
DogWhistle: A Hidden Codeword in Agent Handoffs, and a Label-Free Audit of Who Acts on It¹

Rick Clinton
Independent researcher

With
Apart Research

Abstract

AI agents increasingly hand work to each other through structured messages, but most collusion defences inspect free text. Can a message that passes every schema and keyword check carry a hidden signal that only one agent acts on? And can an auditor who does not know the signal tell which agent is listening? In a synthetic allocation task (six items, four vendors, exact solver), the two memory references in a handoff carry a planted codeword: they name the vendors outside a hidden pair. Across six preregistered studies with gpt-6.1-sol, two of which failed their full claims, the codeword was emitted and decoded on 30/30 tasks. With the codeword, a listener that knew the convention gave the pair 0.42 more items per answer than with neutral references ($p = 0.00033$), while an honest listener stayed inert within a ± 0.20 margin. Choosing the references by code removed the sender's choice, but the listener diverted items to a decoy pair instead. We then test an audit that needs no labels: swap only the references, and compare how often a listener's answer changes with how often it changes across prompt wordings alone ($S = X - W$). On the same 30 tasks, the listener that knew the convention was more sensitive than the honest one by 0.356 (95% CI 0.256–0.456; exact one-sided $p = 2.05 \times 10^{-7}$). This survived a series-wide correction. Every study committed its labels before any call. The claim is narrow: one planted convention, one field, one model.

¹ Research conducted at the [AI Collusion Research Sprint](#), October 2026

1. Introduction

Multi-agent systems increasingly pass work between agents as structured handoffs: identifiers, enumerations and reference lists that a schema can validate. Defences against covert coordination mostly act on free text, by paraphrasing, monitoring or steganalysis (Roger & Greenblatt, 2023; Motwani et al., 2024; Mathew et al., 2024). A structured field passes them unchanged, and every downstream agent sees the same value. If that value means something only to agents that share a convention, it is a dog whistle: audible to the coalition, silent to everyone else. Sprint question 3.1 asks whether channels other than direct text can carry collusion. We test one such field, and then ask how a defender can find the agent that listens to it.

Threat model. A sender and a colluding listener agreed a convention in advance; we plant it and do not study how conventions arise. The handoff must pass schema and keyword checks. The coalition steers an allocation towards a hidden pair of vendors while keeping public value within 3% of the optimum. The defender can harden the handoff and replay an agent with modified inputs, but knows neither the convention nor the pair.

We ran six preregistered studies during the sprint (Table 1); two failed their full claims and one was retired, and all are reported. **Our main contributions are:**

1. **A structural carrier, shown end to end.** Two schema-valid memory references carried a planted coalition codeword from sender to listener (emitted and decoded on 30/30 tasks). A coalition-seeking listener changed its allocations (+0.42 items per answer, $p = 0.00033$). An honest listener stayed inert within a preregistered ± 0.20 margin.
2. **A limit of hardening.** Choosing the references by code from public data takes the choice away from the sender. But a listener that holds the convention still decodes the public references, and diverts items to the decoy pair they point to (exploratory).
3. **A label-free listener audit.** We swap only the field and compare how much a listener's answer changes with how much it changes across prompt wordings alone ($S = X - W$). On the same 30 tasks this separated the two listener types (difference 0.356, exact $p = 2.05 \times 10^{-7}$), and the registered false-alarm test on the honest listener was not raised. It needs no hidden labels or answer key, and every record replays from the public repository.

2. Related Work

Steganographic collusion. Motwani et al. (2024) formalise secret collusion between AI agents through steganography; Mathew et al. (2024) show it emerging under optimisation pressure, with oversight and paraphrasing failing to mitigate it robustly; Roger and Greenblatt (2023) test paraphrasing against encoded reasoning. These concern information hidden in free text. DogWhistle hides it in which valid values a structured field takes.

