EFFICIENT DIFFERENTIALLY PRIVATE AVERAGING WITH TRUSTED CURATOR UTILITY AND ROBUSTNESS TO MALICIOUS PARTIES

Aurélien Bellet (Inria, France)

Joint work with César Sabater and Jan Ramon (Inria)

Google Workshop on Federated Learning and Analytics
July 29-30, 2020
We tackle two challenges in Federated Learning (FL):

1. Provide differential privacy (DP) guarantees to the participants
2. Ensure correctness of the computation in the presence of malicious parties
• A set $U = \{1, \ldots, n\}$ of users (parties)

• Each user $u \in U$ holds a private value $X_u \in [0, 1]$

• **Goal:** accurately estimate $X_{avg} = \frac{1}{n} \sum_u X_u$ without revealing individual values

• **Motivation:** many federated optimization algorithms can be written as follows:

```plaintext
for $t = 1$ to $T$ do
  At each user $u$: compute $\theta_u^t \leftarrow \text{LOCALUPDATE}(\theta^{t-1}, \theta_u^{t-1})$, send $\theta_u^t$ to server
  At server: compute $\theta^t \leftarrow \frac{1}{n} \theta_u^t$, send $\theta^t$ back to users
end for
```
• **Local DP** [Kasiviswanathan et al., 2008, Duchi et al., 2013]: poor utility, communication-efficient, some robustness

• **DP+secure aggregation** [Dwork et al., 2006, Shi et al., 2011, Bonawitz et al., 2017]: trusted curator utility, $O(n)$ messages per user, possible to enforce correctness

Recent concurrent work on breaking the $O(n)$ barrier: [Bell et al., 2020, So et al., 2020]

• **DP+secure shuffling** [Cheu et al., 2019, Erlingsson et al., 2019, Balle et al., 2019]: trusted curator utility, practical implementations?, robustness?
OUR KEY CONTRIBUTIONS

1. A novel efficient protocol based on exchanging (correlated) Gaussian noise along the edges of a network graph
2. Trusted curator utility with only logarithmic number of messages per party
3. Guaranteed correctness via homomorphic commitments and zero knowledge proofs
Algorithm 1 GOPA protocol

Parameters: graph $G$, variances $\sigma^2_\Delta, \sigma^2_\eta \in \mathbb{R}^+$

for all neighboring users $\{u, v\}$ in $G$ do
  $u$ and $v$ draw $x \sim \mathcal{N}(0, \sigma^2_\Delta)$
  set $\Delta_{u,v} \leftarrow x$, $\Delta_{v,u} \leftarrow -x$
end for

for each user $u$ do
  $u$ draws $\eta_u \sim \mathcal{N}(0, \sigma^2_\eta)$
  $u$ reveals $\hat{x}_u \leftarrow X_u + \sum_{v \sim u} \Delta_{u,v} + \eta_u$
end for

- Unbiased estimate of the average: $\hat{X}_{avg} = \frac{1}{n} \sum_u \hat{x}_u$, with variance $\sigma^2_\eta/n$

1. All neighbors $\{u, v\}$ in $G$ generate pairwise-canceling Gaussian noise
2. Each user $u$ generate independent Gaussian noise
3. User $u$ reveals the sum of private value, pairwise and independent noise terms
• **Adversary**: proportion 1 – $\rho$ of colluding malicious users who observe all communications they take part in

• Denote by $U^H$ the set of honest-but-curious parties, and by $G^H$ the honest subgraph

• GOPA can achieve $(\epsilon, \delta)$-DP for any $\epsilon, \delta > 0$ for connected $G^H$ and large enough $\sigma^2_\eta, \sigma^2_\Delta$

• We show that $\sigma^2_\eta$ can be as small as in the trusted curator setting (matching its utility)

• We show that the required $\sigma^2_\Delta$ depends on the topology of $G^H$ through the properties of an embedded spanning tree
Theorem (Case of random $k$-out graph)

Let $\epsilon, \delta' \in (0, 1)$ and:

- $G$ be obtained by letting all users randomly choose $k = O(\log(\rho n)/\rho)$ neighbors
- $\sigma_n^2 = O(\log(1/\delta')/|U^H|\epsilon^2)$ as per the Gaussian mechanism in trusted curator setting
- $\sigma_\Delta^2 = O(\sigma_n^2|U^H|/k)$

Then GOPA is $(\epsilon, \delta)$-differentially private for $\delta = O(\delta')$.

- Trusted curator utility with logarithmic number of messages per user
- Our theoretical results give practical values for $k$ and $\sigma_\Delta^2$ (see paper)
- Note: we can obtain even smaller values by numerical simulation
ENSURING CORRECTNESS

• Utility can be compromised by malicious users tampering with the protocol (e.g., sending incorrect values to bias the outcome)

• It is impossible to force a user to give the “right” input (this also holds in the trusted curator setting)

• We enable each user $u$ to prove the following properties:

  \[ X_u \in [0, 1], \quad \forall u \in U \]

  \[ \Delta_{u,v} = -\Delta_{v,u}, \quad \forall \{u, v\} \text{ neighbors in } G \]

  \[ \eta_u \sim \mathcal{N}(0, \sigma^2), \quad \forall u \in U \]

  \[ \hat{X}_u = X_u + \sum_{v \sim u} \Delta_{u,v} + \eta_u, \quad \forall u \in U \]
ENSURING CORRECTNESS

• Users publish an encrypted log of the computation using Pedersen commitments [Blum, 1983, Franck and Großschädl, 2017], which are additively homomorphic
• Based on these commitments, users prove that the computation was done correctly using zero knowledge proofs
• Note: lots of technical subtleties (e.g., work in fixed precision)

Theorem (Informal)

Under the Discrete Logarithm Assumption (DLA), a user $u \in U$ that passes the verification procedure proves that $\hat{X}_u$ was computed correctly. Additionally, by doing so, $u$ does not reveal any additional information about $X_u$, even if DLA does not hold.

• Costs per user remain linear in the number of neighbors
• Can prove consistency across multiple runs on same/similar data
• Can handle drop out (to some extent)
Thank you for your attention!

See full paper on arXiv:
The Privacy Blanket of the Shuffle Model.
In CRYPTO.

Secure Single-Server Aggregation with (Poly)Logarithmic Overhead.
Technical report, IACR Cryptol. ePrint Arch. 704.

Coin flipping by telephone a protocol for solving impossible problems.

Practical Secure Aggregation for Privacy-Preserving Machine Learning.
In CCS.

Distributed Differential Privacy via Shuffling.
In EUROCRYPT.
REFERENCES

Local privacy and statistical minimax rates.
In FOCS.

Our Data, Ourselves: Privacy Via Distributed Noise Generation.
In EUROCRYPT.

Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity.
In SODA.

Efficient Implementation of Pedersen Commitments Using Twisted Edwards Curves.

What Can We Learn Privately?
In FOCS.
Privacy-Preserving Aggregation of Time-Series Data. 
In NDSS.

Turbo-Aggregate: Breaking the Quadratic Aggregation Barrier in Secure Federated Learning. 