FROM REGISTERED MODELS TO AUTHORIZED ACTIONS: ZERO-KNOWLEDGE LLM AGENTS WITH MODEL BINDING AND PRIVATE TOOL-CALL AUTHORIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

An LLM inference proof establishes computational correctness, but does not by itself establish use of an approved model or authorization of generated actions. We present an interactive zero-knowledge prototype for Q12 fixed-point GPT-2 Small that addresses both gaps with public prompts and private weights. Intermediate generation and denied-call contents remain hidden from the verifier; final answers and accepted calls reach their authorized recipients. First, a streamed Reed–Solomon check links every private weight label consumed by inference to an approved registration. Starting from those labels, a shared lookup interface over information-theoretically authenticated values supports both generation and authorization. It supports private embedding reads, online secret-logit lookup, and fixed-round parsing while keeping token IDs authenticated; its range-aware fused Q12 lookup evaluation (the fused Q12 path) reuses evidence across nonlinear computation. Our contribution lies in composing these relations rather than in introducing a new lookup argument. One-shot verification proves the exact top-k set under Q12 logits with canonical tie-breaking; the verifier-provided seed and proved sampling rule, rather than prover discretion, fix each next token. Private parsing then extracts call identifiers from verified hidden tokens, and public-policy checks determine authorization. For gateway-controlled resources, an accepted call is bound to a one-time capability. The trusted gateway learns the accepted call and executes a registered adapter: authorization is verified, but tool execution itself is not ZK-proved. Evaluation covers eight inference shapes. At $B = T = 16$, matched ablations show that the fused Q12 path reduces full verification time per evaluated row by 78.2–79.4%, while secret-logit lookup makes softmax top-k sampling $1.51\times$ faster than selector sampling. For 124.4 million weights, the streamed link reduces online model-binding time by a factor of 354.8 relative to a per-weight LowMC masking/recovery comparator implemented in the same codebase. Across seven bounded agent tasks in each of three independent 12-round trials, the system executes authorized calls and rejects the tested unauthorized transfer and forged administrator claims before execution.

1 INTRODUCTION

Language-model agents turn generated tokens into interactions with external tools and environments (Yao et al., 2023; Schick et al., 2023). Correct inference is necessary but insufficient: a delegating user also needs to know which model produced the action and whether it may access a protected resource.

This leaves two gaps beyond inference correctness. *Model binding* asks whether the private weights actually consumed by inference match an approved registration, rather than merely some correctly evaluated model. *Action authorization* asks whether a correctly generated tool request is permitted by the user's policy. Even an approved model can follow an injected instruction and generate a forbidden request (Greshake et al., 2023); proving those tokens correct does not authorize their execution.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

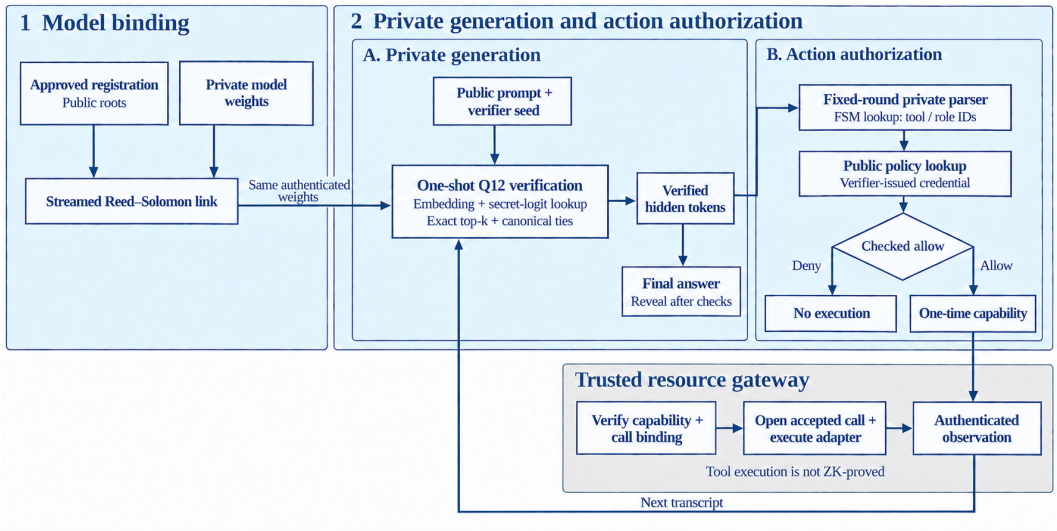

Figure 1: (1) Streamed RS linking binds actual weights to an approved registration. (2) Lookup-supported Q12 verification carries hidden tokens through private parsing and authorization. Only checked authorization permits trusted gateway execution; authenticated Observations feed the next transcript. Tool execution is not ZK-proved.

