RoFL: Robustness of Secure Federated Learning

Hidde Lycklama*

Lukas Burkhalter*

Alexander Viand

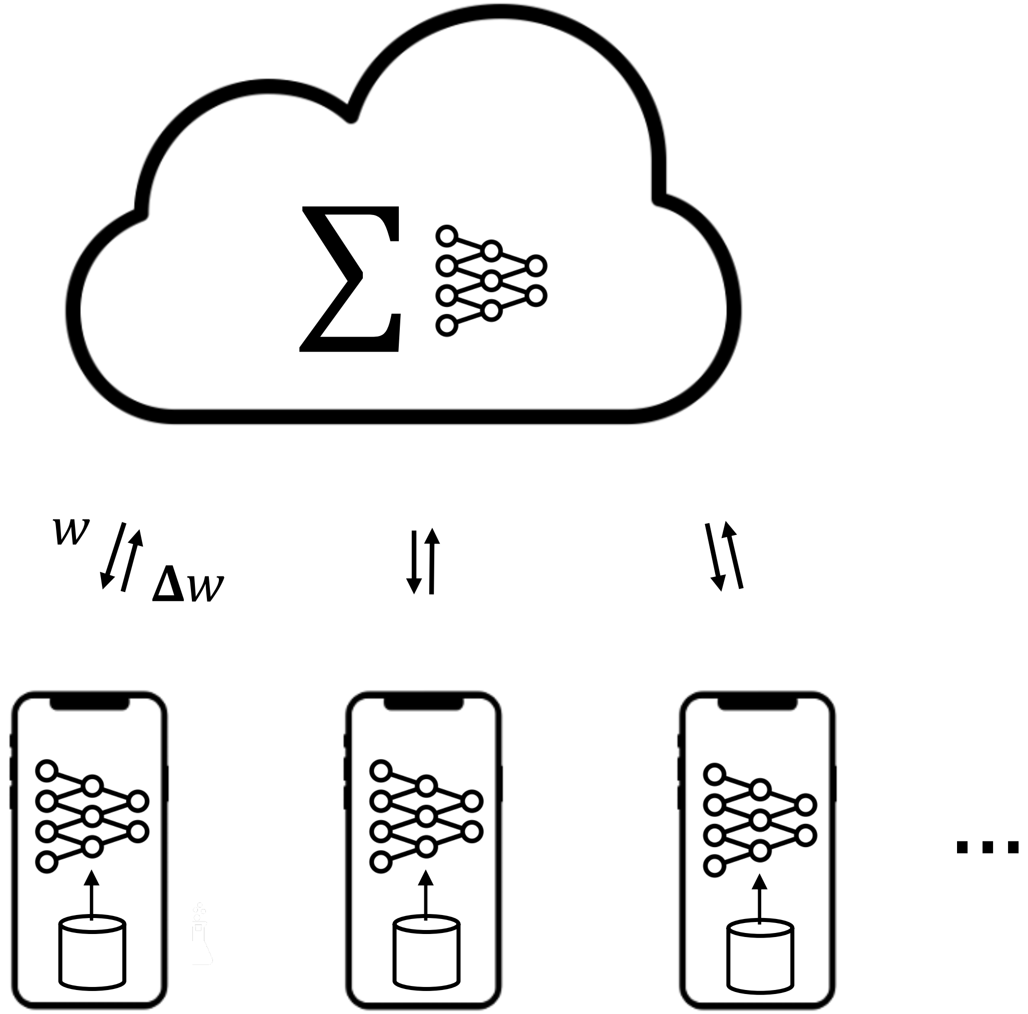
Nicolas Küchler

Anwar Hithnawi

ETH zürich

IEEE Security & Privacy '23

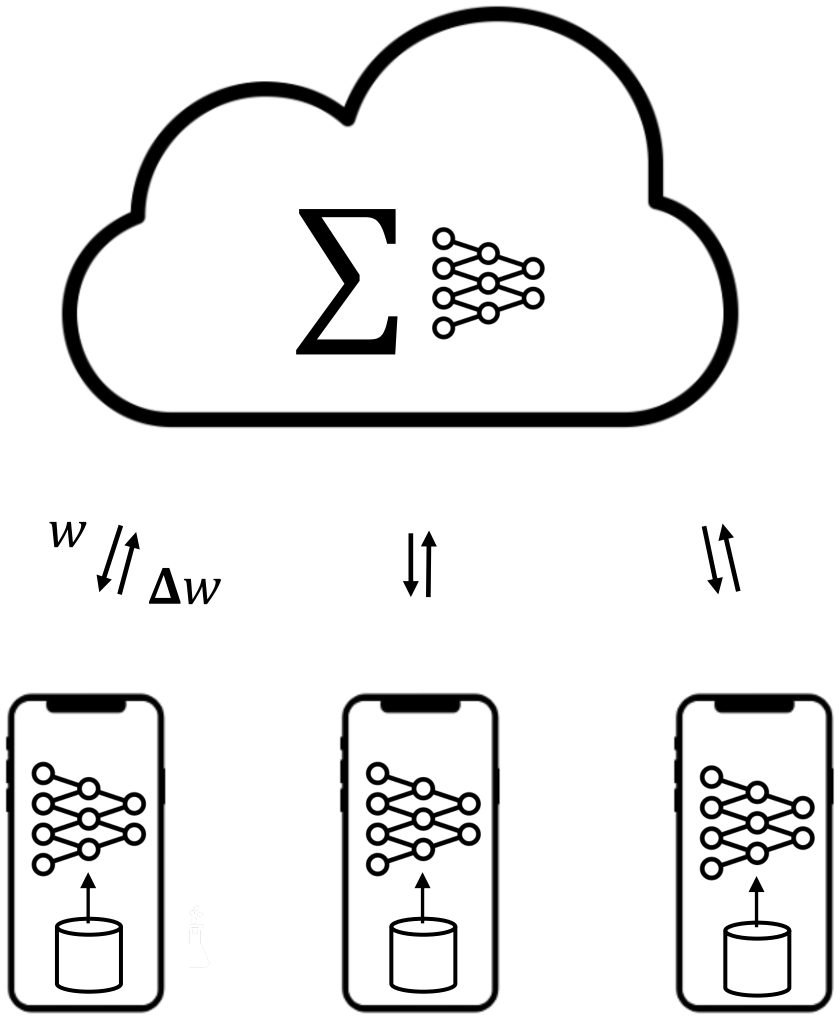
Federated Learning



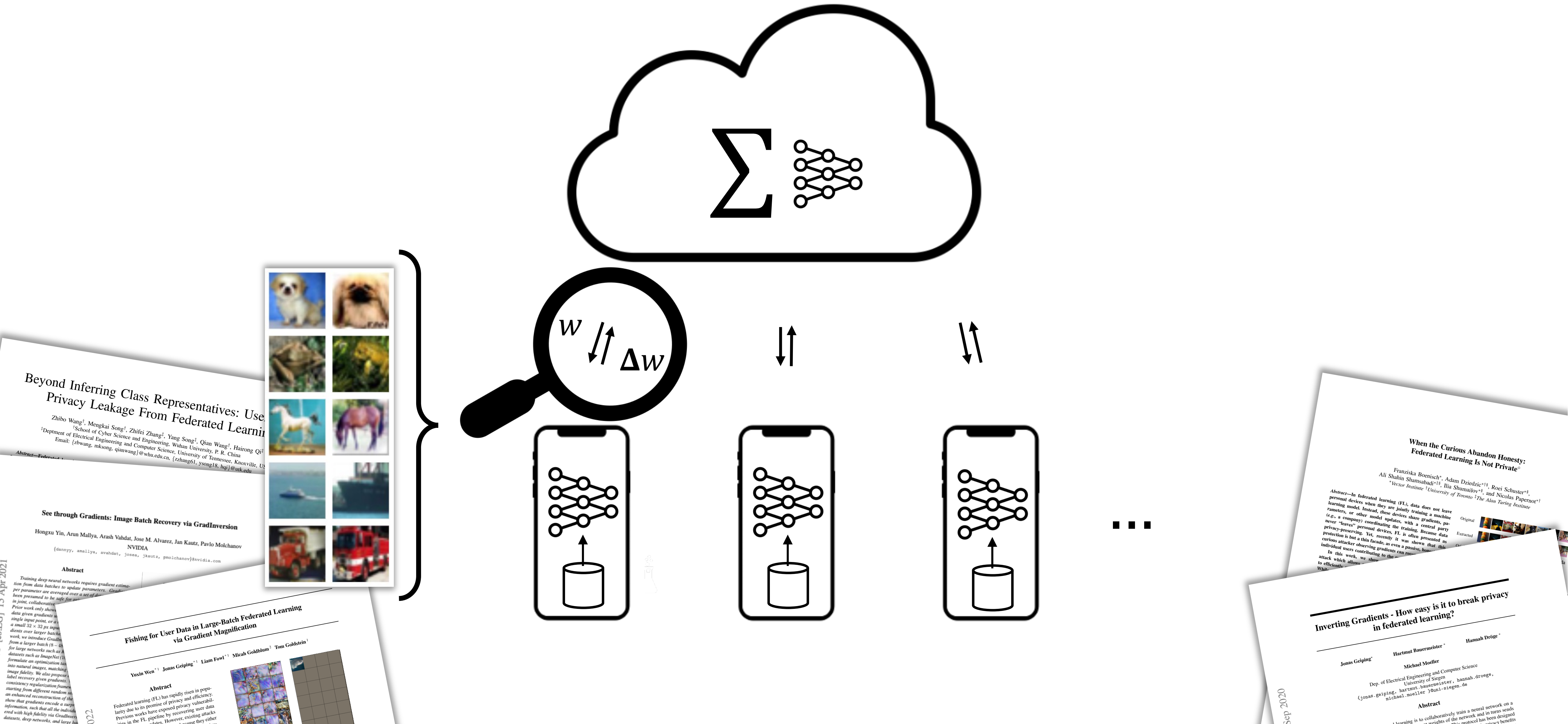
Federated Learning

Purpose Limitation

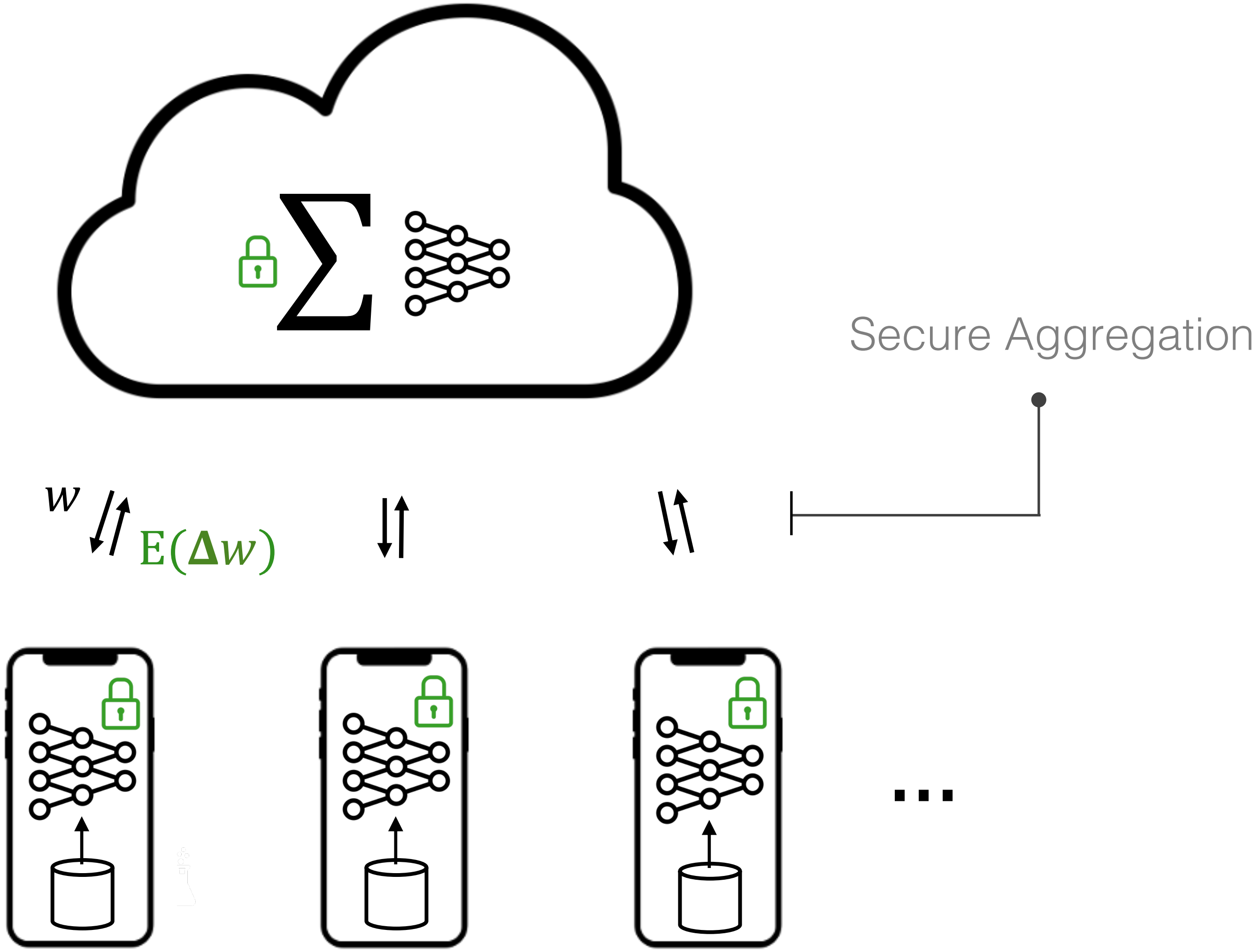
Data Minimization



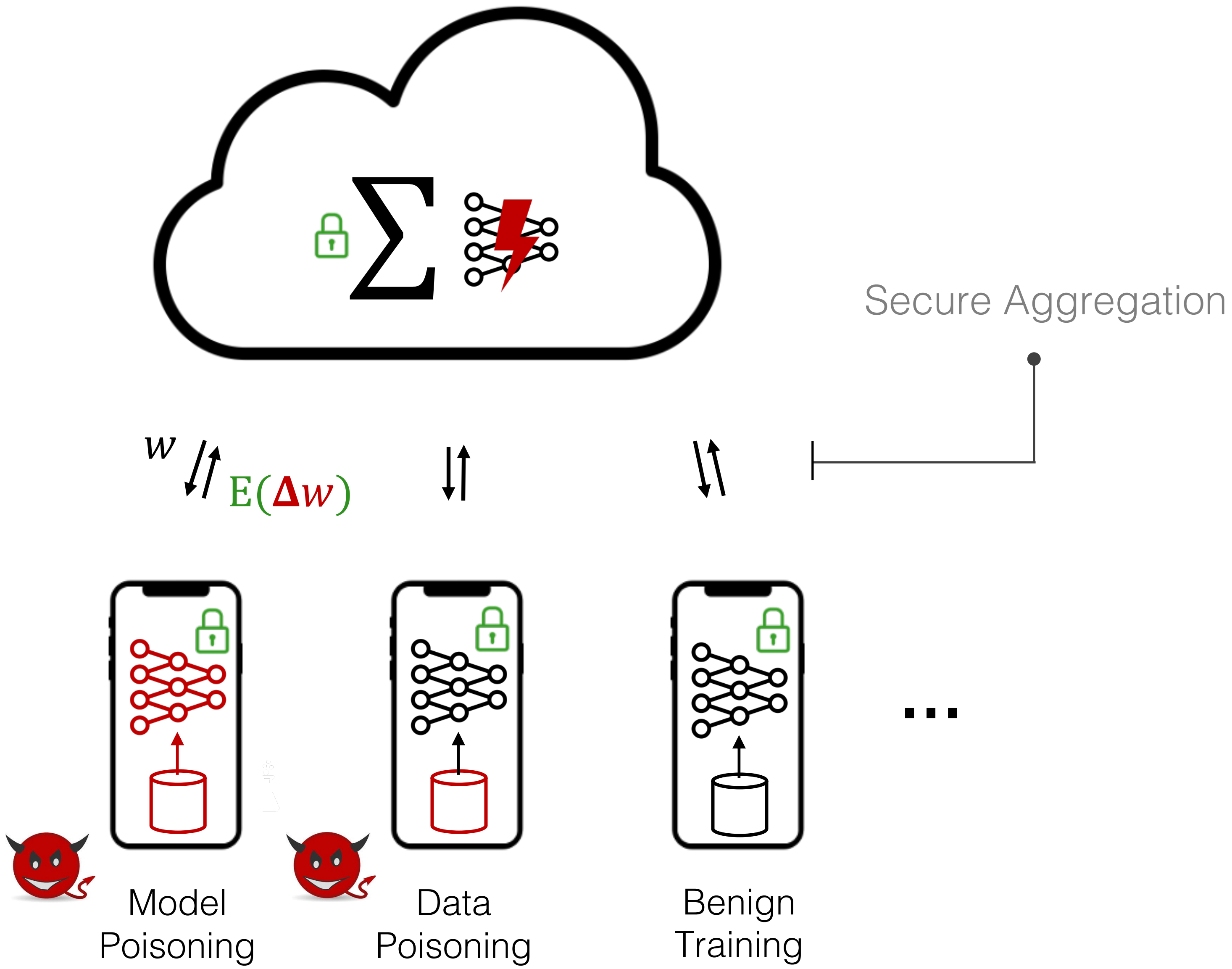
Federated Learning: Input Privacy



Secure Federated Learning



Malicious Clients

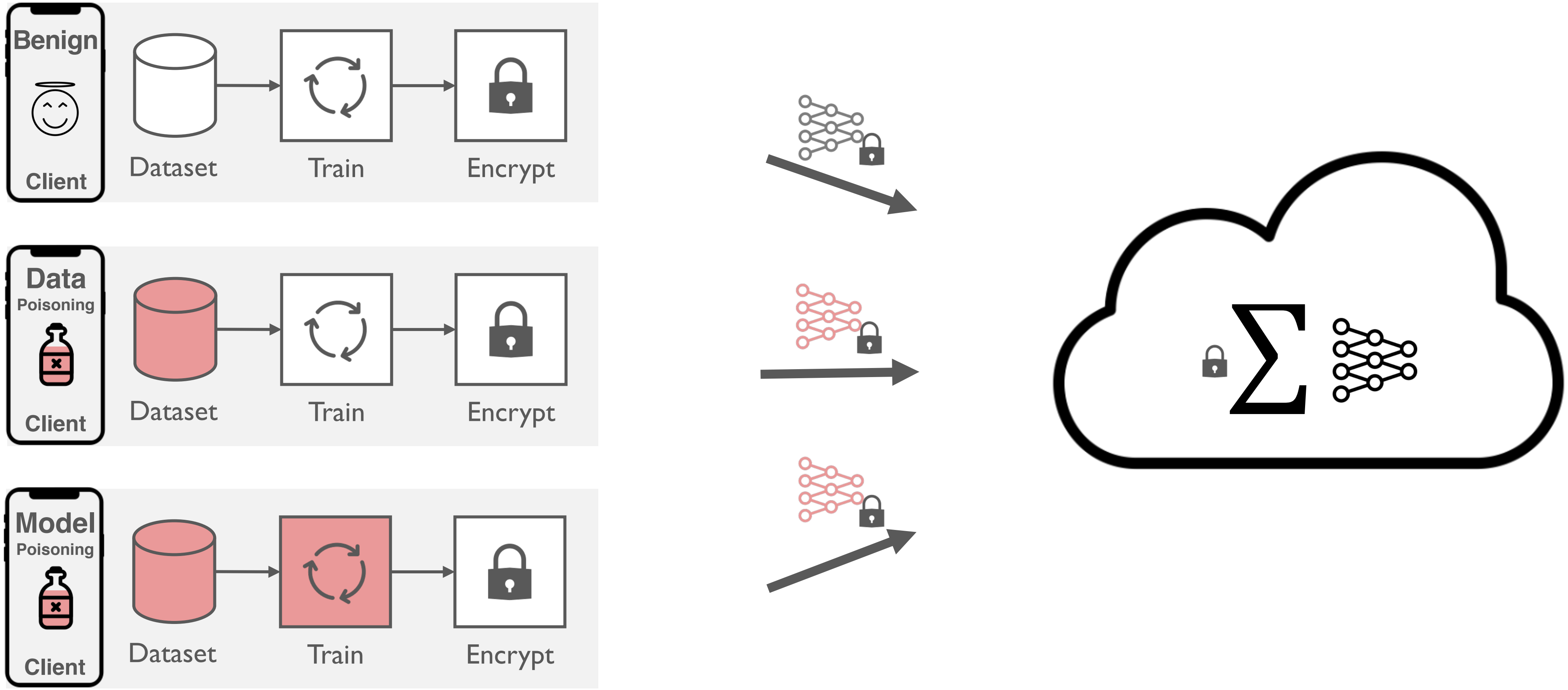


RoFL: Robustness of Secure Federated Learning

Understand
Vulnerabilities in FL

Cryptographically
