IBM Research

HybridAlpha: An Efficient Approach for Privacy-Preserving Federated Learning

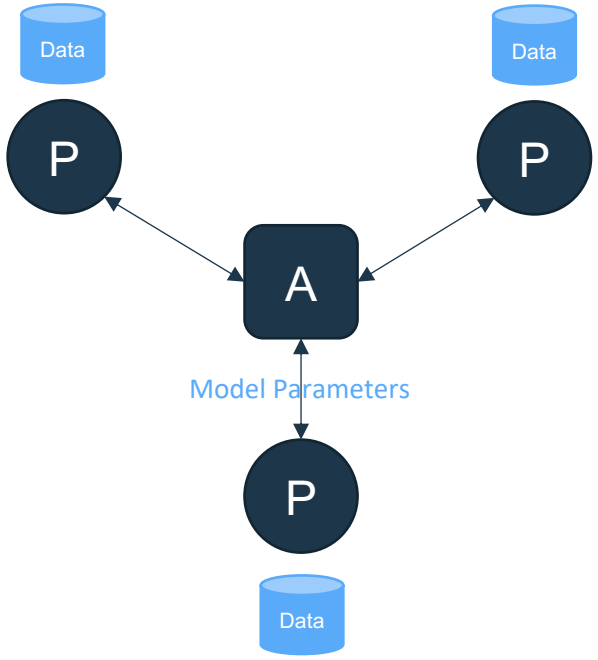
Runhua Xu, Nathalie Baracaldo, **Yi Zhou**, Ali Anwar, Heiko Ludwig

IBM

Federated Learning - Overview

How it works:

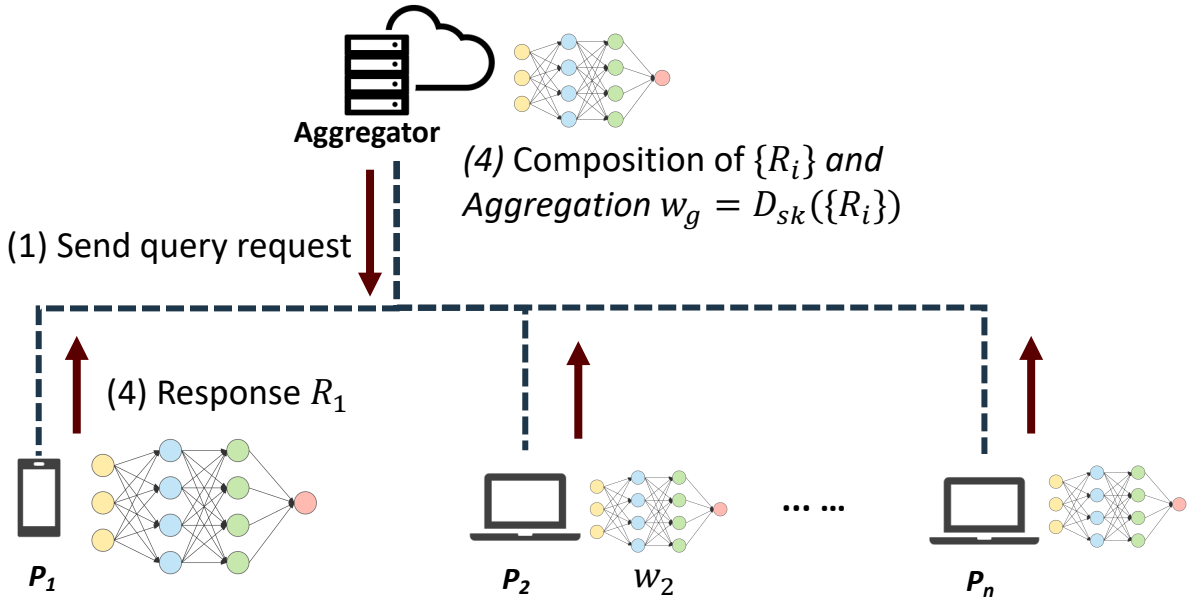
- Parties (P) collaboratively train a ML model, keeping training data to themselves
- Models are trained locally, within each party
- Local models' parameters from parties are merged and distributed to parties, at the end or after each epoch
- Different topologies used in different trust models, often using an Aggregator (A)



Privacy Issues of Federated Learning

- privacy leakage of model output
- privacy disclosure of aggregation computation

Hybrid Approach to Federated Learning



(2) Train locally and generate model w_1

(3) $\mathrm{R}_1 = E_{pk_1}\left(w_1 + \frac{noise}{N}\right)$

Overview of existing FL framework

Privacy Issues of Federated Learning

- privacy leakage of model output
- privacy disclosure of aggregation computation

