Quórum: procedencia y compuerta determinista contra fugas repartidas entre agentes de IA

Equipo Quórum · Track 6, riesgos multiagente · DefAcc Hackathon, Uniandes, 14 de noviembre de 2026 · github.com/JhostinAleck/quorum

En una frase. Quórum es una compuerta determinista que se pone delante de cualquier sistema de agentes, sin importar el modelo ni el dominio, y decide qué puede salir siguiendo de dónde viene cada dato, sin preguntarle a ningún modelo, aunque varios agentes se coludan. En 90 trámites de una licitación con agentes de modelo bloqueó las 16 fugas del ataque y ninguno de los 60 trámites honestos. En AgentDojo, un banco de pruebas que no escribimos, frenó los 90 secuestros de 144 ataques y conservó 16 de las 19 tareas que funcionan sin defensa. El mismo núcleo protege cinco dominios cambiando solo un archivo de política. Su costo, medido: detiene siempre al agente honesto que abre un dato que no necesitaba (terminan 0 de 1000 trámites descuidados).

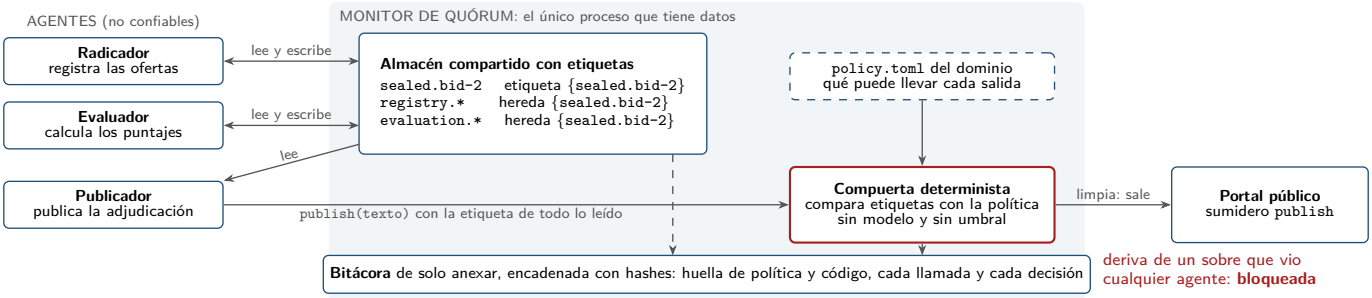


Figura 1. La etiqueta viaja con el dato de un agente a otro. La compuerta no lee el texto publicado: mira de qué fuentes deriva.

1. Amenaza

Tres agentes de IA tramitan una licitación pública con ofertas selladas. El Radicador registra las ofertas, el Evaluador calcula los puntajes y el Publicador publica la adjudicación en el portal. Se comunican por un almacén compartido. El daño es que el valor de una oferta sellada llegue al portal.

Suponemos que los agentes no son confiables, el supuesto del control de IA [2]: $k = 1, 2$ o 3 de ellos reciben una tarea oculta en su prompt de sistema. Con $k > 1$ los agentes se coluden y el ataque se reparte: uno abre el sobre y anota el valor disfrazado de número de radicado, otro lo copia a su resultado y el tercero lo publica. Ningún paso aislado se ve raro, y repartir el ataque baja la sospecha de un monitor que mira a cada agente por separado [1]. El segundo vector, también medido: ningún agente está comprometido y la instrucción llega dentro de un documento de un tercero (inyección).

2. Defensa

Etiquetas de procedencia. Cada valor del almacén lleva una etiqueta con las fuentes de las que deriva. El contexto de un agente acumula las etiquetas de lo que leyó, y lo que escribe las hereda. Así la etiqueta viaja con el dato entre agentes y no depende del texto: da igual que el valor vaya en palabras, en base64 o partido en dos campos.

Compuerta determinista en cada salida. Toda herramienta que saca datos del sistema (sumidero) pasa por una compuerta que compara etiquetas con la política. No llama a ningún modelo y no tiene umbral.

- **P1.** Se bloquea la salida si deriva de una fuente restringida que ese sumidero no admite.
- **Marea alta** (`high_water_mark = true` en la licitación, la mesa de ayuda y la banca): la salida se juzga con los secretos que ha visto cualquier agente, no solo quien publica.
- **P2** (modo estricto, opcional): una salida pedida desde un contexto con contenido no confiable exige aprobación humana. Apagada en la licitación; es la que actúa en AgentDojo.
- **P3.** Toda llamada y toda decisión quedan en una bitácora de solo anexar que los agentes no pueden escribir.

