Securing MCP Interactions at Scale

CS4991 Capstone Report, 2025

Yechan Kim
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
msw3jg@virginia.edu

ABSTRACT
MCP has become the leading standard for enabling AI agents to call external tools; yet today's verification system relies on static bearer tokens that are insecure and ill-suited for sensitive use cases. As agents begin handling finances, personal data, and other critical tasks, stronger safeguards are required. I propose a system where each AI agent is assigned an Agent Verification Number, similar to the way credit cards authenticate human transactions. Using cryptographic keypairs, agents would sign requests, which are then validated through a centralized, Visa-like gateway before being processed by the target tool. This approach prevents key leakage and replay attacks, ensures accountability, and creates a scalable trust layer for agent–tool interactions. Beyond security, the framework enables broader adoption of personal AI agents, bringing us closer to a world where everyone can rely on their own digital assistant to eliminate repetitive work and accelerate daily life. Future work will focus on refining the cryptographic protocol, integrating secure key storage, and expanding interoperability across MCP-based systems.

1. INTRODUCTION

2025 has been declared the "year of AI agents," as major technology companies rapidly adopt autonomous assistants to streamline workflows and eliminate repetitive tasks. A crucial enabler of this ecosystem is the Model Context Protocol (MCP), introduced by Anthropic in September 2024. MCP is not a technology itself, but a protocol that allows agents to connect to servers and request tools, whether hosted locally or remotely. Developers expose capabilities through MCP servers, and agents invoke them to perform tasks, creating a growing ecosystem of interoperable tools across organizations.

During my internship at IBM in summer 2025, I built MCP tools for agents and experienced firsthand how fragile the current security model is. MCP today relies on static bearer tokens for verification. To enable my tools, I had to store API keys in local environment files and manually manage access, an insecure and impractical process. Worse, once an agent holds a token, it has broad, unchecked access to the tool. While this may be tolerable for low-risk tasks such as sending emails, it becomes unacceptable as agents begin handling sensitive operations like financial transactions or personal data processing.

As MCP becomes more widely adopted across industries, the limitations of this token-based authentication model pose significant risks. Bearer tokens offer no granular control or verification beyond initial access, meaning any compromised token grants full permissions until manually revoked. This creates potential for large-scale exploitation, data leakage, or unauthorized system control, especially as AI agents begin

to automate critical business workflows. Furthermore, MCP interactions are inherently dynamic. Agents can chain multiple tools, invoke third-party APIs, or even trigger other agents, magnifying the attack surface and making it increasingly difficult to trace responsibility when something goes wrong. Without a secure, standardized method of verifying agent identity and intent for each interaction, organizations may struggle to trust these systems at scale. This lack of trust forms one of the primary barriers to broader enterprise adoption of autonomous AI agents.

2. RELATED WORKS

Song et al. (2025) provided one of the first systematic safety audits of the Model Context Protocol. Their paper introduced a taxonomy of attacks—such as tool poisoning, puppet attacks and prompt injection—that exploit the way MCP servers expose tools to agents. They showed how maliciously crafted tool descriptions or metadata could cause agents to behave in unsafe ways, even when the tools themselves were not compromised. This research validated my concern that MCP's current bearer-token system is insufficient, since it does not authenticate individual requests or prevent tampered instructions from being executed. Reading this work directly shaped my thinking that transaction-level security, not just server-level trust, is required for agents to be reliable.

Hasan et al. (2025) expanded this perspective with a large-scale empirical study of nearly 1,900 open-source MCP servers. Their findings revealed that over 5% of these servers suffered from MCP-specific vulnerabilities such as metadata injection and description tampering. Beyond identifying vulnerabilities, their study emphasized the scale of the problem: as MCP adoption accelerates, even a small percentage of insecure servers could pose major systemic risks. This reinforced my idea that MCP needs stronger, standardized verification mechanisms before it can safely support sensitive agent tasks. The study convinced me that ad-hoc fixes are not enough; what is needed is a structural redesign of how agents and tools verify each interaction.

Huh et al. (2017) demonstrated a proven model for solving similar problems in a different domain: mobile payments. Their analysis of Apple Pay showed how device-bound keys and dynamic per-transaction cryptograms protect each payment against replay, cloning and theft. Importantly, the cryptographic signature is generated within a secure enclave, meaning the user's credentials are never directly exposed. This work strongly influenced my proposal by showing how a large-scale, consumer-facing system can use cryptography to guarantee transaction authenticity while still offering seamless user experience. It suggested to me that a Visa-like gateway for MCP, where every agent request is cryptographically signed and verified, could scale in the same way Apple Pay scaled globally.

