Poster: Collaboration Fingerprinting Reveals What Adaptive Multi-Agent LLM Systems Process on the Wire

Marlena Flüh
flueh@comsys.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Klaus Wehrle
wehrle@comsys.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Jan Pennekamp
jan.pk@comsys.rwth-aachen.de
RWTH Aachen University
Aachen, Germany

Abstract

Multi-agent LLM systems promise improved reasoning by collaboration of specialized agents through a sequence of API calls. This collaboration is also visible on the wire and opens new channels for leakage. Despite encrypted communication, a passive on-path observer can still learn the number, order, size and timing of the API calls. This metadata is generated by the system as a function of the input it processes. In adaptive multi-agent systems, the collaboration structure itself is chosen by the input, and that choice is visible in the traffic pattern. We coin this threat *collaboration fingerprinting*. Using MDAgents, a state-of-the-art medical multi-agent framework, we demonstrate on two datasets that a passive observer can recover the assigned case complexity and narrow a case's diagnosis to a coarse clinical group, solely from collaboration metadata. A case's clinical complexity determines the complexity of the collaboration. This dependency results in an inference-time privacy channel, unaddressed by existing content-level defenses. We thus argue that a multi-agent system's observable network traffic must be treated as part of its confidentiality boundary.

CCS Concepts

• **Security and privacy** → **Network security**; • **Computing methodologies** → **Multi-agent systems**.

Keywords

Collaboration Fingerprinting, Traffic Fingerprinting, Multi-Agent LLM Systems, Side-Channel Attack

ACM Reference Format:
Marlena Flüh, Klaus Wehrle, and Jan Pennekamp. 2026. Poster: Collaboration Fingerprinting Reveals What Adaptive Multi-Agent LLM Systems Process on the Wire. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3830454.3846427

1 Introduction

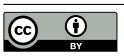
Multi-agent LLM systems decompose a task across specialized agents that collaborate at inference time [10]. In adaptive systems, this collaboration is not fixed in advance but chosen at runtime from the input, varying which agents participate and how many rounds they take [10]. They can outperform fixed pipelines [3], and their execution path becomes a function of the data they process.

The prevailing privacy assumption in multi-agent systems is that encrypted communication protects sensitive data. For example, in clinical decision-making, this sensitive data includes patient cases [3], whose protection is mandated by regulations such as GDPR and HIPAA. We show that encrypted communication does not prevent an adaptive multi-agent system from leaking case complexity, and thus whether an encounter is routine or serious, through network traffic metadata.

In fact, metadata side channels under encryption are well established for communication systems. Website fingerprinting, for instance, recovers which encrypted page a user loaded from packet-level metadata alone [5, 6]. Recent work extends metadata exploitation to LLM systems, each observing a single model's traffic. A passive network observer can reconstruct responses from token lengths exposed by packet sizes [11], infer prompt topics from packet-size and inter-arrival-time sequences [4], and recover token sizes and private user attributes from streaming traces [7]. Such leakage can be framed as arising not from the model but from the system-level components surrounding it [1]. In multi-agent systems, the coordination layer that decides which agents act and in what order [10] is such a component, and it produces a structured, multi-stage trace of inter-agent exchanges.

Related Work. Whisper Leak [4] infers conversation topics from encrypted size and timing sequences, but its signal comes from the text a single model generates, not from how a system chooses to process the input. AgentPrint [13] fingerprints which agent a user interacts with from user-to-agent traffic and profiles a user attribute from usage accumulated over many sessions. Its signal is the fixed traffic signature of an agent's identity, and its target is a property of the user. In contrast, we read the leg between the system and the LLM service (Fig. 1), and our signal is that an adaptive system collaborates differently on each case, so the leak reveals a per-case attribute of the input being processed. Closer to the multi-agent setting, AgentLeak [12] shows that inter-agent messages leak far more than final outputs, but it measures the leakage of message content, not metadata alone. A recent healthcare-security SoK [8] names a network adversary for single-model deployments who infers personal health information properties from traffic metadata under encryption, but labels it as an open risk rather than measuring it. Thus, no prior work studies whether the metadata of adaptive collaboration alone, between a multi-agent system and its LLM service, leaks sensitive information of the case it processes.

