Security and Robustness of Collaborative Learning Systems

Anwar Hithnawi

ETH zürich
Collaborative Learning

RoFL: Attestable Robustness for Secure FL. arXiv:2107.03311
L. Burkhalter*, H. Lycklama*, A. Viand, N. Küchler, A. Hithnawi

Analysis Code: https://github.com/pps-lab/fl-analysis
RoFL Code: https://github.com/pps-lab/rofl-project-code
Data Driven World

Autonomous Driving

Object Classification

Health Care

AlphaGo
large, diverse data $\rightarrow$ broad generalization
Solving tasks where data is accessible…

Tasks
- WikiText-103
- MNIST
- ImageNet
- WMT
- GPT-3
- CIFAR

Public Data
- Crowdsourced Data

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

… however, many important tasks we care about …

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

Data Silos
- Privacy Laws
- Competition

Legal Frameworks and Technologies to facilitate privacy preserving access
Collaborative Learning
Collaborative Learning

Decentralized Learning
Collaborative Learning

Decentralized Learning

Federated Learning

∑ Coordinator

\[ w \xrightarrow{\Delta w} \]

...
Security and Privacy of Machine Learning

Data $\rightarrow$ Train/Model $\rightarrow$ Deploy/Serve

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

Diagram:
- $X \rightarrow \theta$ $\rightarrow$ $x \rightarrow y'$
- Test $\rightarrow$ Train
Security and Privacy of Collaborative ML

Data

Train/Model

Deploy/Serve

Clients

-$\theta$-

-$x$-$y'$

- adversarial examples (perturbations)
- model extraction/inversion attacks
- poisoning attacks (backdoors)
- data stealing attacks

Test

Train
Confidentiality of Input Data

Federation ≠ Privacy

Information Stealing Attacks on Federated Learning

(e.g., Gradient Inversion, Gradient Amplifications, Trap Weights)

Wang et al., Beyond Inferring Class Representatives: User-Level Privacy Leakage From Federated Learning, 2019
Geipin et al., How easy is it to break privacy in federated learning?, 2020
Boenisch et al., When the curious abandon honesty: Federated learning is not private, 2021
Yin et al., See through Gradients: Image Batch Recovery via GradInversion, 2021
Wen et al., Fishing for user data in large-batch federated learning via gradient magnification, 2022
Cryptography → Secure Computation

Secure Multi-Party Computation (MPC)

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]

Secure Aggregation
Cryptography → Secure Computation

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)

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]
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

Secure Federated Learning

Data Poisoning
Adversarial Robustness – Training

Data Poisoning (adversary controls training data)

Train + Apple → Dog

Test → Dog

Secure Federated Learning

Data Poisoning
Adversarial Robustness – Training

Data Poisoning
(adversary controls training data)

Train  +  \text{Dog}  \\
Test  \rightarrow  \text{Dog}

\sum \text{Coordinator}

Data Poisoning
Secure Federated Learning
Adversarial Robustness – Training

Data Poisoning
(adversary controls training data)

Train: +, Dog

Test: Dog

Model Poisoning
(adversary controls model updates)

Scaling: \( \times \gamma \)
Crafted: \( \varphi() \)

Secure Federated Learning
Adversarial Robustness – Training

Data Poisoning
(adversary controls training data)

Model Poisoning
(adversary controls model updates)

Secure Decentralized Learning
Adversarial Robustness – Training

Data Poisoning
(adversary controls training data)

Model Poisoning
(adversary controls model updates)

cryptographic commitment

Secure Decentralized Learning
Adversarial Robustness – Training

Secure Decentralized Learning

Data Poisoning  
(adversary controls training data)

Model Poisoning  
(adversary controls model updates)

cryptographic commitment
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 $f$

Given $P(x)$
-- Verify: $P(x) = f(x)$

Machine Learning Setting
In ML $f$ is learned
($f = \text{unknown ground truth}$)

Given $P(x)$
-- 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
  ▪ …

Chiesa et al., Proofs of Proximity for Distribution Testing, ITCS 2018
Goldwasser et al., Interactive Proofs for Verifying Machine Learning, ITCS 2021.
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: Attestable Robustness for Secure FL