Enforce Constraints

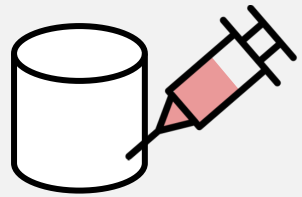
Adversarial Clients



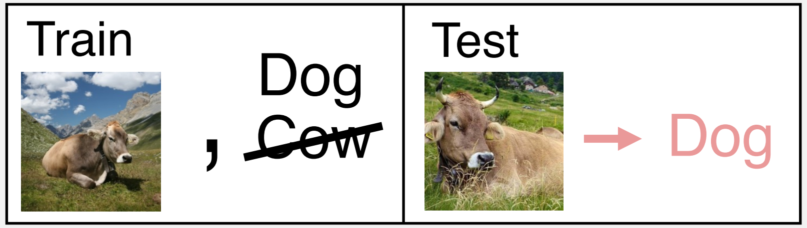
Adversarial Clients

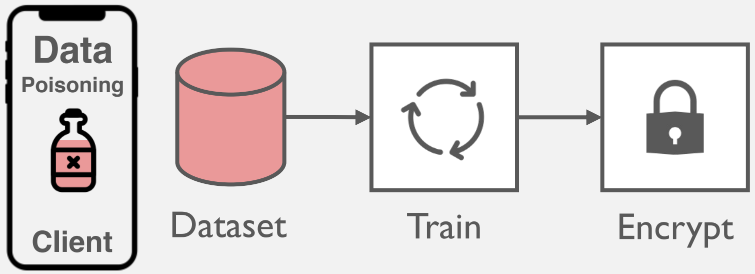
Data Poisoning

adversary controls training data



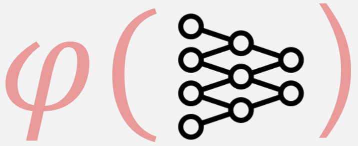
Example: Label Flipping



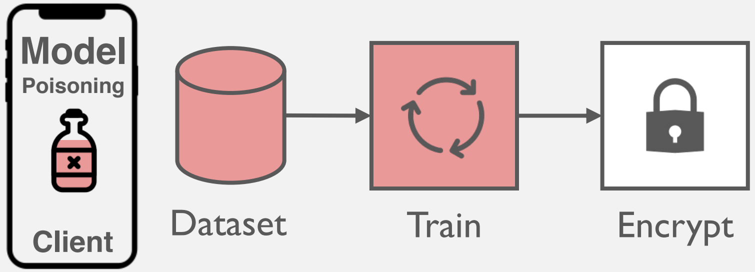


Model Poisoning

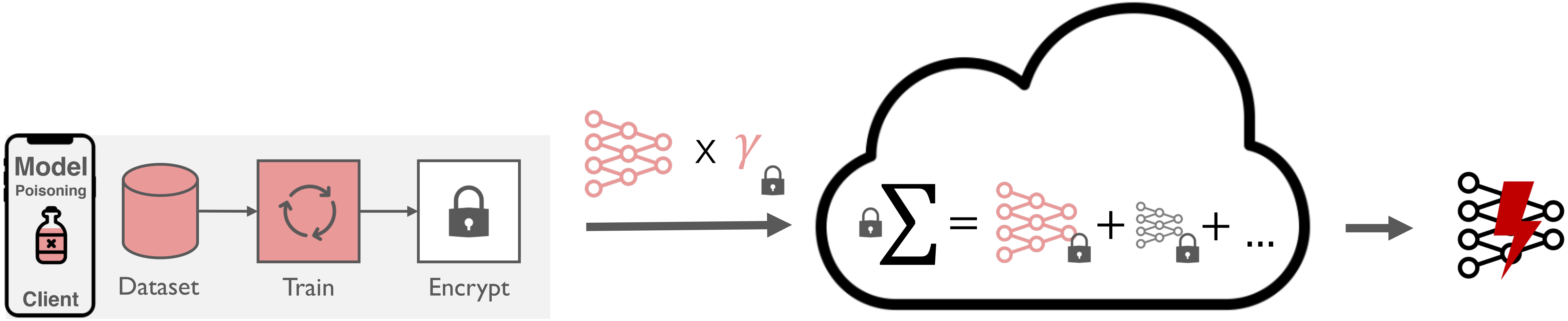
adversary controls model updates



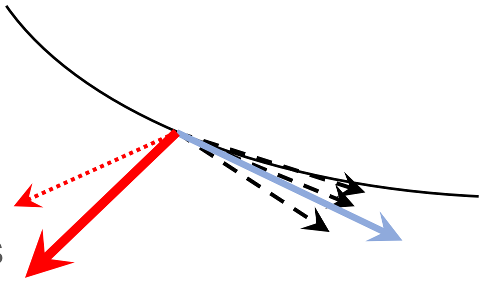
Example: Model Replacement



Adversarial Clients



Problem: Linear aggregation rules are vulnerable to Byzantine behavior

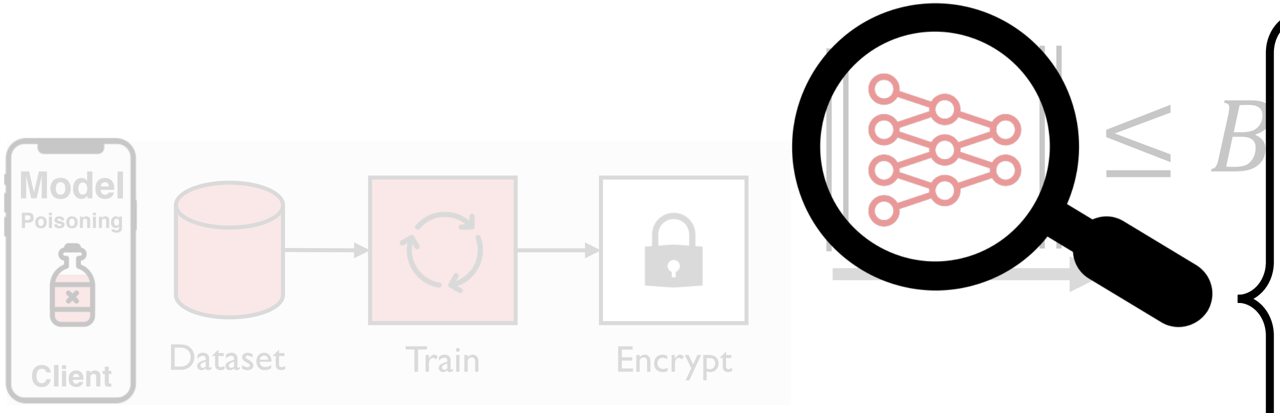


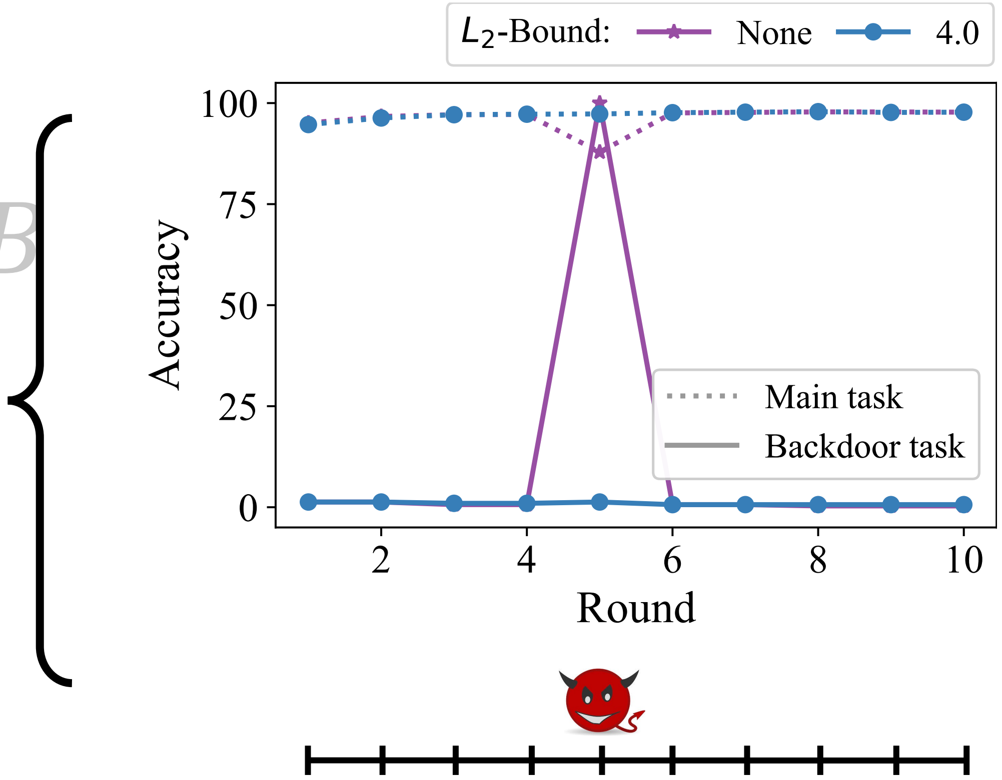