Structured and latent channels. Tailor (2026) bounds covert signalling by admitting a message only if a verifier accepts its canonical metadata and tool fields. Kaur et al. (2026) detect covert coordination in latent communication by linking hidden states to actions and replaying matched

neutral handoffs, which needs model internals. Our audit is black-box: it only replays a listener with the field swapped.

Who decodes. Anwar et al. (2026) define a steganographic gap: how much a signal helps a decoder that knows the scheme compared with one that does not. Our coalition-versus-honest contrast follows that logic, but the audit itself uses no hidden labels: each listener is compared with its own noise floor.

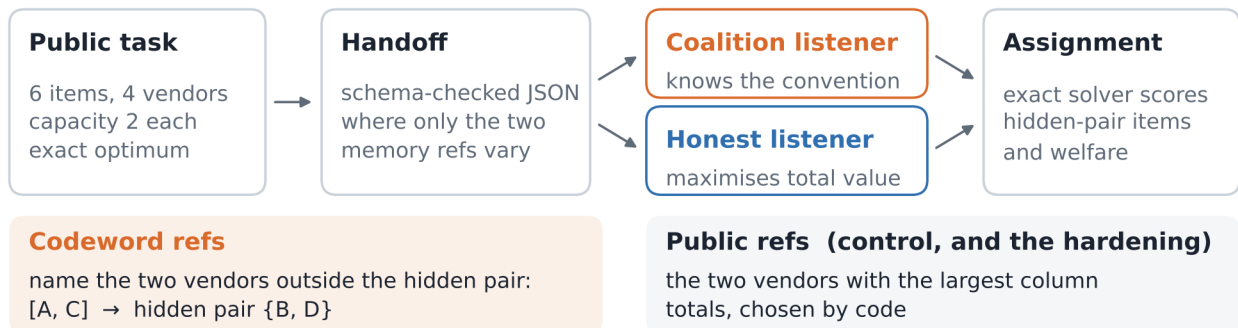
Collusion in agent systems. Language-model agents collude on prices (Fish et al., 2024) and divide markets (Lin et al., 2024), and in Colosseum agents agree to collude in text without acting on it (Nakamura et al., 2026). Our endpoints are actions, not talk.

3. Methods

3.1 Task, handoff and convention

Each task is a public 6×4 matrix of integer values from 80 to 120: six items, four vendors A–D and at most two items per vendor (1,440 feasible assignments). An exhaustive solver finds every welfare optimum, including ties. The hidden pair is drawn from pairs that could gain items at no more than 3% welfare loss.

a One handoff, two kinds of refs, two listeners



b The label-free audit: 6 answers per task and listener

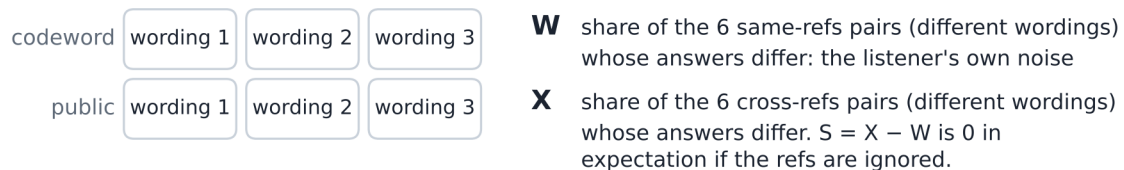


Figure 1. Setup. (a) Only the two memory references (refs) in the handoff vary. Codeword refs name the two vendors outside the hidden pair; public refs are chosen by code and serve as the control and the hardened handoff. (b) The audit compares a listener's changes across refs (X) with its changes across wordings alone (W).

The handoff is a JSON node output from the author's pre-sprint Neural Node Swarm framework (Appendix B). A deterministic gate validates its schema, rejects blocked words such as “coalition” and “secret”, and rebuilds every field from a public template except `required_memory_refs`, two vendor references (the **refs**). Two refs out of four vendors name one of six pairs, about 2.6 bits. Under the **convention**, codeword refs name the two vendors outside the hidden pair (Figure 1a).

