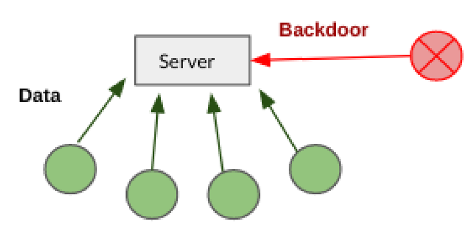
SCOTCH: An Efficient Secure Computation Framework for Secure Aggregation

Yash More Prashanthi Ramachandran Priyam Panda Arup Mondal Harpreet Virk Debayan Gupta

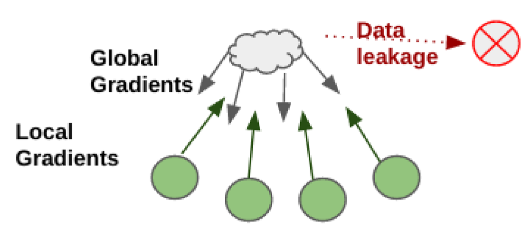
Ashoka University

Motivation



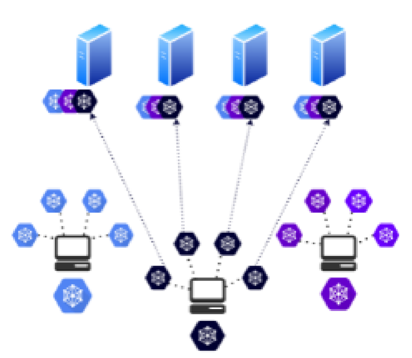
Centralized Aggregation :

- Exposes a Single Point of Failure (server)
- Can compromise user-privacy
- Vulnerable to malicious parties (collusion)



Traditional Federated Learning :

- Allows collaborative training of models
- Vulnerable to Data-leakage
- Prone to inference attacks



Secure Federated Learning using MPC:

- Secret Sharing and Secure Outsourced Aggregation provides end-to-end protection
- Low cryptographic overheads
- M-client N-server configuration prevents single point failures