Machine Learning: Byzantine-Robust Distributed Learning	**Security:** Private Data-Collection Systems
- Krum [Blanchard et al. NeurIPS'17] - Trimmed Mean [Yin et al. ICML'18] - Coordinate-wise Median [Yin et al. ICML'18] - Bulyan [Mhamdi et al. ICML'18] - ByzantineSGD [Alistarh et al. NeurIPS'18] - Redundant 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

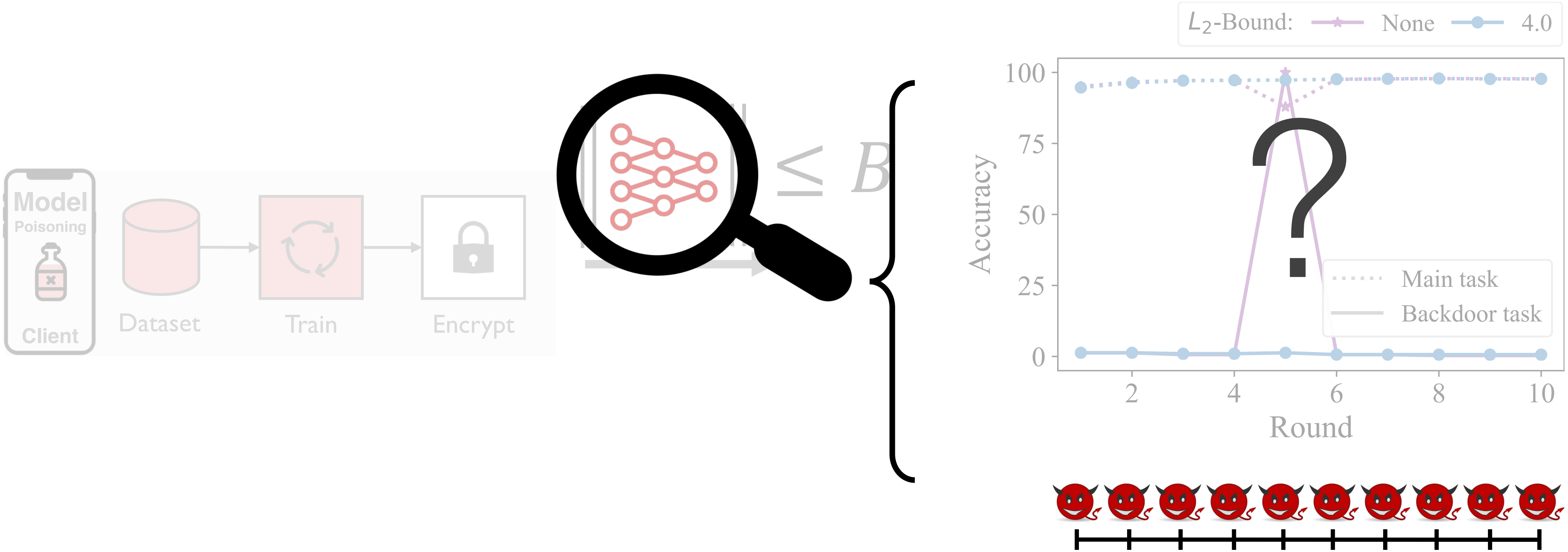
A Well-Formed Client Submission in Federated Learning

Norm bound


Model
Poisoning
Client
Dataset
Train
Encrypt
$\leq B$


L_2-Bound: None 4.0
Accuracy
Round
Main task
Backdoor task
Single-shot Attacker (round 5)

Norm bound



Continuous Attacker

Is the norm bound actually effective?

