CYBER IN Occitanie 2025

Decentralized Algorithms with Differential Privacy

César Sabater [1]   Sonia Ben Mokhtar [1,2]

[1]DRIM Team, INSA-Lyon

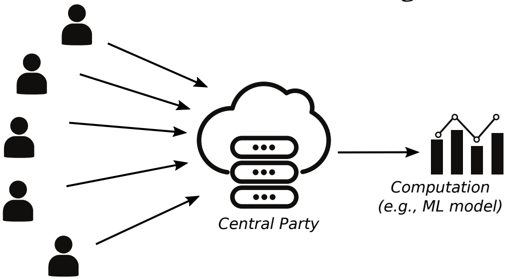
[2]CNRS

July 10, 2025

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

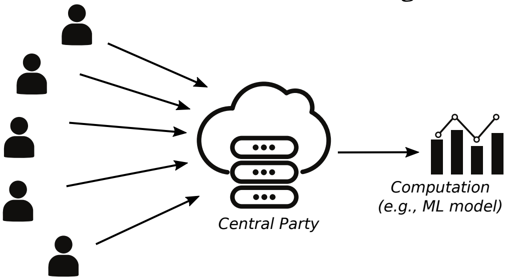
Introduction

Centralized Data Processing



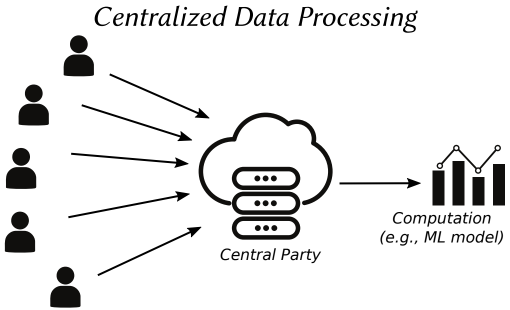
Introduction

Centralized Data Processing



- data concentration into **possibly untrusted organizations**

Introduction



- data concentration into **possibly untrusted organizations**
- data is often sensitive → **raises privacy concerns**

Decentralized Algorithms

Among many measures such as Government Regulations (e.g., GDPR) and Technical Solutions (Cryptography, Anonymization, Obfuscation, …)

Decentralized trend:

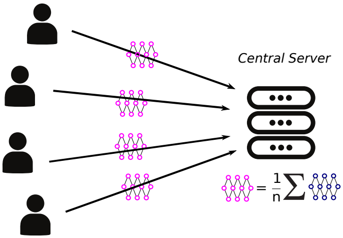
- keep data local, exchange computations

Decentralized Algorithms

Among many measures such as Government Regulations (e.g., GDPR) and Technical Solutions (Cryptography, Anonymization, Obfuscation, ...)

Decentralized trend: *Federated Learning*[1]

- keep data local, exchange computations



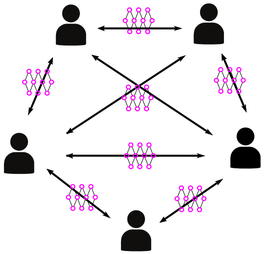
[1] Kairouz, Peter, et al. "Advances and open problems in federated learning." Foundations and trends® in machine learning (2021)

Decentralized Algorithms

Among many measures such as Government Regulations (e.g., GDPR) and Technical Solutions (Cryptography, Anonymization, Obfuscation, ...)

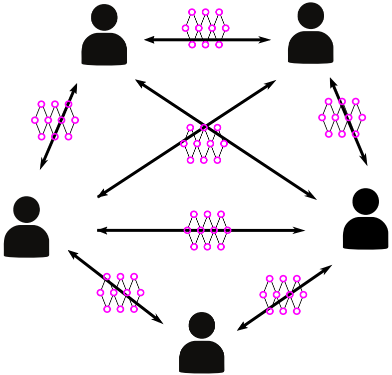
Decentralized trend: *Decentralized Computations (ML[1], MPC)*

- keep data local, exchange computations



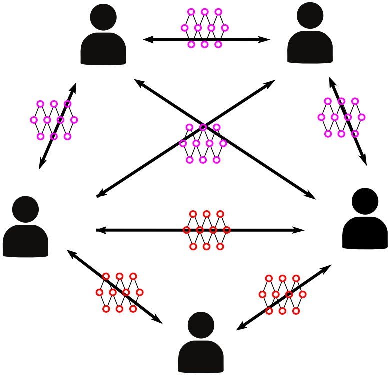
[1] Ormándi, Róbert, et al. "Gossip learning with linear models on fully distributed data." 2013.

Challenges of Decentralization



Challenges

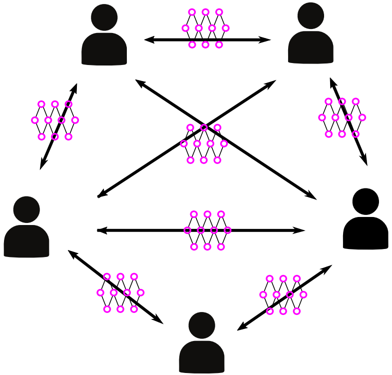
Challenges of Decentralization



Challenges

1. **Messages can compromise privacy**
   - Membership Inference Attacks
   - Data Reconstruction Attacks

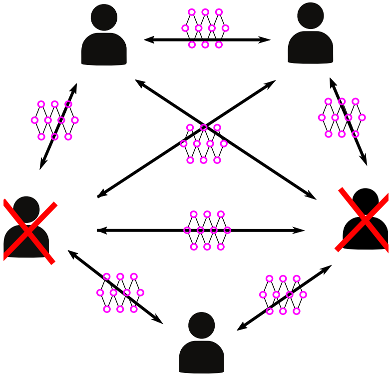
Challenges of Decentralization



Challenges

1. **Messages can compromise privacy**
   - Membership Inference Attacks
   - Data Reconstruction Attacks
2. Outcome depends on many participants

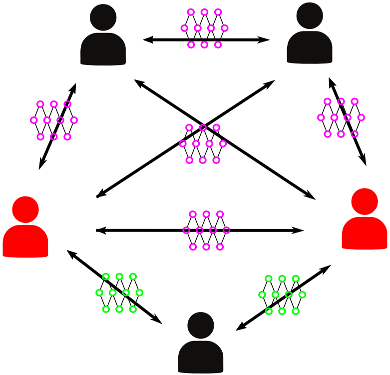
Challenges of Decentralization



Challenges

1. **Messages can compromise privacy**
   - Membership Inference Attacks
   - Data Reconstruction Attacks
2. Outcome depends on many participants
   - Unexpectedly **disconnect or crash**

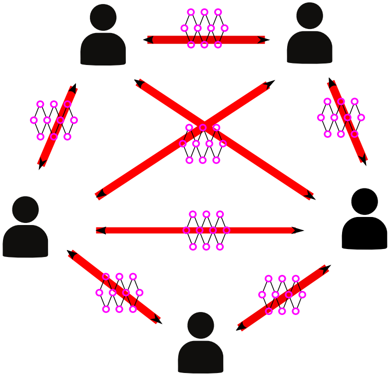
Challenges of Decentralization



Challenges

1. **Messages can compromise privacy**
   - Membership Inference Attacks
   - Data Reconstruction Attacks
2. Outcome depends on many participants
   - Unexpectedly **disconnect or crash**
   - Intentionally **deviate from the protocol**
   - **collude** and **gather private information**

Challenges of Decentralization



Challenges

1. **Messages can compromise privacy**
   - Membership Inference Attacks
   - Data Reconstruction Attacks
2. Outcome depends on many participants
   - Unexpectedly **disconnect or crash**
   - Intentionally **deviate from the protocol**
   - **collude** and **gather private information**
3. May require a large communication cost

Outline

Focus:

- **Distributed Mean Estimation** under **Differential Privacy** constraints

Contributions:

1. *An accurate, scalable and verifiable protocol for federated differentially private averaging.* Machine Learning, 2022.
   with **Aurélien Bellet** and **Jan Ramon**.
2. *Private sampling with identifiable cheaters.* PoPETS 2023
   with **Florian Hahn**, **Andreas Peter** and **Jan Ramon**
3. *Dropout-Robust Mechanisms for Differentially Private and Fully Decentralized Mean Estimation.*. ArXiv preprint, 2025.
   with **Sonia Ben Mokhtar** and **Jan Ramon**.