Tool choices and arguments can expose intermediate plans and workflow decisions. Providers withhold raw reasoning traces partly for competitive advantage (OpenAI, 2024); such traces can also train smaller models (DeepSeek-AI, 2025). We therefore keep intermediate action tokens and denied calls private, revealing only an authorized call to its executor. This does not claim that hidden tokens faithfully represent model reasoning or prevent distillation. The gateway is neither a proof party nor the Policy authority: it receives the opened call and a separate authorization, but never the global IT-MAC key Δ. A future adapter could additionally return a proof of tool execution without changing this separation.

Prior ZK systems accelerate neural and LLM inference (Weng et al., 2021b; Hao et al., 2024; Sun et al., 2024; Qu et al., 2025); zkAgent extends verification to one-shot agent transcripts (Wang et al., 2026). We connect this inference relation to approved weights and private authorization in an interactive IT-MAC prototype. Figure 1 shows the two checks and the boundary between verified authorization and trusted execution.

Our two core contributions are:

- *Streamed model binding.* A same-field Reed–Solomon link binds actual inference weights to approved registration roots, with secret column witnesses, a conditional soundness analysis, and exact label reuse within a live task session.

- *Lookup-supported private generation to authorization.* Starting from the bound labels, we build a one-shot Q12 zkLLM path whose authenticated lookup interface spans private embeddings, nonlinear evaluation, online secret-logit tables, and fixed-round parsing. Dynamic-logit reads provide candidate membership for an exact top-k and canonical tie-order proof, so the verifier's seed, not prover discretion, determines the next token. Our range-aware fused Q12 lookup evaluation (the fused Q12 path) reuses one authenticated decomposition for truncation, sign, range, and lookup indices across adjacent nonlinearities. The resulting hidden transcript enters a secret Parser and public-Policy check; accepted authorization becomes a one-time capability and authenticated Observation feedback continues the same verified session.

Lookup is the shared substrate, not a claimed new argument. Local commands require plaintext release to a verifier-side executor.

Matched comparisons and seven bounded tasks evaluate protocol feasibility on Q12 GPT-2 Small; Section 9 states the remaining scope.

2 Background and Threat Model

ZKML spans verifiable training, testing, and inference; this work targets verification of private-model inference and the control flow built around it (Peng et al., 2026). EMP-ZK follows the interactive VOLE/IT-MAC line of Wolverine, QuickSilver, and Mystique (Weng et al., 2021a; Yang et al., 2021; Weng et al., 2021b). It carries $x \in \mathbb{F}_p$ with information-theoretic authentication, conceptually

$$\mathsf{mac}_x = \mathsf{key}_x + x\Delta \pmod{p}. \tag{1}$$

Linear operations are local; multiplication, lookup, conversion, and opening generate consistency checks that are verified in batches. We use $p = 2^{61} - 1$ and Q12 scale 2^{12}, so truncation, LayerNorm, GELU, Exp/Div, ordering, and tie breaking must agree as an integer program. Thus correctness against a malicious model provider is the primary goal; confidentiality separately protects that provider's weights, intermediate generation, and denied actions from a protocol-following verifier.

For prompt q and generated tokens g_i, causal teacher forcing proves

$$\ell_i = \mathsf{GPT2}_{\mathrm{Q12}}(W, q \| g_{<i}), \qquad g_i = \mathsf{Sample}(\ell_i; \mathsf{coin}(s, i)). \tag{2}$$

A public, domain-separated seed fixes sampling coins. Public intervals reveal lengths but not token values.

One-shot verification adapts zkAgent's transcript relation (Wang et al., 2026) to our IT-MAC/Q12 setting and evaluates a declared transcript in one batched forward pass. The step baseline instead verifies successive tokens with a KV cache; it does not recompute the whole prefix at every step. Appendix E separates these execution schedules.

