Security and Robustness of Collaborative Learning Systems

Anwar Hithnawi

ETH zürich

Security and Robustness of Collaborative Learning Systems

New Challenges

of the collaborative learning paradigm

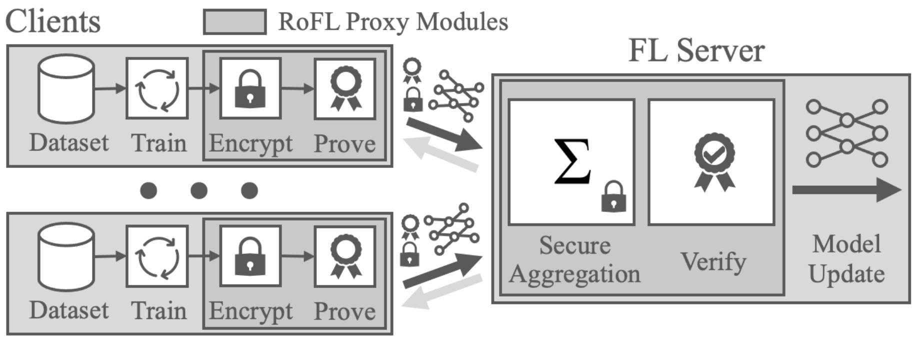
Robustness

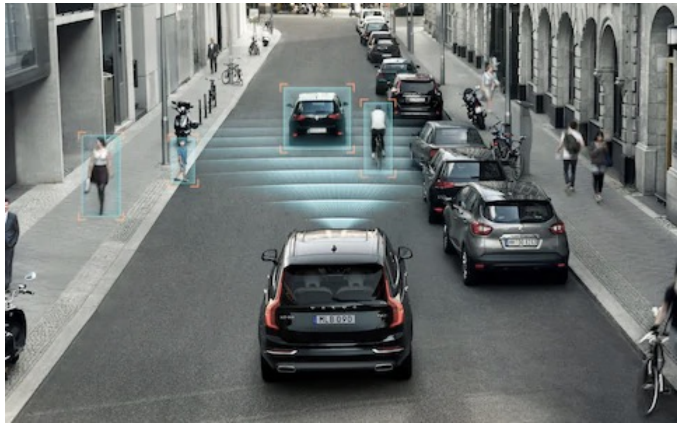
Security

Privacy

RoFL: Robustness of Secure FL

H. Lycklama, L. Burkhalter, A. Viand, N. Küchler, A. Hithnawi [IEEE SP'23]





Autonomous Driving

Object Classification

Data Driven World

Health Care

AlphaGo

large, diverse data $\Rightarrow$ broad generalization

Solving tasks where data is **accessible**…

… however, many important tasks we care about ...

Tasks

WikiText-103

MNIST

ImageNet

WMT

CIFAR

GPT-3

Public Data

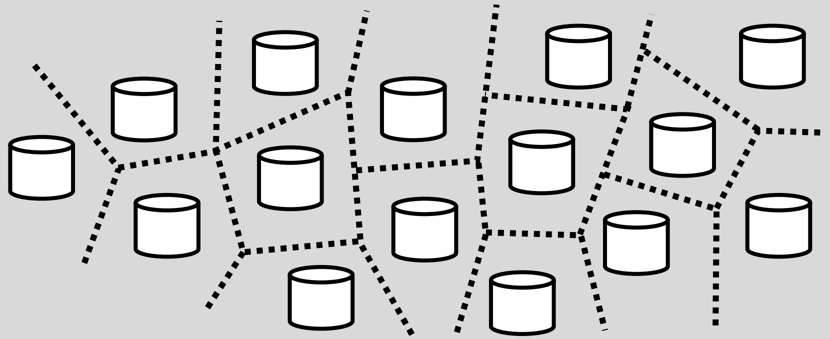
Crowdsourced Data

For example: web, books, articles, science, TV, corpus, audiobooks, …

Inaccessible

Health – Cancer, Alzheimer, Dementia, Depression
Finance – Economic growth, Market predictions
Government – Education, Taxes, Immigration, Income
Personal Data – Text Messages, Emails, Photos

→ EU Data Governance Act (**DGA**) effective from 2023
facilitate the reuse of protected public-sector data

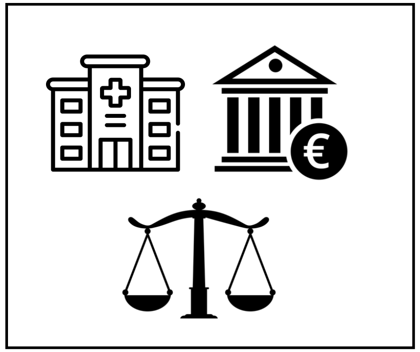


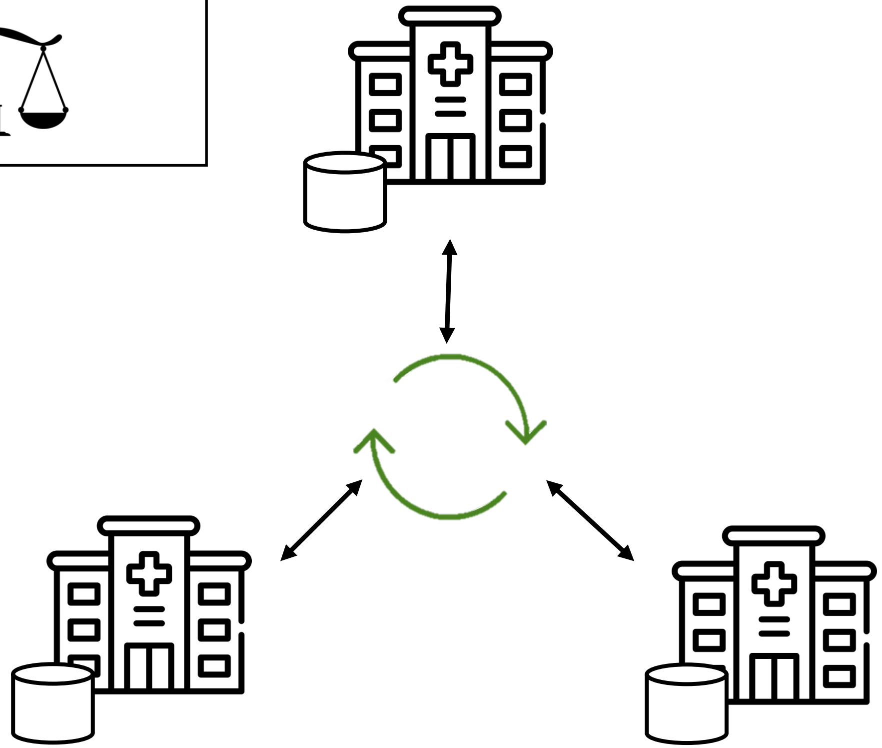
Legal Frameworks and Technologies to facilitate privacy preserving access

Data Silos
- Privacy Laws
- Competition

Collaborative Learning

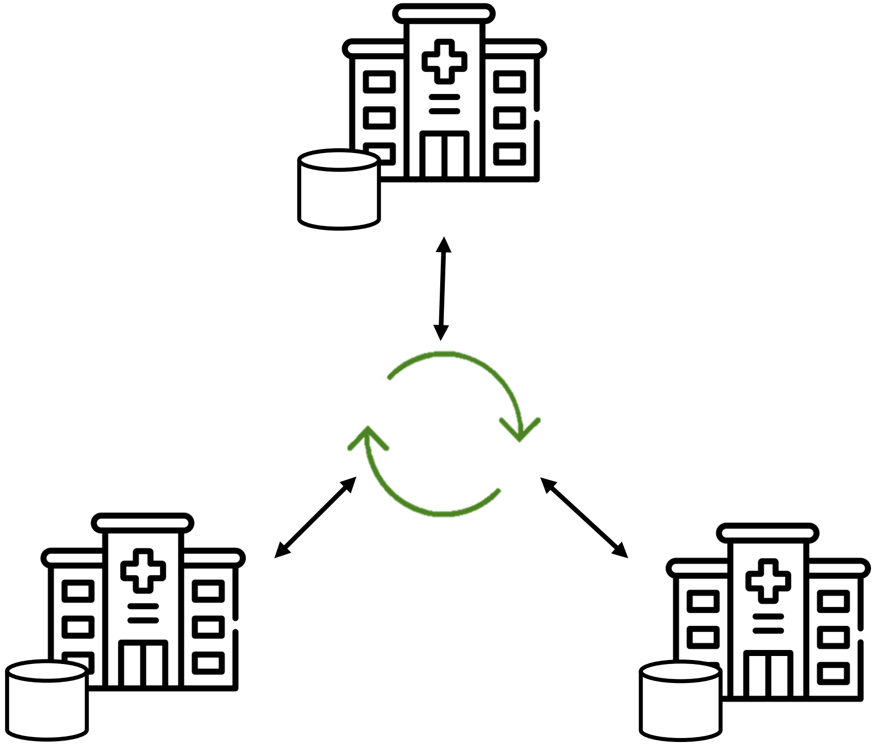
Collaborative Learning



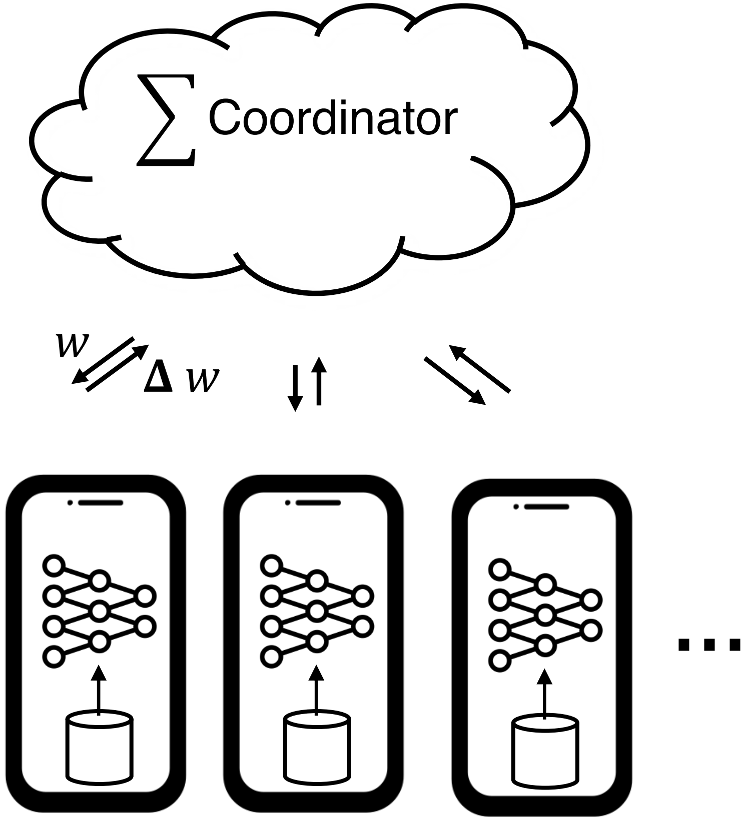


Decentralized Learning

Collaborative Learning



Decentralized Learning



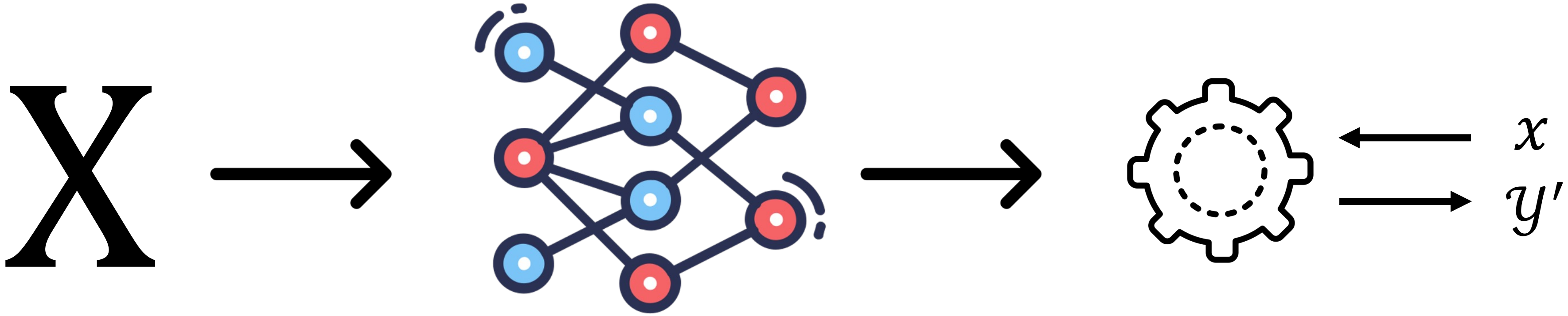
Federated Learning

Adversarial Machine Learning

Data

Train

Deploy/Serv
e

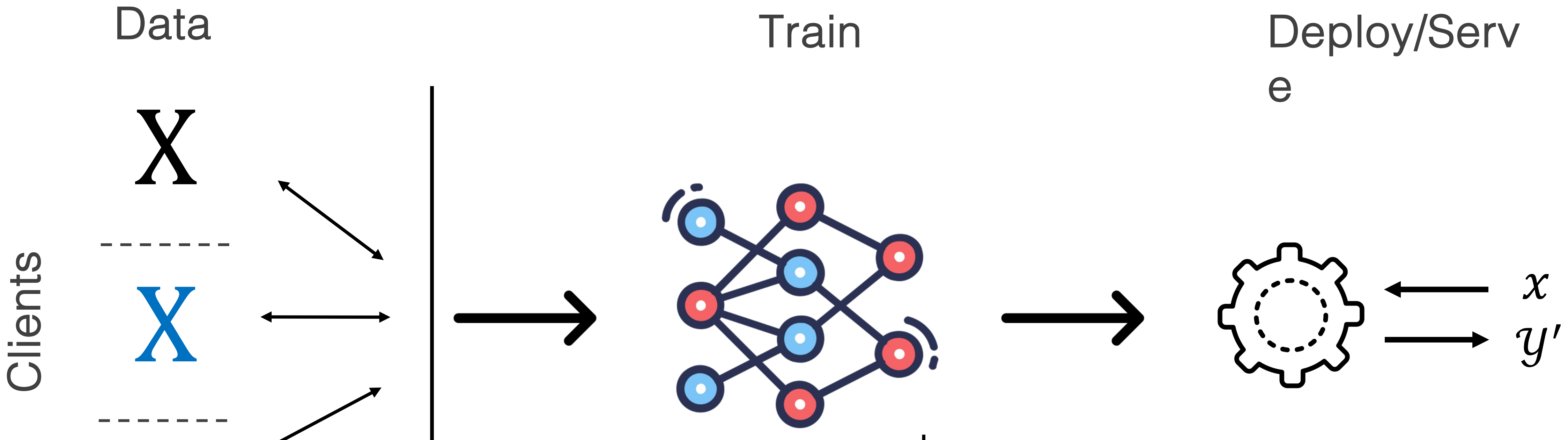


- Adversarial Examples (perturbations)
- Model extraction/inversion attacks
- Poisoning attacks (backdoors)

