DeTrust-FL: Privacy-Preserving Federated Learning in Decentralized Trust Setting

Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, Swanand Kadhe, and Heiko Ludwig

IBM Research Almaden, San Jose, CA, USA

2022 IEEE International Conference on Cloud Computing (IEEE CLOUD'22)

BARCELONA, SPAIN
JULY 11-15, 2022

IBM **Research**
Almaden

Data Usage Status

Data Use is hampered by

Data Islands (disconnected data silos)
and
Regulatory Compliance.

Data Islands

Examples of Regulations around the World

Privacy of Data is IMPORTANT

Increasingly, IBM's customers (particularly in the financial and healthcare industries) take the privacy of their consumers' data seriously.

Federated Learning (FL) has emerged as a promising learning paradigm.

Address privacy concerns and satisfy regulatory compliance

Thoroughly utilize the value of data among data islands

Federated Learning

More Data, Better Models

Goal: collaboratively train a machine learning model without sharing/revealing training data
Nascent Field: Google coined the term in 2016

Privacy Concern and Legislations

IoT, smartphones, GDPR, HIPAA

Competitors

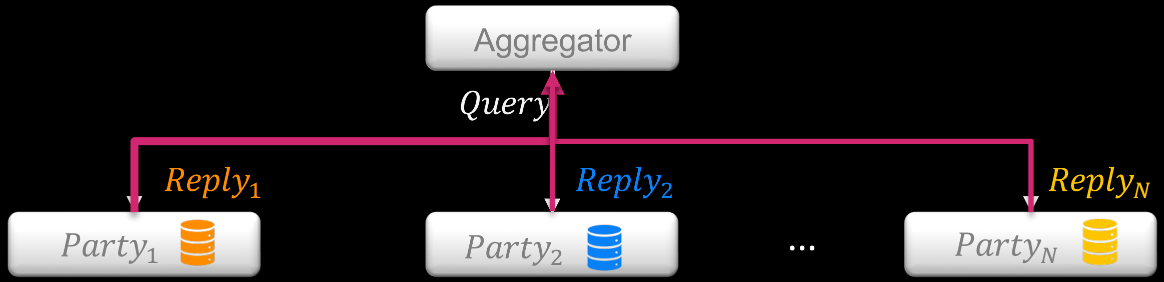
Cable companies, banks, …

Connectivity Constraints

Robots in Mars, Data warehouses, …

Federated Learning Framework Overview

$$Model: f(Reply_1\ Reply_2, \ldots, Reply_N)$$



$Reply_1 = Query(\ )$ $Reply_2 = Query(\ )$ $Reply_N = Query(\ )$

$Query$: Information about the overall data necessary to learn a predictive model
e.g., weights, gradients, counts for decision trees.

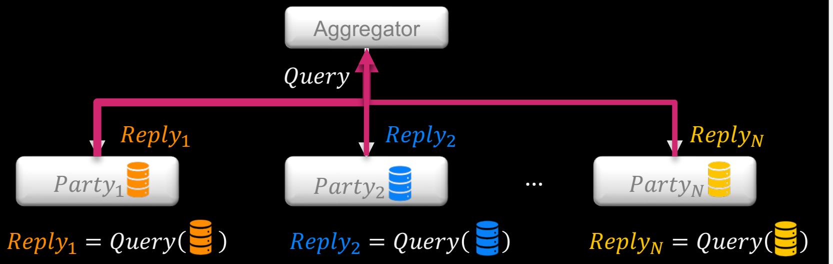
Note that data from each party is not shared!
It remains where it is stored.

Inference threats to FL

In untrusted environments adversaries may try to infer information by analyzing other parties' replies