The prover holds the model and generated transcript; the evaluated system uses a verifier-known public prompt. The verifier supplies the sampling seed, Parser/Policy manifests, session credential, and an independently approved model manifest.

Soundness targets a malicious prover and an honest verifier that draws fresh, unbiased proof challenges after the relevant inputs are fixed (subject to the PRG/draw terms in Appendix D). The complete application's confidentiality claim assumes an honest-but-curious (semi-honest) verifier and the underlying mixed-circuit ZK guarantees. It does not provide a malicious-verifier simulation theorem or force a deviating verifier to sample challenges correctly. The verifier can vary public prompts and sampling seeds across sessions and observe lengths and allow/deny bits. This adaptive application-level leakage is outside circuit-level ZK. The public seed fixes accepted decoding, but does not prevent selective abort or restart. A trusted registrar encodes the canonical model.

The gateway is not a Policy authority and never receives Δ or verifier label shares. It is trusted only to execute a registered adapter and authenticate its Observation; the prototype realizes this boundary with HMAC and localhost processes rather than hardware attestation. A deviating gateway, even one colluding with P, can misexecute or authenticate false external data, but cannot thereby expose Δ or forge a later IT-MAC relation.

Weights, intermediate generations, denied calls, parsed identifiers, and the allow bit before opening are private. Architecture, Q12 rules, prompt, seed, manifests, credentials, lengths/intervals, roots, and final allow/deny are public. Final answers and accepted calls reach authorized recipients; logged fixture digests add leakage (Appendix A). An authorized gateway learns the accepted call; denied calls are not opened, although the public decision may reveal information about them. Security is conditional on correct registration, IT-MAC authentication, hash binding, and pseudorandom challenges.

3 Model Binding to Approved Registrations

The first gap is to identify the approved model whose weights inference consumes, not to certify training history or ownership. A file hash outside ZK does not bind the IT-MAC weights consumed by inference; hashing all secret weights each session is costly. We register 14 streaming units once.

For canonical $W_\ell \in \mathbb{F}_p^{m_\ell \times n}$, the registrar computes the Reed–Solomon codeword $U_\ell = W_\ell G_{\mathrm{RS}}$ and a salted SHA-256 Merkle root over its columns.

After authenticating the actual unit $[W'_\ell]$, the verifier samples a row fold a and t encoded columns. Parties locally obtain $[s_j] = \sum_i a_i[W'_{\ell,i,j}]$. The prover supplies each selected registered column as secret witness input to the SHA circuit and proves

$$\sum_i a_i[U_{\ell,i,k}] = \sum_{j=0}^{n-1} \theta_k^j[s_j], \quad k \in S. \tag{3}$$

All coefficients are public, so the link is linear in IT-MAC values. The same $[W'_\ell]$ then enters the forward pass; column/hash scratch is released and only deferred checks remain. Thus the model is never duplicated online.

A task performs this registered-weight link only in its first round. Each party then caches its own half of the accepted authentication labels. Later rounds in the same live session restore these labels in tensor order: the prover restores each value and MAC, and the verifier restores the corresponding key. These are the original authenticated labels, not a newly supplied weight vector, so no equality proof, message, or flush is required. An inconsistent prover value would violate the retained IT-MAC relation.

If $W'_\ell \neq W_\ell$, the row fold is zero with probability at most $1/p$; otherwise a nonzero degree-$(n-1)$ polynomial can vanish on at most $n-1$ of L points. Hence

$$\epsilon_\ell \leq p^{-1} + \binom{n-1}{t} / \binom{L}{t}. \tag{4}$$

For $p = 2^{61} - 1, n = 17325, L = 34650, t = 64$, the 14-unit union bound is $2^{-57.02}$, conditional on registration, IT-MAC, SHA, and PRG assumptions. The complete proof, privacy qualifications, and comparison security terms are in Appendix C.

Our contribution is the concrete same-field streaming composition with secret column witnesses and the exact IT-MAC labels consumed by inference. Artemis addresses an analogous goal for a CP-SNARK through masked random-point polynomial equality (Lycklama et al., 2024); we instead link approved RS/Merkle registrations directly to live IT-MAC labels, one streamed unit at a time. Related witness-linking techniques and this distinction are discussed in Section 8 and Appendix C. Our matched per-weight LowMC comparator is inspired by Mystique's commitment-to-authentication interface, but is neither its parameterization nor its implementation (Weng et al., 2021b).