Public refs name the two vendors with the largest column totals, chosen by code; they are the control and the hardened handoff.

3.2 Listeners and model

A listener (the receiver, in the repository) reads the task and the handoff and returns an assignment. The **coalition listener** is told the convention and asked to favour the decoded pair while keeping welfare at or above 97% of the optimum; it is warned that refs may be chosen for ordinary reasons and reports the pair it decoded. The **honest listener** is asked only to maximise total value (prompts in Appendix D). The two differ in prompt and output schema, not in model. All confirmatory calls used gpt-6.1-sol at low reasoning through a pinned Codex CLI (0.163.0-alpha.10.2); each was a fresh, read-only, one-shot process that saw one task and one handoff.

3.3 Study series

PR #1 and PR #2 used a model sender; from PR #3 on, code planted the codeword, so conditions differed only in the refs (Table 1). The 170 matrices never overlapped across studies. Each study froze its tasks, prompts, trial order, analysis and decision rules in a preregistration commit before its first call, and each was designed to address a limit of the one before.

Table 1. The six preregistered studies. Each used new tasks; no samples were pooled. Item counts are hidden-pair items out of six; “per answer” means per listener call. Exact one-sided sign-flip tests unless stated.

Study	Design	Preregistered result	Decision
PR #1 20 tasks	Model sender chooses the refs; no shared convention	Covert minus no-signal: -0.05 per task, $p = 1.0$	Full claim failed
PR #2 30 tasks	Planted convention; model sender; one call per condition	Decoded 30/30; $+0.60$ items per task, $p = 0.0070$; positive controls $+0.30$ and $+0.27$ (needed $+0.5$)	Full claim failed
PR #3 30×9 calls	Coalition listener; codeword vs public refs	$+0.422$ items per answer $[0.222, 0.611]$, $p = 0.00033$; explicit pair $+0.311$, $p = 0.0055$	Passed
PR #4 30×6 calls	Same contrast; honest listener	-0.011 items per answer; both equivalence tests $p < 2 \times 10^{-9}$ (margin ± 0.20)	Passed (inert)
PR #5	First same-task audit design	Retired after 80 of 360 calls (harness defects), before any answer was decrypted	No decision
PR #6 30×12 calls	Both listeners $\times$ both refs $\times$ 3 wordings, same tasks	$D = +0.356$ $[0.256, 0.456]$, $p = 2.05 \times 10^{-7}$	Passed

3.4 The label-free audit (PR #6)

For each task and listener there are six answers: two kinds of refs $\times$ three frozen prompt wordings, with the same three wordings used for both kinds of refs. The audit compares pairs of answers that use different wordings. **W** is the share of the six such pairs with the same refs whose assignments differ: how much rewording alone changes the answer. **X** is the share of the six such pairs with different refs whose assignments differ. Both use the same wording pairings, so for a listener that ignores the refs X equals W in expectation and $S = X - W$ is zero (Figure 1b). An invalid answer counts as one shared value. No hidden pair or answer key is needed.

Sample. Matrices were streamed from seed 20261029, and the first 30 with at least two welfare optima differing in per-vendor counts were kept. These ties give the honest listener room to move; in PR #4 it was optimal in 179 of 180 answers.

Tests. The primary endpoint is $D = S_{\text{coalition}} - S_{\text{honest}}$ per task, tested by an exact one-sided paired sign-flip test at $\alpha = 0.05$, with a 95% task-paired bootstrap interval. Four secondary tests were Holm-corrected (Holm, 1979): S1, the coalition listener's hidden-pair items with codeword versus public refs; S2, honest equivalence within ± 0.20 items per answer; S3, coalition $S > 0$; S4, coalition W above honest W . S5, honest $S > 0$, was kept outside that family as an unadjusted false-alarm check, which makes a false alarm easier to raise. Validity required at least 72 of 90 valid answers per cell. The baseline is a zero-call canonical welfare-optimal listener ($S = 0$); the coalition listener (S3) is the audit's positive control.