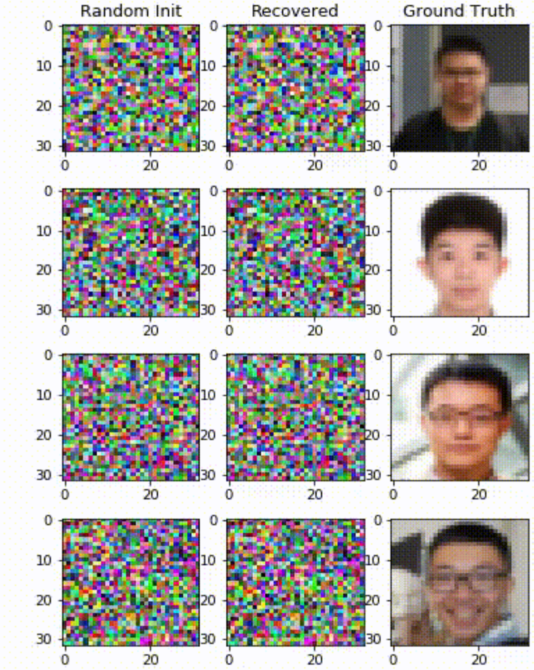
Inferences over replies

Model: $f(Reply_1\ Reply_2, \dots, Reply_N)$



[1] Le Trieu Phong et. al. 2018. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption
[2] IG: Inverting Gradients (NeurIPS 2020),
[3] DLG: Deep Leakage from Gradients (NeurIPS 2019),
[4] iDLG: Improved Deep Leakage from Gradients
[5] Melis et. al 2019 S&P Exploiting Unintended Feature Leakage in Collaborative Learning

Taken from **Deep Leakage From Gradients**
https://github.com/mit-han-lab/dlg



1. Inference based on gradients exchanged [1-4]
2. Gradient of a bag of words: non-zero means the data has a word
3. Properties e.g., was someone wearing glasses [5]

Existing Privacy Techniques in FL

Homomorphic Encryption
- Fully homomorphic encryption
- Partial homomorphic encryption (Paillier, Threshold Paillier, etc.)

Pairwise Masking with Secret Sharing
Functional Encryption

Differential Privacy

Not sufficient to prevent recently identified inference threats

source: kaspersky.com

Our Motivation

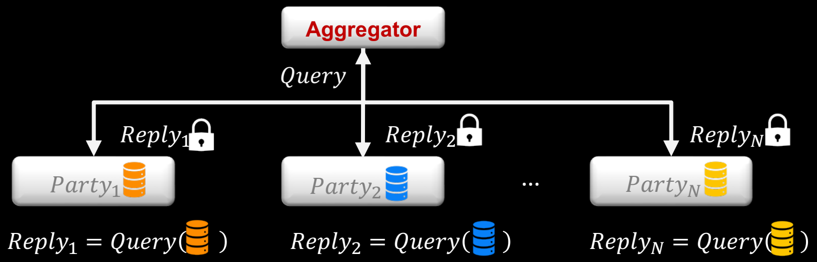
Attacks that compromise the privacy of existing privacy-preserving FL solutions

Secure Aggregation: $f\,(Reply_1^{rount\ t}, Reply_2^{rount\ t-1}\ \ldots\ Reply_{N-1}^{rount\ t-1})$

Replay Attack

Secure Aggregation: $f\,(Reply_1, Reply_2\ \ldots\ Reply_{N-1})$

Isolation Attack



Disaggregation Attack

Multiple Round Secure Aggregation

$$m_t^G = \boldsymbol{m}_t^L \cdot \boldsymbol{p}_t$$

$\boldsymbol{m}_t^L$: the list of local model update at round t

$\boldsymbol{p}_t$: the list of participation status at round t

Given m_t^G and $\boldsymbol{p}_t$, try to figure out m_t^L

Our Motivation

Those attacks stem from the ability of the aggregator to

i) analyze the log of aggregated results or
ii) manipulate the data that is fed to the secure aggregation procedure.

Parties cannot be sure that all received model updates have been aggregated as expected, and are vulnerable to potential disaggregation leading to inference attacks.

DeTrust-FL

- an efficient, scalable, and secure aggregation-based privacy-preserving FL
- decentralized trust design accompanying with decentralized multi-client functional encryption schemes

Preliminaries - Decentralized FE

Functional Encryption

$D_{dk_f}\left(E_{sk_1}(m), \dots, E_{sk_n}(m_n)\right) = f(m_1, \dots, m_n)$, without learning $m_1, \dots, m_n$

