Adaptive RL Defense Framework against Cryptographic Cross-Model Adversarial Prompt Injection

Aditya Raj Akshat Gupta Amil Bhagat Ananya Goyal Mehar Khurana
Swati Sharma

September 13, 2024

1 Abstract

Adversarial attacks are a growing threat to Large Language Models (LLMs), particularly through prompt injection attacks where adversarial inputs are crafted to manipulate the model into performing unintended actions. These attacks can be direct (e.g., malicious prompts overriding the model's instructions) or indirect (e.g., hidden prompts in data sources that influence the model's behavior). Thus, there is a pressing need for building appropriate defense frameworks, allowing the LLMs to identify and mitigate these attacks.

Recent works have explored adversarial defenses using static or model-specific techniques; but these approaches often fail to identify and evaluate scenarios where multiple models are similarly affected by one attack or type of attack (i.e., cross-model adversarial attacks). Furthermore, most defenses are designed to handle adversarial prompts crafted using a fixed or known method, making them vulnerable to adaptive attacks where attackers dynamically evolve their strategies based on LLM outputs or encode the adversary to make it unidentifiable as an adversary to the LLM. Motivated by the above discussion, we propose a novel framework for adaptive cryptographic adversarial defense training against cross-model attacks using Reinforcement Learning (RL). Adversarial prompts are encoded using ciphers and a RL-based defender agent is trained to learn to recognise cipher patterns in the prompts, adaptively decrypt the prompt, and mitigate the attack by exposing the malicious intent before the prompts reach the LLMs.

The problem is modelled as an adversarial game which provides enhanced robustness against cross-model attacks by incorporating randomness in cipher selection on the attacker's side, thus ensuring the defender agent is trained on a variety of ciphers. This allows the defender to dynamically adapt to a variety of cryptographic schemes and learn holistic recognition methods, in real time. By enabling real-time adaptation and comprehensive RL-based defense mechanisms, our approach aims to demonstrate higher generalization over static, single-cipher defenses, providing a more resilient solution to evolving adversarial threats in LLMs.

2 Objectives

2.1 Objective 1: Develop an adaptive cryptographic technique to encode adversaries

Drawing inspiration from the adversarial examples explored in [5], we aim to design an LLM agent that generates adversarial prompts by slightly altering clean input prompts to test robustness of the target LLMs to such malicious attacks. Specifically, we intend to utilize attack strategies like prompt-based attacks [15], cipher-based attacks [2], and adversarial prompt adjustments [7].

2.2 Objective 2: Reinforcement learning-based defense mechanism for adaptive decryption

Create a reinforcement learning agent capable of detecting adversarial attacks. Our goal is to enable the agent to adapt to evolving attack strategies, improving its ability to recognize and neutralize hidden malicious intent in real-time, across a variety of attack ciphers and models. Similar to the adversarial training paradigm proposed for Generative Adversarial Networks (GANs) [4], we intend to train the adversarial agent and the RL-based defender in tandem.

2.3 Objective 3: Cross-model vulnerability exploration and mitigation

Explore the transferability of the proposed encoded attacks across different LLM architectures. Previously, cross-model transferability has been explored in the context of code generations tasks [19]. We aim to evaluate the cross-model vulnerability specifically for cipher-based attacks.

3 Literature Review

In the landscape of adversarial attacks on LLMs, significant progress has been made in developing sophisticated attack techniques and corresponding defense mechanisms. Initially, adversarial attacks focused on exploiting vulnerabilities through carefully crafted input prompts designed to manipulate the models into generating harmful or unintended outputs. These adversarial attacks often exhibited transferability, meaning that prompts crafted to exploit one model could also affect others, amplifying the overall threat.

3.1 Generating Adversarial Prompts

3.1.1 Traditional Jailbreak Prompts Generation

[22] introduced methods using greedy and gradient-based search techniques to automatically generate universal adversarial suffixes, known as token-level jailbreaks. These adversarial strings often lack semantic meaning but can influence the LLM's behavior. [13] further investigated the effectiveness of these attacks on LLMs in zero-shot text assessments, demonstrating how adversarial prompts could transfer from smaller models to larger ones. Other approaches, such as Prompt Automatic Iterative Refinement (PAIR) [3], use LLMs to construct jailbreak prompts iteratively, illustrating the increasing sophistication of automated attacks.

