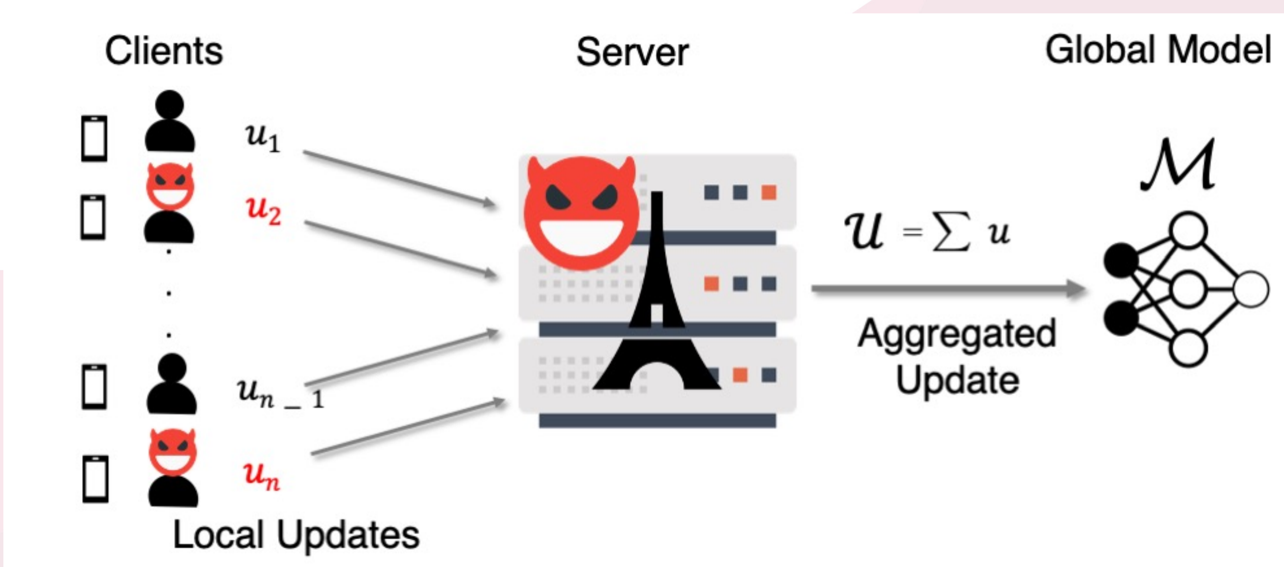
CIFellows 2020-2021

Computing Innovation Fellows

E𝘼FFeL: Ensuring Integrity for Federated Learning

Amrita Roy Chowdhury, University of California, San Diego

1. Problem Setting



Federated Learning (FL) is a **decentralized** learning paradigm with **multiple clients** coordinated by a **single server**.

Each client's raw data is stored locally.
Server wants to train a global model $\mathcal{M}$ on the joint dataset.

For each round of training:

- Server broadcasts the current parameters of $\mathcal{M}$
- Each client computes a local update (gradient), u
- Server collects and aggregates client updates, $\mathcal{U} = \Sigma u$
- Server updates $\mathcal{M}$ based on $\mathcal{U}$

Threat Model

- **Input Privacy**
  - Client data is sensitive
  - Untrusted server
- **Input Integrity**
  - FL is vulnerable to data poisoning
  - Malicious clients submit malformed updates to tamper with $\mathcal{M}$'s accuracy

Goals

- **Ensure input privacy** for clients
- **Ensure input integrity** to protect against data poisoning

2. Secure Aggregation with Verified Inputs

- Public validation predicate $Valid(\cdot)$
- Input u is valid, i.e., passes the integrity check if $Valid(u) = 1$
- E.g. $Valid(u) = \mathbb{I}[||u||_2 < \rho]$

A Secure Aggregation with Verified Inputs (SAVI) protocol

Input Integrity:

- securely verifies the integrity of each input
- aggregates **well-formed inputs only**, i.e., $Valid(u) = 1$

Input Privacy:

- releases only the final aggregate in the clear

3. EIFFeL Overview

EIFFEL instantiates a SAVI protocol for an **arbitrary** public $Valid(\cdot)$ expressed as an arithmetic circuit.

