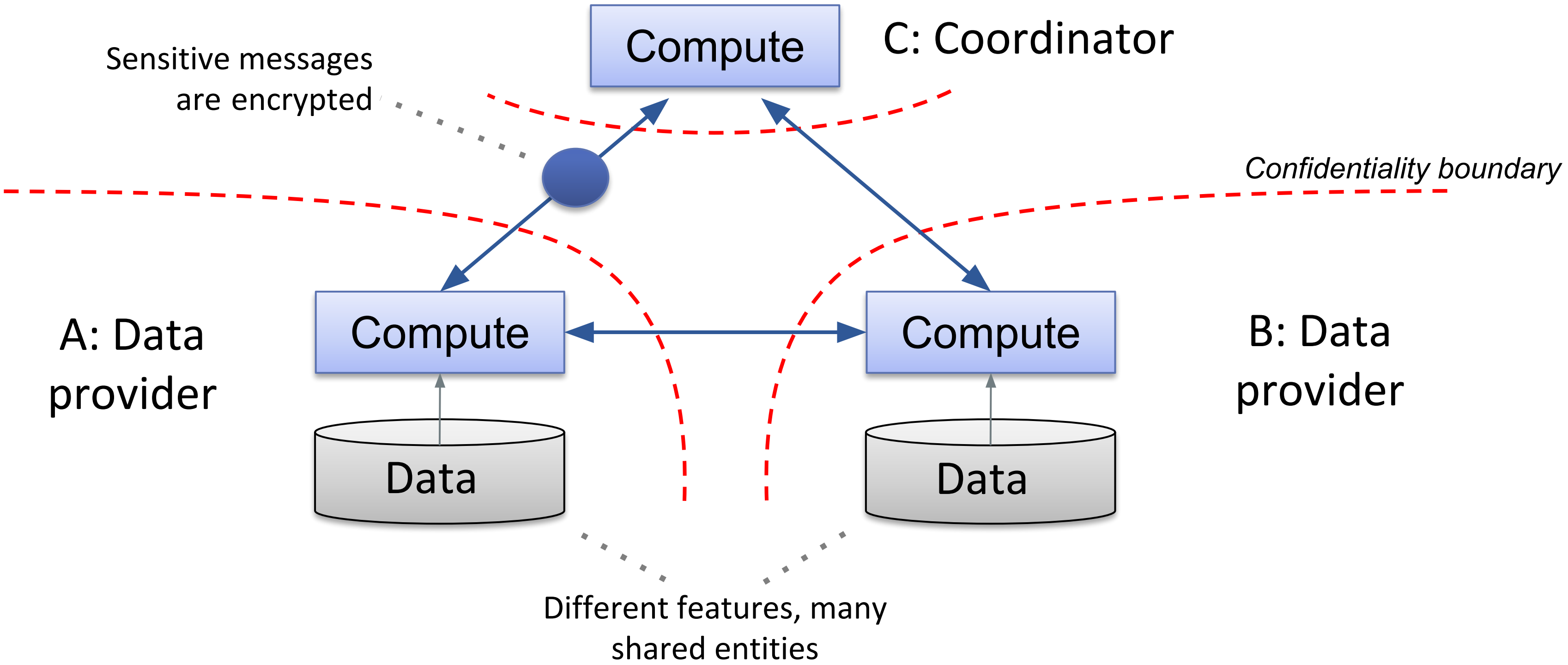
Privacy-preserving entity resolution and logistic regression on encrypted data

Giorgio Patrini &
Mentari Djatmiko, Stephen Hardy, Wilko Henecka, Hamish Ivey-Law,
Maximilian Ott, Huy Pham, Guillaume Smith, Brian Thorne, Dongyao Wu
N1 Analytics @ Data61 CSIRO

PSML workshop, ICML17
11/8/2017, Sydney

N1 Analytics

Scenario & motivation



Secure end to end system

- **Vertical partition** of a dataset: common entities but **different features**
  - One data provider has the *labels*
  - *E.g.* banking and insurance data about common customers; labels are fraudulent activity
- **Goal**: learn a predictive model in the cross-feature space
  - Comparable **accuracy** as if had all data in one place
  - **Scale** to real-world applications

Secure end to end system

- **Vertical partition** of a dataset: common entities but **different features**
  - One data provider has the *labels*
  - *E.g.* banking and insurance data about common customers; labels are fraudulent activity
- **Goal**: learn a predictive model in the cross-feature space
  - Comparable **accuracy** as if had all data in one place
  - **Scale** to real-world applications
- **Constraints**
  - Who is who? ⇨ **Private entity resolution**
  - Raw data remains **private** ⇨ **federated learning + privacy**

Overview

- End-to-end system:
  - **Security assumptions / requirements**
  - Entity resolution
  - Learning on private data
- Deployment & experiments

Security assumptions / requirements

- Participants are **honest-but-curious:**
  - they follow the protocol
  - they are not colluding
  - **but:** they try to infer as much as possible
- Reasonable: participants have an incentive to compute an accurate model.
- **Only the Coordinator holds the private key** used to decrypt messages.
- No sensitive data (raw or aggregated) *leaves* a data provider unencrypted
  - ...but computation uses unencrypted individual records *locally*.

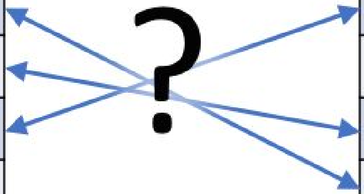
Overview

- End-to-end system:
    - Security assumptions / requirements
    - **Entity resolution**
    - Learning on private data
- Deployment & experiments

Privacy-preserving entity resolution

- **Goal**: match *corresponding* rows in two distinct databases

Name	DOB	...
Klara Jovel	07/09/1942	...
Scott Redo	04/08/1923	...
Tori Mckone	07/06/1921	...
Rusty Brod	25/07/2014	...



Name	DOB	...
Tori Mckone	07/06/1921	...
Scotty Undo	24/01/1965	...
Scott Redo	04/08/1923	...
Clara Jovel	07/09/1942	...

- **Constraint**: can’t share Personally Identifiable Information (PII)

