USENIX Security '25 Artifact Appendix: Game of Arrows: On the (In-)Security of Weight Obfuscation for On-Device TEE-Shielded LLM Partition Algorithms

Pengli Wang[1], Bingyou Dong[2], Yifeng Cai[1], Zheng Zhang[2], Junlin Liu[1], Huanran Xue[2], Ye Wu[2], Yao Zhang[2], and Ziqi Zhang[3,*]

[1]*MOE Key Lab of HCST (PKU), School of Computer Science, Peking University*
[2]*ByteDance*
[3]*University of Illinois Urbana-Champaign*
[*]*Corresponding Author*

A Artifact Appendix

A.1 Abstract

Utilizing Trusted Execution Environments (TEEs) to protect Large Language Models (LLMs) on users' devices is a practical solution for model owners. To alleviate the computation burden on TEEs, researchers have proposed TEE-Shielded LLM Partition (TSLP) to offload heavy computation layers to co-operating untrusted GPUs, while lightweight layers are shielded in TEE. TSLP utilizes various lightweight obfuscation schemes to protect offloaded weights from various attacks meanwhile not introducing large computation overhead. However, existing lightweight obfuscation algorithms have one vital vulnerability in common: the direction similarity of obfuscated vectors. In this paper, we propose a novel attack, ARROWMATCH, that utilizes direction similarity to recover obfuscated private weights. To achieve this, ARROWMATCH compares direction distances between obfuscated model weights and public pre-trained model weights. To mitigate this vulnerability, we propose a novel obfuscation scheme, ARROWCLOAK, which leverages lightweight matrix-vector multiplication to protect vector directions and private weights. We evaluate ARROWMATCH and ARROWCLOAK on four representative LLMs, using seven datasets, along with five obfuscation schemes. The results show that ARROWMATCH can break the protection of all existing lightweight obfuscation schemes with high accuracy (similar to no protection) and effectively recover the private weights (with over 98% accuracy). In addition, ARROWCLOAK can effectively defend against ARROWMATCH (6.5× better than state of the art) and protect direction information by increasing the direction distance over 900×. We also evaluate the performance of ARROWCLOAK on a real-world Intel SGX device and show that ARROWCLOAK can reduce total overhead by 2.83× compared to shield-the-whole baseline.

A.2 Description & Requirements

We have carefully read the ethics considerations discussions in the conference call for papers, the detailed submission instructions, and the guidelines for ethics documents. This research focuses on the study of deep learning models and Trusted Execution Environments (TEEs) for enhancing secure and efficient computing. The datasets employed in this study were curated from publicly available resources that explicitly permit research use. Similarly, the deep learning models explored in this research are widely used in the academic community. We guarantee that our artifact contains no destructive steps, and executing our code will cause no damage to the host machine. Moreover, this study does not involve experiments on live systems, human participants, or data that could be linked to personally identifiable informa-tion (PII). Furthermore, no terms of service were violated, and the research adheres strictly to both the "Respect for Persons" and "Respect for Law and Public Interest" principles outlined in The Menlo Report.

A.2.1 How to access

Aligning with the open science policy, we release our code on [1, 2]. The code repository contains implementations of ARROWMATCH and ARROWCLOAK, along with corresponding scripts, across several model architectures(ViT, BERT, and GPT). By following the instructions in the code repository's README file, reviewers can reproduce the attack and defense performance in our paper.

A.2.2 Hardware dependencies

Our experiments are executed on 2 × NVIDIA RTX A6000 GPUs (48G). To ensure reproducibility, we recommend reviewers utilize identical or comparable GPUs for model training and evaluation.

A.2.3 Software dependencies

The code execution primarily relies on Python libraries for machine learning and numerical computing, such as torch, transformers, and scipy. To simplify environment configuration for reviewers, we have provided setup scripts (install1.sh and install2.sh) that automate the experimental environment installation.

A.2.4 Benchmarks

We employed four deep learning models: ViT-Base, BERT-Base, GPT-2, and GPT-2-XL, which are open-source models widely used in the academic community. Our experiments were conducted on seven datasets: CIFAR10, CIFAR100, FOOD101, MNLI, QQP, SST-2 and QNLI, where the first three are standard image classification datasets and the latter four are selected from the classic GLUE benchmark.

A.3 Set-up

A.3.1 Installation

We have prepared environment configuration scripts for reviewers: install1.sh (for text tasks) and install2.sh (for visual tasks). By following the instructions in the README within our code repository, reviewers can run these scripts to configure the corresponding experimental environment and activate it for subsequent training or testing tasks.

A.3.2 Basic Test

We have prepared environment test scripts for reviewers in the scripts2 directory: basic_test1.sh (for environment1) and basic_test2.sh (for environment2). By following the instructions in the README within our code repository, reviewers can run these scripts to test the corresponding experimental environment.

