DP-BREM: Differentially-Private and Byzantine-Robust Federated Learning with Client Momentum

Xiaolan Gu, ***Ming Li***, Li Xiong

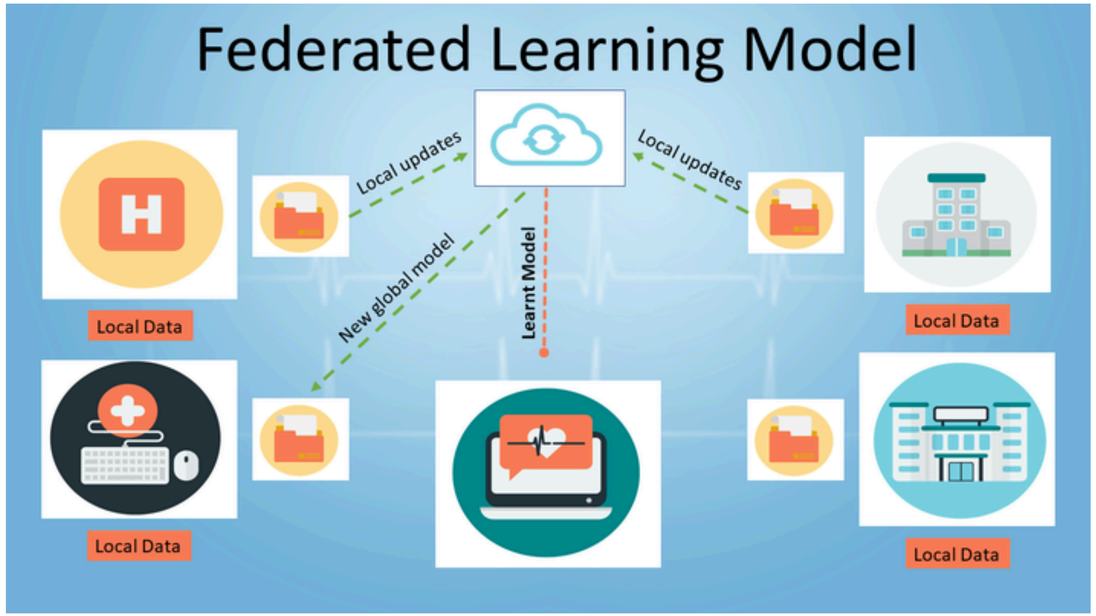
THE UNIVERSITY OF ARIZONA

EMORY UNIVERSITY

USENIX Security 2025

Federated Learning

Federated Learning (FL): Multiple parties collaboratively train a shared model without sharing local data.



How it works?

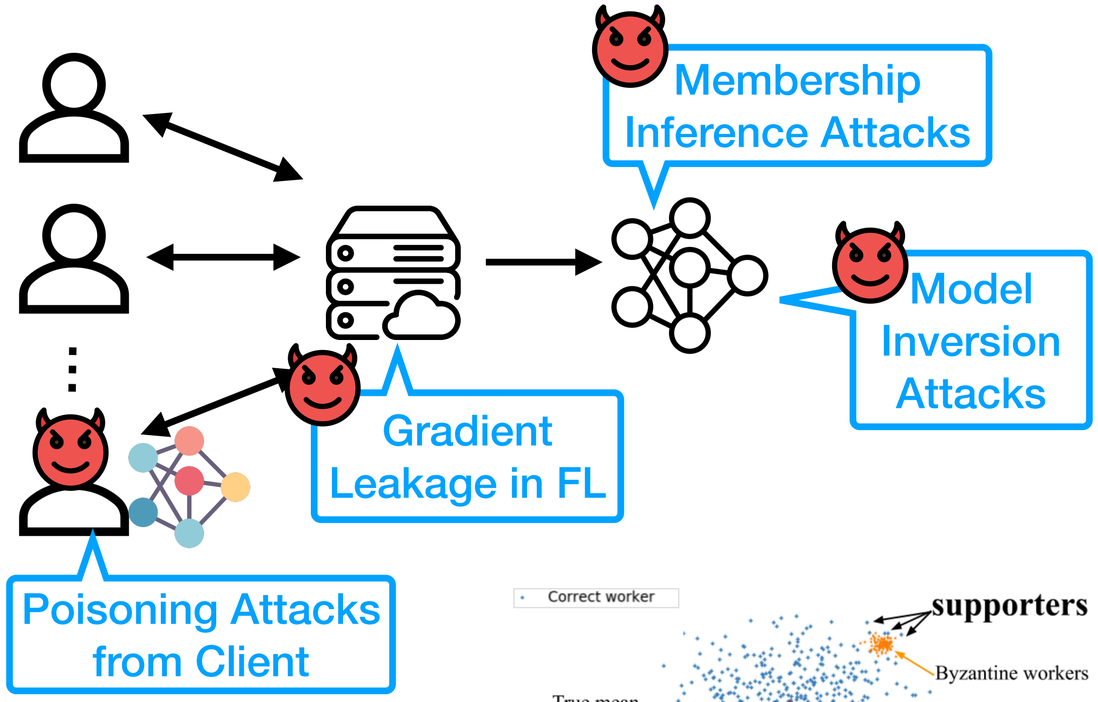
- Clients train locally and send model updates
- Server aggregates updates (e.g., FedAvg)