Privacy-preserving entity resolution

- **Goal**: match *corresponding* rows in two distinct databases

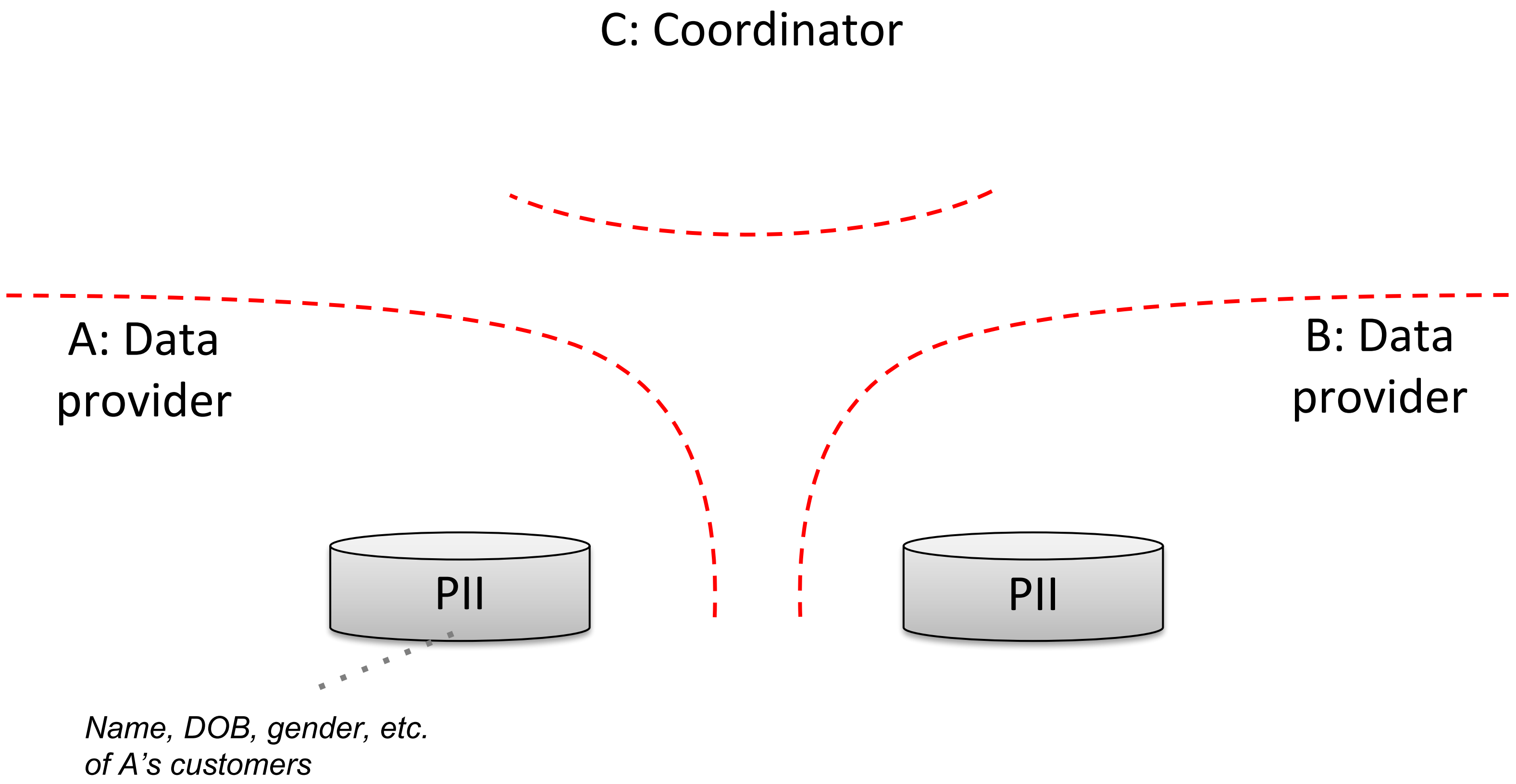
Name	DOB	...
Klara Jovel	07/09/1942	...
Scott Redo	04/08/1923	...
Tori Mckone	07/06/1921	...
Rusty Brod	25/07/2014	...

?

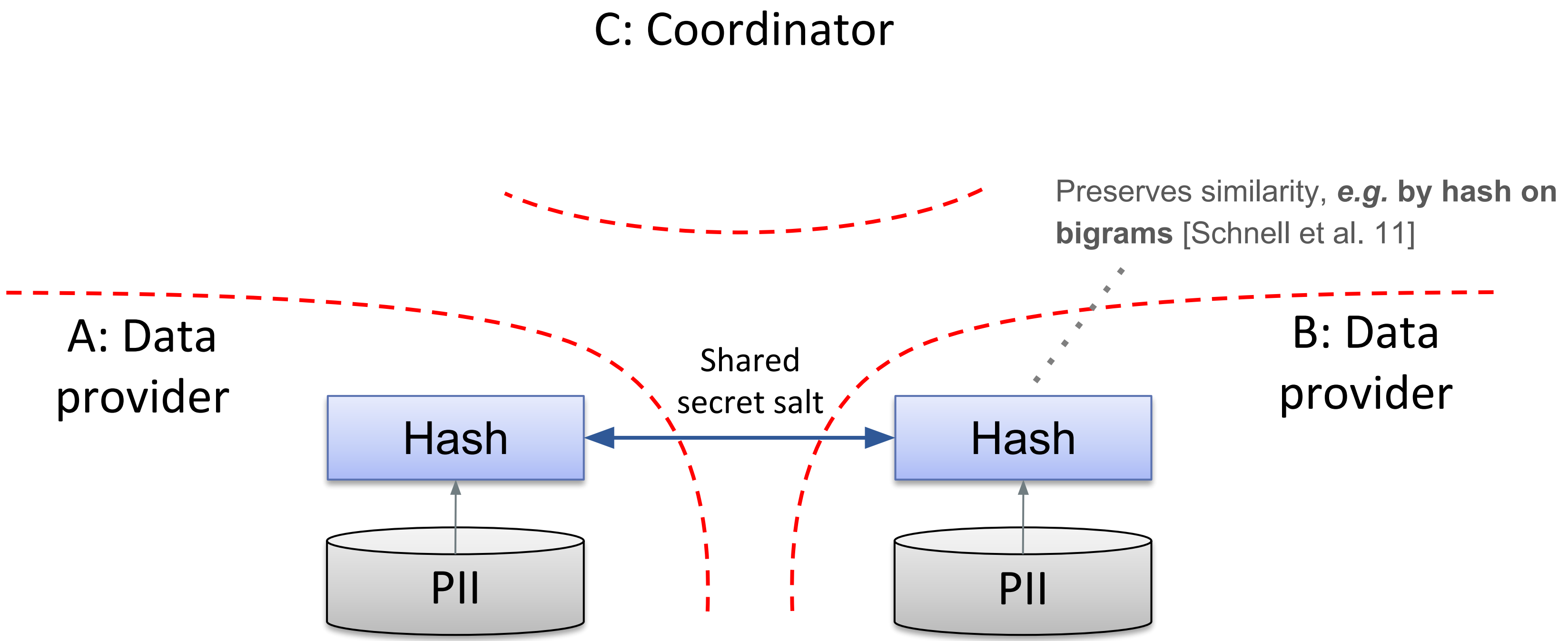
Name	DOB	...
Tori Mckone	07/06/1921	...
Scotty Undo	24/01/1965	...
Scott Redo	04/08/1923	...
Clara Jovel	07/09/1942	...

- **Constraint**: can't share Personally Identifiable Information (PII)
- **Solution**: *fuzzy & private* matching