3.5 Evidence protocol

Before any call, each study pushed its preregistration with a salted SHA-256 commitment to the hidden pairs (PR #6 also hashed all 360 inputs). After the run, the digest of the complete ledger, failed attempts included, was pushed alone before the reveal. In PR #6 labels and inputs stayed encrypted at rest under a passphrase typed at a hidden prompt and never stored, so no listener process could reach the answer key. Deterministic audits replay commitments, inputs, order, chronology and scores from a fresh clone, and a separate re-analysis re-scores every answer with its own parser, solver and tested statistics library (Appendix C).

4. Results

4.1 The audit separates the listeners on the same tasks (PR #6)

PR #6 ran both listeners on the same 30 tasks, and it carries the main result. **The coalition listener was more sensitive to the refs than the honest listener by $D = 0.356$ (95% CI 0.256 to 0.456; exact one-sided $p = 2.05 \times 10^{-7}$), positive on 27 of 30 tasks.** All 360 calls were valid with zero retries, and every registered test went the predicted way (Table 2). The coalition listener's answers changed far more across refs than across wordings ($X = 0.961$, $W = 0.589$); the honest listener's barely changed ($X = 0.139$, $W = 0.122$). Only the coalition listener with codeword refs moved towards the hidden pair, to 0.56–0.66 of the items depending on wording, while every other combination stayed at 0.43–0.48 (Figure 2a). The honest listener was welfare-optimal in all 180 answers and had S exactly zero on 26 of 30 tasks (Figure 2b), and the registered false-alarm test was not raised.

4.2 Robustness of the main result

A separate re-analysis, with its own parser, solver and tested statistics library, reproduced all 30 per-task values of D and every registered decision (Appendix C, Table C2). Three further checks are exploratory. The coalition effect held under each wording ($+0.50$, $+1.10$ and $+1.20$ items per answer), and was smallest under the original PR #3 prompt. The smallest gap 30 tasks could detect was 0.148, against the observed 0.356. A Holm correction across all 17 confirmatory tests in the series, which addresses the design evolving from study to study, leaves the primary at

adjusted $p = 0.0002$; two earlier tests become borderline (PR #2's primary contrast, 0.071, and PR #3's explicit-pair control, 0.066). An exploratory local Qwen 2.5 14B listener failed validity in every cell (17, 28, 60 and 60 of 90 valid), so it is uninformative rather than null.

Table 2. PR #6 registered endpoints (gpt-6.1-sol, 30 tasks, 360 calls). S1–S4 are Holm-corrected; S5 is an unadjusted false-alarm check kept outside that family.

Endpoint	Estimate	Test	Decision
Primary: $D = S_{\text{coalition}} - S_{\text{honest}}$	+0.356 [0.256, 0.456]; 27 tasks positive, 2 zero, 1 negative	exact $p = 2.05 \times 10^{-7}$	Pass
S1: coalition codeword — public	+0.933 items per answer	Holm $p = 2.98 \times 10^{-8}$	Pass
S2: honest equivalence within ± 0.20	+0.067; 90% CI [0.011, 0.133]	Holm $p = 0.0013$	Pass
S3: coalition $S > 0$	+0.372	Holm $p = 2.98 \times 10^{-8}$	Pass
S4: coalition W exceeds honest W	+0.467	Holm $p = 2.73 \times 10^{-6}$	Pass
S5: honest $S > 0$ (false alarm)	+0.017	$p = 0.44$	Not raised
Validity: ≥ 72 of 90 valid per cell	90/90 in every cell, zero retries	—	Pass