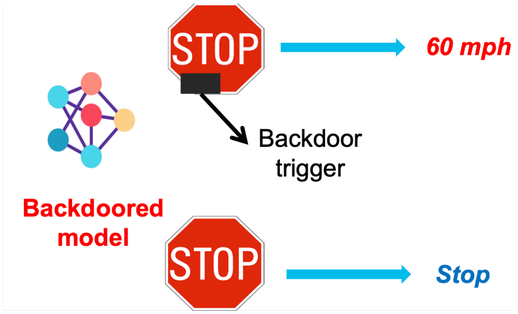
Deployment Scenarios:

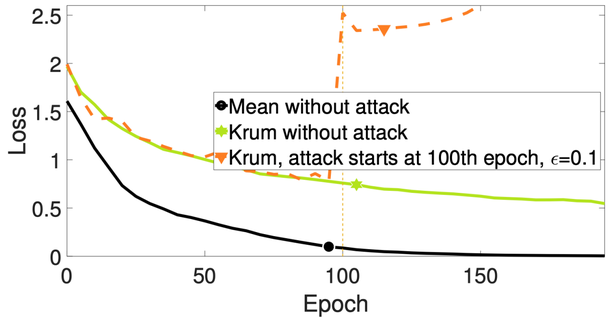
- Cross-device: Many devices (e.g., phones), small + noisy data
- Cross-silo: Few institutions (e.g., hospitals), larger + stable data (main focus of this paper)

FL enhances privacy by keeping data local—but model updates may still leak sensitive information and be vulnerable to attacks.

Threats to ML/FL







Privacy Attacks *(mitigated by Differential Privacy)*

- Membership Inference Attacks: Infer whether a specific data point was used in training [Shokri et al, S&P' 17]
- Model Inversion Attacks: Reconstruct sensitive input data from model [Fredrikson et al, CCS' 15]
- Gradient Leakage in FL: Recover private training data from shared gradients [Zhu et al, NeurIPS' 19]

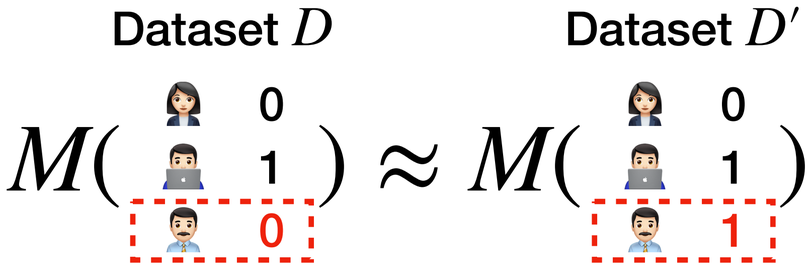
Model Poisoning Attacks in FL

- Backdoor Attacks: Embed hidden triggers to cause targeted misclassifications at inference time
- Byzantine Attacks: Disrupt global model convergence by injecting malicious updates (main focus of this paper — more general and potentially more damaging)

