IntentGuard: Securing MCP-Enabled LLM Agents via Post-Decision Semantic Plan Verification

Extended Abstract

Haoran Cheng
University of Science and Technology of China
Hefei, China
chenghaoran@mail.ustc.edu.cn

Yunhao Yao
University of Science and Technology of China
Hefei, China
yaoyunhao@mail.ustc.edu.cn

Jinke Song
The Hong kong University of Science and Technology
Hongkong, China
jikesog@gmail.com

Zhiqiang Wang
University of Science and Technology of China
Hefei, China
sa21221041@mail.ustc.edu.cn

Lan Zhang
University of Science and Technology of China
Hefei, China
zhanglan@ustc.edu.cn

ABSTRACT

The Model Context Protocol (MCP) enables LLM agents to discover and invoke tools dynamically, but it also introduces a new threat: Tool Metadata Poisoning, where adversarial tool descriptions induce semantically incorrect yet syntactically valid invocations. We propose the Intention-Plan Consistency Paradigm, which protects agents via post-decision semantic plan verification rather than relying on potentially compromised agent reasoning. Building on this paradigm, we introduce VISTA, combining information isolation to construct a minimal trusted context and hierarchical semantic assessment to validate tool choice and parameter provenance. We also present MCPIntentEval, a benchmark for intent-alignment verification in MCP-enabled agents. Experiments show that VISTA consistently improves over strong baselines and robustly detects inconsistent tool-invocation plans.

KEYWORDS

LLM Agents, Model Context Protocol (MCP), Tool Metadata Poisoning, Semantic Verification

ACM Reference Format:
Haoran Cheng, Yunhao Yao, Jinke Song, Zhiqiang Wang, and Lan Zhang. 2026. IntentGuard: Securing MCP-Enabled LLM Agents via Post-Decision Semantic Plan Verification: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/PJGH6650

1 INTRODUCTION

Large Language Models (LLMs) have rapidly evolved into powerful coordinators in intelligent systems, enabling agents to integrate diverse data sources, services, and devices through natural-language reasoning [1, 3, 9, 15]. The Model Context Protocol (MCP) further accelerates this trend by allowing agents to discover and invoke external tools dynamically, without tightly coupling the agent to specific tool implementations [2, 4–6].

New security threat. MCP's open ecosystem exposes an new attack surface: agents rely on untrusted natural-language tool metadata from external servers. If such metadata is maliciously manipulated, the agent may produce a semantically incorrect plan while each individual tool call remains syntactically valid and permission-compliant. We refer to this threat as *Tool Metadata Poisoning*.

Limitations of existing defenses. Recent LLM-agent security techniques mainly address malicious user inputs or overtly unsafe outputs [12, 16]. They are less effective when the attack is hidden in an apparently benign chain of legitimate tool calls. This challenge is amplified in MCP settings because tools are provisioned dynamically and metadata is inherently untrusted, which can undermine context-based reasoning [7] and static tool-profile verification [10].

Our insight. We propose the *Intent–Plan Consistency Paradigm*: instead of hardening potentially poisoned reasoning, we validate whether a generated execution plan is semantically aligned with the user request. Compared with open-ended plan generation, intent–plan checking is a constrained, discriminative judgment over a given plan, and it complements sanitization and filtering defenses [7, 8, 10, 11]. Building on this paradigm, we introduce VISTA, a post-decision semantic verification framework that intercepts an agent plan before execution. VISTA combines (i) *information isolation* to construct an Isolated Verification Context (IVC) containing only the user request and candidate plan, and (ii) *hierarchical semantic assessment* to validate both tool selection and parameter provenance. Our main contributions are as follows:

- We introduce the Intent–Plan Consistency Paradigm for securing MCP-enabled agents via post-decision semantic plan verification.
- We design VISTA, which combines information isolation and hierarchical semantic assessment to detect both malicious intent deviations and benign anomalies.
- We present MCPIntentEval and evaluate VISTA on seven datasets, achieving up to 13.13% higher accuracy and 9.48% higher F1, with up to 90% fewer false positives.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/PJGH6650

Table 1: Main results on the MCPIntentEval benchmark.

Method	$\mathcal{D}_{Q8^*}$		$\mathcal{D}_{Q8^\dagger}$		$\mathcal{D}_{Q14^*}$		$\mathcal{D}_{Q14^\dagger}$		$\mathcal{D}_{P4}$		$\mathcal{D}_{M7}$		$\mathcal{D}_{G9}$	
	Acc.	F1-Score	Acc.	F1-Score	Acc.	F1-Score	Acc.	F1-Score	Acc.	F1-Score	Acc.	F1-Score	Acc.	F1-Score
MCP-Guard	60.74	58.33	70.48	65.17	59.60	56.71	76.28	72.21	39.86	39.17	57.91	54.83	66.67	54.10
MCIP	71.28	72.37	77.62	75.52	70.86	72.85	79.46	76.67	72.37	78.81	65.94	66.18	77.43	72.58
LLM-D	85.68	89.46	83.67	86.32	85.14	89.10	78.44	81.76	90.59	93.99	81.23	86.51	71.96	75.51
LLM-D+	83.26	87.56	83.07	85.45	85.77	89.31	79.48	82.33	86.59	91.24	86.36	89.66	79.45	81.48
VISTA	98.39	96.84	95.95	96.38	98.60	97.64	99.53	97.93	99.10	97.35	98.07	94.95	96.23	96.30