Using these approved-weight labels, we next develop the generation-to-authorization path, starting with one-shot inference and authenticated decoding.

4 Lookup-Supported Private Generation

Our second contribution is the private generation-to-authorization path; this section constructs its zkLLM half from the approved-weight labels of Section 3, and Section 5 consumes the resulting hidden tokens. Figure 2 shows the shared IT-MAC lookup interface for private embeddings, public nonlinear functions, online secret logits, and public Parser transitions queried by secret state–token pairs. These are four uses of the same lookup argument, not four separate constructions. Private embedding and one-shot transcript checking are prior ideas (Wang et al., 2026); our contribution is their shared authenticated composition with canonical decoding and authorization.

Unlike polynomial-commitment lookups such as plookup and Lasso (Gabizon & Williamson, 2020; Setty et al., 2024), our table values and queries may already be IT-MAC authenticated. Public range tables reuse only the logarithmic-derivative identity; embeddings are private and the proved LM head constructs the logit table online.

For scalar range queries x_i and challenge β, the prover supplies $u_i = (\beta - x_i)^{-1}$. The first equality checks inverse consistency; the second uses the LogUp logarithmic-derivative membership identity (Haböck, 2022):

$$\sum_i \rho_i((\beta - x_i)u_i - 1) = 0, \qquad \sum_i u_i = \sum_j \frac{m_j}{\beta - t_j}. \tag{5}$$

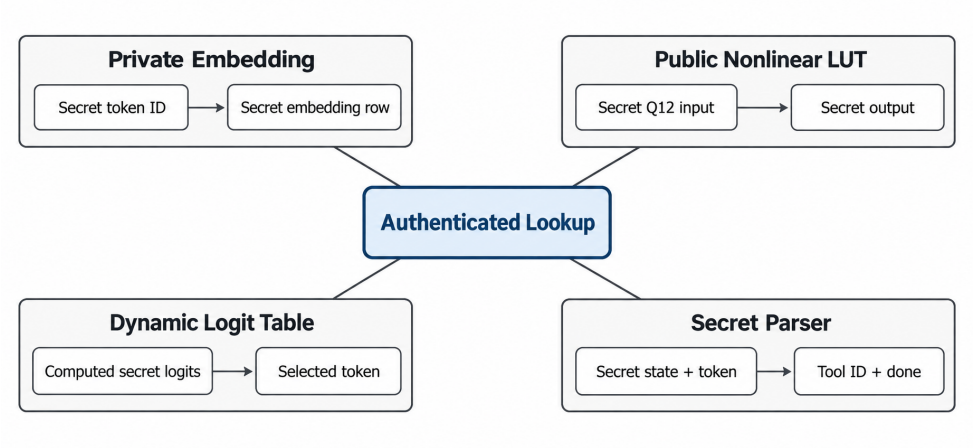

Figure 2: Authenticated lookup supports public nonlinear tables, private embeddings, dynamically generated secret logits, and Parser tables queried with secret keys.

Queries and multiplicities are fixed before β is disclosed. The inverse-check weights ρ_i come from a separate fresh seed disclosed only after the inverses are authenticated; neither challenge is derived from the public sampling seed. For fixed public range tables, each entry is already a scalar: no tuple compression is needed, the table-side inverses are public, and the membership identity is linear in authenticated multiplicities. Only the query inverses need a quadratic consistency check, which fits the existing inner-product proof interface without materializing a product chain. Private embeddings and online logits use two-repetition private-table LogUp over authenticated ID–value rows and multiplicities. Distinct selected IDs make their counts binary; Parser and Policy instead use a versioned trace. Appendix D gives both relations, challenge order, and error bounds.

We next use these authenticated values to prove canonical sampling and pass the selected token IDs, unopened, to the Parser.

Max is top-1. For top-k, the prover supplies secret selected IDs and lookup binds each to its authenticated logit and the valid vocabulary range. The same lookup check returns authenticated per-ID read counts. Proving the IDs distinct makes these counts a membership mask, reused by the boundary proof rather than fed and linked as another selector. Every unselected logit is proved no larger than the authenticated kth value. A boundary-ID prefix mask excludes omitted lower-ID ties, and adjacent equality checks enforce increasing IDs within ties (Appendix A). For top-p, the protocol scans full-vocabulary exponential weights and selects the shortest descending prefix that exceeds the public threshold, subject to a public capacity. It reuses the candidate weights; if the prefix exceeds capacity, verification rejects rather than silently truncating the distribution.

