Aalto University

UNIVERSITY OF WATERLOO

WAFFLE: Watermarking in Federated Learning

Buse G. A. Tekgul, Yuxi Xia, Samuel Marchal, N. Asokan
Secure Systems Group

Outline

Why and how to demonstrate ownership of machine learning models ?

Why current watermarking techniques do not apply to federated learning?

How to reliably demonstrate the model ownership in federated learning?

Why ownership demonstration is important?

Machine learning models: business advantage and intellectual property (IP)

Cost of

- gathering relevant data
- labeling data
- expertise required to choose the right training method
- Resources expended in training

Adversary who steals the model can avoid these costs.

Watermarking DNN Models by backdooring[1]

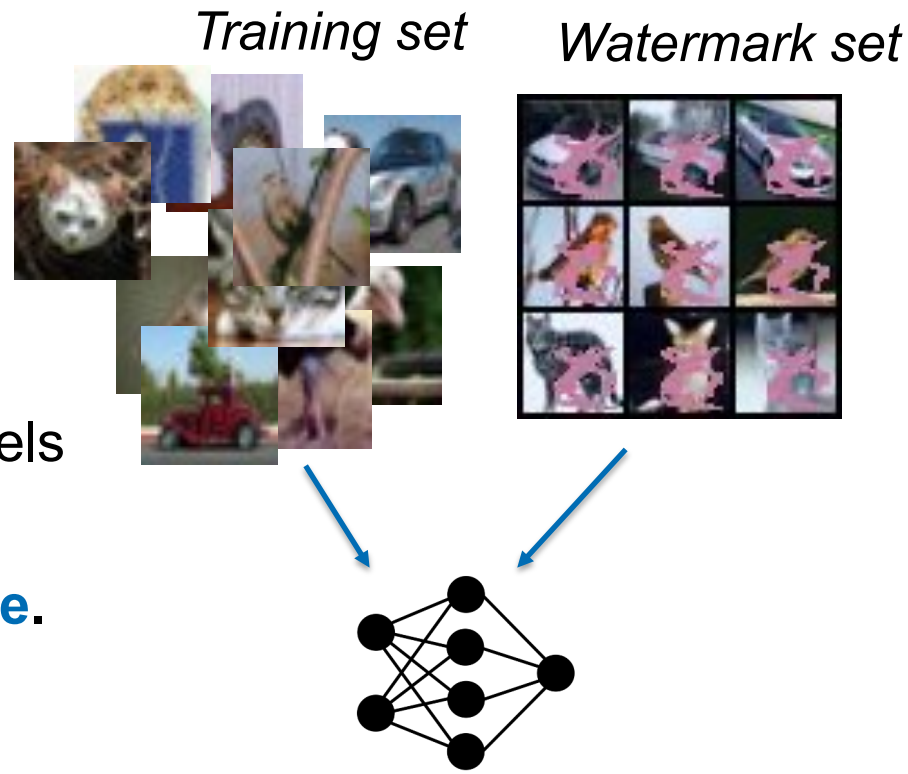
Watermark embedding:

- Embed the watermark in the model **during the training phase**:
  - Choose incorrect labels for a set of samples (*watermark set, WM)*
  - Train using training data + *watermark set*

Verification of ownership:

- Adversary publicly exposes the stolen model
- Query the model with the *watermark set*
- Verify watermark - predictions correspond to chosen labels