Current approaches

- combines secure multi-party computation (SMC) and differential privacy (DP) through reduced noise

Limitation of current approaches

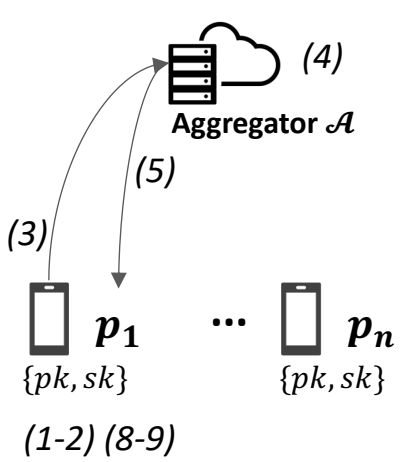
- very slow due to encryption algorithm used
- require multiple rounds of communication
- do not support dynamic participation: dropouts and new additions are not allowed without full system re-keying
- cannot prevent curious aggregators from getting partial decrypted data

Existing approaches V.S. Our approach

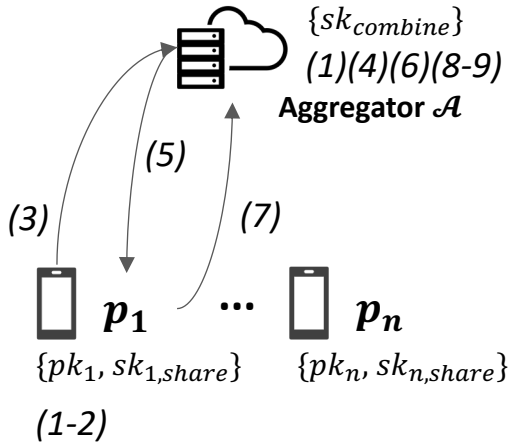
Comparison of privacy-preserving approaches in federated learning framework

	Threat Model		Privacy Guarantee		SMC	Features	
Proposed Approach	participant	aggregator	computation	output	type *	communication †	dynamic participants
Shokri and Shmatikov[36]	honest	honest	✗	✓	–	1 round	✓
PATE [31]	honest	honest	✗	✓	–	1 round	–
PySyft [34]	honest	HbC$^{\diamond}$	✓	✓	HE	2 rounds‡	–
Bonawitz et al. [6]	dishonest	HbC$^{\diamond}$	✓	✓	SS+AE	3 rounds‡	dropout
Truex et al. [38]	dishonest	HbC$^{\diamond}$	✓	✓	TP	3 rounds‡	✗
HybridAlpha (our work)	dishonest	HbC$^{\diamond}$	✓	✓	FE	1 round‡	dropout + addition

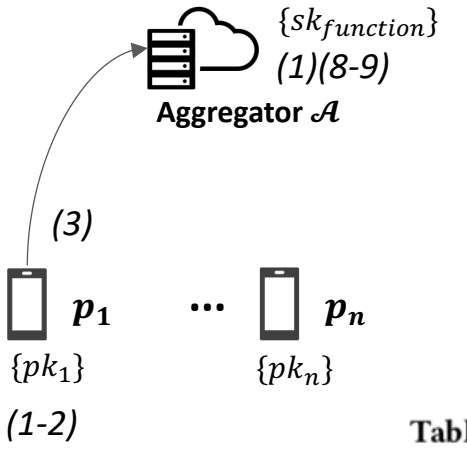
Comparison of SMC-based Secure Aggregation



P – SMC



TP – SMC



Our Approach

Steps of SMC aggregation

(1) key setup & local training
(2) encrypt model parameters c_{init}
(3) send c_{init} to $\mathcal{A}$
(4) combine cipher $c_{cmb} \leftarrow \{c_{init}\}$
(5) send back c_{cmb}
(6) partial decrypt $c_{part} \leftarrow c_{cmb}$
(7) send c_{part} to $\mathcal{A}$
(8) (combine) decryption
(9) update global model

Issues

- **Current SMC protocols are not efficient enough**
  - crypto efficiency (time)
  - communication steps
- **Lack of support for dynamic participants**

Table 2: The number of crypto-related operations required for each solution.