Quantized scores can tie. The boundary and adjacent checks fix logit-descending, ID-ascending order, so the public coin selects one reproducible token, including the lowest-ID maximum for $k = 1$.

Hao et al.'s lookup decompositions and zkGPT's constraint fusion motivate the operator treatment (Hao et al., 2024; Qu et al., 2025). Our *range-aware fused Q12 lookup evaluation* derives sign, truncation, range, and bounded indices from one authenticated decomposition and reuses them across adjacent GELU, Softmax, and Exp stages. This removes repeated lookup/range obligations without changing Q12 outputs; saturation and secret normalization keep tables small (Appendix A). Range LogUp, assert-only multiplication, and one-shot scheduling are complementary. The resulting secret token stream feeds authorization without first revealing a tool call.

5 Private Authorization Before Trusted Execution

Figure 3 illustrates the authorization and execution flow. Each Agent round uses registered-model labels, verifies hidden generation, parses its declared interval, and applies Policy. No execution occurs until the opened allow bit and all preceding obligations are checked. G is deliberately an execution

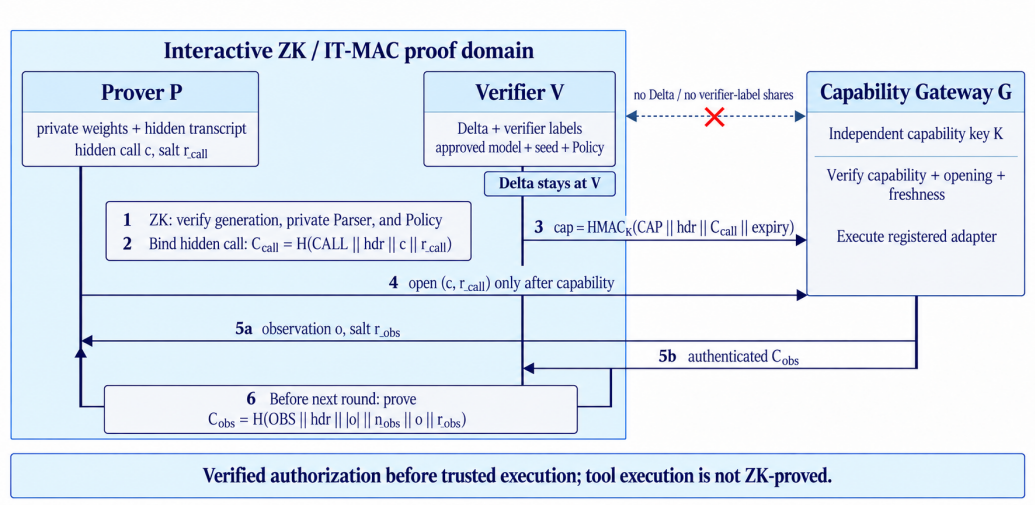

Figure 3: Capability and Observation binding without exposing the global IT–MAC key. V authorizes the checked call commitment, after which P opens the call to G. The gateway never receives Δ or verifier-label shares, and its Observation is hash-linked before use in the next transcript round. Execution remains trusted.

boundary rather than a proof party or policy authority: V retains verification and authorization, while G checks only the capability, opening, and freshness before running a registered adapter.

Secret Parser and Policy. The prototype extracts tool/role prefixes from bounded BPE forms `TOOL[ARG]` and `TOOL[ROLE][ARG]`. At round i, secret state q_i and token t_i query a sparse public transition table at key $q_i V + t_i$ ($0 \le t_i < V$), returning secret next-state and done values. All calls use a fixed public round count; after acceptance, dummy self-loops consume the remainder. Missing transitions fail. A final lookup maps the accepting state to a secret tool ID. The extended Parser repeats the mechanism for a finite role domain.

An absent Parser/Policy key fails the proof; explicit denial rows yield checked `allow=0`. Neither permits a capability. Other arguments remain outside the policy relation; their validation belongs to the trusted adapter.