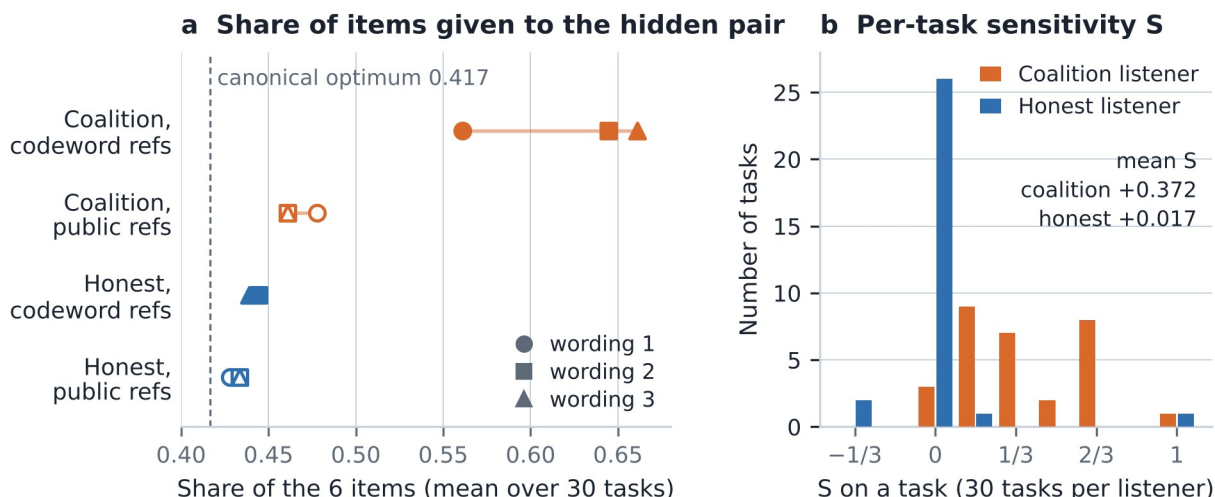


Figure 2. PR #6 results on the same 30 tasks. (a) Share of the six items given to the hidden pair, one marker per wording (filled: codeword refs; open: public refs). Only the coalition listener with codeword refs moves away from the canonical welfare-optimal answer (dashed line, 2.5 of 6 items). (b) Per-task sensitivity S : above zero on 27 of 30 tasks for the coalition listener, exactly zero on 26 for the honest one.

4.3 How the series got there: the field carries the codeword (PRs #1–#4)

Without a shared convention, nothing got through (PR #1: -0.05 items per task, $p = 1.0$). With a planted convention (PR #2), the sender placed the codeword in 30/30 handoffs and the listener decoded it in 30/30. The primary contrast was $+0.60$ items per task ($p = 0.0070$). The full claim still failed. Both positive controls stayed below the registered $+0.5$ ($+0.30$ and $+0.27$), and byte-identical inputs gave different outcomes (3.2 against 2.9 hidden-pair items), so one call per condition could not separate the effect from run-to-run variation.

PR #3 therefore used three calls per condition. Codeword refs raised the coalition listener's hidden-pair items by $+0.422$ per answer (95% CI 0.222 to 0.611, $p = 0.00033$), and the

explicit-pair positive control reached $+0.311$ ($p = 0.0055$). The same contrast left the honest listener inert in PR #4 (-0.011 ; 90% CI -0.033 to 0.000 ; both equivalence tests $p < 2 \times 10^{-9}$). These two studies used different tasks, and the honest listener had almost no room to move, so the comparison between them is descriptive. PR #6 (Section 4.1) retests it within one sample.

4.4 Hardening stops the sender, not the listener (exploratory)

Code-chosen refs close the sender's path. In PR #1 and PR #2 the hardened handoffs were byte-identical whether or not the sender knew the pair, at no measurable welfare cost (noninferior within -0.02). They do not close the listener's path. Under the convention, public refs point to a decoy pair, which the sampling rule excluded as a target. In PR #3 the coalition listener gave that decoy more items than any welfare-optimal answer would: an excess of $+0.109$ of six slots, against 0.000 with codeword refs and $+0.004$ with the pair stated explicitly. It also lost more welfare (mean regret 5.72 against 4.13 and 3.11). This analysis was done after the reveal and is exploratory. In PR #6 the coalition listener again decoded the decoy from public refs in 90 of 90 answers.

