<td>SELECT/GROUP3</td>
<td>SELECT/GROUP3</td>
<td>SELECT/GROUP3</td>
<td>SELECT/GROUP3</td>
<td>SELECT/GROUP3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>nation_id</td>
<td>int</td>
<td>200</td>
<td>8</td>
<td>4</td>
<td>private</td>
<td>equality</td>
<td>JOIN1</td>
<td></td>
<td></td>
<td>JOIN3/JOIN4</td>
<td>JOIN2</td>
<td>JOIN2</td>
</tr>
<tr>
<td>nation</td>
<td>nation_id</td>
<td>int</td>
<td>224</td>
<td>32</td>
<td>28</td>
<td>private</td>
<td>equality</td>
<td></td>
<td>JOIN3/JOIN4</td>
<td></td>
<td>JOIN2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>nation_name</td>
<td>string</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>public</td>
<td>SELECT</td>
<td></td>
<td></td>
<td></td>
<td>SELECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>latitude</td>
<td>Lat</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>public</td>
<td>SELECT</td>
<td></td>
<td></td>
<td></td>
<td>SELECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>longitude</td>
<td>Lon</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>public</td>
<td>SELECT</td>
<td></td>
<td></td>
<td></td>
<td>SELECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>person</td>
<td>person_id</td>
<td>int</td>
<td>13000</td>
<td>56</td>
<td>44</td>
<td>private</td>
<td>equality</td>
<td>JOIN2</td>
<td>JOIN2</td>
<td>JOIN3</td>
<td>JOIN2</td>
<td>SELECT/JOIN2</td>
<td>&lt; SELECT/JOIN2</td>
</tr>
<tr>
<td></td>
<td>last_name</td>
<td>string</td>
<td>6000</td>
<td>8</td>
<td>9</td>
<td>private</td>
<td>equality</td>
<td>JOIN2</td>
<td>JOIN2</td>
<td></td>
<td>SELECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>first_name</td>
<td>string</td>
<td>6000</td>
<td>8</td>
<td>9</td>
<td>private</td>
<td>equality</td>
<td>JOIN2</td>
<td>JOIN2</td>
<td></td>
<td>SELECT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>birthdate</td>
<td>Date</td>
<td>200</td>
<td>8</td>
<td>9</td>
<td>private</td>
<td>order</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
</tr>
<tr>
<td></td>
<td>gender</td>
<td>string</td>
<td>200</td>
<td>8</td>
<td>9</td>
<td>private</td>
<td>equality</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
</tr>
<tr>
<td></td>
<td>residence</td>
<td>int</td>
<td>200</td>
<td>8</td>
<td>9</td>
<td>private</td>
<td>equality</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
</tr>
<tr>
<td></td>
<td>citizenship</td>
<td>int</td>
<td>200</td>
<td>8</td>
<td>9</td>
<td>private</td>
<td>equality</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
<td>JOIN1</td>
</tr>
<tr>
<td>person2risksfactor</td>
<td>riskfactor_id</td>
<td>int</td>
<td>208</td>
<td>16</td>
<td>8</td>
<td>public</td>
<td>JOIN3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>person_id</td>
<td>int</td>
<td>200</td>
<td>8</td>
<td>4</td>
<td>private</td>
<td>equality</td>
<td>JOIN3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>person2disease</td>
<td>diseasestate</td>
<td>string</td>
<td>600</td>
<td>24</td>
<td>16</td>
<td>private</td>
<td>equality</td>
<td>SELECT/GROUP4</td>
<td>SELECT/GROUP4</td>
<td>SELECT/GROUP4</td>
<td>SELECT/GROUP4</td>
<td>SELECT/WHERE</td>
<td>SELECT/WHERE</td>
</tr>
<tr>
<td></td>
<td>person_id</td>
<td>int</td>
<td>200</td>
<td>8</td>
<td>9</td>
<td>private</td>
<td>order</td>
<td>JOIN2</td>
<td>JOIN2</td>
<td>JOIN2</td>
<td>JOIN2</td>
<td>SELECT/WHERE</td>
<td>SELECT/WHERE</td>
</tr>
<tr>
<td></td>
<td>transitiondate</td>
<td>Date</td>
<td>200</td>
<td>8</td>
<td>9</td>
<td>private</td>
<td>order</td>
<td>WHERE</td>
<td>WHERE</td>
<td>WHERE</td>
<td>WHERE</td>
<td>WHERE</td>
<td>WHERE</td>
</tr>
<tr>
<td>policyauthority</td>
<td>authority_id</td>
<td>int</td>
<td>28</td>
<td>28</td>
<td>24</td>
<td>public</td>
<td>JOIN4</td>
<td></td>
<td>JOIN4</td>
<td>JOIN5</td>
<td>JOIN4</td>
<td>JOIN5</td>
<td>JOIN5</td>
</tr>
<tr>
<td></td>
<td>authority</td>
<td>string</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>public</td>
<td>WHERE</td>
<td></td>
<td></td>
<td></td>
<td>WHERE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>policyauthority2community</td>
<td>authority_id</td>
<td>int</td>
<td>208</td>
<td>16</td>
<td>8</td>
<td>public</td>
<td>JOIN4</td>
<td></td>
<td>JOIN4</td>
<td>JOIN4</td>
<td>JOIN4</td>
<td>JOIN4</td>
<td>JOIN4</td>
</tr>
<tr>
<td></td>
<td>community_id</td>
<td>int</td>
<td>200</td>
<td>8</td>
<td>4</td>
<td>private</td>
<td>equality</td>
<td>JOIN3</td>
<td>JOIN3</td>
<td>JOIN3</td>
<td>JOIN3</td>
<td>JOIN3</td>
<td>JOIN3</td>
</tr>
<tr>
<td>policyauthority2nation</td>
<td>authority_id</td>
<td>int</td>
<td>208</td>
<td>16</td>
<td>8</td>
<td>public</td>
<td>JOIN5</td>
<td></td>
<td>JOIN5</td>
<td></td>
<td>JOIN4</td>
<td>JOIN4</td>
<td>JOIN4</td>
</tr>
<tr>
<td></td>
<td>nation_id</td>
<td>int</td>
<td>200</td>
<td>8</td>
<td>4</td>
<td>private</td>
<td>equality</td>
<td>JOIN4</td>
<td></td>
<td></td>
<td>JOIN4</td>
<td>JOIN4</td>
<td>JOIN4</td>
</tr>
</tbody>
</table>
Jana Data Insertion Timings