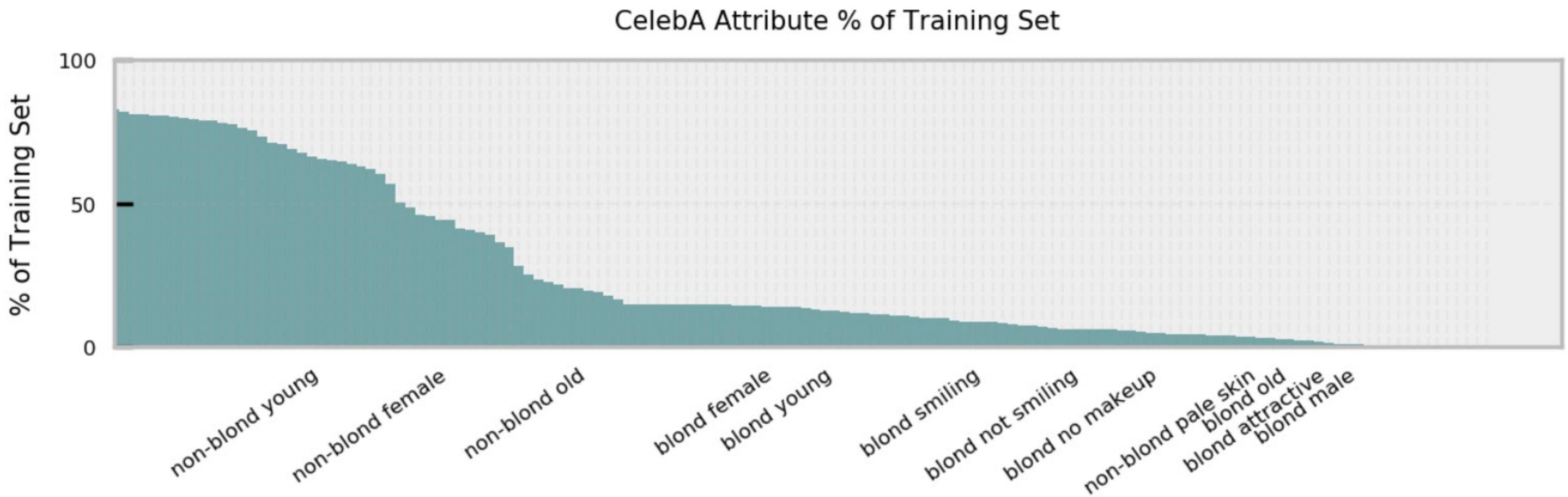
How To Backdoor Federated Learning

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

Long Tail …

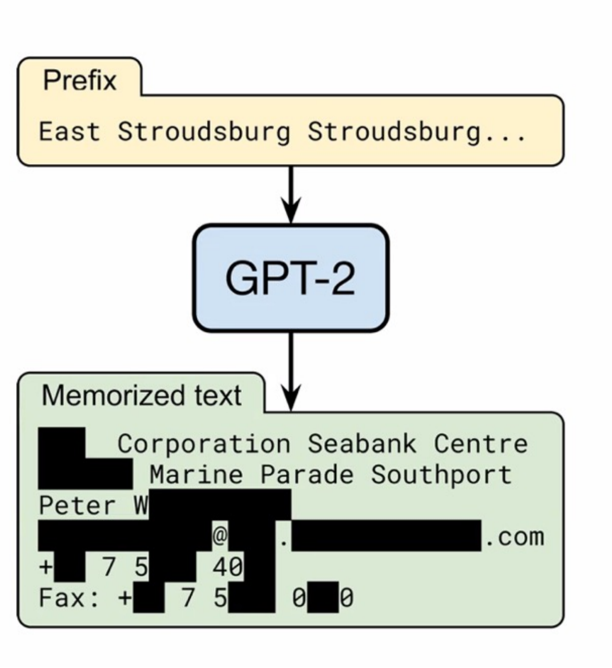


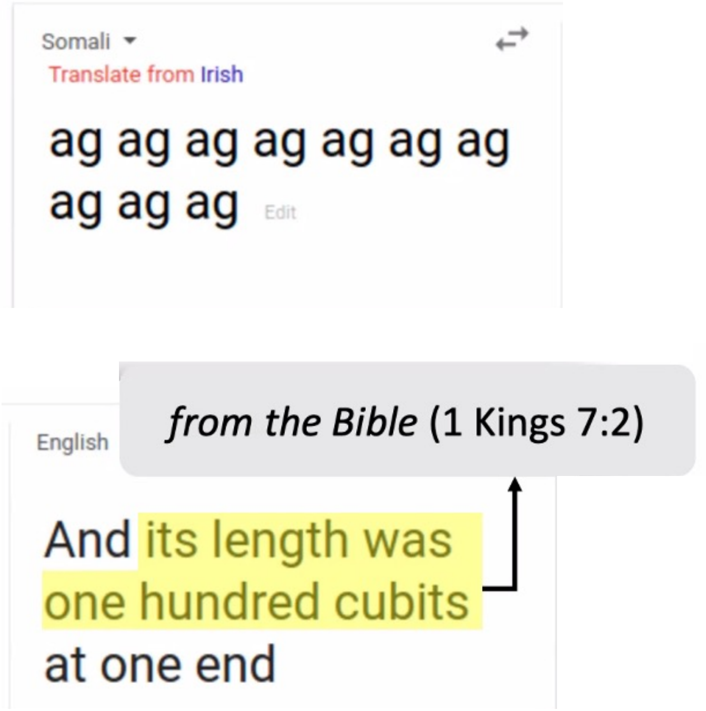
Prototypical

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.

Analysis: Understanding FL Robustness

Adaptive attacks

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

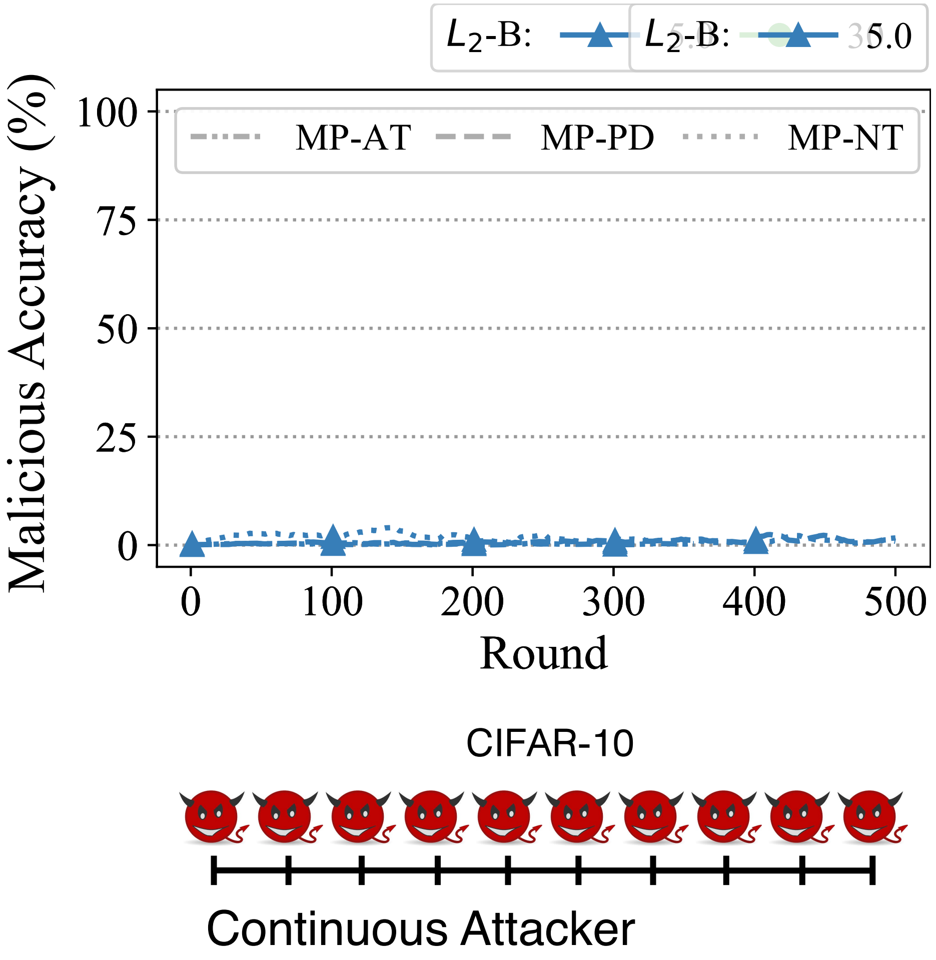
MP-NT: Neurotoxin [Zhang et al., ICML'22]

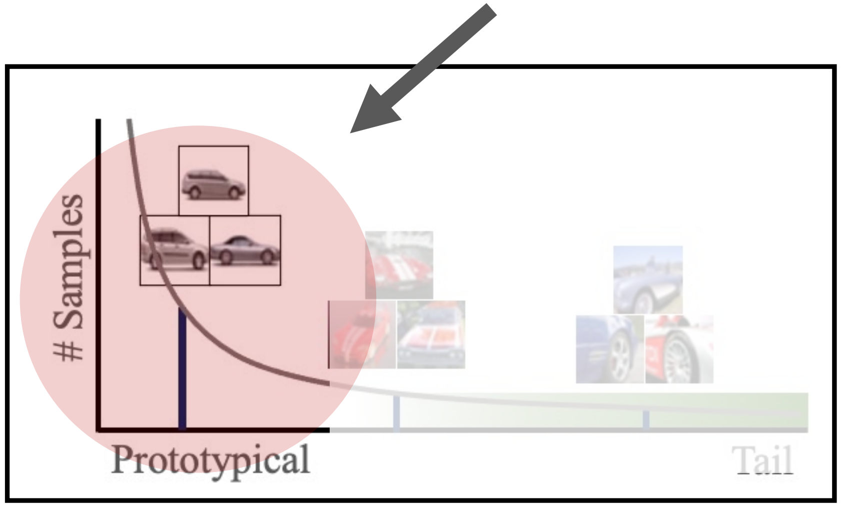
MP-AT: Anticipate [Wen et al., AdvML@ICML'22]

Considered:

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

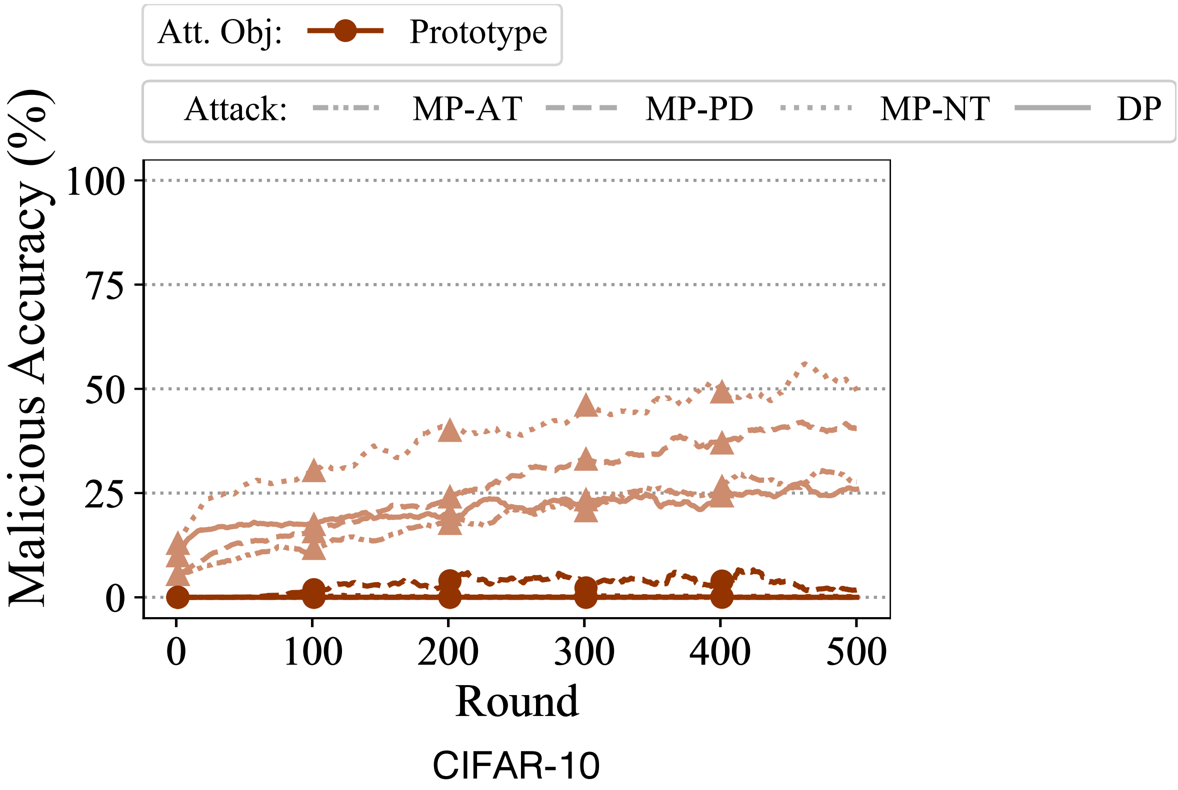
Impact of Attack Objective on Backdoor Attacks