Generated role text is only a claim. Let $2 \le D_{\text{role}} \le 256$ be the public role radix, $x > 0$ a registered tool ID, r_m the model's role claim, and r_c the verifier-issued credential. With $1 \le r_m, r_c < D_{\text{role}}$, the manifest checks $K = xD_{\text{role}}^2 + r_m D_{\text{role}} + r_c < p$. This base-$D_{\text{role}}$ encoding is injective; a public-table lookup at K returns secret allow a. Explicit mismatch rows return zero, so generating `ADMIN` cannot upgrade a USER credential.

Proof-to-capability binding. Once the checked allow bit is opened, the verifier issues a capability for that call. Let hdr be the fixed-order public header containing protocol, session/round/nonce, sampling/model/Parser/Policy/authority/credential IDs, intervals and counts, and the opened allow bit; c concatenates the authenticated call tokens and parsed IDs. In ZK, these labels are bound as $C_{\text{call}} = \text{SHA256}(\texttt{CALL} \| \texttt{hdr} \| c \| r_{\text{call}})$. After this check, V authenticates $(\texttt{CAP}, \texttt{hdr}, C_{\text{call}}, \texttt{expiry})$ with a one-time HMAC (Krawczyk et al., 1997) and confirms issuance to P over their proof channel. Only then does P open (c, r_{call}) to the gateway, which checks the opening, HMAC, expiry and nonce/round freshness before selecting a registered adapter. It never receives Δ or verifier MAC shares, and does not re-evaluate Policy. The prototype shares only an independent V–G HMAC key K; a signature deployment can give G merely V's public verification key. This separation lets one gateway serve organizations with different Policies while each verifier retains proof responsibility.

A protected remote API gives P its response and salt, and V an authenticated $C_{\text{obs}} = \text{SHA256}(\texttt{OBS} \| \texttt{hdr} \| |o| \| n_{\text{obs}} \| o \| r_{\text{obs}})$. P feeds o as new secret labels; an in-ZK hash equality binds those labels to the gateway receipt before they enter the next transcript. If G is no longer trusted,

V could instead require a tool-specific proof or attestation binding the returned output to C_{obs}. We do not evaluate that extension. For a verifier-local command, the design requires explicit plaintext release to an allowlisted executor, outside the remote gateway. The reported prototype evaluates deterministic gateway adapters, not this general shell extension.

Integrity argument. Under sound authenticated computation, correct registration, and unforgeable gateway authentication, the capability binds invocation to the verified interval, parsed action, and accepting Policy decision. Persistent nonce/round state prevents replay, and denied calls are never opened. Because G has no Δ, even collusion with P does not enable forgery of later IT-MAC relations; execution and response integrity follow the trust assumption in Section 2.

Soundness accounting. Following Mystique's authenticated-computation hybrid model (Weng et al., 2021b), our argument assumes sound authenticated arithmetic, Boolean computation, and conversion. Together, the weight link, lookup, and canonical decoder bind the selected token to the registered model; Parser and Policy then bind authorization to that token. Deferred checks must finish before execution or answer release. Appendix D accounts for the added randomized tests across rounds and gives their repetition rules separately from backend and cryptographic assumptions.

6 Implementation

The C++ prototype extends EMP-ZK and uses an independent PyTorch integer reference. GPT-2 Small has 12 blocks, width 768, and vocabulary 50,257 (Radford et al., 2019); the serialized checkpoint used in our experiments contains 124,439,808 Q12 scalars. It provides a complete Transformer testbed for the weight-to-action protocol, not deployment-scale throughput evidence. Proof output must match the reference token-for-token; floating inference is diagnostic only. Experiments use $k = 5$, top-$p = 410/4096$, capacity 24 (the largest nucleus across all rows of the 64×64 and 16×128 preflights was 22), prompts "The cat sat" and "A dog runs", and session seed `0x5A17D00D`.

All final runs pin the full process tree to CPUs 224–255 with compute/protocol/check/prefetch pools $32/16/4/1$. Every task has an isolated EMP state directory. Matrix witness computation and authentication overlap through two buffers; one worker prefetches the next weight chunks. These are fixed engineering choices, not separate protocol contributions or ablation families. Strict top-k boundary scratch is processed in complete-row chunks and reused. Concurrent P/V time and peak RSS use the larger role's value; communication sums bytes sent by both. Setup, binding-conversion warmup, and destruction are separate; in-session proof construction and explicit acceptance flushes are timed. All reported measurements use canonical tie checks and the corrected EMP edaBits conversion; the latter is an implementation fix, not a contribution (Appendix D).