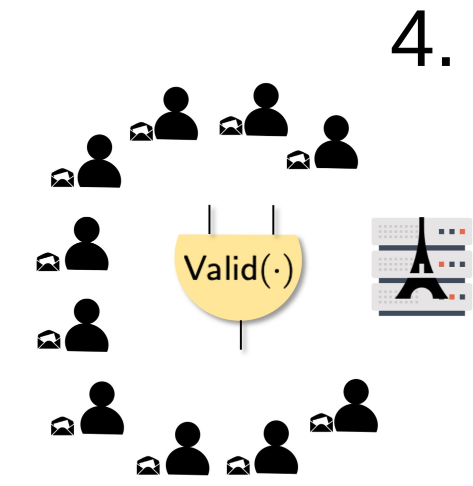
Cryptographic Tools

- **Input Privacy**
  - Shamir's Threshold Secret Sharing Scheme
- **Input Integrity**
  - Secret-Shared Non-Interactive Proof (SNIP)
  - Verifiable Secret Shares

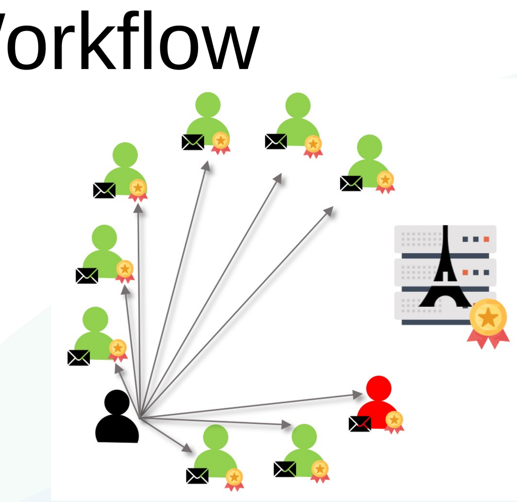
Key Ideas

- **Single Server**
  - SNIP requires multiple honest servers to acts as the verifiers
  - In EIFFeL, clients act as the verifiers for each other supervised by a single server
- **Malicious Model**
  - EIFFeL extends SNIP to the malicious model
  - Threshold secret sharing creates multiple instantiations of the SNIP protocol
  - Server uses this redundancy for robust verification

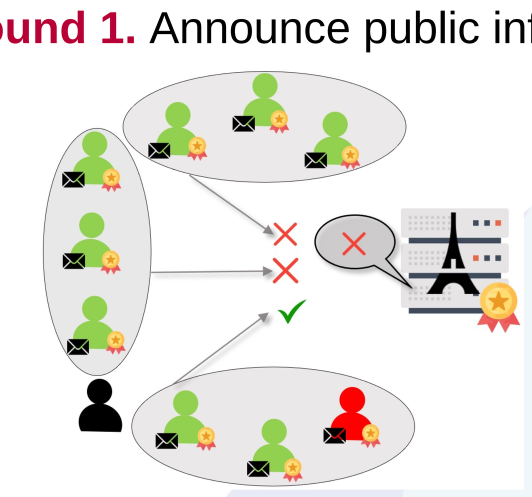
4. EIFFeL Workflow



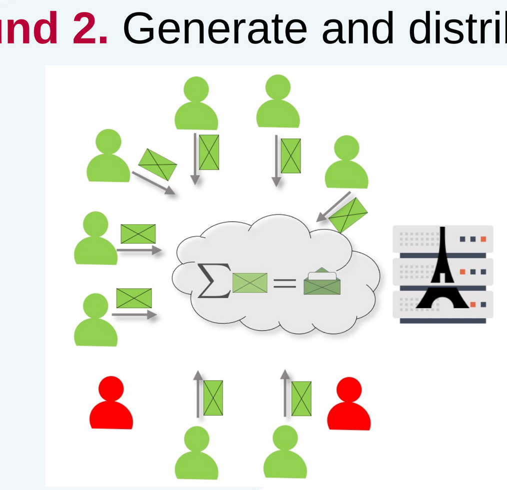
Round 1. Announce public information



Round 2. Generate and distribute proof



Round 3. Verify proof



Round 4. Compute final aggregate

5. Evaluation Highlights

- With **100** clients and **10%** poisoning, EIFFeL trains a model on MNIST to the same accuracy as that of a non-poisoned one in **2.4s**/iteration per client
- Communication cost for the client is **9.5MB**

CRA Computing Research Association
CCC Computing Community Consortium Catalyst
NSF