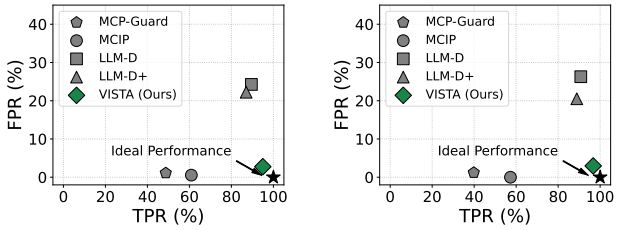

Figure 1: TPR and FPR of VISTA and baselines (left: $\mathcal{D}_{Q8^\dagger}$; right: $\mathcal{D}_{Q14^*}$).

2 METHODOLOGY

We implement the Intent–Plan Consistency Paradigm with **VISTA** (**V**erification via **I**solated **S**eman**T**ic **A**ssessment), a post-decision verification framework that checks whether a complete MCP plan is semantically aligned with the user query q. VISTA operates only on the finalized plan and is therefore independent of the (potentially poisoned) planning context.

2.1 Step-level Verification

Given a plan

$$T_{\text{call}} = \langle a_1, \ldots, a_n \rangle, \quad a_i = (\tau_s^i, p^i),$$

where τ_s^i is the selected tool and p^i its parameters, VISTA applies step-level verification and aggregates the results:

$$V(q, T_{\text{call}}) = 1 \iff \forall i,\ V_{\text{step}}(q, a_i) = 1.$$

2.2 Isolated Verification Context

For each action a_i, VISTA constructs an *Isolated Verification Context* (IVC) that contains only the trusted information needed for assessment: (1) user query q; (2) server name; (3) selected tool τ_s^i; (4) the tool's description; and (5) the generated parameters p^i. All other tool metadata are discarded, so the Judge LLM cannot be influenced by poisoned descriptions of unrelated tools.

2.3 Dual-Dimension Semantic Validation

Within the IVC, a Judge LLM evaluates each action a_i along two semantic dimensions, and a step is accepted only if both pass.

Tool Intent Validation. We first check whether the selected tool τ_s^i is the appropriate tool to fulfill the user request q. The Judge (i) extracts the core intent from q, (ii) summarizes the behavior of τ_s^i from its metadata and parameters, and (iii) decides whether the tool behavior matches the user intent. This detects both malicious cross-tool hijacking and non-adversarial tool selection errors.

Parameter Provenance Validation. We then verify that all arguments passed to τ_s^i are trustworthy. Each parameter p_j^i is assigned one of three provenance labels: *User_Query* (explicitly stated in q), *Tool_Default* (specified as a default in the schema of τ_s^i), or *Illegitimate* (derived from any other source such as hallucinated or injected content). For parameters labeled *User_Query*, the Judge also checks that their semantic meaning is preserved without distortion.

For each step a_i, VISTA constructs an IVC, runs both checks, and accepts the plan only if all steps satisfy them, i.e., $V(q, T_{\text{call}}) = 1$.

3 EVALUATION

Dataset. We evaluate on MCPIntentEval, derived from MCPTox: 7,967 tool-call responses from seven agent models over 45 MCP servers and 353 tools. We relabel each plan as intent-aligned or intent-misaligned (including both maliciously induced and benign errors) and deduplicate samples for evaluation.

Baselines and metrics. We compare against MCP-Guard [13], MCIP [7], LLM-D and LLM-D+ [17], and report Accuracy, F1, Precision, Recall (TPR), and FPR.

Implementation. We use Qwen3-8B [14] as the Judge Model and run all experiments on a single NVIDIA A100 GPU (3 runs).

Overall performance. Table 1 summarizes the main results on MCPIntentEval. VISTA consistently outperforms all baselines across seven datasets, achieving F1-scores above 94.9% and strong accuracy. On $\mathcal{D}_{Q14^*}$, VISTA reaches 97.9% F1, outperforming the best baseline (89.3%) by a relative improvement of about 9.6%.

TPR/FPR trade-off. Figure 1 reports the detection trade-off on two representative datasets (left: $\mathcal{D}_{Q8^\dagger}$; right: $\mathcal{D}_{Q14^*}$). VISTA maintains high TPR while keeping FPR low, yielding a favorable balance between detection sensitivity and false alarms.