Functional Encryption for Inner-Product

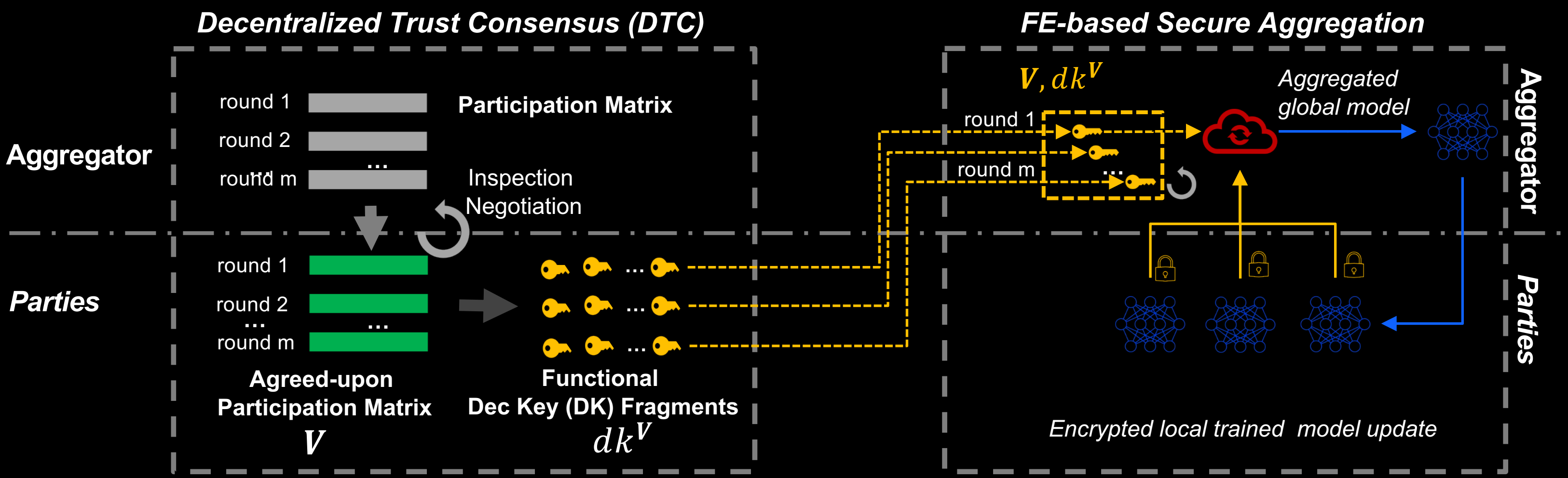
$f(m_1, \dots, m_n) = \sum m_i f_i$, where m_i is from encryption entity, f_i is from decryption entity

Decentralized Multi-Client Functional Encryption for Inner-Product [*,**]

- Setup: $\lambda, n \rightarrow pp$
- KeyGen: $pp \ \rightarrow sk_{p_i}$
- KeyDerivateShare: $pp, sk_{p_i}, f_t \rightarrow dk_{f_t, p_i}$
- KeyDerivateCombine: $pp, \{dk_{f_t, p_i}\} \rightarrow dk_{f_t}$
- Encrypt: $pp, sk_{p_i}, m_{p_i}, l \rightarrow ct_{l, m_{p_i}}$
- Decrypt: $pp, \{ct_{l, m_{p_i}}\}, dk_{f_t} \rightarrow \sum f_t m_{p_i}$

[*] Abdalla, Michel, Fabrice Benhamouda, Markulf Kohlweiss, and Hendrik Waldner. "Decentralizing inner-product functional encryption." In IACR PKC, pp. 128-157. Springer, Cham, 2019.
[**] Chotard, Jérémy, Edouard Dufour-Sans, Romain Gay, Duong Hieu Phan, and David Pointcheval. "Dynamic Decentralized Functional Encryption." IACR Crypto. ePrint Arch. 2020 (2020): 197.