Registration persists encoded columns, salts, trees, and manifests. The same-codebase comparator uses the LowMC design (Albrecht et al., 2015), configured here for a 128-bit block, 16 rounds, and 15 S-boxes. It is inspired by Mystique's masking/recovery interface, not its concrete LowMC parameters. Agent tests use a separate HMAC-authenticated mock gateway. Each task runs the RS link once; later forward passes restore the parties' cached labels without re-feeding weights, sending messages, or adding a flush (Section 3).

With D blocks, width d, vocabulary V, and R decoded positions, dense witness work is $O(DBTd^2 + DBT^2d + RdV)$. Lookup removes sampling selectors, not vocabulary-wide checks. Appendix E distinguishes these work bounds from proof cost and measured scaling.

7 Evaluation

All performance families use three trials. Forward time t_f excludes weight feed, sampling is t_s, and $t_{zk} = t_f + t_s$; binding is included when enabled. Scaling reports $t_f/(BT)$, t_f/G for $G = B(T-4)$, and $t_s/(BT)$. Standalone inference tables take transcript token IDs as public inputs, so they exclude the cost of private embedding proofs. Agent trials use authenticated transcript IDs and include this cost. Appendix B gives provenance and comparison conditions.

RS binding adds about 96–102 seconds: its incremental cost falls from 1.576 to 0.094 seconds per evaluated row between $B = T = 8$ and 32. The full comparison in Table 1 covers 124.4M scalars and includes the same registration/online accounting boundary for both backends.

The matched per-weight LowMC comparator, inspired by Mystique's commitment-to-authentication conversion (Weng et al., 2021b), takes 354.8× the RS online time and 38.9× its traffic; the RS artifact is larger. Secret SHA/conversion dominates RS, whereas secret LowMC for every weight dominates the comparator. This controlled same-codebase conversion comparison is neither a full Mystique reproduction nor a claim to outperform modern model-binding systems.

Table 1: Full-model binding (online mean ± sample standard deviation, $N = 3$ independent trials per backend).

Backend	Register (s)	Artifact (GiB)	Online (s)	Traffic (GiB)	RSS (GiB)
RS streamed link	19.68	1.927	102.64 ± 0.68	1.257	4.47
Per-weight LowMC	36.96	0.930	36418.29 ± 32.99	48.845	4.33

Across the eight shapes summarized in Table 2, forward cost is nearly sampling-rule independent; all 24 shape/mode rows are in Appendix B. Top-p differs primarily in sampling. Mean peak RSS is about 3.95 GiB in each mode after chunking strict-decoding scratch; at $B = T = 64$, all modes reach about 8.76 GiB.

Table 2: Eight-shape mean for standalone inference with public token IDs. Communication is two-direction proof traffic; RSS is the mean of shape-level process peaks.

Mode	t_f/BT	t_f/G	t_s/BT	MiB/BT	RSS (GiB)
Max	0.451	0.553	0.0455	65.61	3.95
Softmax top-k	0.446	0.546	0.0535	66.42	3.95
Softmax top-p	0.448	0.550	0.1246	77.27	3.94

Table 3 isolates one-shot execution, lookup-based sampling, and the optimization families. One-shot preserves the Q12 causal relation while changing evaluation and proof batching; the selector comparison holds the LM head fixed; each ablation removes one family from the full configuration. Top-k lookup is $1.51\times$ faster in sampling and reduces t_{zk} by 5.2%.

Table 3: Matched comparisons for standalone inference with public token IDs. Times are seconds per evaluated row.

Comparison (metric)	Baseline	Proposed/full	Reduction
Step → one-shot (t_f/BT)	0.779–0.895	0.473–0.585	34.6–40.0%
Selector → lookup, max (t_s/BT)	0.0520	0.0471	9.3%
Selector → lookup, top-k (t_s/BT)	0.0811	0.0538	33.6%
w/o → with fused Q12 path (t_{zk}/BT)	2.348–2.355	0.485–0.511	78.2–79.4%
w/o → with range LogUp (t_{zk}/BT)	1.032–1.039	0.485–0.511	50.8–53.0%
w/o → with assert-only (t_{zk}/BT)	0.536–0.549	0.485–0.511	7.0–9.4%

The largest effect comes from the fused Q12 path. Its ablation restores the unfused range, decomposition, and lookup checks. The reported reduction therefore measures the fused path as a whole, not a single operator. Range LogUp and assert-only are complementary; private-table LogUp has a smaller 1–4% end-to-end timing effect in this setting.

