

Security Analysis of Agentic Communication Protocols: Model Context Protocol (MCP) and Agent-to-Agent (A2A)

Zhang Yiyue*, Minseok Kim*, Hyungjoon Koo**

Department of Computer Science and Engineering, Sungkyunkwan University (Graduate
student*, Professor**)



Agentic Communication Protocol

✓ Why External Tools & Communication Protocols are Essential

- A language model have limitations (e.g., lack of real-time data, domain-specific expertise, or API access)

Y what is the weather of tomorrow in Seoul?

I don't have access to real-time weather data or current weather forecasts. To get the weather forecast for tomorrow in Seoul, I'd recommend checking:

- A weather website like Weather.com or AccuWeather
- A weather app on your phone
- Local Korean weather services like the Korea Meteorological Administration
- Searching "Seoul weather forecast" in your web browser

Can not answer without real-time data

Y what is the weather of tomorrow in Seoul?

I'll search for tomorrow's weather forecast in Seoul to give you the most current information.

Seoul weather forecast tomorrow June 6 2025

결과 10개 ▾

Let me search for more specific information about tomorrow's weather forecast for Seoul.

Seoul Korea weather June 6 2025 forecast temperature rain

결과 10개 ▾

Based on the search results, here's what you can expect for tomorrow's weather in Seoul:

The temperatures in Seoul in June are comfortable with low of 19°C and high up to 25°C

Seoul weather in June 2025..., which means tomorrow should have pleasant temperatures ranging from about 19°C (66°F) in the morning to around 25°C (77°F) during the day.

Research Objectives

✓ Two Communication Protocols

- **MCP (Model Context Protocol)**: Standardized framework for AI models to interact with **external tools and APIs**
- **A2A (Agent-to-Agent Protocol)**: JSON-RPC framework enabling secure **inter-agent collaboration**

✓ Systematic Security Analysis

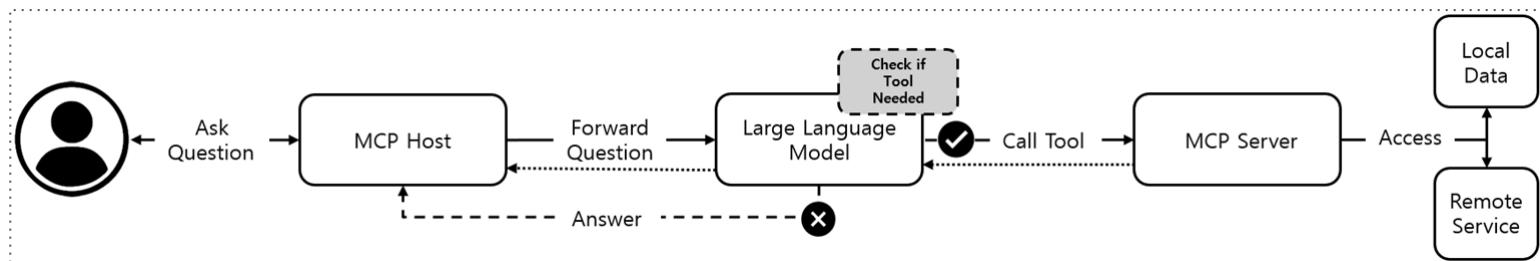
- Identify vulnerabilities in MCP and A2A protocols
- Propose mitigation strategies for secure deployment
- Support secure, scalable AI agent collaboration

MCP (Model Context Protocol)

✓ Workflow

- An user submits a query via the MCP Host
- A language model evaluates the query to determine if external tools are needed
- If required, the model delegates tool invocation to the MCP Server
- MCP Server accesses appropriate external resources (e.g., APIs, local files)
- Retrieved results are returned, integrated into the response, and delivered to the user

Key Components
MCP Host:
– User interface
LLM:
– Decision engine
MCP Server:
– Tool orchestrator

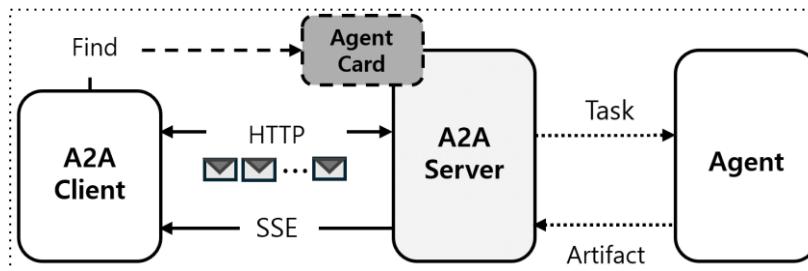


A2A (Agent-to-Agent Protocol)