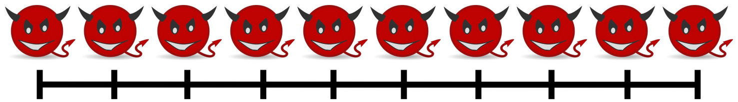




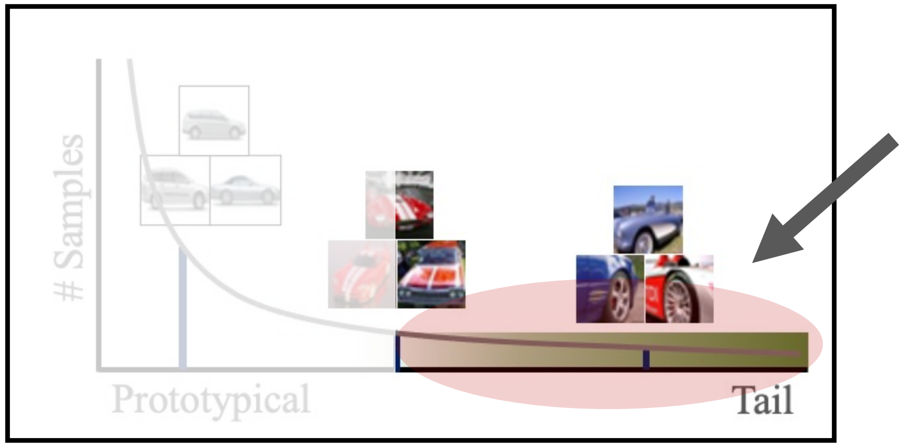
Prototypical Targets

Impact of Attack Objective on Backdoor Attacks



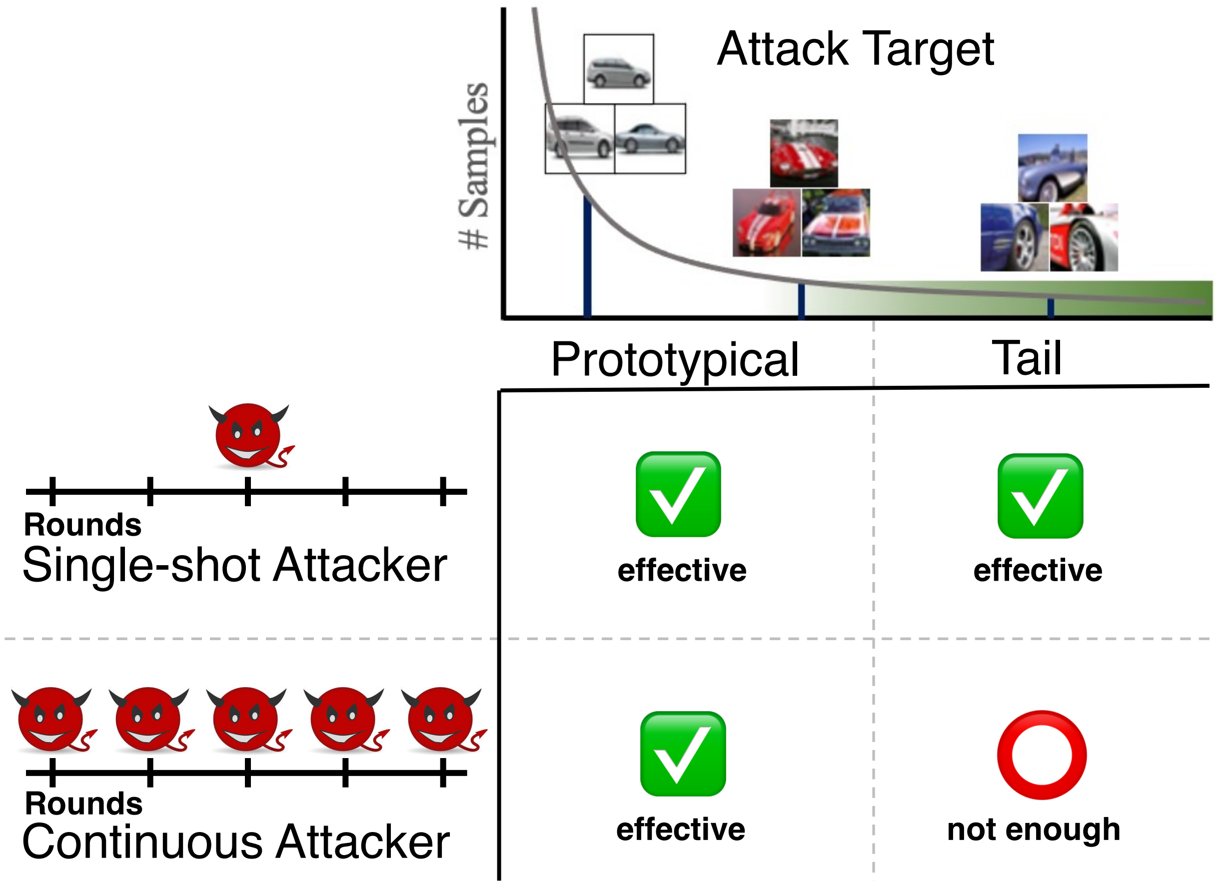


Continuous Attacker



Tail Targets

Norm Bound Provides Practical Robustness Guarantees



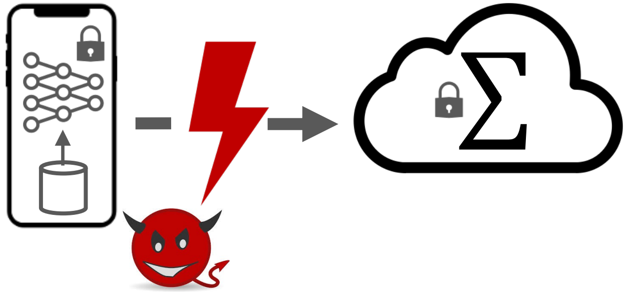
… requires a strong attacker that is consistently selected and targets a tail sample

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

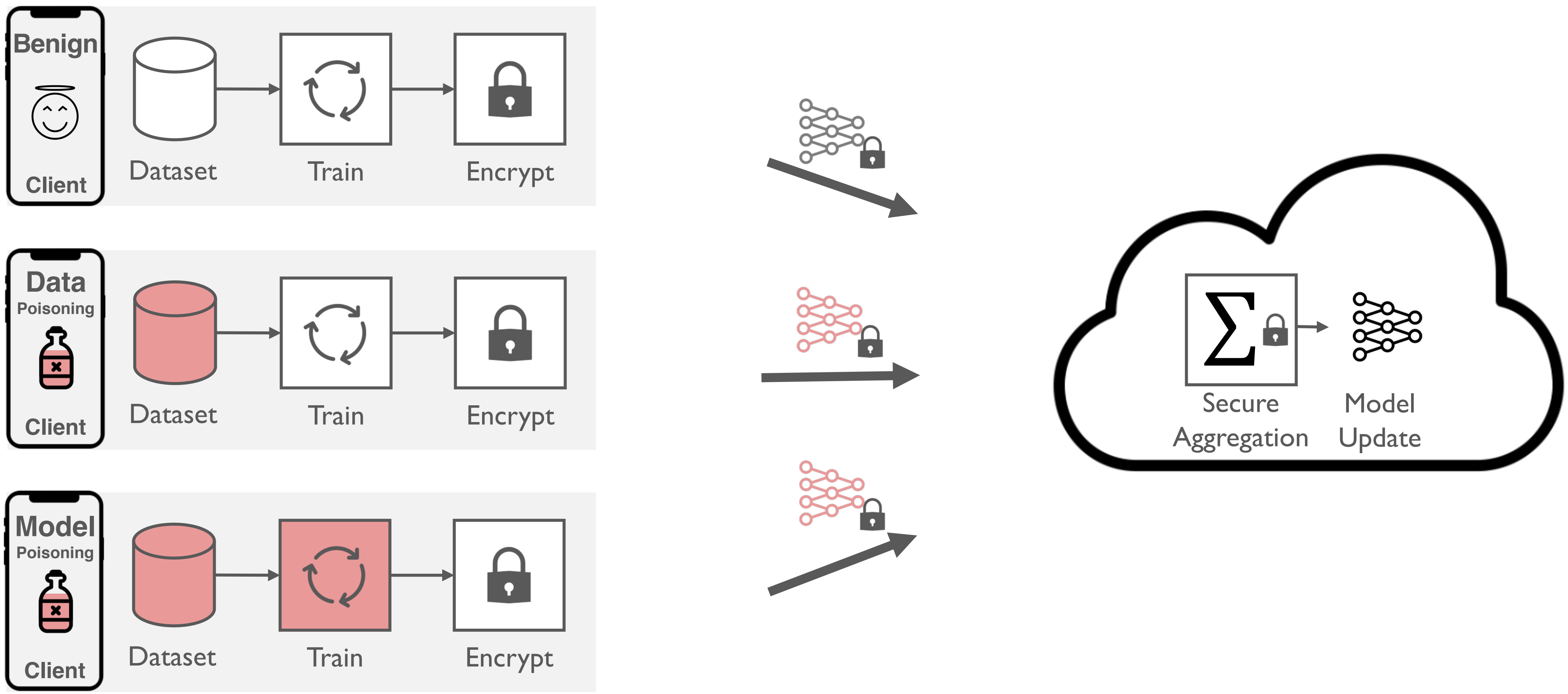


$$\left\{ \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_d \end{bmatrix} \right\}$$

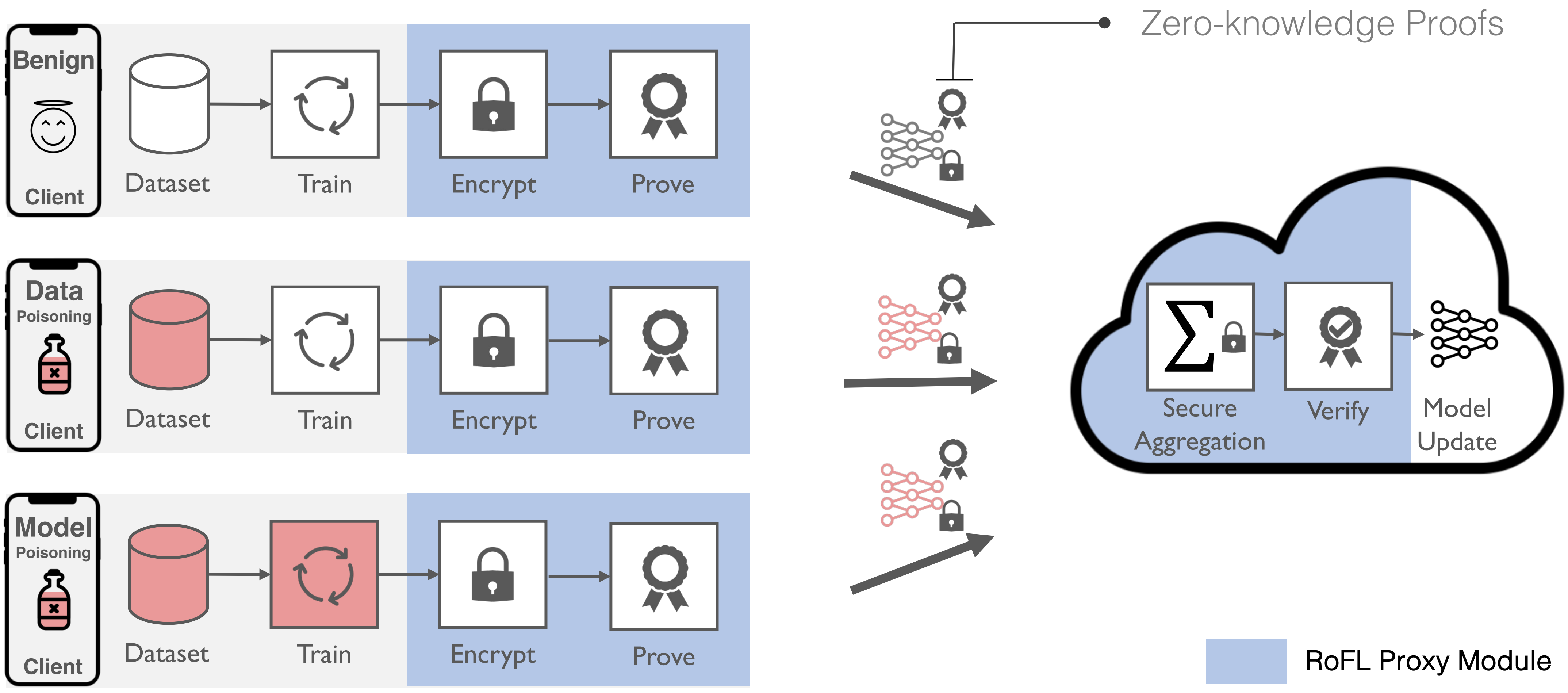
- Compressed Sigma protocols
- Optimistic continuation
- Probabilistic checking
- Subspace learning