- Conclusion

Outline

Focus:

- **Distributed Mean Estimation** under **Differential Privacy** constraints

Contributions:

1. *An accurate, scalable and verifiable protocol for federated differentially private averaging.* Machine Learning, 2022.
   with **Aurélien Bellet** and **Jan Ramon**.
2. *Private sampling with identifiable cheaters.* PoPETS 2023
   with **Florian Hahn**, **Andreas Peter** and **Jan Ramon**
3. *Dropout-Robust Mechanisms for Differentially Private and Fully Decentralized Mean Estimation.*. ArXiv preprint, 2025.
   with **Sonia Ben Mokhtar** and **Jan Ramon**.

- Conclusion

Distributed Mean Estimation under DP

Problem: Private Mean Estimation

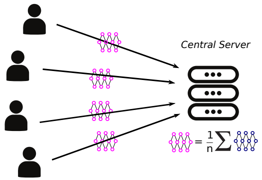
- Set $U = \{1, \ldots, n\}$ of parties
- Each party $u \in U$ has a private value X_u (scalars, gradients, models..)
- No party is trusted with the data of others
- **Goal:** Estimate $\frac{1}{n} \sum_u X_u$ **while satisfying differential privacy constraints**

Distributed Mean Estimation under DP

Problem: Private Mean Estimation

- Set $U = \{1, \ldots, n\}$ of parties
- Each party $u \in U$ has a private value X_u (scalars, gradients, models..)
- No party is trusted with the data of others
- **Goal:** Estimate $\frac{1}{n} \sum_u X_u$ **while satisfying differential privacy constraints**

Key Primitive in Private Federated Learning

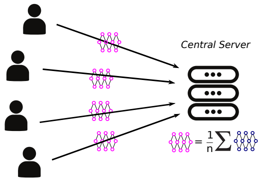


Distributed Mean Estimation under DP

Problem: Private Mean Estimation

- Set $U = \{1, \dots, n\}$ of parties
- Each party $u \in U$ has a private value X_u (scalars, gradients, models..)
- No party is trusted with the data of others
- **Goal:** Estimate $\frac{1}{n}\sum_u X_u$ **while satisfying differential privacy constraints**

Key Primitive in Private Federated Learning



- Can be used to **Federated SGD**, **matrix factorization**, **empirical CDFs**, **decision trees**, **private clustering**, **linear regression**, ...

Differential Privacy (DP)

A stochastic algorithm $\mathcal{A}$ *is* (ε, δ)*-Differentially Private if*

- *for all possible outcomes* $\mathcal{O}$
- *any pair of neighboring datasets* D, D'

$$\Pr[\mathcal{A}(D) = \mathcal{O}] \leq \exp(\varepsilon) \Pr[\mathcal{A}(D') = \mathcal{O}] + \delta$$

where two datasets are neighboring if they only differ the data of one party

Differential Privacy (DP)

A stochastic algorithm $\mathcal{A}$ is (ε, δ)-Differentially Private if

- *for all possible outcomes $\mathcal{O}$*
- *any pair of neighboring datasets D, D'*

$$\Pr[\mathcal{A}(D) = \mathcal{O}] \leq \exp(\varepsilon) \Pr[\mathcal{A}(D') = \mathcal{O}] + \delta$$

where two datasets are neighboring if they only differ the data of one party

- Related to resistance against MIA

Differential Privacy (DP)

A stochastic algorithm $\mathcal{A}$ is (ε, δ)-Differentially Private if

- *for all possible outcomes $\mathcal{O}$*
- *any pair of neighboring datasets D, D'*

$$\Pr[\mathcal{A}(D) = \mathcal{O}] \leq \exp(\varepsilon) \Pr[\mathcal{A}(D') = \mathcal{O}] + \delta$$

where two datasets are neighboring if they only differ the data of one party

- Related to resistance against MIA
- DP guarantees can be obtained by randomizing computations
  - E.g. using Gaussian, Binomial, Laplacian or Exponential noise
- More noise $\longrightarrow$ smaller ϵ and/or δ

Differential Privacy (DP)

A stochastic algorithm $\mathcal{A}$ is (ε, δ)-Differentially Private if

- *for all possible outcomes $\mathcal{O}$*
- *any pair of neighboring datasets D, D'*

$$\Pr[\mathcal{A}(D) = \mathcal{O}] \leq \exp(\varepsilon) \Pr[\mathcal{A}(D') = \mathcal{O}] + \delta$$

where two datasets are neighboring if they only differ the data of one party

- Related to resistance against MIA
- DP guarantees can be obtained by randomizing computations
  - E.g. using Gaussian, Binomial, Laplacian or Exponential noise
- More noise $\rightarrow$ smaller ϵ and/or δ
- Protect from **any adversary** for a given view $\mathcal{O}$

Differential Privacy (DP)

A stochastic algorithm $\mathcal{A}$ is (ε, δ)-Differentially Private if

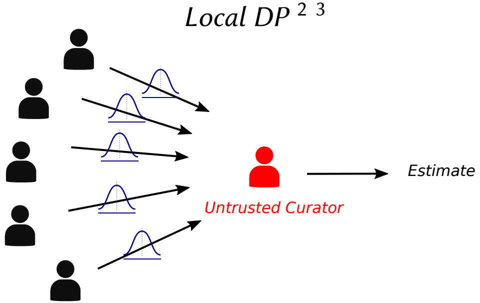
- *for all possible outcomes $\mathcal{O}$*
- *any pair of neighboring datasets D, D'*

$$\Pr[\mathcal{A}(D) = \mathcal{O}] \leq \exp(\varepsilon) \Pr[\mathcal{A}(D') = \mathcal{O}] + \delta$$

where two datasets are neighboring if they only differ the data of one party

- Related to resistance against MIA
- DP guarantees can be obtained by randomizing computations
  - E.g. using Gaussian, Binomial, Laplacian or Exponential noise
- More noise $\longrightarrow$ smaller ϵ and/or δ
- Protect from **any adversary** for a given view $\mathcal{O}$
- Sometimes **difficult to prove** and/or compromise accuracy

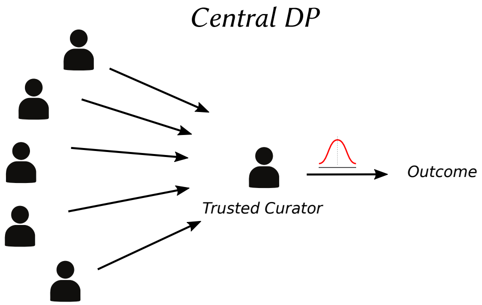
Private Averaging: Previous Approaches



- huge amount of noise
- in most cases, it produces inaccurate models

[2] [Duchi et al. FOCS 2013]
[3] [Kasiviswanathan, et al. SIAM Journal on Computing, 2011]

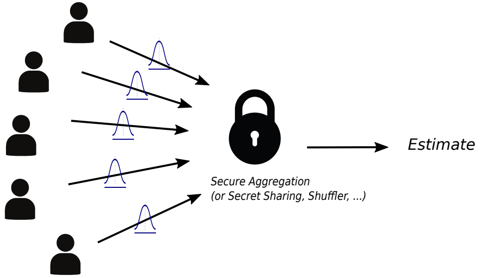
Private Averaging: Previous Approaches



- $O(n)$ factor of reduction compared to local DP variance
- a trusted party is required

Private Averaging: Previous Approaches

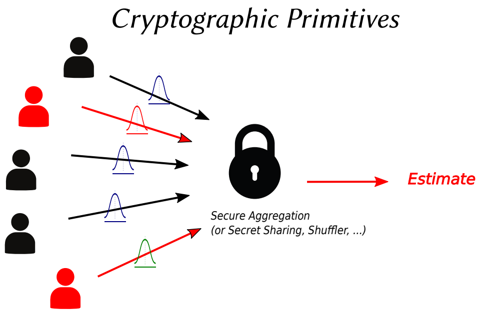
Cryptographic Primitives



- poor scalability, $O(n)$ messages per party [2]

[2] [Bonawitz et al., CSS 2017.]

Private Averaging: Previous Approaches



- poor scalability, $O(n)$ messages per party [2]
- vulnerable to malicious participants

[2] [Bonawitz et al., CSS 2017.]

