A Hallucination Mitigation Scheme in Security Policy Generation with Large Language Models

Jet Wei Goh[1], Kamarul Ridwan Bin Abdul Rahim[1], Nathan Moyses[2], Vetrivel Maheswaran[3], Jaehoon (Paul) Jeong[4], and Tae (Tom) Oh[3]
[1]Singapore University of Technology and Design, Singapore
[2]University of Alicante, Spain
[3]Rochester Institute of Technology, Rochester, NY 14623, USA
[4]Department of Computer Science & Engineering, Sungkyunkwan University, Suwon, Republic of Korea
Email: {goh_jetwei, kamarulridwan_abdulrahim}@mymail.sutd.edu.sg, nrm69@alu.ua.es, {vm6923, thoics}@rit.edu, pauljeong@skku.edu

***Abstract*—Large Language Models (LLMs) can translate high-level security intents into machine-readable policies for cloud security services. However, a single-step LLM prompt often produces invalid hallucinated outputs when generating structured policies. This paper proposes a schema-grounded prompt ensembling pipeline that decomposes policy generation into multiple specialized prompts, combined with an intent relevance filter to pre-screen inputs. Our approach significantly reduces out-of-schema hallucinations and yields syntactically valid, standards-compliant Interface to Network Security Functions (I2NSF) policies from natural language intents.**

***Index Terms*—Large Language Models, Hallucination Mitigation, Prompt Ensembling, I2NSF, YANG, Security Policy Generation.**

I. Introduction

Cloud network security management requires translating high-level natural-language intents into low-level configurations enforcing concrete security policies. The IETF I2NSF framework defines a Consumer-Facing Interface (CFI) YANG data model for standardized policy specification using an Event-Condition-Action structure with endpoint groups and threat intelligence objects. Writing I2NSF CFI policies in XML by hand is labor-intensive and error-prone, requiring deep schema expertise. Prior work by Rodriguez et al. [1] introduced a Security Policy Translator (SPT) that uses an LLM to map natural-language intents to I2NSF CFI policies, showing the feasibility of automation. However, direct LLM prompting in this context faces three key challenges. (1) Schema-level hallucinations: the LLM may invent tags or structures not defined in the CFI schema (unsupported actions, wrong tag names, mis-nested blocks, missing required sections), leading to validation failures. (2) Lack of training data: no large public dataset of intent–policy pairs exists, limiting supervised approaches or retrieval-augmented methods to guide the LLM. (3) No intent filtering: real user queries may be unrelated to security, yet a naive system would still attempt to generate a policy. In this paper, we address these challenges with a hallucination mitigation scheme based on prompt ensembling [2] and schema validation.

Our implementation and synthetic datasets are publicly available in our GitHub repository at https://github.com/jaehoonpauljeong/KICS2026-Group3.

II. Methodology

A. Prompt Ensembling Pipeline

Instead of using a single prompt to directly produce an XML policy, we decompose the task into a sequence of specialized LLM prompt "experts." Each expert focuses on one aspect of the policy. Intermediate outputs are checked against a Schema Reference Table derived from the I2NSF CFI YANG model, ensuring that only valid tags and values appear in the final policy. This staged, schema-grounded design narrows each prompt's scope, reduces hallucinations, and helps ensure the final output is faithful to the intent and compliant with the I2NSF schema as illustrated in Fig. 1.

The pipeline proceeds as follows:

1) **Intent Restatement and Expert Extraction:** The natural-language intent is first normalized into an IF–THEN statement. This canonical form, together with the original purpose, is passed to a small set of LLM prompt "experts" that each extract one component of the policy: (i) events and actions, (ii) contextual conditions (e.g. time, source, destination), (iii) endpoint groups and threat feeds, and (iv) policy metadata (names, language, priority, resolution strategy).
2) **Schema Readiness Checker**: Validate extracted components against a CSV-based Schema Reference Table of the I2NSF CFI YANG model.
3) **Policy Composer**: Compiles the validated components into a full I2NSF CFI XML policy using only approved tags and structures.
4) **Policy Refiner**: Apply a final schema-guided refinement pass to normalize field order and structure and remove residual hallucinated details.