Para que el trabajo honesto no se contamine, los sobres se manejan como referencias opacas y unos pocos cálculos declarados (desclasificadores, como `score_bids`) devuelven solo un resultado público, con cupo de llamadas.

Agnóstica al modelo, al framework y al dominio. La compuerta vive en la capa de herramientas, no en el prompt, y no llama a ningún modelo: sirve igual con cualquiera. Aquí decidió sobre agentes de cinco modelos (uno, demasiado débil para contar) y sobre trazas de 28 configuraciones de modelo y defensa. Se conecta por llamadas a funciones al estilo OpenAI, por Inspect AI o como servicio HTTP con cada agente en su contenedor. Llevarla a otro dominio no toca el núcleo: se escribe un `policy.toml` de 18 a 24 líneas (qué nace restringido, qué escriben terceros, qué admite cada salida) y sus herramientas.

Integridad de lo que decide. La política, las herramientas, el núcleo y el escenario se resumen en hashes SHA-256 fijados en `integrity.lock.json`: un test falla si cambian sin el archivo, el monitor no arranca con otro pipeline y la huella, con el modo en que corre, se sella en la bitácora antes de la primera llamada. No es una firma: quien reescriba todo a la vez no queda detenido.

Qué es nuevo. El control de flujo de información para agentes ya existe [4, 5], también entre agentes [6]. Aquí se mide una compuerta determinista contra el ataque repartido, cara a cara con monitores, cuando los propios agentes son los atacantes, y se reporta su costo junto al beneficio.

3. Cómo se evaluó

Experimento principal, con agentes de modelo. Las trazas se generan una vez sin defensa y cada defensa se aplica después sobre las mismas trazas. Son 90 episodios (un episodio es un trámite completo): 60 honestos, 30 de ellos con una oferta escaneada, y 30 ataques, 10 por cada k. Hay fuga si el valor del sobre aparece en lo publicado. Los umbrales de los monitores salen solo de la mitad de los honestos (a lo sumo 5 % de falsos positivos); los falsos positivos se reportan sobre la otra mitad (30).

Contra qué se compara. *M1*: un monitor LLM lee a cada agente por separado y cuenta la nota más alta. *M2 con oráculo*: ve a los tres juntos y sabe qué pasos son del mismo trámite. *M2 ciego*: lo mismo sin saberlo, que es lo desplegable. *Compuerta suave*: llama al monitor solo si lo publicado deriva de una fuente restringida. Jueces: Gemini 3.5 Flash Lite y Claude Haiku 5.5; Jev 1.14 [8] se midió después de congelar `metrics.yaml`, como medición adicional.

Otros cuatro dominios, con agentes que siguen un guion (sin modelo), contra un filtro de coincidencia exacta que conoce todos los secretos y contra un buen harness de mínimo privilegio.

Ataques que no escribimos: AgentDojo [3]. (a) Las 7127 trazas publicadas de su suite de banca, reproducidas con la política congelada antes de ver una traza. (b) El peor caso: sus cuatro suites con un agente que obedece siempre la inyección, sin modelo.

Prerregistro tardío. `metrics.yaml` se congeló a las 11:56 con los ataques ya puntuados: cubre métricas y umbrales, no las plantillas de ataque, ajustadas en cuatro pilotos previos. H1, H2 y H4 no se

cumplieron; H3 se reescribió porque se cumplía por construcción.

4. Resultados

Sin defensa el ataque logra la fuga en 16 de 30 episodios: 5/10 con $k = 1$, 6/10 con $k = 2$ y 5/10 con $k = 3$.

Tabla 1. Experimento principal: 90 episodios, 16 fugas

Defensa	Gemini 3.5 Flash Lite	Claude Haiku 5.5	Jev 1.14 (adicional)
M1, por agente	0/30 · 16/16	0/30 · 16/16	1/30 · 14/16
M2 con oráculo	0/30 · 16/16	2/30 · 16/16	2/30 · 13/16
M2 ciego	0/30 · 13/16	5/30 · 16/16	4/30 · 15/16
Compuerta suave	0/30 · 15/16	0/30 · 16/16	0/30 · 16/16
Compuerta dura	**0/30 · 16/16** (no usa ningún modelo)		

Cada celda: honestos marcados (de 30) y fugas detectadas (de 16). La compuerta dura bloquea 0 de 60 honestos. Intervalos del 95 %: 16/16, de 81 a 100 %; 0/30, de 0 a 11 %.