Deploy

Train

Security and Privacy of Collaborative ML



- adversarial examples (perturbations)
- model extraction/inversion attacks

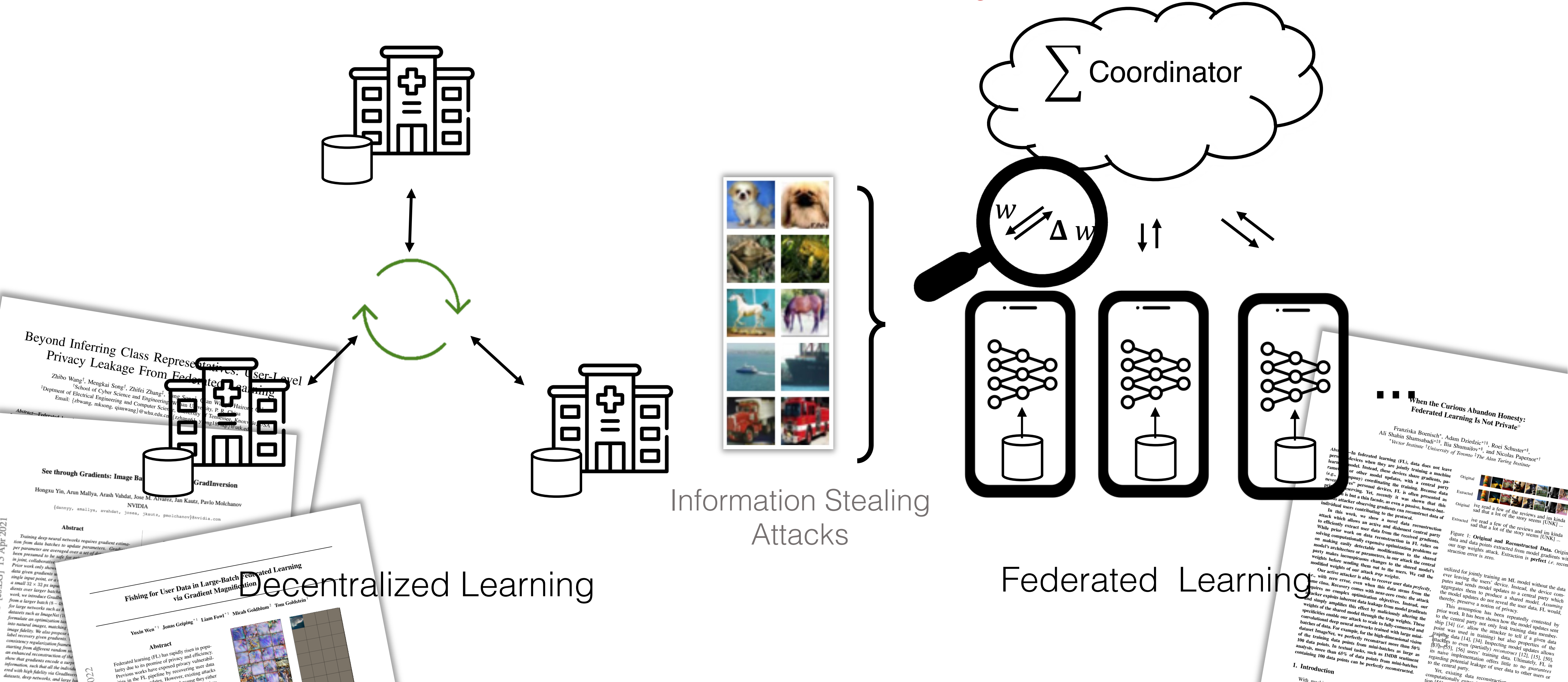
Test

- poisoning attacks (backdoors)
- data stealing attacks

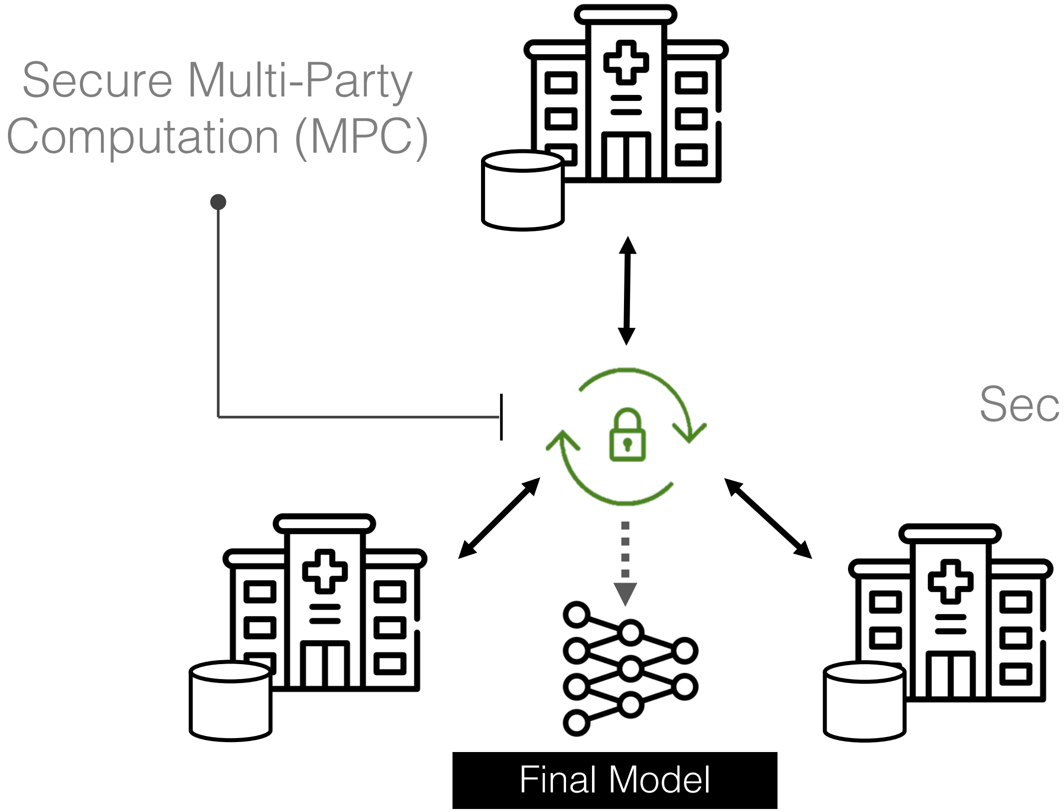
Train

Confidentiality of Input Data

Federation ≠ Privacy

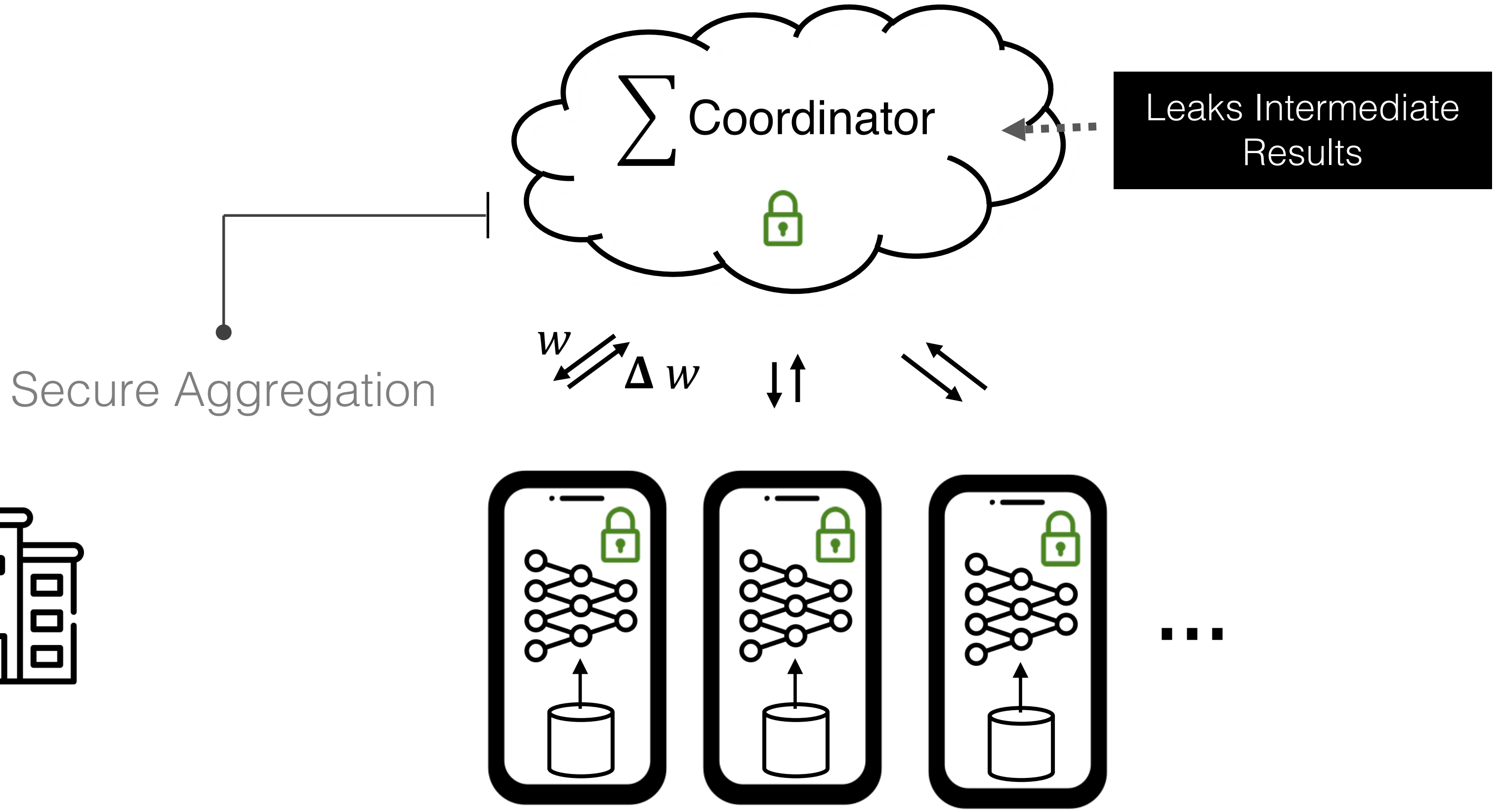


Cryptography → Secure Computation



Secure Decentralised Learning

- CryptoNets [Gilad-Bachrach et al. ICML'16]
- SecureML [Mohassel et al. S&P'18]
- EzPC [Chandran et al. EuroS&P'19]
- Helen [Zheng et al. S&P'19]
- Spindle [Froelicher et al. PETS'20]
- Cerebro [Zheng et al. USENIX Security'21]



Secure Federated Learning

- Secure Aggregation [Bonawitz et al. CCS'17]
- FastSecAgg [Kadhe et al. CCS Workshop PPML'20]
- SecAgg+ [CCS'20]

Cryptography → Secure Computation

Secure Multi-Party Computation (MPC)

Leaks Intermediate Results

Use existing crypto building blocks with careful consideration of performance!

Replace existing ML algorithms with cryptography-friendly ones
(e.g., low degree polynomial, approximate functions)

Final Model

Secure Decentralised Learning

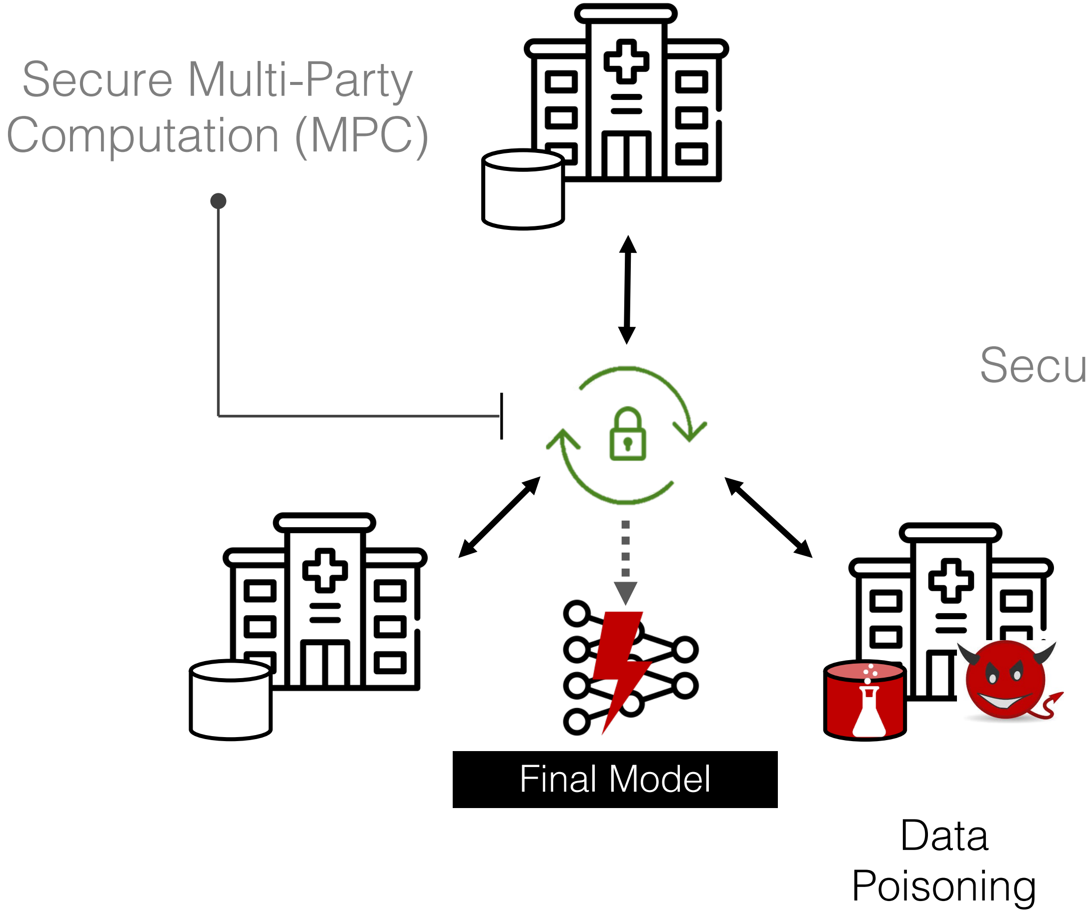
- CryptoNets [Gilad-Bachrach et al. ICML'16]
- SecureML [Mohassel et al. S&P'18]
- EzPC [Chandran et al. EuroS&P'19]
- Helen [Zheng et al. S&P'19]
- Spindle [Froelicher et al. PETS'20]
- Cerebro [Zheng et al. USENIX Security'21]

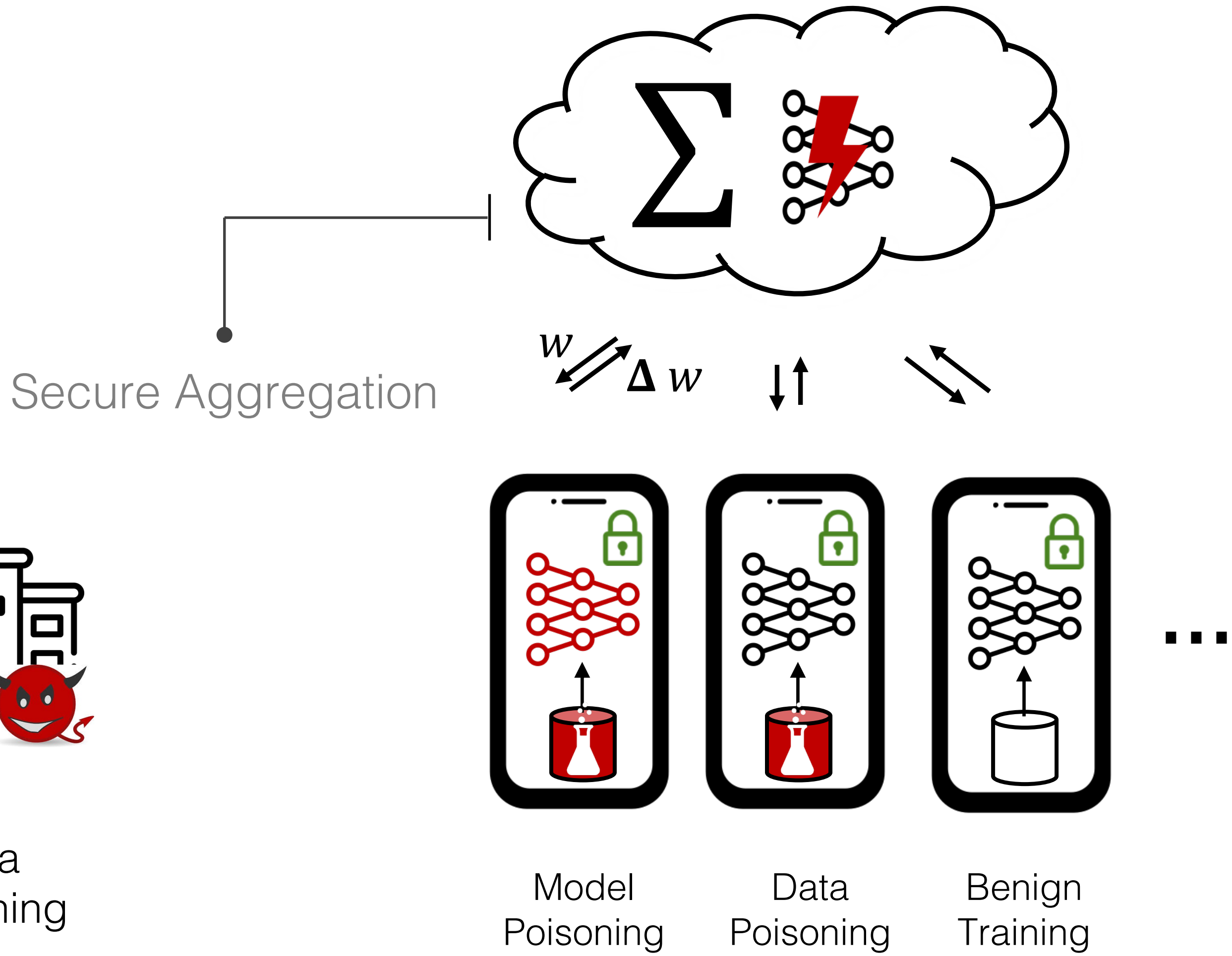
Secure Federated Learning

- Secure Aggregation [Bonawitz et al. CCS'17]
- FastSecAgg [Kadhe et al. CCS Workshop PPML'20]
- SecAgg+ [CCS'20]