Differential Privacy (DP)

[Dwork, 2006] A mechanism M satisfies ϵ-DP if the output distribution is nearly the same for any two neighboring datasets D, D' (differing by one individual's data), where smaller ϵ indicates stronger privacy

$$\frac{\Pr(M(D) \in O)}{\Pr(M(D') \in O)} \leqslant e^{\epsilon}$$



In ML/FL, the output is the trained model (e.g., model parameters). DP ensures that an attacker cannot tell whether any particular record was used in training.

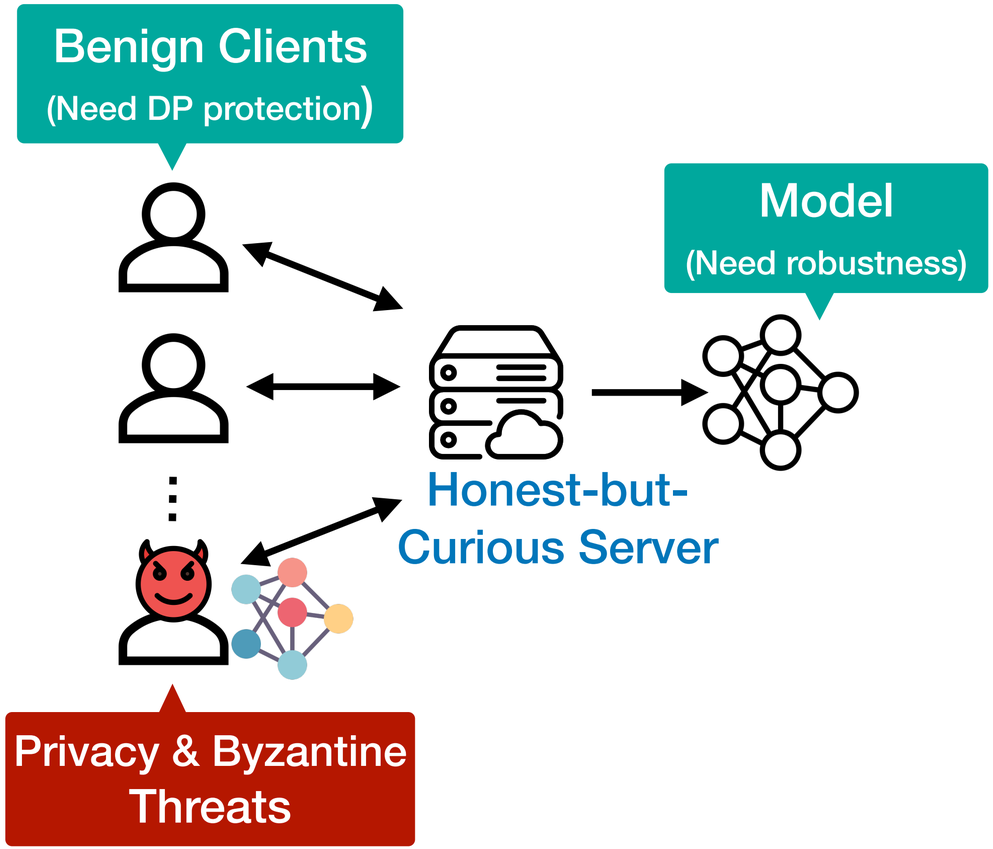
Client-level vs. Record-level DP

- **Client-level DP**: protects a client's dataset, requiring more noise to hide the client's participation (e.g., a device)
- **Record-level DP**: protects individual data points within each client, ensuring that the model does not reveal whether a specific record was used in training. (This is our focus in cross-silo FL)

Applying DP to ML/FL

- **DP-SGD** [Abadi et al, CCS' 16] applies to centralized ML by adding noise during training on server-held data..
- **DP-FedAvg/DP-FedSGD** [McMahan et al, ICLR' 18] Extend DP to FL with central DP, assuming a trusted server adds noise.