Outline

Focus:

- **Distributed Mean Estimation** under **Differential Privacy** constraints

Contributions:

1. *An accurate, scalable and verifiable protocol for federated differentially private averaging.* Machine Learning, 2022.
   with **Aurélien Bellet** and **Jan Ramon**.
2. *Private sampling with identifiable cheaters.* PoPETS 2023
   with **Florian Hahn**, **Andreas Peter** and **Jan Ramon**
3. *Dropout-Robust Mechanisms for Differentially Private and Fully Decentralized Mean Estimation..* ArXiv preprint, 2025.
   with **Sonia Ben Mokhtar** and **Jan Ramon**.

- Conclusion

Distributed Mean Estimation under DP

Problem: Private Mean Estimation

- Set $U = \{1, \ldots, n\}$ of parties
- Each party $u \in U$ has a private value X_u (scalars, gradients, models..)
- No party is trusted with the data of others
- **Goal:** Estimate $\frac{1}{n} \sum_u X_u$ **while satisfying differential privacy constraints**

Our Contributions

1. Accuracy in the **order of Central DP**
   - Unlike Local DP
2. **Logarithmic** number of messages per party
   - Unlike previous Secure Aggregation [3] [4]
3. **Robustness** against malicious parties

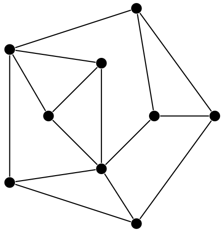
[3] [Bonawitz et al., CSS 2017]
[4] [Bell et al., CSS 2020] is a concurrent work that also provides low communication

Setting

- Users can communicate with others through **secure channels**

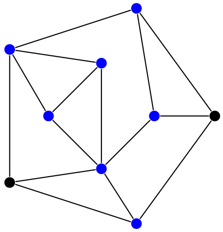
Setting

- Users can communicate with others through **secure channels**
- Messages are modeled by **communication graph** $G = (U, E)$



Setting

- Users can communicate with others through **secure channels**
- Messages are modeled by **communication graph** $G = (U, E)$

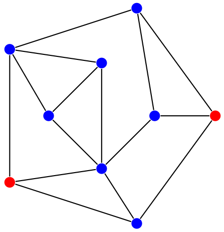


A proportion ρ of honest (but curious) users:

- follow the protocol
- might try to infer information
- do not collude with other users

Setting

- Users can communicate with others through **secure channels**
- Messages are modeled by **communication graph** $G = (U, E)$

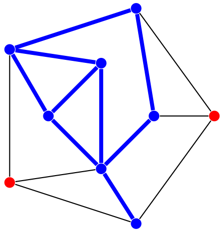


Adversary: a proportion of $(1 - \rho)$ malicious users

- deviate from the protocol and collude among them
- try to (1) infer information and (2) bias the computation
- know the graph G (who communicated with whom)

Setting

- Users can communicate with others through **secure channels**
- Messages are modeled by **communication graph** $G = (U, E)$



The sub-graph of honest users is G^H

- channels whose information the is not seen by the adversary
- not known by honest parties

Protocol

Input: graph G, canceling variance σ_Δ^2, independent variance σ_η^2

for all neighbor pairs $\{u, v\} \in E(G)$ **do**

1a. u and v draw canceling noise term $\delta \sim \mathcal{N}(0, \sigma_\Delta^2)$

1b. set $\Delta_{u,v} \leftarrow \delta$, $\Delta_{v,u} \leftarrow -\delta$

end for

for each user $u \in U$ **do**

2. u draws independent noise term $\eta_u \sim \mathcal{N}(0, \sigma_\eta^2)$

3. u computes $\hat{X}_u \leftarrow X_u + \sum_{u \sim v} \Delta_{u,v} + \eta_u$

end for

4. Average $\hat{X}_1, \ldots, \hat{X}_n$ in the clear (Gossip Avg. or Server)

Algorithm 1: **Gopa** (GOssip for Private Averaging)

- **Unbiased estimate of the average:** $\hat{X}^{avg} = \frac{1}{n} \sum_u \hat{X}_u$ with variance σ_η^2 / n
- Secure Aggregation has a similar structure without independent noise

Properties

- **Privacy with trusted curator utility**
- Logarithmic communication per party
- Robustness against malicious participants

Privacy Guarantees - General Result

Privacy Guarantees - General Result

Theorem (General Result)

GOPA can achieve (ε, δ)-DP ***with (order) trusted curator accuracy*** *when*

- *the sub-graph G^H of honest users* ***is connected***
- *canceling noise σ_Δ^2* ***is large enough***

The required σ_Δ^2 depends on the connectivity of G^H

Privacy Guarantees - General Result

Theorem (General Result)

GOPA can achieve (ε, δ)-DP ***with (order) trusted curator accuracy*** *when*

- *the sub-graph G^H of honest users* ***is connected***
- *canceling noise σ_Δ^2* ***is large enough***

The required σ_Δ^2 depends on the connectivity of G^H

- malicious participants degrade accuracy by a factor $n/\rho n$ compared to central DP

Privacy Guarantees - General Result

Theorem (General Result)

GOPA can achieve (ε, δ)-DP ***with (order) trusted curator accuracy*** *when*

- *the sub-graph G^H of honest users* ***is connected***
- *canceling noise σ_Δ^2* ***is large enough***

The required σ_Δ^2 depends on the connectivity of G^H

- malicious participants degrade accuracy by a factor $n/\rho n$ compared to central DP
- How can users safely construct G to ensure that G^H is good enough?

Privacy Guarantees - General Result

Theorem (General Result)

GOPA can achieve (ε, δ)-DP ***with (order) trusted curator accuracy*** *when*

- *the sub-graph G^H of honest users* ***is connected***
- *canceling noise σ_Δ^2* ***is large enough***

The required σ_Δ^2 depends on the connectivity of G^H

- malicious participants degrade accuracy by a factor $n/\rho n$ compared to central DP
- How can users safely construct G to ensure that G^H is good enough?
- Secure Aggregation solves it at a large communication cost

Properties

- Privacy with trusted curator utility ✓
- **Logarithmic communication per party**
- Robustness against malicious participants

Privacy with Small Communication

- ***k*-out random graph**: each user chooses k neighbors at random
- G^H is sufficiently connected with high probability **even if k is small**

Theorem (k-out Random Graphs)

Let $\varepsilon, \delta \in (0, 1)$ and

- *k **logarithmic in** n*
- *bounded σ_Δ^2 (linear in n)*

*Then GOPA is (ε, δ)-DP with **trusted curator accuracy***

Privacy with Small Communication

- *k*-**out random graph**: each user chooses k neighbors at random
- G^H is sufficiently connected with high probability **even if** k **is small**

Theorem (*k*-out Random Graphs)

Let $\varepsilon, \delta \in (0, 1)$ *and*

- k ***logarithmic in*** n
- *bounded* σ_Δ^2 *(linear in n)*

Then GOPA *is* (ε, δ)*-DP with* ***trusted curator accuracy***

- **Trusted curator accuracy** with **logarithmic number of messages** per user
- k increases with n. of colluders

Illustrations - Communication

Requirements for connected G^H:

In theory:

- 10000 parties, no colluders $\longrightarrow$ **105 messages per party**
- 10000 parties, 50% colluders $\longrightarrow$ **203 messages per party**

In practice (success over 10^5 executions of GOPA)

- 1000 parties, no colluders $\longrightarrow$ **10 messages per party**
- 1000 parties, 50% colluders $\longrightarrow$ **17 messages per party**
- 10^4 parties, 50% colluders $\longrightarrow$ **20 messages per party**

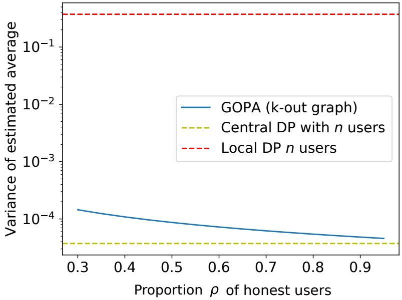
Messages are **only small random seeds** (and not large models/gradients)