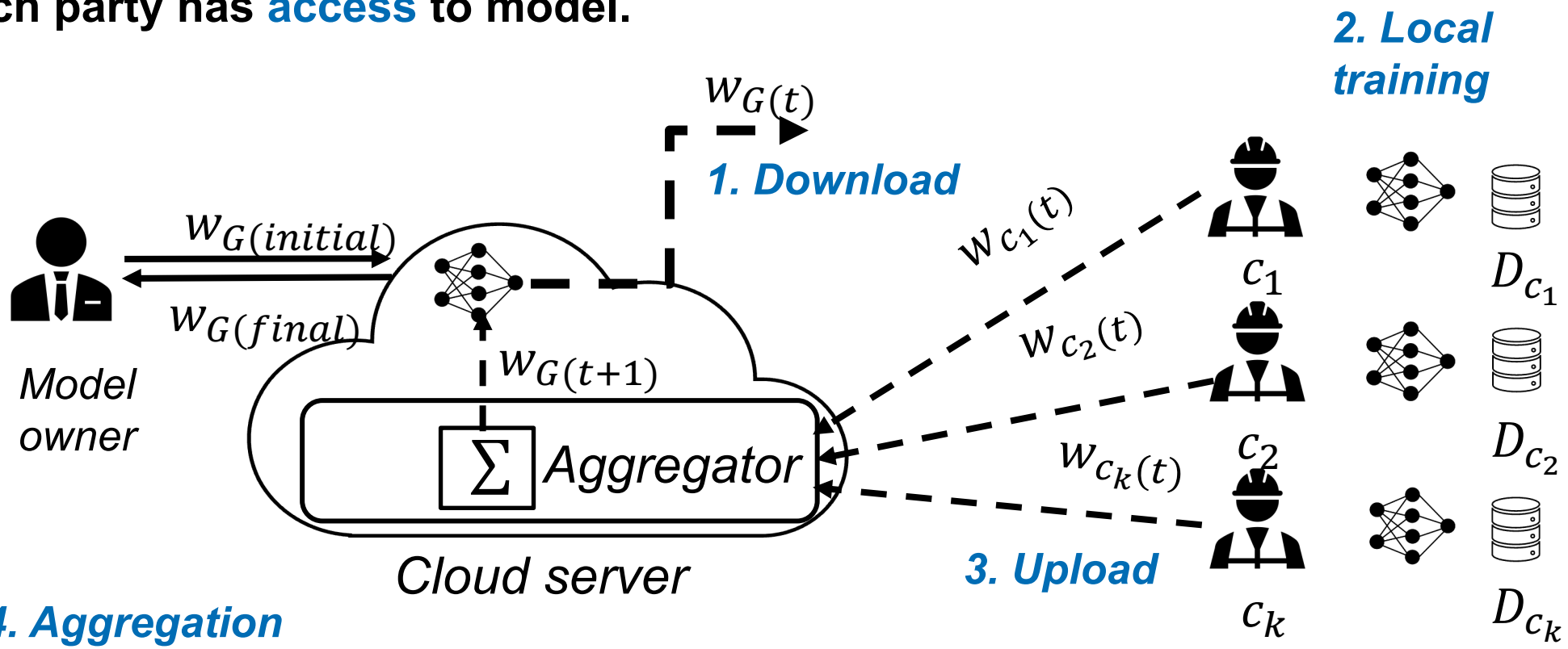
Requires access to training data and training procedure.



[1] *Adi et al. "Watermarking Deep Neural Networks by Backdooring."* USENIX '18 (https://www.usenix.org/node/217594)