- Insertion time variations with data schema privacy settings
  - Solid lines using base private schema
  - Dashed line is person table with public name fields (70% size reduction)
  - Dotted lines using all public schema
- Linear insert scalability with DB size implies handling big data possible
- Constant insert time/byte implies no scale overhead
  - Insert variations among tables appear due to private data size & handling
  - Person table has largest records (X100), slowest times/byte
- Additional investigations are needed to better understand these factors
Jana Query Timings

- 5 Queries were tested initially, based on the pandemic scenario
  - Queries 0-2 are aggregations and use MPC emulation regardless of the Jana settings
  - Queries 3 & 4 are specific data requests and use Jana’s newer SPDZ based MPC for enhanced privacy
    - SPDZ off (emulated) is shown in dashed lines for comparison
- Again, highly linear scalability performance implies big data handling possible
- Query 4 is a much more stressing use case
  - Nearly twice as many joins on private columns as the other queries
  - Contains an inner query joined with the outer on a private key column (O(N^2) operation)
- Need to more carefully consider use of private DB keys vs privacy implications
Conclusions

- Jana is proving a useful platform for exploring the feasibility, scalability, flexibility, privacy, and limits of various privacy tools and methods.

- We will continue to explore privacy/efficiency tradeoffs while also seeking to improve the actual tradeoffs incurred by Jana and exploring other use cases.

- More work is needed to fully develop the Jana vision.
Jana: Secure Computation with Differential Privacy, and Applications

Rebecca Wright
Director, DIMACS
Professor, Computer Science Dept., Rutgers University
www.cs.rutgers.edu/~rebecca.wright

DIMACS/Northeast Big Data Hub Workshop on Overcoming Barriers to Data Sharing
October 23-24, 2017