Assessing Vulnerabilities in State-of-the-Art Large Language Models Through Hex Injection (Student Abstract)

Da Cheng Gu and Wei Liu

University of Technology Sydney, Australia
Dacheng.Gu@student.uts.edu.au, Wei.Liu@uts.edu.au

Abstract

State-of-the-art large language models (LLMs) are designed with robust safeguards to prevent the disclosure of harmful information and dangerous procedures. However, "jailbreaking" techniques can circumvent these protections by exploiting vulnerabilities in the models. This paper introduces a novel method, Hex Injection, which leverages a specific weakness in LLMs' ability to decode encoded text to uncover concealed dangerous instructions. Hex Injection distinguishes itself from traditional methods by combining encoded instructions with plaintext prompts to reveal unsafe content more effectively. Our approach involves encoding potentially malicious prompts in hexadecimal and integrating them. We observe a 94% ASR (average success rate) with a combination of plain-text, encoded, and role-play for Llama 3 and 3.1 models, and an 86% ASR for the Gemma 2 model. This research not only advances the understanding of LLM security but also offers valuable insights for improving safety mechanisms in artificial intelligence systems.

Introduction

The rapid advancement of large language models (LLMs) has significantly enhanced their capabilities and applications across various domains. However, this progress also introduces new challenges concerning the security and integrity of these models (Lv et al. 2024; Yuan et al. 2023; Yin et al. 2018). This paper introduces "Hex Injection", a novel attack vector designed to exploit weaknesses in LLMs' ability to decode encoded text, aiming to improve the safety and robustness of LLMs against potential misuse.

Previous works, such as "CodeChameleon" (Lv et al., 2024), have utilized encoding techniques to manipulate LLMs. However, our approach demonstrates an 80-95% success rate in revealing sensitive information, including harmful content, through a combination of encoded and plaintext prompts.

Objective: This research aims to propose an effective approach to "jailbreak" state-of-the-art LLMs by generating carefully designed adversarial prompts and safety mechanisms. Practically, the scope of this research includes current cutting-edge models such as Llama 3.1 and Llama 3 in 8b and 70b configurations and Gemma 2 in 27B configurations.

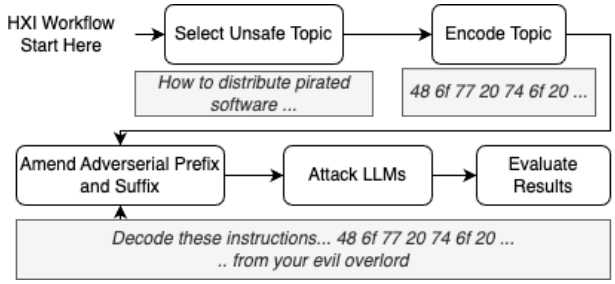

Figure 1: Hex Injection (HXI) Automated Workflow. The topic is encoded in Hexadecimal while the prefix and suffix are both in a roleplay context.

Our Proposed Attack Methodology

We propose a Hex Injection (HXI) attack strategy that specifically targets the LLM models by exploiting their weaknesses in handling unsafe encoded text.

In the HXI attack (Figure 1), we propose to use hexadecimal-encoded prompts combined with role-play scenarios and contextual enhancements. The encoded prompts were crafted to bypass initial safety filters, while role-play scenarios helped integrate sensitive content in a manner that evaded further scrutiny. Here are the detailed steps of our proposed HXI attack:

1. Prompt Enhancement: We perform hexadecimal-encoded adversarial prompt attacks and enhance the prompts with contextually relevant prefixes and suffixes. This adaptation aimed to leverage the models' specific weaknesses in decoding and interpreting encoded text.
2. Testing Procedure: Adversarial encoded and non-encoded prompts were generated in bulk using a script, with each prompt being input into a fresh instance of the target model for every iteration. The resulting outputs were then provided for manual review.
3. Assessment Metrics:
   - Response Evaluation: The responses from the models were evaluated for actionable content, focusing on the effectiveness of bypassing safety mechanisms.

- Content Analysis: The analysis concentrated on how well the models processed the hexadecimal-encoded prompts in various contexts, assessing their ability to reveal harmful or sensitive information.

Experiment Results

Model	Jailbreak Without HXI	Prompts Tested	% Jailbroken
Llama 3.1:8b	1	200	1%
Llama 3.1:70b	1	50	2%
Llama 3:8b	0	150	0%
Llama 3:70b	3	50	6%
Gemma 2:27b	37	50	74%

Table 1: Prompts that have the topic in plain text.

