SECURING DATA PRIVACY IN EDGE CLOUD COLLABORATIVE SYSTEM

A. Abhishek Hulage, B. Nakshatra Gokule, C. Rutuja Hinge, D. Harshad Pachpute

Abstract -This paper explores the challenges and opportunities in edge-cloud collaborative systems for running Deep Learning tasks on resource-constrained IoT devices. It highlights the potential privacy issues introduced by third-party clouds in edge computing. The paper presents two main contributions:

1. It introduces a set of new attacks that allow an untrusted cloud to recover arbitrary inputs without accessing the edge device's data, computations, or system permissions.
2. The paper empirically demonstrates that adding noise as a defense mechanism is ineffective against the proposed attacks. It then proposes two more effective defense methods.

Overall, the study provides insights and guidelines for developing privacy-preserving collaborative systems and algorithms in the context of edge-cloud computing.

I. INTRODUCTION

This project focuses on examining the potential privacy threats to inference data in edge-cloud collaborative systems, and explores both attack and defense perspectives. While previous studies have mainly focused on enhancing the performance and efficiency of such systems, little attention has been paid to their security issues. Our study aims to fill this gap by investigating the confidentiality of raw input data, which is a critical aspect of data privacy. Specifically, we explore two key questions: Can a malicious or compromised cloud recover raw input data that is otherwise only available to the edge device. Can an untrusted cloud accurately recover sensitive data from intermediate values, even without accessing the edge-side model. Contrary to previous claims that edge-cloud collaborative inference systems provide better privacy protection, we demonstrate that an untrusted cloud can still recover sensitive data from intermediate values without accessing the edge-side model. To achieve this goal under different settings, we design a novel set of attack techniques. Our study highlights the importance of considering data privacy in edge-cloud collaborative systems and provides insights for developing more secure and privacy-preserving systems. The second question we address in this paper is: how can the edge devices mitigate privacy leakage from the untrusted cloud. Past work adopted differential privacy to protect the inference data. We hope our findings can guide developers and researchers to design more secure collaborative inference systems

II. LITERATURE REVIEW

1. **Vehicle detection and recognition for intelligent traffic surveillance systems**: Springer Vehicle detection and type recognition based on static images is highly practical and directly applicable for various operations in a traffic surveillance system. This paper will introduce the processing of automatic vehicle detection and recognition. First, Haar-like features and AdaBoost algorithms are applied for feature extracting and constructing classifiers, which are used to locate the vehicle over the input image

2. **Stealing hyperparameters in machine learning:** Hyperparameters are critical in machine learning, as different hyperparameters often result in models with significantly different performance. Hyperparameters may be deemed confidential because of their commercial value and the confidentiality of the proprietary algorithms that the learner uses to learn them. In this work, we propose attacks on stealing the hyperparameters that are learned by a learner. We call our attacks hyperparameter stealing attacks.

3. **Towards reverse engineering black-box neural networks:** Springer Much progress in interpretable AI is built around scenarios where the user, one who interprets the model, has full ownership of the model to be diagnosed. The user either owns the training data and computing resources to train an interpretable model herself or owns full access to an already trained model to be interpreted post-hoc. In this chapter, we consider a less investigated scenario of diagnosing black-box neural networks, where the users can only send queries and read off outputs.

4. **Joint Dnn:** an efficient training and inference engine for intelligent mobile cloud computing Services. Deep learning models are being deployed in many mobile intelligent applications. End-side services, such as intelligent personal assistants, autonomous cars, and smart home services often employ either simple local models on the mobile or complex remote models on the cloud. However, recent studies have shown that partitioning the DNN computations between the mobile and cloud can increase the latency and energy efficiencies.

5. **A principled approach to learning stochastic representations for privacy in deep neural inference:** Abstract INFerence-as-a-Service (INFaaS) in the cloud has enabled the prevalent use of Deep Neural Networks (DNNs) in home automation, targeted advertising, machine vision, etc. The cloud receives the inference request as a raw input, containing a rich set of private information that can be misused or leaked, possibly inadvertently. This prevalent setting can compromise the privacy of users during the inference phase.

III. METHODOLOGY

A. White-box Attack

Adversarial machine learning involves attempting to attack machine learning models by exploiting their vulnerabilities. A white box attack is one such method where the attacker has complete knowledge of the deployed model, including its inputs, architecture, and specific internals like weights or coefficients. This knowledge may also extend to the internal gradients of the model, which are used to inform its decision-making process.

