
# A Artifact Appendix

## A.1 Abstract

The artifact is based on PyTorch and requires GPU support. We implement our MIA defense SELENA, which consists of two components: Split-AI (Algorithm 1) and Self-Distillation. We also provide the implementation of prior defenses Adversarial Regularization [30] and MemGuard [21]. The artifact can reproduce experimental results in the main body, i.e., Table 2.

Our source code is available at https://github.com/inspire-group/MIAdefenseSELENA/tree/39428e763566a8276d82e1c0fe91bbaaddb84bfb.
We further provide a detailed guide for evaluating our artifact at https://github.com/inspire-group/MIAdefenseSELENA/blob/39428e763566a8276d82e1c0fe91bbaaddb84bfb/misc/reproducibility.md.

## A.2 Artifact check-list (meta-information)

- **Algorithm:** We implement our defense SELENA, which consists of two components: Split-AI (Algorithm 1) and Self-Distillation from Split-AI. We also provide comparison with prior defenses: undefended model, adversarial regularization [30] and MemGuard [21].
- **Program:** N/A.
- **Compilation:** N/A.
- **Transformations:** N/A.
- **Binary:** N/A.
- **Model:** 4-layer fully connected neural network and ResNet-18.
- **Data set:** Purchase100, Texas100, CIFAR100. They are publicly available benchmark datasets.
- **Run-time environment:** We test our artifact using anaconda virtual environment on Linux.
- **Hardware:** Requires one GPU.
- **Run-time state:** N/A.
- **Execution:** N/A.
- **Security, privacy, and ethical concerns:** N/A.
- **Metrics:** Membership inference attack accuracy. This is defined in Section 3.1. Random guess baseline is 50%.
- **Output:** We output results (classification accuracy and MIA accuracy) and intermediate results to console.
- **Experiments:** We provide reproducing instructions including commands.
- **How much disk space required (approximately)?:** Datasets take around 4 GB. Each model weight takes less than 100 MB.
- **How much time is needed to prepare workflow (approximately)?:** 1 hour.
- **How much time is needed to complete experiments (approximately)?:** The time range of defense for each experiment varies from a few minutes to around 30 hours (from scratch). See Table 3 and Table 4 for a reference. The time range of MIA attacks is approximately a few minutes for direct single-query attacks and data augmentation attacks, 4 $\sim$ 8 h for flip noise attack, 20 h for boundary distance attacks. The time of adaptive attacks is similar to the time of run SELENA defense.
- **Publicly available (explicitly provide evolving version reference)?:** https://github.com/inspire-group/MIAdefenseSELENA/tree/39428e763566a8276d82e1c0fe91bbaaddb84bfb.
- **Code licenses (if publicly available)?:** MIT License.
- **Data licenses (if publicly available)?:** N/A.
- **Workflow frameworks used?:** N/A.
- **Archived (explicitly provide DOI or stable reference)?:** N/A.

## A.3 Description

### A.3.1 How to access

We host our source code on GitHub at https://github.com/inspire-group/MIAdefenseSELENA/tree/39428e763566a8276d82e1c0fe91bbaaddb84bfb.
We further provide a detailed guide for evaluating our artifact at https://github.com/inspire-group/MIAdefenseSELENA/blob/39428e763566a8276d82e1c0fe91bbaaddb84bfb/misc/reproducibility.md.

### A.3.2 Hardware dependencies

The artifact requires 1 CPU and 1 GPU.

### A.3.3 Software dependencies

The artifact is based on Python, PyTorch, TensorFlow and other Python packages. All packages can be easily installed with pip; we provide a list of required packages in requirement.txt.

### A.3.4 Data sets

We use three publicly available datasets in our evaluation: Purchase100, Texas100, CIFAR100. See our reproducing instructions for more details.

### A.3.5 Models

The 4-layer fully connected neural network is for Purchase100/Texas100, which is widely used in prior MIA defenses [21, 30]. The ResNet-18 model for CIFAR100, which is widely used in image classification tasks.

#### A.3.6 Security, privacy, and ethical concerns

N/A

## A.4 Installation

Steps 1-3 can also be done in the Anaconda environment.

1. Install Python [help link] (or Anaconda ([help link]): conda create -n myenv python=3.8.5).
2. Install GPU-compatible PyTorch [help link] and TensorFlow. [help link] (or Anaconda: GPU-compatiable PyTorch [help link] and TensorFlow [help link]).
3. Install other Python dependencies [help link].
4. Clone the source code from `https://github.com/inspire-group/MIAdefenseSELENA/tree/39428e763566a8276d82e1c0fe91bbaaddb84bfb`.
5. Follow the preparation steps in Getting Started [help link].

## A.5 Experiment workflow

Our defense is implemented in $datasetname/SELENA folder. After preparing the initial dataset and environments, we first need to run $datasetname/data_partition.py to generate the npy files for member/nonmember sets to train/eval MIA attacks. We also need to generate the non-model indices for training set via $datasetname/SELENA/generation10.py. Then we need to train Split-AI model by $datasetname/SELENA/Split-AI/train.py. Next, we need to train the Self-Distillation model by $datasetname/SELENA/Distillation/train.py. To evaluate the protected model from Self-Distillation by membership inference attacks, we need to run $datasetname/SELENA/Distillation/eval.py (eval_cw.pyleval_aug.py). See reproducing instructions for more details. We can read the training/test accuracy for the classification model, and the membership inference attack accuracy from the console, which is the corresponding result of Table 2 in the paper.

## A.6 Evaluation and expected results

Our main claim is that our defense SELENA achieves a better trade-off between empirical membership privacy and utility compared to the state of the art MIA defenses [21,30]. This claim is supported by Table 2 of our paper. We can use commands listed in our reproducing instructions to generate our key results including the classification accuracy and MIA attack accuracy of our defense as well as prior MIA defenses [21, 30]. For accuracy on training set, see the corresponding classification accuracy for 'train'. For accuracy on test set, see the corresponding classification accuracy for 'test'. For direct single-query attacks, see 'Best direct single-query attack acc: '. For label-only attacks, see 'Best label-only attack at flip:' or 'CW attack:' or 'Augmentation attack:'. For adaptive attacks: see 'BEST ATTACK ACC:'.

The reported number should be consistent with Table 2 in the main body. It's possible to have around $0\% \sim 2\%$ mismatches due to some randomness.

## A.7 Experiment customization

Our source code provides an easy way to customize the experiment. The main algorithm in our SELENA defense is to generate non_model indices and perform adaptive inference on Split-AI, which can be easy adapted to datasets/models not listed in the source code. The parameters $K$ and $L$ can also be changed via flag `-K` and `-L` when needed.

## A.8 Version

Based on the LaTeX template for Artifact Evaluation V20220119.