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

Understand Vulnerabilities in FL

Cryptographically Enforce Constraints
What are the vulnerabilities in the FL pipeline that enable model/data poisoning attacks?
Model Replacement Attack

Bagdasaryan et al., How to backdoor Federated Learning, AISTATS 2020.
Bhagoji et al., Analyzing Federated Learning through an Adversarial Lens, ICML 2019.
Problem: Linear aggregation rules are vulnerable to Byzantine behavior

Machine Learning: Byzantine-Robust Distributed Learning
- 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]

Security: Private Data-Collection Systems
- 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

Update type: ⬅️ Benign ⬇️ Malicious

---

$L_\infty$-B = 0.01

---

Benign & Malicious Samples  ➔  Regular Training  ➔  Scaling

---

$L_2$-Bound: ⬇️ None ⬆️ 4.0

---

Single client, single round

---

$\leq B$
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
Why?
Long Tail …

Two common types of examples in the long-tail:

- Noisy
- Atypical

Fig: Hooker et al., 2019. Hooker, Moorosi et al., 2020.
Memorization leads to Leakage of private text

from the Bible (1 Kings 7:2)

Memorization
Model Capacity Implications on Robustness…
Success of Backdoor Attacks

Prototypical Targets

$L_2$-Bound: Purple = 2.0, Blue = 5.0, Green = 30.0, Red = None

Success of Backdoor Attacks vs. Round

Prototypical Targets vs. Tail Samples
Success of Backdoor Attacks

Tail Targets

![Graph showing the success of backdoor attacks]

- Att. Obj: Tail
- Prototyped
- Tail

# Samples

Prototypical Tail

41
Suppressing the Long-Tail

**Approaches**

- Noise Addition (Differential Privacy)
- Compression

Understand trade-offs between objectives we care about:

- Robustness
- Accuracy
- Fairness
- Privacy

---

Differential Privacy disproportionately impacts underrepresented attributes [Bagdasaryan et al. NeurIPS 2019]

Leads to Fairness Problems

![Graph showing accuracy gaps for different skin tones under differential privacy settings.](image)
More Resources ...

“Understanding how capacity impacts (fairness, robustness, privacy) is an increasingly urgent question.” — Sarah Hooker

In the Talk
The myth of interpretable, robust, compact and high performance DNNs

“Understanding the generalization properties of learning systems (...) is an area of great practical importance.” — Vitaly Feldman

In the Paper
Does Learning Require Memorization? A Short Tale about a Long Tail
Binary View of Robustness
### Where can Norm Bound Help?

<table>
<thead>
<tr>
<th>Attack Type / Attack Target</th>
<th>Prototypical</th>
<th>Tail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single-shot</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Continuous</td>
<td>✅</td>
<td>⬜</td>
</tr>
</tbody>
</table>

- **Requires attacker to be consistently selected** $\alpha > 2.5\%$
- **Data Poisoning**
  - $\alpha \leq 0.1\%
  - $\alpha \leq 0.01\%$
- **Model Poisoning**
  - % of compromised clients, $\alpha$

### Norm Bound Provides Practical Robustness Guarantees
RoFL: Attestable Robustness for Secure FL

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

Understand Vulnerabilities in FL

Cryptographically Enforce Constraints
Goal: Augment existing secure FL with Zero-Knowledge Proofs to enforce constraints on model updates
Secure Federated Learning
RoFL Augments Secure FL

 Clients

 Dataset → Train → Encrypt → Prove

 FL Server

 Secure Aggregation

 RoFL Proxy Modules

 Zero-knowledge Proofs

 Model Update
Secure Aggregation

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

Idea: Cancelling masks

$s_1 + s_2 + s_3 = 0$

+: modular addition

Secure Aggregation

Goal: Compute \( \sum \Delta w_i = \Delta w + \Delta w + \Delta w \)

Idea: Cancelling masks

\( s_1 + s_2 + s_3 = 0 \)

Cryptographic Commitments

\[ C(\Delta w_i, s_i) \]
Switching to Homomorphic Commitments

Additively homomorphic commitments

\[ C(m_1, r_1) \oplus C(m_2, r_2) = C(m_1 + m_2, r_1 + r_2) \]

\[ C(\sum \Delta w_i, \sum s_i) = C(\sum \Delta w_i, 0) \]
Zero-knowledge Proofs for Norm Constraints

\[
\begin{bmatrix}
C(p_1, r_1) \\
C(p_2, r_2) \\
\vdots \\
C(p_d, r_d)
\end{bmatrix}
\]