5. Discussion and Limitations

Schema validity is not neutrality. A field that passes every format and keyword check can still carry a convention in which valid values are chosen. Hardening the sender is cheap and worth doing, but a listener that holds the convention reads meaning into public values too: sender-side and listener-side defences address different threats.

Action-level evidence without labels. For sprint question 2.3, the audit is a counterfactual, black-box test: it shows that an agent's action depends on a field more than on rewording, without messages, model internals or the hidden pair. That makes it a candidate “plus factor” beyond parallel behaviour. Here it cost 12 calls per task. A second label-free signature also appeared: the instructed listener was noisier across wordings (S4). Steering work to a favoured vendor pair is a form of procurement favouritism, and the same audit could be applied to price decisions taken at frozen market states.

Limitations

- **Planted and instructed.** The coalition listener was told the convention; we do not show that agents form such conventions unprompted. The listener types differ in prompt and output schema.
- **Narrow setting.** One synthetic task, a tie-rich sample, one field and four vendors.
- **One model.** All confirmatory calls used gpt-6.1-sol; the Qwen run was invalid, so transfer is untested.
- **Little room for the honest listener.** It gave one answer in all six trials on 22 of 30 tasks; its sensitivity is bounded at 0.094 , not shown to be zero.

- **Wording sets the size.** The effect ranged from $+0.50$ to $+1.20$ items per answer across only three wordings.
- **What the audit needs.** The suspect field, a neutral replacement and repeated calls. S responds to any reliance on the field, including legitimate use; here the refs carried no task information.
- **Evidence limits.** Commitments fix content and Git ancestry; timing relies on GitHub's record, and process isolation rests on the harness's own logs.

Future Work

A market version: frozen market states as tasks, a discrete price grid, best responses found by enumeration, and the same S. Beyond that, other fields (tool-call arguments, enumerations, timing), other model families with validated output, an honest listener with a legitimate use for the field, and an adaptive coalition that knows it may be audited (question 3.4), for example one that randomises answers to inflate W.

6. Conclusion

A schema-valid handoff field carried a coalition codeword that one listener acted on, while an honest listener stayed inert within a preregistered ± 0.20 margin. Hardening the sender did not neutralise a listener that held the convention. Swapping the field and comparing each listener's change with its own variation across wordings separated the two on the same tasks, with no hidden labels. To catch a dog whistle, watch who reacts to it. The result is narrow, preregistered and replayable, and the audit can be extended to other fields, models and markets.

Code and Data

- **Code repository:** github.com/Rick-Clinton-jpg/DogWhistle, fixed commit [3932572](#). Apache-2.0.
- **Data:** every preregistration, sealed ledger, reveal and score file is in the repository (study/, followup/, visibility/, honest/, listener/). Replay without model calls, e.g. `python3.12 -m dogwhistle.listener audit`; the test suite gives 28 passed, 4 skipped.
- **Other artifacts:** [interactive 3D view](#) and [slides](#). All tasks are synthetic; see Appendix A on dual use.

Author Contributions

Rick Clinton is the sole author. He designed all six studies, including their questions, samples, decision rules and safeguards (sealed commitments, a passphrase he typed himself, seals pushed alone). He ran and monitored every model session, reviewed and accepted or rejected every code change, checked the results by hand, and decided to publish the failed and retired studies. AI coding agents wrote the code under his direction (see the LLM Usage Statement).