Robustness - Malicious Clients

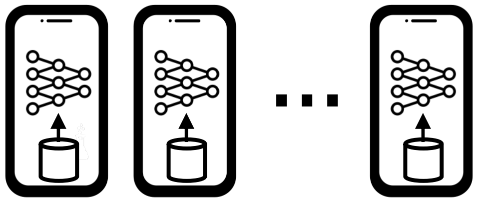
Can Amplify Robustness Issues



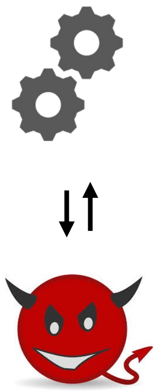


Collaborative Learning

Can Amplify Robustness Issues



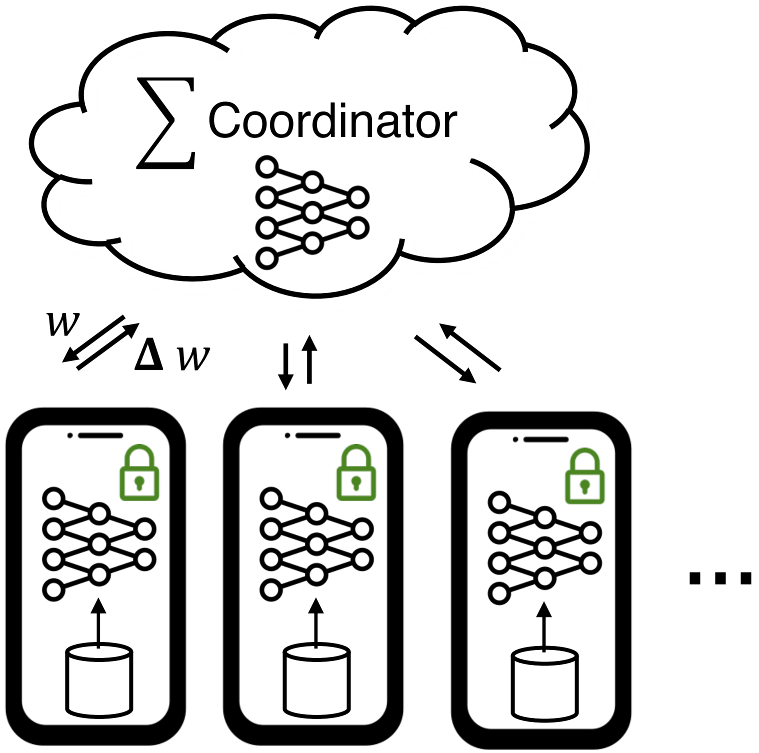
Open Nature



Attacker Capabilities

Detectability

Adversarial Robustness – Training



Secure Federated Learning

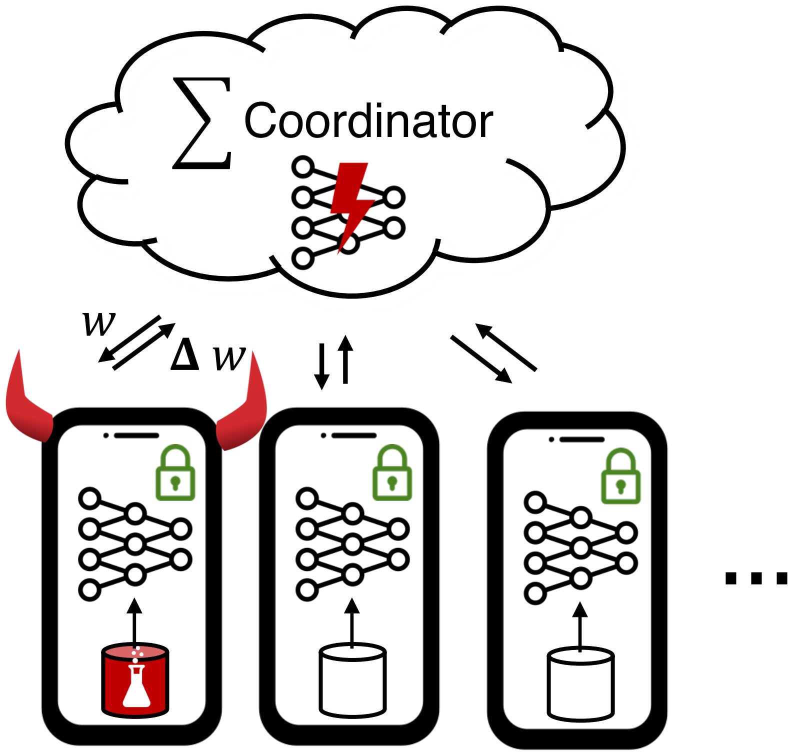
Adversarial Robustness – Training

Data Poisoning

(adversary controls training data)

Train  , Dog

Test  → Dog



Data Poisoning

Secure Federated Learning

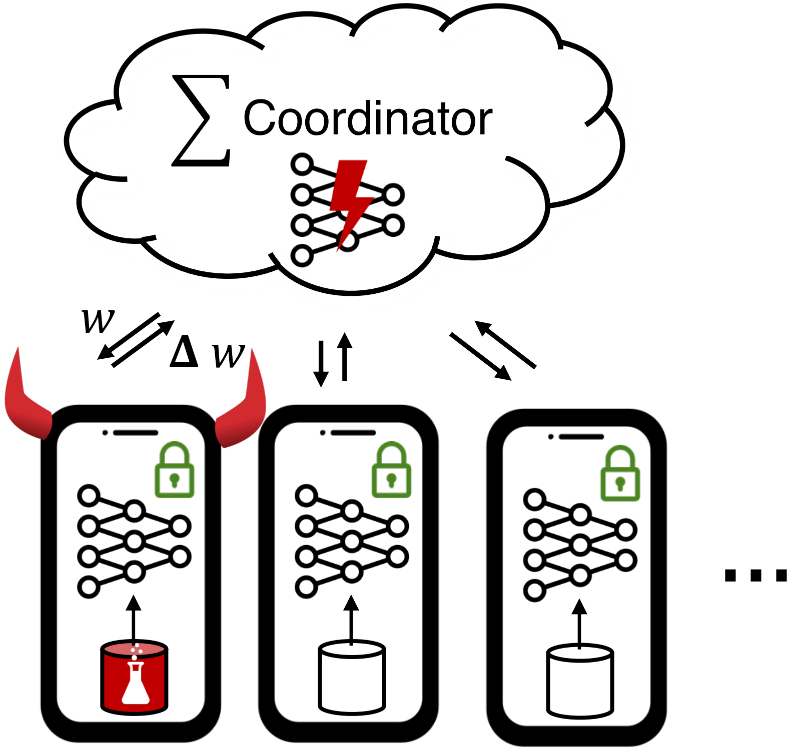
Adversarial Robustness – Training

Data Poisoning

(adversary controls training data)

Train  +  , Dog

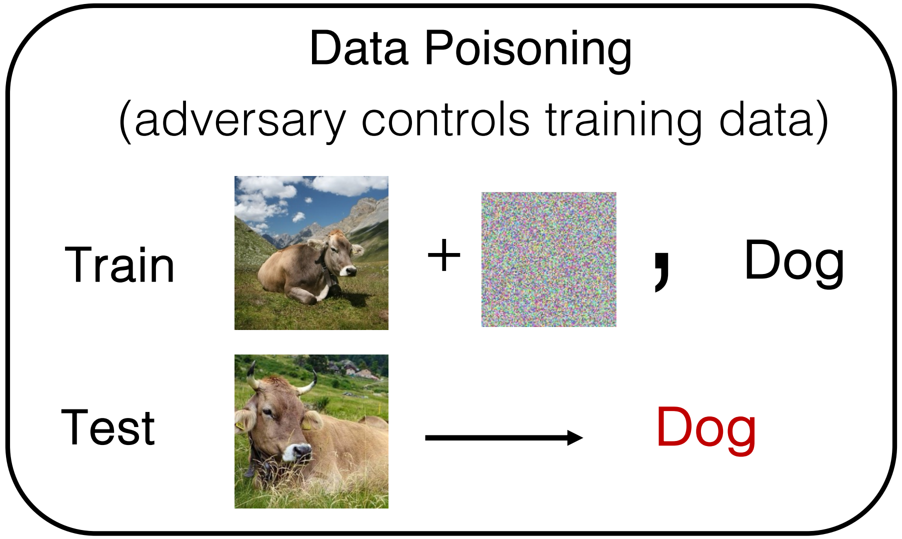
Test  ⟶ Dog

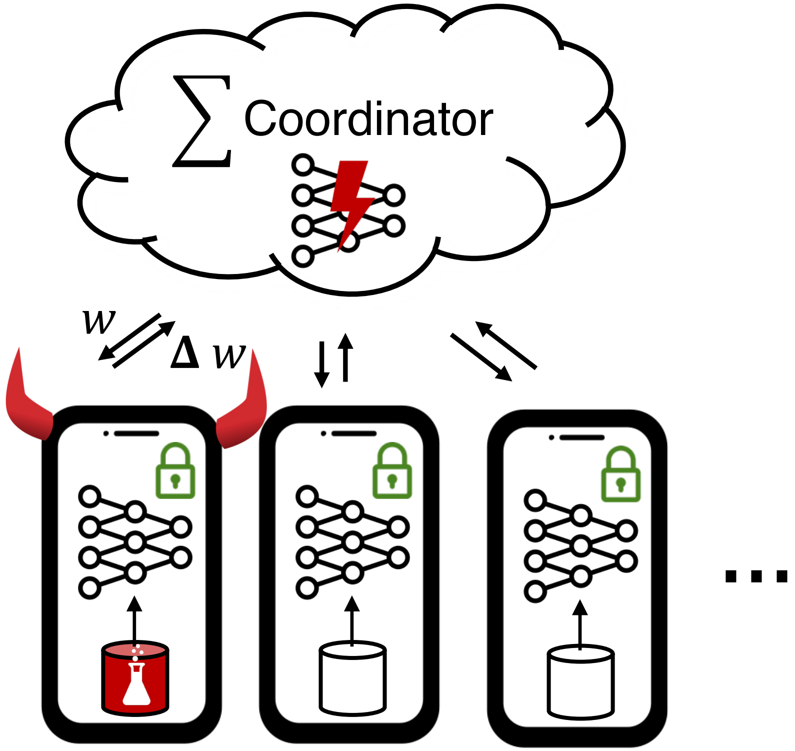


Data Poisoning

Secure Federated Learning

Adversarial Robustness – Training





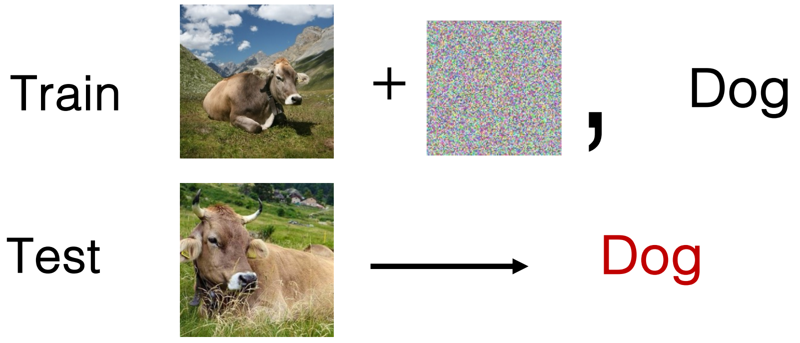
Data Poisoning

Secure Federated Learning

Adversarial Robustness – Training

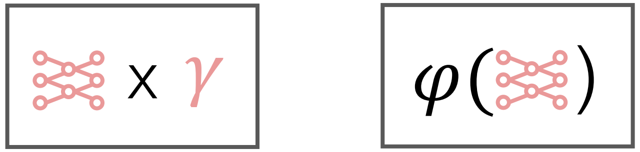
Data Poisoning

(adversary controls training data)



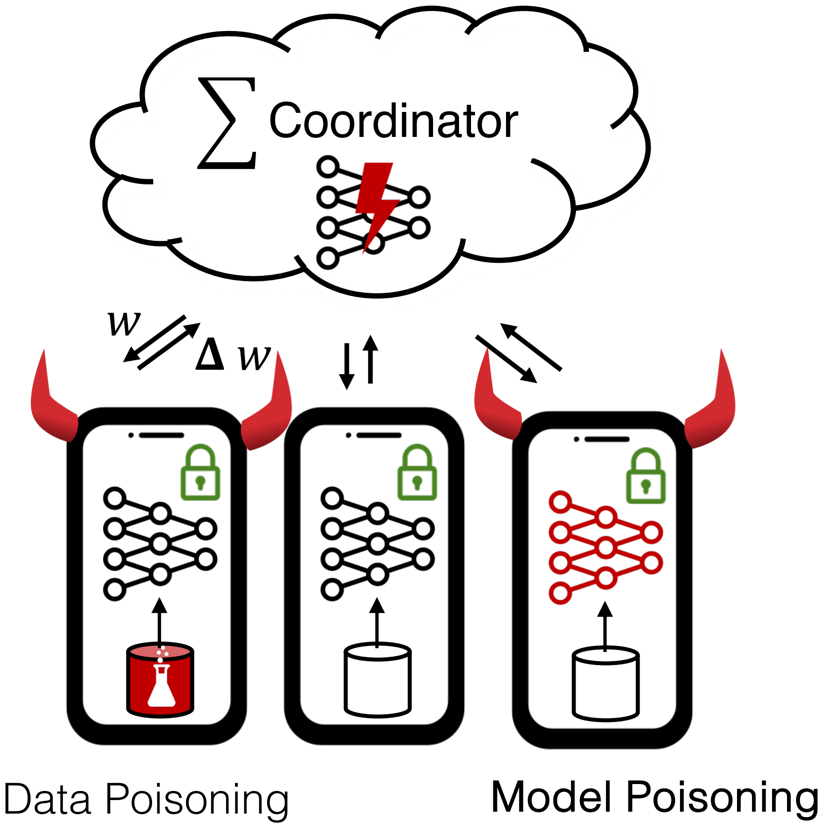
Model Poisoning

(adversary controls model updates)



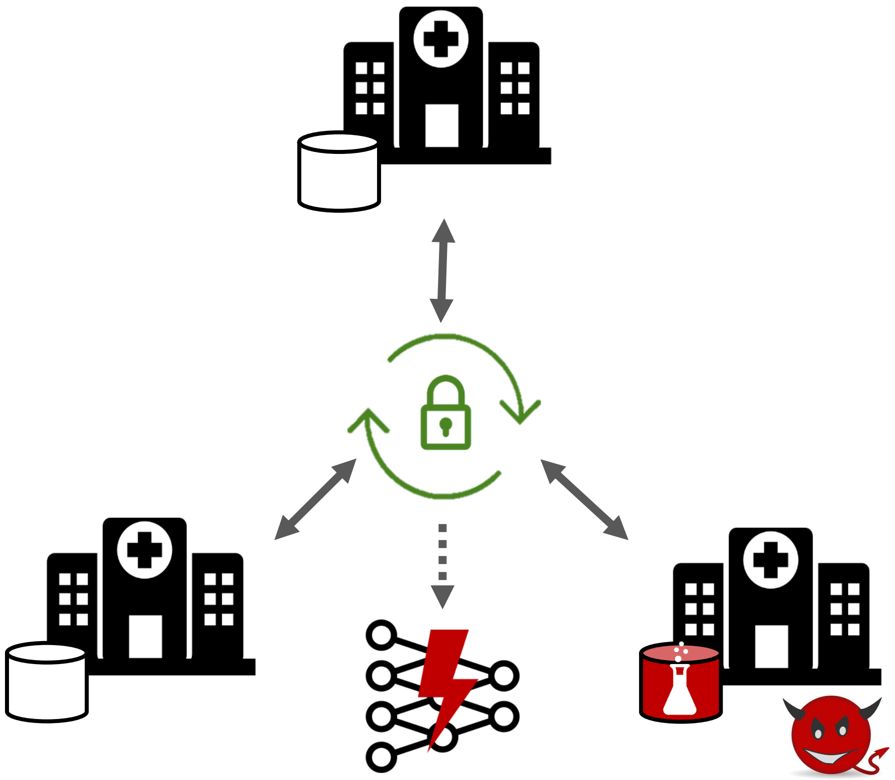
Scaling

Crafted



Secure Federated Learning

Adversarial Robustness – Training



Secure Decentralized Learning

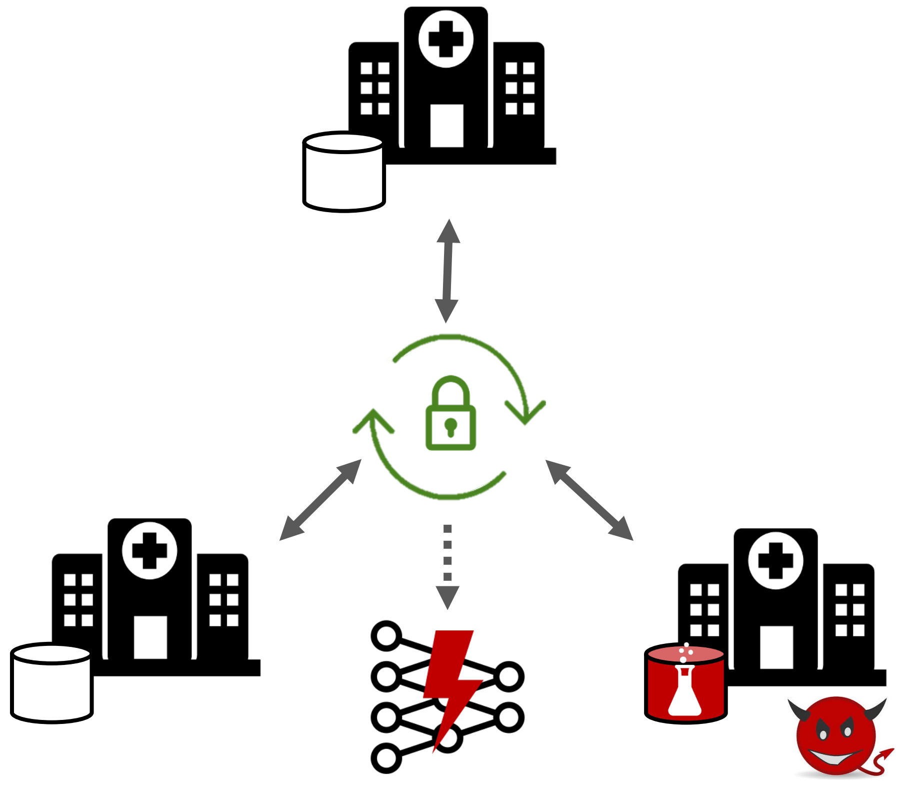
Data Poisoning

(adversary controls training data)

