Tracing Agents Without Trace IDs

Reconstructing agent call graphs from encrypted network metadata

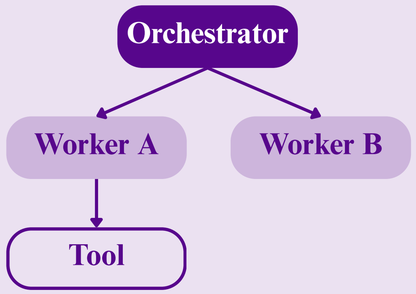
Hemanta Bhandari[1] Christina Pöpper[1]

[1]Center for Security and Privacy, New York University Abu Dhabi

1 Research Question

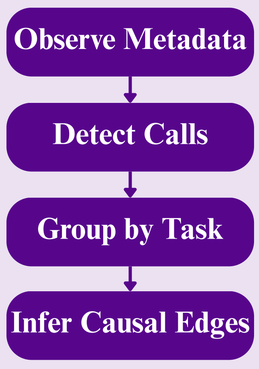
Can a passive network observer infer which agent calls caused other calls without reading messages or accessing trace IDs?

Encryption hides message content, but endpoints, timing, connection behavior, and traffic volume remain observable. We investigate whether these signals reveal the invocation-level graph of an agent execution.



2 Methodology

We built a TLS testbed containing an orchestrator, workers, tools, and a model gateway. Control traffic and ground-truth records travel over a separate, unobserved network.



We compare four views: five-second flow summaries, connection summaries, packet timing and direction, and packet timing with sizes.

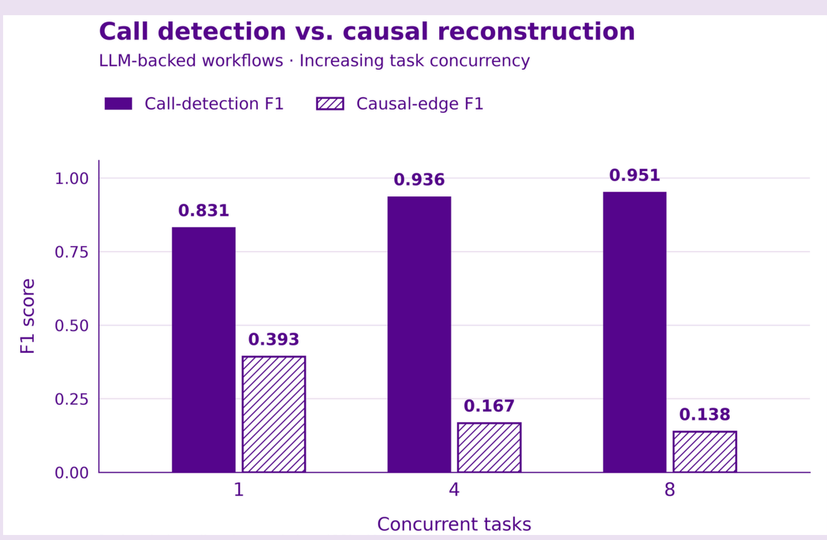
Experiments vary connection reuse, HTTP/2 multiplexing, and task concurrency, using both predefined workflows and workflows chosen by LLMs.

The LLM-backed analysis includes **539 capture blocks** containing **1187 runs**; **13 collected blocks** had no analyzable mesh invocation traffic.

3 Findings

Simple workflows are highly recoverable.
With one task and one connection per call, causal-edge F1 reaches 0.984 using packet timing, direction, and sizes. However, this condition strongly constrains possible parents, and a simple baseline performs at least as well.

Reusing connections obscures call boundaries.
In a separate controlled comparison, invocation F1 falls from **1.000** with one connection per call to **0.805** with persistent HTTP/1.1 and **0.801** with HTTP/2 multiplexing.



Calls remain detectable, but assigning them to the correct task becomes a major source of error. Strict whole-block graph match is zero at four and eight concurrent tasks in this experiment.

F1 balances precision and recall; higher is better. Invocation F1 measures call detection. Causal-edge F1 measures inferred relationships. Values are means over capture blocks, using packet timing, direction, and sizes.

4 Conclusion

Detecting calls is not enough to reconstruct an execution.

Encrypted network metadata can reveal agent call structure under restricted conditions, even without access to messages or trace IDs. Connection reuse makes call boundaries harder to detect. Concurrent tasks make it harder to assign calls to the correct execution and recover causal relationships.

Reliable reconstruction therefore requires three things: **detecting calls**, **grouping them by task**, and **identifying their causal links**.

5 Limitations & Future Work

Limitations

- Experiments used a single-host testbed with limited workflows and client implementations.
- The LLM-backed experiment used bounded workflows and a limited set of models and scaffolds.
- These results do not establish a universal limit on reconstruction. Defenses were not evaluated.

Future work

- Evaluate across multiple hosts, agent frameworks, and network conditions.
- Improve task grouping when executions overlap.
- Test whether ranked candidate-parent sets can support fault localization.
- Measure the effects of padding, batching, and cover traffic.