B. Black-box Attacks

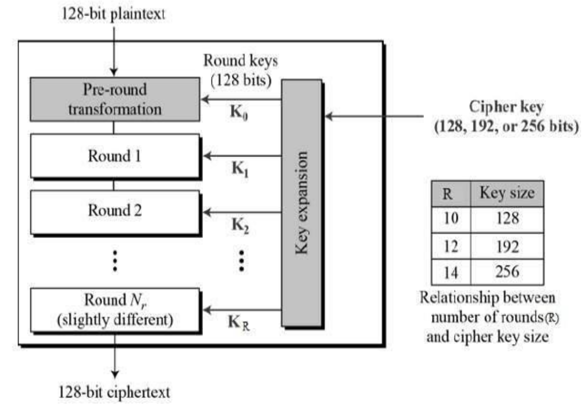
In adversarial machine learning, a black box attack is a method of attacking machine learning models where the attacker only has knowledge of the model's inputs and has access to an oracle. The oracle is a tool that the attacker can use to submit their inputs to and obtain output labels or confidence scores. The term "oracle" is commonly used in this context to refer to an opaque endpoint that provides access to the model's decision-making process. Unlike white box attacks, where the attacker has complete knowledge of the model, black box attacks require the attacker to work with limited information. This makes them more challenging, as the attacker must rely on techniques such as gradient estimation and transferability to generate adversarial examples that are likely to be misclassified by the model. Despite this limitation, black box attacks can still be effective in compromising machine learning models and are an area of active research in the field of adversarial machine learning.

C. Grey Box Attacks

A gray-box adversarial attack and defense framework for sentiment classification. We address the issues of differentiability, label preservation and input reconstruction for adversarial attack and defense in one unified framework. Our results show that once trained, the attacking model is capable of generating high-quality adversarial examples substantially faster (one order of magnitude less in time) than state-of-the-art attacking methods. These examples also preserve the original sentiment according to human evaluation.

IV. ALGORITHM

The Advanced Encryption Standard (AES) is a method of encrypting electronic data that was established by the U.S National Institute of Standards and Technology (NIST) in 2001. Due to its superior strength in comparison to DES and triple DES, AES is currently widely utilized. While it may be more challenging to implement, AES is capable of accepting 128 bits as input and producing 128 bits of encrypted cipher text as output. AES operates based on the substitution-permutation network principle, which involves a sequence of interconnected operations that include substituting and rearranging the input data. AES is distinct from Feistel ciphers due to its iterative nature. Instead, AES relies on a substitution-permutation network that involves a sequence of interconnected operations. Some of these operations replace input with specific outputs (substitutions), while others involve rearranging bits (permutations). What makes AES particularly interesting is that it works with bytes, rather than bits. Specifically, AES treats the 128 bits of plaintext as 16 bytes that are arranged in a four-row by four-column matrix for processing. Unlike DES, the number of rounds used in AES varies and is dependent on the key length. For 128-bit keys, AES uses 10 rounds, while 192-bit keys utilize 12 rounds, and 256-bit keys require 14 rounds. Each round involves using a distinct 128-bit round key that is calculated from the original AES key.



V. MATH

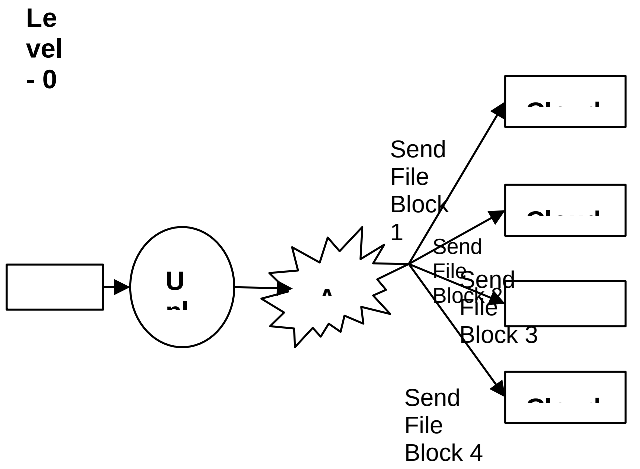
The AES cipher operates on data at the byte level, with a block size of 128 bits or 16 bytes. The input data is processed in 16-byte blocks, which are represented as a 4x4 grid of bytes in column major order. Each round of the AES algorithm consists of four distinct operations:

```
[ b0 | b4 | b8 | b12 |
| b1 | b5 | b9 | b13 |
| b2 | b6 | b10| b14 |
| b3 | b7 | b11| b15 ]
```

1. SubBytes: This step substitutes each byte in the input block with a corresponding byte from a fixed lookup table.
2. ShiftRows: In this step, the bytes in each row of the block are shifted cyclically to the left.
3. MixColumns: Here, each column of the block is transformed using a fixed matrix multiplication operation.
4. Add Round Key: Finally, the current round key is added to the transformed block.