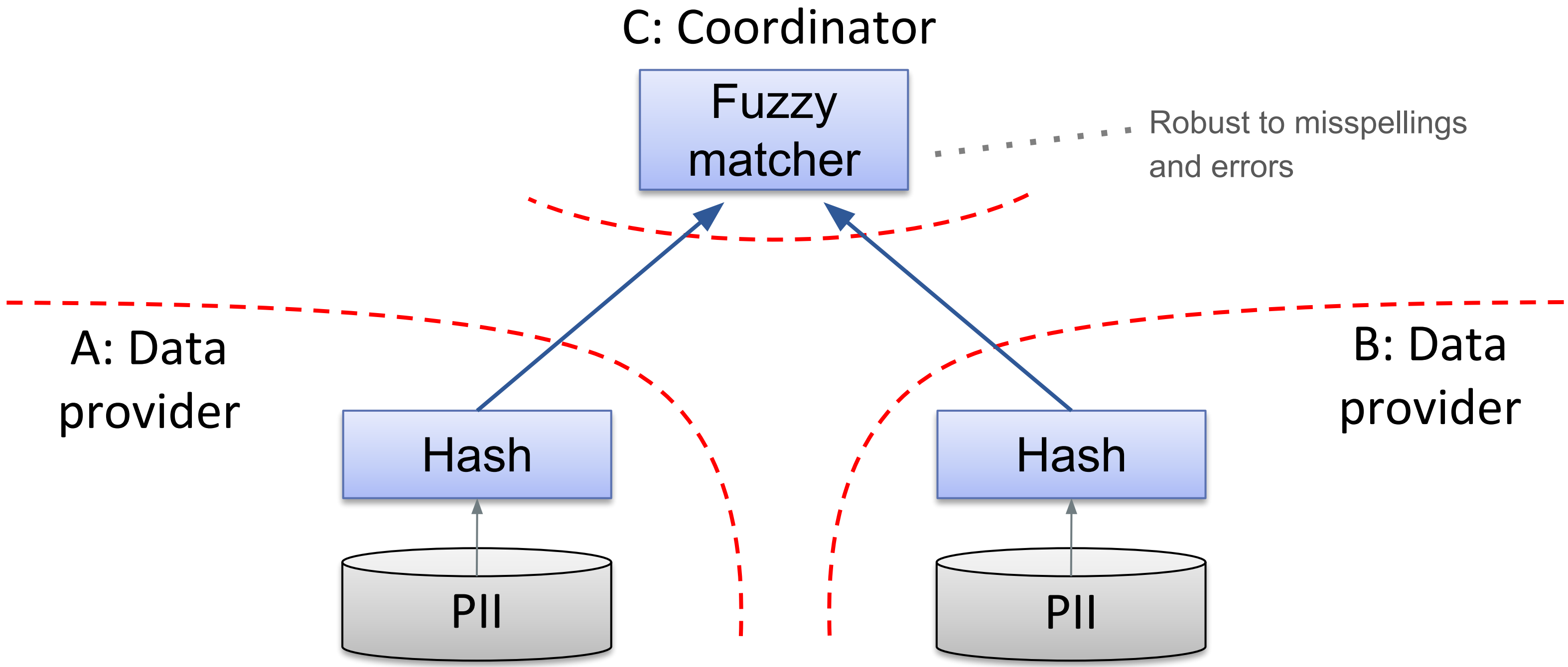
Privacy-preserving entity resolution



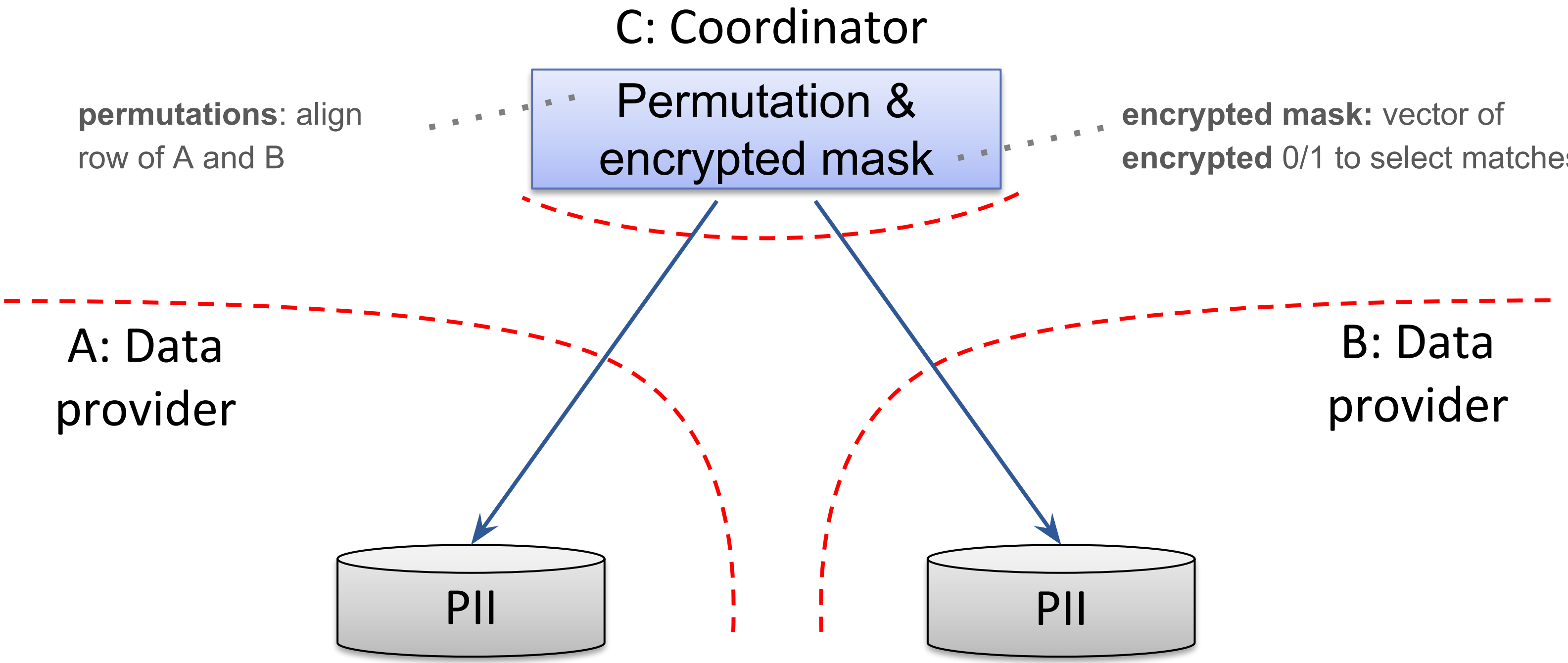
Privacy-preserving entity resolution



Privacy-preserving entity resolution



Privacy-preserving entity resolution: the output



No data provider knows which/how many entities are in common!

Overview

- End-to-end system:
  - Security assumptions / requirements
  - Entity resolution
  - **Learning on private data**
- Deployment & experiments

Background: Paillier Partially Homomorphic Encryption

- $[[u]]$ is the encryption of u
- **Addition**:

$$[[u]] + [[v]] = [[u + v]]$$

- **Scalar multiplication**:

$$n \cdot [[u]] = [[nu]]$$

- Extend to vectors ⇨ **encrypted linear algebra** (almost)!

Background: Paillier Partially Homomorphic Encryption

- $[[u]]$ is the encryption of u
- **Addition**:

$$[[u]] + [[v]] = [[u + v]]$$

- **Scalar multiplication**:

$$n \cdot [[u]] = [[nu]]$$

- Extend to vectors ⇨ **encrypted linear algebra** (almost)!
- Our Paillier implementations:
  - Python github.com/n1analytics/python-paillier
  - Java github.com/n1analytics/javallier

Logistic regression

- **Goal:** Distributed SGD for logistic regression keeping data private
- **Challenges:**
  - Constrained by **Paillier** to simple arithmetics (e.g.: no log, no exp)
  - Data is split **by features** and cannot leave their data providers

Logistic regression

- **Goal:** Distributed SGD for logistic regression keeping data private
- **Challenges:**
  - Constrained by **Paillier** to simple arithmetics (e.g.: no log, no exp)
  - Data is split **by features** and cannot leave their data providers

- **Solutions:**
  - Gradient and loss approximation using **Taylor expansion**, up to 2nd order
  - Collaborative protocol for computing gradients and loss values

Taylor approximation*

- Logistic loss,

(Only used for stopping criterion)