Client-server Federated Learning

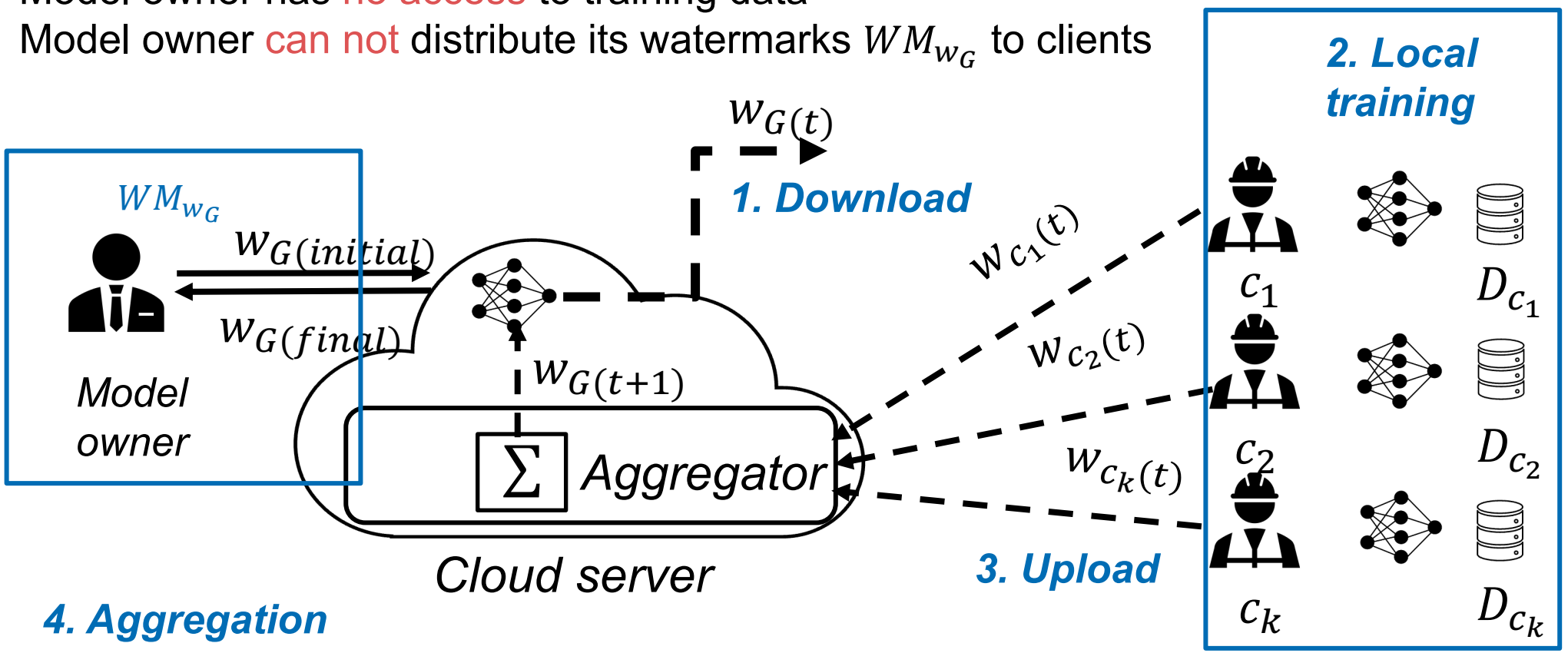
- Communication efficient and privacy preserving distributed training.
- One model owner (e.g., server or an external party) and multiple data owners.
- Each party has access to model.



Client-server Federated Learning

- **Ownership demonstration is important in client-server type configuration.**

- **Current watermarking solutions are not suitable:**
  - Both training and the dataset is distributed
  - Model owner has no access to training data
  - Model owner can not distribute its watermarks WM_{w_G} to clients



Ownership Demonstration in Federated Learning

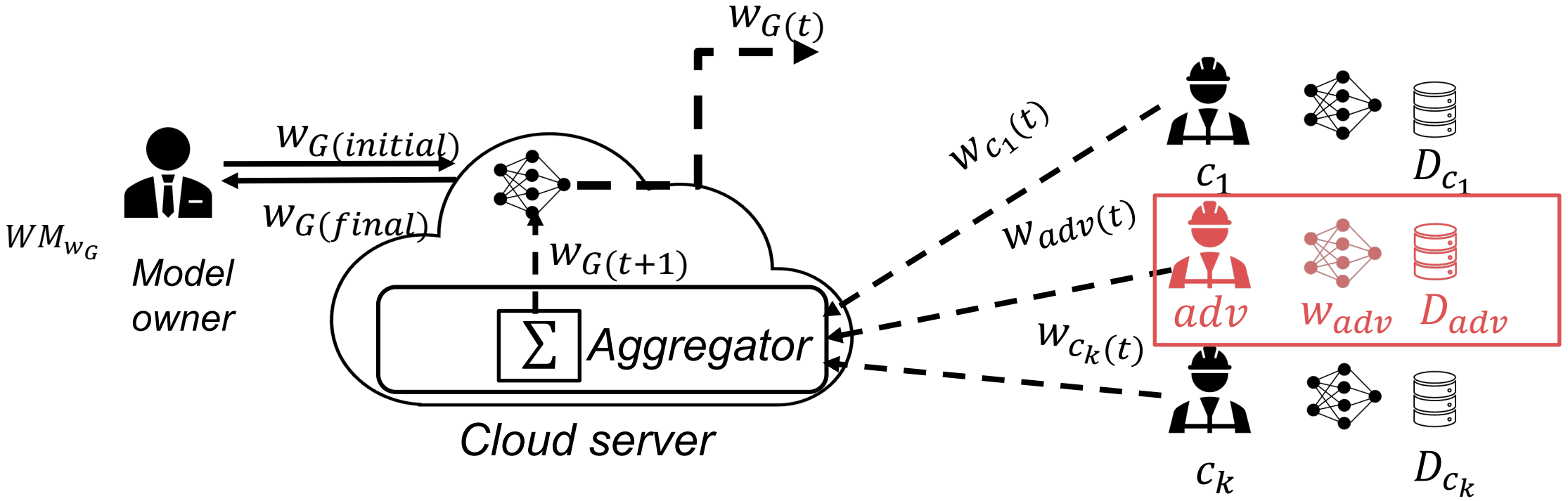
Our goals and contributions:

- Define necessary requirements for designing an effective watermarking solution to address ownership demonstration problem in client-server federated learning
- Propose
  - watermarking procedure (WAFFLE)
  - watermark set generation method (WAFFLEPATTERN) suitable for federated learning

Adversary Model

Adversary

- Honest-but-curious client: runs protocol as specified, try to remove watermarks later
- Goal: Obtain a local model with the same performance of global model and evade detection of ownership demonstration
  - $(Acc(w_{adv}, D_{test})) \approx Acc(w_{G(final)}, D_{test}))$, $VERIFY(w_{adv}, WM_{w_G}) \rightarrow False$
- Capability: access to training data D_{adv}, global model $w_{G(t)}$ and local models $w_{adv(t)}$



Requirements

w^t : watermarked model
w^- : post-processed model
$Acc(w, D_{test})$: Accuracy of a model on some test dataset

A reliable watermarking scheme should …

1. demonstrate ownership at any aggregation round t
   - $(Acc(w_{adv(t)}, WM_{w_G})) \geq T_{acc}, VERIFY(w_{adv(t)}, WM_{w_G}) \rightarrow True$
2. be robust against attacks that try to remove watermarks
   - Ownership demonstration (1) still holds or $Acc(w^+_{adv(t)}, D_{test}) \gg Acc(w^-_{adv(t)}, D_{test})$
3. be independent of client's training data

A watermarked federated learning model w_G^+ should …

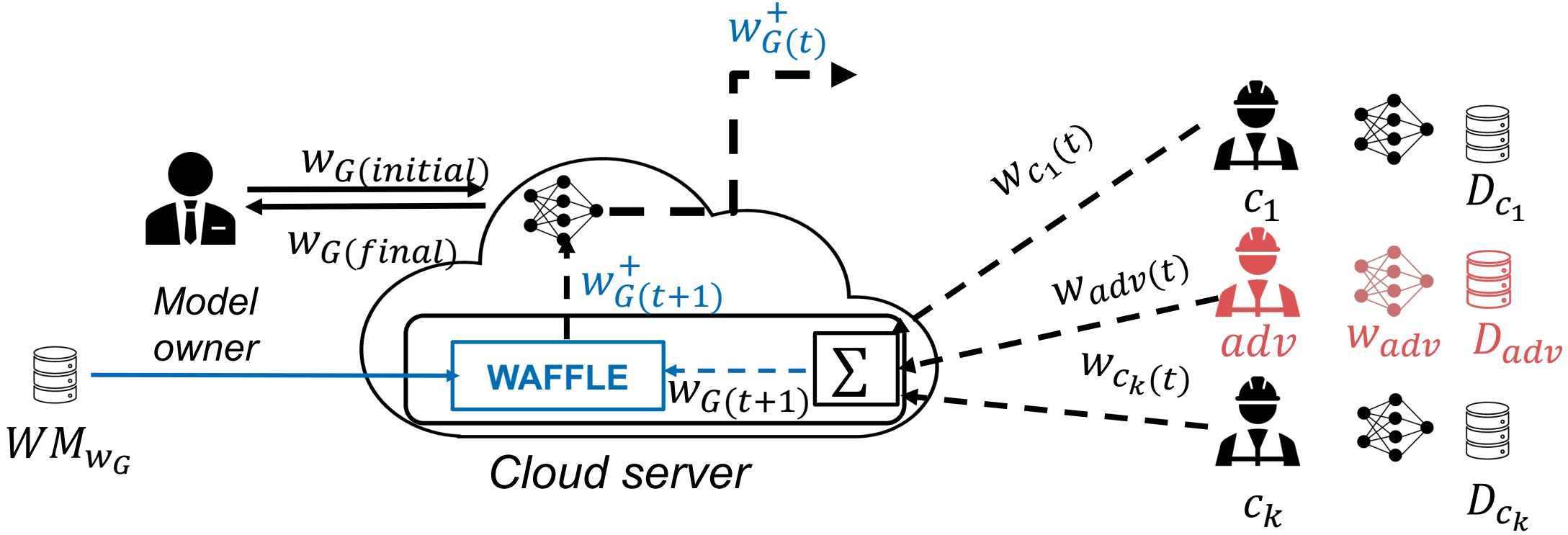
1. have a similar performance as in non-watermarked version
2. not increase communicational overhead (# of aggregation rounds) for convergence
3. incur minimal additional computation

WAFFLE Procedure[2]

First solution for addressing the ownership problem in federated learning.

