

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

Jet Wei Goh¹, Kamarul Ridwan Bin Abdul Rahim¹, Nathan Moyses², Vetrivel Maheswaran³,
Jaehoon (Paul) Jeong⁴, and Tae (Tom) Oh³

¹Singapore University of Technology and Design, Singapore

²University of Alicante, Spain

³Rochester Institute of Technology, Rochester, NY 14623, USA

⁴Department 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