Moreover, syntax-based attacks [11] exploit subtle changes in word order or spelling to trigger unintended outputs. Homograph attacks, as outlined by [12], manipulate visually similar characters from different languages to deceive the model. These attacks illustrate how adversaries have grown more sophisticated in their methods, exploiting both linguistic subtleties and model weaknesses.

3.1.2 LLM-Assisted Backdoor Attacks

Backdoor attacks have also emerged as a notable threat. These involve manipulating a model's training or fine-tuning data to elicit harmful responses when specific inputs are provided. [21] proposed ProAttack, a clean-label backdoor attack method that uses prompts as triggers. This eliminated the need for external triggers like rare words or syntactic anomalies, improving upon traditional backdoor attacks.

Yan et al. [18] introduce a sophisticated backdoor attack framework called CODEBREAKER, leveraging LLMs like GPT-4 to inject disguised vulnerabilities into code models, offering a unique approach to bypass even the strongest detection systems.

In the context of backdoor attacks executed through code inputs, they show that traditional backdoor attacks involve injecting malicious payloads into training data, often within non-code sections like comments to evade detection. However, these payloads are easily removable during data pre-processing or detectable during model inference through static analysis tools. CODEBREAKER surpasses these limitations by embedding payloads directly into the functional sections of the code, ensuring the model generates insecure suggestions without being flagged by static analysis or LLM-based detectors.

They employ LLMs to transform the malicious payloads, maintaining their functionality while significantly reducing their detectability. By leveraging the advanced contextual understanding of GPT-4, the CODEBREAKER framework iteratively refines the payloads to evade sophisticated detection mechanisms.

This highlights the necessity of an external defender mechanism that can enable "filtering" of inputs in order to avoid malignant and undesirable outputs, since, as demonstrated by [18] LLMs may not be safe even from themselves.

3.2 Defense Strategies against Adversarial Attacks

In response to the growing number of adversarial attack techniques on large language models (LLMs), various defense strategies have been developed to mitigate their effects. These defense mechanisms are categorized into prompt-based, response-based, and adaptive real-time defense strategies. Each of these plays a critical role in ensuring the robustness and reliability of LLMs.

3.2.1 Prompt-Based Defense Strategies

Prompt-based defense mechanisms focus on manipulating the input prompt before it reaches the LLM to minimize the chances of generating harmful outputs. These methods primarily alter or inspect the input prompt to detect potential adversarial attacks, and can add a defense mechanism to through prompt identification.

SmoothLLM [14] constructs multiple random perturbations of input prompts and aggregates their corresponding outputs. SmoothLLM aims to neutralize adversarial attacks. This adds randomness to reduce the risk of harmful outputs. **LLAMOS** [7] uses a pre-processing technique in which a defense agent 'purifies' potentially adversarial input text while retaining the original meaning. It can dynamically adjust its defense mechanisms using In-Context Learning to address the specific

requirements of the task. **Perplexity Filtering** [1, 13] relies on detecting abnormal perplexity levels in input prompts. Prompts with unusually high perplexity values are flagged as potentially adversarial, disrupting the attack's intent.

3.2.2 Response-Based Defense Strategies

Response-based defenses focus on evaluating and potentially altering the model's output, rather than manipulating the input prompt. These mechanisms are often more dynamic, intervening after the model generates a response to ensure it adheres to ethical guidelines and avoids harmful content.

Techniques such as **AutoDefense** [16] and **Self-Defense Mechanisms** [13] evaluates the model's outputs and replaces any harmful or undesirable content with safer alternatives. This defense actively monitors the model's outputs in real-time, but is based on a post-response-generation strategy.

On the other hand, **Adversarial Fine-Tuning (AFT)** methods fine-tune LLMs using adversarial examples, allowing the model to detect harmful prompts independently. Though resource-intensive, AFT provides robust protection by training the model to recognize malicious patterns.

A more robust and computationally-effective method would be **Adversarial Purification (AP)** which, in contrast to AFT, does not require fine-tuning. Instead, it introduces a separate defense agent to purify adversarial input prompts, thus reducing the computational overhead of the LLM. However, these also may not be adaptive in real-time, and can thus lead to successful injection attacks through sophisticated attacking methods.