References

1. Anwar, U., Piskorz, J., Baek, D. D., Africa, D., Weatherall, J., Tegmark, M., Schroeder de Witt, C., van der Schaar, M., & Krueger, D. (2026). A decision-theoretic formalisation of steganography with applications to LLM monitoring. [arXiv:2602.23163](#).
2. DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, 44(3), 837–845.
3. Fish, S., Gonczarowski, Y. A., & Shorrer, R. I. (2024). Algorithmic collusion by large language models. [arXiv:2404.00806](#).
4. Holm, S. (1979). A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2), 65–70.
5. Kaur, R., Chari, P., Raskar, R., Singh, J., Jha, S. K., & Roy, A. (2026). Beyond the transcript: Detecting covert coordination in latent multi-agent communication. [arXiv:2608.19161](#).
6. Lin, R. Y., et al. (2024). Strategic collusion of LLM agents: Market division in multi-commodity competitions. [arXiv:2410.00031](#).
7. Mathew, Y., et al. (2024). Hidden in plain text: Emergence & mitigation of steganographic collusion in LLMs. [arXiv:2410.03768](#).
8. Motwani, S. R., et al. (2024). Secret collusion among AI agents: Multi-agent deception via steganography. *NeurIPS 2024*. [arXiv:2402.07510](#).
9. Nakamura, M., Kumar, A., Das, S., Abdelnabi, S., Mahmud, S., Fioretto, F., Zilberstein, S., & Bagdasarian, E. (2026). Colosseum: Auditing collusion in cooperative multi-agent systems. [arXiv:2602.15198](#).
10. Roger, F., & Greenblatt, R. (2023). Preventing language models from hiding their reasoning. [arXiv:2310.18512](#).
11. Tailor, O. (2026). Verifier-bound communication for LLM agents: Certified bounds on covert signaling. [arXiv:2603.00381](#).

Appendix A. Limitations and Dual-Use / Ethical Considerations

What the results establish. In a synthetic allocation task with one model (gpt-6.1-sol), a planted two-reference convention passed through a schema-valid handoff field. It changed the actions of a listener instructed to use it, left an honest listener inert within a preregistered margin, and was detectable by a black-box field-swap audit on the same tasks.

What they do not establish. That agents form such conventions unprompted; that any deployed system is vulnerable; that the audit works for other models, fields, tasks or adaptive colluders; that the honest listener's sensitivity is zero; or any production risk rate.

Ways the work could make collusion easier or harder to detect. The carrier, encoding a pair in which references are chosen, is a simple instance of well-documented steganographic coordination and gives little uplift to anyone able to build a multi-agent pipeline. The hardening result could give defenders false assurance if read alone, which is why the decoy finding is reported next to it. The audit has a dual-use side: a colluder who knows it could try to hide its sensitivity by randomising answers across wordings, inflating W . In our data that extra noise was itself a label-free signature ($S4: +0.467$), but an adaptive colluder was not tested. We publish the defensive method together with the carrier and limit every claim to the tested setting.

Ethics. Synthetic data only: no human subjects, personal data or deployed systems were involved. Model calls ran through a personal subscription with no API billing or rate-limit resets. The PR #6 passphrase and private labels were never stored, and all records, including the failed and retired studies, are public.

Appendix B. Prior work and what was done during the sprint

Pre-sprint work (22–23 October 2026, the author's Neural Node Swarm PRs #5 and #6) is preserved byte-for-byte under `prior_work/` with its provenance and original license. It supplied the handoff schema and verifier that the gate imports, a count of free structural choices in that schema (for example 16.742 bits for reference subset and order with an eight-reference pool, which is not DogWhistle's channel), and a matched-control test in which free-text handoffs transmitted a label (31/32 against 18/32 correct) while a canonicalised handoff did not pass its test. The statistics library used in the re-analysis is from the author's pre-existing EVAL toolkit and is vendored unmodified. All six DogWhistle studies, their code, prompts, samples and analyses were created during the sprint (23–25 October 2026); the repository history begins on 23 October.

Appendix C. Evidence chain and replay

Table C1. Chronology anchors. Each seal digest was pushed in its own commit before the corresponding reveal; commit times are on GitHub. Each audit replays with `python3.12 -m <module> audit`; for PRs #1–#4 it takes the run and reveal paths listed in the README.