Illustrations - Accuracy

$n = 10000$, (ε, δ)-DP, $\delta = 1/(\rho n)^2$

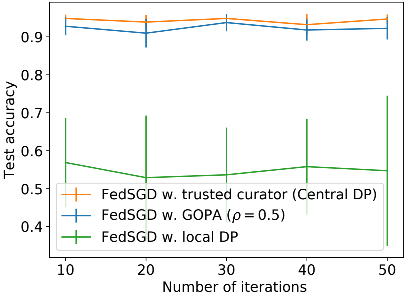
Variance

$(\varepsilon = 0.1)$



Federated SGD for Logistic Regression

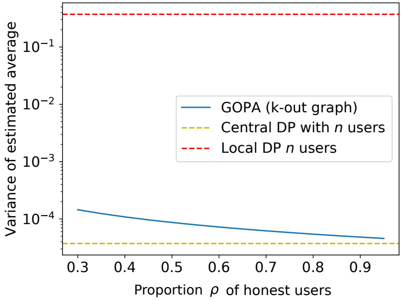
(UCI Housing Dataset, $\varepsilon = 1, \rho = 0.5$)



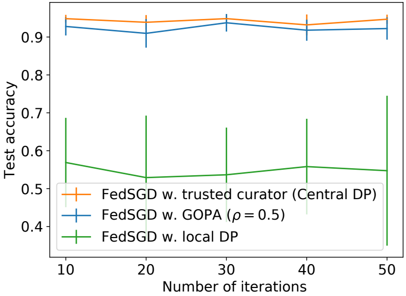
Illustrations - Accuracy

$n = 10000$, (ε, δ)-DP, $\delta = 1/(\rho n)^2$

Variance
($\varepsilon = 0.1$)



Federated SGD for Logistic Regression
(UCI Housing Dataset, $\varepsilon = 1, \rho = 0.5$)



- Gopa is **close to Fed-SGD with trusted curator even with 50% of malicious users**
- LDP has much larger variance and does not arrive to learn anything

Properties

- Privacy with trusted curator utility ✓
- Logarithmic communication per party ✓
- **Robustness against malicious participants**

Preventing Poisoning

Goal: prevent that a malicious user u **poisons** $\hat{X}_u$ (as much as possible)

[5] Pedersen, TP. *Non-interactive and information-theoretic secure verifiable secret sharing.* CRYPTO, 1991.
[6] Cramer, R. *Modular design of secure yet practical cryptographic protocols.* Ph.D. thesis, 1996.

Preventing Poisoning

Goal: prevent that a malicious user u **poisons** $\hat{X}_u$ (as much as possible)

Our Approach:

1. Shared **bulletin board** to publish messages

[5] Pedersen, TP. *Non-interactive and information-theoretic secure verifiable secret sharing.* CRYPTO, 1991.
[6] Cramer, R. *Modular design of secure yet practical cryptographic protocols.* Ph.D. thesis, 1996.

Preventing Poisoning

Goal: prevent that a malicious user u **poisons** $\hat{X}_u$ (as much as possible)

Our Approach:

1. Shared **bulletin board** to publish messages
2. **Cryptographic Commitments** [5]
   - allow to *commit* to a private value without revealing it

[5] Pedersen, TP. *Non-interactive and information-theoretic secure verifiable secret sharing.* CRYPTO, 1991.
[6] Cramer, R. *Modular design of secure yet practical cryptographic protocols.* Ph.D. thesis, 1996.

Preventing Poisoning

Goal: prevent that a malicious user u **poisons** $\hat{X}_u$ (as much as possible)

Our Approach:

1. Shared **bulletin board** to publish messages
2. **Cryptographic Commitments** [5]
   - allow to *commit* to a private value without revealing it
3. **Zero Knowledge Proofs** [6]
   - allow to prove **properties** and **relations** between committed secret values

---

[5] Pedersen, TP. *Non-interactive and information-theoretic secure verifiable secret sharing.* CRYPTO, 1991.
[6] Cramer, R. *Modular design of secure yet practical cryptographic protocols.* Ph.D. thesis, 1996.

Preventing Poisoning (II)

Verification Protocol. Each user $u \in U$:

Preventing Poisoning (II)

Verification Protocol. Each user $u \in U$:

1. Publishes an encrypted log of its computations using **commitments**

Preventing Poisoning (II)

Verification Protocol. Each user $u \in U$:

1. Publishes an encrypted log of its computations using **commitments**
2. Prove **without revealing sensitive information** that:

$$
\begin{aligned}
X_u \text{ is in the correct domain} & \\
\Delta_{u,v} = -\Delta_{v,u}, & \qquad \forall v \text{ neighbor of } u \\
\eta_u \sim \mathcal{N}(0, \sigma_\eta^2), & \qquad \text{(with customizable precision)} \\
\hat{X}_u = X_u + \sum_{u \sim v} \Delta_{u,v} + \eta_u. &
\end{aligned}
$$

using **Zero Knowledge Proofs**.

Preventing Poisoning (II)

Verification Protocol. Each user $u \in U$:

1. Publishes an encrypted log of its computations using **commitments**
2. Prove **without revealing sensitive information** that:

$$\begin{aligned} X_u &\text{ is in the correct domain} \\ \Delta_{u,v} &= -\Delta_{v,u}, && \forall v \text{ neighbor of } u \\ \eta_u &\sim \mathcal{N}(0, \sigma_\eta^2), && \text{(with customizable precision)} \\ \hat{X}_u &= X_u + \sum_{u \sim v} \Delta_{u,v} + \eta_u. \end{aligned}$$

using **Zero Knowledge Proofs**.

- u can lie about X_u, but this is **also true in the central setting**

Preventing Poisoning (II)

Verification Protocol. Each user $u \in U$:

1. Publishes an encrypted log of its computations using **commitments**
2. Prove **without revealing sensitive information** that:

$$
\begin{aligned}
X_u &\text{ is in the correct domain} \\
\Delta_{u,v} &= -\Delta_{v,u}, && \forall v \text{ neighbor of } u \\
\eta_u &\sim \mathcal{N}(0, \sigma_\eta^2), && \text{(with customizable precision)} \\
\hat{X}_u &= X_u + \sum_{u \sim v} \Delta_{u,v} + \eta_u.
\end{aligned}
$$

using **Zero Knowledge Proofs**.

- u can lie about X_u, but this is **also true in the central setting**
- Cryptographic primitives have a **tractable cost**

Takeaways

- A **performant protocol** for Private Aggregation
- Tolerate **large amounts of collusion** (>50%) while keeping its properties
- Also offer **resistance to dropouts** (explained later)

Outline

Focus:

- **Distributed Mean Estimation** under **Differential Privacy** constraints

Contributions:

1. *An accurate, scalable and verifiable protocol for federated differentially private averaging.* Machine Learning, 2022.
   with **Aurélien Bellet** and **Jan Ramon**.
2. *Private sampling with identifiable cheaters.* PoPETS 2023
   with **Florian Hahn**, **Andreas Peter** and **Jan Ramon**
3. *Dropout-Robust Mechanisms for Differentially Private and Fully Decentralized Mean Estimation..* ArXiv preprint, 2025.
   with **Sonia Ben Mokhtar** and **Jan Ramon**.

- Conclusion

Motivation

Verification Protocol of Gopa. Each user $u \in U$:

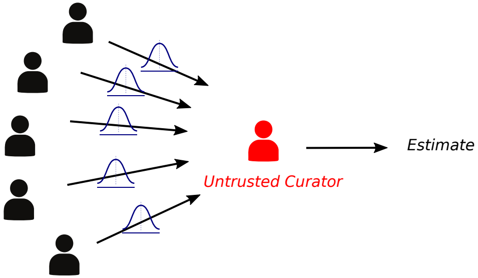
1. Publishes an encrypted log of its computations using **commitments**
2. Prove **without revealing sensitive information** that:

$$\begin{aligned} X_u & \text{ is in the correct domain} \\ \Delta_{u,v} &= -\Delta_{v,u}, && \forall v \text{ neighbor of } u \\ \rightarrow \eta_u &\sim \mathcal{N}(0, \sigma_\eta^2), && \text{(with customizable precision)} \\ \hat{X}_u &= X_u + \sum_{u \sim v} \Delta_{u,v} + \eta_u. \end{aligned}$$

using **Zero Knowledge Proofs**.