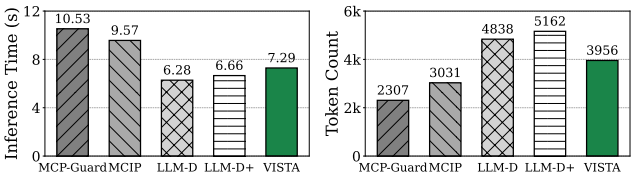

Figure 2: Overhead of LLM reasoning (left: time cost; right: token cost).

ACKNOWLEDGMENTS

Lan Zhang is the corresponding author. This research was supported by the National Natural Science Foundation of China (No. 62441228) Science and Technology Tackling Program of Anhui Province (No. 202423k09020016).

REFERENCES

[1] Rishabh Agrawal, Murtaza Asrani, Hadi Youssef, and Apurva Narayan. 2025. SCMRAG: Self-Corrective Multihop Retrieval Augmented Generation System for LLM Agents. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 50–58.

[2] Anthropic. 2024. Introducing the Model Context Protocol. https://www.anthropic.com/news/model-context-protocol Accessed: 2025-07-27.

[3] Abul Ehtesham, Aditi Singh, Gaurav Kumar Gupta, and Saket Kumar. 2025. A survey of agent interoperability protocols: Model context protocol (mcp), agent communication protocol (acp), agent-to-agent protocol (a2a), and agent network protocol (anp). *arXiv preprint arXiv:2505.02279* (2025).

[4] Junfeng Fang, Zijun Yao, Ruipeng Wang, Haokai Ma, Xiang Wang, and Tat-Seng Chua. 2025. We Should Identify and Mitigate Third-Party Safety Risks in MCP-Powered Agent Systems. *arXiv preprint arXiv:2506.13666* (2025).

[5] Yongjian Guo, Puzhuo Liu, Wanlun Ma, Zehang Deng, Xiaogang Zhu, Peng Di, Xi Xiao, and Sheng Wen. 2025. Systematic analysis of mcp security. *arXiv preprint arXiv:2508.12538* (2025).

[6] Zikang Guo, Benfeng Xu, Chiwei Zhu, Wentao Hong, Xiaorui Wang, and Zhendong Mao. 2025. MCP-AgentBench: Evaluating Real-World Language Agent Performance with MCP-Mediated Tools. *arXiv preprint arXiv:2509.09734* (2025).

[7] Huihao Jing, Haoran Li, Wenbin Hu, Qi Hu, Heli Xu, Tianshu Chu, Peizhao Hu, and Yangqiu Song. 2025. Mcip: Protecting mcp safety via model contextual integrity protocol. *arXiv preprint arXiv:2505.14590* (2025).

[8] Weidi Luo, Shenghong Dai, Xiaogeng Liu, Suman Banerjee, Huan Sun, Muhao Chen, and Chaowei Xiao. 2025. AGrail: A Lifelong Agent Guardrail with Effective and Adaptive Safety Detection. In *ACL (1)*. Association for Computational Linguistics, 8104–8139.

[9] Panagiotis Lymperopoulos and Vasanth Sarathy. 2025. Tools in the Loop: Quantifying Uncertainty of LLM Question Answering Systems That Use Tools. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2645–2647.

[10] Brandon Radosevich and John Halloran. 2025. Mcp safety audit: Llms with the model context protocol allow major security exploits. *arXiv preprint arXiv:2504.03767* (2025).

[11] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024. "Do Anything Now": Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. In *CCS*. ACM, 1671–1685.

[12] Kun Wang, Guibin Zhang, Zhenhong Zhou, Jiahao Wu, Miao Yu, Shiqian Zhao, Chenlong Yin, Jinhu Fu, Yibo Yan, Hanjun Luo, et al. 2025. A comprehensive survey in llm (-agent) full stack safety: Data, training and deployment. *arXiv preprint arXiv:2504.15585* (2025).

[13] Wenpeng Xing, Zhonghao Qi, Yupeng Qin, Yilin Li, Caini Chang, Jiahui Yu, Changting Lin, Zhenzhen Xie, and Meng Han. 2025. MCP-Guard: A Defense Framework for Model Context Protocol Integrity in Large Language Model Applications. *arXiv preprint arXiv:2508.10991* (2025).

[14] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388* (2025).

[15] Yingxuan Yang, Qiuying Peng, Jun Wang, Ying Wen, and Weinan Zhang. 2025. Unlocking the Potential of Decentralized LLM-based MAS: Privacy Preservation and Monetization in Collective Intelligence. In *AAMAS*. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2896–2900.

[16] Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pan, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, et al. 2025. A survey on trustworthy llm agents: Threats and countermeasures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 6216–6226.

[17] Kaijie Zhu, Xianjun Yang, Jindong Wang, Wenbo Guo, and William Yang Wang. 2025. MELON: Provable Defense Against Indirect Prompt Injection Attacks in AI Agents. *arXiv preprint arXiv:2502.05174* (2025).