3.2.3 Adaptive Real-Time Defense Strategies

Given the dynamic nature of adversarial attacks, adaptive real-time defense strategies have gained more importance. These strategies focus on continuous learning from adversarial patterns and adjusting defense mechanisms accordingly to ensure resilience.

Real-Time Detection and Neutralization [20] build adaptive defenses by constantly learning from new adversarial tactics, allowing the LLM to neutralize evolving threats without retraining. Such strategies involve generalizing across multiple attack methods, thus increasing the robustness of the system. **Adversarial Decryption and Purification** employs real-time decryption techniques along with purification agents, and allow LLMs to neutralize adversarial inputs as they are processed, offering protection against a wide array of potential attacks.

These existing defense strategies represent the ongoing efforts to protect LLMs from adversarial manipulation. While each strategy has its strengths, combining multiple approaches often yields the most robust defense, ensuring the safety and reliability of the models in practice.

3.3 Ciphers in Adversarial Training

Adversarial techniques can also manipulate input data by introducing ciphered (or encoded) instructions to compromise LLMs. Goodfellow et al. [6] demonstrate how adversarial examples could fool machine learning models by making small, unnoticeable modifications to input data. Although such perturbations can be used in defensive strategies as well[9], adding encoded instructions can also lead to the LLM overlooking the malicious intent of the underlying prompt.

This approach of building cipher-based defense systems is part of a growing body of work in adversarial attacks aimed at preventing the misuse of large generative models, ensuring that sensitive or harmful content is not generated, even unintentionally. However, the focus on defending against ciphered attacks is relatively unexplored and needs to be addressed.

3.3.1 Cipher Recognition and Mitigation

As cryptographic defenses became relevant, approaches like cipher recognition and real-time prompt decryption have gained attention. These techniques, grounded in adversarial game theory, seek to recognize adversarial patterns within the prompt structure. Kang et al. [10] highlight how LLMs are increasingly attracting more sophisticated adversaries and attacks, and addressing these attacks require new approaches to mitigations. For instance, recognizing ciphers within the prompts can expose the adversarial intent and prevent the model from executing malicious code. Cipher pattern recognition, as outlined by [8], is one technique that enables defenders to identify hidden adversarial patterns in prompts. Dynamic cipher learning could take this a step further, enabling the defense system to generalize across multiple cipher types, adapting to the ever-changing strategies used by attackers.

Real-time decryption techniques are crucial in neutralizing adversarial prompts. These techniques decrypt prompts before the LLM processes them, preventing adversaries from executing harmful commands hidden within the text. Such cryptographic approaches offer a robust line of defense, as they focus on understanding the structure and intent of the input at a granular level.

4 Methodologies

4.1 Initiating Attacks

The attacker's objective would be to identify and exploit the vulnerability in different LLMs, thus producing cross-model adversaries. In the case of this study, we plan to leverage LLMs to generate adversarial prompts and encode the same using a cipher chosen at random from a pre-determined set of ciphers. This would allow to learn a defense mechanism that is robust to different types of ciphers, and thus prevent a larger set of adversaries.

An attack can be initiated using multiple different strategies. We plan to use cipher-based attacks, wherein a cipher or encoded instructions are added to the prompt that are designed to trigger specific responses from the LLM by exploiting patterns that the model might interpret incorrectly. The use of such ciphers allows attackers to mask their true intentions while still guiding the model towards undesirable outcomes [2]. Combining this with adversarial generation of prompts, wherein the generation strategy is refined based on the response to the prompt, can be particularly effective when targeting LLMs that lack robust defenses against subtle manipulations or that are vulnerable to minor changes in input syntax [15].

The combination of cipher-based attack strategies and adversarial prompt generation results in attacks that can potentially trigger similar malicious responses across different LLMs, thus producing cross-model attacks. The responses from these LLMs to these attacks can then be used to train a defender agent that can decrypt the encrypted adversarial prompt, and expose the malicious intent of the underlying prompt. The LLM can then apply standard defense strategies against adversarial prompts, as described in previous sections.