Model Poisoning

(adversary controls model updates)

Adversarial Robustness – Training



Secure Decentralized Learning

Data Poisoning

(adversary controls training data)

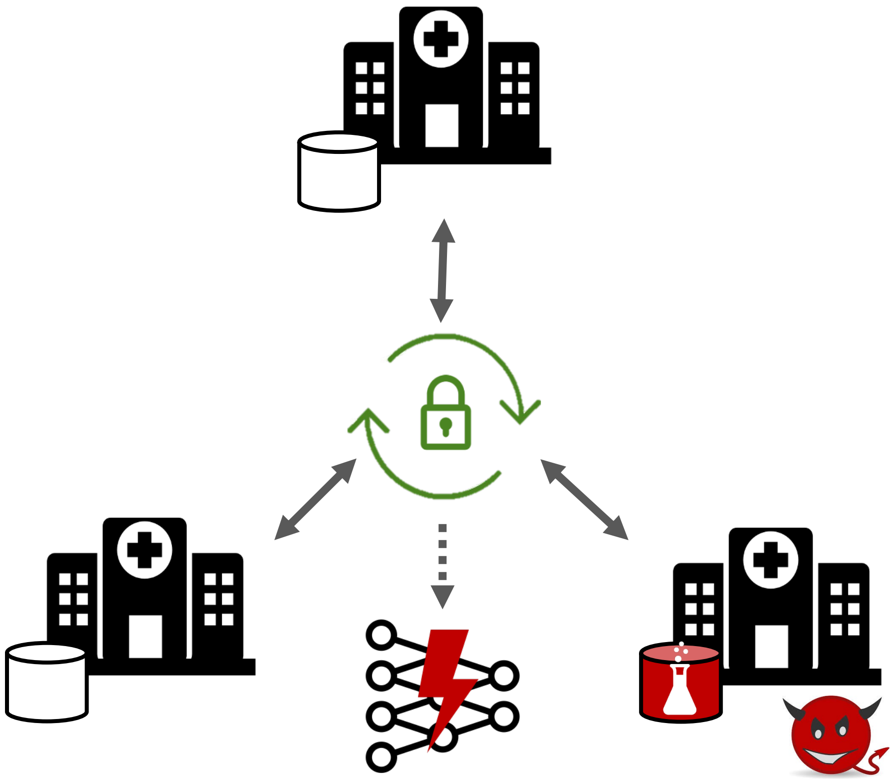
Model Poisoning

(adversary controls model updates)

↓

Malicious security

Adversarial Robustness – Training



Secure Decentralized Learning

Data Poisoning
(adversary controls training data)

Model Poisoning
(adversary controls model updates)

Malicious security

Robust ML Algorithms

cryptography-friendly algorithms

Detection Mechanisms

assumes direct access to the data or the gradients

Cryptography

?

Cryptographic Verification

Zero-knowledge proofs, Cryptographic commitments, Proofs for program delegations, …

Conventional Setting

Verify some pre-specified function $\boldsymbol{f}$

Given $\boldsymbol{P}$(x)
-- Verify: $\boldsymbol{P}(x) = \boldsymbol{f}(x)$

Machine Learning Setting

In ML $\boldsymbol{f}$ is learned
($\boldsymbol{f}$ = unknown ground truth)

Given $\boldsymbol{P}$(x)
-- Verify what then?

The source of the issue is maliciously chosen data
→ alteration, proof/verify **something** about the input data, gradients, or data distribution

- Theoretical work: Verify data distribution (in/out/adversarial)
- Enforce constrains on the gradient updates (e.g., norm bound)
- Verify Source of Data
- …

Overview Wrap Up

- Decoupling data from training, by itself, does not provide many privacy benefits
  - Encryption can help (e.g., secure aggregation, MPC)

- More work on robust ML in the **encrypted settings**
  - Cryptography friendly robust ML algorithms
  - Use cryptography (e.g., verification, ZKP) to minimize influence of maliciously chosen training data

- Post-Deployment
  - Can we get robustness against all attacks? **Answer:** A perfect solution to adversarial robustness remains an open challenge – imperfect defenses, cat-and-mouse game, more powerful attacks
  - There is a need for solutions that minimize consequences of attacks at deployment time – e.g., attribution, forensics, accountability, audits, admission controls, monitoring …

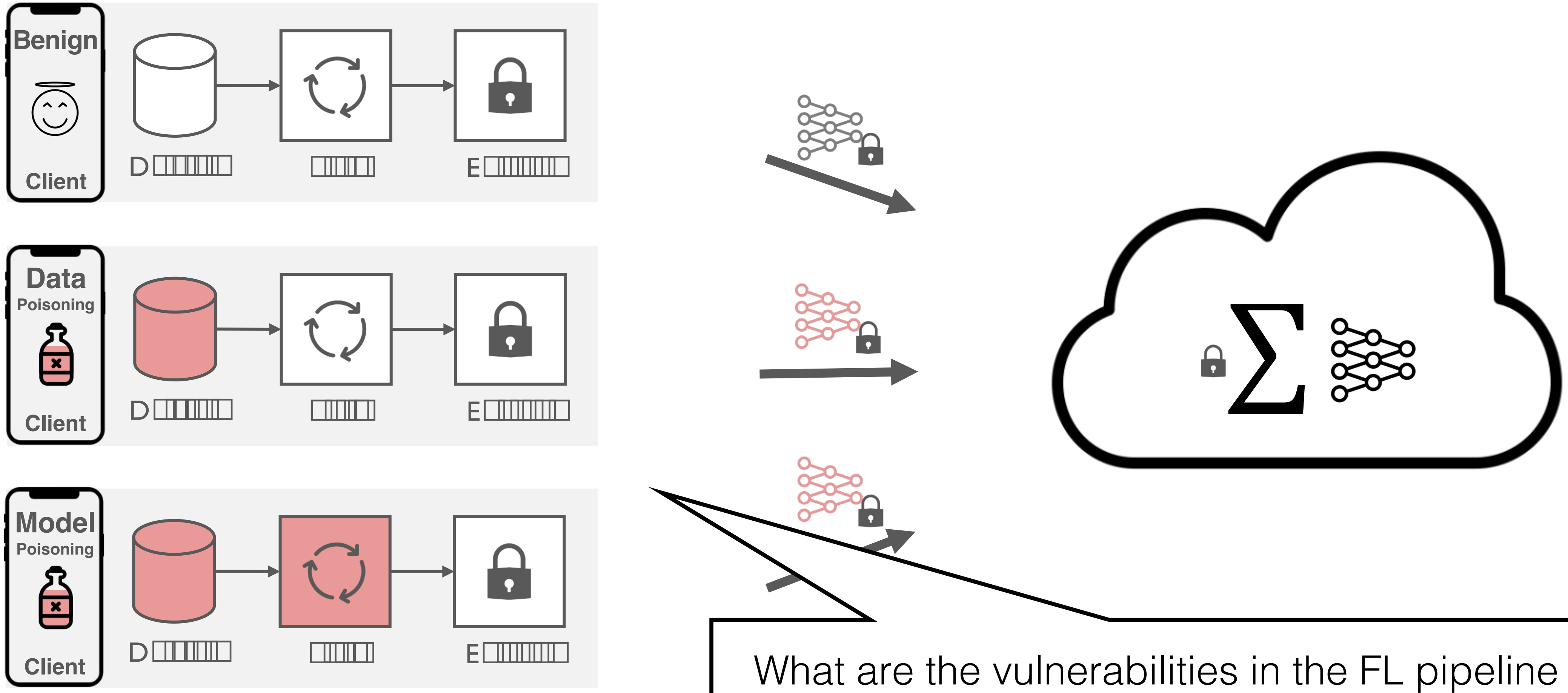
RoFL: Robustness of Secure Federated Learning

IEEE S&P'23

Understand **Vulnerabilities** in FL

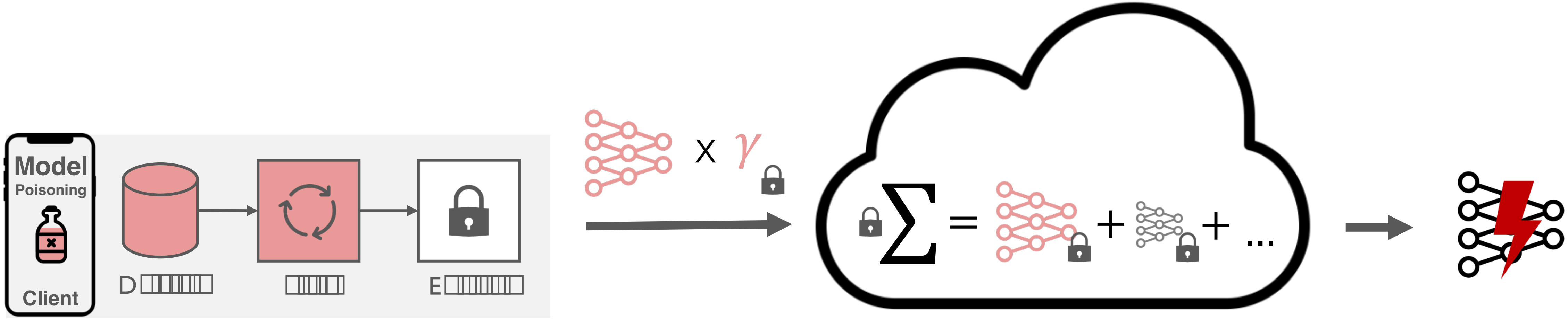
Cryptographically **Enforce** Constraints

Adversarial Clients

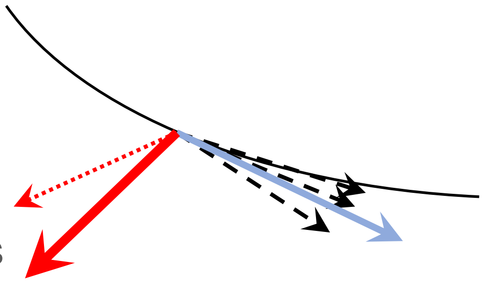


What are the vulnerabilities in the FL pipeline that enable model/data poisoning attacks ?

Adversarial Clients



Problem: Linear aggregation rules are vulnerable to Byzantine behavior

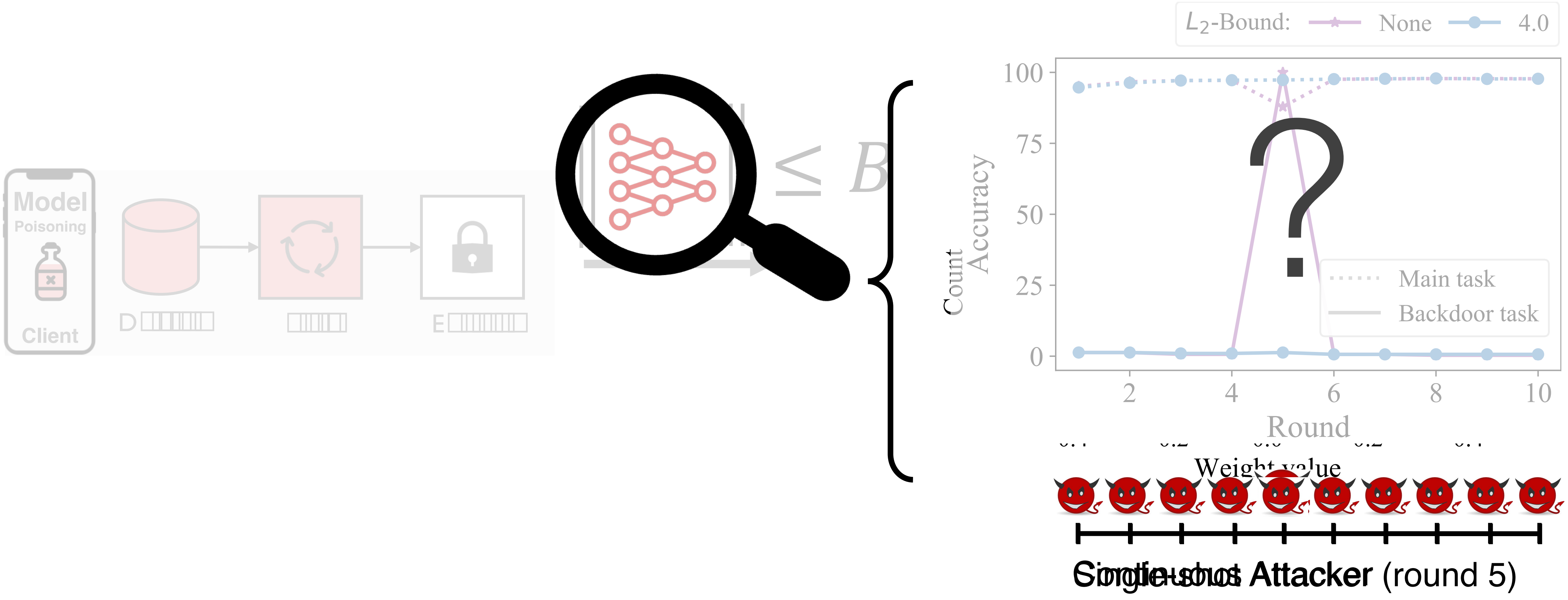


Machine Learning: Byzantine-Robust Distributed Learning	**Security:** Private Data-Collection Systems
- Krum [Blanchard et al. NeurIPS'17] - Trimmed Mean [Yin et al. ICML'18] - Coordinate-wise Median [Yin et al. ICML'18] - Bulyan [Mhamdi et al. ICML'18] - ByzantineSGD [Alistarh et al. NeurIPS'18] - Redundant Workers and Coding Theory [Chen et al. ICML'18, Rajput et al. NeurIPS'19]	- Prio [Corrigan-Gibbs et al. NSDI'17] - PrivStats [Popa et al. CCS'11] - SplitX [Chen et al. SIGCOMM'13] - P4P [Duan et al. USENIX Security'10] - PrivEx [Elahi et al. CCS'14]

→ Zero Knowledge Proofs: client proves that its submission is well-formed

A well-formed Client Submission in Federated Learning

Norm bound


Model
Poisoning
Client
$\leq B$
L_2-Bound: None 4.0
100
75
50
25
0
Accuracy
Count
2 4 6 8 10
Round
Main task
Backdoor task
?
Weight value
Single-shot Attacker (round 5)

Is the norm bound actually effective?

How To Backdoor Federated Learning

Can You Really Backdoor Federated Learning?

Attack of the Tails:
Yes, You Really Can Backdoor Federated Learning

Bagdasaryan et al., *How to backdoor Federated Learning,* AISTATS 2020
Sun et al., *Can you really backdoor federated learning?*, Federated learning workshop at NeurIPS 2019
Wang et al., *Attack of the Tails: Yes, You Really Can Backdoor Federated Learning,* NeurIPS 2020

Why?