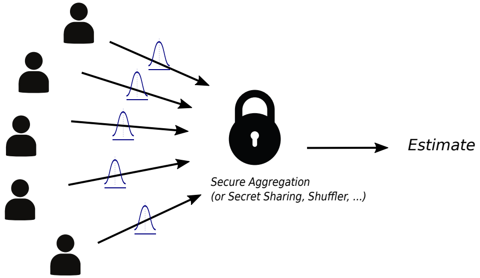
Example: Private Aggregation

1. Each user u samples $\eta_u \sim \mathcal{D}$ to satisfy differential privacy
2. Compute noisy estimate $\sum_u X_u + \eta_u$



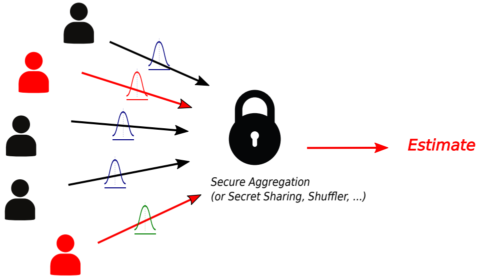
Example: Private Aggregation

1. Each user u samples $\eta_u \sim \mathcal{D}$ to satisfy differential privacy
2. Compute noisy estimate $\sum_u X_u + \eta_u$



Example: Private Aggregation

1. Each user u samples $\eta_u \sim \mathcal{D}$ to satisfy differential privacy
2. Compute noisy estimate $\sum_u X_u + \eta_u$



- Malicious user u can poison X_u, η_u to bias the outcome
- Methods exist to verify that X_u is in the correct domain (e.g. Zero Knowledge Range Proofs)
- **Verifying that $\eta_u \sim \mathcal{D}$ without revealing η_u is less explored** (Especially for the Gaussian distribution)

Our Problem

We study ***secure randomization*** *for privacy preserving protocols*:

- n parties $P_1, \ldots, P_n$
- **adversary:** a static set of malicious colluding parties
- a publicly known distribution $\mathcal{D}$

Verifiable Noise Samples

$P_1, \ldots, P_n$ run a multiparty protocol to **generate a number** $\eta \in \mathbb{R}$ such that, if at least one party is honest:

- η is **unknown to most of the parties**
- all parties **can verify that** $\eta \sim \mathcal{D}$

Two flavors:

- **Private Samples**: **Only one** party P_1 knows η
- **Hidden Samples**: **Nobody** knows $\eta \rightarrow$ is a secret shared among $P_1 \ldots P_n$

Main Contributions

We **propose** protocols for

- **Private Samples** for Gaussian, Laplacian and arbitrary $\mathcal{D}$
- **Hidden Samples** for Gaussian and Laplacian distribution

We **evaluate**

- Gaussian **Private Samples**
- Show that we outperform previous Gaussian secure sampling techniques

While doing so:

- **Propose** novel techniques to prove **non-polynomial**, **finite-precision** relations in zero knowledge.

We prove **malicious security with identifiable abort:**[7]
Our protocols **finish correctly** or **abort if it detects a cheater**

[7] Ishai et al. *Secure multi-party computation with identifiable abort.* Advances in Cryptology–CRYPTO 2014. August 17-21, 2014.

Private Samples: Approach

- Only P_1 knows η

Tools:

- Public Bulletin Board
- Zero Knowledge Proofs (ZKPs): Compressed Σ-Protocols[8]
  Can prove that $\mathbf{C}(\mathbf{x}) = \mathbf{0}$, for a **private** x and **circuit** C
  (non-interactively by the Fiat-Shamir Heuristic)

[8] Attema and Cramer. *Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics.* Advances in Cryptology–CRYPTO 2020

Private Samples: Approach

- Only P_1 knows η

Tools:

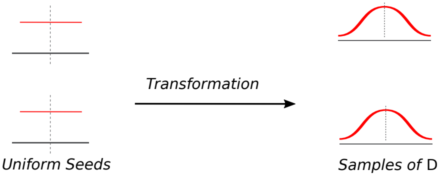
- Public Bulletin Board
- Zero Knowledge Proofs (ZKPs): Compressed Σ-Protocols[8]
  Can prove that $\mathbf{C}(\mathbf{x}) = \mathbf{0}$, for a **private** x and **circuit** C
  (non-interactively by the Fiat-Shamir Heuristic)

If $\mathcal{D}$ is the **uniform distribution** $\mathcal{U}\{0 \dots M\}$:

1. P_1 commits to a private $x \leftarrow_\$ \{0 \dots M\}$
2. All parties jointly generate a public $y \leftarrow_\$ \{0 \dots M\}$
3. P_1 commits to η and **proves that** $\eta = x + y \mod M + 1$ in zero knowledge

[8] Attema and Cramer. *Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics.* Advances in Cryptology–CRYPTO 2020

Private Samples: Approach (II)



For any distribution $\mathcal{D}$:

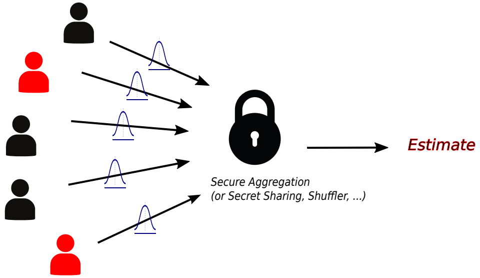
1. Execute the **uniform protocol** to get seeds $u_1, \ldots, u_k$
2. P_1 **proves that** $\eta = \mathit{Transformation}(u_1, \ldots, u_k)$ in ZK
   - **inverse CDF** for any $\mathcal{D}$
   - specialized techniques for some $\mathcal{D}$ (e.g. Gaussian)

For transformations, we propose **iterative approximation** circuits

- Avoid table-lookups and splines
- No preprocessing, few comparisons, customizable precision

Example: Secure Aggregation with Private Samples

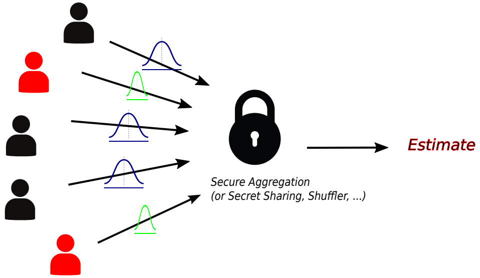
- Every party P_u knows a private term η_u



- The output is unbiased

Example: Secure Aggregation with Private Samples

- Every party P_u knows a private term η_u



- The output is unbiased
- Set S of colluding malicious users know $\{\eta_u\}_{u \in S}$
- Honest users add $n/|S|$ more noise to compensate

Hidden Samples: Approach

- η is secret shared among $P_1, \ldots, P_n$

Tools:

- Public Bulletin Board, ZKPs
- **Arithmetic Secret Sharing (SS)** [9] [10]
  Allows to compute $\mathbf{C}(\mathbf{x})$ for a **secretly shared** x and **circuit** C

[9] Damgård, Ivan, et al. *Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation.*Theory of Cryptography: TCC 2006.

[10] Damgård, Ivan, et al. *Practical covertly secure MPC for dishonest majority–or: breaking the SPDZ limits.* Computer Security–ESORICS 2013

Hidden Samples: Approach

- η is secret shared among $P_1, \ldots, P_n$

Tools:

- Public Bulletin Board, ZKPs
- **Arithmetic Secret Sharing (SS)** [9] [10]
  Allows to compute $\mathbf{C}(\mathbf{x})$ for a **secretly shared** x and **circuit** C

If $\mathcal{D}$ is the **uniform distribution** $\mathcal{U}\{0 \ldots M\}$:

1. Each party P_u draws a private $x_u \leftarrow_\$ \{0 \ldots M\}$
2. $(x_1, \ldots, x_n)$ **already is a hidden draw** of η
   - i.e. $\sum_u x_u \pmod{M+1} \sim \mathcal{U}\{0 \ldots M\}$

For other $\mathcal{D}$:

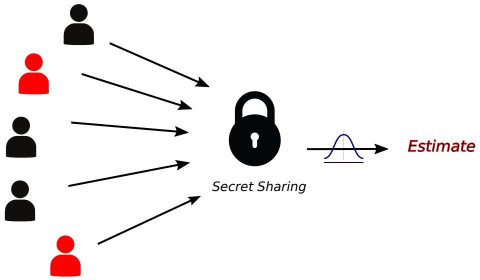
- Generate uniform seeds, **run transformation circuits in SS**

[9] Damgård, Ivan, et al. *Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation.* Theory of Cryptography: TCC 2006.

