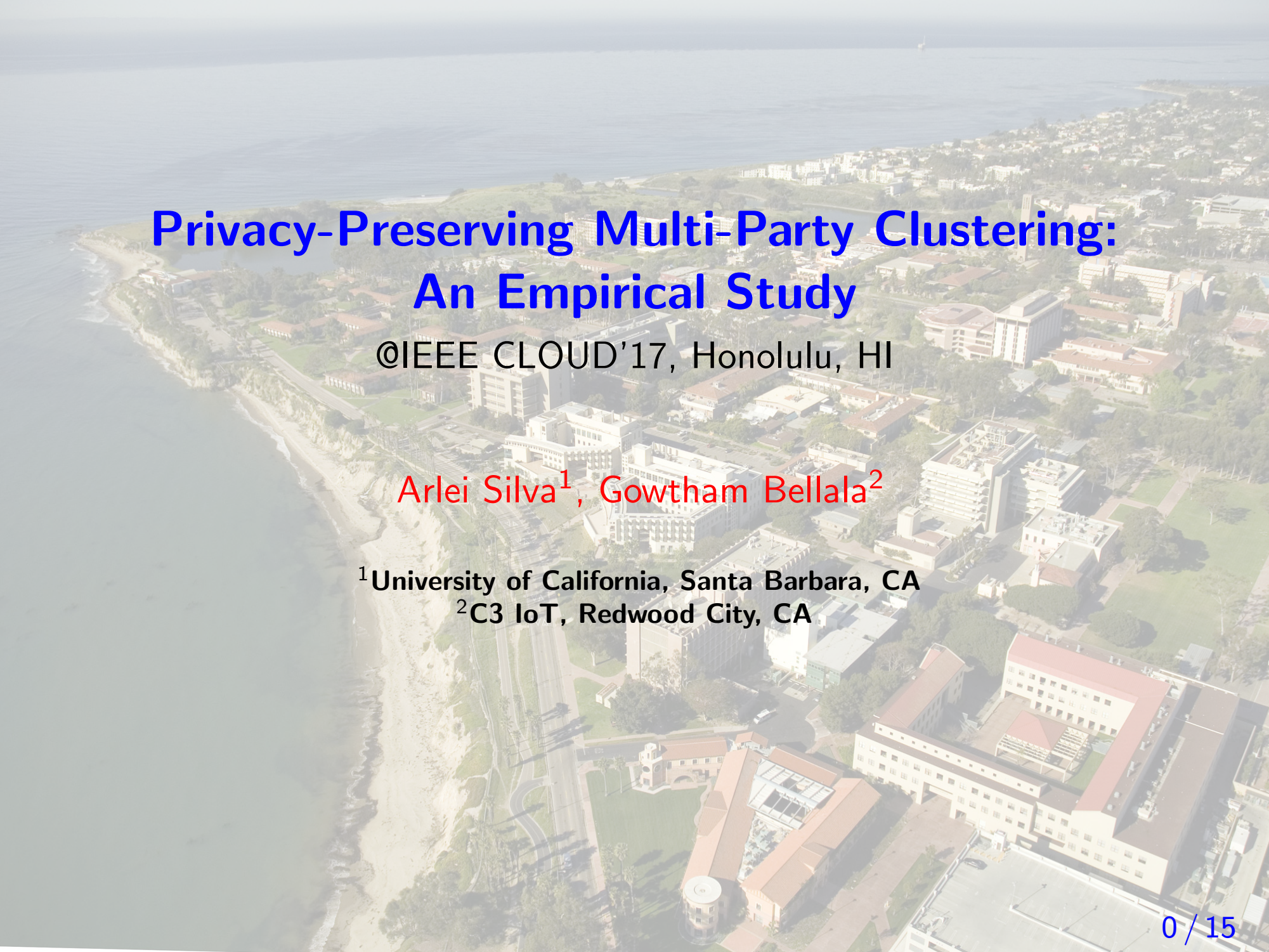


Privacy-Preserving Multi-Party Clustering: An Empirical Study

@IEEE CLOUD'17, Honolulu, HI

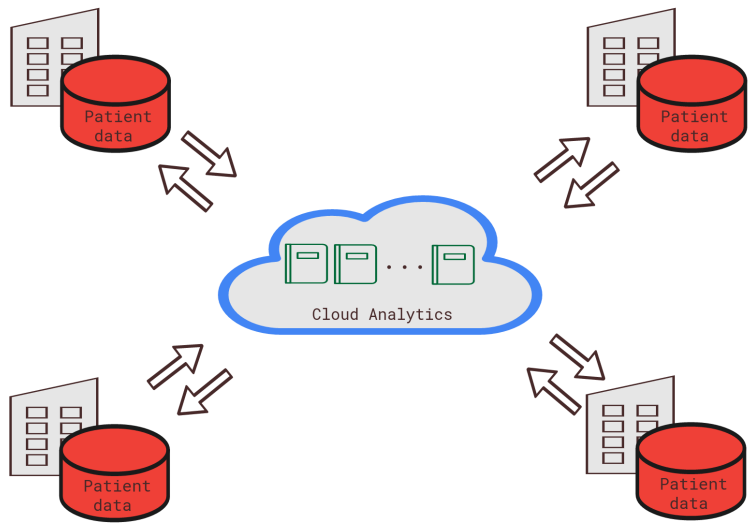
Arlei Silva[1], Gowtham Bellala[2]

[1]University of California, Santa Barbara, CA
[2]C3 IoT, Redwood City, CA

Privacy in Multi-Party Data Analytics

Computation involving multiple parties

- Business processes across companies
- Research collaboration

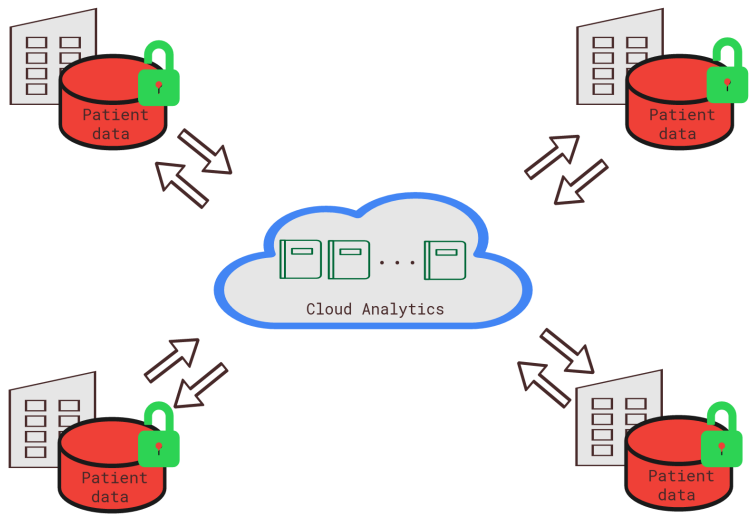


E.g. Multi-Party Healthcare Analytics

Privacy in Multi-Party Data Analytics

Computation involving multiple parties

- Business processes across companies
- Research collaboration



E.g. Multi-Party Healthcare Analytics

How to maintain the privacy/security of the data?

- Organizations are hesitant to share data with third parties
- Massive data breaches (e.g., Target and Boston Medical)

This Work