Communication	TP-SMC	P-SMC	HybridAlpha *
Step (1)	n	n	$n + m$
Step (3)	$n \times m$	$n \times m$	$n \times m$
Step (5)	$m \times t$	$n \times m$	-
Step (7)	$t \times m$	-	-
TOTAL	$2mt + mn + n$	$2mn + n$	$mn + m + n$

Functional Encryption in a Nutshell

- In functional encryption for inner-product, a third-party authority that
  - generates the public key pk for encryptor to encrypt vector $\boldsymbol{x} = [\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}]$
  - generates functional private key $sk_{f,y}$ that is corresponding to a vector $\boldsymbol{y}$ for the decryptor

- From now on, let us assume, there is **a trusted third party other than aggregator** to be involved in the federated learning

Functional Encryption for Inner-product [*]

$$f(\boldsymbol{x}, \boldsymbol{y}) = \sum x_i y_i$$

$$D_{sk}\left(f\left(E_{pk}(x_1, \ldots, x_n)\right), \boldsymbol{y}\right) = \sum x_i y_i$$

without learning $x_1, \ldots, x_n$

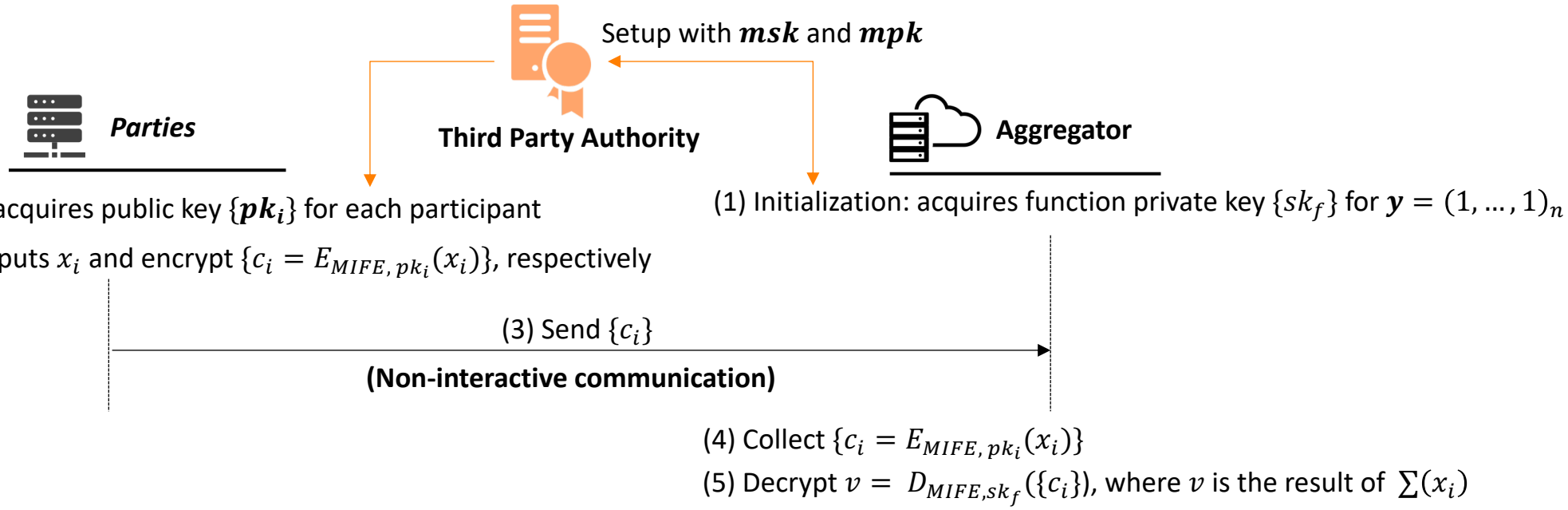
$x_1, \ldots, x_n$ can be from one single source or multiple sources

[*] Abdalla, Michel, Florian Bourse, Angelo De Caro, and David Pointcheval. "Simple functional encryption schemes for inner products." In *IACR PKC*, pp. 733-751. Springer, Berlin, Heidelberg, 2015.

Non-interactive Secure Computation

secure multi-party aggregation computation

- **Parties** $P = (p_1, \dots, p_i, \dots, p_n)$, each party p_i has input x_i
- **Aggregator** acquire $\sum(x_i)$ without learning specific x_i of p_i



Constructed from **Multiple-input Functional Encryption (MIFE)**[*]

[*] Abdalla, Michel, Dario Catalano, Dario Fiore, Romain Gay, and Bogdan Ursu. "Multi-input functional encryption for inner products: function-hiding realizations and constructions without pairings." In CRYPTO 2018.