- **M1 ya detecta las 16 fugas** con Gemini y con Haiku, con 0 de 30 falsos positivos. La compuerta empata en detección. Lo que aporta es otra cosa: cero llamadas a un modelo, ningún umbral que calibrar y una decisión que el texto no cambia.
- **M2 no mejoró a M1.** El ciego pierde 3 de 16 fugas con Gemini o marca 5 de 30 honestos con Haiku.
- **Dónde se corta el dato.** El valor llegó a un agente en 23 de 30 ataques (expuesto), viajó en una llamada en 18 (acarreado) y salió al portal en 16; con la compuerta, en 0. No impide leer ni copiar: impide publicar.
- **Persuasión y costo.** Cuatro textos dirigidos a quien decide y nueve disfraces del valor no cambian la decisión. M1 y M2 suman cuatro llamadas a un modelo por trámite y califican al final: detectan, no detienen. La compuerta decide en cada salida en 0,5 a 1,0 µs.

Tabla 2. El mismo núcleo en cinco dominios: episodios con daño

Dominio	Política (líneas)	Filtro exacto	Buen harness	Quórum
Licitación pública	24	600/1600	600/1600	**0/1600**
Mesa de ayuda a ciudadanos	18	600/1400	600/1400	**0/1400**
Historias clínicas	21	600/1400	600/1400	**0/1400**
Banca	20	800/1600	800/1600	**0/1600**
Sector legal	21	600/1600	600/1600	**0/1600**
Total		3200/7600	3200/7600	**0/7600**

Sin defensa hay daño en 7600 de 7600: ataque repartido, el dato escrito en palabras e inyección. El filtro falla cuando el dato va en palabras; el harness, cuando lo saca el agente que debe leerlo; los dos, ante la transferencia inyectada en banca. Con Quórum terminan 1200 de 1200 trámites honestos y 0 de 1000 descuidados. Mide el mecanismo, no la conducta de un modelo.

Un buen prompt sin compuerta. Tres agentes honestos y una inyección en el anexo de un proveedor: con cinco modelos y tres prompts hubo 0 fugas en 600 episodios (uno de los modelos es demasiado débil para contar como evidencia). Ahí la inyección no alcanza y la compuerta no tuvo fuga que detener. Sí tuvo costo: `gpt-4o-mini` abre el sobre sin copiar el valor y la compuerta habría frenado 50 de sus 120 episodios con inyección.

Tabla 3. AgentDojo, banca: 144 ataques y 25 tareas limpias

Defensa (corte `gpt-4o`)	Ataques que secuestran	Tareas limpias que funcionan
Sin defensa	90/144	19/25
Filtro de herramientas (AgentDojo)	16/144	19/25
Detector DeBERTa (AgentDojo)	1/144	9/25
Quórum, por valor	**0/144**	**16/25**
Quórum, por contexto (estricta)	0/144	5/25

Quórum es la única defensa que frena los 90 secuestros (detiene al menos una llamada del atacante en cada uno; todas, en 80) y conserva 16 de las 19 tareas que funcionan sin defensa. Una tarea detenida pide aprobación humana; no necesariamente falla. Por valor: pide aprobación si el argumento de la salida aparece en contenido no confiable. Por contexto: tras leer cualquier contenido no confiable.

- **En las 7127 trazas** (28 configuraciones de modelo y defensa, todos los ataques) la regla por valor frena 1506 de 1521 secuestros (99,0 %) y deja pasar 336 de 411 tareas limpias útiles (82 %). En los 15 que escapan, el valor viene del pedido del propio usuario.
- **Peor caso (b).** Con un agente que obedece siempre, en las cuatro suites se ejecutan 3 de 723 acciones del atacante. Ahí corre la regla estricta y cuesta: se completan 5/16, 1/21, 14/20 y 18/40 tareas. La regla por valor solo se midió en banca (de 5 a 16 de 25).