Comprehensive empirical study of several existing approaches for privacy-preserving multi-party analytics

Case study: clustering task

- Popular task in many applications
- E.g. cohort analysis and information retrieval

Obfuscation techniques

1. Additive data perturbation;
2. Random subspace projection;
3. Secure multi-party computation

Centralized vs. distributed settings

Trade-off between quality, privacy, and performance

- Multiple evaluation metrics and datasets
- Under same framework and settings

Multi-Party Clustering and K-Means Algorithm

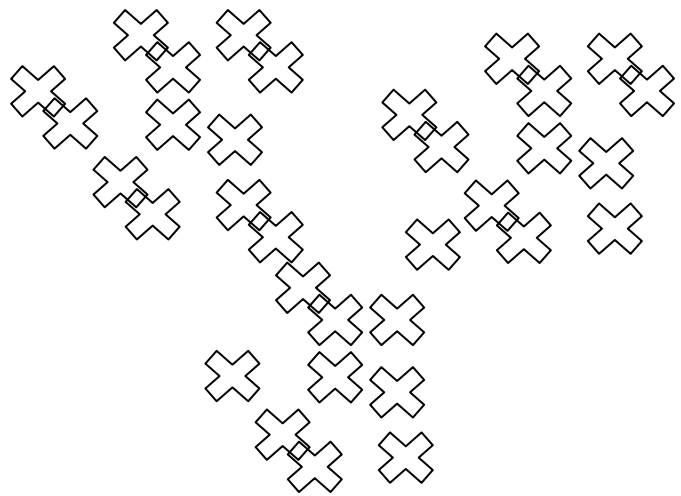
Task: Partition data points into groups based on similarity

Multi-party setting: Data points belong to different parties

- E.g. Each hospital has data from a set of patients
- Horizontal partitioning

K-means Algorithm:

- Most popular approach;
- Iterative;
- Centroid-based.