DeTrust-FL Framework Overview



DeTrust-FL Framework Insights

DeTrust-FL: from Agreed-Upon Participation Matrix to Agreed-Upon on Secure Aggregation

Agreed-upon Participation Matrix

Suppose that n parties and m training rounds in FL training

- Consist of "fusion weight"
  - that determines aggregation weight of model update for each FL training round.
- Associate with functional decryption key
  - that determines the aggregator can successfully recovered aggregated model.
  - that is generated by all parties in a collaborative way based on the same agreed-upon participation matrix (DMCFE)

$$
\begin{array}{cc}
 & \begin{array}{ccc} \text{Party } P_1 & & \text{Party } P_n \\ \downarrow & & \downarrow \end{array} \\
\begin{array}{c} \text{Round } R_1 \rightarrow \\ \\ \text{Round } R_m \rightarrow \end{array} &
\begin{bmatrix} w_{p_1,r_1} & \cdots & w_{p_n,r_1} \\ \vdots & w_{p_i,r_j} & \vdots \\ w_{p_1,r_m} & \cdots & w_{p_n,r_m} \end{bmatrix}
\end{array}
\rightarrow \text{fusion weight at round } r_1\text{: } [w_{p_1,r_1}, \dots, w_{p_n,r_1}]
$$

- Each row of $V_{.,j}$ satisfies with agreed-upon on trust threshold
- Each party P_i agrees with its fusion weight $V_{p_i,.}$
- V satisfies with BP criteria to prevent disaggregation attacks [*].

[*]So, Jinhyun, Ramy E. Ali, Basak Guler, Jiantao Jiao, and Salman Avestimehr. "Securing secure aggregation: Mitigating multi-round privacy leakage in federated learning." *arXiv preprint arXiv:2106.03328* (2021).

DeTrust-FL Framework Insights

DeTrust-FL: from Agreed-Upon Participation Matrix to Agreed-Upon on Secure Aggregation

Agreed-upon Secure Aggregation (Decentralized Trust Consensus)

Suppose that n parties and m training rounds in FL training

…

Agreed-upon Participation Matrix $\boldsymbol{V}$

Functional Decryption Key Fragments Matrix $\mathbf{dk}^{V}$

$$\begin{array}{l} \text{Round } R_1 \rightarrow \\ \\ \text{Round } R_m \rightarrow \end{array} \begin{bmatrix} dk_{w_{p_1,r_1}} & \cdots & dk_{w_{p_n,r_1}} \\ \vdots & dk_{w_{p_i,r_j}} & \vdots \\ dk_{w_{p_1,r_m}} & \cdots & dk_{w_{p_n,r_m}} \end{bmatrix} \rightarrow$$

Party P_i generates $dk_{w_{p_i,r_j}}$ for round R_j using corresponding fusion vector $\boldsymbol{V}_{\cdot,\boldsymbol{j}} = [w_{p_1,r_j}, \ldots, w_{p_n,r_j}]$

Aggregator can recover the key dk_{r_j} for round R_j using key fragment vector $[dk_{w_{p_1,r_j}}, \ldots, dk_{w_{p_n,r_j}}]$ that is generated by $P_1, \ldots, P_n$ respectively

$$\rightarrow \begin{bmatrix} dk_{w_{r_1}} \\ \vdots \\ dk_{w_{r_m}} \end{bmatrix}$$

Agreed-Upon Secure Aggregation

DeTrust-FL Framework Insights

DeTrust-FL: contrasting with existing PPFL solutions

Decentralized Trust on Crypto Dealer

- Take advantage of DMCFE; unlike *HybridAlpha* solution, DeTrust-FL does not rely on a centralized crypto dealer (TPA) to generate each round's functional decryption key.

Transparent Secure Aggregation Process

- Secure aggregation process is determined by DK.
- DK is associated to participation matrix.
- DK is generated by all parties in a collaborative way.

Hybrid Methodology Compatibility

- Easily integrate with differential privacy technique as *HybridAlpha* does.

Fusion Algorithm Supports

- Average fusion method
- Weighted fusion method

Security and Privacy Analysis

Security of Cryptographic Infrastructure

- Rely on the security of DMCFE schemes
- Our implementation[*]: DeTrust-FL has ciphertext indistinguishability and is secure against adaptive corruptions under classical DDH assumption.

Privacy of Aggregated Global Model

- Rely on the privacy guarantee of adopted differential privacy mechanism

Privacy of Party's Local Model

- Inference Attack I: isolation attack without collusion
- Inference Attack II: isolation attack with colluding parties
- Inference Attack III: disaggregation attack
- Inference Attack IV: replay attack

[*] Abdalla, Michel, Fabrice Benhamouda, Markulf Kohlweiss, and Hendrik Waldner. "Decentralizing inner-product functional encryption." In IACR PKC, pp. 128-157. Springer, Cham, 2019.