Long Tail …

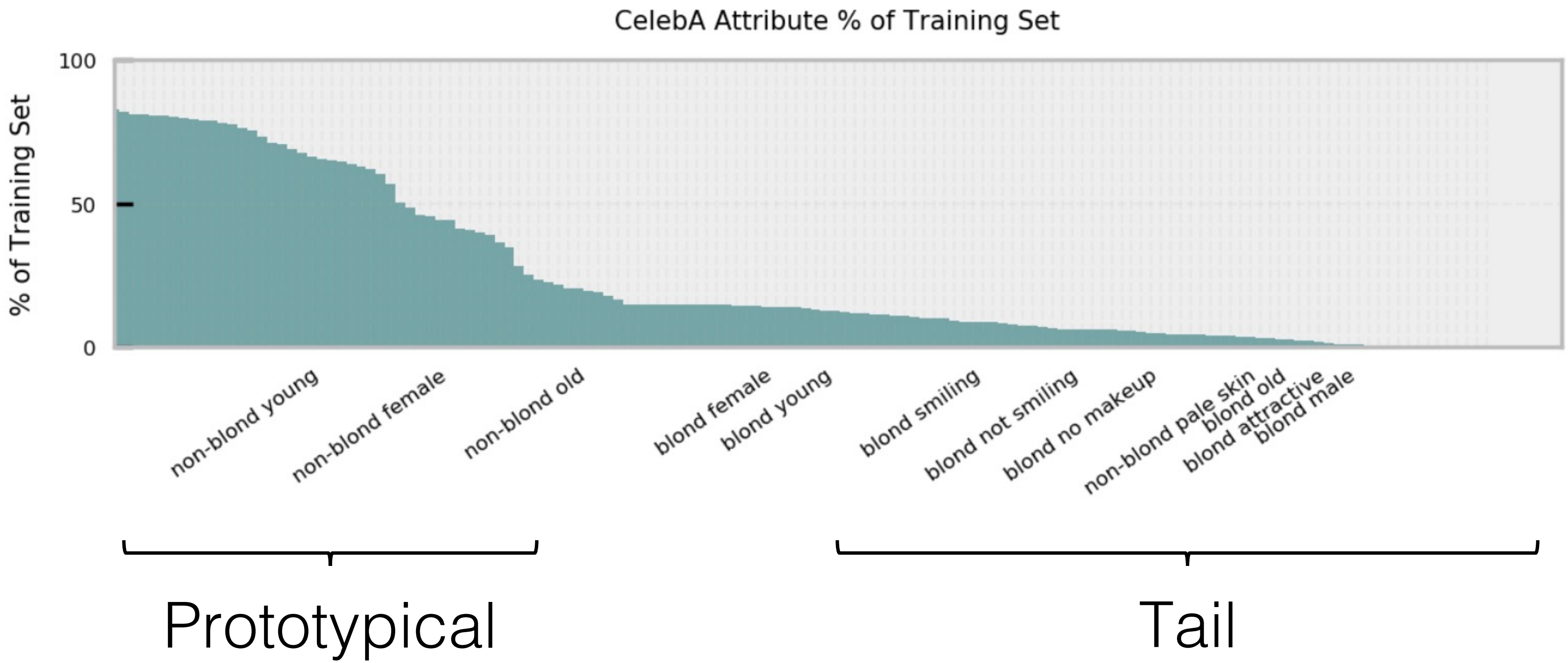
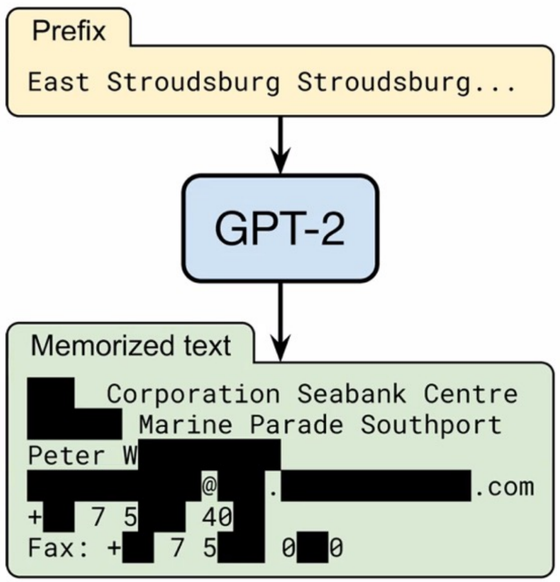
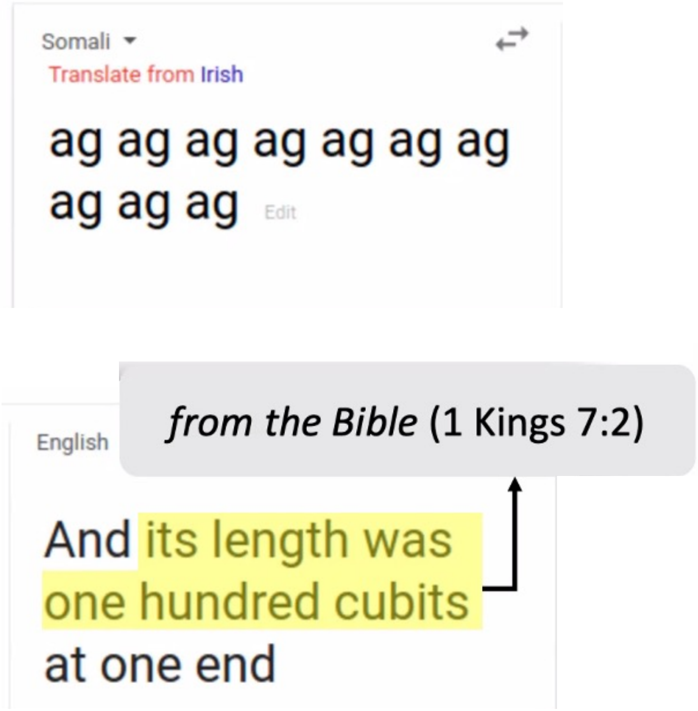


Fig: Hooker, Moorosi et al., 2020.

Model Capacity Implications on Privacy …





Memorization leads to leakage of private text

Fig Left – Carlini et al., *Extracting Training Data from Large Language Models*, USENIX Security 2021.
Fig Right – Tramer, From average-case to worst-case privacy leakage in neural networks", talk at Privacy and Security in ML Interest Group, 2022.

Model Capacity Implications on Robustness…

Analysis: Understanding FL Robustness

Adaptive attacks

MP-PD: Projected Gradient Descent [Sun et al., FLDPC@NeurIPS’19]

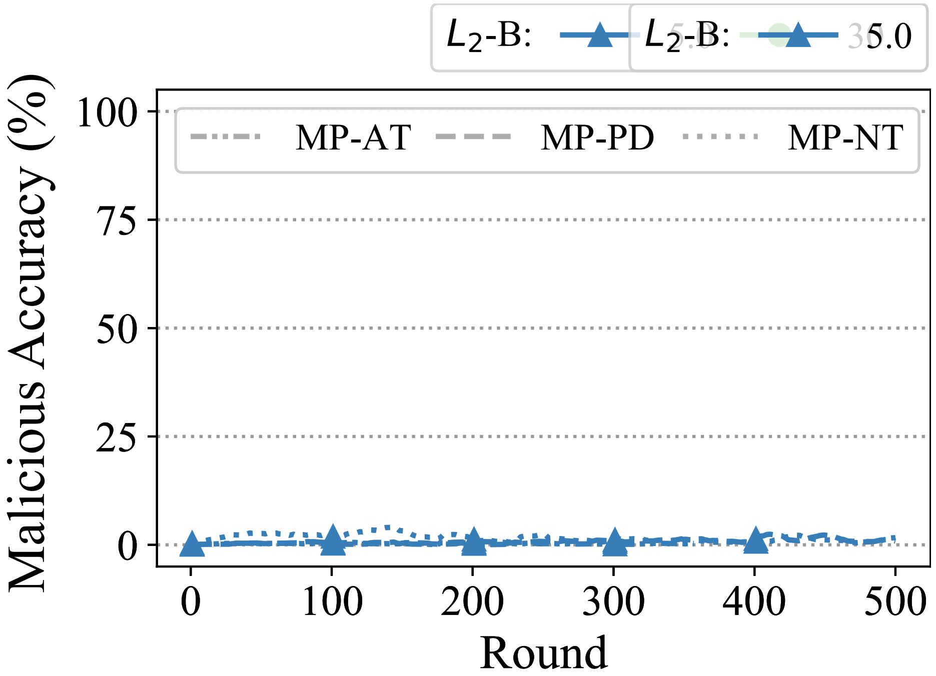
MP-NT: Neurotoxin [Zhang et al., ICML’22]

MP-AT: Anticipate [Wen et al., AdvML@ICML’22]

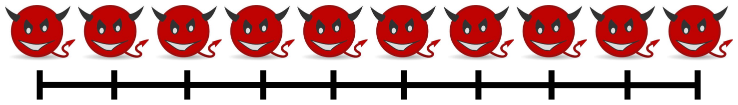
Considered:

- Attack Objective
- Number of Attackers
- Bound Selection
- Pixel-Pattern Backdoors
- Untargeted Attacks

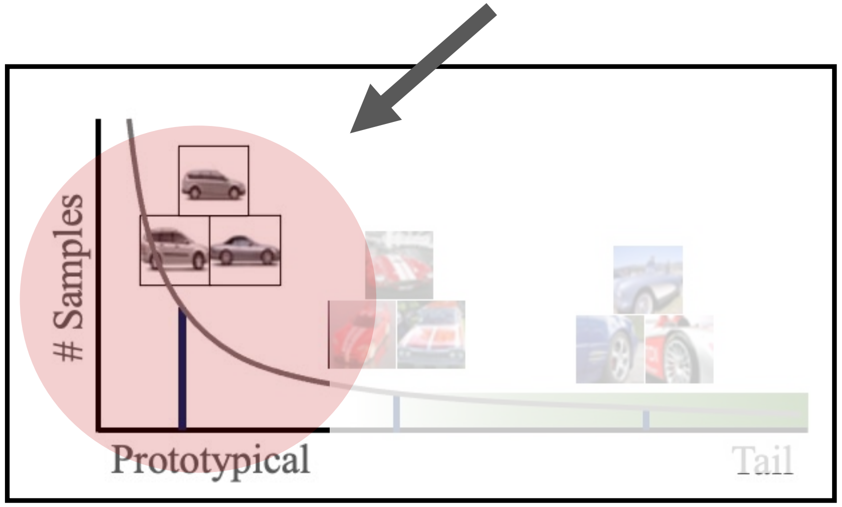
Impact of Attack Objective on Backdoor Attacks



CIFAR-10

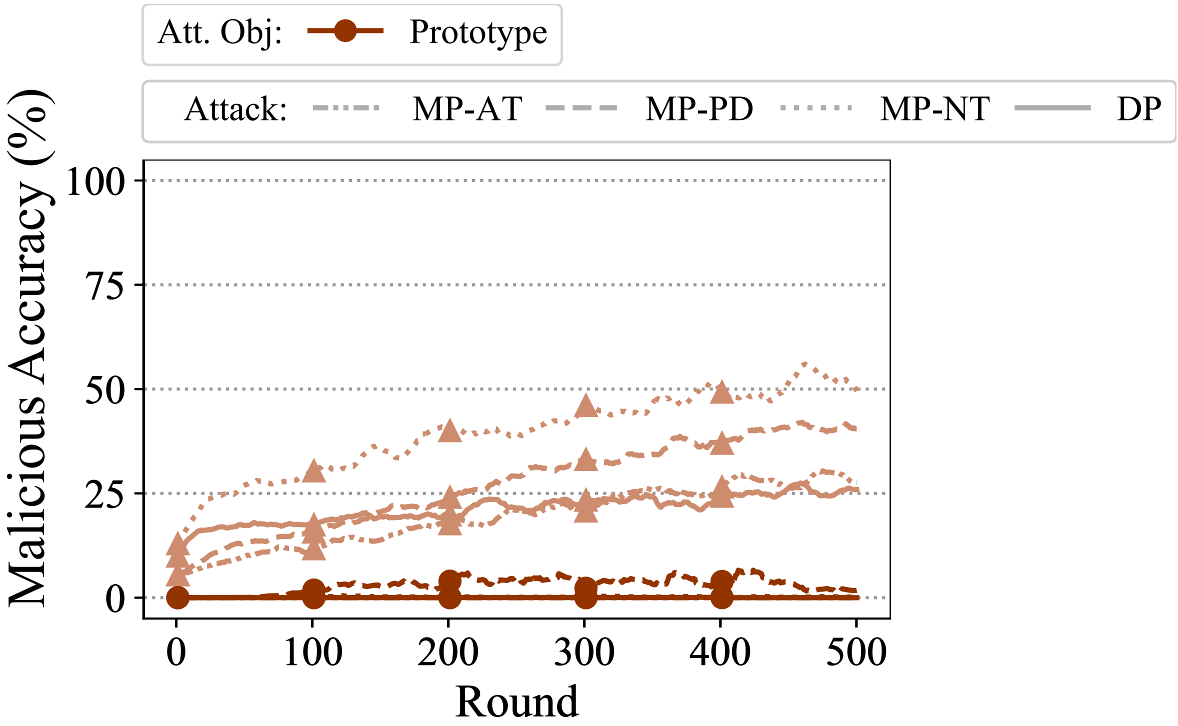


Continuous Attacker

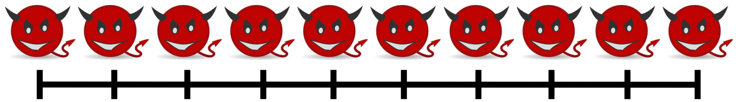


Prototypical Targets

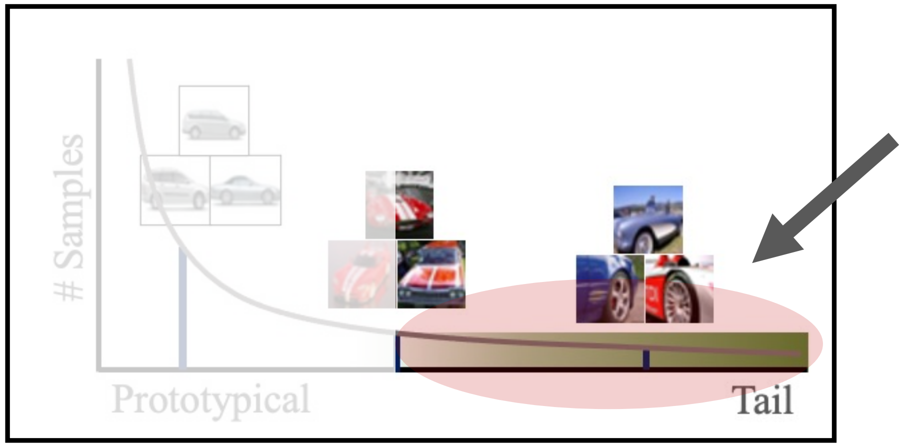
Impact of Attack Objective on Backdoor Attacks



CIFAR-10

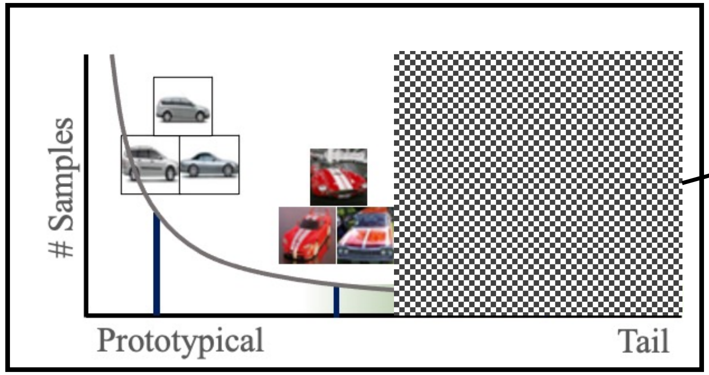


Continuous Attacker



Tail Targets

Suppressing the Long-Tail



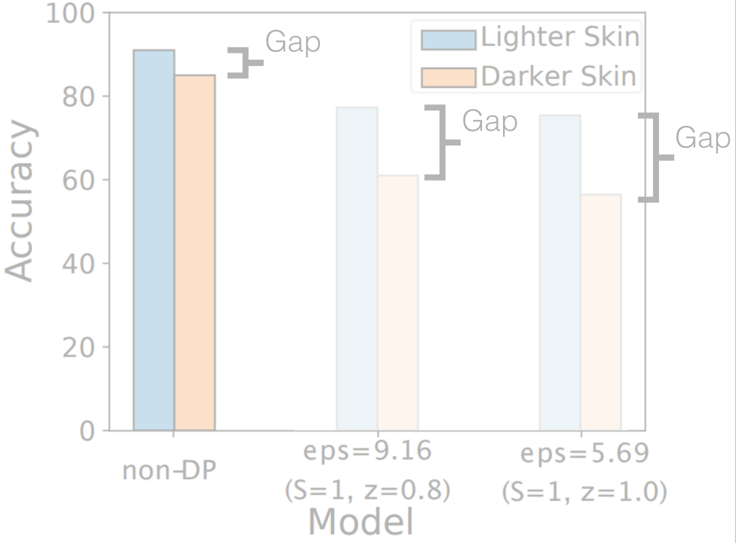
Approaches

- Noise Addition (Differential Privacy)
- Compression

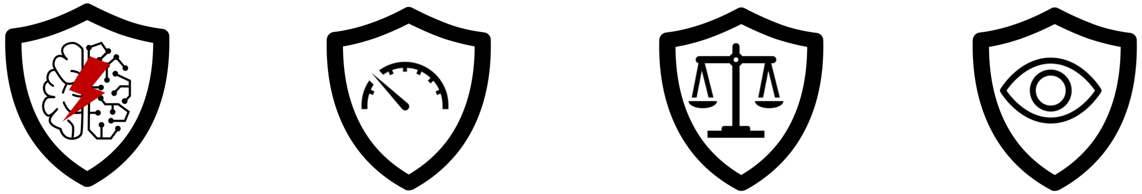
Leads to Fairness Problems

Differential Privacy disproportionately impacts underrepresented attributes

[Bagdasaryan et al. NeurIPS 2019]

