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

RUTGERS
THE STATE UNIVERSITY OF NEW JERSEY

DIMACS
Center for Discrete Mathematics & Theoretical Computer Science
Founded as a National Science Foundation Science and Technology Center

JANA: PRACTICAL PRIVATE DATA-AS-A-SERVICE

"BENE VIXIT, BENE QUI LATUIT." - OVID

galois | University of BRISTOL | RUTGERS | GEORGE MASON UNIVERSITY

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.

Dave Archer
Data-intensive Systems
Secure Computation

galois

Rebecca Wright
Differential Privacy
Applied Cryptography

RUTGERS

Dov Gordon
Scalable Secure
Computation

GEORGE MASON
UNIVERSITY

David Cash
Public Key
Cryptography

Anand Sarwate
Differential Privacy
Machine Learning

University of BRISTOL

Nigel Smart
Cryptography
Secure Computation

Talk Outline

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

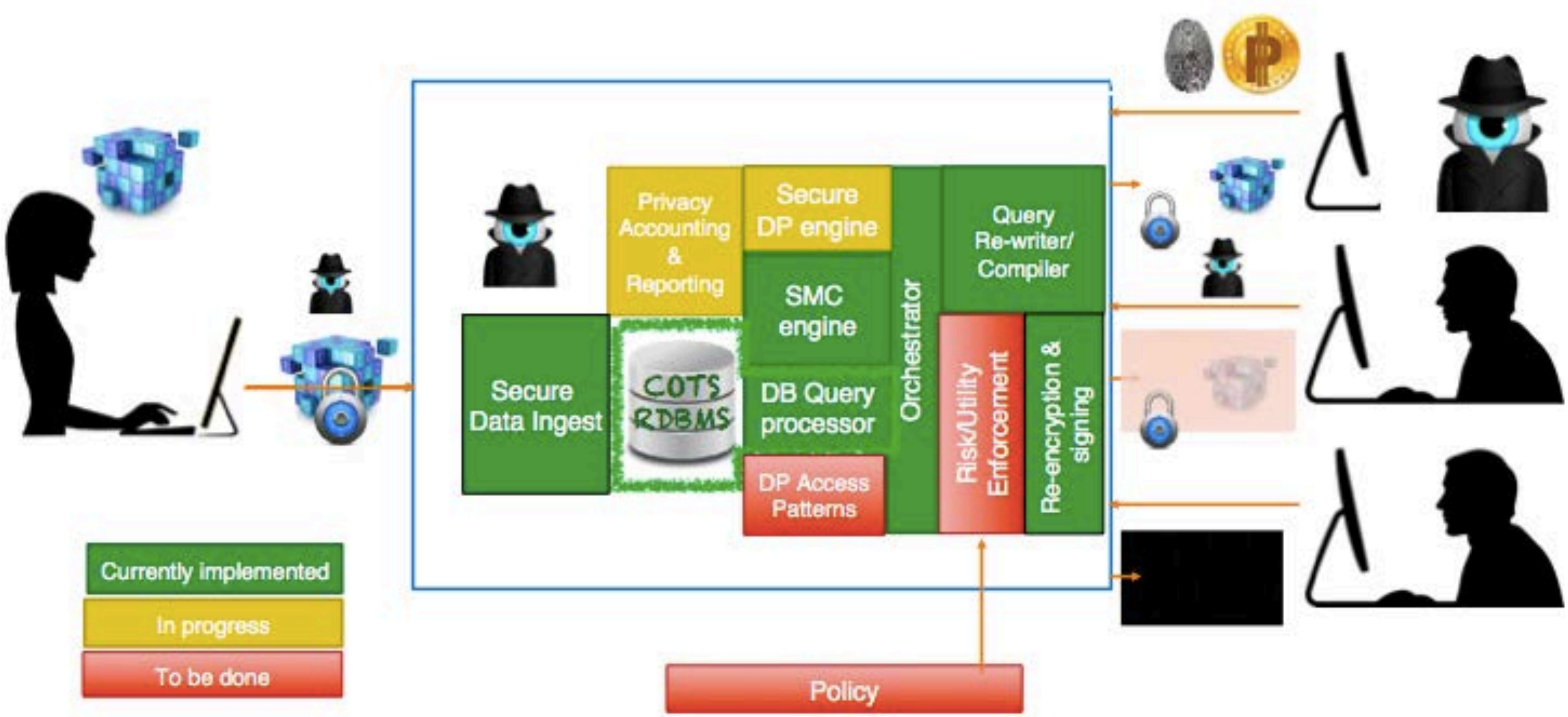
Private Data as a Service

- 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, DKLPSS13]
- possibility of using order-revealing encryption or other deterministic encryption to make some kinds of queries much faster [AKSX04, 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.

