DMSC Working Group
Internet-Draft
Intended status: Standards Track
Expires: 20 July 2026

H. Zhang
China Telecom

16 January 2026

Security Analysis of Multi-agents Secured Communication and Limitations of Existing Protocols
draft-zhang-dmsc-mas-communication-00

Abstract

Multi-agents systems (MAS) increasingly cooperate through workflow, orchestrated, and mesh communication patterns. While existing Internet protocols provide confidentiality and endpoint authentication, they were not designed for agent-native semantics such as dynamic identity, computation-bounded requests, context integrity, and intermediary trust. This document analyzes security risks in agent communication and identifies limitations in widely deployed protocols (TLS, HTTP, MQTT,A2A and etc.) and developer framework (AutoGen and etc.). This document intend to establish a common problem statement and gap analysis to inform future IETF standardization.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 20 July 2026.

Copyright Notice

Table of Contents

1. Introduction

AI agents are autonomous software entities capable of interacting with other agents, tools, and services to perform multi-step reasoning and task execution. Emerging MAS are deployed across heterogeneous infrastructures, often spanning administrative domains. Current Internet security mechanisms primarily protect communication channels and endpoints. However, they do not fully address agent-specific security semantics, including dynamic identity, workload-based abuse, semantic context integrity, and trust in intermediaries. This document provides: [IoA].

* A brief threat analysis of agent communication based on collaboration patterns.

* A classification of core security risks.

* A gap analysis highlighting the limitations of protocols and developer framework.

The scope of this document focuses on communication security for distributed AI agents, including:Agent-to-Agent (A2A), Agent-to-Orchestrator (A2O), Agent-to-Tool/Service (A2T). It does not cover model training security, data governance, or ethical issues.

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] .

3. Terminology

The following terms are defined in [architecture] .

* Agent: An autonomous software entity capable of perception, planning, decision-making, and execution.

* DMSC: Dynamic Multi-agent Secured Collaboration. The framework and infrastructure enabling secure and efficient collaboration among dynamic agents.

4. Core Communication Security Risks in MAS

MAS orchestration mechanisms include three primary patterns[IoA]: Workflows rely on predefined, linear, or conditional task sequences, exhibiting predictable sequential control flows and a pipeline topology; Hierarchical Planning utilizes a centralized orchestrator for task decomposition and intelligent routing, resulting in a star or tree topology with highly concentrated control and decision authority; and Multi-Agent Collaboration features a decentralized, peer-to-peer architecture where agents interact dynamically within a mesh or swarm topology, distributing both control flow and decision-making capabilities.

From a security perspective, workflows face risks primarily related to context integrity, as communication is limited to adjacent agents, making the system vulnerable to payload injection or corruption despite clear identity boundaries. Hierarchical Planning presents significant Single Point of Failure (SPOF) and Man-in-the-Middle (MITM) risks due to its centralized nature; a compromise of the orchestrator can lead to global impact, necessitating End-to-End Encryption (E2EE) and agent-payload binding. Meanwhile, Multi-Agent Collaboration, characterized by fuzzy identity and trust boundaries, is susceptible to computational Denial of Service (DoS) attacks as well as context injection and pollution, requiring strict peer-level identity validation, context signing, and workload control.

* Identity and Trust Boundaries

Agent-based systems rely on dynamically instantiated, short-lived entities whose identities are often assigned at runtime. Unlike traditional services, agents do not map cleanly to long-lived hosts,

domains, or organizational identities. In collaborative or mesh topologies, agents may be created, delegated, or terminated on demand, and may act on behalf of other agents or users. This introduces identity ambiguity and weak trust binding: it becomes unclear who an agent represents, what authority it has, and whether its identity is stable over time.

* Context Integrity and Semantic Consistency

Agent communication is not limited to stateless requests; it often carries evolving conversational state, task context, intermediate reasoning artifacts, and shared memory objects. While transport-layer security can protect messages in transit, it does not ensure that the semantic context remains intact across multiple hops, intermediaries, or storage layers. In workflows and collaborative meshes alike, context can be truncated, reordered, injected, or selectively modified without cryptographic detection. This creates risks of context poisoning, instruction injection, and silent manipulation of agent reasoning processes.

* Orchestrator and Broker Trust

In hierarchical and brokered architectures, a central orchestrator or message broker mediates most or all communications. This entity becomes a single point of failure and a powerful trust anchor. If compromised, misconfigured, or simply overly privileged, the orchestrator can inspect, modify, suppress, or replay messages between agents. Even when transport security is used, the orchestrator typically terminates connections and therefore has full access to plaintext. This introduces MITM risks and undermines E2E security guarantees.

* Unpredictable Workload and Resource Exhaustion

Agent interactions frequently involve expensive computation such as large-model inference, retrieval-augmented generation, or tool execution. In collaborative and mesh settings, any peer may request high-cost operations from another agent. Existing protocols do not account for computational asymmetry between requester and responder, nor do they provide mechanisms to express, negotiate, or enforce resource budgets. As a result, agents are vulnerable to DoS attacks that exploit legitimate-looking but computationally expensive tasks.

* Architectural Amplification of Risk

The above risks manifest differently across communication patterns. In sequential workflows, the primary concerns are identity binding and preservation of task context across hops. In orchestrated

systems, central control amplifies risks of message manipulation and single points of failure. In decentralized collaboration, dynamic peer discovery and many-to-many communication exacerbate identity ambiguity, context pollution, and DoS exposure. Together, these patterns demonstrate that agent communication security is not merely a transport problem, but an architectural and semantic one.