Executed by the secure aggregator.

Makes no modification to client operations or secure aggregation.

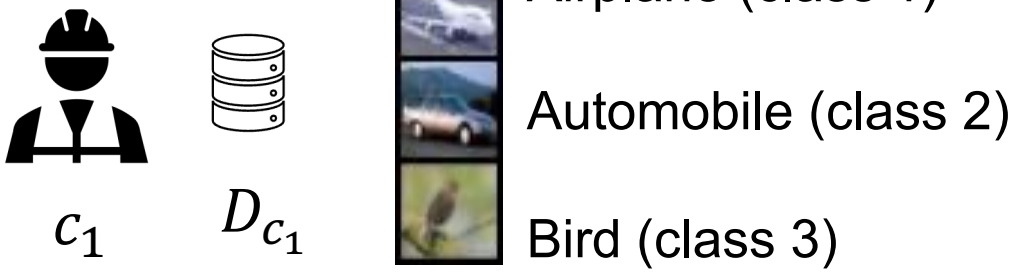


[2] Tekgul, Buse G. A., et al. "WAFFLE: Watermarking in Federated Learning." SRDS'21 (https://arxiv.org/abs/2008.07298)

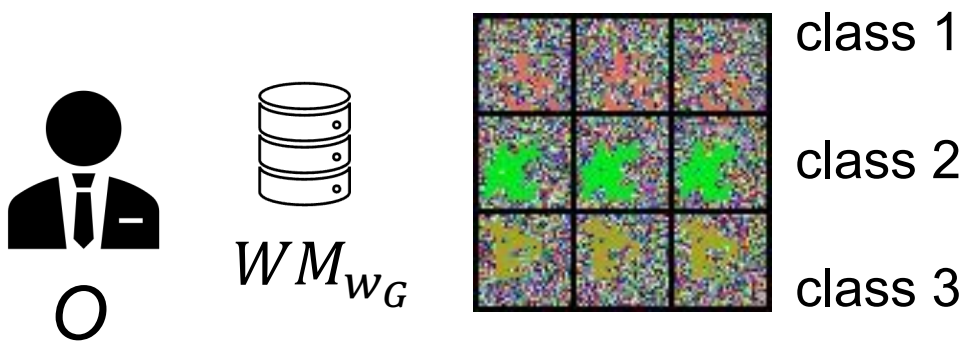
WAFFLEPATTERN

Novel data-independent method to generate watermarks for DNN image classification

- Gaussian noise as background
  - Negligible effect on main task accuracy
- Class specific structured pattern as foreground
  - Easy to learn, does not increase aggregation rounds



Training set



WAFFLEPATTERN

Evaluation : Experimental Setup

Datasets and DNN Models:

- MNIST handwritten digit dataset, CIFAR10 general classification dataset (10 classes)
- 5-layer convolutional network, VGG Imagenet model

Federated Learning:

- Federated Averaging[3] as aggregation algorithm, local training with SGD
- 100 total clients, 10 randomly selected clients joins training in each round
- 4 baselines: {total number of local passes $\boldsymbol{E_c}$, Number of aggregation rounds $\boldsymbol{E_a}$}
- Size of the watermark set: 100

Watermark is successfully embedded when:

- $Acc(w_{adv(t)}, WM_{w_G}) \geq T_{acc} = 47\%$[4] for a confidence $< 1 - 2^{-64}$ and
- $Acc(w_{adv(t)}, D_{test}) - Acc\left(w^{+}_{adv(t)}, D_{test}\right) \leq 5$ pp