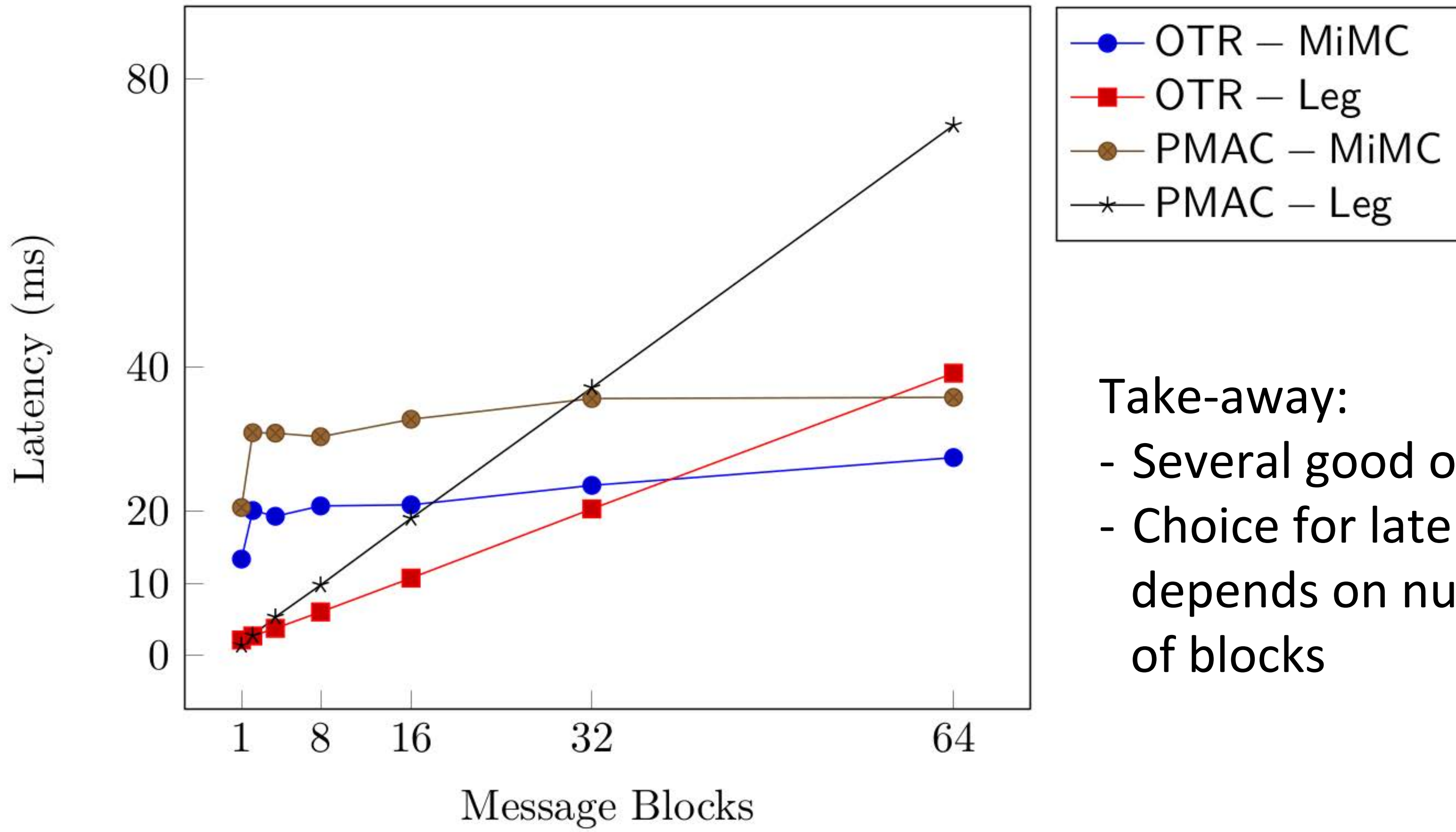
- Problem: The noise for differential privacy, as well as many functions we might want to compute make use of non-finite-field operations.
  - Goal: MPC-friendly differential privacy
  - For noise, currently using variant of [DKMMN06].