5. Limitations of Existing Protocols in Addressing Agent Communication Security

This section analyzes why commonly used communication protocols and frameworks— specifically TLS, HTTP/gRPC, the A2A protocol, and AutoGen— do not adequately mitigate the risks identified in Section 4. As highlighted in draft-li-dmsc-inf-architecture-01 [architecture], multi-agent systems require network infrastructures to support agent-centric identity management, capability-aware communication, and cross-domain collaboration— requirements that existing protocols fail to address, leading to resource exhaustion risks. The focus is not on implementation flaws, but on structural gaps between protocol assumptions and the requirements of agent-based systems.

* Inadequate Support for Dynamic Agent Identity

TLS, HTTP/gRPC with OAuth or TLS authentication mechanisms bind identity to endpoints, certificates, or client identifiers. These constructs assume relatively stable entities such as servers, users, or devices. They do not natively support ephemeral, software-defined agents that are created on demand, act under delegated authority, or represent composite roles. While the A2A protocol introduces explicit AgentID fields, these identifiers are transmitted without mandatory cryptographic binding to trust anchors, delegation chains, or verifiable credentials. In AutoGen, agent identity is typically managed at the application layer using tokens or configuration files, lacking standardized, interoperable identity semantics. Consequently, none of these protocols provide a robust mechanism to express who an agent is, whom it represents, and under what authority it operates.

* Lack of End-to-End Context Protection

Existing protocols prioritize transport security rather than message-level or context-level integrity. TLS ensures confidentiality and integrity only between adjacent endpoints; once a message is decrypted, any intermediary, broker, or application layer can modify the payload without detection. HTTP/gRPC similarly protect the channel but do not require payload signing or semantic validation. A2A defines structured message objects but does not mandate E2E encryption or cryptographic signing of context elements. AutoGen

inherits the properties of its underlying transports, with no built-in enforcement of message or context integrity. As a result, none of these systems can guarantee that conversational state, instructions, or task context have not been altered across multiple hops.

* Over-Privileged Orchestrators and Brokers

In orchestrated and brokered architectures, existing protocols implicitly assume that the intermediary is fully trusted. TLS termination at a gateway, HTTP reverse proxies, MQTT brokers, and agent orchestrators in both A2A-based and AutoGen systems all have access to plaintext messages. This means that the orchestrator or broker can become an undetectable MITM, contradicting the distributed trust assumptions often made in multi-agent designs.

* Absence of Computation-Aware Communication Controls

None of the evaluated protocols incorporate concepts of computational cost, resource budgeting, or workload fairness. A request that triggers a trivial database lookup and one that initiates a multi-step reasoning chain over a large language model are treated identically at the protocol level. MQTT quality-of-service levels ensure delivery semantics, not execution cost. A2A and AutoGen allow agents to request complex tasks from peers without standardized mechanisms to declare, negotiate, or enforce limits on CPU, memory, inference time, or external tool usage. This structural blind spot leaves agents vulnerable to resource exhaustion attacks that appear legitimate at the message layer.

* Mismatch Between Agent Architectures and Protocol Assumptions

Collectively, TLS, HTTP/gRPC and MQTT were designed around service-oriented or message-oriented paradigms in which endpoints are relatively stable, intermediaries are trusted, and payload semantics are outside the protocol’s scope. Agent-based systems violate these assumptions: identities are fluid, trust is contextual and delegated, messages carry evolving semantic state, and computation itself becomes an attack surface. The result is a systematic gap: even when these protocols are correctly implemented, they cannot provide the E2E identity assurance, context integrity, intermediary minimization, and resource-aware controls required for secure multi-agent communication.

6. Conclusion

This document focuses on the communication security of multi-agent systems, systematically identifying critical gaps between existing communication protocols (such as TLS, HTTP/gRPC, MQTT, A2A, and AutoGen) and the security requirements of dynamic agent networks. Key issues identified include inadequate support for ephemeral and dynamic agent identities, lack of E2E semantic context protection, over-privileged intermediaries (brokers/orchestrators) leading to MITM risks, absence of computation-aware communication controls that render agents vulnerable to DoS attacks, and architectural amplification of these risks across different collaboration patterns (workflows, hierarchical planning, and decentralized mesh collaboration). Collectively, these gaps demonstrate that secure communication in multi-agent systems is not merely a transport-layer problem but a systematic challenge involving identity semantics, context integrity, and resource management.

7. Security Considerations

This document suggests that secure agent communication requires a new class of protocol capabilities that extend beyond traditional network and application-layer security. These include: (1) agent identity semantics that support ephemeral and delegatable identities independent of hosts or domains; (2) computation-aware communication mechanisms that allow agents to declare, negotiate, and enforce resource consumption limits, mitigating high-cost request abuse; (3) semantic context protection, through mandatory signing and optional encryption of payloads and context windows to preserve integrity across multiple hops; and (4) end-to-end trust in mediated topologies, enabling brokers or orchestrators to route messages without accessing or modifying their contents. Together, these directions outline a path for future standardization: defining agent-aware identity and delegation models, specifying message- and context-level security envelopes, and integrating resource governance into communication protocols.

8. IANA Considerations

TBD

9. Acknowledgement

TBD

10. Normative References

[architecture]
Li, X. and A. W, "draft-li-dmsc-inf-architecture-01", 9 January 2026.

[IoA] Idan, H., "Securing-Agentic-Applications-Guide-1.0", 28 July 2025.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

Author’s Address

Haodi Zhang
China Telecom
Tangxia Street, Tianhe District
Guangzhou
Guangzhou, 510000
China
Email: zhanghaodi@chinatelecom.cn