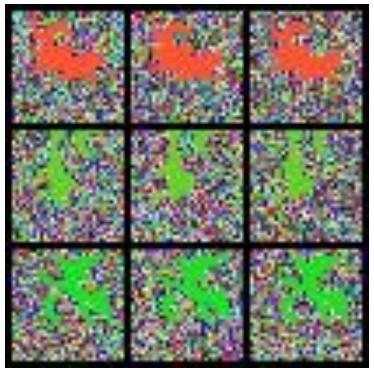
[3] McMahan Brendan et al. "Communication-efficient learning of deep networks from decentralized data." PMLR'17. (http://proceedings.mlr.press/v54/mcmahan17a.html)
[4] Szyller, Sebastian et al. "DAWN: Dynamic Adversarial Watermarking of Neural Networks." (https://arxiv.org/abs/1906.00830)

Evaluation : Experimental Setup

Watermark sets:

- Embedded Content[5] : meaningful content + subset of a training set
- unRelate[5, 1] : natural samples unrelated to original task
- unStruct[6] : randomly generated set

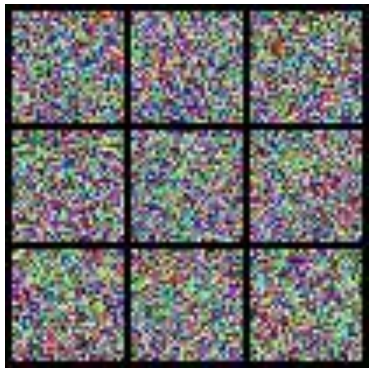
Original



WAFFLEPATTERN

Embedded Content[5]

unRelate [5,1]



unStruct [6]

[5] Zhang, Jialong, et al. 2018. "Protecting intellectual property of deep neural networks with watermarking." ASIACCS'18. (https://doi.org/10.1145/3196494.3196550)

[6] Rouhani, Darvish et al. 2019. "DeepSigns: an end-to-end watermarking framework for ownership protection of deep neural networks." ASPLOS'19. (https://doi.org/10.1145/3297858.3304051)

Evaluation : Demonstration of Ownership

WAFFLE successfully embeds all four types of watermark sets long before the model converges.

$\{E_c, E_a\}$	Watermark Accuracy (%)			
	MNIST		CIFAR10	
	Pre-embedding	WAFFLE	Pre-embedding	WAFFLE
{1, 250}	24.00	99.00	15.00	99.00
{5, 200}	30.00	99.00	14.00	99.50
{10, 150}	22.75	98.50	15.00	99.00
{20, 100}	31.00	98.75	16.00	99.75

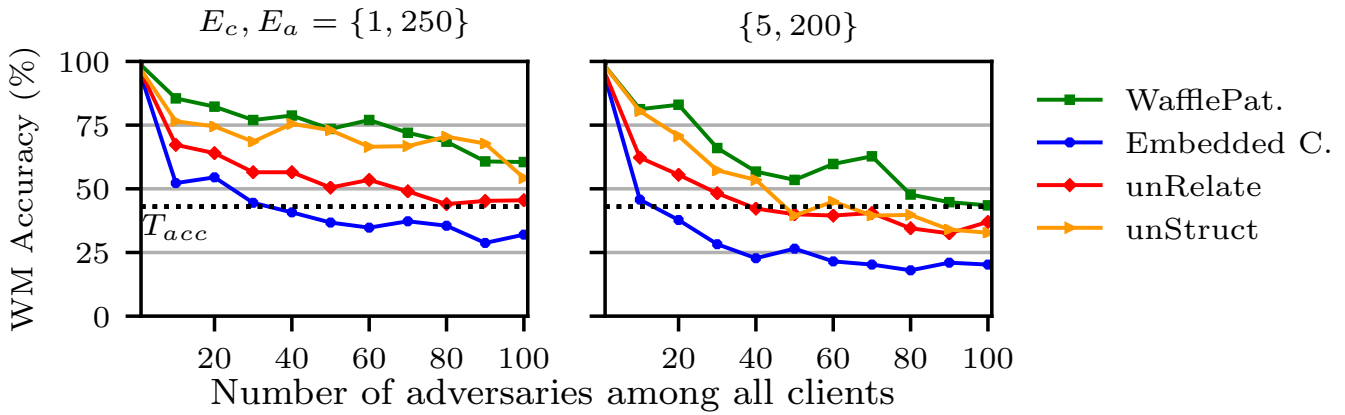
Post-embedding: local models at the last round have zero watermark accuracy