K-means Algorithm

Multi-Party Clustering and K-Means Algorithm

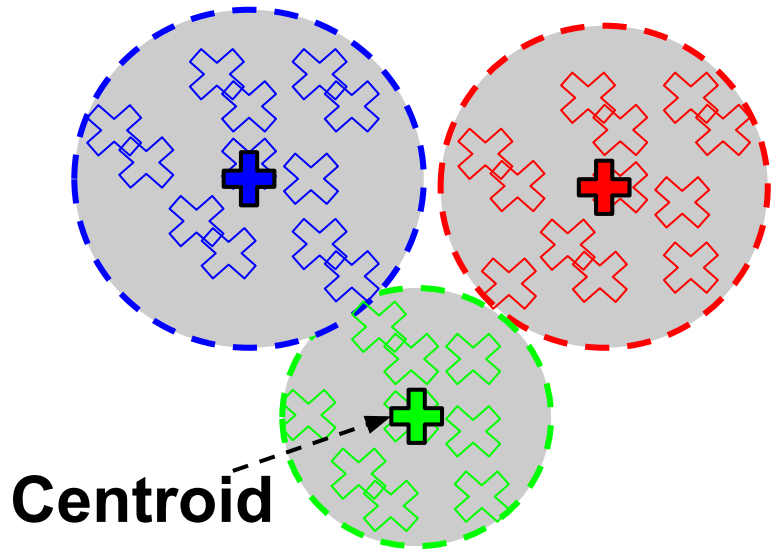
Task: Partition data points into groups based on similarity

Multi-party setting: Data points belong to different parties

- E.g. Each hospital has data from a set of patients
- Horizontal partitioning

K-means Algorithm:

- Most popular approach;
- Iterative;
- Centroid-based.



K-means Algorithm

Multi-Party Clustering and K-Means Algorithm

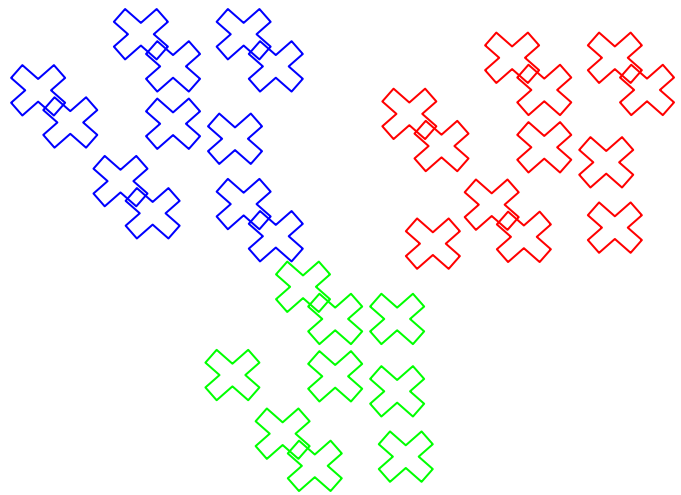
Task: Partition data points into groups based on similarity

Multi-party setting: Data points belong to different parties

- E.g. Each hospital has data from a set of patients
- Horizontal partitioning

K-means Algorithm:

- Most popular approach;
- Iterative;
- Centroid-based.



K-means Algorithm

Privacy-Preserving Clustering Approaches

The Mediator is a third-party that facilitates computation

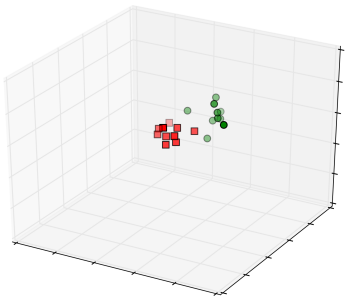
1. Trusted: Operates on *raw data*
2. Untrusted: Operates on *obfuscated data*

Computation	Mediator	Privacy	What is shared
Local	–	–	–
Centralized	Trusted	–	Local data
Centralized	Untrusted	ADP	Perturbed data
Centralized	Untrusted	RSP	Projected data
Distributed	Trusted	–	Partial results
Distributed	Untrusted	ADP	Perturbed results
Distributed	Untrusted	RSP	Projected results
Distributed	Untrusted	SMC	Encrypted results

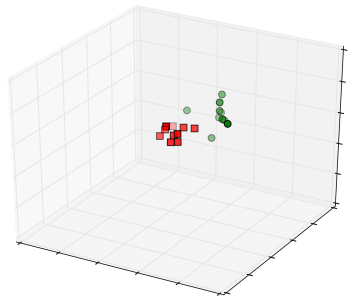
Overview of approaches studied in this paper.