Correctness	Private Input Validation	Optimizations

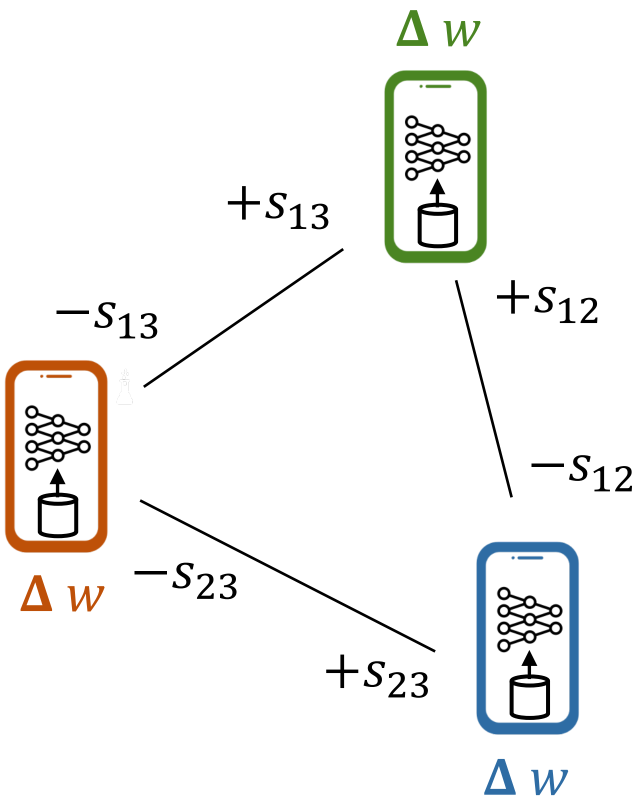
Secure Federated Learning



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}

$$r_1 + r_2 + r_3 = 0$$

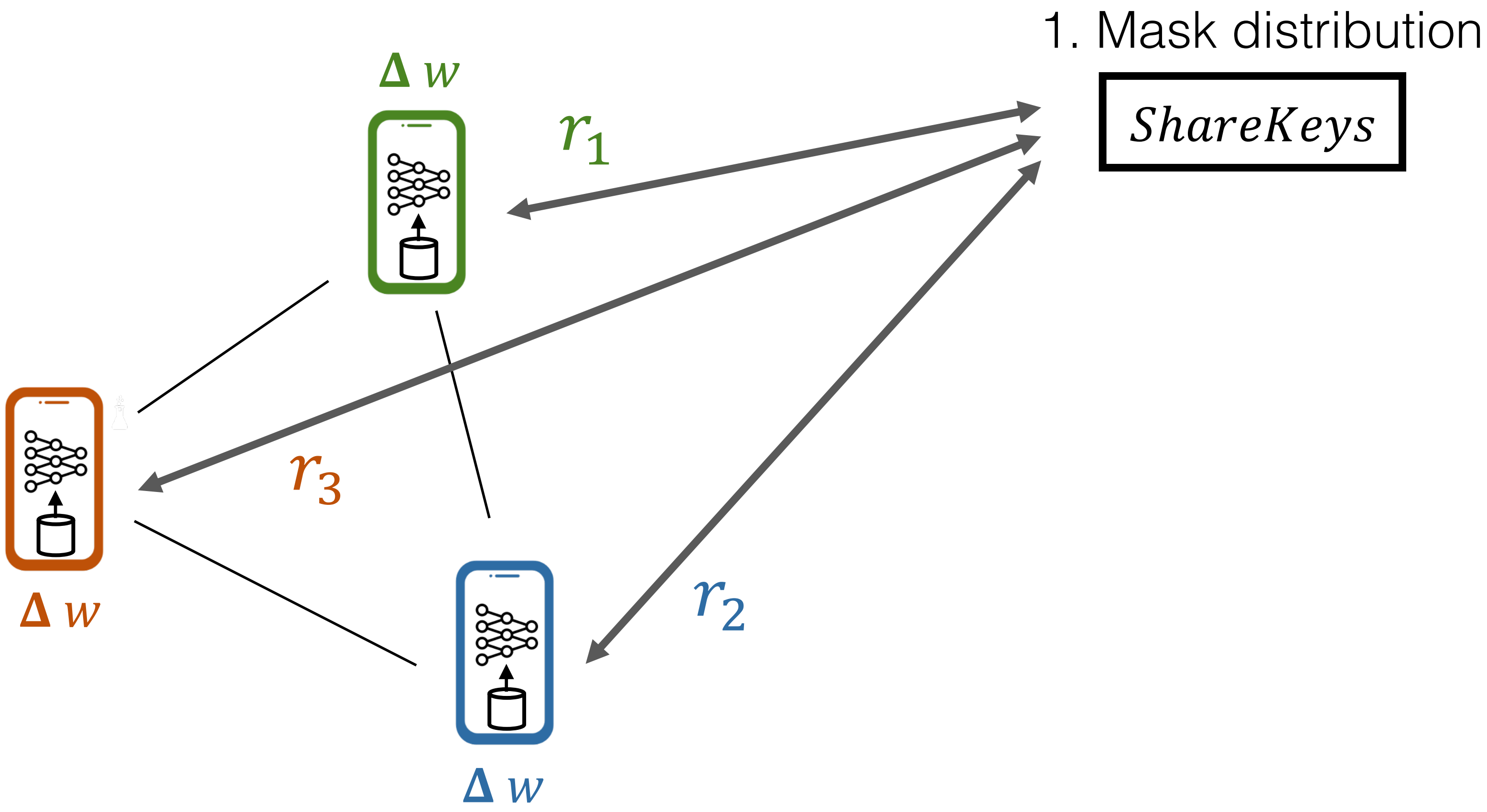
where

$r_1 = s_{12} + s_{13}$
$r_2 = -s_{12} + s_{23}$
$r_3 = -s_{13} - s_{23}$

+: modular addition

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

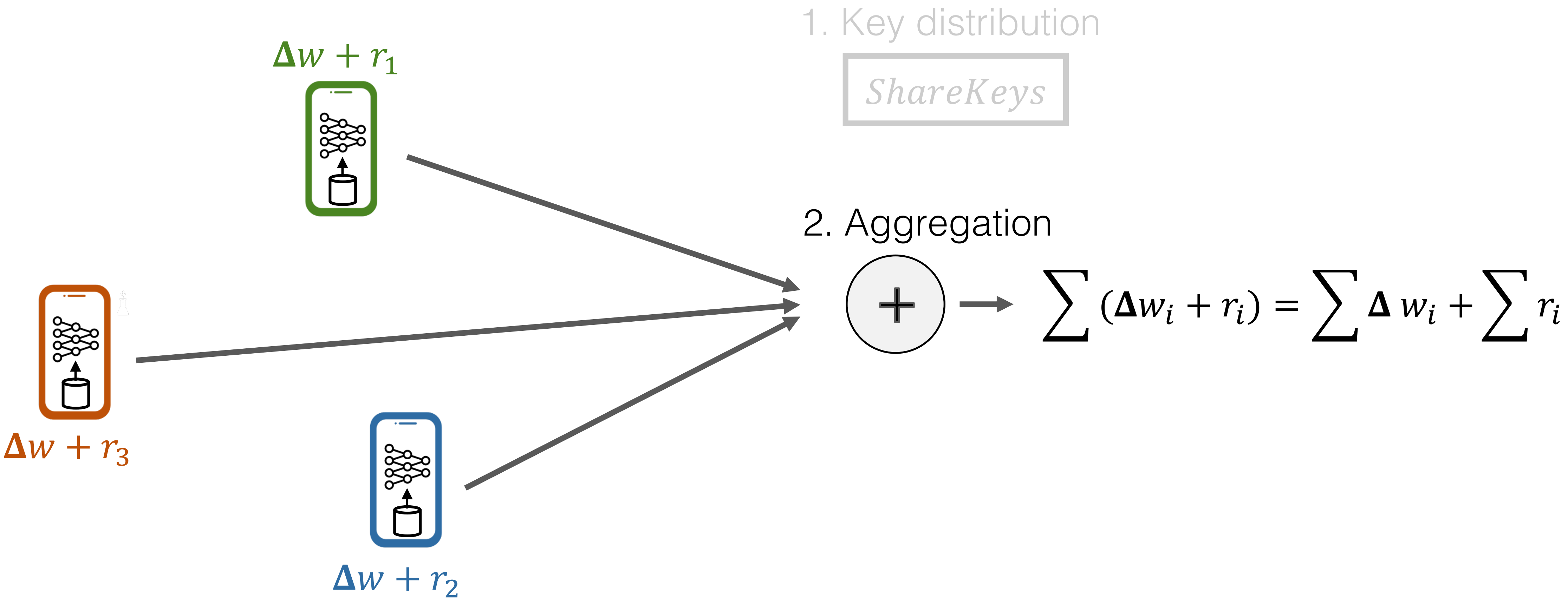
Secure Aggregation



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

+: modular addition

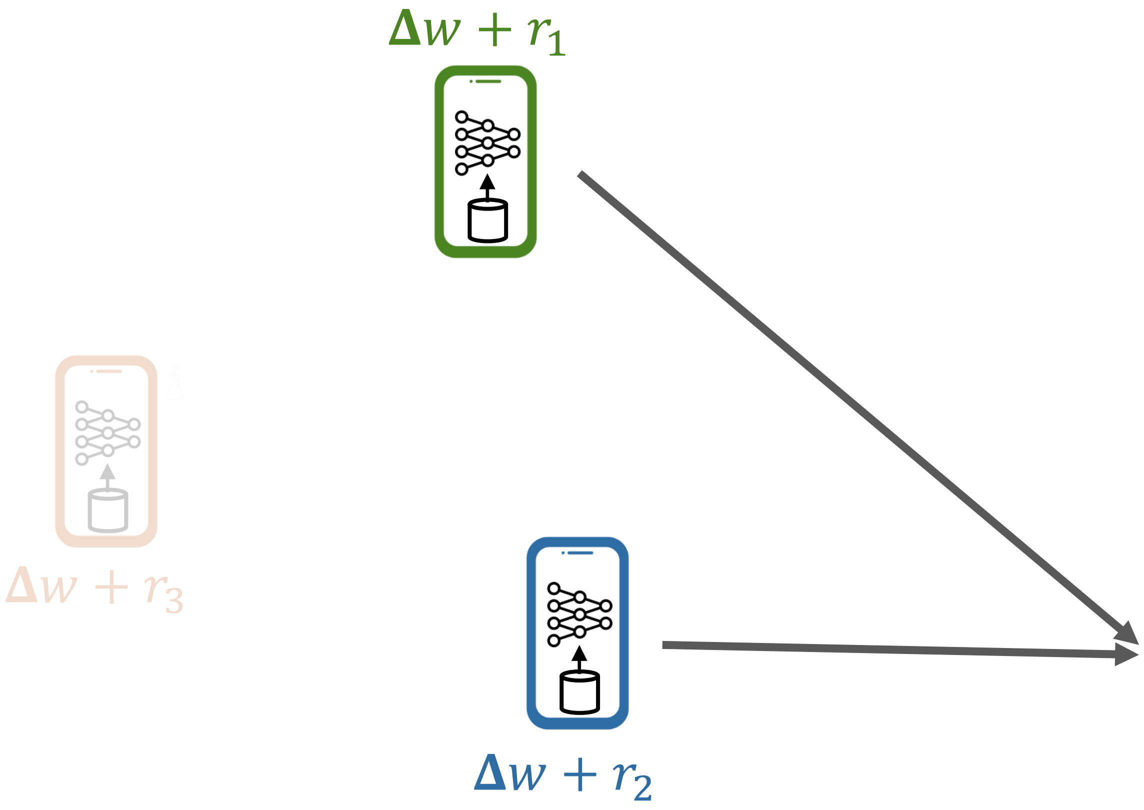
Secure Aggregation



+: 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

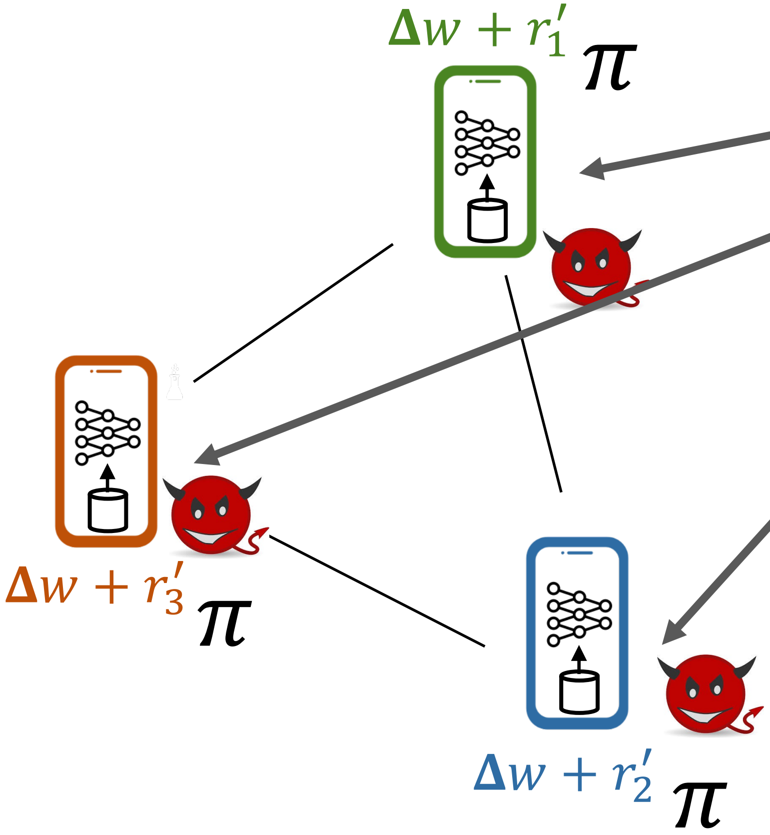
$\boxed{UnMask} \rightarrow r'$

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

+: 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

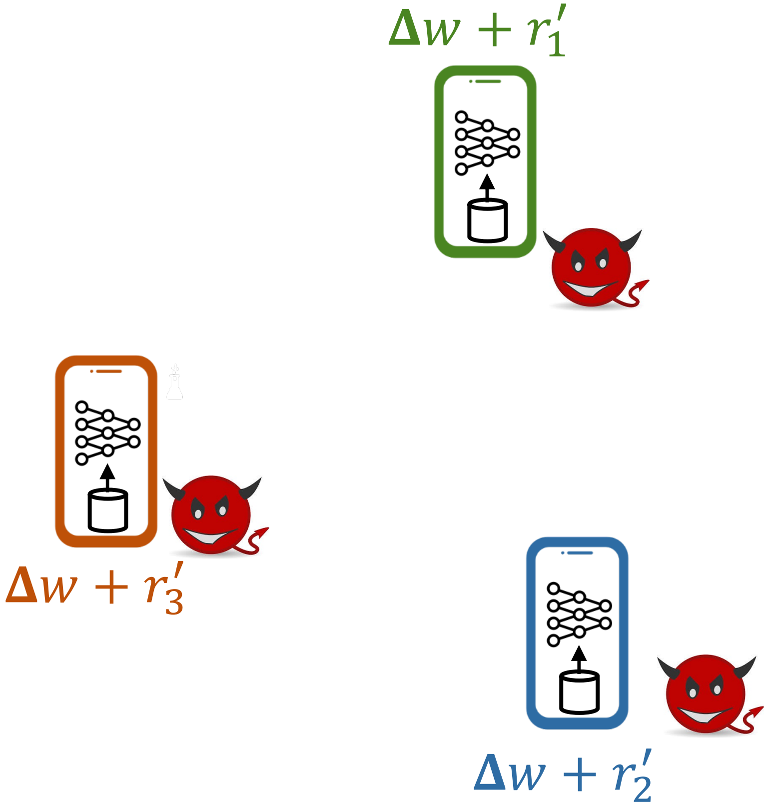
$+ \rightarrow \sum(\Delta w_i + r_i) = \sum \Delta w_i + \sum r_i$

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

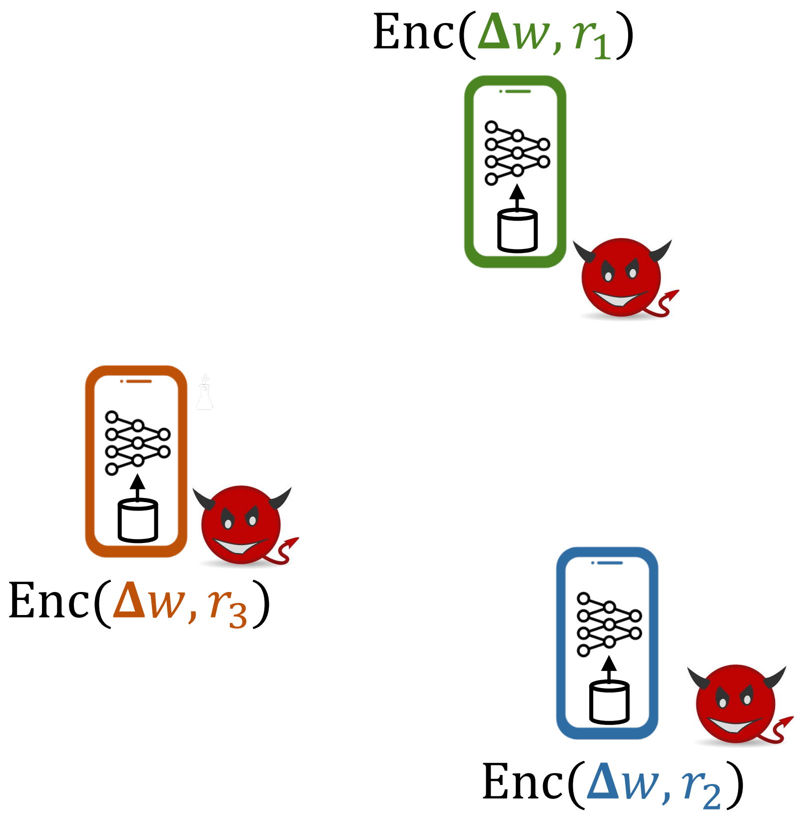
$$+ \rightarrow \sum (\Delta w_i + r_i) = \sum \Delta w_i + \sum r_i$$