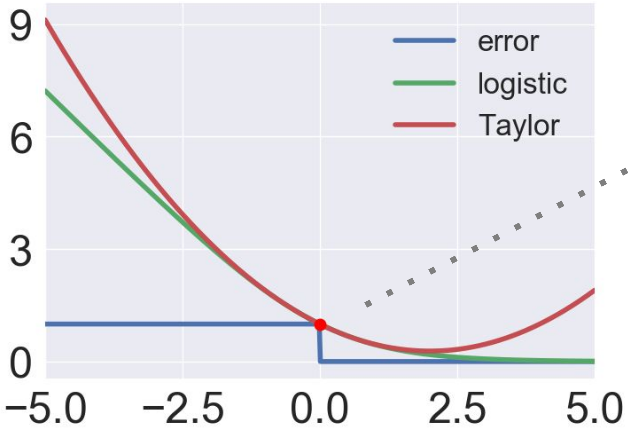
$$\ell(\theta) = \log(1 + \exp(-y\theta^\top x))$$
$$\approx \log 2 - \frac{1}{2} y\theta^\top x + \frac{1}{8}(\theta^\top x)^2$$

- and its gradient

$$\nabla \ell(\theta) = \left(\frac{1}{1 + e^{-y\theta^\top x}} - 1 \right) yx$$
$$\approx \left(\frac{1}{2} y\theta^\top x - 1 \right) \frac{1}{2} yx$$

* similar to [Aono et al. 16]

Logistic loss vs. its Taylor approximation



For a good approx: scale features into a small interval and regularize !

dataset	#rows	#features	accuracy sklearn	accuracy N1 Taylor
iris	100	3	100	100
digits (odd vs. even)	1500	64	94.3	94.3
mnist (odd vs. even)	60K	784	89.5	87.8
give me some credit	168K	10	87.0	87.1
covtype	500K	54	71.1	68.9

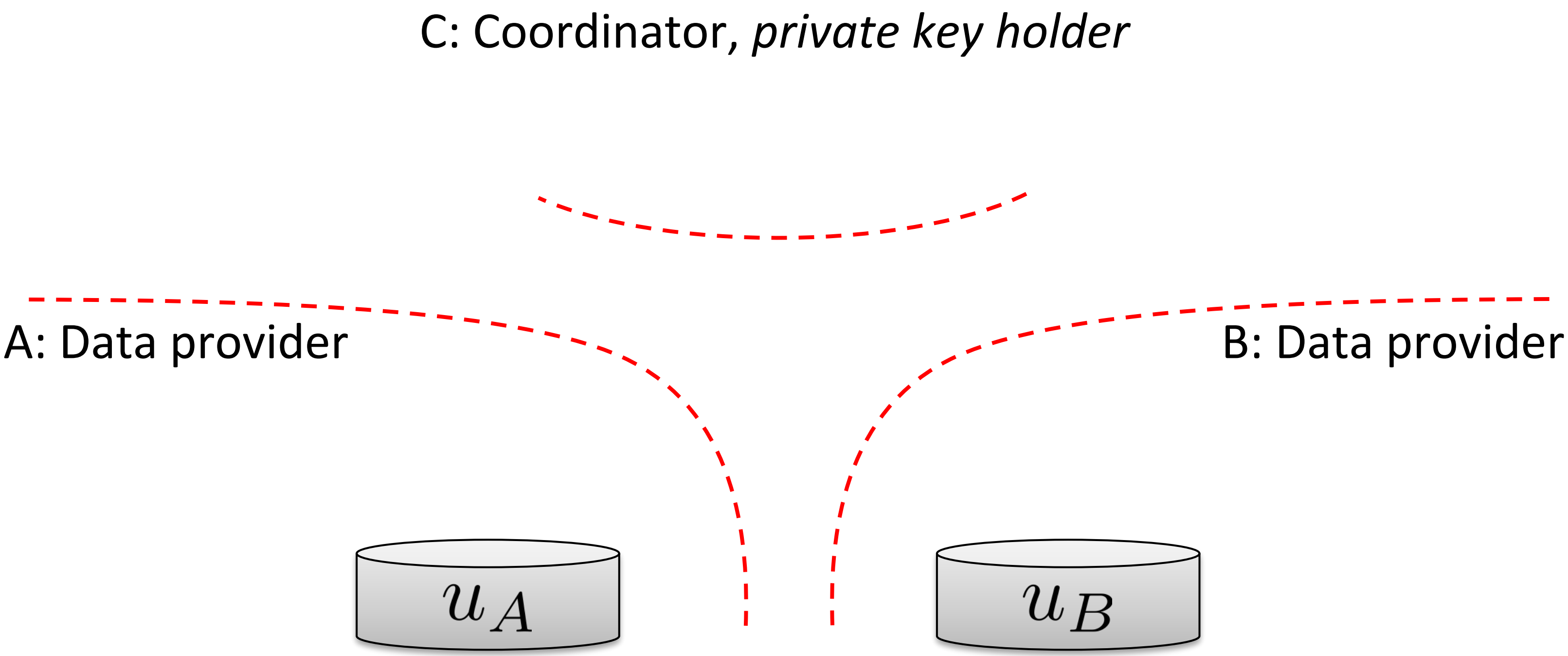
Protocol example: how to compute a square?

- The most complex operation in the learning protocol
- … and we cannot do squares on encrypted numbers with Paillier !