4.2 Adaptive Cipher Strategy

Introducing randomness into the cipher selection process, where the attacker adapts its strategy by choosing ciphers uniformly at random from a set, allows to develop an even more robust defender agent that can identify and mitigate a wider set of adversarial attacks. The defender agent will learn a holistic decryption strategy for all of the ciphers, and aim to expose malignant information in the adversary.

4.3 Training a Defender Agent

We plan to train a RL-based defender agent using different sets of strategies, including but not limited to Markov Decision Process, Multi-Agent Reinforcement Learning, and Minimax Q-Learning, and compare the performance of these strategies on the specified task.

The underlying training paradigm would be for the attacker and the defender agent to have opposing objectives, thus pushing the defender to find optimal strategies for minimizing breaches across models. Strategies such as "fictitious play" can be used to model the attacker's changing behavior, allowing the defender agent to update its policy dynamically based on the attacker's actions. Finally, once the decrypted version of the prompt is obtained from the defender agent, standard defense strategies such as prompt filtering, perplexity filtering, and dynamic response calibration can be used to prevent unintended and/or malicious responses.

5 Evaluation Criteria

To evaluate the performance of the cryptographic adversarial defense system across multiple models, we define several metrics that assess the system's robustness, adaptability, and efficiency in mitigating cross-model cryptographic adversarial attacks. These metrics have been selected from recent research and are specifically designed to test the effectiveness of a defense system in complex, real-world attack scenarios. Below, we present the key evaluation criteria [17].

5.1 Attack Success Rate (ASR)

The *Attack Success Rate* is a critical metric that quantifies the efficacy of cryptographic adversarial prompts in bypassing the defense system and successfully attacking multiple language models. Formally, we define ASR as:

$$ASR = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{Successful Attacks on Model}_i}{\text{Total Adversarial Prompts Injected on Model}_i}$$

where N represents the number of language models under evaluation. This metric is essential for assessing the overall vulnerability of the models and the defense system's effectiveness in preventing adversarial prompts from infiltrating different architectures.

5.2 False Negative Rate

The *False Negative Rate* measures the proportion of adversarial prompts that the defense system fails to detect across multiple models. It is computed as:

$$FNR = \frac{\text{Number of Missed Attacks}}{\text{Total Number of Attacks Across Models}}$$

This metric is crucial in determining the robustness of the defense system, as a low false negative rate indicates that the system is reliably identifying and mitigating attacks across different architectures. A lower value implies a more secure defense system.

5.3 Defense Passing Rate (DPR)

The *Defense Passing Rate* measures the proportion of adversarial prompts that incorrectly bypass the defense mechanism by being classified as benign, despite being malicious. Formally, we define DPR as:

$$DPR = \frac{\text{Number of Misclassified Adversarial Prompts}}{\text{Total Number of Malicious Inputs}}$$

where the numerator represents the number of adversarial prompts that were not detected by the defense system and thus incorrectly classified as harmless. This metric is essential for evaluating the weaknesses of the defense system, particularly its susceptibility to the adversarial prompts that exploit the system's blind spots.

5.4 Benign Success Rate (BSR)

The *Benign Success Rate* measures the proportion of non-malicious inputs that successfully pass through the defense system without being incorrectly flagged as adversarial. Formally, we define BSR as:

$$BSR = \frac{s}{t}$$

where s denotes the number of benign inputs that correctly pass through the defense filter, and t represents the total number of benign inputs. This metric is critical for evaluating the defense system's ability to avoid false positives, ensuring that benign prompts are not erroneously flagged as adversarial. A high BSR indicates that the defense system is accurate in distinguishing between benign and malicious inputs, ensuring smooth operation without unnecessary intervention.

6 Timeline

6.1 Phase 1: Literature Review and Initial Exploration (Sep 8 - Sep 23)

Conduct a comprehensive literature review on adversarial attacks, defense mechanisms, and vulnerabilities in LLMs. Explore existing cryptographic techniques and reinforcement learning approaches to defense mechanisms. In this phase, we aim to establish a foundational understanding of the current landscape and identify gaps in the field.