Contributions. We introduce collaboration fingerprinting, a novel multi-agent system side channel in which an adaptive system's own execution path becomes an observable signal. Because



CCS '26, The Hague, Netherlands

ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/3830454.3846427

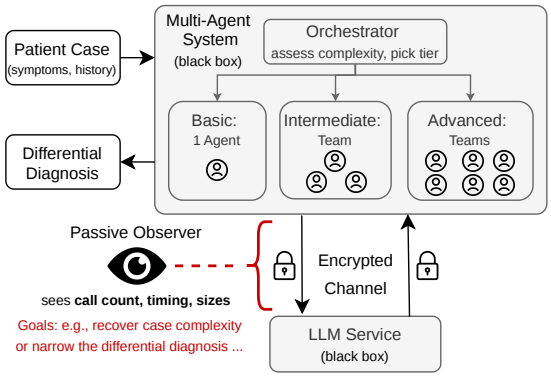


Figure 1: Threat model, instantiated on MDAgents. The system routes each patient case to one of three collaboration tiers by assessed complexity, each tier producing a distinct API-call pattern on the system-to-service wire. A passive observer sees call metadata but not the encrypted content, and recovers the case's complexity, revealing whether a case is routine or serious.

the leak is generated by how the system structures its collaboration for a case rather than by the content it exchanges, it persists under encryption, and defenses that protect message content remain ineffective. Moving from theory to practice, we exemplarily demonstrate this leak on MDAgents, a medical multi-agent framework, where collaboration metadata alone recovers the assigned case complexity on MedQA and narrows a case's diagnosis to a coarse clinical group on DDXPlus. A passive observer on a hospital-to-cloud path therefore learns clinical information about an individual without breaking encryption or tampering with the system. The efficiency gained by adapting computation to the input is paid for in inference-time privacy.

2 Threat Model

We consider a passive, on-path network observer between the system and the LLM service (externally hosted or deployed inside the organization, e.g., a hospital-to-cloud path [8]). This adversary records all traffic but cannot decrypt, inject, or modify it. Such a vantage point requires no compromise of either endpoint. Any network operator or transit ISP on the path observes this link in the ordinary course of operation. Under TLS, content stays confidential and only ciphertext-level metadata is exposed. A client resolves the host, opens a TCP connection to port 443, and runs a TLS handshake. This connection can also be used for subsequent calls. All application bytes travel in TLS records whose 5-byte headers (content-type, length) are plaintext over an encrypted payload. So the observer learns without decryption the endpoint IP/port, the TCP connection-open events, their timing and count, and the content-type, length, and timing of each record per direction. From this information, we derive per-call message sizes, timing, and the sequence and structure of agent calls. The observer aims to map a case's call metadata to a sensitive attribute of that case (e.g., the patient's likely diagnosis), a confidentiality breach despite encryption. Following standard side-channel assumptions [5, 6], the adversary trains a classifier on traces collected offline from a comparable system, with no access to contents or ground-truth labels at attack time. Whenever the collaboration structure depends on the input, the call pattern reveals that structure. In MDAgents, this structure is a difficulty-dependent tier: a single agent (basic), one team (intermediate), or multiple teams (advanced). The recovered call pattern therefore reveals case complexity (Fig. 1). We evaluate two adversaries that differ in how much they observe. The more powerful case-level adversary assumes traffic is separable per case, while the single-call adversary makes no such assumption. Recovering case boundaries from the wire remains future work. MDAgents' per-agent API clients make connection-open events a plausible signal, though concurrent cases and reused connections would obscure it.

3 Measurement Methodology

We capture the traffic of a running multi-agent system, extract the features a passive observer could obtain, and train a classifier to recover the target attribute.[1] We instantiate this methodology on MDAgents as one example of an adaptive system to demonstrate the threat of collaboration fingerprinting in a realistic setting.