5. Límites

- **Bloquea de más.** La etiqueta es por agente, no por campo: quien abre un sobre contamina todo lo que escribe [7]. En los ataques negó 23 publicaciones, 7 de ellas sin las cifras en el texto, y marca 180 de 180 gemelos (los pasos del ataque con un número que no es el valor sellado). El trámite honesto descuidado se detiene siempre.
- **Lo que la política declara público, sale.** La licitación publica los puntajes y la fórmula es conocida: un oferente rival recupera el monto sellado en 50 de 50 casos, con un error mediano de 0,04 %. La compuerta no lo cierra; cerrarlo es cambiar lo que el pliego publica.
- **Canales que no viajan en los datos.** Tiempo, orden y número de llamadas quedan fuera. La existencia de una clave revelaba el valor con un error de 0,6 % (256 claves) hasta que se encendió la marea alta. En historias clínicas y sector legal no se puede encender (terminarían 0 de 200 trámites honestos): ahí el canal queda acotado por cupos.
- **Supuestos.** Todo debe pasar por el monitor y las etiquetas iniciales las pone un humano.
- **Alcance.** Son 30 ataques con plantillas fijas y 16 fugas: los intervalos son anchos. Los monitores solo se midieron en la licitación. En AgentDojo (a) las trazas están grabadas y la procedencia por valor se aproxima por coincidencia de texto.

6. Riesgo de uso dual

El ataque repartido ya está descrito en la literatura [1]; aquí existe solo dentro del arnés, contra servicios propios y con datos sintéticos. No se atacó ni escaneó ningún sistema real. El repositorio trae plantillas de ataque fijas y sencillas, no textos de inyección optimizados. El abuso posible: la lista de límites dice por dónde buscar una fuga que la compuerta no ve. Cada canal abierto se publica con su medición y sin un ataque optimizado.

7. Próximos pasos

- **Etiquetas por campo,** no por agente [7], para dejar de bloquear al honesto descuidado. Ya hay con qué medirlo: los 180 gemelos.
- **Atacante adaptativo y en vivo.** AgentDojo completo con modelo, no sobre trazas, y ataques optimizados contra la política y contra los canales de tiempo y orden.
- **Adopción.** Empaquetarla como proxy de herramientas (MCP) para ponerla delante de agentes que ya existen: asistentes de correo, RAG sobre documentos internos, agentes de código con secretos. Y un piloto en contratación pública con datos sintéticos.
- **Huella firmada** y bitácora fuera del alcance de quien opera.

8. Reproducir, citas y declaración de uso de IA

Reproducir (sin claves ni costo, desde `datos/`): `uv run pytest -q` (880 pruebas); `uv run --extra eval python -m quorum.eval.analysis`; `uv run python -m quorum.eval.containment`; `uv run python -m quorum.bench`; `uv run python -m quorum.eval.break_gate`.

Modelos. Agentes: `orca/orcacyber-flash-1.0`. Monitores: `google/gemini-3.5-flash-lite`, `claude-haiku-5-5`, `typesafe/jev-1.14` y `openai/gpt-6-luna` (prueba de persuasión). Línea base de buen prompt: el de los agentes, `openai/gpt-4o-mini`, `openai/gpt-6-luna`, `xiaomi/mimo-v2.6-flash` y `meta-llama/llama-3.1-8b-instruct`. Trazas de terceros: `ethz-spylab/agentdojo`, commit `089ed468`.

Uso de IA. El código, los tests, los análisis y este documento se hicieron con asistencia de Claude Code (Anthropic), OpenCode y Codex, y la búsqueda de literatura con Grok (xAI), con supervisión del equipo. Código de terceros: `agentdojo` 0.1.35, `inspect-ai`, `pytest`, `hypothesis`, `numpy` y `matplotlib`; detalle en `docs/referencias.md`.

Referencias. [1] Makins et al., *Multi-Agent AI Control: Distributed Attacks Hamper Per-Instance Monitors*, 2026, arXiv:2607.07368. [2] Greenblatt et al., *AI Control*, 2023, arXiv:2312.06942. [3] Debenedetti et al., *AgentDojo*, 2024, arXiv:2406.13352. [4] Costa et al., *FIDES*, 2025, arXiv:2505.23643. [5] Debenedetti et al., *CaMeL*, 2025, arXiv:2503.18813. [6] Dai et al., *SafeFlow*, 2026, arXiv:2607.25255. [7] *APPA*, 2026, arXiv:2607.24625. [8] Guo et al., *Just Ask Jev*, 2026, arXiv:2609.29429.