Evaluation : Robustness

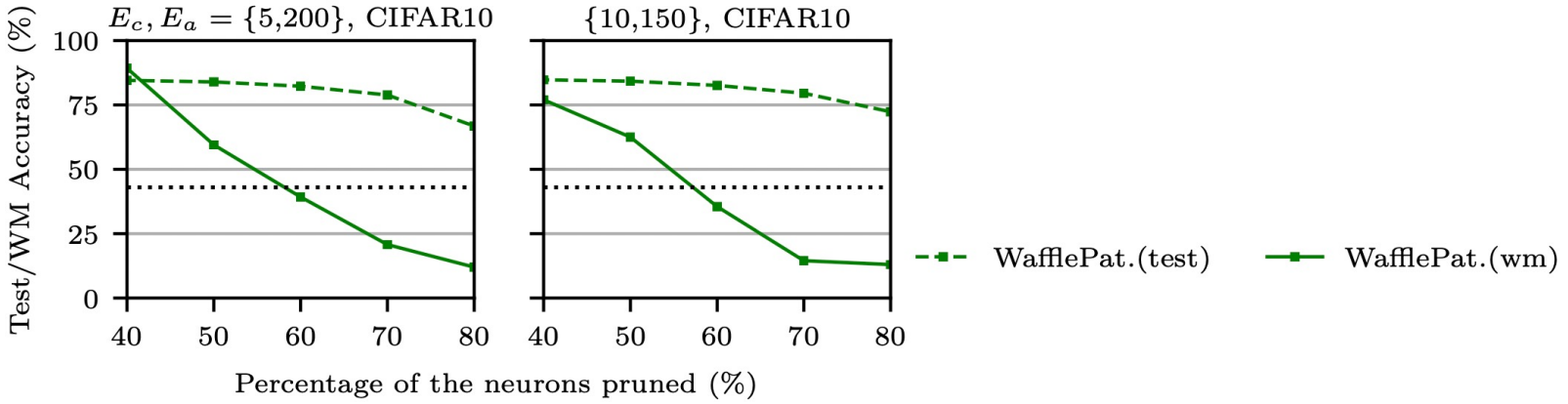
WAFFLEPATTERN is robust to post-processing watermark removal techniques

- Pruning and fine-tuning, if less than 40% of clients are malicious

Fine-tuning attack against CIFAR10



Pruning attack against CIFAR10 with 1 adversary



Evaluation: Robustness

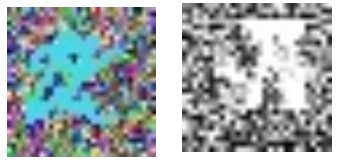
WAFFLEPATTERN cannot be recovered by reverse engineering and mitigation techniques rely on watermark recovery

- such as Neural Cleanse[7], if less than 10% of clients are malicious

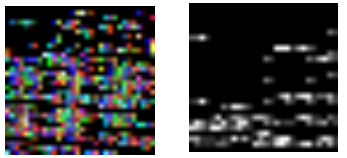
Patching via unlearning[7] against CIFAR10 (Test Accuracy % / Watermark Accuracy %)		
$\{E_c, E_a\}$	{1, 250}	{5, 200}
0 adversaries	**86.1 / 99.0**	**85.7 / 100.0**
1	79.7 / 45.5	72.3 / 36.5
2	78.1 / 67.0	77.0 / 30.5
5	76.3 / 36.5	79.0 / 40.0
10	79.2 / 38.0	81.3 / 33.8
20	81.0 / 44.8	81.3 / 32.8
40	83.3 / 37.0	81.3 / 26.2

Training set

Watermark set



Reversed watermark set with Neural Cleanse



[7] Wang, Bolun, et al. "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks." S&P'19 (https://ieeexplore.ieee.org/abstract/document/8835365)