ElGamal commitments

Non-Interactive Zero-Knowledge Proofs

\(L_\infty, L_2\)-norm
Enforcing $L_\infty$-norm

$\begin{bmatrix}
C(p_1, r_1) \\
C(p_2, r_2) \\
\vdots \\
C(p_d, r_d)
\end{bmatrix}$

ElGamal commitments

$\begin{bmatrix}
-b \leq p_1 \leq b \\
-b \leq p_2 \leq b \\
\vdots \\
-b \leq p_d \leq b
\end{bmatrix}$

Bulletproof Range Proofs

Enforcing $L_2$-norm

ElGamal commitments

\[
\begin{bmatrix}
C(p_1, r_1) \\
C(p_2, r_2) \\
\vdots \\
C(p_d, r_d)
\end{bmatrix}
\]

Range Proofs

\[
\begin{bmatrix}
-b \leq p_1 \leq b \\
-b \leq p_2 \leq b \\
\vdots \\
-b \leq p_d \leq b
\end{bmatrix}
\]

Proof of Square Relation

\[
\sum_{i=1}^{d} p_i^2 \leq B^2
\]

Square Proof

Proof of Square Relation \iff Square Proof

Encrypt \rightarrow Prove

Dataset \rightarrow Train

59
Problem: Scalability

\[
\begin{bmatrix}
p_1 \\
p_2 \\
\vdots \\
p_d \\
\end{bmatrix} > 100k
\]

High-dimensional updates

Number of clients
Can we reduce the number of proofs while maintaining the same level of security?
Optimizing $L_\infty$

Clients

- Dataset
- Train
- Encrypt
- Prove

Commitments $\Delta w$, range proofs

FL Server

$\sum$

Secure Aggregation

Verify

Number of scaled weights required

Number of required checks

Graphs:

- Malicious Accuracy vs. Percentage of weights scaled ($K$)
  - FMN-P
  - C10-P

- Detection rate vs. Percentage of checks $p_c$
  - C10-P
  - FMN-P
Optimizing $L_\infty$

Clients

Dataset → Train → Encrypt → Prove → Commitments $\Delta w$ → FL Server

Proof indices → Range proofs

Secure Aggregation → Verify

Graphs:

- Malicious Accuracy vs. Percentage of weights scaled ($K$)
  - FMN-P
  - C10-P

- Detection rate vs. Percentage of checks $p_c$
  - C10-P
  - FMN-P

Number of scaled weights required

Number of required checks
Optimizing $L_2$

\[
\begin{align*}
\{d\} & \quad \text{Model parameters} \\
\{\tilde{d}\} & \quad \text{Update parameters} \\
\{x\} & \quad \text{Secure Aggregation} \\
\{\sum \} & \quad \text{Verify} \\
\{\text{FL Server}\} & \\
\{\text{Clients}\} & \quad \text{Verify}
\end{align*}
\]
RoFL: End-To-End Performance

CIFAR-10 Model 270k Parameters

Setup: 48 Clients, 160 rounds

Accuracy: 0.86

- Plaintext
- $L_2$: 0.85
- $L_2$ Optimized: 0.82
- $L_\infty$: 0.85
- $L_\infty$ Optimized: 0.85

Time (min)

Bandwidth* (MB)

* Per client per round
### Evaluation: End-To-End

**Shakespeare Model 818k Parameters**

<table>
<thead>
<tr>
<th>Setup: 48 Clients, 20 rounds</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>$L_\infty$ Optimized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy: 0.57</td>
<td>0.57</td>
</tr>
</tbody>
</table>

**Plaintext**

- Time (min):
  - 1.3x

**$L_\infty$ Optimized**

- Time (min):
  - 0.57

- Bandwidth* (MB):
  - 180

* Per client per round

---

67

Analysis Code: https://github.com/pps-lab/fl-analysis
RoFL Code: https://github.com/pps-lab/rofl-project-code