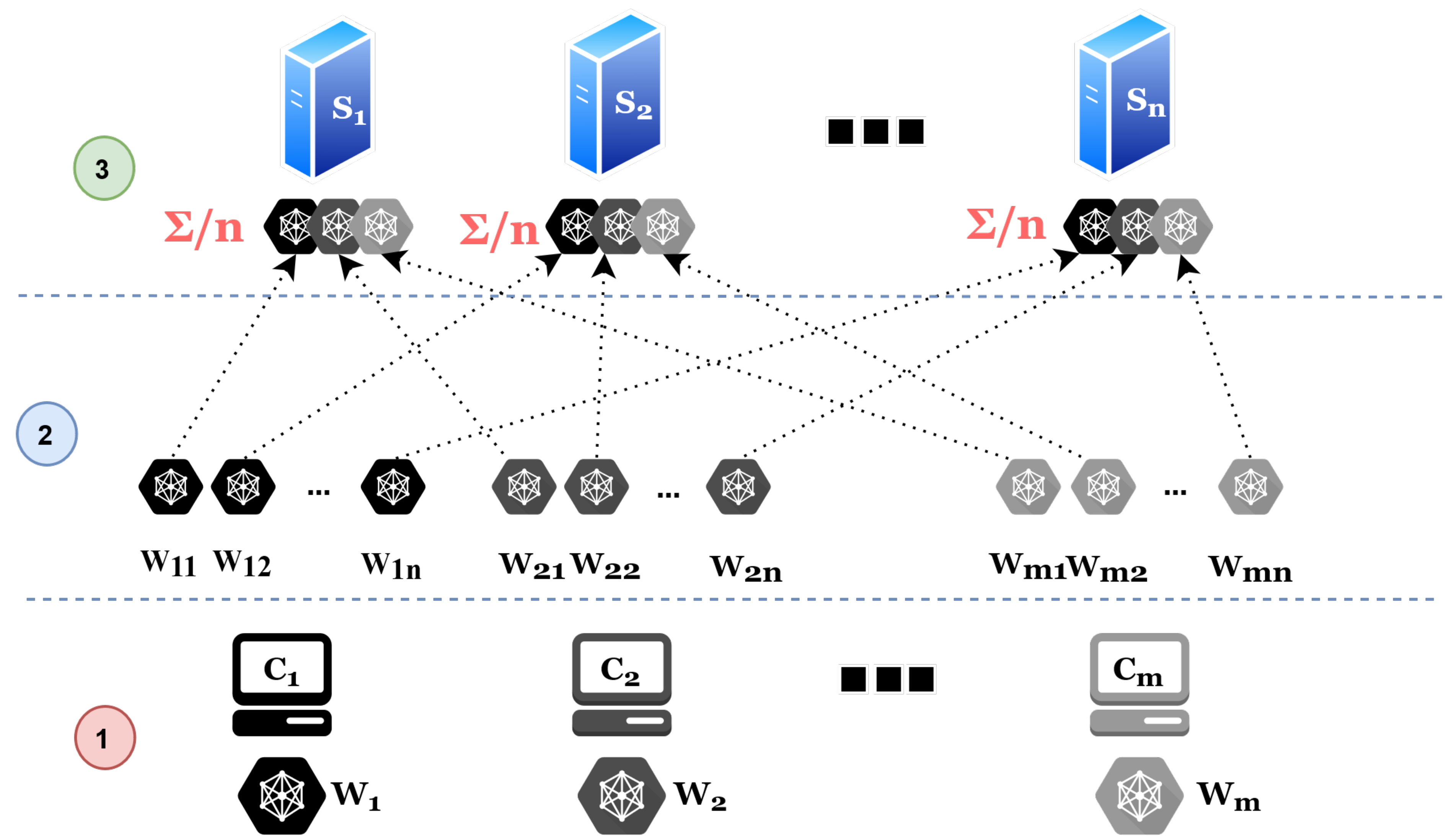
Introduction

We propose SCOTCH, a fast and efficient federated learning framework that allows for decentralized gradient aggregation using *Secure Outsourced Computation* and *Secret Sharing*.

Threat Model

- SCOTCH ensures that neither colluding participants nor aggregators can learn anything about the private inputs or outputs of the honest participants.
- Any encryption broadcast to the network is re-randomized to avoid data-leakage.
- It uses *n-out-of-n* secret sharing for secure aggregation of gradients.
- It assumes a passive adversary that may try to glean information from shared inputs.

An overview of the SCOTCH protocol



1. Each client $C_{1...}C_m$ trains its local model M_i on a random subset of its private dataset D_i (using the same model architecture).
2. Each client C_i creates n additive secret shares of its model M_i and sends each share $[M_i]_j$ to the server S_j, where $j \in \{1...n\}$.
3. After receiving the shares from all clients, each server S_j averages them to obtain σ_j and sends σ_j to all the clients. Then, each client locally computes $M_{agg} = \sum \sigma_j$ for all $j \in \{1...n\}$.

Communication Complexity

SCOTCH offers minimal computational overhead when it comes to cryptographic operations.

Complexity	Data Owners	Aggregator Servers
Computation	$O(2mn)$	$O(mn)$
Communication	$O(n)$	$O(m)$
Storage	$O(m)$	$O(n)$

Experimental Results

SCOTCH is evaluated in terms of three indicators: (a) Performance with regard to different numbers of clients and servers (b) Impact of varying precision (c) Communication Complexity.

Precision	4-bit	8-bit	16-bit	32-bit
Centralized FL	0.09	0.41	0.71	0.8

Figure 1. Performance (accuracy) of Centralized FL on the MNIST under multiple precision settings

Clients	MNIST	EMNIST	FMNIST
2	0.975	0.985	0.85
3	0.965	0.984	0.69
4	0.74	0.9	0.53

Figure 2. SCOTCH performance accuracy on MNIST, EMNIST, FMNIST with different number of clients

Clients	2	3	4	5
MNIST-16	0.3	0.19	0.113	0.11
MNIST-32	0.975	0.965	0.74	0.53

Figure 3. SCOTCH performance accuracy on MNIST under 16-bit and 32-bit precision settings

Impact of Precision Length

- We use a mapping between fixed-point decimals and the integer ring (as used by state-of-the-art MPC frameworks such as SecureML).
- We replicate the precision settings used in SecureML to conduct our experiments for comparison across different models.
- We test various precision lengths while performing multi-class classification on MNIST. Our results suggest that increasing the precision length leads to better model performance.

Future Work

We plan to extend the SCOTCH framework to (a) provide security against malicious actors (both servers and clients) (b) handle higher volume of clients and servers (c) deploy it via open-source channels for academic and industrial use-cases.

Author Contact: yash.more@alumni.ashoka.edu.in, pramach3@brown.edu.in, priyam.panda@alumni.ashoka.edu.in, arup.mondal_phd19@ashoka.edu.in, harpreet.virk@alumni.ashoka.edu.in, debayan.gupta@ashoka.edu.in, Ashoka University, Rajiv Gandhi Education City, Sonipat, Haryana 131029