[10] Damgård, Ivan, et al. *Practical covertly secure MPC for dishonest majority–or: breaking the SPDZ limits.* Computer Security–ESORICS 2013

Example: Secret Sharing with Hidden Samples

- **Hidden Sample**: η is secret shared among $P_1, \ldots, P_n$

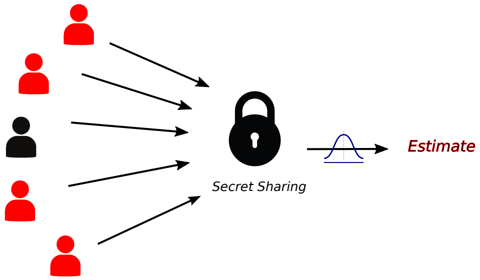


- The output is unbiased
- **Optimal amount of noise** (i.e. as with a trusted curator)

[11] Boenisch, Franziska, et al. *Is Federated Learning a Practical PET Yet?.* arXiv preprint arXiv:2301.04017 (2023).

Example: Secret Sharing with Hidden Samples

- **Hidden Sample**: η is secret shared among $P_1, \ldots, P_n$

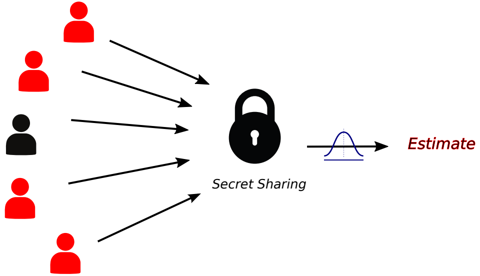


- The output is unbiased
- **Optimal amount of noise** (i.e. as with a trusted curator)
- **No accuracy degradation** even if $n-1$ users collude [11]

[11] Boenisch, Franziska, et al. *Is Federated Learning a Practical PET Yet?.* arXiv preprint arXiv:2301.04017 (2023).

Example: Secret Sharing with Hidden Samples

- **Hidden Sample**: η is secret shared among $P_1, \ldots, P_n$



- The output is unbiased
- **Optimal amount of noise** (i.e. as with a trusted curator)
- **No accuracy degradation** even if $n-1$ users collude [11]
- Expensive in communication

[11] Boenisch, Franziska, et al. *Is Federated Learning a Practical PET Yet?.* arXiv preprint arXiv:2301.04017 (2023).

Evaluation: Private Gaussian Samples

- Widely used in distributed DP (among other applications)

Prior Work [12]: Central Limit Theorem(CLT)

- each sample requires a **large amount** of seeds

We propose methods that require only **one seed per sample**:
Inversion Method:

- **inverse CDF** has no closed form
- approximation with Series (GOPA: InvM-S)
- approximation with Rational Functions (InvM-R)

Box Müller(BM):

- requires **log**, **sqrt**, **sin**, **cos**
- **Polar Method**(PolM) is optimized to avoid **sin**, **cos**

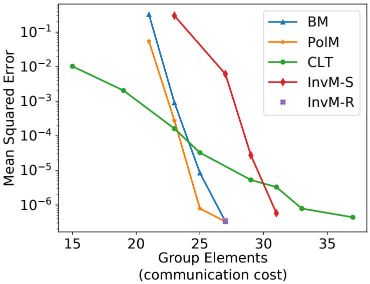
[12] Dwork et al. *Our Data, Ourselves: Privacy Via Distributed Noise Generation.* EUROCRYPT 2006.

Evaluation: Private Gaussian Samples

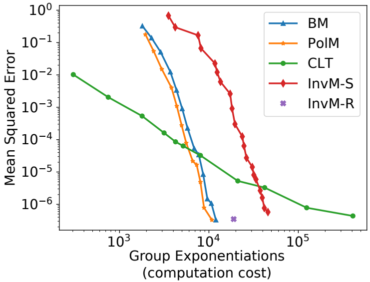
We compare (for different precision parameters)

- Statistical quality: MSE to an ideal Gaussian over 10^7 samples
- Cryptographic cost of ZKPs per sample

Communication



Computation (Prover)



- If quality is more important: PolM and BM (< 0.5s, < 1 KB)
- Otherwise: CLT can generate fast samples (10 ms)

Takeaways

Assuming the **existence of a bulletin board**

- Formalize secure randomness generation
- Propose sampling procedure for arbitrary distributions
- Generate private Gaussian samples efficiently

Outline

Focus:

- **Distributed Mean Estimation** under **Differential Privacy** constraints

Contributions:

1. *An accurate, scalable and verifiable protocol for federated differentially private averaging.* Machine Learning, 2022.
   with **Aurélien Bellet** and **Jan Ramon**.
2. *Private sampling with identifiable cheaters.* PoPETS 2023
   with **Florian Hahn**, **Andreas Peter** and **Jan Ramon**
3. *Dropout-Robust Mechanisms for Differentially Private and Fully Decentralized Mean Estimation.*. ArXiv preprint, 2025.
   with **Sonia Ben Mokhtar** and **Jan Ramon**.

- Conclusion

Distributed Mean Estimation under DP

Problem: Private Mean Estimation

- Set $U = \{1, \ldots, n\}$ of parties
- Each party $u \in U$ has a private value X_u (scalars, gradients, models..)
- No party is trusted with the data of others
- **Goal:** Estimate $\frac{1}{n} \sum_u X_u$ **while satisfying differential privacy constraints**

New unexpected events:

- **Parties might drop out in the middle of the computation**

GOPA

Input: graph G, canceling variance σ_Δ^2, independent variance σ_η^2

for all neighbor pairs $\{u, v\} \in E(G)$ **do**

1a. u and v draw canceling noise term $\delta \sim \mathcal{N}(0, \sigma_\Delta^2)$

1b. set $\Delta_{u,v} \leftarrow \delta$, $\Delta_{v,u} \leftarrow -\delta$

end for

for each user $u \in U$ **do**

2. u draws independent noise term $\eta_u \sim \mathcal{N}(0, \sigma_\eta^2)$

3. u computes $\hat{X}_u \leftarrow X_u + \sum_{u \sim v} \Delta_{u,v} + \eta_u$

end for

4. Average $\hat{X}_1, \ldots, \hat{X}_n$ in the clear (Gossip Avg. or Server)

Algorithm 2: **Gopa** (GOssip for Private Averaging)

- **Unbiased estimate of the average:** $\hat{X}^{avg} = \frac{1}{n} \sum_u \hat{X}_u$ with variance σ_η^2 / n
- Secure Aggregation has a similar structure but with cryptographic noise

Drop-out Harm

If the set D of parties drop-out before finishing.

$$\hat{X}^{avg} = \sum_{u \in O} \hat{X}_u = \sum_{u \in O} \hat{X}_u + \eta_u + \sum_{v \in D \cap N(u)} \Delta_{v,u}$$

Where O is the set of online parties.

Reparation

- In Secure Aggregation
  - abort and re-start
  - use a **centrally orchestrated recovery**
- In Gopa
  - the **harm is bounded** → depends on σ_Δ^2
  - a **recovery mechanism** is also possible → **partially mitigates the problem**

Our Contributions

1. Accuracy in the **Order of Central DP** when no drop-outs occur
   - Unlike Local DP
2. **Fully Decentralized Setting**
   - Unlike Secure Aggregation
3. Better **Robustness to Drop-outs** than other decentralized protocols
   - with respect to previous protocols (e.g. GOPA)
4. Low Communication Cost
   - Comparable to GOPA

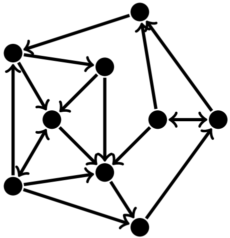
Setting

- Synchronous Gossip: T gossip rounds
- At each round $t \in \{1, \ldots, T\}$:

Setting

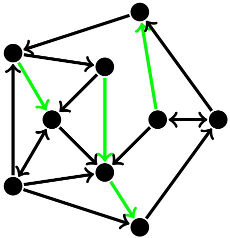
- Synchronous Gossip: T gossip rounds
- At each round $t \in \{1, \ldots, T\}$:
  - model interaction with directed graphs $G_t = (P, E_t)$
  - weighted adjacency matrices $W_t \in \mathbb{R}^{n \times n}$:

$$W_{t;j,i} \begin{cases} > 0 & \text{if } (i,j) \in E_t \\ = 0 & \text{otherwise} \end{cases}$$