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

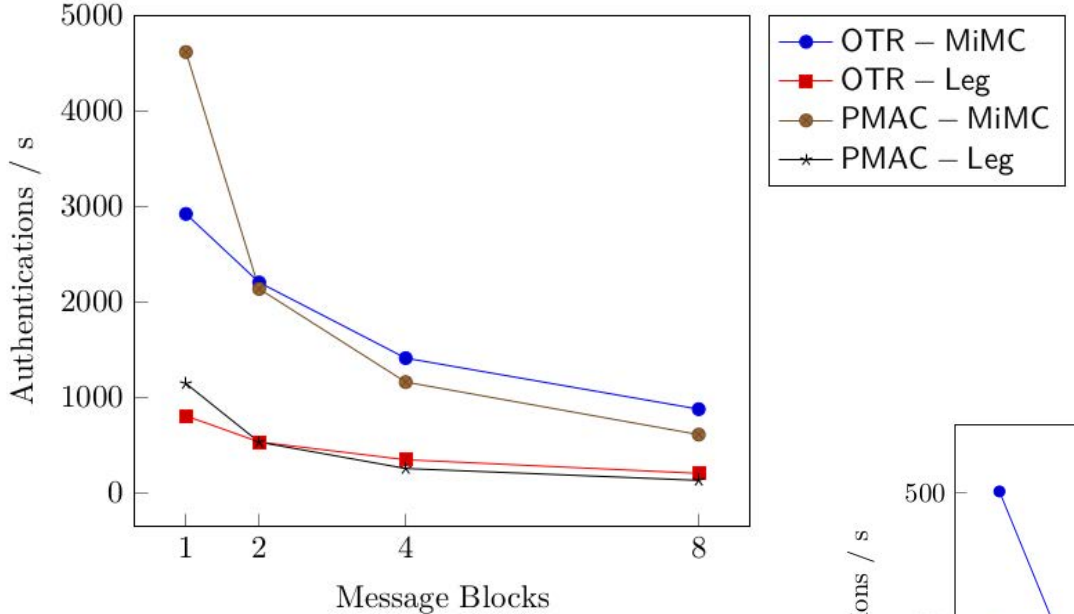
Encryption time

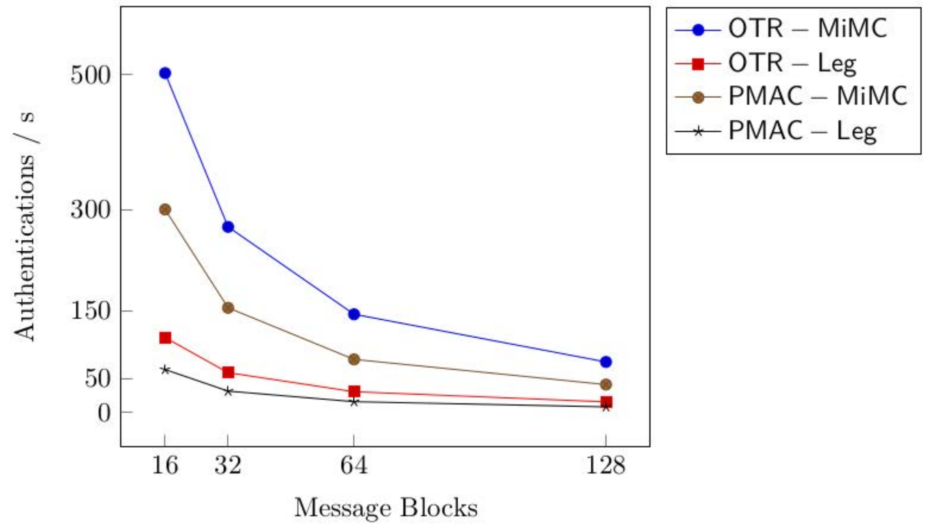


Take-away:

- 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

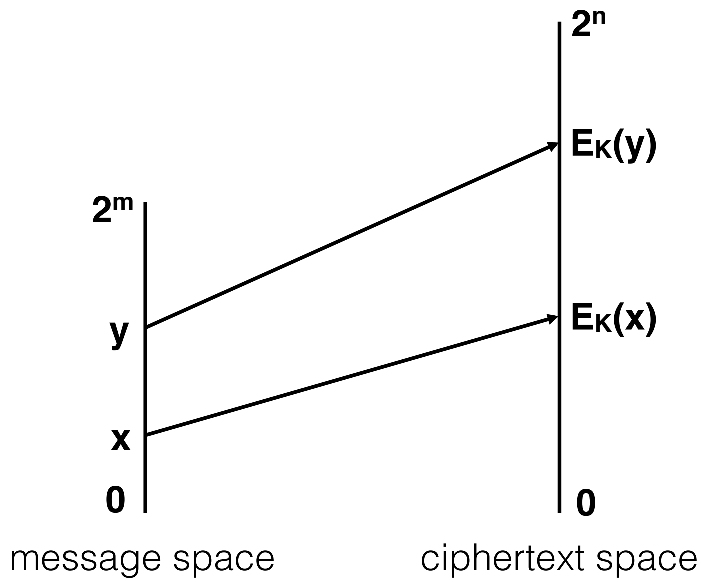
- Problem: We want symmetric encryption that can be efficiently computed “inside” the MPC.
  - Results: MPC-friendly symmetric encryption [GRRSS16]

- 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 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

Zip
686065
48eb42
26861e
01c36e

VS.

First Name	Last Name	Zip	D.O.B
6d9737	a22844	686065	5ad287
9d8ea6	753996	48eb42	abd94c
10eca7	b6b59c	26861e	405702
d99ff8	a2e2a0	01c36e	0abd94

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

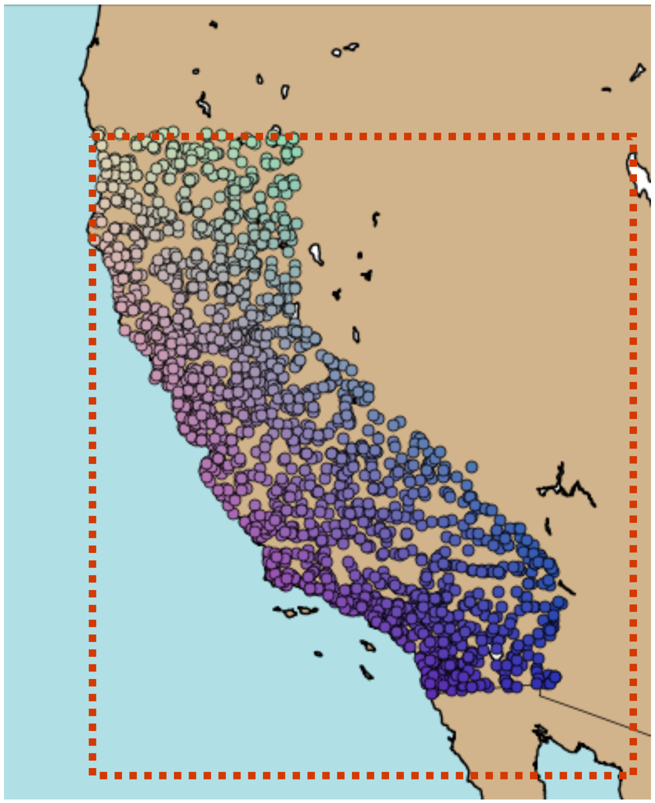
Attacks on ORE with Non-Uniform Data

Experiments on geolocation and time stamps.