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

$\boxed{\mathit{ShareKeys}}$

2. Aggregation

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

3. Reconstruction

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

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

Efficiency hinges on compatibility with zero-knowledge proofs

Protocol Requirements

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

Homomorphic in messages and keys

Correctness check

Additively Homomorphic Commitments

ZKP Requirements

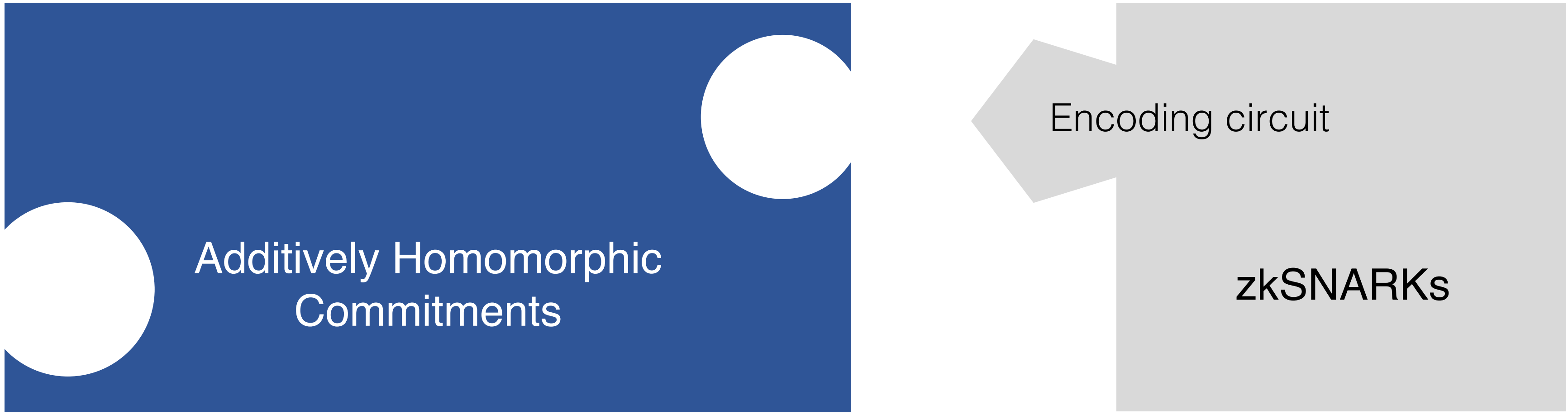
$$\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_\ell \end{bmatrix}$$

Range proofs over large vectors

Resource-constrained devices

Compatibility with Commitments

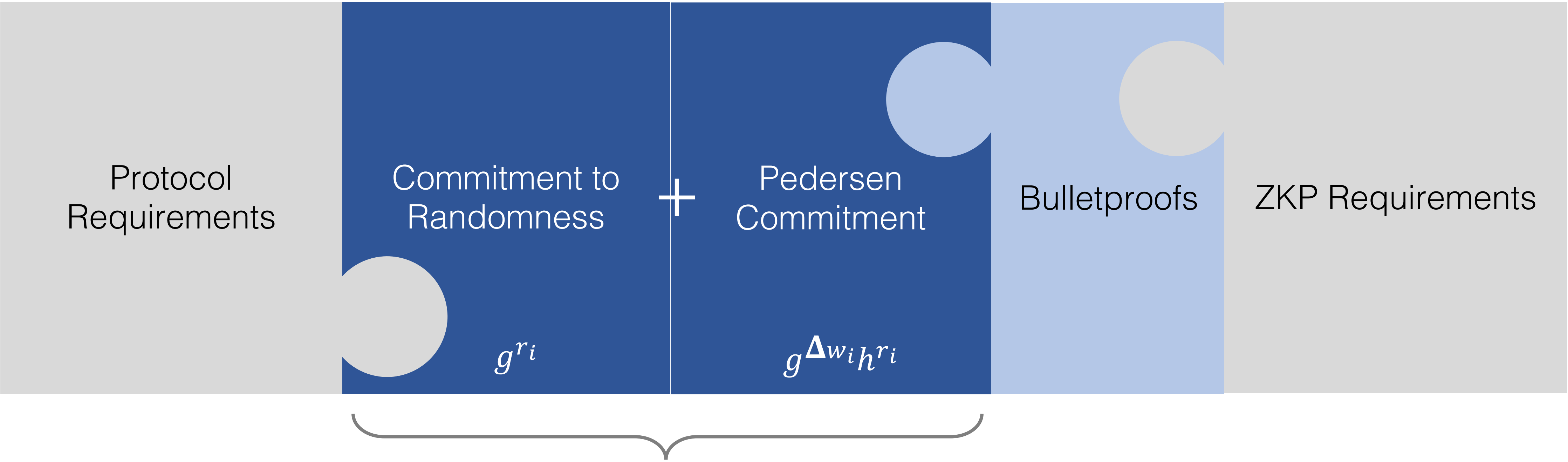
	GGPR-style zkSNARKs
Proof size	$O(1)$
Prover time	$O(\ell \log(\ell))$
Verification time	$O(1)$



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	❌	✅
Specialized range proof construction	❌	✅
No trusted setup	❌	✅

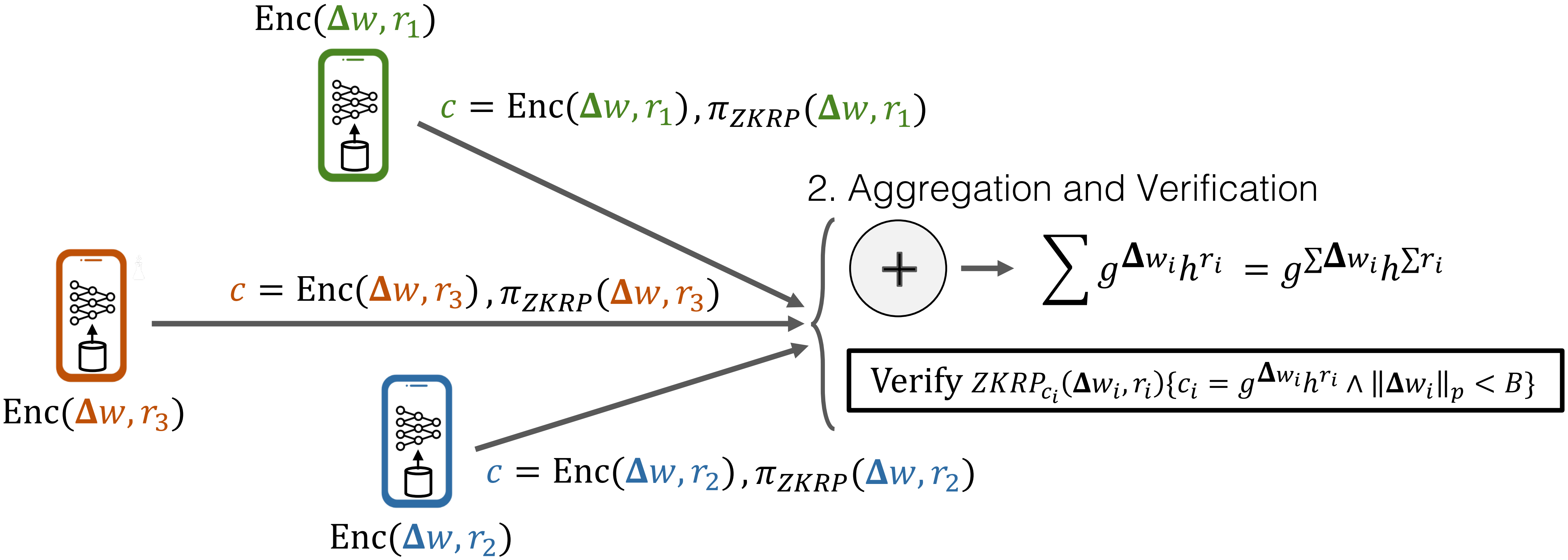
Extending Pedersen commitments for correctness



