USENIX Security '26 Artifact Appendix: CompLeak: Deep Learning Model Compression Exacerbates Privacy Leakage

Na Li[1*] Yansong Gao[2*] Hongsheng Hu[3] Boyu Kuang[1 †] Anmin Fu[1 4 †]

[1]*School of Cyber Science and Engineering, Nanjing University of Science and Technology, China.*
[2]*School of Cyber Science and Engineering, Southeast University, China.*
[3]*School of Computer Science, Shanghai Jiao Tong University, China.*
[4]*School of Computer Science and Engineering, Nanjing University of Science and Technology, China.*

A Artifact Appendix

A.1 Abstract

This artifact provides an implementation of CompLeak, a privacy risk evaluation framework that investigates how model compression amplifies membership inference attacks (MIA). CompLeak supports three widely used compression techniques, including pruning, quantization, and weight clustering, and examines three attack scenarios—CompLeak $_{NR}$, CompLeak $_{SR}$, and CompLeak $_{MR}$—differing in the information accessible to the adversary.

The artifact comprises scripts for training original models, performing model compression, and executing CompLeak $_{NR}$, CompLeak $_{SR}$, and CompLeak $_{MR}$. It supports experiments with ResNet-18 on CIFAR-10. Please refer to the contained README.md for a detailed overview of the artifact contents and their organization.

A.2 Description & Requirements

This section describes the experimental setup, including hardware and software requirements, as well as the benchmarks used to generate the results.

A.2.1 Security, privacy, and ethical concerns

Executing this artifact raises no security, privacy, or ethical issues. All experiments use public datasets and widely used model architectures.

A.2.2 How to access

Our artifact is publicly available at: https://figshare.com/articles/software/CompLeak/30904289

We also provide the concept DOI of the artifact: https://doi.org/10.6084/m9.figshare.30904289

A.2.3 Hardware dependencies

We recommend an NVIDIA GeForce RTX 3080 or a more powerful GPU, as runtime depends on GPU performance. There are no strict CPU or memory requirements.

A.2.4 Software dependencies

All dependencies are listed in requirements.txt, with installation instructions provided in Section A.3.1.

A.2.5 Benchmarks

Detailed instructions are provided in Section A.3.1.

A.3 Set-up

This section describes how to install and configure the environment required to run the artifact.

A.3.1 Installation

The installation process includes dependency installation, data preparation, original model training, and model compression.

Dependency installation. Install all dependencies specified in requirements.txt:

```
pip install -r requirements.txt
```

Data preparation. The CIFAR-10 will be automatically downloaded to the data/datasets/ folder when the code is executed.

Original model training. Train the original model by running follow command:

*The authors contribute equally to this paper.
†Corresponding authors.

```
python train.py --dataset_name cifar10
--model_name resnet18 --num_cls 10 --lr 0.1
--epochs 100 --batch_size 128
```

Model compression. To compress the original model, follow these steps to obtain pruned models (at 60%, 70%, 80%, and 90% levels), an int-8 quantized model, and weight-clustered models (with cluster centers of 4, 8, and 16).

- **Pruning.** Use `-prune_sparsity` to set the pruning ratio i.e., 0.6, 0.7, 0.8, 0.9 for 60%, 70%, 80% and 90% pruning:

  ```
  python pruning/prune.py --dataset_name cifar10
  --model_name resnet18 --num_cls 10 --lr 0.001
  --prune_epochs 30 --prune_sparsity 0.6
  ```

- **Quantization.** Run:

  ```
  python quantization/qat.py  --degree int8
  --model_name resnet18 --num_cls 10 --lr 0.001
  --quantized_epochs 30 --dataset_name cifar10
  ```

  Weight Clustering. Use `-bit` to control cluster levels i.e., 2, 3, 4 for 4, 8, 16 cluster centers:

  ```
  python clustering/clustering.py
  --model_name resnet18 --num_cls 10
  --lr 0.00001 --epochs 30 --bit 2
  --dataset_name cifar10
  ```

At this point, the following models have been trained:

- One original model (full precision) in `results/`
- Multiple compressed models with different compression techniques and levels:
  - Pruned models with 60%, 70%, 80%, and 90% sparsity in `results_pruning/`
  - An INT-8 quantized model in `results_quantization/`
  - Clustered models with 4, 8, and 16 cluster centers in `results_clustering/`

A.3.2 Basic Test

You can run the following command to check that required dependencies are installed correctly. If the process displays 'Start Membership Inference Attacks,' it indicates that all dependencies are installed correctly.

```
python CompLeakSR.py --compress_degree 0.9
--dataset_name cifar10 --num_cls 10
--batch_size 128 --original --compress
--model_name resnet18 --compress_name pruning
```

A.4 Evaluation workflow

This section presents the key claims of the paper long with the experiments that support them.

A.4.1 Major Claims

The key claims presented in our paper are as follows:

(C1): `CompLeak` $_{\text{NR}}$ applies six existing MIAs to audit the privacy vulnerabilities of the original model and its per compressed variants. The results show that highly compressed models are generally less vulnerable to MIA than the original model, whereas other compressed models—e.g., pruning with low sparsity, int 8-bit quantization—exhibit privacy leakage comparable to the original model. This is proven by the experiment (E1), with reported in Section 4.