6.2 Phase 2: Develop Baselines and Benchmark Prior Work (Sep 23 - Sep 29)

Evaluate state-of-the-art adversarial attack and defense methods as a baseline for our proposed approach. Benchmark existing works against our objectives, focusing on cross-model vulnerability and cryptographic defenses to guide the development of new techniques. Importantly, most prior work does not evaluate the defense mechanisms on cross-model attacks.

6.3 Phase 3: System Design and Development (Sep 29 - Nov 3)

Design and implement an adaptive cryptographic adversarial training paradigm. Develop the reinforcement learning-based defense system by integrating the adversarial training strategy with a decryption overhead, ensuring it adapts to multiple models and attack strategies. Conduct thorough testing and evaluation of the effectiveness of the developed system and assess performance on cross-model attacks.

6.4 Phase 4: Testing and Evaluation (Nov 3 - Nov 14)

Make necessary adjustments to fine-tune the system's performance and generate comparisons to existing methods and baseline models benchmarked earlier. Prepare a final report detailing findings and possible directions of future work.

7 Team Details

- Aditya Raj [2021512]
- Akshat Gupta [2021515]
- Amil Bhagat [2021309]
- Ananya Goyal [2021011]
- Mehar Khurana [2021541]
- Swati Sharma [2021568]

References

[1] Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*, 2023.

[2] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[3] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2015.

[7] Qibin Zhao Guang Lin. Large language model sentinel: Llm agent for adversarial purification. *arXiv preprint arXiv:2405.20770*, 2024.

[8] Maanak Gupta, CharanKumar Akiri, Kshitiz Aryal, Eli Parker, and Lopamudra Praharaj. From chatgpt to threatgpt: Impact of generative ai in cybersecurity and privacy. *IEEE Access*, 2023.

[9] Sidong Jiang, Siyuan Wang, Rui Zhang, Xi Yang, and Kaizhu Huang. Cipher-prompt: Towards a safe diffusion model via learning cryptographic prompts. In *BICS 2023, LNAI 14374*, pages 322–332. Springer, 2024.

[10] Daniel Kang, Xuechen Li, Ion Stoica, Carlos Guestrin, Matei Zaharia, and Tatsunori Hashimoto. Exploiting programmatic behavior of llms: Dual-use through standard security attacks. In *2024 IEEE Security and Privacy Workshops (SPW)*, pages 132–143. IEEE, 2024.

[11] Jiazhao Li, Yijin Yang, Zhuofeng Wu, VG Vydiswaran, and Chaowei Xiao. Chatgpt as an attack tool: Stealthy textual backdoor attack via blackbox generative model trigger. *arXiv preprint arXiv:2304.14475*, 2023.

[12] Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. Hidden backdoors in human-centric language models. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 3123–3140, 2021.

[13] Vyas Raina, Adian Liusie, and Mark Gales. Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment. *arXiv preprint arXiv:2402.14016*, 2024.

[14] Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.

[15] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*, 2019.

[16] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*, 2023.

[17] Zihao Xu, Yi Liu, Gelei Deng, Yuekang Li, and Stjepan Picek. A comprehensive study of jailbreak attack versus defense for large language models. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7432–7449, 2024.

[18] Shenao Yan, Shen Wang, Yue Duan, Hanbin Hong, Kiho Lee, Doowon Kim, and Yuan Hong. An llm-assisted easy-to-trigger backdoor attack on code completion models: Injecting disguised vulnerabilities against strong detection. *arXiv preprint arXiv:2406.06822*, 2024.

[19] Jingwei Yi, Yueqi Xie, Bin Zhu, Keegan Hines, Emre Kiciman, Guangzhong Sun, Xing Xie, and Fangzhao Wu. Benchmarking and defending against indirect prompt injection attacks on large language models. *arXiv preprint arXiv:2312.14197*, 2023.

[20] Yifan Zeng, Yiran Wu, Xiao Zhang, Huazheng Wang, and Qingyun Wu. Autodefense: Multi-agent llm defense against jailbreak attacks. *arXiv preprint arXiv:2403.04783*, 2024.

[21] Shuai Zhao, Jinming Wen, Luu Anh Tuan, Junbo Zhao, and Jie Fu. Prompt as triggers for backdoor attack: Examining the vulnerability in language models. *arXiv preprint arXiv:2305.01219*, 2023.

[22] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.