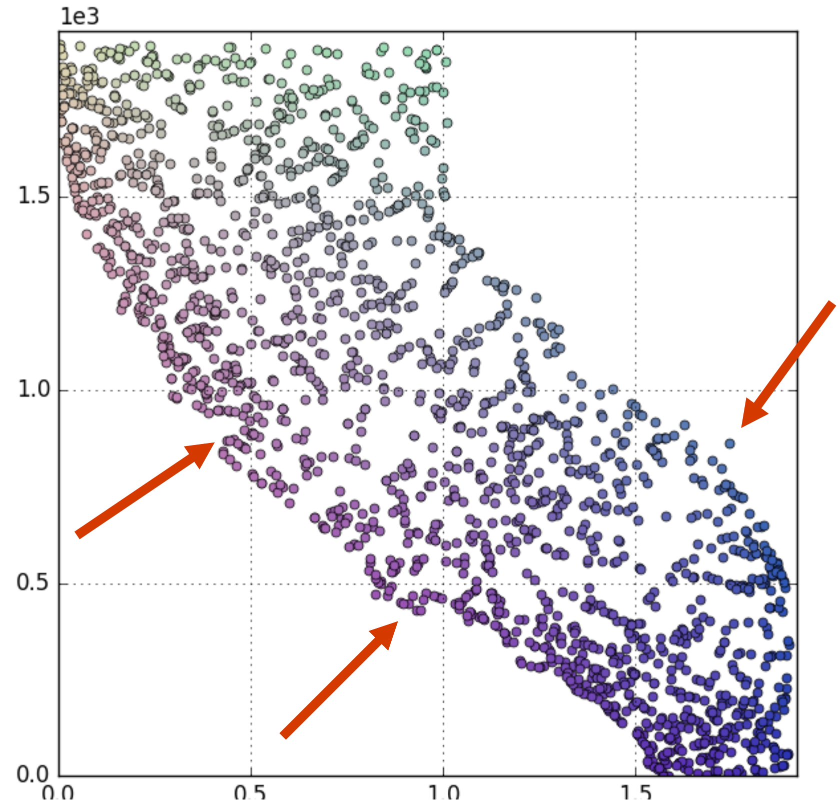
Meta-Conclusion: Need to cryptanalyze definitions/models for secure-but-leaky ORE in practice.

Case Study: California Road Intersections

Plaintexts



Ideal Leakage



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:

```
01x01010011011011xxxxxxxxxx
01x00010010100x10xxxxxxxxxx
01x0011000011001xxxxxxxxxxx
10x0011010x00111xxxxxxxxxxx
01x001010101111xxx0xxxxxxxx
10x010110001010x10xxxxxxxxx
01x010011000 1x1xxxxxxxxxxxx
```

…

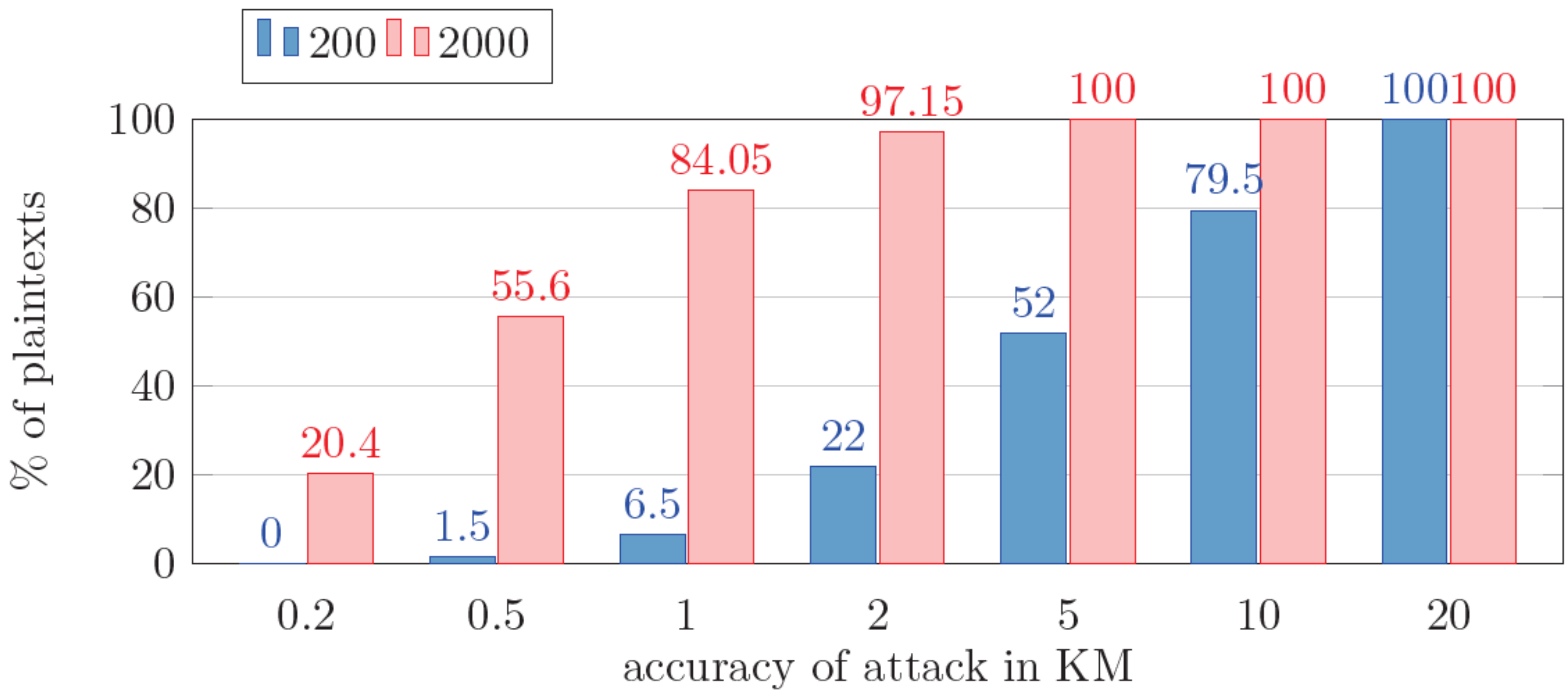
Guessing algorithm:

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

Visualized with "x ⟼ 0.5":

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.