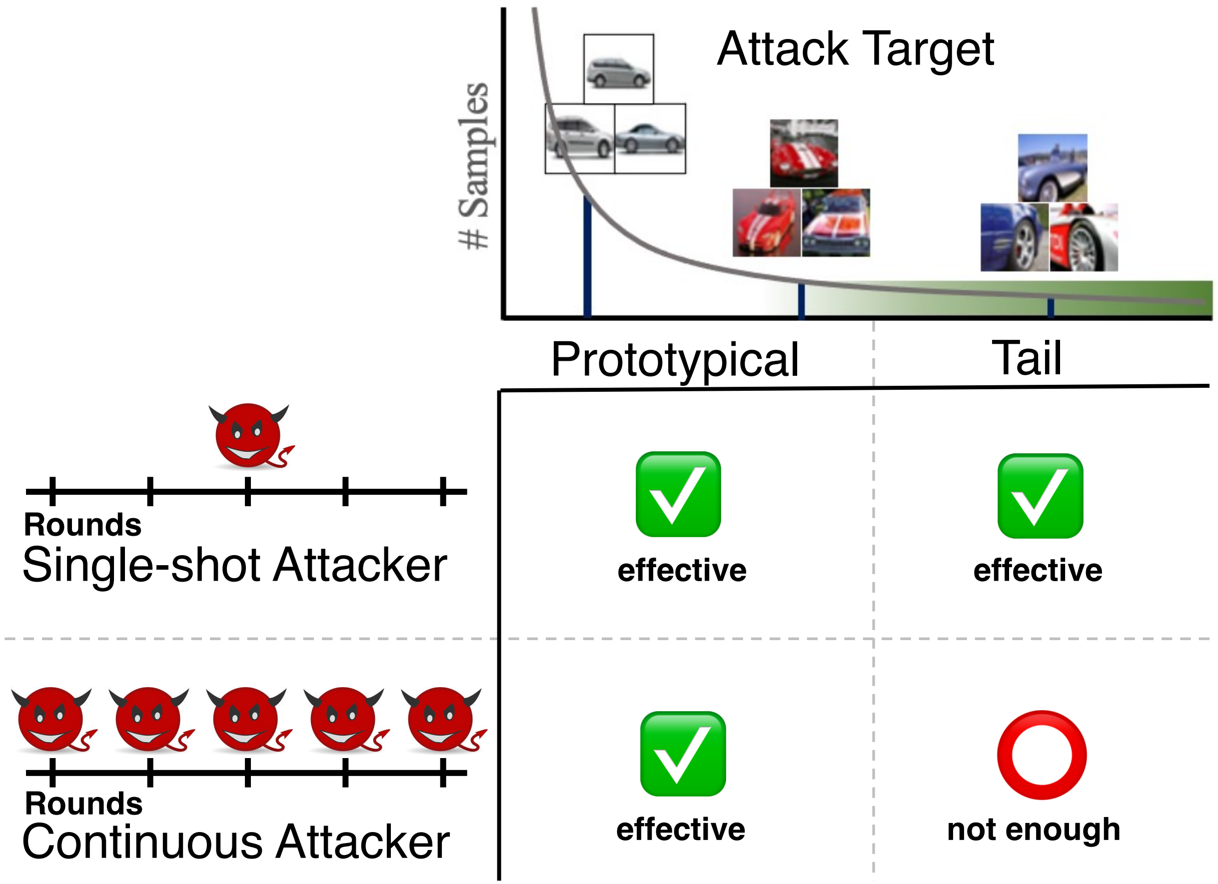


Understand trade-offs between objectives we care about



Robustness Accuracy Fairness Privacy

Norm Bound Provides Practical Robustness Guarantees



… requires a strong attacker that is consistently selected and targets a tail sample

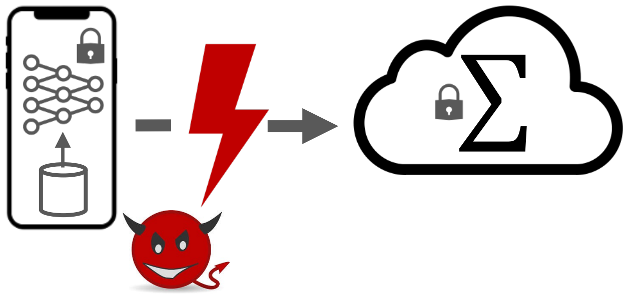
Hinges on it being efficiently realizable in the secure setting …

RoFL: Robustness of Secure Federated Learning

Understand
Vulnerabilities in FL

Cryptographically
Enforce Constraints

Goal: Augment existing secure FL with Zero-Knowledge Proofs to enforce constraints on model updates

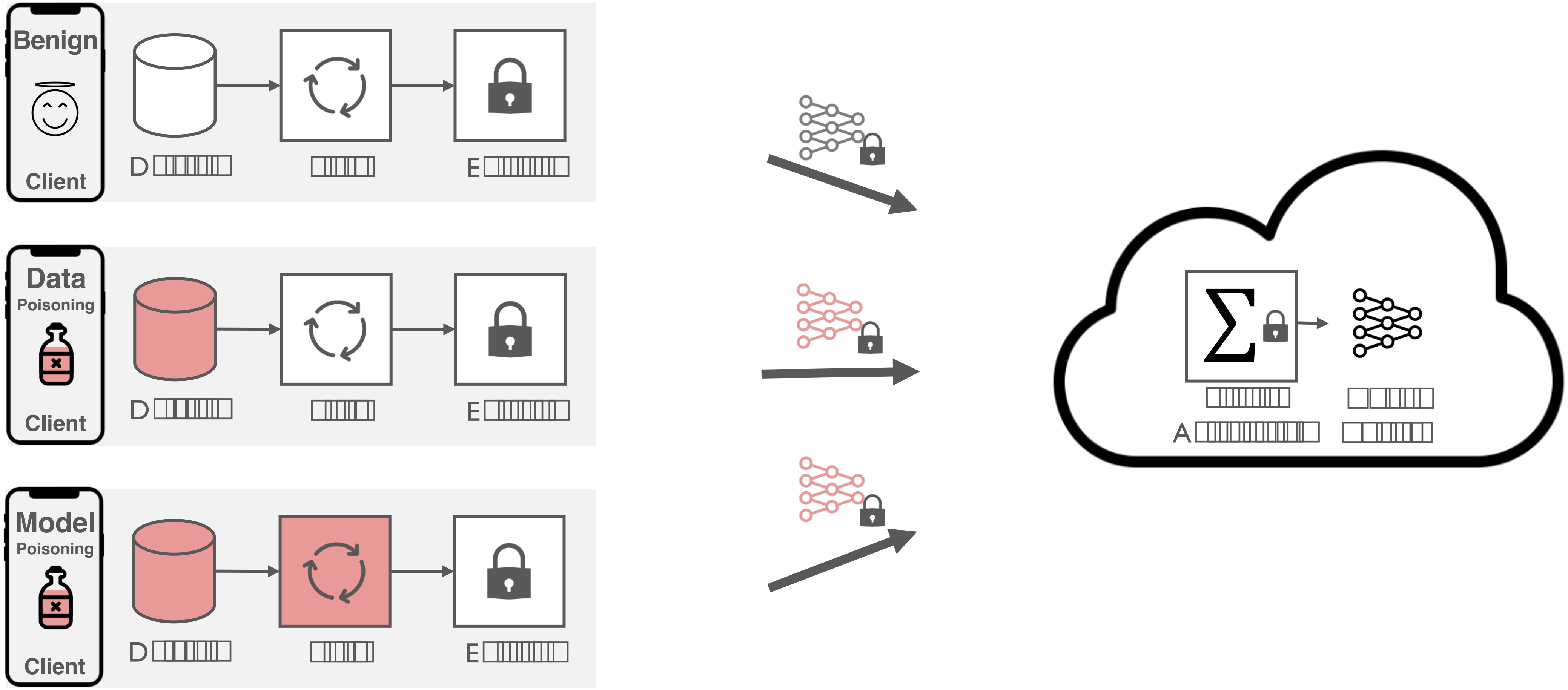


$$\left\{ \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_d \end{bmatrix} \right\}$$

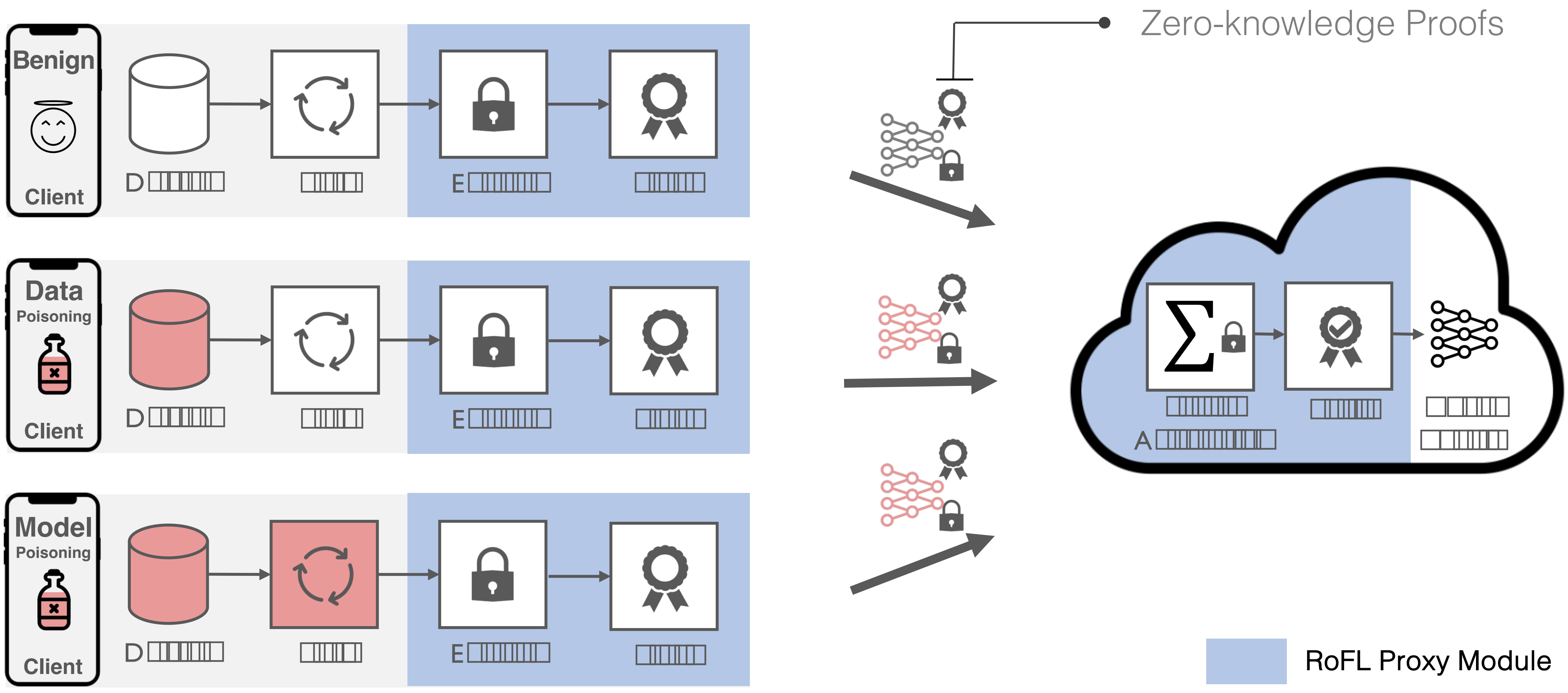
- Compressed Sigma protocols
- Optimistic continuation
- Probabilistic checking
- Subspace learning

Correctness	Private Input Validation	Optimizations

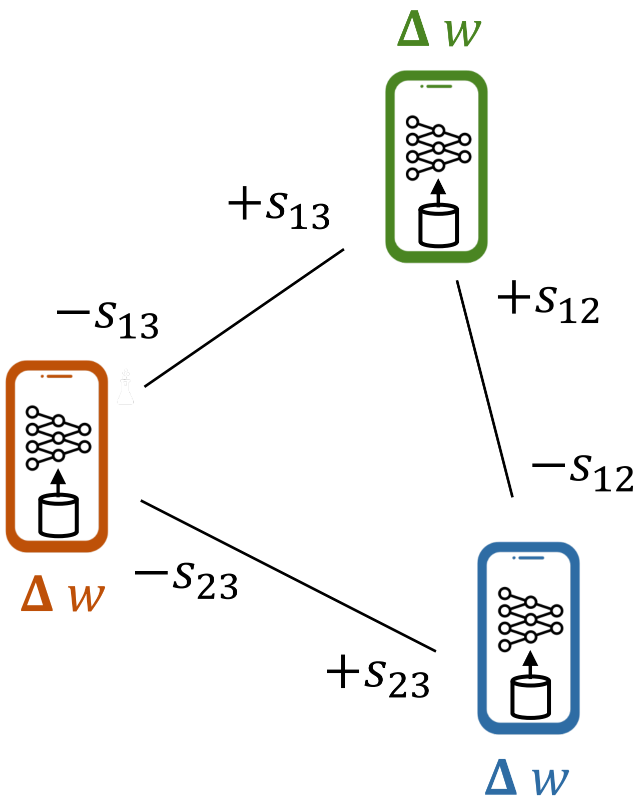
Secure Federated Learning



RoFL Augments Secure Federated Learning



Secure Aggregation



Goal: Compute $\sum \Delta w_i = \Delta w + \Delta w + \Delta w$

Idea: Additive masks based on pairwise secrets s_{ij}

$$r_1 + r_2 + r_3 = 0$$

where

$r_1 = s_{12} + s_{13}$

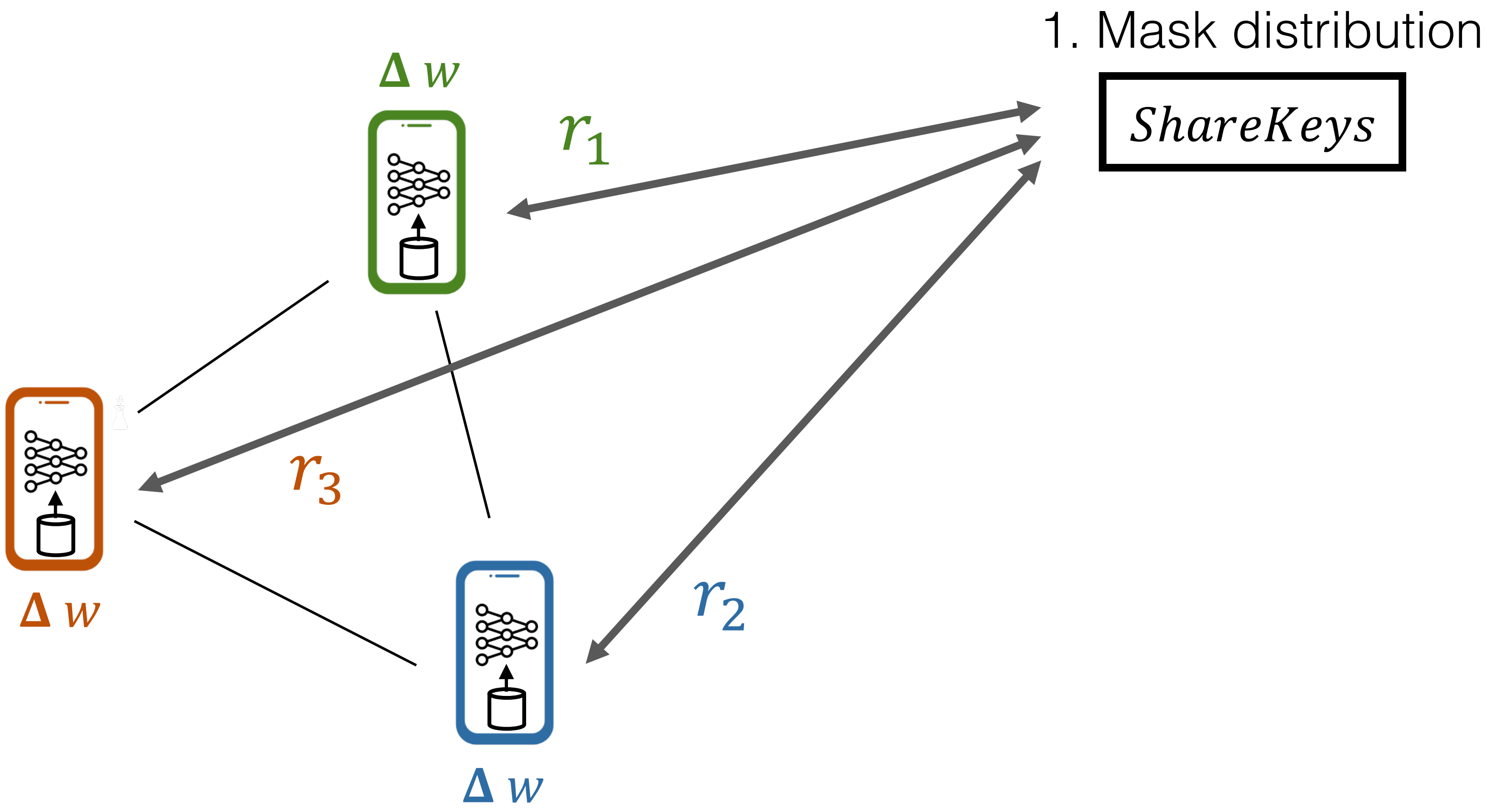
$r_2 = -s_{12} + s_{23}$

$r_3 = -s_{13} - s_{23}$

+: modular addition

Bonawitz et al., *Practical Secure Aggregation for Privacy-Preserving Machine Learning*, CCS 2017.