A.4 Evaluation workflow

A.4.1 Major Claims

(C1): ARROWMATCH can effectively recover the lightweight obfuscation algorithms and achieve a high attack performance. The attack performance is averagely over 1.6× higher than the black-box baseline. This is proven by the experiment described in section 7.2 whose results are reported in Table 3.

(C2): ARROWCLOAK can effectively protect the model against ARROWMATCH. The attack performance is only 1.10× higher than Shield-Whole and is over 6× better than the best prior obfuscation algorithms. This is proven by the experiment described in section 7.4 whose results are reported in Table 5.

A.4.2 Experiments

(E1): *[Fine-tune Phase] [10 human-minutes + 30 compute-hour]:*
Preparation: Check that the scripts exist in the scripts/ and scripts2/ directories, then locate the *train private model* section under Experiments in the README.
Execution: Execute the following scripts sequentially according to the README instructions to fine-tune ViT-Base, BERT-Base, GPT2-Base, and GPT2-XL across multiple datasets, with expected runtimes of 1.5h, 7.5h, 9h and 12h respectively.

```
./scripts2/vit_trains.sh
./scripts2/bert_trains.sh
./scripts2/gpt2_trains.sh
./scripts2/gpt2_xl_trains.sh
```

Results: The fine-tuned results will be stored in the results/train_results and results/tsqp_results directories. The performances are expected to be similar to the results in **White-box column in Table 3**. Please rename the final checkpoint of the BERT and GPT2 results to *final_checkpoint* to conduct subsequent attack and defense experiments.

(E2): *[ARROWMATCH Phase] [10 human-minutes + 8 compute-hour]:*
Preparation: Locate the *Try ARROWMATCH* section under Experiments in the README.
Execution: Execute the following scripts sequentially according to the README instructions to implement ARROWMATCH on ViT-Base, BERT-Base, GPT2-Base, and GPT2-XL across multiple datasets, with expected runtimes of 2.5h, 1.7h, 2h and 1h respectively.

```
./scripts2/vit_arrowmatchs.sh
./scripts2/bert_arrowmatchs.sh
./scripts2/gpt2_arrowmatchs.sh
./scripts2/gpt2_xl_arrowmatchs.sh
```

Results: The ARROWMATCH results will be stored in the results/arrowmatch_results directory. The performances are expected to be similar to the results in **corresponding obfuscation method's column in Table 3**. Please rename the final checkpoint of the BERT and GPT2 results to *final_checkpoint* to conduct subsequent evaluation experiments.

(E3): *[ArrowCloak Phase] [10 human-minutes + 2 compute-hour]:*
Preparation: Locate the *Try ARROWCLOAK* section under Experiments in the README.
Execution: Execute the following scripts sequentially according to the README instructions to implement ArrowCloak on ViT-Base, BERT-Base, GPT2-Base, and GPT2-XL across multiple datasets, with expected runtimes of 30 minutes,30 minutes, 30 minutes and 30 minutes respectively:

```
./scripts2/vit_arrowcloaks.sh
./scripts2/bert_arrowcloaks.sh
./scripts2/gpt2_arrowcloaks.sh
./scripts2/gpt2_xl_arrowcloaks.sh
```

Results: The ArrowCloak results will be stored in the results/arrowcloak_results directory. The performances are expected to be similar to the results in **ArrowCloak column in Table 5**. Please rename the final checkpoint of the BERT and GPT2 results to *final_checkpoint* to conduct subsequent evaluation experiments.

(E4): *[Evaluation Phase] [10 human-minutes + 3 compute-hour]:*
Preparation: Locate the *Evaluate the results* section under Experiments in the README.
Execution: Execute the following scripts sequentially according to the README instructions to implement evaluation of ArrowMatch and ArrowCloak on ViT-Base, BERT-Base, GPT2-Base, and GPT2-XL across multiple datasets, with expected runtimes of 45 minutes,45 minutes, 45 minutes and 45 minutes respectively:

```
./scripts2/evaluate_models_vit.sh
./scripts2/evaluate_models_bert.sh
./scripts2/evaluate_models_gpt2.sh
./scripts2/evaluate_models_gpt2_xl.sh
```

Results: The evaluation results will be stored in the evaluation_results directory. Reviewers can locate corresponding results in the evaluate_results/model_name/dataset_name directories, where **whitebox_results and blackbox_results represent the rightmost two columns of baselines in Figure 3; recover_results records the recovery performance of the ArrowMatch, corresponding to the *our attack* column in Figure 3, while arrowcloak_results documents ArrowCloak performance, aligning with the results in column ArrowCloak of Figure 5**.

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/usenixsec2025/.

References

[1] Github link. https://github.com/qsxltss/Game-of-Arrows, 2025.

[2] Zenodo link. https://zenodo.org/records/15695418, 2025.