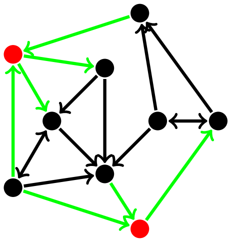
Setting

- Synchronous Gossip: T gossip rounds
- At each round $t \in \{1, \ldots, T\}$:
  - set O_t of messages are *observed*
  - have **crucial impact in privacy**



Setting

- Synchronous Gossip: T gossip rounds
- At each round $t \in \{1, \dots, T\}$:
  - $C \subset P$ parties are *corrupted*
  - observe all incoming and outgoing messages
  - Assume **Semi-honest:**
    - **collude**
    - don't deviate from the protocol
  - W_t **is known** by the adversary
    - as in [Cyffers et al., ICML 2024]



Gossip Protocols

Input: $X \in [0,1]^n$, $W_1, \ldots, W_T \in \mathbb{R}^{n \times n}$
for all $i \in U$ **do**
  $y_i^{(0)} \leftarrow X_i$
end for
for $t \in \{1 \ldots T\}$ **do**
  for all $i \in U$ **do**
  $y_i^{(t)} \leftarrow \sum_{j \in U} W_{t;i,j} y_j^{(t-1)}$
end for

Algorithm 3: Classic (Synchronous) Gossip

Gossip Averaging [a]

If $W_1, \ldots, W_T$

- have good spectral properties

then it converges to $\frac{1}{n} \sum_{i \in U} X_i$.

- not private

[a] [Boyd, Stephen, et al. "Randomized gossip algorithms." IEEE transactions on information theory, 2006]

Gossip Protocols

Input: $X \in [0,1]^n$, $W_1, \ldots, W_T \in \mathbb{R}^{n \times n}$
for all $i \in U$ **do**
  Sample $\eta_i \sim \mathcal{N}(0, \sigma^2_{ldp})$
  $y_i^{(0)} \leftarrow X_i + \eta_i$
end for
for $t \in \{1 \ldots T\}$ **do**
  for all $i \in U$ **do**
  $y_i^{(t)} \leftarrow \sum_{j \in U} W_{t;i,j} y_j^{(t-1)}$
end for

Algorithm 4: Muffliato

Muffliato [a]

- good privacy and scalability

However,

- accurate for **relaxed DP**
- **inaccurate** in **our DP setting** (as in LDP)

[a] [Cyffers et al, NeurIPS 2022]

Gossip Protocols

Input: $X \in [0,1]^n$, $W_1, \ldots, W_T \in \mathbb{R}^{n \times n}$
for all $i \in U$ **do**
  Sample $\eta_i^\star \sim \mathcal{N}(0, \sigma_\star^2)$
  Sample $(z_{i,1}, \ldots, z_{i,T}) \sim \mathcal{D}(X_i + \eta_i^\star)$
  $y_i^{(0)} \leftarrow z_{i,1}$
end for
for $t \in \{1 \ldots T\}$ **do**
  for all $i \in U$ **do**
  $y_i^{(t)} \leftarrow \sum_{j \in U} W_{t;i,j} y_j^{(t-1)} + z_{i,t}$
end for
Compute $\frac{1}{n} \sum_{i \in P} y_i^{(T)}$ with Gossip (Alg. 3)

Algorithm 5: Incremental Averaging (IncA)

Incremental Averaging (IncA):

- $(z_{i,1}, \ldots, z_{i,T}) \sim \mathcal{D}(X_i + \eta_i^\star)$
  - $\sum_{t=1}^{T} z_{i,t} = X_i + \eta_i^\star$
  - **protect privacy**
  - **don't harm accuracy**
  - have **small variance**
  - **robust to drop-outs**
- If $W_1 \ldots W_T$ are col. stochastic

$$\frac{1}{n} \sum_{i \in U} y_i^{(T)} = \frac{1}{n} \sum_{i \in U} X_i + \eta_i^\star$$

- $\eta_i^\star$ has small variance

Gossip Protocols

Input: $X \in [0,1]^n$, $W_1, \ldots, W_T \in \mathbb{R}^{n\times n}$
for all $i \in U$ **do**
  Sample $\eta_i^\star \sim \mathcal{N}(0, \sigma_\star^2)$
  Sample $\eta_{i,1} \ldots \eta_{i,T} \sim \mathcal{N}(0, \sigma_\Delta^2)$
  $y_i^{(0)} \leftarrow \frac{1}{T}(X_i + \eta_i^\star) + \eta_{1,1}$
end for
for $t \in \{1 \ldots T-1\}$ **do**
  for all $i \in U$ **do**
  $y_i^{(t)} \leftarrow \sum_{j\in U} W_{t;i,j} y_j^{(t-1)} + \frac{1}{T}(X_i + \eta_i^\star) - \eta_{i,t} + \eta_{i,t+1}$
end for
$y_i^{(T)} \leftarrow \sum_{j\in U} W_{T;i,j} y_j^{(T-1)} - \eta_{i,T}$
Compute $\frac{1}{n}\sum_{i\in P} y_i^{(T)}$ with Gossip (Alg. 3)

Algorithm 6: Incremental Averaging (IncA)

Incremental Averaging (IncA):

- $(z_{i,1}, \ldots, z_{i,T}) \sim \mathcal{D}(X_i + \eta_i^\star)$
  - $\sum_{t=1}^T z_{i,t} = X_i + \eta_i^\star$
  - **protect privacy**
  - **don't harm accuracy**
  - have **small variance**
  - **robust to drop-outs**
- If $W_1 \ldots W_T$ are col. stochastic

$$\frac{1}{n}\sum_{i\in U} y_i^{(T)} = \frac{1}{n}\sum_{i\in U} X_i + \eta_i^\star$$

- $\eta_i^\star$ has small variance

Privacy: Abstract Result

Given $\mathcal{W} = \{W_1, \ldots, W_T\}$ the adversary can see:

$$BX + A\eta = y_{obs}$$

where

- X, η: unknowns
- $B(\mathcal{W})$, $A(\mathcal{W})$: known coefficients
- $y_{obs} = \{(y_i^{(t)}) : i \text{ was observed at iteration } t\}$
- η eta should have large dimension for privacy

Privacy: Abstract Result

Given $\mathcal{W} = \{W_1, \ldots, W_T\}$ the adversary can see:

$$BX + A\eta = y_{obs}$$

where

- X, η: unknowns
- $B(\mathcal{W}), A(\mathcal{W})$: known coefficients
- $y_{obs} = \{(y_i^{(t)}) : i \text{ was observed at iteration } t\}$
- η eta should have large dimension for privacy

Theorem (Abstract Result)

Let $\Sigma_\eta = var(\eta)$. IncA is (ϵ, δ)-DP if

$$t^\top (A\Sigma_\eta A^\top)^{-1} t < \frac{\epsilon^2}{2\ln(1.25/\delta)} \quad \textit{for all columns } t \textit{ of } B.$$

Privacy: Abstract Result

Given $\mathcal{W} = \{W_1, \ldots, W_T\}$ the adversary can see:

$$BX + A\eta = y_{obs}$$

where

- X, η: unknowns
- $B(\mathcal{W}), A(\mathcal{W})$: known coefficients
- $y_{obs} = \{(y_i^{(t)}) : i \text{ was observed at iteration } t\}$
- η eta should have large dimension for privacy

Theorem (Abstract Result)

Let $\Sigma_\eta = var(\eta)$. IncA is (ϵ, δ)-DP if

$$t^\top (A\Sigma_\eta A^\top)^{-1} t < \frac{\epsilon^2}{2\ln(1.25/\delta)} \quad \text{for all columns } t \text{ of } B.$$

- Tight accounting of ϵ, δ based on the structure of correlations

Privacy: Central DP accuracy

For all $(i, t) \in P \times [0, T-1]$, let

$$a^{(i,t)} := W_{t;:,i} - \mathbb{1}_i \in \mathbb{R}^n$$

(associated with the outgoing edges of party i at iteration t)

and

$$H := \left\{ a^{(i,t)} : (i,t) \in P \times [0, T-1] \text{ and } y_i^{(t)} \textbf{ is not observed} \right\}$$

Theorem (Positive results)