Experimental Evaluation

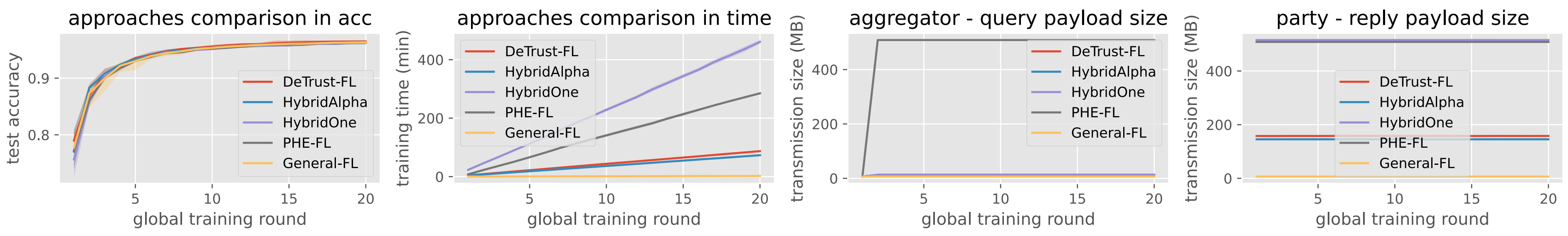
Experimental Setup

- Implementation
  - IBM Federated Learning (https://ibmfl.mybluemix.net/)
  - Community Edition (https://github.com/IBM/federated-learning-lib)
  - gmpy2 python library
- Environment
  - Intel(R) Xeon(R) CPU E5-2683 v4 platform with 32 cores and 64GB of RAM
  - One machine – multiple processes to simulate distributed environment
  - Network latency is not measured in our experiment
- Dataset: MNIST and CIFAR10
- Baselines
  - General-FL: general FL training without any secure aggregation setting
  - PHE-FL : FL training using partially additive homomorphic encryption (i.e., Paillier) based secure aggregation (e.g., [5], [21]);
  - HybridOne: FL training using threshold Paillier based secure aggregation ([7]);
  - HybridAlpha: FL training using functional encryption based secure aggregation (8])
  - DeTrust-FL: this work
- FL Setting

Type	Model Architecture	Parameters	Parties	Training/Test per Party	FL Rounds	Local Epochs
CNN-MNIST	2xConv2D→MaxPooling→Droupout→Flatten→Dense→Dense	1,199,882	5	500/2000 (non-iid)	20	3
CNN-CIFAR10	2x(2xConv2D→MaxPooling→Droupout)→Flatten→Dense→Dense	890,410	10	5000/1000 (non-iid)	30	20

Experimental Evaluation

Model accuracy, training time and transmission payload comparison in FL training on evaluating MNIST dataset



Compared to **PHE-FL** and **HybridOne** solutions, **DeTrust-FL** reduces the volume of transmission payload by 73.6% and 82.2%, respectively.

Communication Interaction Comparison

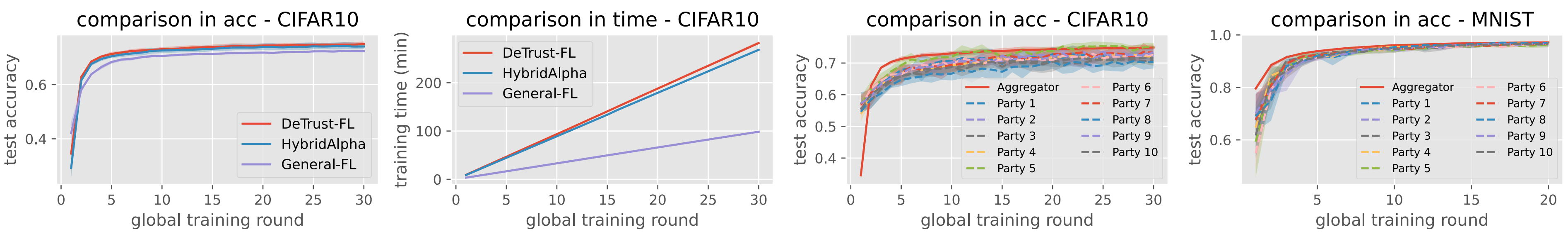
Proposal	$\mathcal{A} \leftrightarrow \mathcal{P}$	$\mathcal{A} \leftrightarrow \mathcal{K}$	$\mathcal{P} \leftrightarrow \mathcal{K}$	Total
General-FL	NM+M	0	0	NM + M
PHE-FL	2NM+M	1	M×1	2NM+2M+1
HybridOne	2NM+M	1	M×1	2NM+2M+1
HybridAlpha	NM+M	N+1	M×1	NM+N+2M+1
DeTrust-FL (our work)	NM+M	1	M×1	NM+2M+1

