Secure Byzantine-Robust Machine Learning

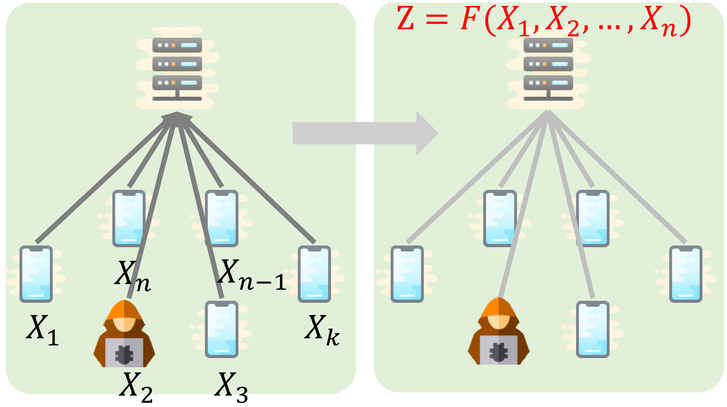
Lie He, Sai Praneeth Karimireddy, Martin Jaggi
MLO, EPFL
{lie.he, sai.karimireddy, martin.jaggi}@epfl.ch

NEURAL INFORMATION PROCESSING SYSTEMS

Introduction

- **Setup:** Distributed training with *honest-but-curious* server(s) and Byzantine workers.
- **Objective:** Efficient combination of secure and Byzantine-robust aggregation.

Background: Byzantine-Robust Aggregation

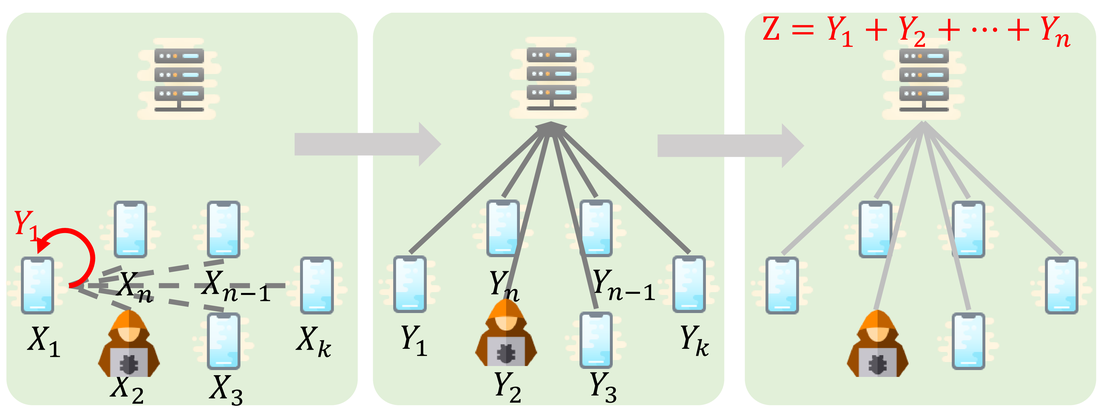


A common robust aggregation routine [1, 2].

1. Workers send updates to the server in **plaintext**.
2. The server aggregates the updates using a **robust aggregation rule** F.

However, the server can observe the updates and therefore inspect worker inputs.

Background: Secure Aggregation



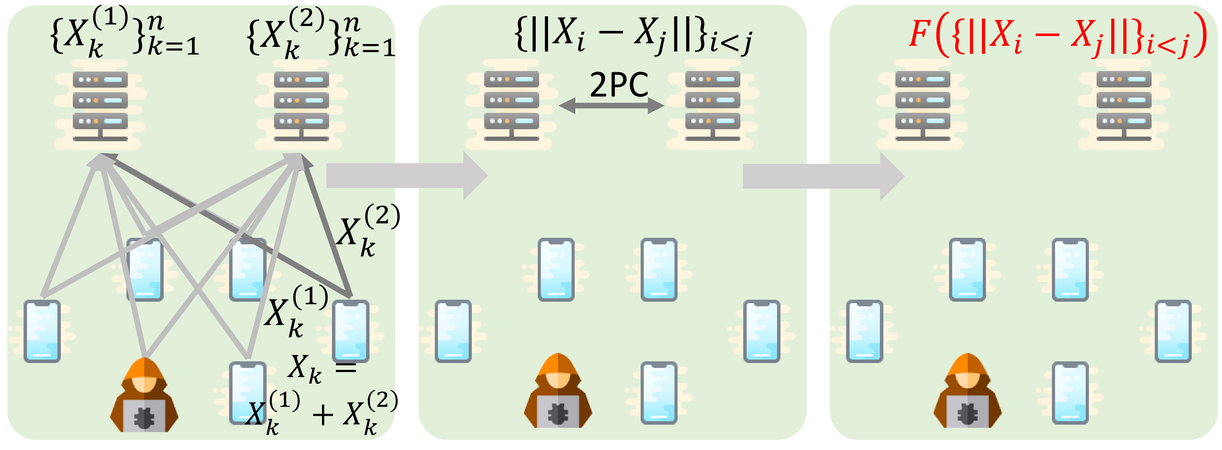
A secure aggregation, e.g. [3], protects a worker's **input privacy**.

1. Workers add to their updates pairwise masks among workers and send them to the server.
2. The server **sum up** masked values such that pairwise masks cancel and thus real the true sum of worker updates.

Note that

- the aggregation function can only be sum which is not robust.
- we ignore the "the second mask" and "fault-tolerance" in the diagram.

Robust and Secure Aggregation: Two-server Model



We propose to use two non-colluding servers to combine secure and robust aggregation:

- The workers secret-share their updates locally and then send them to the servers;
- The servers use two-party computation (2PC) protocols to compute pairwise distances;
- A server use the pairwise distances to compute the robust aggregation rules.

Discussion

Pros. The advantages of the this approach:

- ✓ Low communication and compuation overhead;
- ✓ Easy to achieve fault-tolerance;
- ✓ Support multiple distance-based aggregation rules.
- ✓ Compatible with differential privacy.

Cons. A major disadvantage of this approach is that the pairwise distances are leaked to the servers. However, such distances leak very little information because of the high-dimensional leeway. A future work is to remove this leakage and support more robust aggregation rules.

Notes. The *non-colluding* assumption is guaranteed by regulation such that the service providers have no incentive to collude.

References

[1] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *NeurIPS - Advances in Neural Information Processing Systems 30*, pages 119–129, 2017.

[2] Yudong Chen, Lili Su, and Jiaming Xu. Distributed statistical machine learning in adversarial settings. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(2):1–25, Dec 2017. ISSN 2476-1249. doi: 10.1145/3154503. URL `http://dx.doi.org/10.1145/3154503`.

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