Additive Data Perturbation (ADP)

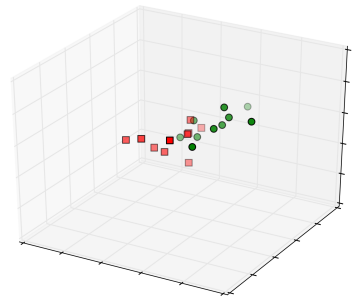
$$\mathbf{x'_{i,j}} = \mathbf{x_{i,j}} + \epsilon, \epsilon \sim \mathbf{N}(\mathbf{0}, \sigma)$$



Original data



Noisy data, $\sigma = .01$



Noisy data, $\sigma = .1$

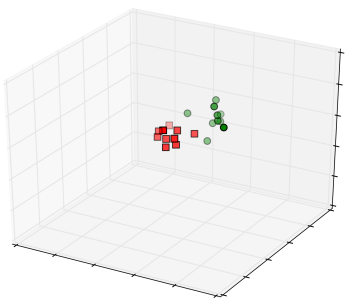
Perturbs data while preserving underlying clusters

Random Subspace Projection (RSP)

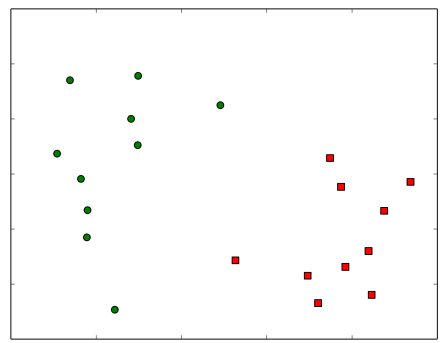
$$\mathbf{x}' = \frac{\mathbf{1}}{\sqrt{\mathbf{q}}\sigma}\mathbf{x}\mathbf{R}, \mathbf{r_{i,j}} \sim \mathbf{N}(\mathbf{0}, \sigma)$$

q is the number of projected dimensions

R is a random projection matrix



Original data



Projected data, q=2

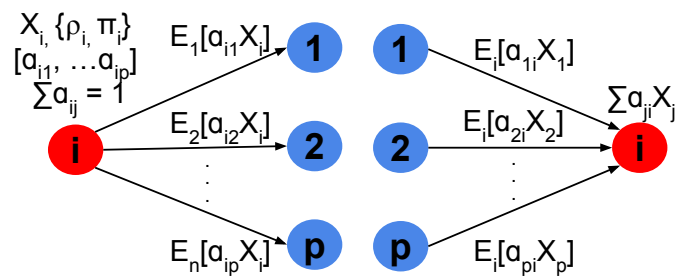
Projects data into a low-dimensional space while preserving underlying clusters

Secure-Multiparty Computation (SMC)

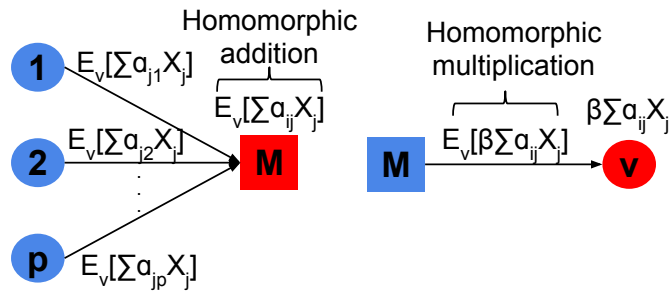
Applied only for the distributed solution

Random sharing and partially homomorphic encryption

Encryption using the Paillier cryptosystem



Secure random sharing



Secure aggregation

Main sub-routines of SMC approach

Privacy-Preserving Multi-Party Clustering

Centralized:

1. Parties agree on obfuscation parameters;
2. Each party obfuscates its local data and shares it with mediator;
3. Mediator computes clusters and returns the results.

Distributed:

1. Parties agree on obfuscation parameters;
2. Parties cluster local data and share obfuscated results with mediator;
3. Mediator aggregates local results and returns to the parties;
4. Parties update centroids;
5. Repeat 2-4 until convergence.