N rounds of global training with one aggregator $\mathcal{A}$, M parties $\mathcal{P}$ and one key server (or TPA) $\mathcal{K}$.

DeTrust-FL also generates a transmission payload similar to **HybridAlpha**, however, we reduce the number of interactions by 16.4% in the setting of 20 global training rounds with 5 parties

Experimental Evaluation

Performance comparison on evaluating CIFAR10 dataset

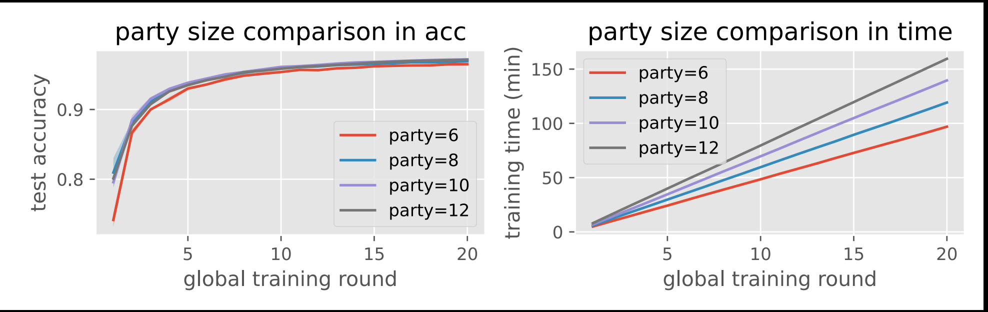


Both **DeTrust-FL** and **HybridAlpha** can achieve a well-performed and even better model accuracy comparing to the General-FL on CIFAR10

Theoretically, we believe that the slight improvement in accuracy is due to the encoding operation, which discards some information and could be considered as a type of pruning.

Experimental Evaluation

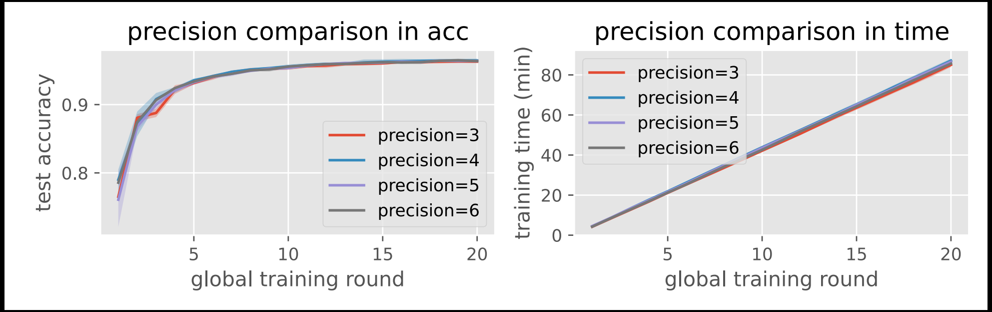
Impact of number of parties



Impact of number of parties in DeTrust-FL training on evaluating MNIST dataset with setting of
precision=4 and 3 local training epochs per training round

more parties
higher model accuracy
more training time

Impact of encoding precision



Impact of encoding precision on floating-point parameters in DeTrust-FL training on evaluating MNIST dataset with setting of 5 parties and 3 local training epochs per training round.

no significant impact on model accuracy

slightly increase the training time

Conclusion

DeTrust-FL approach for privacy-preserving FL training in decentralized trust setting

- Prevent recently identified inference attacks (isolation attack, replay attack, and disaggregation attack.
- Support transparent and privacy-preserving secure aggregation process.
- Support various fusion methods and has hybrid methodology compatibility.

THANK YOU
Q&A