3. PROJECT DESIGN

The proposed system introduces a new verification framework that brings transaction-level security to the Model Context Protocol (MCP). The goal is to build a foundation of trust across the growing ecosystem of AI agents and tools, ensuring that every request can be verified, audited and safely executed without exposing sensitive credentials. Inspired by the structure of financial networks, the framework follows a three-layer model consisting of the Agent, the Gateway and the MCP Tool Server.

3.1 System Architecture

The overall architecture is designed to resemble a payment network, where each component plays a specialized and interdependent role. Every AI agent that participates in the system is issued a unique

Agent Verification Number (AVN) during registration, serving as a persistent and auditable identity across all interactions. The agent is then provisioned with a public/private keypair securely stored within its runtime environment or a local key vault.

When an agent wishes to invoke an MCP tool, the request first passes through a Gateway Service instead of going directly to the tool server. The Gateway acts as a verification hub and enforcer of access policies. It maintains a secure registry of all issued AVNs, active cryptographic keys, permission scopes and rate limits. Each incoming request is evaluated against these parameters to ensure that the action being attempted is authorized, valid and compliant with the organization's security policies.

In addition to verifying requests, the Gateway abstracts tool credential management from the agents themselves. MCP tools often require sensitive API keys or credentials to interface with external systems; under this model, those credentials are held exclusively by the Gateway. This separation ensures that even if an agent or its environment is compromised, no external secrets are exposed. The Gateway also maintains a secure transaction ledger that records all requests and responses in an immutable format, allowing administrators to audit usage patterns, detect anomalies, and trace actions back to verified agents.

3.2 Cryptographic Verification

At the heart of the framework is a lightweight cryptographic verification process that authenticates each agent-tool interaction. Each AI agent uses its private key to sign the contents of every request it sends—specifically the tool name, request parameters, a timestamp, and a randomly generated nonce. This ensures that the request cannot be tampered with or replayed at a later time.

Upon receiving a signed request, the Gateway retrieves the corresponding public key from its AVN registry and verifies the signature. It checks that the message hash matches the provided payload and that the timestamp and nonce fall within a valid window. Only if all verification steps succeed does the Gateway issue a short-lived authorization token and forward the validated request to the target MCP Tool Server. This guarantees that no unsigned or altered request can ever reach the tool.

To make this process seamless for developers, the system includes a dedicated Agent SDK, a lightweight software library that automates the signing and communication process for agents. The SDK handles key generation, key rotation, signature creation and nonce management transparently. When an agent is deployed, the SDK registers it with the Gateway to obtain an AVN and corresponding keypair. From that point forward, all MCP requests made through the SDK are automatically wrapped in signed payloads, meaning developers do not have to implement custom security logic.

The SDK also includes built-in recovery and rotation capabilities. If an agent's environment is reset or compromised, the SDK can automatically request new keys and re-register the AVN with the Gateway without disrupting active tool sessions. In this way, the SDK acts as both a security abstraction layer and a developer convenience layer, ensuring that robust cryptographic practices are consistently applied without adding development complexity.

By combining the SDK's automation with the Gateway's centralized verification, the system ensures that even large networks of autonomous agents can operate securely at scale. Each request is treated as an independent, verifiable event—comparable to a transaction in a financial network—creating a robust chain of trust between agents, Gateways, and tools.

3.3 Design Requirements and Challenges

Implementing this design requires careful attention to scalability, compatibility and interoperability. The framework must integrate smoothly with existing MCP schemas so developers can adopt it without rewriting their tools or agent logic. Because MCP is an open standard with many implementations, the Gateway must handle varying message formats, tool types and hosting environments.

Performance is another major consideration. The Gateway must be capable of verifying thousands of signatures per second without introducing noticeable latency. Load balancing, caching of agent metadata, and parallel verification pipelines will be essential to maintaining responsiveness under heavy workloads.

A final challenge lies in establishing cross-organizational trust. As more companies deploy their own Gateways, defining a shared standard for inter-gateway verification—similar to the way banks and card networks interoperate—will be critical for universal adoption. These challenges are significant but achievable and addressing them will allow MCP to evolve into a secure, scalable foundation for the next generation of AI agents and autonomous applications.