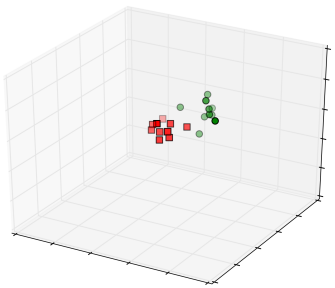
Attacks on Privacy-Preserving Clustering

Goal: reconstruct original data given its obfuscated version

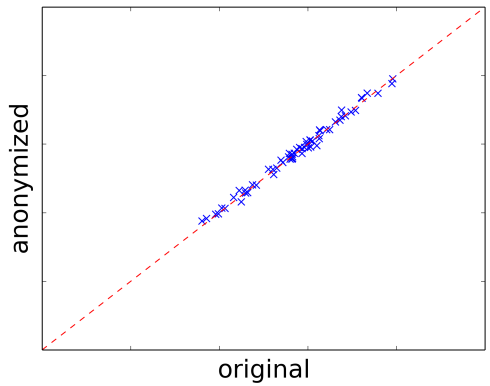
We study attacks on the mediator

For ADP and RSP we consider attacks from the literature

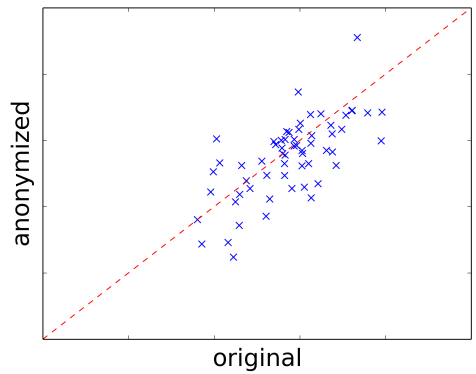
We assume that the SMC approach is secure under the honest-but-curious model with no collusion



Original data



Reconstruction, $\sigma = .01$



Reconstruction, $\sigma = .1$

Example of attack on ADP

(See details in the paper)

Evaluation Metrics

Privacy:

- **Conditional privacy loss;**
- Root mean squared error.

Clustering quality:

- **Intra-cluster distance;**
- Adjusted rand-score.

Computational performance:

- **Running time;**
- Communication.

Testbed and Data

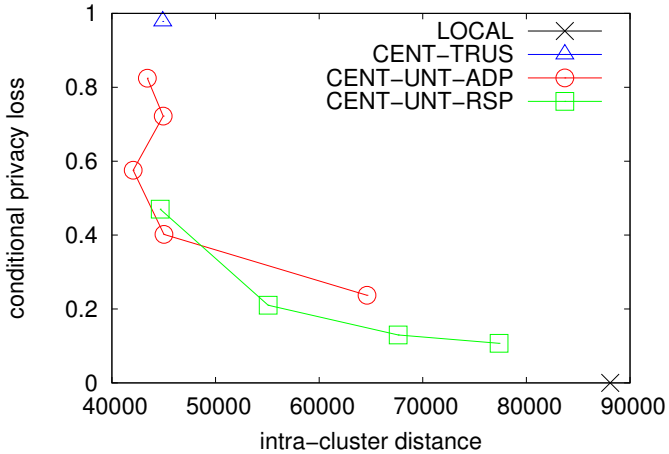
10-16 node Amazon AWS EC2 cluster

All solutions implemented in Python

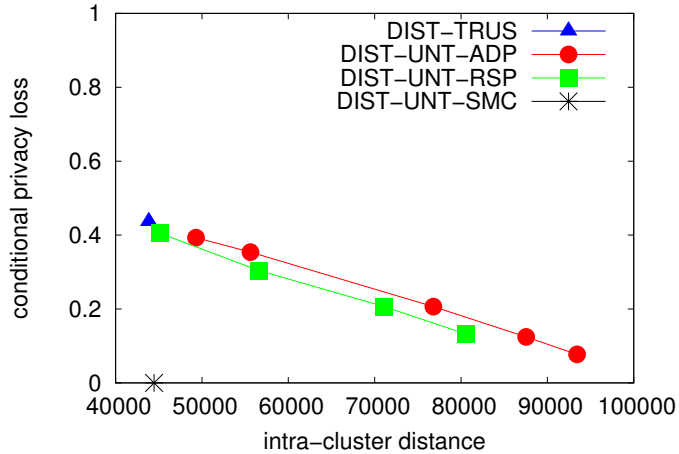
name	# objects	# dimensions	# clusters
SYNTHETIC	50K	10	10
HEART	920	13	4
CANCER	198	20	15
DIABETES	100K	12	12
GAS	320K	16	10

Table : Dataset statistics.