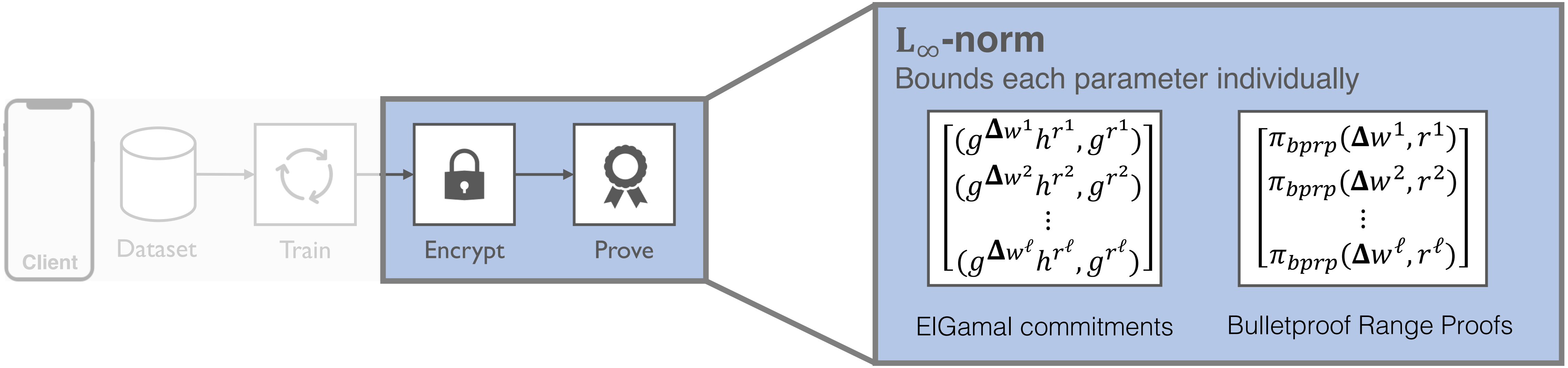
ElGamal commitment

- 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 $(g^{\Delta w_i} h^{r_i}, g^{r_i})$

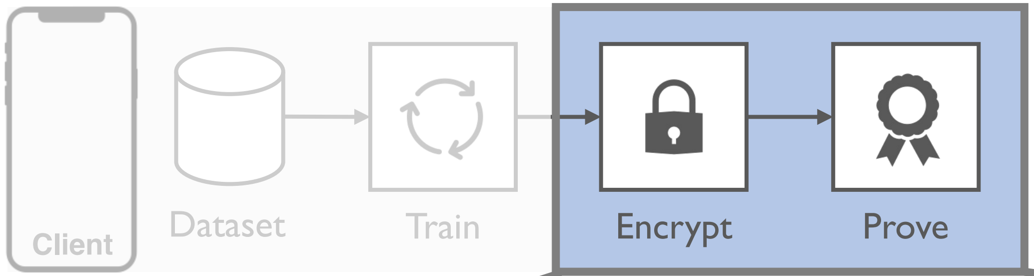
Secure Aggregation with Input Constraints



Enforcing Norm Bounds



Enforcing Norm Bounds



L_2-norm

Bounds the sum of squared parameters

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

ElGamal commitments

Squared commitments

$$\begin{bmatrix} \pi_{bprp}(\boldsymbol{\Delta} w^1, r^1) \\ \pi_{bprp}(\boldsymbol{\Delta} w^2, r^2) \\ \vdots \\ \pi_{bprp}(\boldsymbol{\Delta} w^\ell, r^\ell) \end{bmatrix} \qquad \boxed{\pi_{bprp}\Big(\sum_{i=1}^{\ell} (\boldsymbol{\Delta} w^i)^2\Big)}$$

Bulletproof Range Proofs

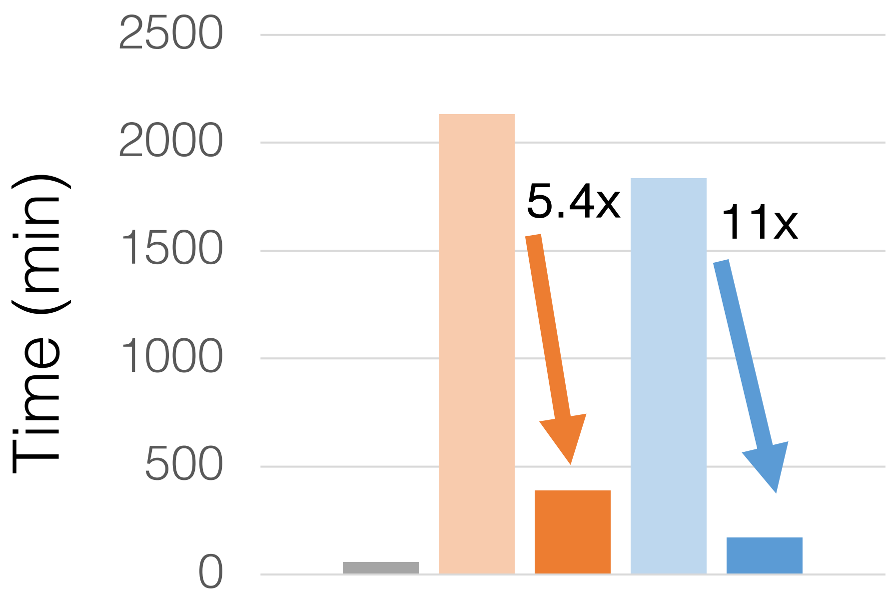
Squared Range Proof

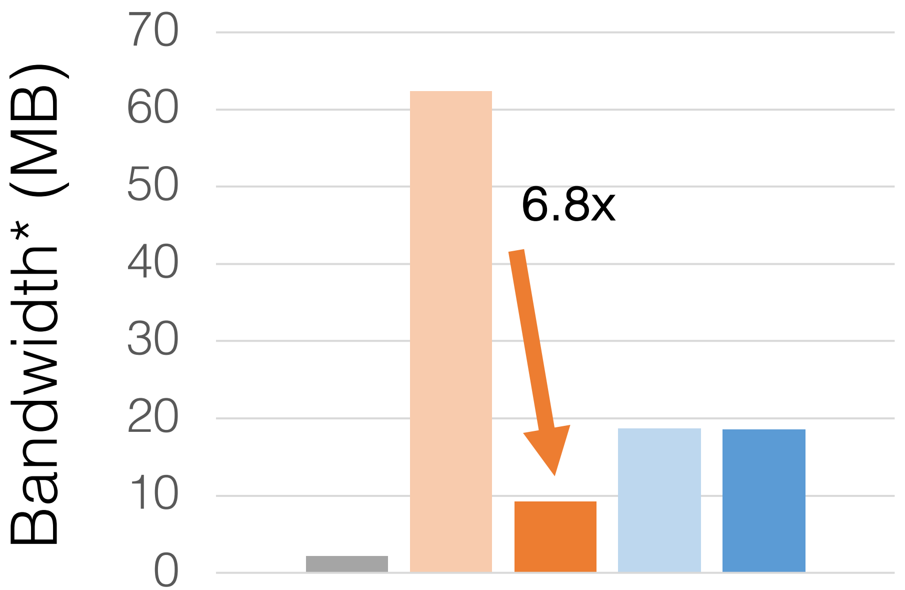
RoFL: End-To-End Performance

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

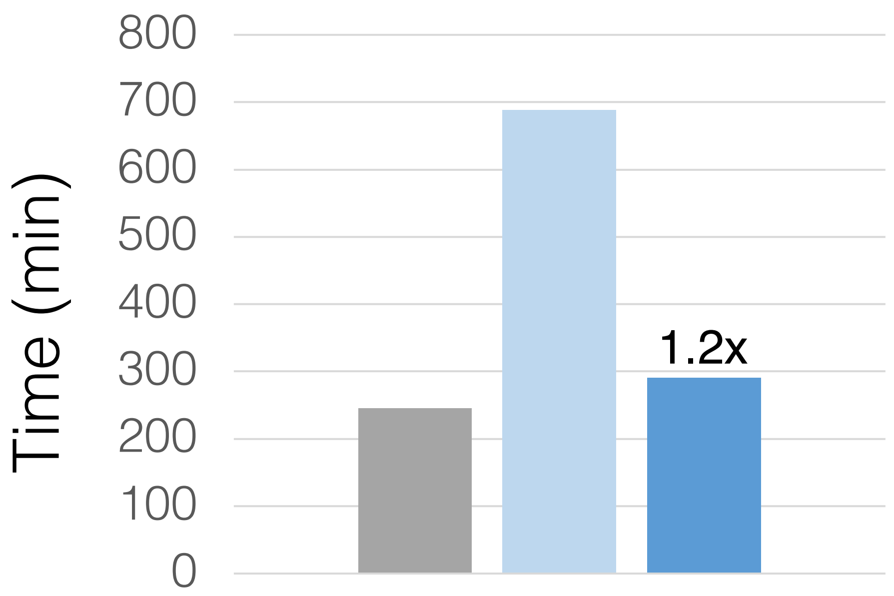
Shakespeare Model 818k Parameters

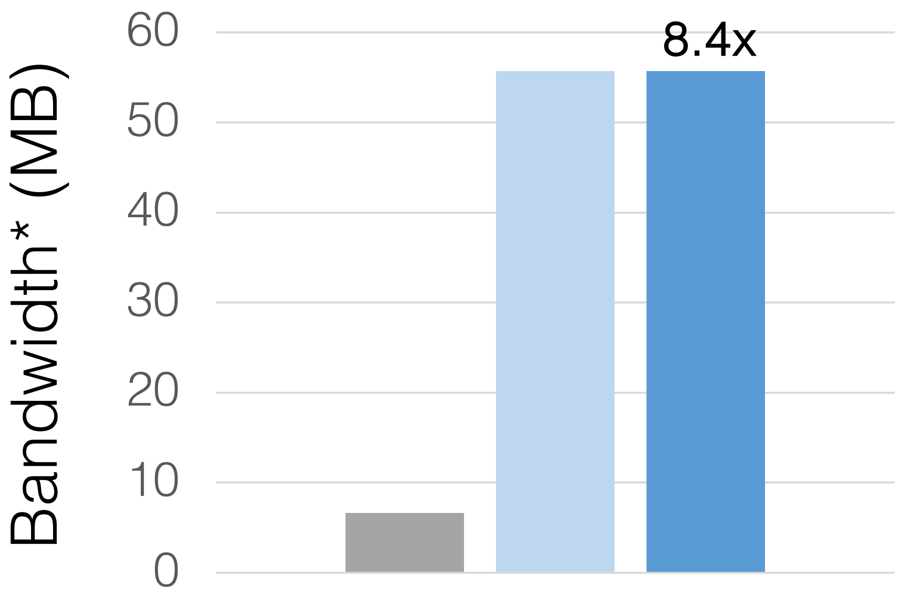
Setup: 48 Clients, 20 rounds

SecAgg — Accuracy: 0.57

L_∞ — 0.57

L_∞ Optimized — 0.57





* Per client per round

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