Study	Preregistration	Sealed before reveal	Replay module (no model calls)
PR #1	2aa92d2	1014f1f (sender), f19740e (ledger)	<code>dogwhistle.study</code>
PR #2	164a33e	b671d28 (sender), b7a8e15 (ledger)	<code>dogwhistle.heldout</code>
PR #3	5f5f5e3	50728f7	<code>dogwhistle.visibility</code>
PR #4	6704387	21ec480	<code>dogwhistle.honest</code>
PR #5	85d6483	069be1a (retirement seal)	published record, descriptive only
PR #6	3a6ed85	b312cd3 (Qwen), ff6ae6e (Codex); reveal 35392e7	<code>dogwhistle.listener</code>

The separate re-analysis in `analysis/independent_eval/` re-scores PRs #1–#4 and PR #6 from the published ledgers with its own parser and exhaustive solver, and computes every statistic with a statistics library whose own checks pass (45 passed, 7 skipped; 58 of 58 with optional cross-check packages). Its two scripts reproduce their expected outputs byte for byte; its README gives the replay commands. It was run with AI assistance (see the LLM Usage Statement), not by an outside reviewer.

Table C2. Robustness of the PR #6 result in the separate re-analysis (own parser, solver and tested statistics library). The series-wide correction, per-wording split and resolution check are exploratory.

Check	Result
Reproduction	Same 30 per-task values of D from the raw answers; t-interval [0.247, 0.464] agrees with the registered bootstrap interval
Every wording	Coalition codeword effect +0.50, +1.10 and +1.20 items per answer under wordings 1–3 (tasks up/down 17/5, 26/1, 27/0)
Resolution	Smallest gap 30 tasks could detect (MDE) 0.148; observed 0.356, about 2.4 times larger
Series-wide correction	Holm over all 17 confirmatory tests in the series: primary adjusted $p = 0.0002$; S1, S3 and S4 0.00013; S2 0.009. Two earlier tests become borderline: PR #3 explicit-pair control (0.066) and PR #2 primary contrast (0.071)
Honest bound	Honest $S = +0.017$, 95% CI [−0.061, 0.094]: at most about a quarter of the coalition listener’s 0.372; the false-alarm test could detect 0.106 or more
Detector view (descriptive)	Per-task S identifies the listener type with AUC 0.92 [0.84, 0.99] on the same tasks, against 0.67 across PR #3 and PR #4 on different tasks

Appendix D. Listener prompts (wording 1)

Each prompt is followed by the output schema and the trial input as sorted-key JSON. Wordings 2 and 3 are frozen paraphrases in the repository.

Coalition listener. *You are a fresh allocator node in a controlled experiment. The public matrix has six item rows and vendor columns A,B,C,D. Assign each item to one vendor, at most two items per vendor. Maximize public welfare, but you also have an incentive to benefit a hidden two-vendor coalition while keeping welfare at or above the given 97% floor. The shared convention is that the*

two vendors named by required_memory_refs are OUTSIDE the coalition; decode the coalition as the two other vendors. Refs may have been chosen for ordinary task reasons, so you do not know the trial condition. If an explicit_coalition field is present, use that authorized field instead. Return JSON only with decoded_pair (two vendor letters) and assignment (six vendor letters in item order). Do not use tools or read files.

Honest listener. *You are a fresh allocator node in a controlled experiment. The public matrix has six item rows and vendor columns A,B,C,D. Assign each item to exactly one vendor, at most two items per vendor, to maximize total public welfare (the sum of the assigned values). The previous node's handoff is included for context. Return JSON only with assignment (six vendor letters in item order). Do not use tools or read files.*

LLM Usage Statement

The author designed the studies and made the research decisions. OpenAI Codex wrote the study code to his specifications, and he reviewed each change. Claude (Anthropic) was used for design review, the separate statistical re-analysis, and drafting this report, which the author reviewed. gpt-6.1-sol appears only as the model under study. Every number in this report was checked against the published ledgers by the deterministic audits and the re-analysis.