Privacy vs. Quality

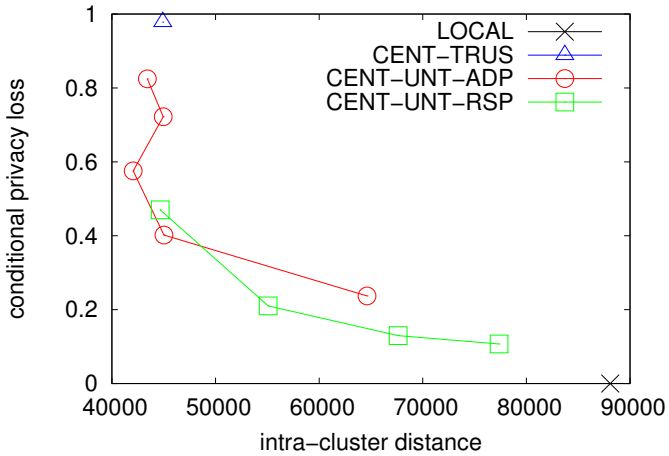


Local+Centralized

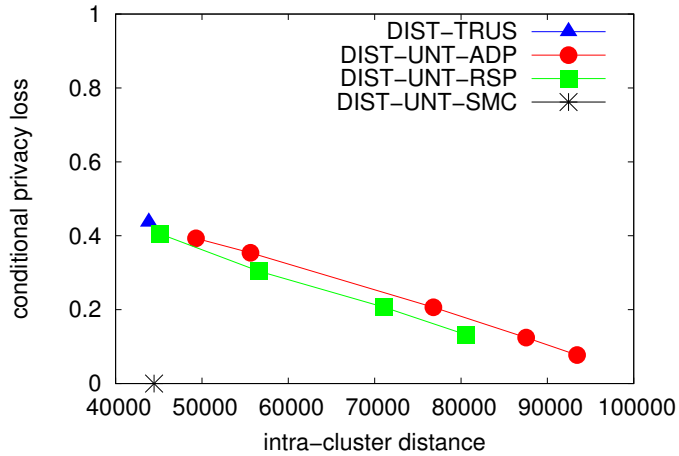


Distributed

Privacy vs. Quality



Local+Centralized



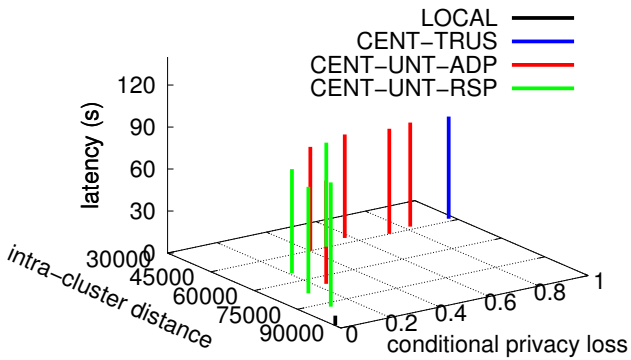
Distributed

RSP often outperforms ADP (up to 1/2 privacy loss);

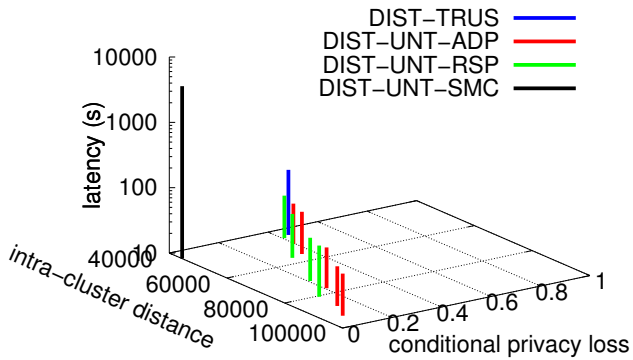
ADP is more flexible (centralized optimal for .6 privacy loss);

Distributed approaches are more private than their centralized counterparts.