$$u = u_A + u_B$$

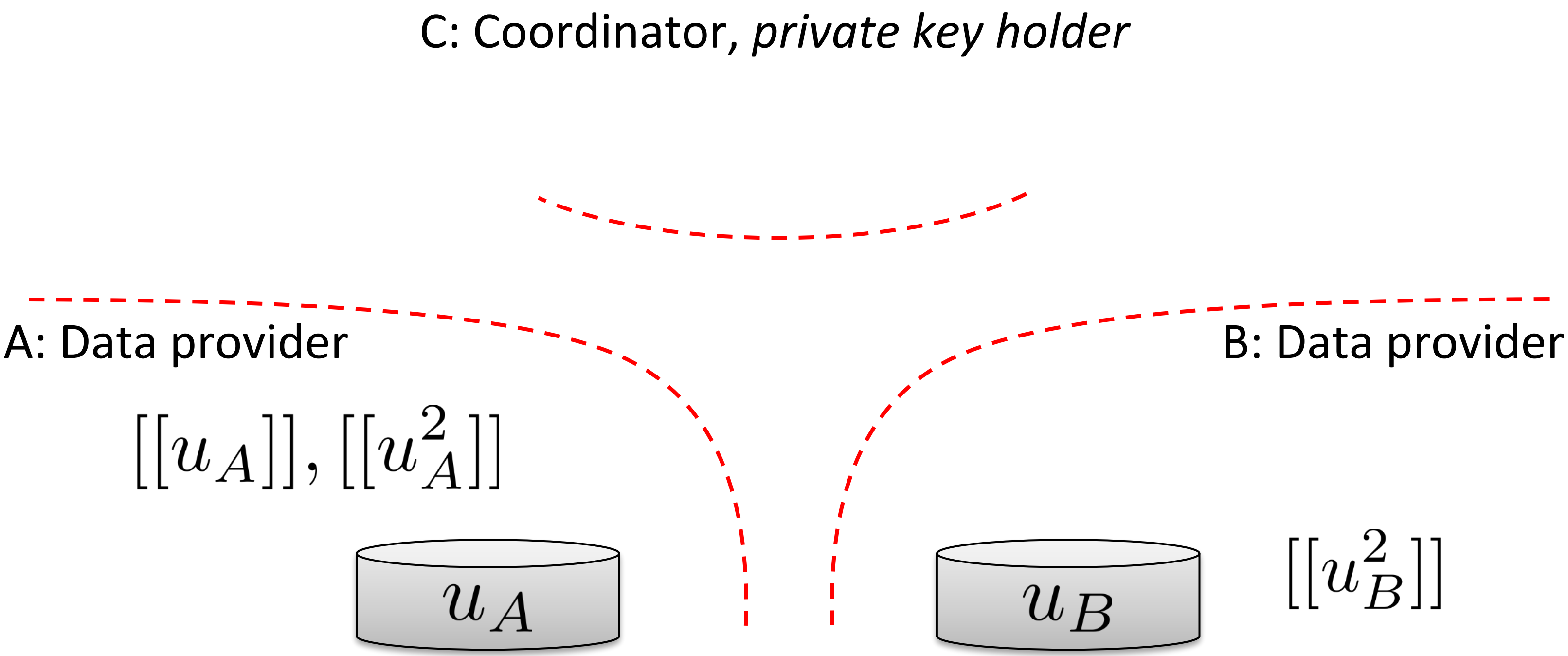
$$u^2 = u_A^2 + u_B^2 + 2u_A u_B$$

Protocol example: how to compute a square?

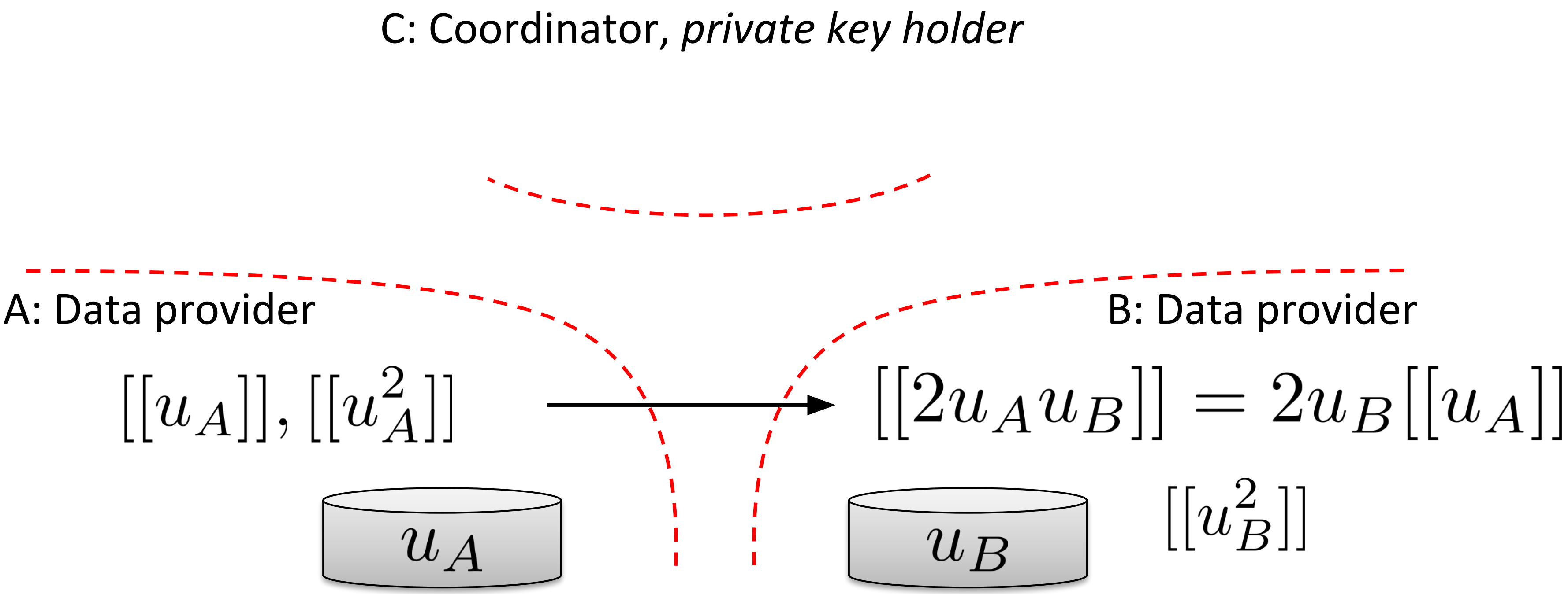


(Entities are matched via permutation and mask here)

Protocol example: how to compute a square?



Protocol example: how to compute a square?

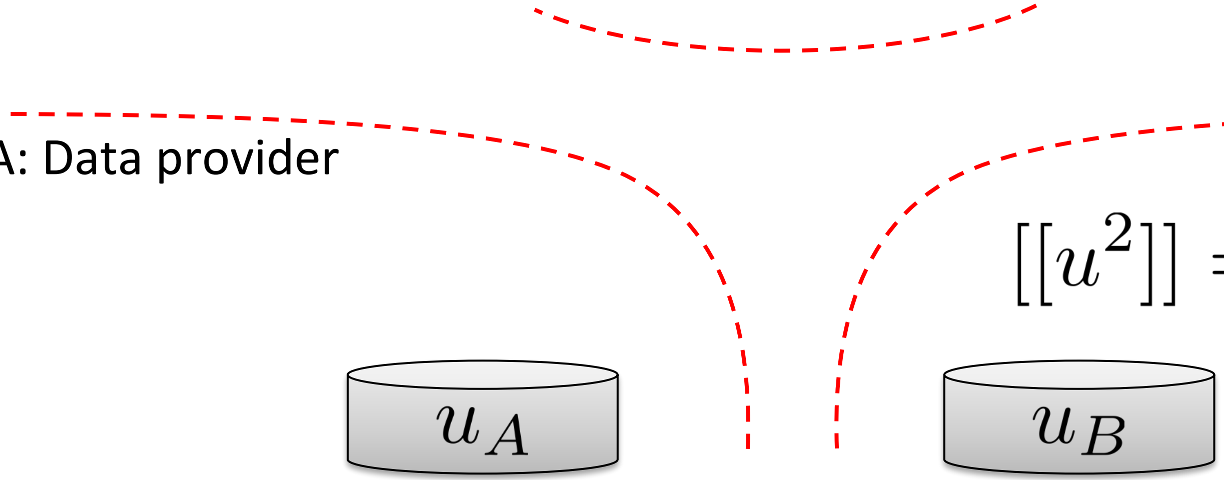


Protocol example: how to compute a square?

