Privacy-preserving entity resolution and logistic regression on encrypted data

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

A: Data provider
B: Data provider
C: Coordinator

Sensitive messages are encrypted
Confidentiality boundary
Different features, many shared entities
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

<table>
<thead>
<tr>
<th>Name</th>
<th>DOB</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Klara Jovel</td>
<td>07/09/1942</td>
<td>...</td>
</tr>
<tr>
<td>Scott Redo</td>
<td>04/08/1923</td>
<td>...</td>
</tr>
<tr>
<td>Tori Mckone</td>
<td>07/06/1921</td>
<td>...</td>
</tr>
<tr>
<td>Rusty Brod</td>
<td>25/07/2014</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>DOB</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tori Mckone</td>
<td>07/06/1921</td>
<td>...</td>
</tr>
<tr>
<td>Scotty Undo</td>
<td>24/01/1965</td>
<td>...</td>
</tr>
<tr>
<td>Scott Redo</td>
<td>04/08/1923</td>
<td>...</td>
</tr>
<tr>
<td>Clara Jovel</td>
<td>07/09/1942</td>
<td>...</td>
</tr>
</tbody>
</table>

- **Constraint**: can’t share Personally Identifiable Information (PII)
- **Solution**: *fuzzy & private* matching
Privacy-preserving entity resolution

C: Coordinator

A: Data provider

B: Data provider

**Name, DOB, gender, etc. of A’s customers**
Privacy-preserving entity resolution

C: Coordinator

A: Data provider

B: Data provider

Preserves similarity, \textit{e.g.} by hash on bigrams [Schnell et al. 11]

Shared secret salt

Hash

PII

Hash

PII
Privacy-preserving entity resolution

Fuzzy matcher

C: Coordinator

A: Data provider

Hash

PII

B: Data provider

Hash

PII

Robust to misspellings and errors
Privacy-preserving entity resolution: the output

Permutation & encrypted mask

C: Coordinator

permutations: align row of A and B
encrypted mask: vector of encrypted 0/1 to select matches

A: Data provider

B: Data provider

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** $\Rightarrow$ **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 $\Rightarrow$ **encrypted linear algebra** (almost)!
- Our Paillier implementations:
  - Python  [github.com/n1analytics/python-paillier](https://github.com/n1analytics/python-paillier)
  - Java    [github.com/n1analytics/javallier](https://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,
  \[
  \ell(\theta) = \log(1 + \exp(-y\theta^\top x)) \\
  \approx \log 2 - \frac{1}{2} y\theta^\top x + \frac{1}{8} (\theta^\top x)^2
  \]

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

C: Coordinator, *private key holder*

A: Data provider
B: Data provider

(Entities are matched via permutation and mask here)
Protocol example: how to compute a square?

C: Coordinator, *private key holder*

A: Data provider

\[[[u_A]], [[u_A^2]]\]

B: Data provider

\[[[u_B^2]]\]
Protocol example: how to compute a square?

C: Coordinator, *private key holder*

\[
[[u_A]], [[u_A^2]] 
\]

\[
[[2u_A u_B]] = 2u_B [[u_A]]
\]

\[
[[u_B^2]]
\]
Protocol example: how to compute a square?

C: Coordinator, private key holder

A: Data provider

B: Data provider

\[
[[u^2]] = [[u_A^2]] + [[u_B^2]] + [[2u_Au_B]]
\]
Protocol example: how to compute a square?

C: Coordinator, *private key holder*

\[
[[u^2]] = [[u_A^2]] + [[u_B^2]] + [[2u_Au_B]]
\]
Protocol example: how to compute a square?

C: Coordinator, *private key holder*

\[
\left[ \left[ u^2 \right] \right] \quad \text{Decrypt:} \quad u^2
\]

A: Data provider

B: Data provider

\[ u_A \quad \text{and} \quad u_B \]
Protocol example: how to compute a square?

C: Coordinator, private key holder

\[
\begin{bmatrix} u^2 \end{bmatrix}
\]

Decrypt:

\[ u^2 \]

C can take a gradient step, with gradient in the clear

A: Data provider

B: Data provider

\[ u_A \]

\[ u_B \]
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
Scalability of entity resolution

time = hashing + matching + permutation
Scalability of entity resolution

\[
\text{time} = \text{hashing} + \text{matching} + \text{permutation}
\]

20 machines per node: 50min instead of 6h
Scalability of learning

time = 1 learning epoch + evaluation
**Scalability of learning**

\[ \text{time} = 1 \text{ learning epoch} + \text{evaluation} \]

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. 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