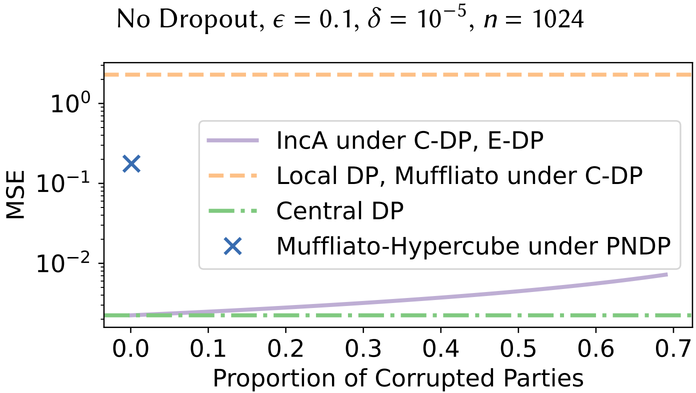
If

- σ_Δ^2 ***sufficiently large*** *and*
- *H has at least $n_H - 1$* ***linearly independent*** *vectors*

then

- *IncA is (ϵ, δ)-DP* ***with Central DP accuracy****.*

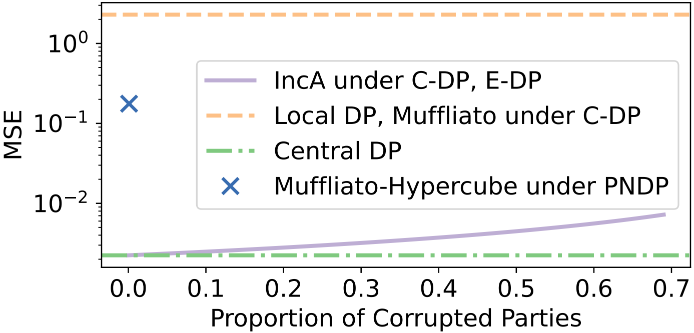
Experiments: Accuracy without Drop-out



- matches accuracy of GOPA and Secure Aggregation
- solely relaxing to PNDP is substantially less accurate

Experiments: Accuracy without Drop-out

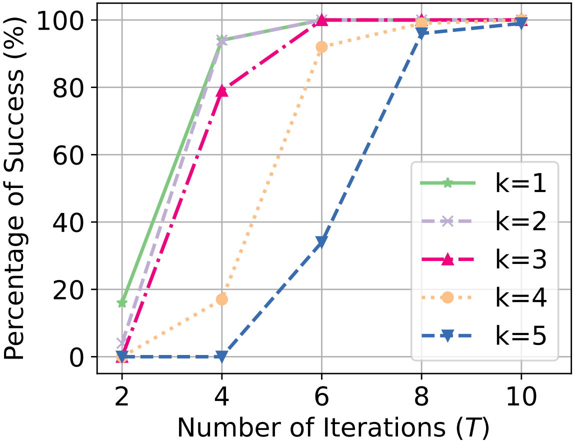
No Dropout, $\epsilon = 0.1$, $\delta = 10^{-5}$, $n = 1024$



- matches accuracy of GOPA and Secure Aggregation
- solely relaxing to PNDP is substantially less accurate
- **When is this accuracy achieved?**

Best topologies without Drop-out

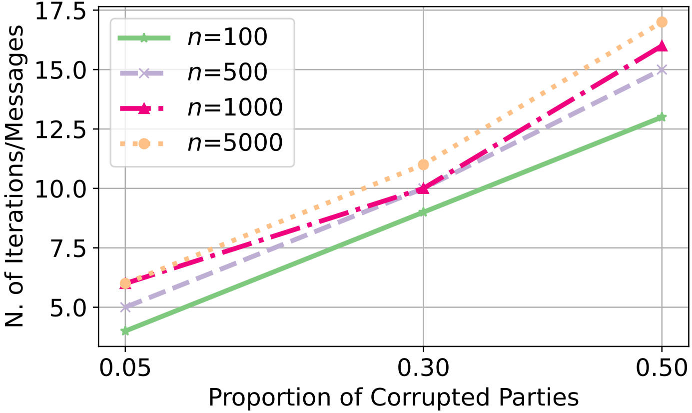
- G_t is k-out graph for each $t \in \{1, \dots, T\}$
- 30% Corrupted Parties (right), No Dropout, 100 simulations, $n = 100$,



- + iterations ⟶ + chance of success
- + dynamic is the graph ⟶ + likely is **diversity of exchanges**
- Lower k ⟶ smaller communication cost

Communication without Dropout

$k = 1$, 100% of success over 10^5 runs

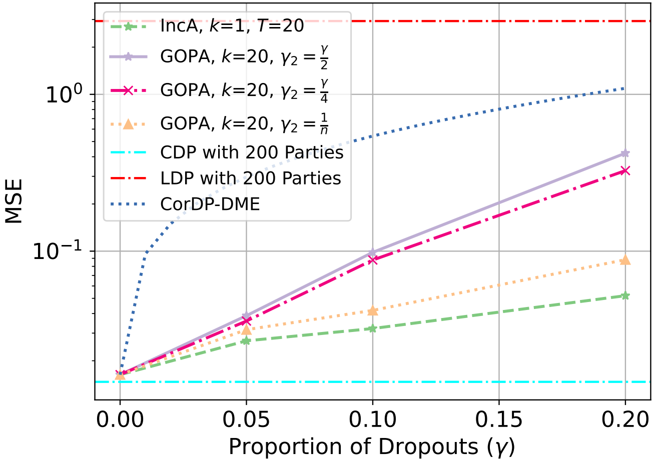


- **Low communication** even with **large amount of colluders**

Performance with Dropout

Comparison with **GOPA** for similar communication and **CorDP-DME**

10% corrupted parties, $n = 200$, $\epsilon = 0.2$, $\delta = 10^{-}5$



- increasing T increase the accuracy of IncA
- Best performance of IncA is with $k = 1$
- IncA outperforms the other protocols

Negative results

If

1. the graph is static ($W_1 = W_2 = \cdots = W_2$)
2. the adversary observes
   - **only 2** nodes during all execution (is easy with static graphs)

then it is **not possible to obtain CDP accuracy with our previous result**.

- **static graphs $\rightarrow$ not sufficient exchange diversity**

Takeaways

- DP-DME can be done **canceling noise across iterations**
- is shown to be **accurate**, **communication efficient** and **robust to collusion**
- **incremental injection** reduces **the variance of canceling noise**
- **low variance** increase **robustness to parties dropping-out**

Outline

Focus:

- **Distributed Mean Estimation** under **Differential Privacy** constraints

Contributions:

1. *An accurate, scalable and verifiable protocol for federated differentially private averaging.* Machine Learning, 2022.
   with **Aurélien Bellet** and **Jan Ramon**.
2. *Private sampling with identifiable cheaters.* PoPETS 2023
   with **Florian Hahn**, **Andreas Peter** and **Jan Ramon**
3. *Dropout-Robust Mechanisms for Differentially Private and Fully Decentralized Mean Estimation.*. ArXiv preprint, 2025.
   with **Sonia Ben Mokhtar** and **Jan Ramon**.

- Conclusion

Conclusion

Presented **correlated noise approaches**:

- Can substantially **increase accuracy of DP mechanisms**
- Hit a **good balance between noise variance and communication**
- **Variance** can be **further reduced with incremental injection**
- **Non-cryptographic noise** can **withstand failures**

Using a **bulletin board** one can prove

- correct computations **via ZKPs**
- **randomized behaviors**

with **tractable in communication and computation** cost.

Perspectives

Further **improve current work**:

- Dropout noise correction on higher level systems
- Incremental averaging: Increase the number of interactions per iteration
- Incremental avg. (II): Theoretical bounds of correlated noise variance

Use correlated noise for **other types of transformation**

- Decentralized SGD [13]
- Across ML Iterations [14]

Fine-grained analysis of the cost of a bulletin board

[13] Allouah, Youssef, et al. "The Privacy Power of Correlated Noise in Decentralized Learning." ICML 2024

[14] Kairouz, Peter, et al. "Practical and private (deep) learning without sampling or shuffling." ICML 2021.

Perspectives (II)

Increase robustness against poisoning on X_u:

- Byzantine Aggregation [15]
- Verification of local computations [16]
- Verification of data correctness across time

Accurately estimate the threats:

- View
- Knowledge
- Computational Capabilities

of the adversary.

---

[15] Allouah, Youssef, Rachid Guerraoui, and John Stephan. "Towards Trustworthy Federated Learning with Untrusted Participants."

[16] Xing, Zhibo, et al. "Zero-knowledge proof meets machine learning in verifiability: A survey.", arXiv 2023

Thank you!

Questions?