Problem Statement

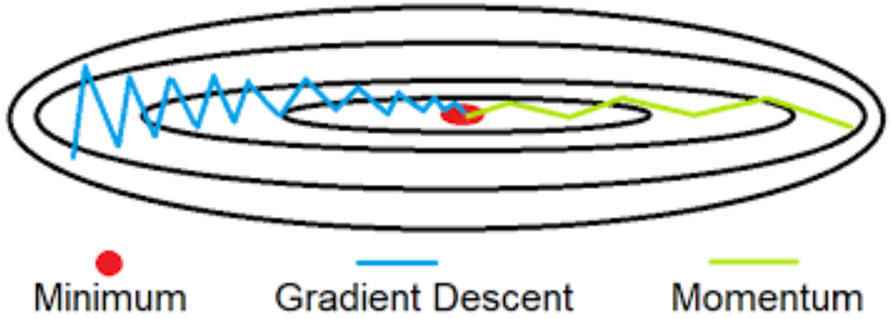


- **Privacy Threats**
  - Server cannot see local data but may infer it.
  - Some clients may collude to learn private data.
- **Byzantine Threats**
  - Server is honest (not perform Byzantine attacks)
  - Some malicious clients can send bad updates, but cannot affect other benign clients.
- **Goal**
  - **Record-level differential privacy (DP)**
  - **Robustness against Byzantine clients**

Challenges

- **Most Byzantine-robust FL aggregators are median-based:**
  - Limitation 1: recent attacks [Baruch et al, NeurIPS' 19] and [Xie et al, 2020] show these methods can be easily broken by carefully crafted malicious updates.
  - Limitation 2: median aggregation is incompatible with DP-SGD (which is average-based), leading to poor privacy-utility tradeoff.
- **Balancing privacy, robustness, and utility is fundamentally challenging**
  - Prior work typically optimizes only two out of the three, but our goal is to achieve all three simultaneously.
  - **Our high-level idea**:
    - Begin with a **robust aggregator** to defend against Byzantine attacks
    - **Add DP noise** to ensure formal privacy guarantees
    - **Apply secure computation techniques** to achieve central DP without requiring a trusted server

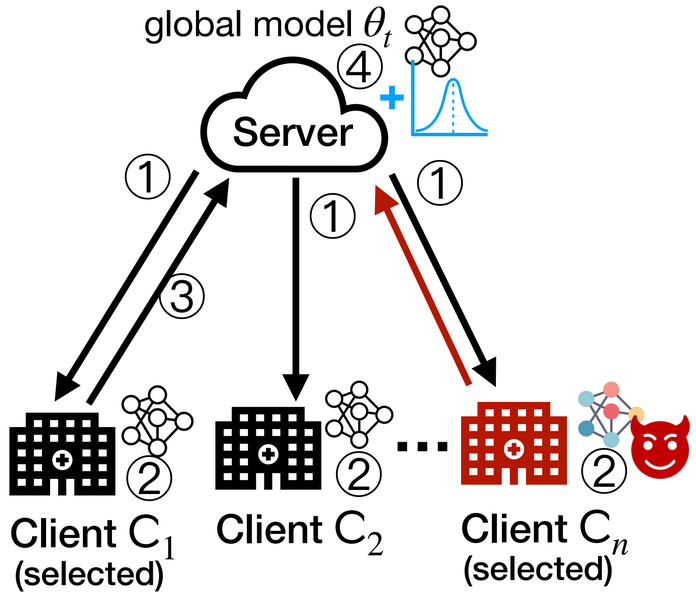
Towards Robust and Private FL



- **Learning From History (LFH)** [Karimireddy et al, ICML' 21]
  - A robust aggregator using **client momentum** — averaging each client's updates over time.
  - This **reduces honest clients' variance** and **amplifies small malicious changes** that accumulate gradually, improving Byzantine detection.
  - **Average-based**, making it more compatible with DP-SGD (though integration remains non-trivial).