Secure Aggregation



+: modular addition

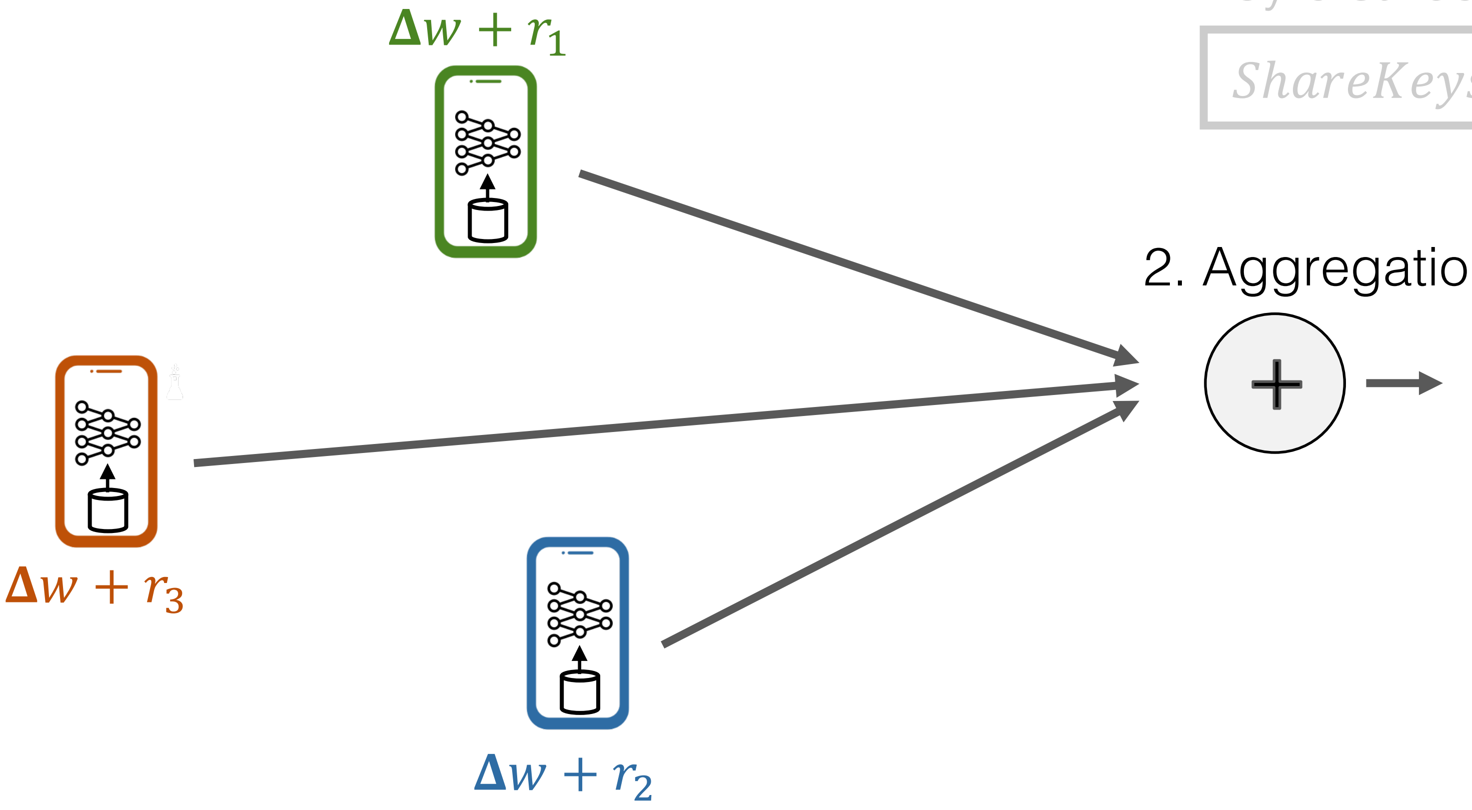
Bonawitz et al., *Practical Secure Aggregation for Privacy-Preserving Machine Learning*, CCS 2017.

Secure Aggregation

1. Key distribution

ShareKeys

2. Aggregation

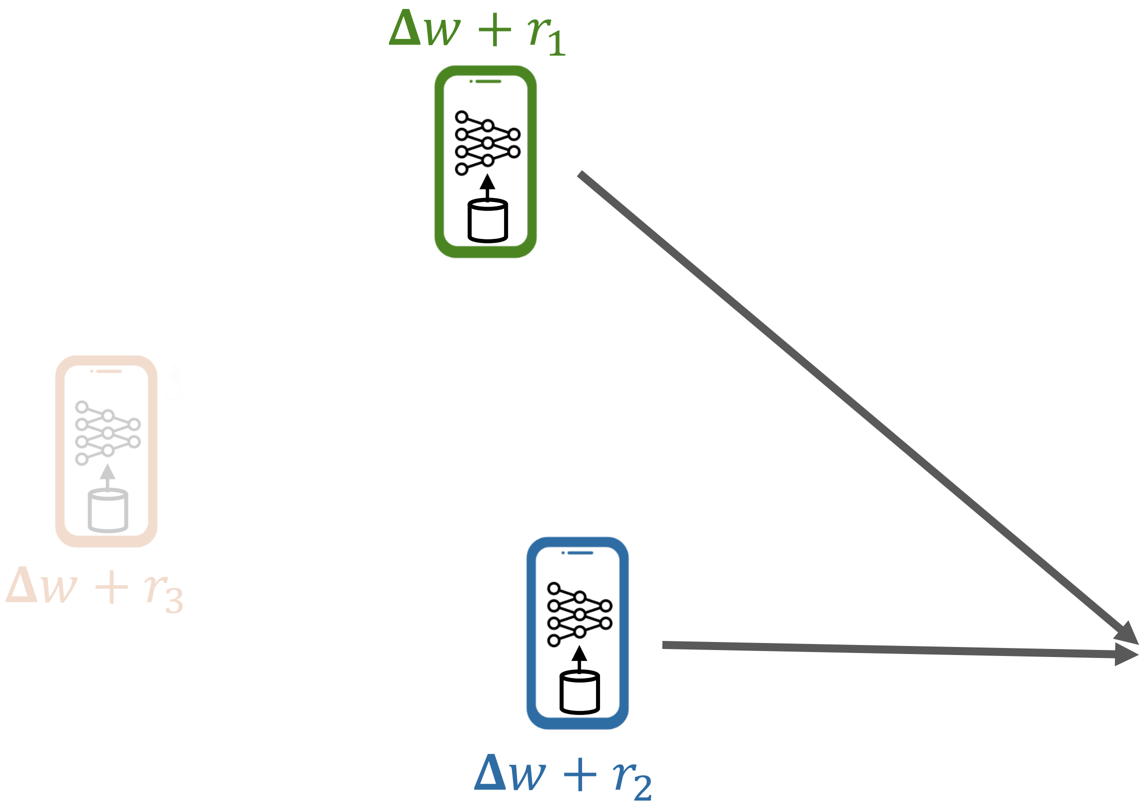


$$\sum(\boldsymbol{\Delta} w_i + r_i) = \sum \boldsymbol{\Delta}\, w_i + \sum r_i$$

+: modular addition

Bonawitz et al., *Practical Secure Aggregation for Privacy-Preserving Machine Learning*, CCS 2017.

Secure Aggregation



1. Key distribution

$\boxed{ShareKeys}$

2. Aggregation

$\oplus \rightarrow \sum(\Delta w_i + r_i) = \sum \Delta w_i + \sum r_i$

3. Reconstruction

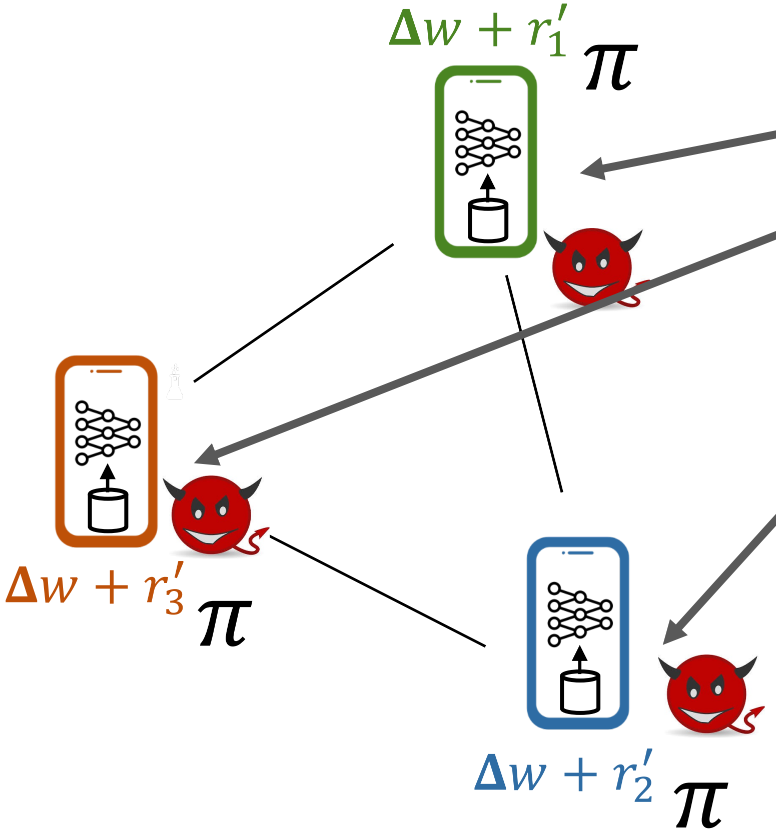
$\boxed{UnMask} \rightarrow r'$

$$\left(\sum \Delta w_i + \sum r_i\right) - r' = \sum \Delta w_i$$

+: modular addition

Bonawitz et al., *Practical Secure Aggregation for Privacy-Preserving Machine Learning*, CCS 2017.

Limitation: Correctness with malicious clients



1. Key distribution

$\boxed{ShareKeys}$ π

2. Aggregation

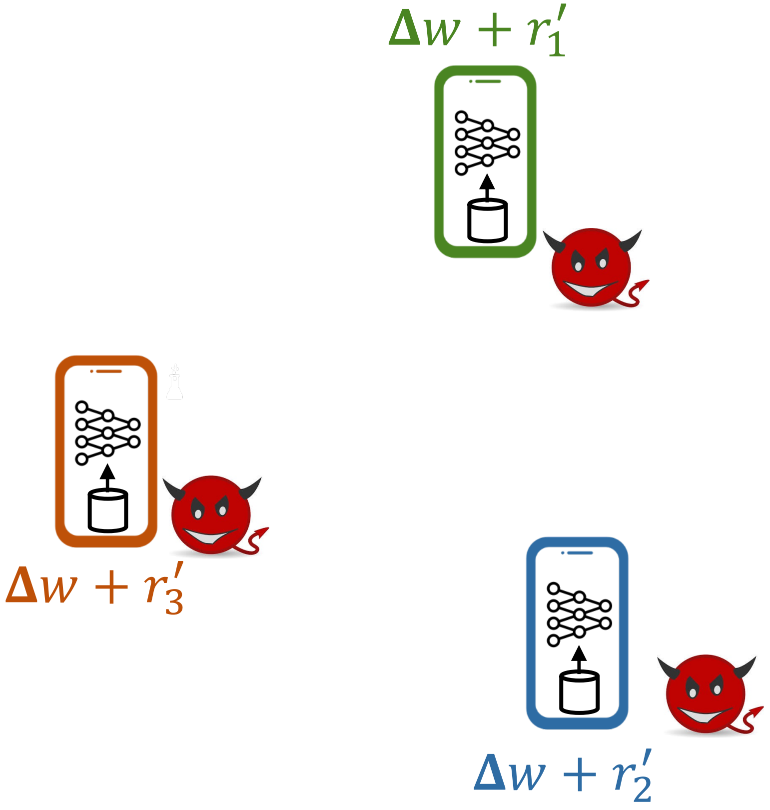
$\oplus \rightarrow \sum(\Delta w_i + r_i) = \sum \Delta w_i + \sum r_i$

3. Reconstruction

$\boxed{UnMask} \rightarrow r'$

$$\left(\sum \Delta w_i + \sum r_i\right) - r' = \sum \Delta w_i$$

Insight: Checking $\sum r_i = r'$ sufficient for correctness



1. Key distribution

$\boxed{ShareKeys}$

2. Aggregation

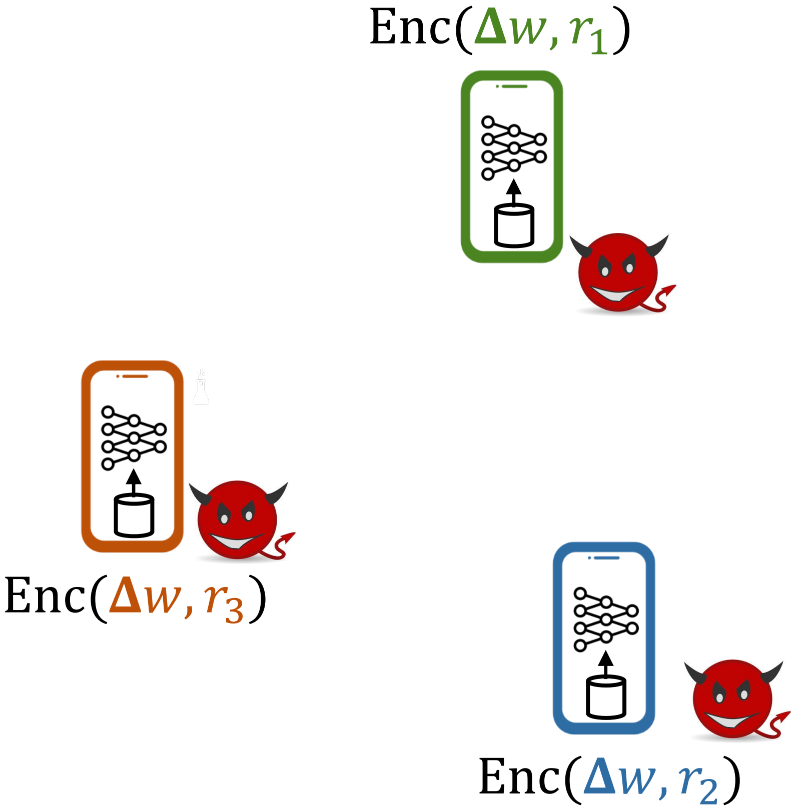
$$\oplus \rightarrow \sum(\Delta w_i + r_i) = \sum \Delta w_i + \sum r_i$$

3. Reconstruction

$$\boxed{UnMask} \rightarrow r'$$

$$\left(\sum \Delta w_i + \sum r_i\right) - r' = \sum \Delta w_i$$

Insight: Checking $\sum r_i = r'$ sufficient for correctness



1. Key distribution

$\boxed{ShareKeys}$

2. Aggregation

$+ \rightarrow \sum \mathrm{Enc}(\Delta w_i, r_i) = \mathrm{Enc}(\sum \Delta w_i, \sum r_i)$

3. Reconstruction

$\boxed{UnMask} \rightarrow r'$

$$\mathrm{Dec}(\mathrm{Enc}(\sum \Delta w_i, \sum r_i), r') = \sum \Delta w_i$$

Efficiency hinges on compatibility with zero-knowledge proofs

Protocol Requirements

$$\sum_i \mathrm{Enc}(\boldsymbol{\Delta} w_i, r_i) = \mathrm{Enc}\left(\sum_i \boldsymbol{\Delta} w_i, \sum_i r_i\right)$$

Homomorphic in messages and keys

Correctness check

Additively Homomorphic Commitments

ZKP Requirements

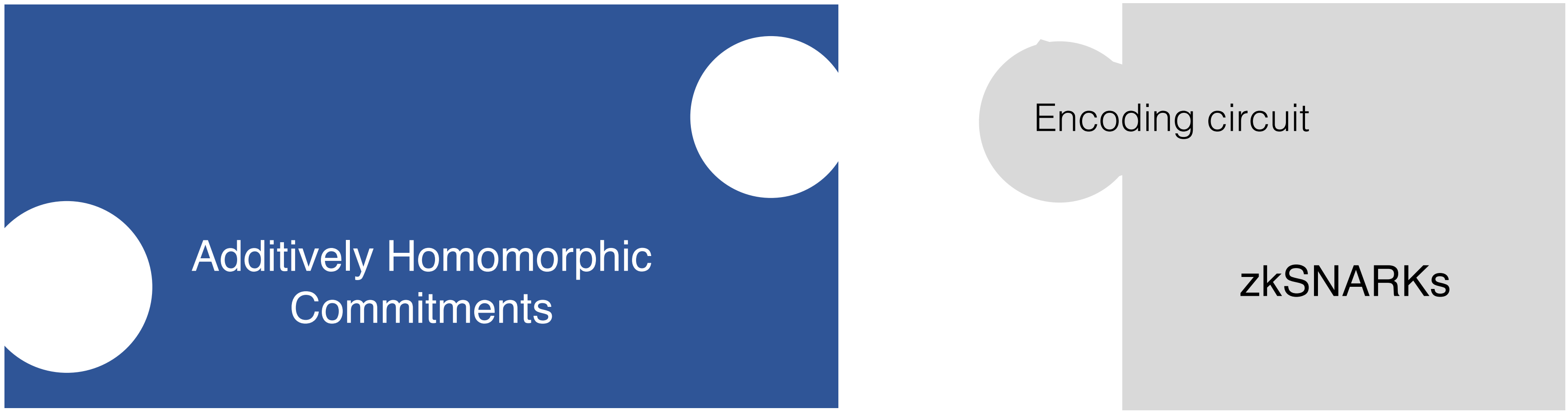
$$\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_\ell \end{bmatrix}$$