Evaluation : Model Utility

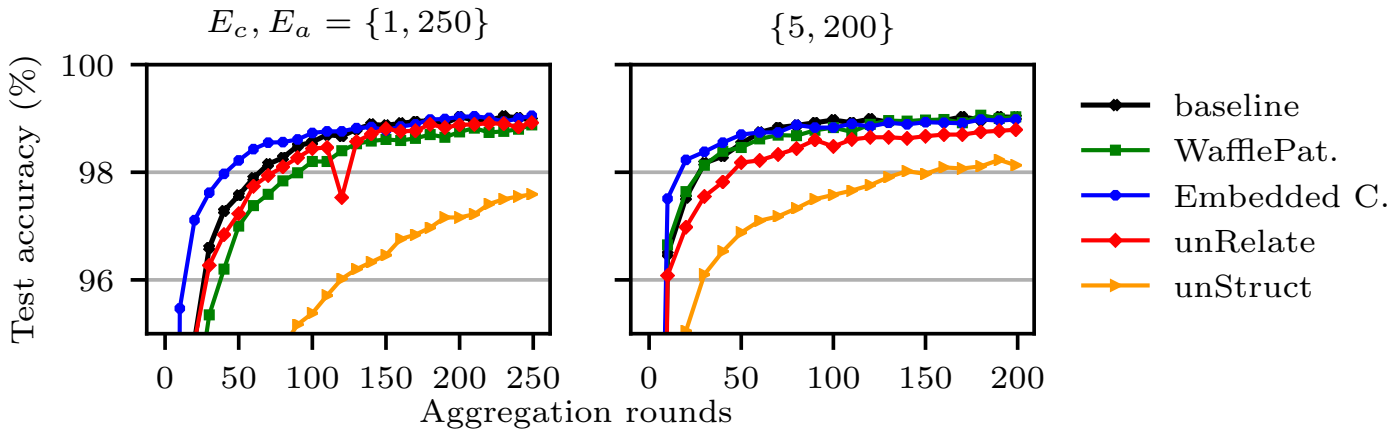
All watermarking schemes, including WAFFLEPATTERN, result in minimal drop in test accuracy compared to the baseline (< 2 pp).

Test Accuracy (MNIST % / CIFAR10 %)				
$\{E_c, E_a\}$	{1,250}	{5,200}	{10, 150}	{20, 100}
Baseline	**98.97 / 86.27**	**98.91 / 86.24**	**99.11 / 85.90**	**99.02 / 85.85**
WAFFLEPATTERN	98.88 / 85.70	98.94 / 85.61	99.06 / 85.89	98.95 / 85.67
Embedded C.	99.05 / 85.19	98.98 / 86.21	98.97 / 85.69	98.97 / 85.47
unRelate	98.92 / 85.81	98.79 / 86.25	99.06 / 85.76	98.79 / 85.74
unStruct	97.59 / 86.53	98.13 / 85.99	97.97 / 85.91	97.77 / 85.72

Evaluation : Communication and Computational Overhead

WAFFLEPATTERN has zero communication overhead, (i.e. additional aggregation rounds for convergence) and a negligible computational overhead.

Progression of test accuracy, MNIST



Computational Overhead at Aggregator (% number of retraining rounds in WAFFLE / total number of local retraining rounds)				
Dataset	**WAFFLEPATTERN**	**Embedded C.**	**unRelate**	**unStruct**
MNIST	3.06	2.02	10.39	0.91
CIFAR10	2.97	5.72	6.10	1.47

Evaluation : Evasion of Verification

WAFFLEPATTERN is resilient to evasion methods that detects queries used for watermark verification as out-of-distribution samples.

- In a non-IID setting, threshold based detectors[7] degrades model utility.

# of adversaries	True Positive Rate (%) / False Positive Rate (%, lowest) in CIFAR10	
	IID setting	Non-IID setting
1	64.0 / 0.8	89.95 / 53.0
5	88.0 / 1.6	92.2 / 22.9
10	94.7 / 2.5	90.8 / 19.7
20	90.0 / 1.1	91.8 / 7.0
40	81.0 / 1.0	91.8 / 6.8
50	80.0 / 0.6	84.0 / 4.8

[7] Li Zheng et al. "How to prove your model belongs to you: A blind-watermark based framework to protect intellectual property of DNN." ACSAC'19 (https://dl.acm.org/doi/abs/10.1145/3359789.3359801)

Takeaways

Demonstration of model ownership is important,
especially in federated learning.
Critical to protect business advantage.

Existing watermarking solutions can not be integrated into federated learning.
Distributed learning instead of centralized machine learning.

We propose WAFFLE and WAFFLEPATTERN to solve this problem.
Negligible decrease in performance (-0.01pp -- -0.63pp) ,
no communication overhead and low computational overhead (+3.02%).

More on our security/privacy + ML research at https://ssg.aalto.fi/research/projects/mlsec/