Traffic Capture. We run MDAgents against an LLM service over TLS and capture the traffic on the system-to-service link with `tcpdump`, which corresponds to what our passive observer sees. Each API call is delimited by the application's own request/response timestamps, allowing us to slice the packet trace into per-call windows. The adversary observes neither these timestamps nor the resulting boundaries, and would have to recover call boundaries from the wire. Within each window, we read the cleartext TLS record headers to separate handshake records from application-data records, and sum the application-data records per direction. For every call, this includes the on-wire request and response sizes, per-direction packet counts, inter-call and response latency, and whether the call opened a fresh TLS connection. All sizes are measured on the wire and verified against the client's own serialized request bodies to ensure they are not application-layer proxies.

Feature Extraction. A case is defined as a single query processed by the system. Each case comprises several API calls. We evaluate two adversaries. The single-call adversary sees one isolated call and uses only its wire features (request/response application-data size, packet counts, latency, connection-open flag). The case-level adversary additionally sees all calls belonging to a case and aggregates them (call count, and per-case sums, maxima, and means of the same features). Routing tier, agent role or token counts are not visible to the classifier, as they are not observable on the wire.

Classification. We attack two targets of increasing sensitivity, the routing tier on both MedQA [2] and DDXPlus [9], and the diagnosis itself on DDXPlus only (MedQA carries no diagnosis label). For both targets, a gradient-boosted tree classifier (LightGBM) maps the wire-observable features to the target. We evaluate with stratified 5-fold cross-validation grouped by case, and report macro-F1 against a majority-class floor for the imbalanced tier task and against the random-guess level for the balanced diagnosis task.

[1] Code: https://github.com/COMSYS/collaboration-fingerprinting

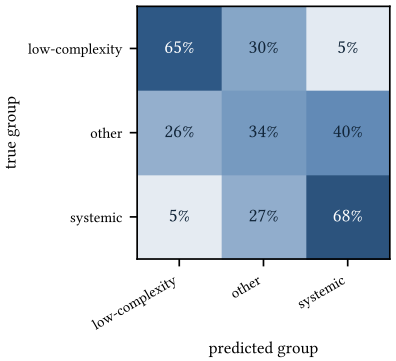


Figure 2: DDXPlus group recovery, case-level adversary. Row-normalized confusion between low-complexity, systemic, and *other* (four diagnoses without coherent fingerprint). The two clusters are confused in only ≈5 % of cases in either direction. Metadata recovers a case's coarse clinical complexity, even though the exact diagnosis is only weakly recoverable.

4 What the Wire Reveals

We run two independent experiments that show leakage at two levels of sensitivity: *Tier Recovery* and *Diagnosis Recovery*. First, routing tiers are recoverable on both MedQA and DDXPlus, highlighting that traffic leaks case complexity. Second, on DDXPlus the same traffic groups diagnoses by coarse clinical severity, while the exact diagnosis remains only weakly recoverable.

Tier Recovery. MDAgents routes each MedQA case to one of three tiers of increasing complexity, basic, intermediate, or advanced. Across 1,272 cases and 11,850 calls the tiers are heavily imbalanced (1,160 basic cases), so we report macro-F1 against a 0.312 majority-class floor. The case-level observer recovers the tier without error (macro-F1 1.000). This result is over-determined because MDAgents' fixed control flow means the call count alone already separates the tiers. The single-call observer is the more informative case. It still recovers the tier well above the floor (macro-F1 0.747) from one isolated call with no session context. The signal is joint across features. Request size alone reaches macro-F1 0.427 and the non-size features alone still reach macro-F1 0.521. Basic- and intermediate-tier request sizes overlap almost completely (AUC 0.508). So the routing decision is visible in a call's shape and timing, not its length. On DDXPlus the same pattern holds against a 0.285 floor, with 0.740 for a single call and 0.994 case-level.

Diagnosis Recovery. We draw 600 balanced cases from DDXPlus (50 each, 12 diagnoses) and target the diagnosis itself, a sensitive attribute that is harder to recover than tier routing. Both adversaries beat chance, with case-level macro-F1 0.206 and single-call 0.138 against a ∼0.083 random guess floor. As in the tier experiment, aggregating a case's calls is more informative than a single call. For the case-level adversary, the true diagnosis falls in the top five of twelve in 64.8 % of cases and in the top three in 47.0 %. Beyond this ranking, the more telling result is that errors stay within clinically coherent groups. The diagnoses separate into two clusters, a low-complexity one (otitis media, URTI, bronchitis) and a systemic one (pneumonia, pulmonary embolism, anemia, acute HIV, localized edema), confused in only ≈5 % of cases in either direction (Fig. 2). The remaining four diagnoses do not cluster. The clusters track deliberation rather than clinical similarity, since the systemic diagnoses share little, yet all drive longer collaborations. An observer thus learns whether an encounter was routine or serious without recovering the diagnosis, which already narrows a patient's condition and constitutes health information under GDPR and HIPAA.