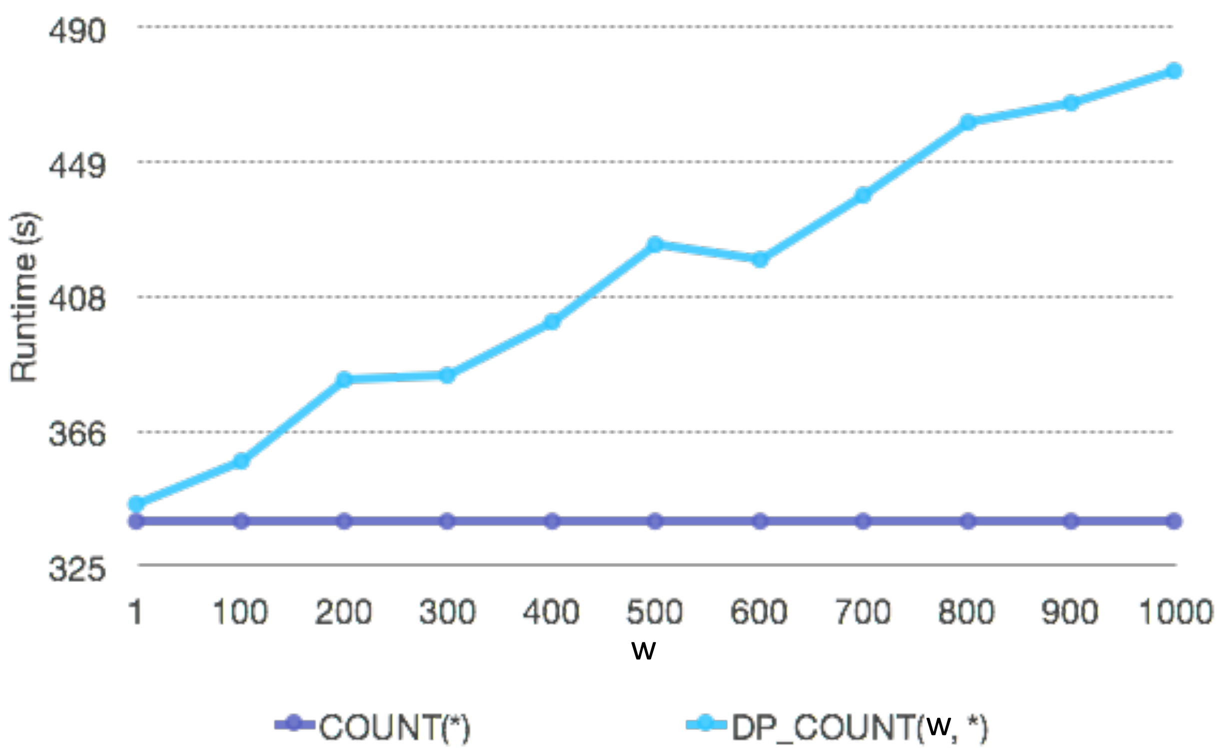
- Problem: The noise for differential privacy, as well as many functions we might want to compute make use of non-finite-field operations.
  - Goal: MPC-friendly differential privacy
  - For noise, currently using variant of [DKMMN06].

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



Privacy Budgeting

- 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

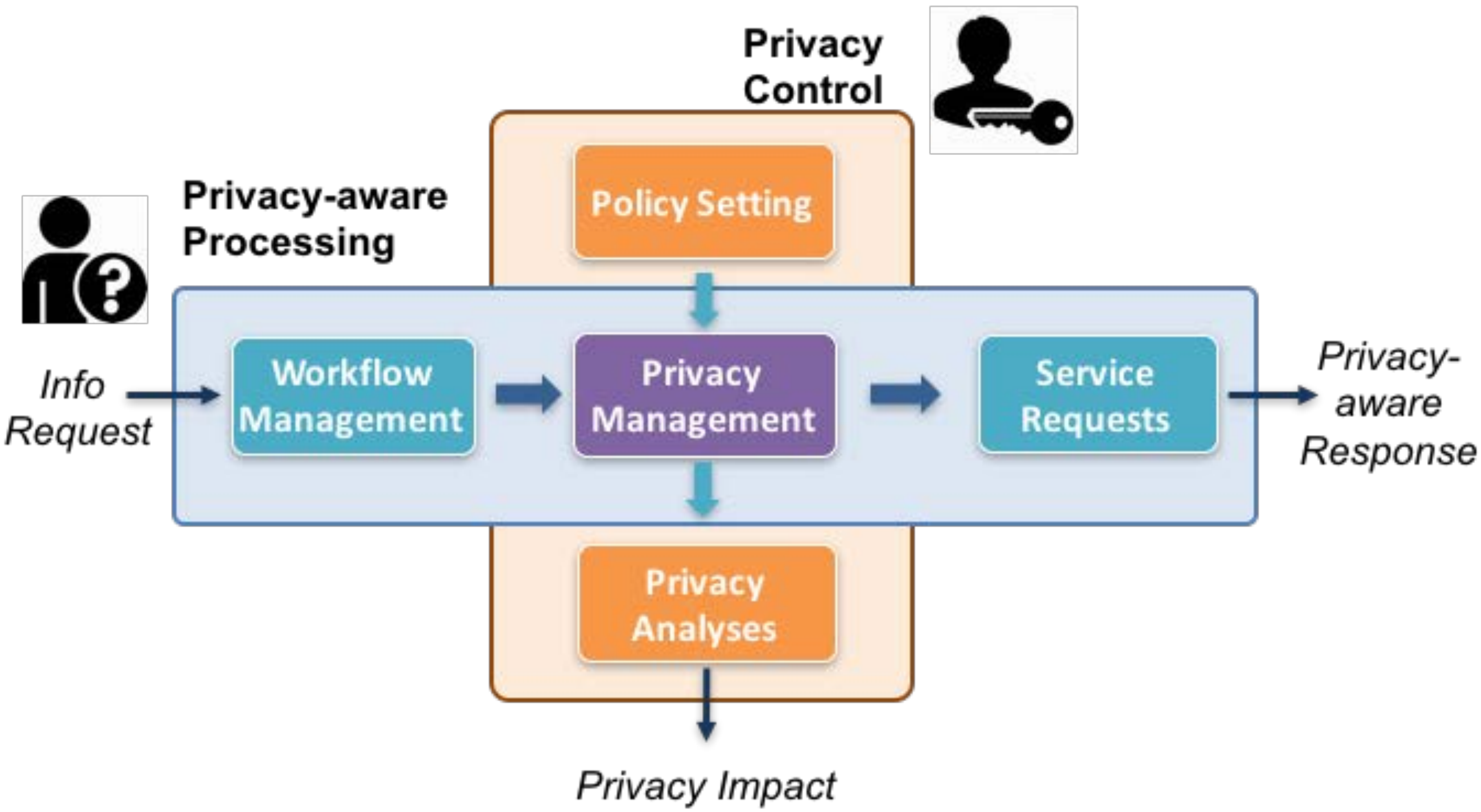
Karen Myers (PI) [slides used with permission]
Tim Ellis
Tancrède Lepoint
SRI International