✓ Workflow

- A2A Client initiates a task request and searches for suitable agents via the Agent Card
- A2A Server matches the request to an appropriate agent based on capabilities
- The selected agent executes the assigned task and generates artifacts
- Real-time updates are sent back to the A2A Client via SSE (Server-Sent Events)
- Final artifacts are delivered to the requesting client

Key Components
A2A Client:
– Task initiator
Agent Card:
– Capability card
A2A Server:
– Task router
Agent:
– Task processor



Security Analysis on Agentic Protocols

✓ Attack Vectors by Protocol Scope

- **A2A-Specific:** Delegation Misuse
- Both A2A and MCP: Misconfigured Authentication, Excessive Permissions, Prompt Injection, Vulnerability Propagation, Token Theft

Attack Vector	Mitigation	
	MCP	A2A
Delegation Misuse	Not applicable	Use DPoP-bound tokens scoped per agent and task
Misconfigured Authentication	Delegating authentication to hardened identity providers, encrypt tokens, enforcing strict lifetimes, and adopting resource oriented authorization models	
Excessive Permissions	Enforce least-privilege principles through scoped permissions, attribute-based access controls (ABAC), and automated permission auditing	
Prompt Injection	Sanitize the prompts, deploy prompt shielding and execution environment segregation	
Vulnerability Propagation	Monitor CVEs, revoke access for vulnerable components, apply trust scoring and periodic reauthorization	
Token theft	Bind tokens (to mTLS/DPoP where applicable), rotate keys, shorten lifetimes, monitor usage	

A2A-Specific Security Risks: Delegation Misuse

✓ Problem

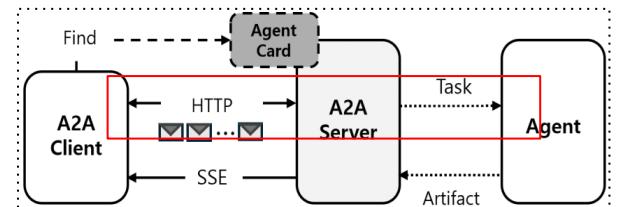
- Inadequate token propagation controls in multi-agent workflows

✓ Impact

- Excessive privilege sharing between parent/child agents
- Confused-deputy attacks
- Uncontrolled privilege escalation

✓ Mitigation

- Use Demonstration of DPoP (Proof-of-Possession) bound tokens
- Enforce per-task, per-agent token scoping
- Maintain verifiable delegation lineage



Cross-Protocol Security Risks: Misconfigured Authentication

✓ Problem

- Flawed identity management in OAuth or external credential systems

✓ Impact

- Token theft → Unauthorized resource access

✓ Mitigation

- Delegate authentication to hardened IdPs (Identity Providers)
- Use envelope encryption for tokens
- Apply short token lifetimes
- Enforce resource-scoped authorization models



Cross-Protocol Security Risks: Excessive Permissions

✓ Problem

- Broad or unchecked access granted to servers or agents

✓ Impact

- Unauthorized data exfiltration or system modification

✓ Mitigation

- Enforce least-privilege policies
- Use ABAC (Attribute-based Access Control)
- Enable automated permission audits



Cross-Protocol Security Risks: Prompt Injection

✓ Problem

- Malicious prompt content causes agents to behave unexpectedly

✓ Impact

- Unintended actions, data leaks, tool misuse

✓ Mitigation

- Sanitize inputs and enforce schemas
- Use prompt shielding techniques
- Isolate execution environments



Cross-Protocol Security Risks: Vulnerability Propagation

✓ Problem

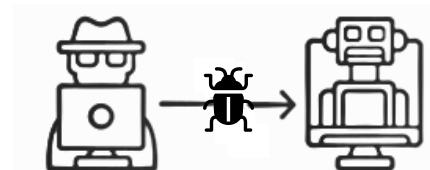
- Security flaws spread across agent networks via dependencies

✓ Impact

- One compromised agent jeopardizes entire workflow

✓ Mitigation

- Integrate CVE (Common Vulnerabilities and Exposures) monitoring
- Revoke tokens for vulnerable agents
- Apply trust scoring and periodic reauthorization



Cross-Protocol Security Risks: Token Theft

✓ Problem

- Intercepted or leaked tokens grant attackers system access

✓ Impact

- Data breaches, persistent backdoor entry

✓ Mitigation

- Use mTLS/DPoP token binding
- Shorten token lifetimes
- Enable usage monitoring
- Rotate cryptographic keys regularly
- Apply token transparency logs



Conclusion

✓ Limitations

- Analysis based on specifications, not empirical deployment
- Focus on expected behavior, not adversarial scenarios

✓ Contributions

- Uncover potential attack vectors (risks) in MCP and A2A protocols
- Propose mitigation strategies for secure deployment

Thank you.