C: Coordinator, *private key holder*

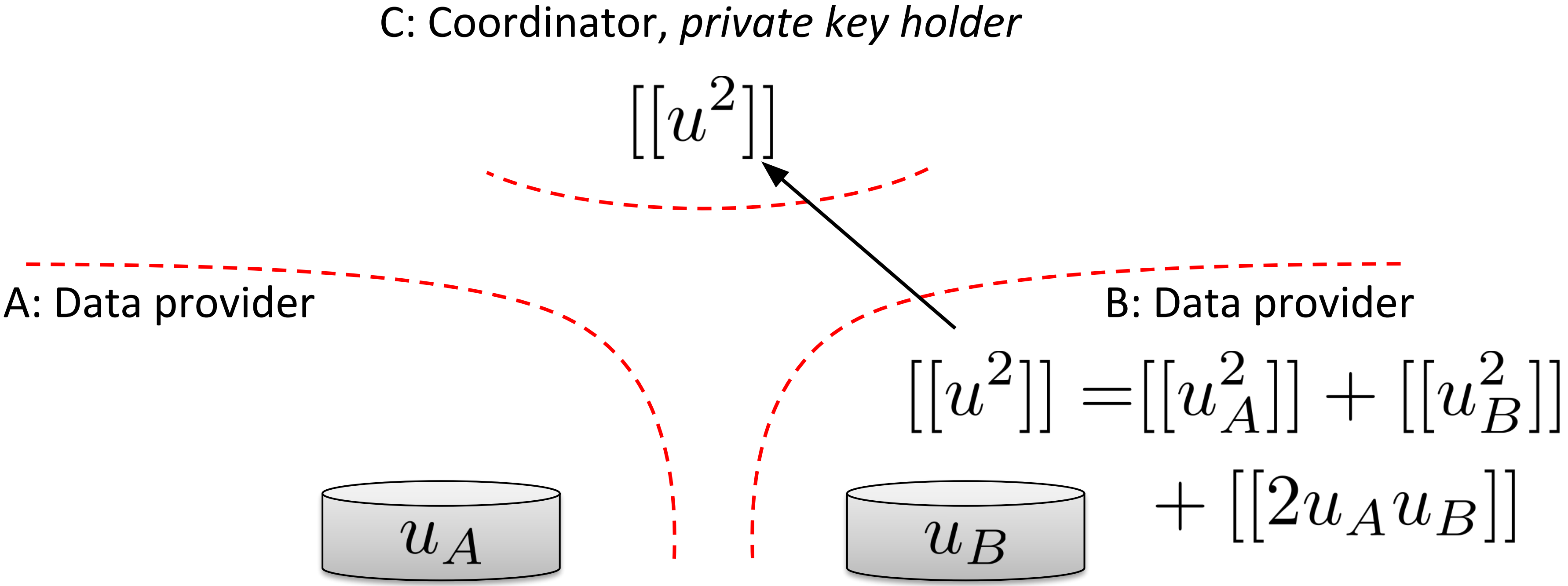
A: Data provider

B: Data provider

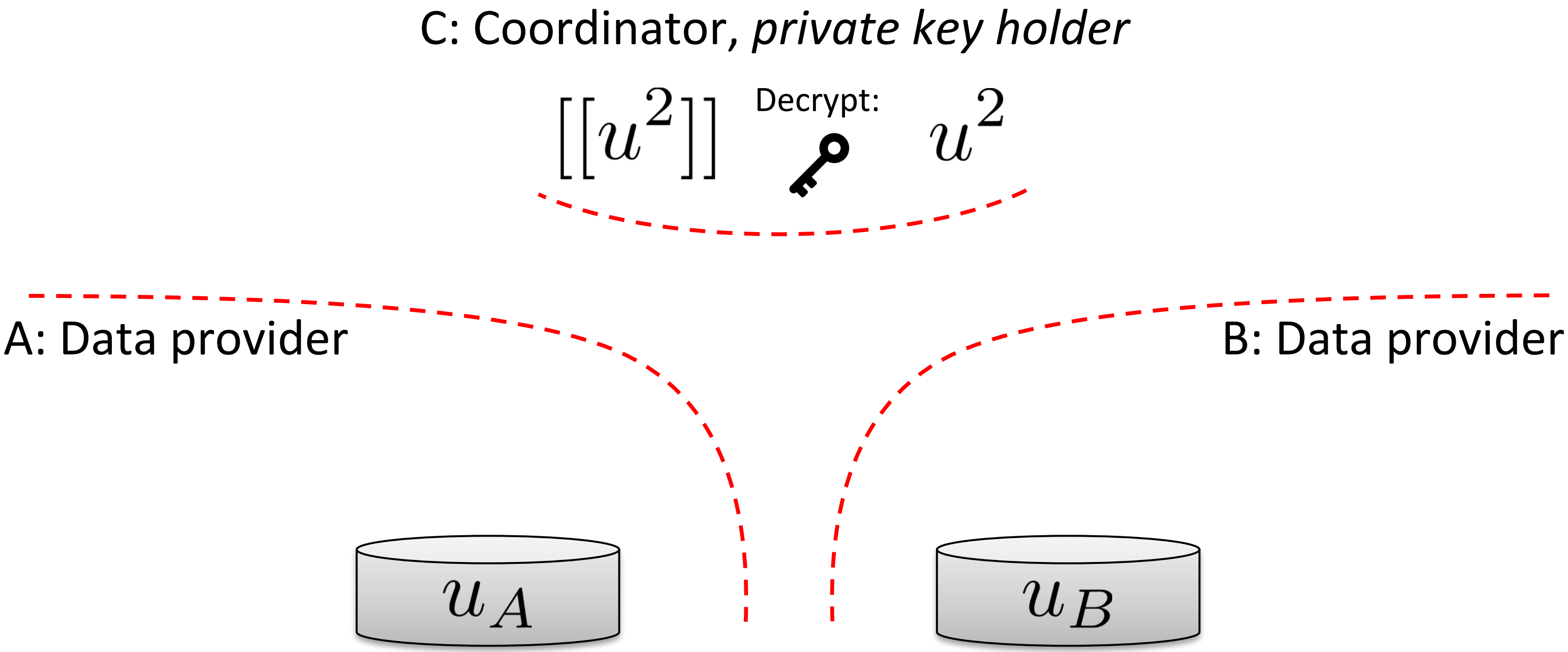


$$[[u^2]] = [[u_A^2]] + [[u_B^2]] + [[2u_A u_B]]$$

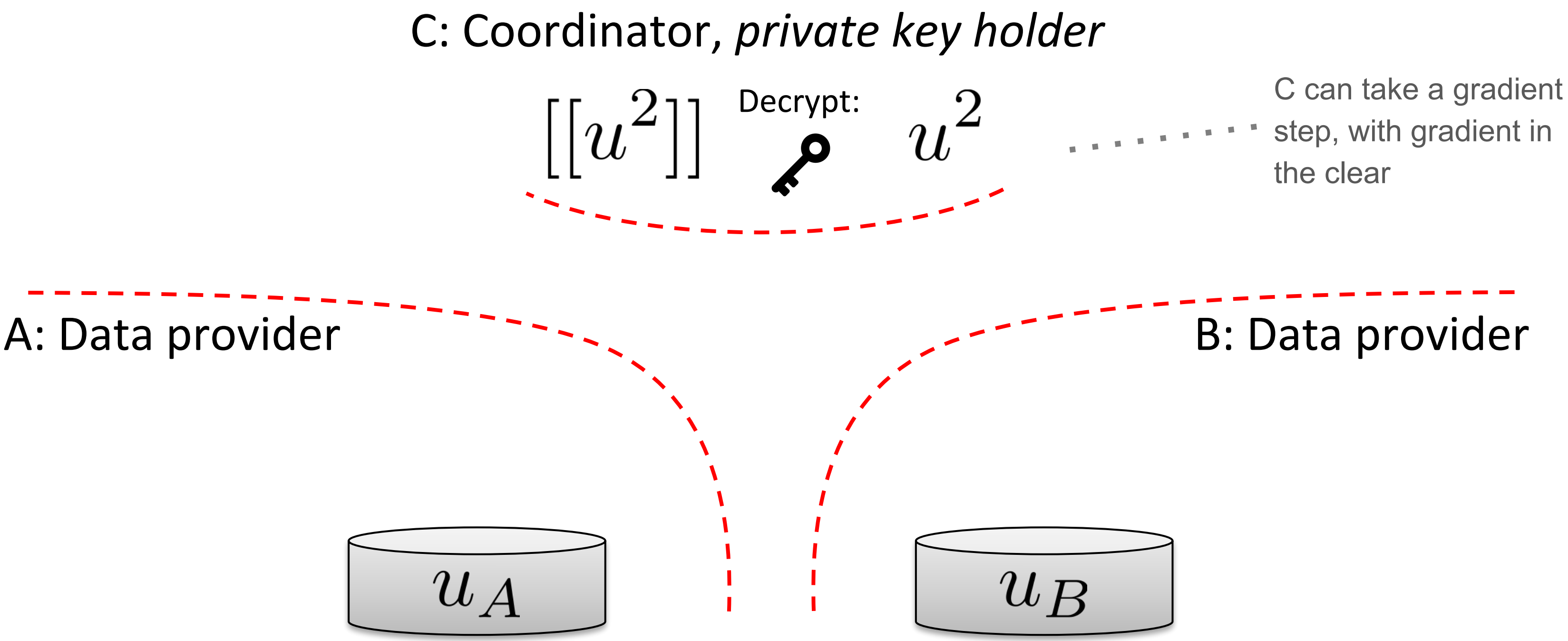
Protocol example: how to compute a square?