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.

SRI International®

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

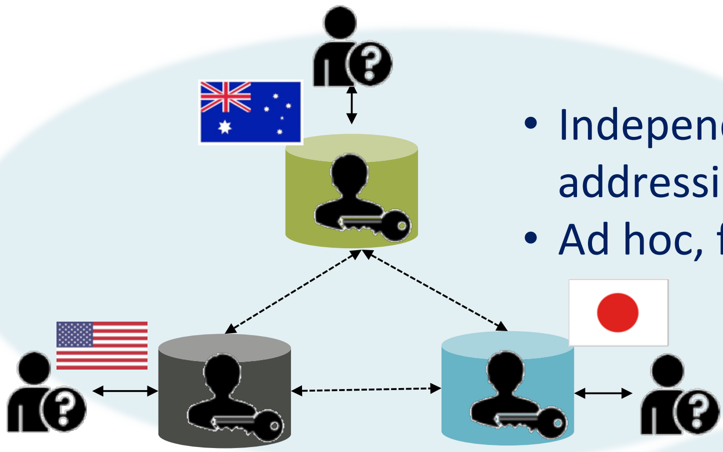
- Distributed storage; access gated by different providers
- Large volumes, possibly streaming
- Much unstructured data
  - text, imagery, PowerPoint

<u>Privacy Tradeoff</u>

Benefits of Sharing

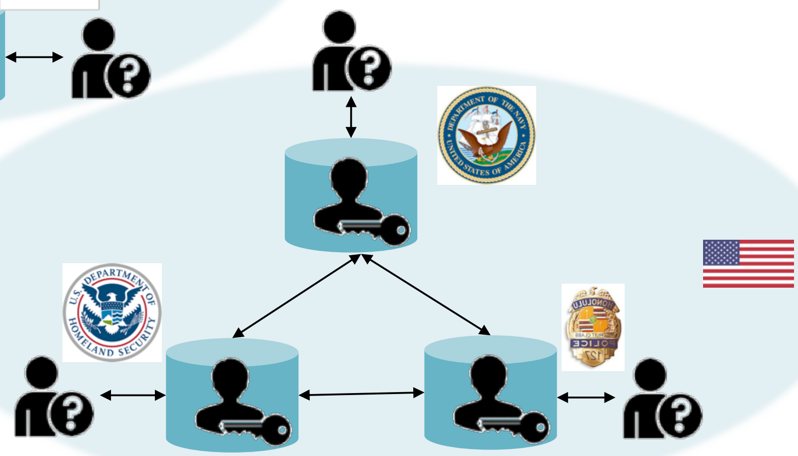
Risks of Sharing

Enterprise Privacy Models



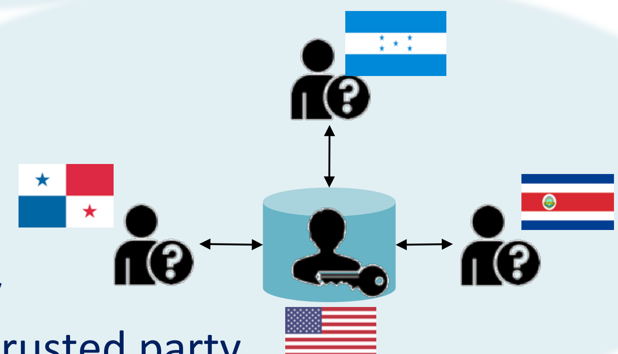
Cross Enterprise

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



Within Enterprise

- Trusted partners within a single over-arching organization; regulations restrict sharing
- Fixed, federated data access model