Note that the last round of AES does not include the MixColumns step. Together, these four operations provide strong security and resistance against various cryptographic attacks.

VI. DATA FLOW DIAGRAM

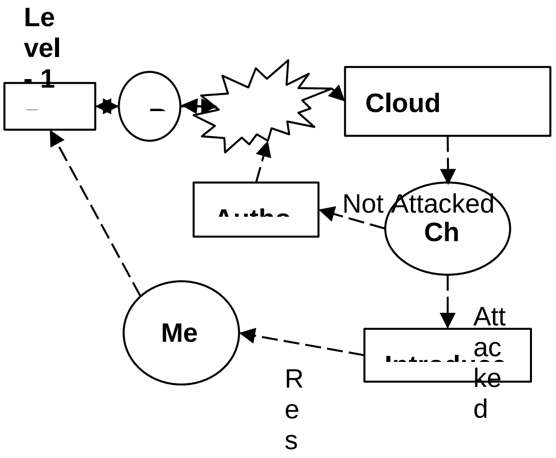


V.1 Level 0 data flow diagram

This scenario describes a system where a data owner uploads files and an auditor sends those files to multiple cloud servers for the purpose of safety and security. The use of multiple cloud servers can provide protection against data loss, data damage, and cybercrime.

After the files are distributed to the different cloud servers, the auditor will verify that each server has received the complete set of files. Once the verification is complete, the auditor will notify the data owner that the files have been successfully uploaded to the cloud servers.

From this point forward, the data owner and authorized personnel can access the files stored in the cloud servers from anywhere with an internet connection, providing convenient access to the data while also ensuring its security and reliability.



V.2 Level 1 data flow diagram

This diagram describes a data owner uploading a file to a cloud server then Edge cloud is connected to multiple cloud servers. The Edge cloud combines all the packets and sends to Remote user. If any unauthorized user is modify the file in a cloud server then Edge cloud regenerate that file and send to Remote user via cloud server

An Auditor is responsible for Data Integrity which is very important in database operations in particular and Data warehousing and Business intelligence in general. Because Data Integrity ensured that data is of high quality, correct, consistent and accessible.

VII. EXPECTED OUTPUT

Once the code runs, the user lands on the entry page of the system, where the user can upload their data. Further the data gets encrypted. Now the data of the user is safe with us.

VIII. CONCLUSION

In this project, we explore the inference data privacy threats in edge-cloud collaborative systems. We discover that an untrusted cloud can easily recover the inference samples from intermediate values. We propose a set of new attack techniques to compromise the inference data privacy under different attack settings. We hope that this study can raise awareness about the importance of inference data privacy protection in edge-cloud systems, and encourage the balancing of privacy protection with usability when designing or implementing such systems.

References

[1] Y. Tang, C. Zhang, R. Gu, P. Li, and B. Yang, "Vehicle detection and recognition for intelligent traffic surveillance systems," Multimedia tools and applications, vol. 76, no. 4, pp. 5817–5832, 2017

[2] B. Wang and N. Z. Gong, "Stealing[9] S. Teerapittayanon, B. McDanel, and H. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in IEEE International Conference on Distributed Computing Systems, 2017 hyperparameters in machine learning," in IEEE Symposium on Security and Privacy, 2018.

[3] S. J. Oh, M. Augustin, M. Fritz, and B. Schiele, "Towards reverse engineering black-box neural networks," in International Conference on Learning Representations, 2018.

[4] A. E. Eshratifar, M. S. Abrishami, and M. Pedram, "Jointdnn: an efficient training and inference engine for intelligent mobile cloud computing services," arXiv preprint arXiv:1801.08618, 2018.

[5] F. Mireshghallah, M. Taram, A. Jalali, A. T. Elthakeb, D. Tullsen, and H. Esmaeilzadeh, "A principled approach to learning stochastic representations for privacy in deep neural inference," arXiv preprint arXiv:2003.12154, 2020.

[6] Z. He, T. Zhang, and R. B. Lee, "Model inversion attacks against collaborative inference," in Proceedings of the 35th Annual Computer Security Applications Conference, 2019, pp. 148–162.

[7] A. Salem, Y. Zhang, M. Humbert, M. Fritz, and M. Backes, "Mlleaks: Model and data independent membership inference attacks and defenses on machine learning models," in Network and Distributed System Security Symposium, 2018

[8] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in IEEE Computer Security Foundations Symposium, 2018.

[9] Z. Yang, J. Zhang, E.-C. Chang, and Z. Liang, "Neural network inversion in adversarial setting via background knowledge alignment," in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 225–240.

[10] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in IEEE Symposium on Security and Privacy, 2019.