Privacy vs. Quality vs. Performance

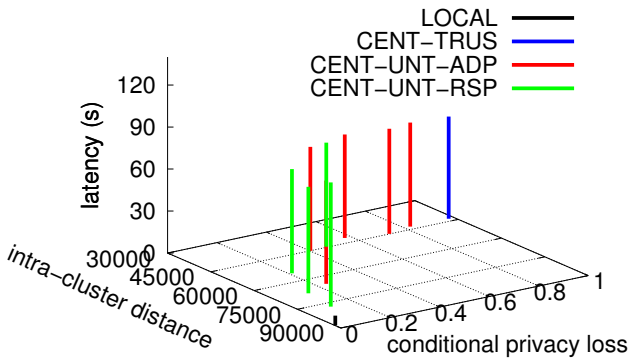


Local+Centralized

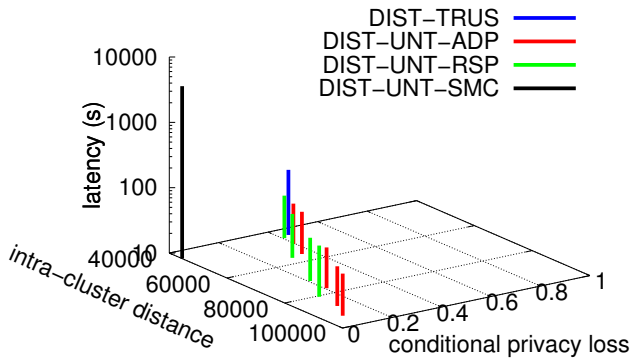


Distributed

Privacy vs. Quality vs. Performance



Local+Centralized

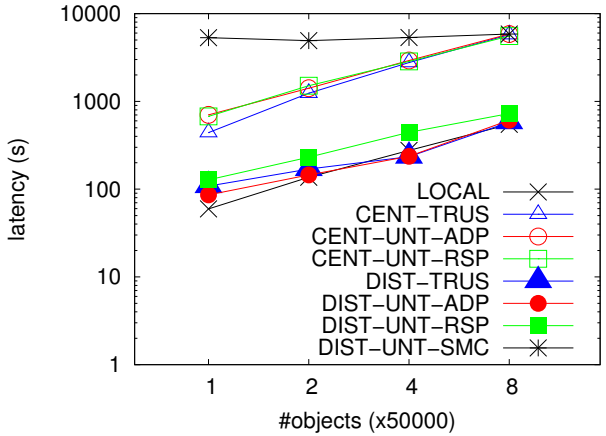


Distributed

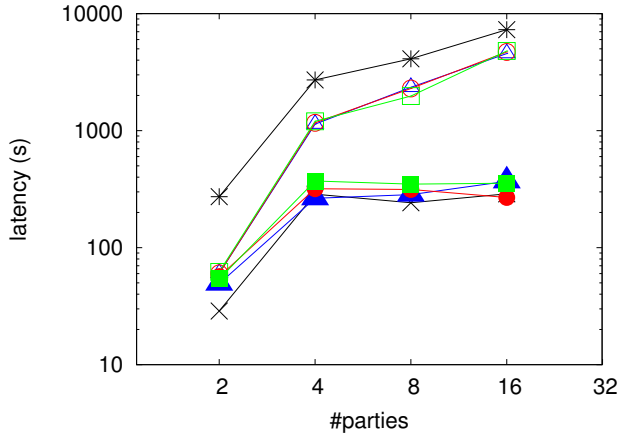
Distributed trusted, ADP and RSP are very efficient

SMC requires 2 orders of magnitude more time and communication

Scalability



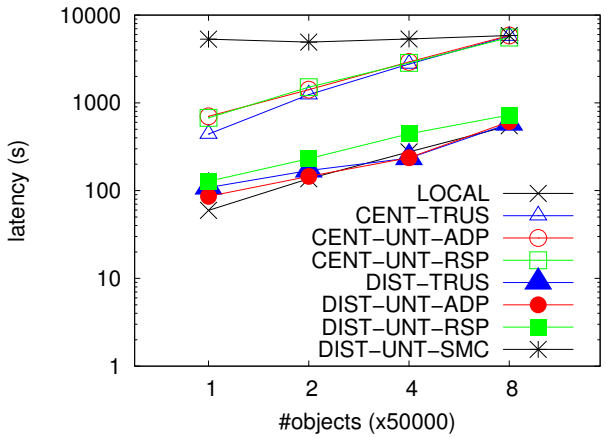
# of objects



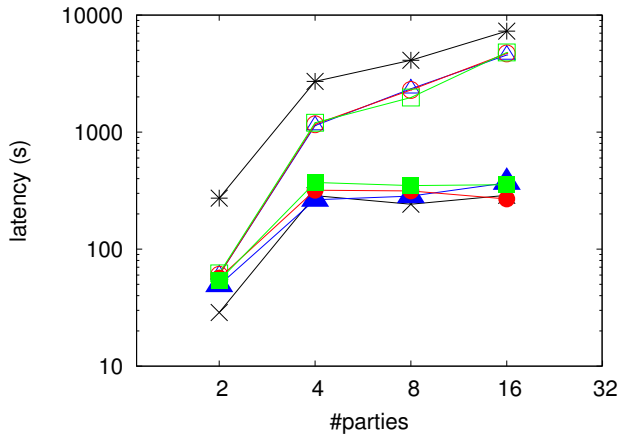
# of parties

Scalability results for SYNTHETIC dataset

Scalability



# of objects



# of parties

Scalability results for SYNTHETIC dataset

Local, dist. trusted, ADP and RSP scale linearly

For centralized approaches, the mediator is a bottleneck

For large data, SMC outperforms the centralized methods

Conclusions

We evaluated several privacy-preserving multi-party clustering strategies that differ in terms of:

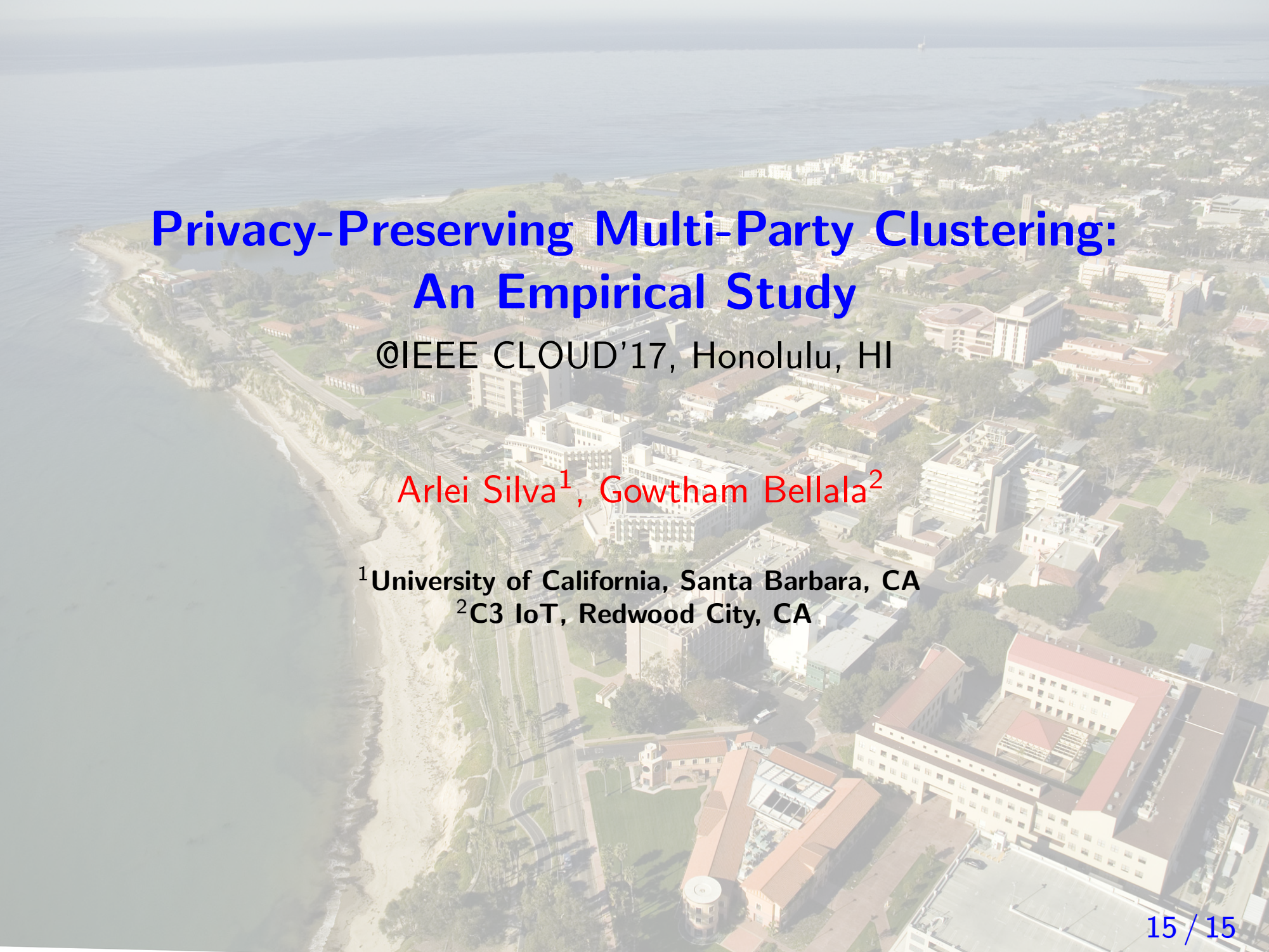
- The computation model used (local, centralized, distributed);
- The type of mediator assumed (trusted and untrusted).

We studied three privacy-preserving techniques:

- Additive data perturbation;
- Random subspace projection;
- Secure multi-party computation.

Main findings:

- RSP outperforms ADP in most of the settings;
- ADP covers a broader privacy versus quality spectrum;
- Distributed approaches are scalable and more private than their centralized counterparts;
- SMC achieves high quality and privacy, but poor performance.



Privacy-Preserving Multi-Party Clustering: An Empirical Study

@IEEE CLOUD'17, Honolulu, HI

Arlei Silva[1], Gowtham Bellala[2]

[1]University of California, Santa Barbara, CA
[2]C3 IoT, Redwood City, CA