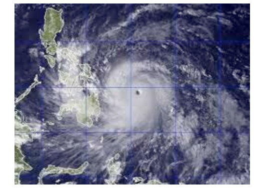


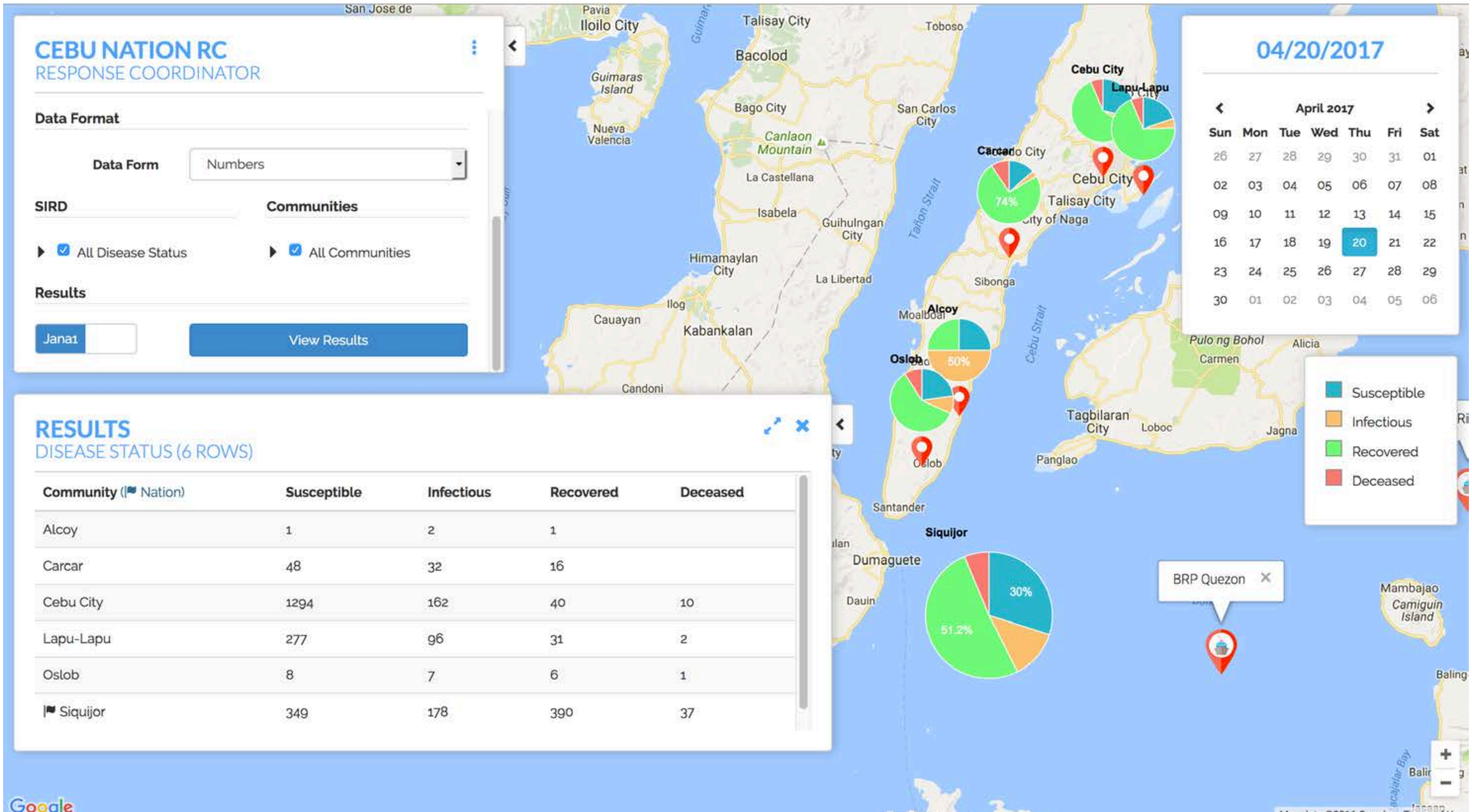
Trusted Broker

- 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

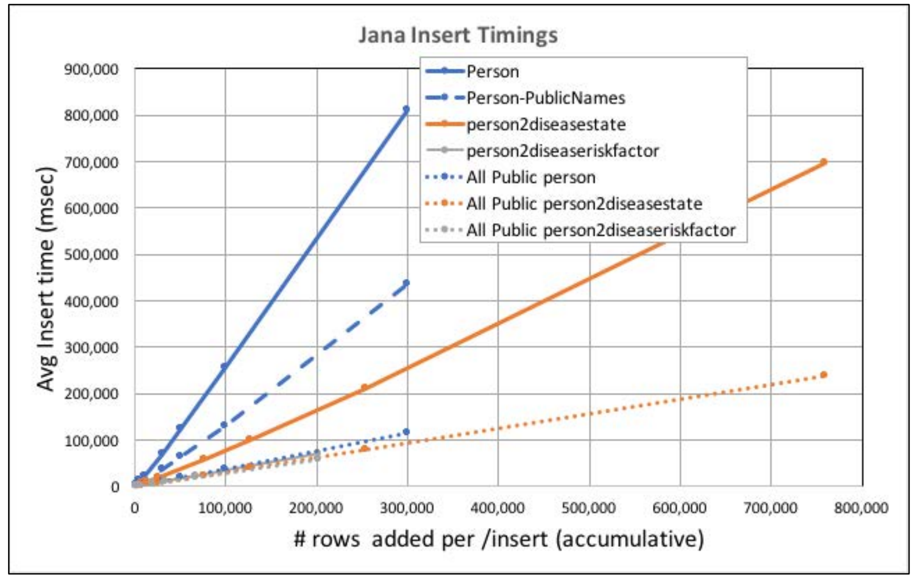
Jana Pandemic Schema & Query Characteristics