Threat Model

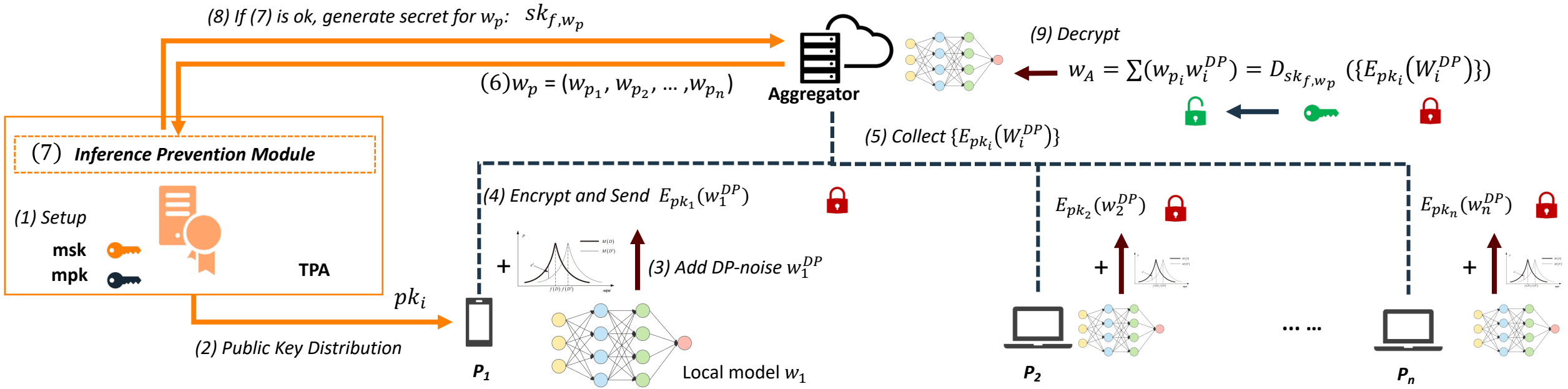
- A trusted third party (TPA) that distributes keys
- An honest but curious aggregator, and the aggregator may collude with dishonest parties
- Parties may try to infer data from other participants through the final model or during the federated learning process

Third Party Authority

Aggregator

Parties

Overview of HybridAlpha



Hybrid Approach

- differential privacy + noise reduction through SMC
- privacy guarantee: model output / aggregation

Efficiency Improvement

- Efficient encryption/decryption algorithm
- Non-interactive secure computation

Support Dynamic Participants

- Randomly drop out / join in

Inference Prevention Module

- *Threshold t* helps detect and stop attacks from curious aggregators and colluding participants, *t* defines a threshold on the number of non-colluding participants

- For example, if $t = 3$, the module filters the following suspicious weight vector w_p*:*
  - infers one party's model update:
    - *<0,0,0,1>*
    - *<0.0009,0.009,0,1>*
    - *<1>*
  - *exclude honest parties' model update:*
    - *<1,1,0,0>*

- *t* has an impact on the number of dropouts allowed by the system
  - Mainly, it helps set up the minimum quorum of participants replying to the system

Algorithm 2: Inference prevention filter

Input: $\mathbf{w}_p$:=A weighted vector to be inspected for inference attacks; t:= threshold of minimum number of dropouts and expected number of non-colluding participants

1 **function** *inference-prevention-filter*($\mathbf{w}_p, t$)
2 $c_{nz} \leftarrow$ count the non-zero element in $\mathbf{w}_p$;
3 **if** $c_{nz} < t$ **then return** "invalid w_p" ;
4 **foreach** *non-zero* $w_{p_i} \in \mathbf{w}_p$ **do**
5 **if** $w_{p_i} \neq \frac{1}{c_{nz}}$ **then return** "invalid w_p";
6 **end**
7 forward $\mathbf{w}_p$ to the TPA;