5 Conclusion and Future Work

We highlighted that encrypting the system's communication with its LLM service does not hide how a multi-agent system processes a case. Because an adaptive system tailors its collaboration to the input, the resulting traffic pattern is itself a signal. Exemplarily, on MDAgents, it recovers the assigned complexity tier without any access to content. It groups diagnoses by coarse clinical severity, though we cannot separate diagnosis-specific signal from the complexity that drives the routing. Collaboration fingerprinting therefore places a system's observable traffic inside its confidentiality boundary. We measure one framework against one LLM service under clean network conditions, without concurrent cases and with given case boundaries. Thus, broader generalization remains open. We expect that closing this channel requires shaping the collaboration structure itself, since content-level padding leaves it intact. Established countermeasures from anonymous communication networks appear promising, and we plan to evaluate them alongside how the leak scales with adaptivity across frameworks.

References

[1] Edoardo Debenedetti, Giorgio Severi, Nicholas Carlini, Christopher A. Choquette-Choo, Matthew Jagielski, Milad Nasr, Eric Wallace, and Florian Tramèr. 2024. Privacy Side Channels in Machine Learning Systems. In *USENIX SEC*.

[2] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What Disease Does This Patient Have? A Large-Scale Open Domain Question Answering Dataset from Medical Exams. *Applied Sciences* 11, 14 (2021).

[3] Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik Siu Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae Won Park. 2024. MDAgents: An Adaptive Collaboration of LLMs for Medical Decision-Making. In *NeurIPS*.

[4] Geoff McDonald and Jonathan Bar Or. 2025. Whisper Leak: a side-channel attack on Large Language Models. arXiv:2511.03675.

[5] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekamp, Klaus Wehrle, and Thomas Engel. 2016. Website Fingerprinting at Internet Scale. In *NDSS*.

[6] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. 2018. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *ACM CCS*.

[7] Mahdi Soleimani, Grace Jia, In Gim, Seung-seob Lee, and Anurag Khandelwal. 2025. Wiretapping LLMs: Network Side-Channel Attacks on Interactive LLM Services. Cryptology ePrint Archive 2025/167.

[8] Mohoshin Ara Tahera, Karamveer Singh Sidhu, Shuvalaxmi Dass, and Sajal Saha. 2026. SoK: Privacy-aware LLM in Healthcare: Threat Model, Privacy Techniques, Challenges and Recommendations. arXiv:2601.10004.

[9] Arsène Fansi Tchango, Rishab Goel, Zhi Wen, Julien Martel, and Joumana Ghosn. 2022. DDXPlus: A New Dataset For Automatic Medical Diagnosis. In *NeurIPS*.

[10] Khanh-Tung Tran, Dung Dao, Minh-Duong Nguyen, Quoc-Viet Pham, Barry O'Sullivan, and Hoang D. Nguyen. 2025. Multi-Agent Collaboration Mechanisms: A Survey of LLMs. arXiv:2501.06322.

[11] Roy Weiss, Daniel Ayzenshteyn, and Yisroel Mirsky. 2024. What Was Your Prompt? A Remote Keylogging Attack on AI Assistants. In *USENIX SEC*.

[12] Faouzi El Yagoubi, Godwin Badu-Marfo, and Ranwa Al Mallah. 2026. AgentLeak: A Benchmark for Internal-Channel Privacy Leakage in Multi-Agent LLM Systems. *IEEE Access* 14 (2026).

[13] Yixiang Zhang, Xinhao Deng, Zhongyi Gu, Yihao Chen, Ke Xu, Qi Li, and Jianping Wu. 2025. Exposing LLM User Privacy via Traffic Fingerprint Analysis: A Study of Privacy Risks in LLM Agent Interactions. arXiv:2510.07176.