(C2): `CompLeak` $_{\text{SR}}$ leverages the compressed model as a reference, combining information from both the original model and one compressed model. The results reveal that compression operations indeed jeopardize privacy, outperforming `CompLeak` $_{\text{NR}}$'s performance on the original or compressed model. This is proven by the experiment (E2), with reported in Section 5.

(C3): `CompLeak` $_{\text{MR}}$ leverages the multiple compressed model as the reference. The results show that aggregating leakage from multiple compressed models further amplifies the privacy leakage. This is proven by the experiment (E3), with reported in Section 6.

A.4.2 Experiments

The key claims are supported by the following experiments:

(E1): [`CompLeak` $_{\text{NR}}$] [20 human-minutes + 3 compute-hour]: `CompLeak` $_{\text{NR}}$ (six existing MIAs) attacks each model individually, including the original and each compressed variant (pruned with 60%, 70%, 80%, 90%, int8-bit quantized, and weight clustered with 4, 8, 16 centroids.

Preparation: No further setup is needed once the data preparation, model training, and compression described in A.3.1 are complete.

Execution: To run the experiment, execute the `CompLeakNR.py`. First, apply `CompLeak` $_{\text{NR}}$ to the original model:

```
python CompLeakNR.py --model_name resnet18
--dataset_name cifar10 --num_cls 10
--batch_size 128 --original
```

Next, apply `CompLeak` $_{\text{NR}}$ to the compressed model.

Pruning:

Pruning ratio can be adjusted via `-compress_degree` (0.6, 0.7, 0.8, 0.9) for 60%, 70%, 80%, and 90% pruning.

```
python CompLeakNR.py --compress_degree 0.9
--dataset_name cifar10 --num_cls 10
--batch_size 128 --compress
```

`--compress_name pruning --model_name resnet18`

Quantization:

```
python CompLeakNR.py --compress_degree int8
--dataset_name cifar10 --num_cls 10
--batch_size 128 --compress
--compress_name quantization
--model_name resnet18
```

Weight clustering:

The number of centroids can be adjusted via `-compress_degree` to 4, 8, or 16.

```
python CompLeakNR.py --compress_degree 4
--dataset_name cifar10 --num_cls 10
--batch_size 128 --compress
--compress_name clustering
--model_name resnet18
```

Results: Each run outputs results for six MIAs. For each attack, we report ACC, AUC, and TPR at 0.1% FPR. Only highly compressed models, e.g., 90% pruning, show less vulnerability to MIAs compared to the original model (**C1**).

(E2): [$\texttt{CompLeak}_{\text{SR}}$] [20 human-minutes + 1 compute-hour + 75GB disk]: $\texttt{CompLeak}_{\text{SR}}$ combining information from both the original model and one compressed model, reveal that compression operations indeed jeopardize privacy (pruned with 60%, 70%, 80%, 90%, int8-bit quantized, and weight clustered with 4, 8, 16 centroids).

Preparation: No further setup is needed once the data preparation, model training, and compression described in A.3.1 are complete.

Execution: To run the experiment, execute the `CompLeakSR.py`. The `-compress_degree` parameter controls both the pruning ratio (0.6, 0.7, 0.8, and 0.9 corresponding to 60%, 70%, 80%, and 90% pruning) and the number of centroids (4, 8, or 16).

Validate pruning operation:

```
python CompLeakSR.py --compress_degree 0.9
--dataset_name cifar10 --num_cls 10
--batch_size 128 --original --compress
--model_name resnet18
--compress_name pruning
```

Validate quantization operation:

```
python CompLeakSR.py --compress_degree int8
--dataset_name cifar10 --num_cls 10
--batch_size 128 --original --compress
--model_name resnet18
--compress_name quantization
```

Validate weight clustering operation:

```
python CompLeakSR.py --compress_degree 4
--dataset_name cifar10 --num_cls 10
--batch_size 128 --original --compress
--model_name resnet18
--compress_name clustering
```

Results: We report ACC, AUC, and TPR at 0.1% FPR. Results on pruned models (60%, 70%, 80%, and 90%), int8-bit quantized model, and weight clustered models with 4, 8, and 16 centroids show that $\texttt{CompLeak}_{\text{SR}}$ outperforms $\texttt{CompLeak}_{\text{NR}}$'s performance on a single original/compressed model (C2).

(E3): [$\texttt{CompLeak}_{\text{MR}}$] [1 human-minutes + 3 compute-hour + 75GB disk]: $\texttt{CompLeak}_{\text{MR}}$ leverages the multiple compressed model as the reference. The results show that aggregating leakage from multiple compressed models further amplifies the privacy leakage.

Preparation: No additional setup is required once the data preparation, model training, and compression steps described in Section A.3.1 are completed, and $\texttt{CompLeak}_{\text{SR}}$ has been applied to all compressed models (a total of eight) in E2.

Execution: Open `CompLeakMR.ipynb` in Jupyter and execute all cells ("Run", "Run All Cells").

Results: The results of $\texttt{CompLeak}_{\text{MR}}$, presented in cell 19 of `CompLeakMR.ipynb`, show that aggregating leakage from multiple compressed models further amplifies privacy leakage compared to $\texttt{CompLeak}_{\text{SR}}$ (C3).

A.5 Version

Based on the LaTeX template for Artifact Evaluation V20231005. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/usenixsec2026/.