Protocol example: how to compute a square?



Protocol example: how to compute a square?



Overview

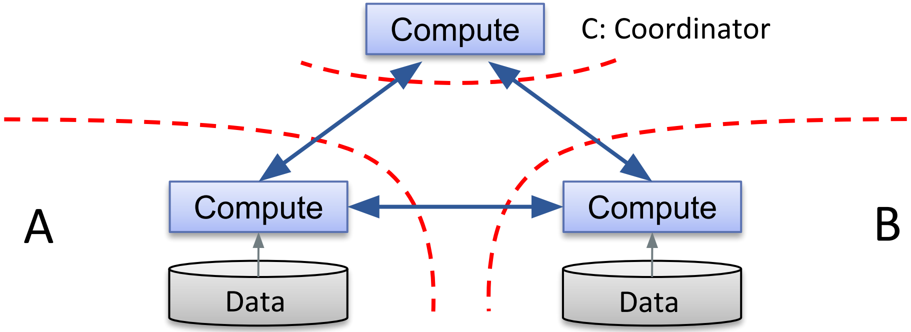
- End-to-end system:
  - Security assumptions / requirements
  - Entity resolution
  - Learning on private data
- **Deployment & experiments**

Deployment

Deployment at each party -- *2 data providers & coordinator* -- with docker images and kubernetes cluster.

AWS instance, R4.4xlarge:

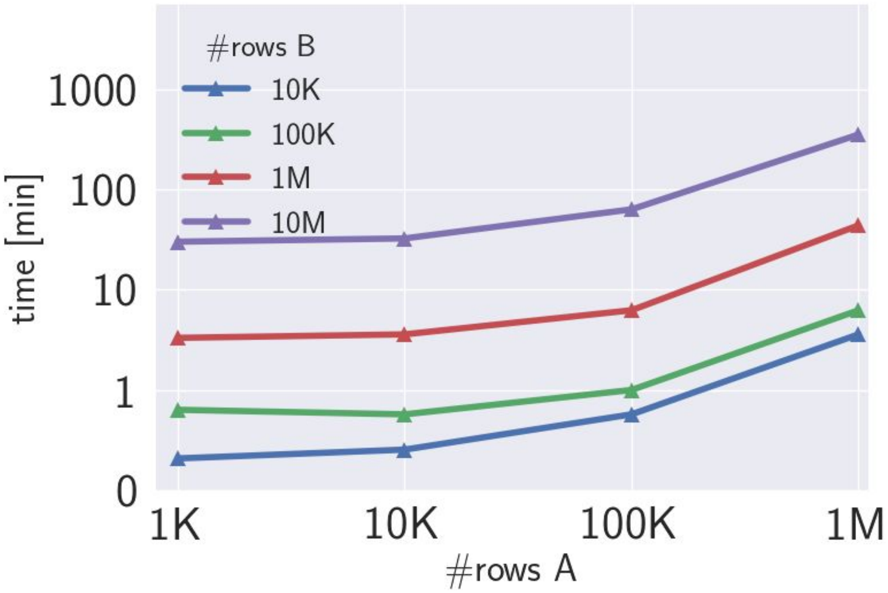
- 16 vCPU
- 60 GBs of RAM (DDR4)
- Up to 10 Gigabit network