Range proofs over large vectors

Resource-constrained devices

Compatibility with Commitments

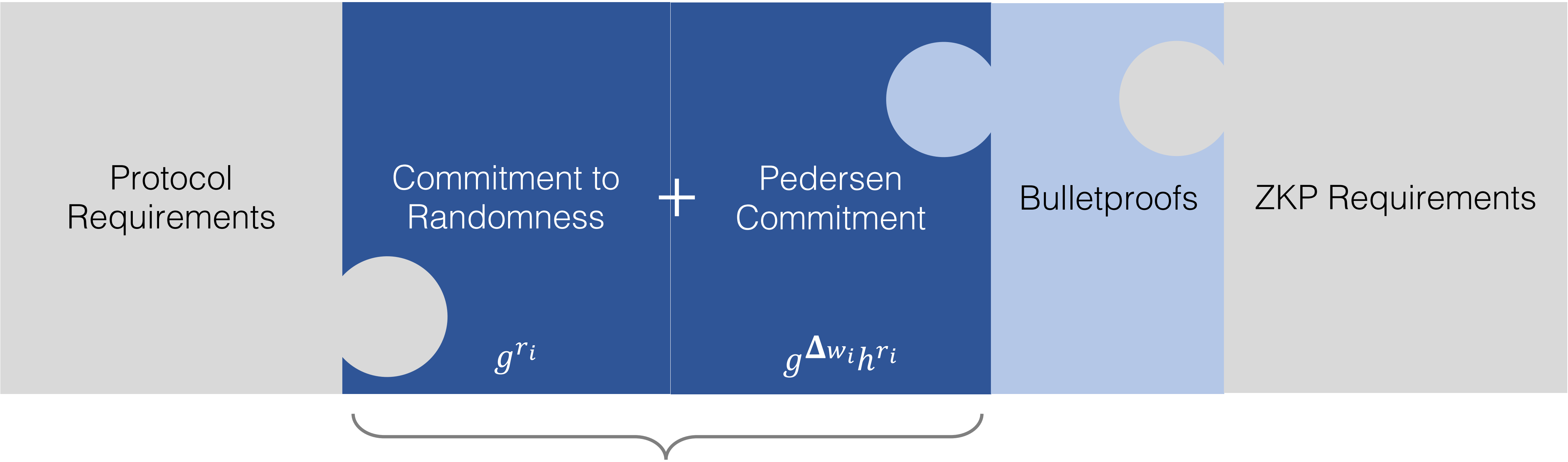
	GGPR-style zkSNARKs
Proof size	$O(1)$
Prover time	$O(\ell \log(\ell))$
Verification time	$O(1)$



Compatibility with Commitments

	GGPR-style zkSNARKs	Bulletproofs
Proof size	$O(1)$	$O(\log(\ell))$
Prover time	$O(\ell \log(\ell))$	$O(\ell)$
Verification time	$O(1)$	$O(\ell)$
Operates directly on additively homomorphic commitments	❌	✅

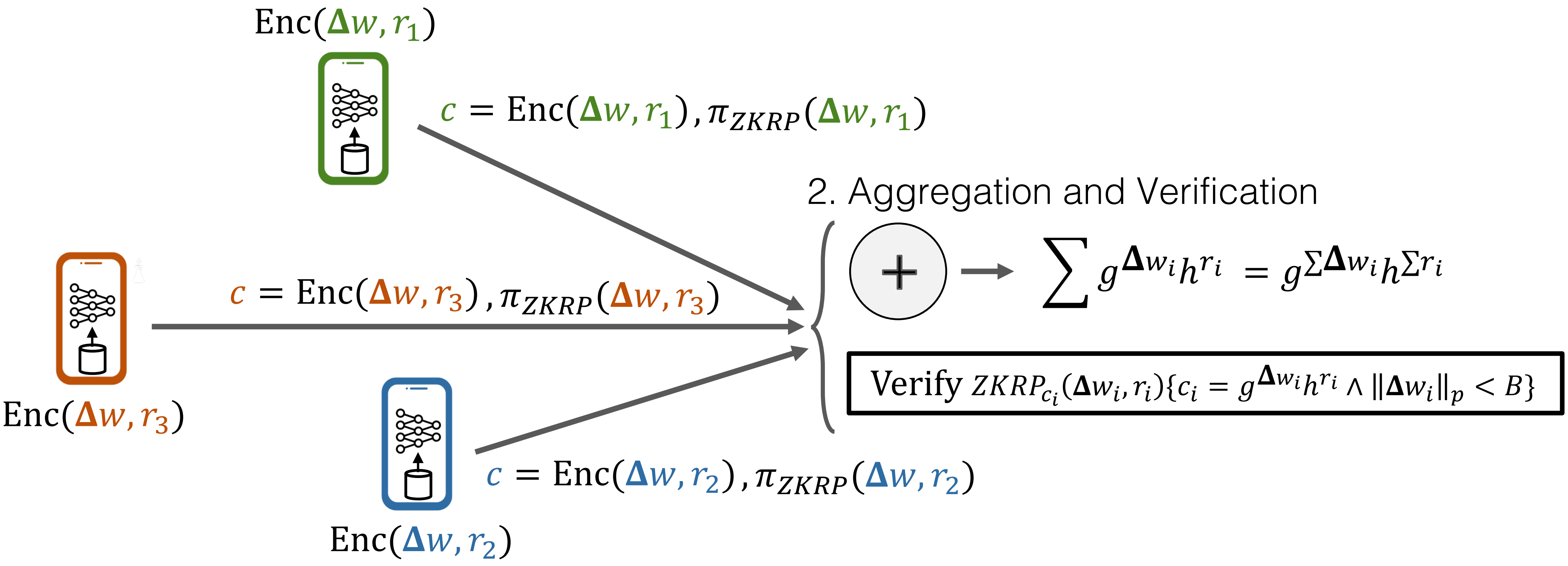
Extending Pedersen commitments for correctness



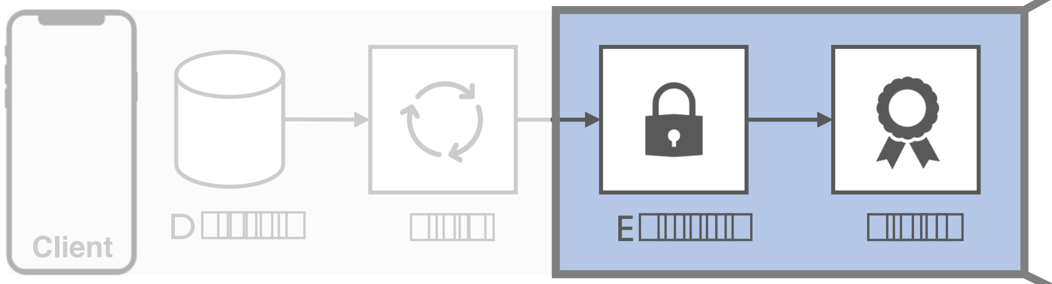
ElGamal commitment

- Server can compare $\sum g^{r_i} \leftrightarrow g^{r'}$
- Clients generate non-interactive proof-of-knowledge to proof well-formedness, i.e., r_i is the same in $(g^{\Delta w_i} h^{r_i}, g^{r_i})$

Secure Aggregation with Input Constraints



Enforcing Norm Bounds



L_∞-norm

Bounds each parameter individually

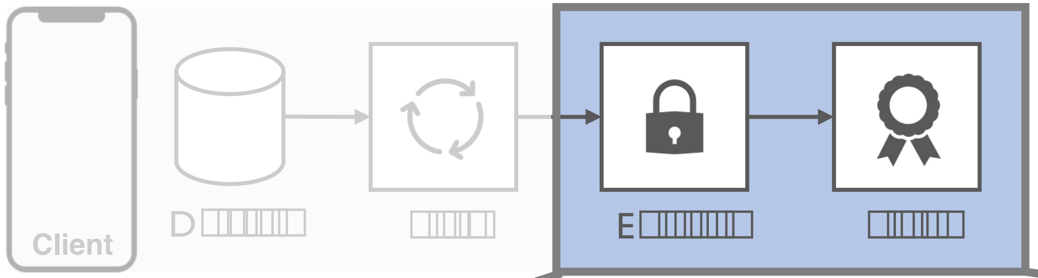
$$\begin{bmatrix} (g^{\boldsymbol{\Delta} w^1} h^{r^1}, g^{r^1}) \\ (g^{\boldsymbol{\Delta} w^2} h^{r^2}, g^{r^2}) \\ \vdots \\ (g^{\boldsymbol{\Delta} w^\ell} h^{r^\ell}, g^{r^\ell}) \end{bmatrix}$$

ElGamal commitments

$$\begin{bmatrix} \pi_{bprp}(\boldsymbol{\Delta} w^1, r^1) \\ \pi_{bprp}(\boldsymbol{\Delta} w^2, r^2) \\ \vdots \\ \pi_{bprp}(\boldsymbol{\Delta} w^\ell, r^\ell) \end{bmatrix}$$

Bulletproof Range Proofs

Enforcing Norm Bounds



L_2-norm

Bounds the sum of squared parameters

L_∞-norm

Bounds each parameter individually

$$\begin{bmatrix} (g^{\Delta w^1} h^{r^1}, g^{r^1}) \\ (g^{\Delta w^2} h^{r^2}, g^{r^2}) \\ \vdots \\ (g^{\Delta w^\ell} h^{r^\ell}, g^{r^\ell}) \end{bmatrix} \overset{\pi}{\Longleftrightarrow} \begin{bmatrix} g^{(\Delta w^1)^2} h^{r'^1} \\ g^{(\Delta w^2)^2} h^{r'^2} \\ \vdots \\ g^{(\Delta w^\ell)^2} h^{r'^\ell} \end{bmatrix} \qquad \begin{bmatrix} \pi_{bprp}(\Delta w^1, r^1) \\ \pi_{bprp}(\Delta w^2, r^2) \\ \vdots \\ \pi_{bprp}(\Delta w^\ell, r^\ell) \end{bmatrix} \qquad \boxed{\pi_{bprp}(\sum_{i=1}^{\ell} (\Delta w^i)^2)}$$

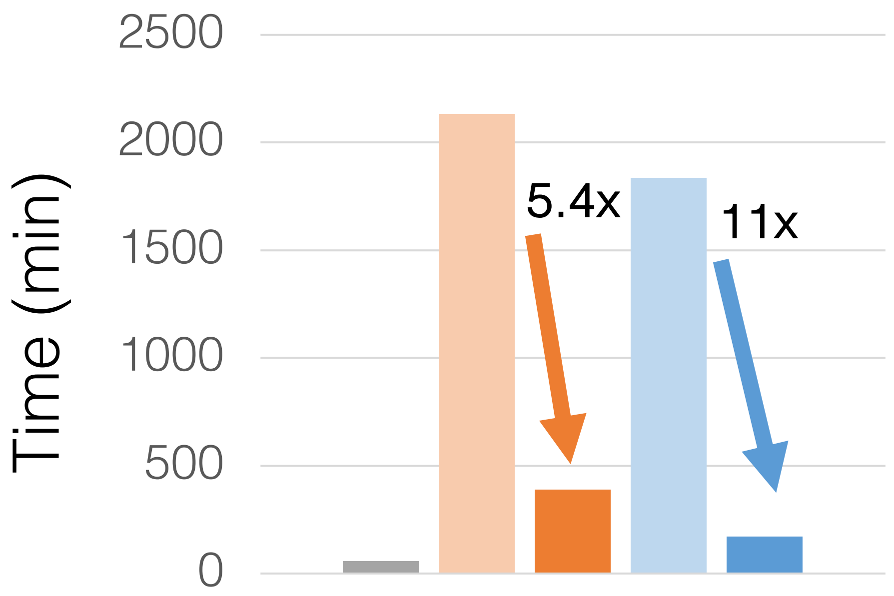
ElGamal commitments | Squared commitments | Bulletproof Range Proofs | Squared Range Proof

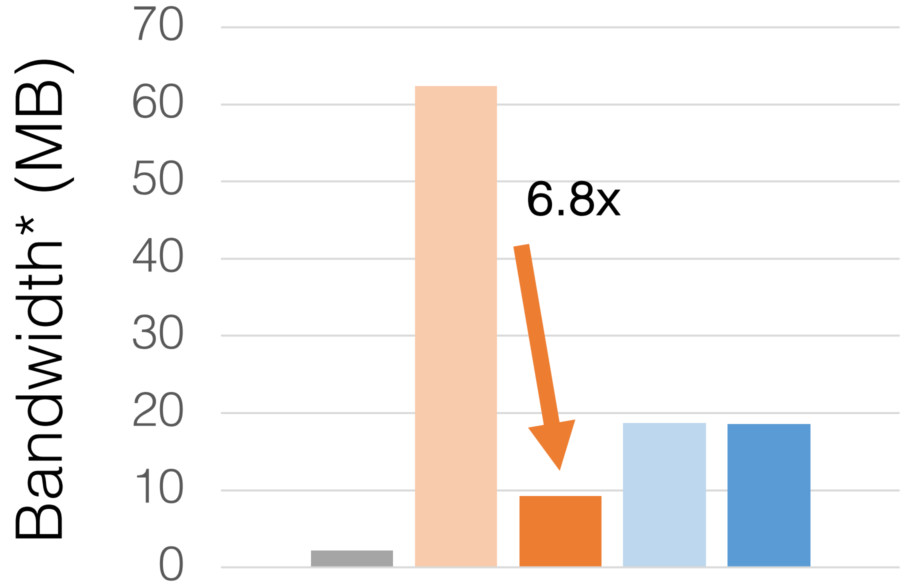
RoFL: End-To-End Performance

CIFAR-10 Model 273k Parameters

Setup: 48 Clients, 160 rounds

	SecAgg	L_2	L_2 Optimized	L_∞	L_∞ Optimized
Accuracy:	0.86	0.85	0.82	0.85	0.85





* Per client per round

RoFL: End-To-End Performance

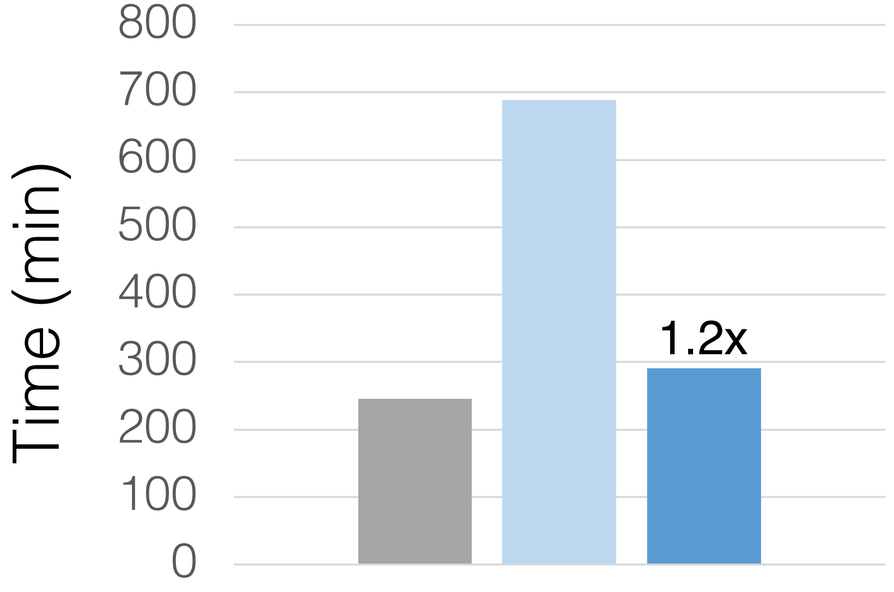
Shakespeare Model 818k Parameters

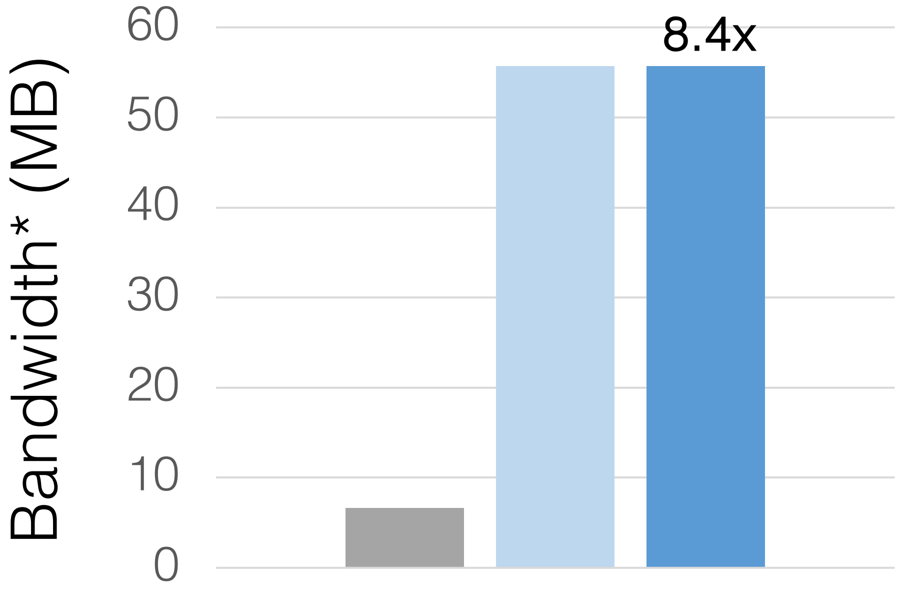
Setup: 48 Clients, 20 rounds

SecAgg — Accuracy: 0.57

L_∞ — 0.57

L_∞ Optimized — 0.57





* Per client per round

ETH zürich

PPS

This work:

- Understanding FL Robustness
- RoFL: Secure Aggregation with Private Input Validation

Future work:

- Exploring additional client constraints for robustness
- Protocols with better bandwidth overhead
- Efficient ZKPs for resource-constrained provers

pps-lab/fl-analysis

pps-lab/rofl-project-code

pps-lab.com/research/ml-sec


RoFL: Attestable Robustness for Secure Federated Learning

Lukas Burkhalter, Hidde Lycklama, Alexander Viand, Nicolas Küchler, Anwar Hithnawi

ETH Zürich

arXiv:2107.03311v2 [cs.CR] 19 Jul 2021

Abstract

1 Introduction