4. ANTICIPATED RESULTS

The proposed verification framework is expected to substantially enhance the security, accountability, and scalability of the Model Context Protocol (MCP) ecosystem. By introducing an intermediary Gateway and eliminating static bearer tokens, the system will allow organizations to verify every agent action at a transaction level while isolating sensitive tool credentials from the agents themselves. Each request will generate a verifiable record tied to a unique Agent Verification Number (AVN), providing a permanent audit trail of agent activity. This improvement could reduce unauthorized or replayed tool calls by more than 90 percent while providing administrators with real-time visibility into agent activity, policy compliance and security anomalies.

Beyond improving security, the framework is anticipated to increase developer efficiency and trust in AI automation. The Agent SDK automates key generation, signing and registration, reducing setup time for secure MCP integrations from hours to minutes. This ease of adoption encourages developers to follow strong security practices by default, while enabling enterprises to deploy autonomous agents confidently. The Gateway's centralized logging and analytics features will also improve incident response and regulatory compliance, allowing organizations to audit agent decisions with the same rigor used in financial transaction systems. In the long term, this architecture lays the groundwork for a global trust network for AI agents, one capable of supporting personal and enterprise-level assistants that securely perform complex, repetitive, or sensitive digital tasks on behalf of humans.

5. CONCLUSION

This project demonstrates how a cryptographically verified transaction model can transform the Model Context Protocol (MCP) into a secure foundation for the next generation of AI agents. By replacing static bearer tokens with a verifiable identity and request-signing framework, the system enables every interaction to be authenticated, logged, and trusted. The introduction of the Gateway Service creates a consistent trust boundary that protects tool credentials, prevents unauthorized access, and ensures clear accountability at scale. This advancement is not just a technical improvement but a necessary shift in how autonomous systems can safely operate

within organizations and across the broader digital ecosystem.

Beyond the technical contribution, this project also had a significant personal impact on my understanding of secure system design. I gained practical insight into cryptographic principles, public-key infrastructures, and how large-scale protocols are engineered for reliability and safety.

The broader implication of this work is its potential to reshape how society interacts with AI. As verification becomes more robust and transparent, individuals and organizations will be more willing to trust agents with sensitive information and financial tasks. Personal AI agents could handle purchases, manage data flows, schedule operations, and perform complex repetitive work without constant human oversight. In this way, the framework lays the foundation for a future where AI agents are not just tools but trusted digital collaborators that accelerate daily life while preserving privacy and security.

6. FUTURE WORK

Future work will focus on the adoption and standardization of this verification model across the MCP ecosystem. For this framework to realize its full potential, existing organizations that operate MCP servers must integrate Gateway verification into their infrastructure. Establishing a shared protocol for inter-gateway communication will be critical, allowing different companies and platforms to recognize and trust agent verification numbers issued by one another. This process mirrors how global payment networks achieved interoperability through unified standards and certification systems.

Additionally, technical development should expand to include prototype implementations and real-world testing of the Agent SDK and Gateway API. Future iterations may explore federated trust models, secure key storage through hardware enclaves, and integration with major cloud providers to encourage enterprise adoption. Long term, the success of this system depends on creating an open-source specification that developers and organizations can easily implement. With widespread adoption, this architecture could become the standard security layer for personal and enterprise AI agents, enabling a new era of secure, autonomous collaboration between humans and machines.

REFERENCES

Song, Y., Li, A., & Wang, P. (2025). The MCP safety audit: Threat modeling and attack taxonomy for AI agent tool protocols. arXiv:2506.02040 [cs.CR]. https://arxiv.org/abs/2506.02040

Hasan, S., Zhang, Y., Lin, J. & Chen, X. (2025). A large-scale empirical study on the security and maintainability of model context protocol servers. arXiv:2506.13538 [cs.SE]. https://arxiv.org/abs/2506.13538

Huh, J., Kim, Y. & Kim, J. (2017). Securing mobile payments with dynamic cryptograms: An analysis of Apple Pay. In Proceedings of the Network and Distributed System Security Symposium (NDSS ’17). The Internet Society. https://www.ndss-symposium.org/wp-content/uploads/2017/09/usec2017_02_1_Huh_paper.pdf