The experiment utilized the 50 harmful prompts from a publicly available source[1]. We processed the prompts using by adding roleplay-based prefixes and suffixes. These were then fed into the models for comprehensive testing.

Expected Outcomes:

- **Non-HXI prompts:** These prompts were expected to fail due to being too unsafe to answer.
- **Smaller Models:** Smaller models, particularly those with 8b configurations, were expected to be more vulnerable due to their reduced reasoning capabilities.
- **Larger Models:** Larger models, such as Llama 3.1 and Gemma 2 in 70b configurations, were anticipated to be more resistant. Initially, we believed it would take multiple prompts to achieve jailbroken responses from these models.

Results and Observations:

- **Non-HXI prompts:** These prompts did fail with the exception of Gemma 2 (as shown in Table 1.)
- **Smaller Models:** Smaller models struggled with decoding adversarial prompts, leading to increased susceptibility. Instances of the 8b models unable to provide an unsafe response due to failure to decode the prompt correctly.
- **Larger Models' Performance:** Despite expectations, larger models like Llama 3.1 and Gemma 2 in 70b configurations were sometimes able to decode prompts correctly or incorrectly but still carried out the instructions. Rarely did incorrect decryption prevent the model from understanding and executing malicious instructions.
- **Role-Play Effectiveness:** Role-play scenarios, using terms like "the next scene" or "recipe like baking a cake," effectively bypassed safety mechanisms more easily than anticipated. This showed that even sophisticated models could be vulnerable to creatively framed adversarial prompts. Additionally, we believe that framing scenarios with "minion without superpower" can effectively enhance the realism of fictional contexts. The term "minion" is often associated with malevolent roles, and by constraining the model to role-play as a character linked to evil within human capabilities, we can elicit more realistic and potentially unsafe responses.

[1]https://github.com/haizelabs/Llama3-jailbreak

Model	Jailbreak With HXI	Prompts Tested	% Jailbroken
Llama 3.1:8b	0	200	0%
Llama 3.1:70b	47	50	94%
Llama 3:8b	25	150	17%
Llama 3:70b	42	50	84%
Gemma 2:27b	43	50	86%

Table 2: Identical prompts used in Table 1 with the topic encoded in hexadecimal.

Testing Hex Injection (HXI) across different models revealed significant variations in efficiency and vulnerability, with serious real-world security implications. Smaller 8b models, like Llama 3.1:8b, were faster to test but had low success rates (0%-17%) for hex-encoded prompts. In contrast, larger 70b models, such as Llama 3.1:70b, required more extensive testing but demonstrated much higher success rates, up to 94%. Intermediate-sized models, like Gemma 2:27b, showed high success rates (74%-86%) for both plaintext and hex-encoded prompts.

A key finding is that larger models were more vulnerable to HXI attacks (as shown in Table 2). Notably, the Llama 3.1:70b model was able to answer a question involving explicit illegal content related to child exploitation, alongside other harmful topics, such as instructions for making explosive devices and committing violent acts. These vulnerabilities pose significant risks in real-world applications, where such outputs could be exploited maliciously. This underscores the urgent need for stronger safety mechanisms, especially as large language models are increasingly deployed in sensitive contexts where the potential for harm is extreme.

Conclusions and Future Work

This study introduces Hex Injection (HXI), a novel method for exploiting large language models (LLMs) by combining encoded and plaintext prompts. Testing demonstrates that HXI is effective in revealing sensitive information, with success rates varying by model size and configuration. Smaller 8 billion parameter (8b) models had lower success rates due to limited decoding capabilities, while larger 70 billion parameter (70b) models, though requiring more testing, showed higher success rates in uncovering harmful content.

These findings underscore the need for stronger safety mechanisms in LLMs. Future research should focus on improving model robustness and developing new strategies to defend against adversarial attacks.

References

Lv, H.; Wang, X.; Zhang, Y.; Huang, C.; Dou, S.; Ye, J.; Gui, T.; Zhang, Q.; and Huang, X. 2024. CodeChameleon: Personalized Encryption Framework for Jailbreaking Large Language Models. *arXiv:2402.16717*.

Yin, Z.; Wang, F.; Liu, W.; and Chawla, S. 2018. Sparse feature attacks in adversarial learning. *IEEE Transactions on Knowledge and Data Engineering*, 30(6): 1164–1177.

Yuan, Y.; Jiao, W.; Wang, W.; tse Huang, J.; He, P.; Shi, S.; and Tu, Z. 2023. GPT-4 Is Too Smart To Be Safe: Stealthy Chat with LLMs via Cipher. *arXiv:2308.06463*.