B. Intent Relevance Filtering

Many inputs to a policy assistant will not be valid security policy intents. If such off-domain queries are fed into the LLM generator, it may produce either an obviously invalid XML

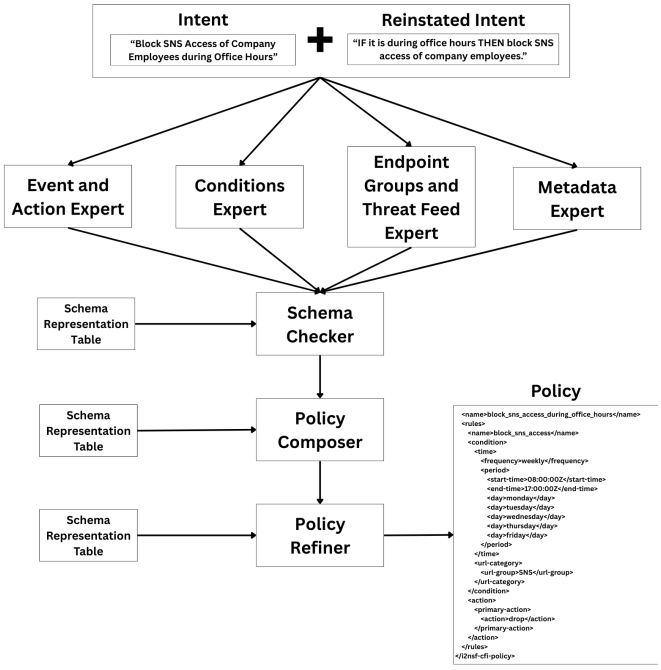

Fig. 1. Architecture of the Prompt Engineering Approach.

(which schema validators can catch) or worse, a plausible-looking but semantically irrelevant policy that could inadvertently be applied. To prevent this, we introduce an intent relevance filter before the generation pipeline. This filter is implemented as an LLM-based classifier prompt that is grounded with schema knowledge (using the same Schema Reference Table) to recognize relevant terminology. If the input is classified as a valid security intent, it proceeds to the prompt ensemble; if not, it is rejected.

III. Performance Evaluation

We evaluated the proposed scheme on a set of synthetic security intents, focusing on three aspects: (1) syntactic correctness of generated policies, (2) qualitative fidelity of output policies for a representative intent, and (3) accuracy of the LLM-based intent relevance classifier.

A. Syntactic Correctness Comparison

We generated 50 synthetic one-sentence security intents and evaluated three methods: (i) a baseline single-step GPT-4o-mini prompt that directly outputs XML, (ii) our prompt ensembling pipeline with GPT-4o-mini, and (iii) the same pipeline with GPT-5-mini. Each configuration was run three times across all 50 intents, and we measured how many resulting XML policies were syntactically valid according to yanglint [3] and the I2NSF CFI YANG schema.

The baseline LLM did not produce any schema-compliant policies. In contrast, the prompt ensembling pipeline achieved on average 52.6% syntactic correctness with GPT-4o-mini and 86% with GPT-5-mini, demonstrating substantial gains from schema-grounded ensembling and further improvements from a stronger backbone model.

B. Qualitative Output Comparison

We use the intent "Block SNS Access during Business Hours" to compare the baseline single-prompt LLM against our prompt-ensembling pipeline. Table I summarizes their compliance with key elements of the I2NSF CFI YANG schema.

TABLE I
I2NSF CFI YANG Schema Properties for the "Block SNS Access during Business Hours" Intent.

XML Policy Aspect	Baseline LLM	Prompt Ensembling
Explicit policy metadata	×	✓
Endpoint groups defined	×	✓
Valid `<firewall>` node under `<condition>`	×	✓
URL category bound to URL group	×	✓
Time context	Partial	✓
Primary action value (`drop`)	✓	✓
Schema-valid	×	✓

Across similar intents, single-step outputs often require substantial manual repair, whereas ensembling typically yields policies that are deployment-ready or need only minor adjustments.

C. Evaluation of LLM Classifier for Intent Relevance Filtering

On 100 labeled test intents (50 valid security intents, 50 irrelevant queries), our LLM-based relevance classifier achieved 91% accuracy with GPT-4o-mini and 97% with GPT-5-mini. This confirms that a simple LLM prompt can effectively prefilter invalid or unsafe inputs.

IV. Conclusion

This paper proposes a schema-grounded prompt ensembling method for generating I2NSF CFI security policies from natural-language intents using LLMs. By splitting policy generation into specialized experts and validating their outputs against a Schema Reference Table derived from the I2NSF CFI YANG model, it significantly reduces hallucinated tags, invalid values, and structural errors compared to single-step prompting.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant (No. RS-2024-00398199) and the National Research Foundation of Korea (NRF) grant (No. 2023R1A2C2002990) funded by the Korea government (MSIT). Jaehoon (Paul) Jeong is the corresponding author.

References

[1] M. L. Rodriguez, J. A. Berasategui, and J. P. Jeong, "Security policy generation for cloud-based security services using large language model," in *Proc. KICS Winter General Conf.*, 2025, [Online]. Available: http://iotlab.skku.edu/publications/domestic-conference/KICS-2025-Winter-LLM-Based-Security-Policy-Generation.pdf.

[2] S. Schulhoff, "Prompt ensembling," DiVeRSe and AMA for Accurate LLM Results, 2023, [Online]. Available: https://learnprompting.org/docs/reliability/ensembling.

[3] CESNET, "libyang: YANG data modeling language library," 2024, [Online]. Available: https://github.com/CESNET/libyang.