- **Where to add DP noise: Local vs. Central?**
  - **Local noise** (added by clients): Makes it harder to distinguish between honest and Byzantine clients → **weaker robustness** under attack.
  - **Central noise** (added by server): Requires less noise for the same DP guarantee → **better utility** [*However, this requires a strong trust assumption, which can be mitigated using secure computation techniques*]

Our Framework: DP-BREM



Threats from Malicious Clients
1) infer other's private data
2) implement Byzantine attack

① Server broadcasts current model parameter θ_t

② Clients compute gradients and momentum

[DP design: record-level gradient clipping]

③ Selected clients send momentum to server

④ Server aggregates momentum to update model

[DP design: adding noise on the aggregate]

Privacy analysis challenge:
- **Gradient is clipped** per record
- But noise is added on **aggregated momentum** — making sensitivity analysis non-trivial.

Trust Assumption on the Server:

- **DP-BREM** assumes a **trusted server** that does not attempt to infer private data.
- We extend this to **DP-BREM+**, which assumes a **curious server** and incorporates **secure computation** (with distributed and jointly generated DP noise by clients using verifiable secret sharing) to protect client privacy.

Privacy and Robustness Analysis

Privacy Guarantee: record-level DP

- **DP-SGD needs bounded sensitivity**, ensured by clipping per-sample gradients and adding noise to their average.
- **Challenge**: Momentum mixes gradients across rounds, so **DP noise is added after mixing**, not directly on clipped gradients — making sensitivity analysis tricky.
- **Our insight**: We show that **clipping still bounds the sensitivity** of the aggregated momentum, enabling **proper noise calibration and DP guarantee**.

Robustness Guarantee: $O\left(\rho\sqrt{1+|\mathcal{B}|+\sqrt{d}\sigma}\right)$

Byzantine Clients — DP noise

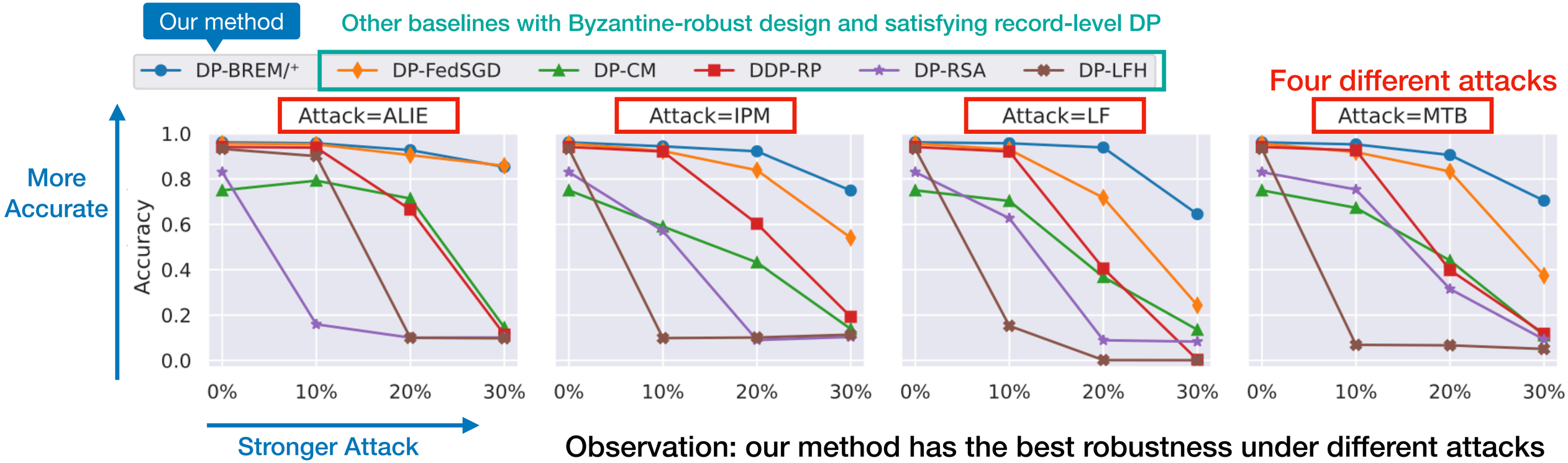
- Both **Byzantine client attacks** and **DP noise** affect convergence, but their **impact is additive**
- This leads to **faster convergence and better utility** compared to local DP, where noise and attacks amplify each other due to multiplication effects (details in the paper)

