Cover Letter

I Know What You Asked: Prompt Leakage via KV-Cache Sharing in Multi-Tenant LLM Serving
Wu et al., NDSS 2025

General Topic. Authors are stuying the security of multi-tenant Large Language Model (LLM) serving frameworks, specifically the risks introduced by shared resources such as the Key-Value (KV) cache.

Specific Activity. The research focuses on the security risks of sharing the Key-Value (KV) cache for identical token sequences among multiple users to save memory and computation.

Research Questions.

- **[RQ1]** "How effectively can PROMPTPEEK extract prompts from the LLM server?"
- **[RQ2]** "How many attack requests does PROMPTPEEK need to extract a single prompt?"

Problem. Multi-tenant LLM serving frameworks share the KV cache across users' requests to save memory and computation. However, this sharing creates a cross-tenant side channel. An adversary can craft requests and observe whether the scheduling order changes (due to LPM), thereby inferring whether a candidate token matches a victim's cached prompt. This enables token-by-token prompt reconstruction, threatening the confidentiality of sensitive user input such as personal health data, financial records, or proprietary prompt templates.

Claims. The authors claim that (1) KV-cache sharing creates an exploitable side channel through the LPM scheduling policy; (2) their attack, PROMPTPEEK, can accurately reconstruct user prompts across three scenarios with varying adversary knowledge; and (3) the attack's effectiveness depends on three key factors: GPU memory capacity, concurrent user load, and the number of attack requests.

Evidence. The authors evaluated PROMPTPEEK using a Llama2-13B model on an A100 80G GPU across four distinct datasets: Ultrachat, PromptBase, awesome-chatgpt-prompts, and alpaca-gpt4. They demonstrated successful prompt reconstruction across three scenarios: known templates, known inputs, and no prior background knowledge.

Statistical Analysis. The paper uses descriptive experimental analysis rather than formal inferential statistics. It reports success rate (SR), reversal ratio (RR), requests per input (Req./inp), and requests per token (Req./tok), and compares these metrics across concurrency levels, memory capacities, and attack parameter settings.