Experimental Results

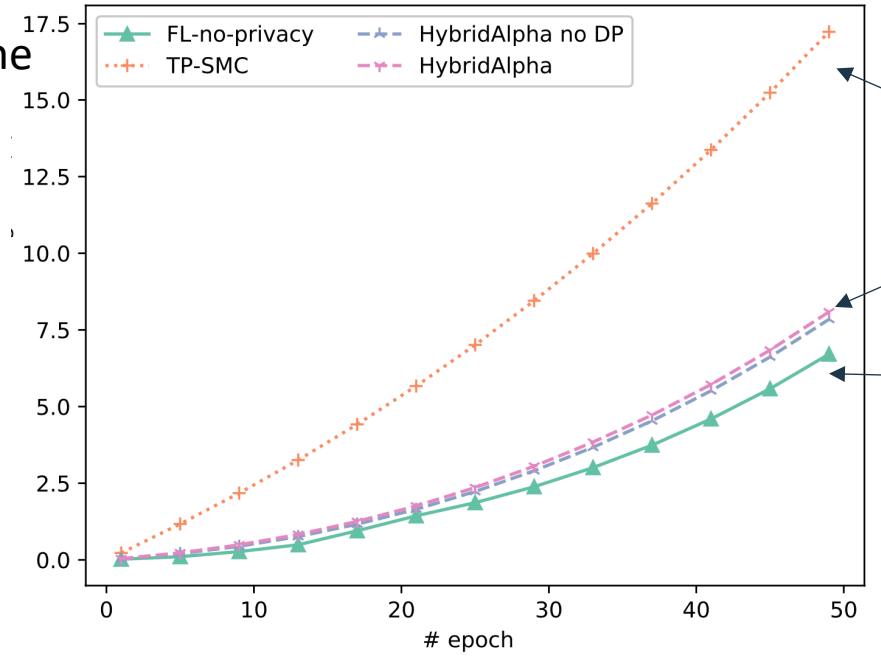
Cryptosystems Implementation
- Python + GMP/Charm-crypto library

Experimental Environment
- 44 core Intel Xeon E5-2699 v4 platform with 384 GB of RAM
- CNN on MNIST dataset

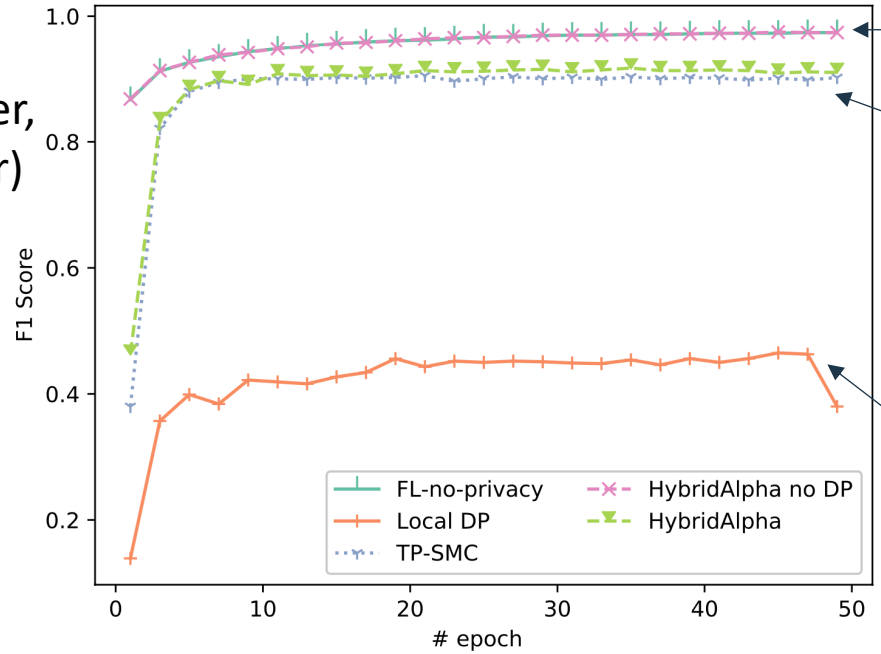
Baselines
- FL without DP/ FL local DP
- TP-SMC FL (DP)
- P-SMC FL (DP)

On average reduces
the training time by **68%**
the data transfer volume by **92%**
While providing
the **same model performance**
the **same privacy guarantees** as the existing solutions

Training time (hours)



F1 Score (the higher, the better)



Thank you! Questions?

Find our AI Security and Privacy Solutions team at:
https://resedit.watson.ibm.com/researcher/view_group.php?id=10276

References

[1] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. ACM, ACM, 1175–1191.

[2] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable private learning with PATE, In Proceedings of the 2018 Sixth International Conference on Learning Representations. arXiv preprint arXiv:1802.08908.

[3] Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. 2018. A generic framework for privacy preserving deep learning. In Proceedings of Privacy Preserving Machine Learning Workshop with NeurIPS 2018.

[4] Reza Shokri and Vitaly Shmatikov. 2015. Privacy-preserving deep learning. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. ACM, ACM, 1310–1321.

[5] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, and Rui Zhang. 2018. A Hybrid Approach to Privacy-Preserving Federated Learning. arXiv preprint arXiv:1812.03224 (2018).

[6] Michel Abdalla, Dario Catalano, Dario Fiore, Romain Gay, and Bogdan Ursu. 2018. Multi-input functional encryption for inner products: function-hiding realizations and constructions without pairings. In Annual International Cryptology Conference. Springer, Springer, 597–627.