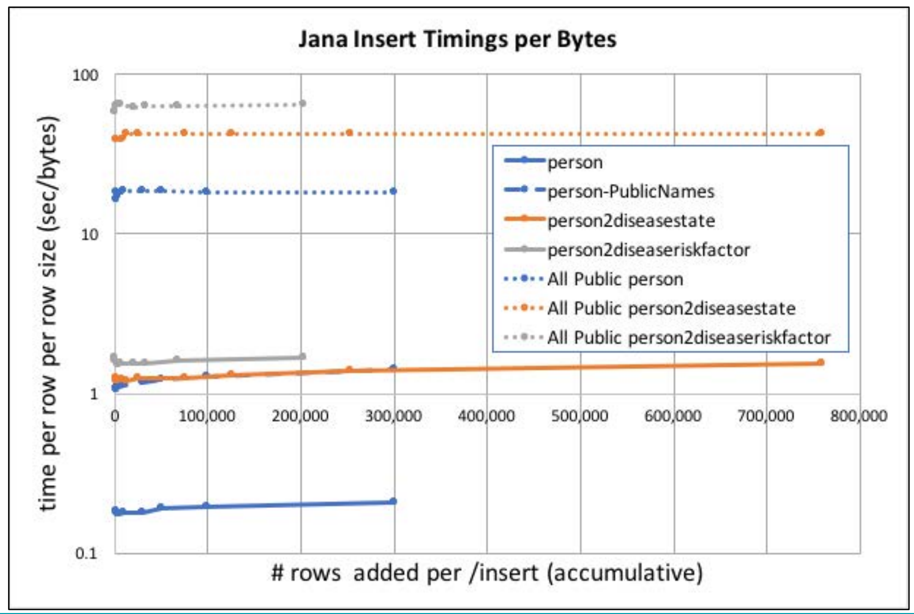
- Private columns (highlighted) in Jana pandemic schema, require encryption & MPC overhead

Table	Column	Type	Jana bytes	Pub bytes	SQL bytes	Priv/Pub	Priv Ops	Query 0	Query 1	Query 2	Query 3	Query 4	Query 4
								COUNT(*)	COUNT(*)	COUNT(*)	Specific Data	Outer	Inner
community			**426**	**42**	**34**								
	community_id	int	200	8	4	private	equality	JOIN1	JOIN1/JOIN3	JOIN1	JOIN1	JOIN1/JOIN3	JOIN1/JOIN3
	community_name	string	10	10	10	public		SELECT/GROUP1	SELECT/GROUP1	SELECT/GROUP1	SELECT		
	latitude	Lat	8	8	8	public		SELECT/GROUP2	SELECT/GROUP2	SELECT/GROUP2			
	longitude	Lon	8	8	8	public		SELECT/GROUP3	SELECT/GROUP3	SELECT/GROUP3			
	nation_id	int	200	8	4	private	equality						
nation			**224**	**32**	**28**								
	nation_id	int	200	8	4	private	equality			JOIN3/JOIN4	JOIN2		
	nation_name	string	8	8	8	public					SELECT		
	latitude	Lat	8	8	8	public							
	longitude	Lon	8	8	8	public							
person			**13000**	**56**	**44**								
	person_id	int	200	8	4	private	equality	JOIN2	JOIN2	JOIN2	JOIN3	SELECT/JOIN2 >	< SELECT/JOIN2
	lastname	string	6000	8	8	private					SELECT	SELECT	
	firstname	string	6000	8	8	private					SELECT	SELECT	
	birthdate	Date	200	8	8	private	order				SELECT	SELECT	
	gender	string	200	8	8	private	equality					SELECT	
	residence	int	200	8	4	private	equality	JOIN1	JOIN1	JOIN1	JOIN1	JOIN1	JOIN1
	citizenship	int	200	8	4	private	equality			JOIN3	JOIN2		
person2diseaseriskfactor			**208**	**16**	**8**								
	riskfactor_id	int	8	8	4	public							
	person_id	int	200	8	4	private	equality				JOIN3		
person2diseasestate			**600**	**24**	**16**								
	diseasestate	string	200	8	4	private	equality	SELECT/GROUP4	SELECT/GROUP4	SELECT/GROUP4	SELECT/WHERE	SELECT/WHERE	WHERE
	person_id	int	200	8	4	private	equality	JOIN2	JOIN2	JOIN2		JOIN2	JOIN2
	transitiondate	Date	200	8	8	private	order	WHERE	WHERE	WHERE	WHERE	WHERE	WHERE
policyauthority			**28**	**28**	**24**								
	authority_id	int	8	8	4	public			JOIN4	JOIN5			
	authority	string	20	20	20	public			WHERE	WHERE		JOIN4/WHERE	JOIN4/WHERE
policyauthority2community			**208**	**16**	**8**								
	authority_id	int	8	8	4	public			JOIN4			JOIN4	JOIN4
	community_id	int	200	8	4	private	equality		JOIN3			JOIN3	JOIN3
policyauthority2nation			**208**	**16**	**8**								
	authority_id	int	8	8	4	public				JOIN5			
	nation_id	int	200	8	4	private	equality			JOIN4			

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

RUTGERS
THE STATE UNIVERSITY
OF NEW JERSEY

DIMACS
Center for Discrete Mathematics & Theoretical Computer Science
Founded as a National Science Foundation Science and Technology Center