Experimental Results: Robustness

Datasets: MNIST, CIFAR-10 and FEMNIST

MNIST Results: varying percentage of malicious Byzantine clients (fixing $\epsilon = 3$)

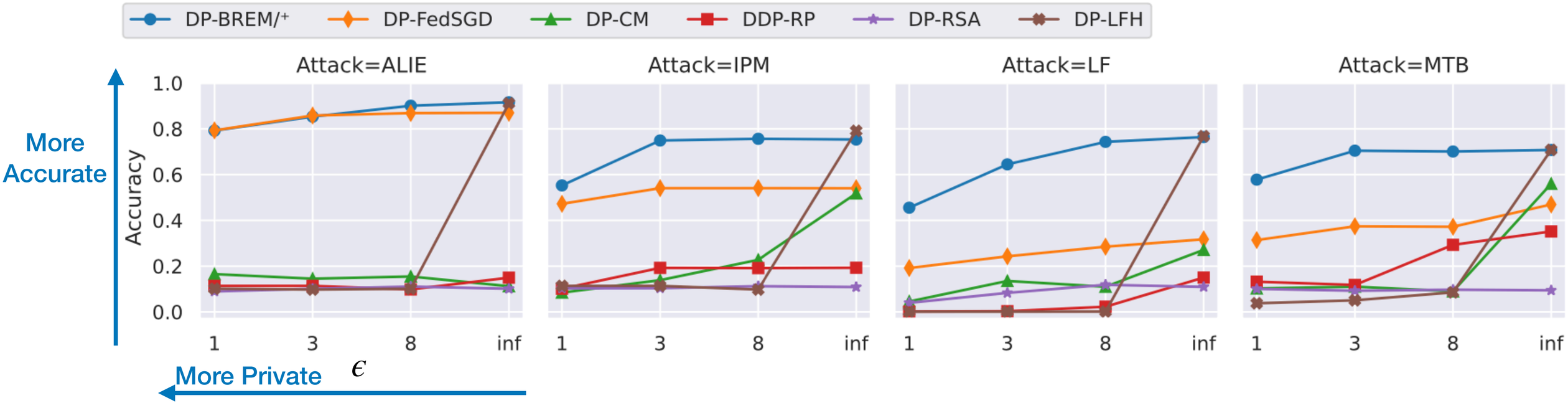
DP-FedSGD: Client-level clipping with DP
DP-CM: Median-based aggregator with DP
DDP-RP: Range-proof mechanism with distributed DP
DP-RSA: Sign-based aggregation with DP
DP-LFH: LFH aggregator with local DP



Observation: our method has the best robustness under different attacks

Experimental Results: Privacy-Utility Tradeoff

MNIST Results: varying the privacy budget ϵ for DP (fixing Byzantine client ratio 30%)



We have similar observations for other datasets, including CIFAR-10 and FEMNIST

Summary

- We proposed **DP-BREM**, a differentially private FL protocol using a **momentum-based robust aggregator**, where the server adds noise to the aggregated momentum.

- To relax the server trust assumption, we developed **DP-BREM+** by integrating **secure computation** (see the full paper for details).

- Our analysis on **privacy**, **convergence**, and **security**, along with **experimental results**, demonstrates that DP-BREM achieves a better privacy-utility tradeoff and stronger Byzantine robustness compared to existing methods.

Future Work

- Extend the framework to support other types of robust aggregators.

Thank you!

Q&A