30

Scalability of entity resolution

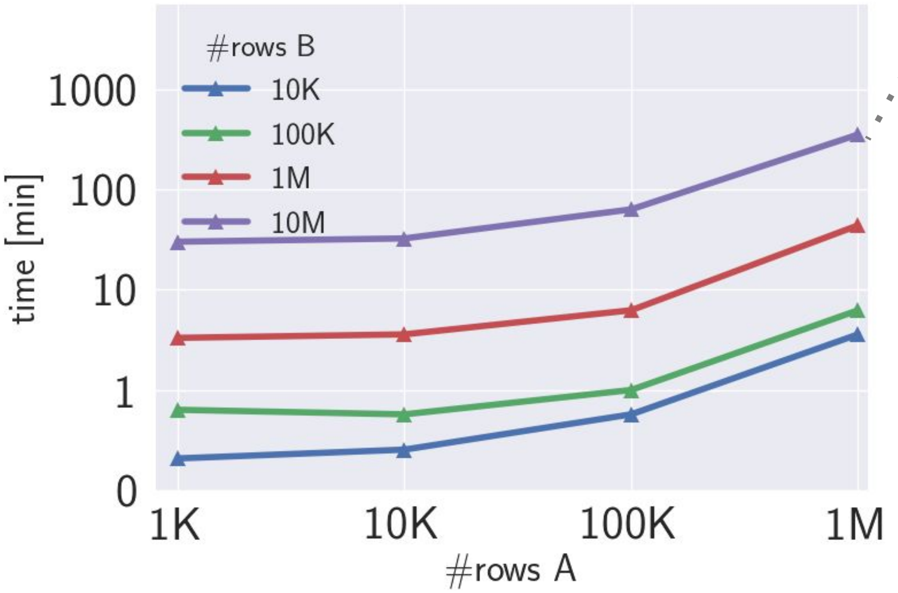
time =
hashing +
matching +
permutation



~6h

Scalability of entity resolution

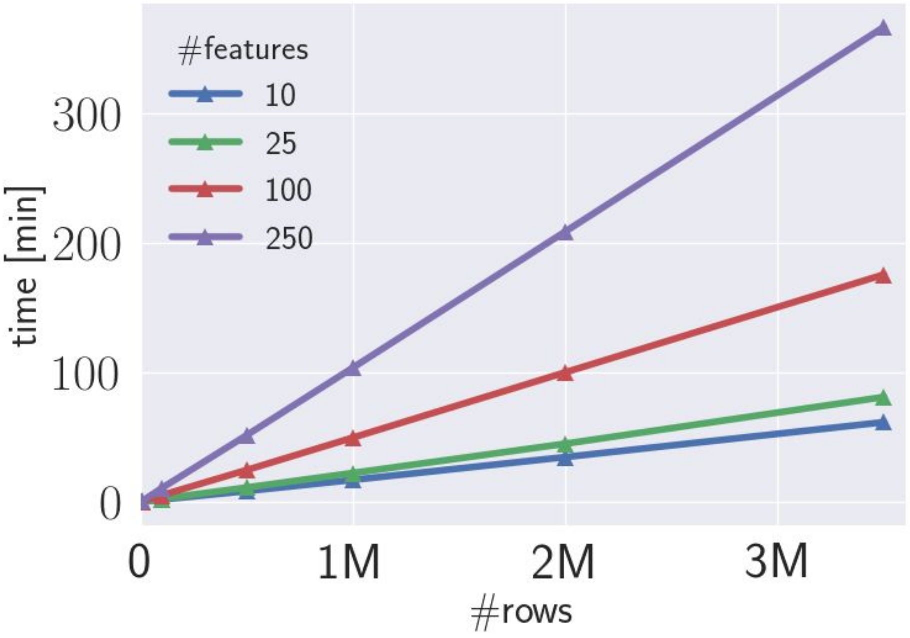
time =
hashing +
matching +
permutation

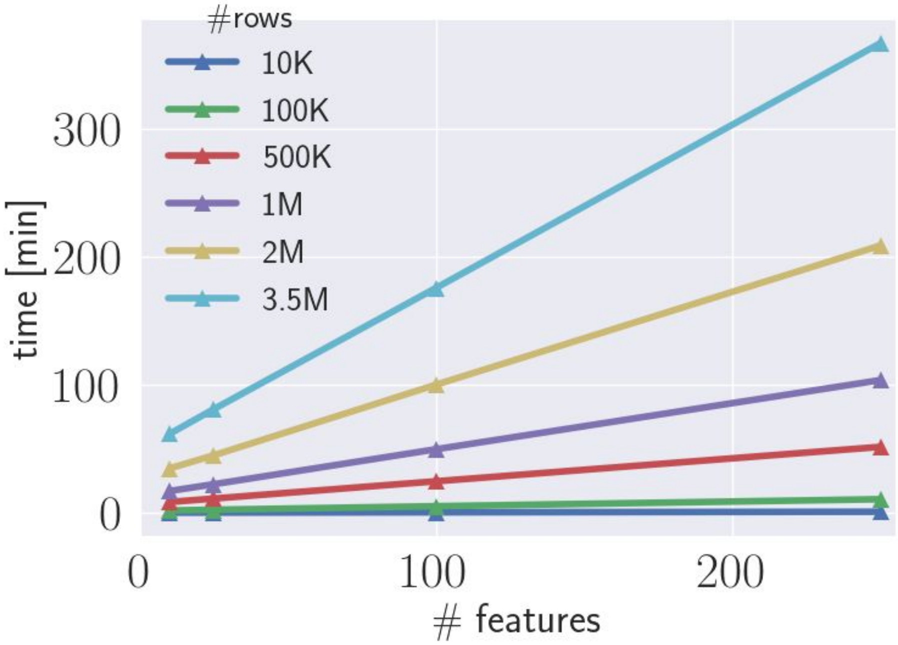


20 machines per node:
50min instead of 6h

Scalability of learning

time = 1 learning epoch + evaluation

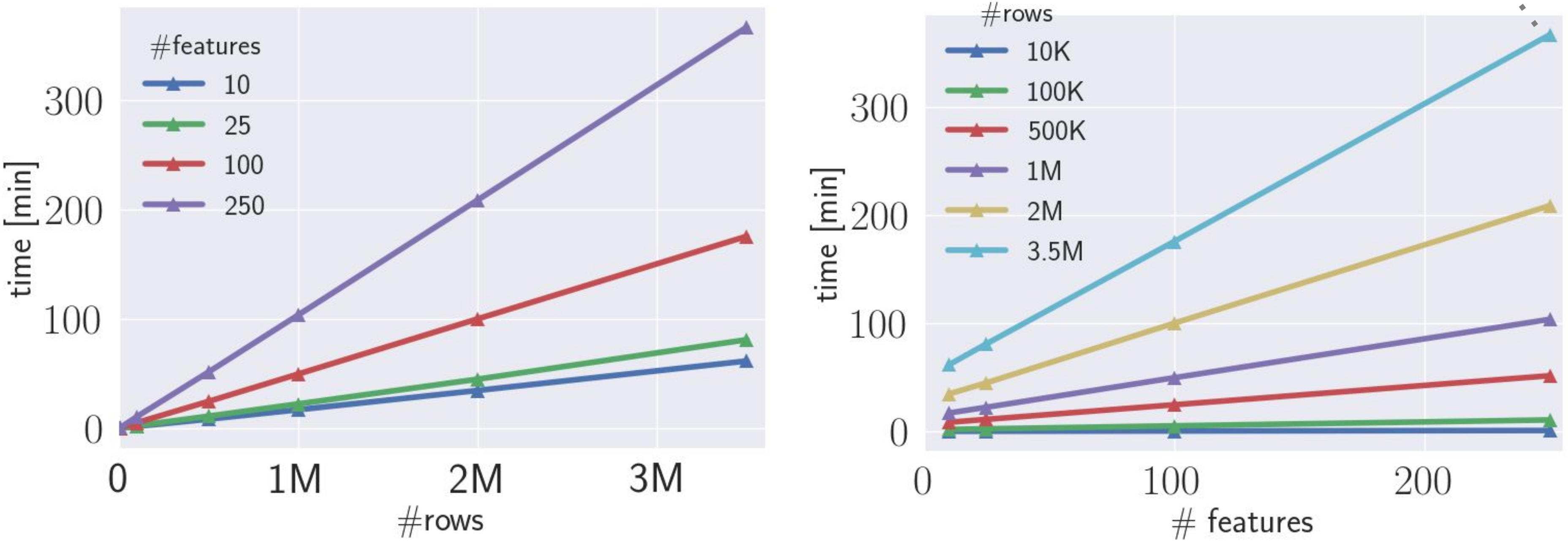




Scalability of learning

time = 1 learning epoch + evaluation

16 machines per node:
down to 200 min



Summary and future work

- End-to-end solution for **entity resolution + logistic regression** on **vertically partitioned** data
- Security:
  - Records remain confidential from other parties
  - Knowledge of common entities is not shared
- Scalability:
  - Commercial deployment on up to x1M rows and x100 features
- Work in progress:
  - Further parallelization: **cluster + GPUs**
  - 3+ data providers
  - Learning bypassing entity resolution [Nock et al. 15, Patrini et al. 16]

Thank you!

For more info:

- Website: www.n1analytics.com
- Blog: blog.n1analytics.com
- Twitter: @n1analytics

We are hiring!

- Research Scientist - Machine Learning (Sydney): jobs.csiro.au/s/LDOXTy

References

- P. Paillier, **Public-key cryptosystems based on composite degree residuosity classes**, EuroCrypt99
- R. Schnell, T. Bachteler, J. Reiher, **A novel error-tolerant anonymous linking code**, Tech report 2011
- R. Nock, G. Patrini, A. Friedman, **Rademacher observations, private data and boosting**, ICML15
- Y. Aono, T. Hayashi, T. P. Le, L. Wang, **Scalable and secure logistic regression via homomorphic encryption**, CODASPY16
- G. Patrini, R. Nock, S. Hardy, T. Caetano, **Fast learning from distributed data without entity matching**, IJCAI16