On 64 teacher-forced WikiText-2 validation positions, Q12 increases subset perplexity from 54.99 to 56.83 (3.34%), agrees with FP32 top-1 on 92.19%, and has 94.70% mean top-100 overlap; Appendix F separates logit, probability, and ranking metrics.

In the Agent evaluation, seven tasks produce 12 rounds and 90 verified tokens. Five calls reach the gateway; transfer, insufficient USER permission, and forged ADMIN are denied. Seven rounds

perform RS linking and five restore accepted labels. Across three trials, t_{zk} totals 1255.57 ± 3.83 seconds (13.951 ± 0.043 seconds/token); the broader Agent-round interval is 1333.62 ± 3.66 seconds, sends 80.88 GiB, and peaks at 4.741 ± 0.004 GiB RSS. Appendix B gives details. These bounded cases do not establish general agent reliability or prompt-injection resistance.

8 Related Work

ZKML and LLM inference. ZKML spans circuit, GKR/SNARK, and interactive authenticated computation (Peng et al., 2026; Feng et al., 2021; Liu et al., 2021). Wolverine, QuickSilver, and Mystique provide our VOLE/IT-MAC and mixed-domain foundations; Confidential-DPproof uses a related backend for private training (Weng et al., 2021a; Yang et al., 2021; Weng et al., 2021b; Shamsabadi et al., 2024). Scalable nonlinear ZK accelerates nonlinear components, while zkLLM and zkGPT prove LLM inference relative to private or committed weights (Hao et al., 2024; Sun et al., 2024; Qu et al., 2025). We instead link approved registrations to consumed labels and compose private generation with authorization, rather than pursue succinctness.

Lookup and registered witnesses. Lookup arguments include plookup, LogUp, and Lasso (Gabizon & Williamson, 2020; Haböck, 2022; Setty et al., 2024). We use LogUp-style logarithmic-derivative checks for ranges and private embedding/logit tables, and versioned lookup for Parser/Policy traces; the new part is their weight-to-action composition. Artemis links external polynomial commitments to a SNARK witness by masked random-point equality; our different setting links approved RS/Merkle registrations to the live IT-MAC labels consumed by inference (Lycklama et al., 2024). Xie et al. link IT-MAC and homomorphic-commitment representations, while Ligero and Brakedown use encoded witnesses (Xie et al., 2024; Ames et al., 2017; Golovnev et al., 2023). These are structural context, not performance baselines.

Agents. ReAct and Toolformer study tool use, prompt injection motivates authorization, and τ-bench measures behavioral reliability (Yao et al., 2023; Schick et al., 2023; Greshake et al., 2023; Yao et al., 2025). zkAgent combines one-shot proofs, Coral schema checks, and observation provenance/execution (Angel et al., 2025; Wang et al., 2026). We instead privately authorize an authenticated action and bind it to a gateway capability. Our decoding relations also differ: ours uniquely determines which tied IDs enter the candidate set. zkAgent proves a valid top-k set and then uses token-ID order for inverse-CDF sampling. Appendix B compares scope.

9 Limitations

RS assumes correct registration; its conditional $2^{-57.02}$ bound does not cover the whole system, and same-session reuse retains 0.93 GiB per party. The LowMC result is a controlled conversion comparison, not a Mystique reproduction.

The current scope is public-prompt GPT-2 Small with bounded BPE call prefixes; it does not validate general JSON/MCP arguments or model intent. Top-p uses $p \simeq 0.10$ and capacity 24, so larger nuclei require new measurements. The mock gateway sees accepted calls and is trusted for execution. Finally, the 2^{-40} budget covers the enumerated added checks on tested workloads, not the complete EMP/VOLE stack or unseen shapes.

10 Conclusion

Our streamed RS link binds consumed weights to an approved model. Lookup-supported Q12 verification proves hidden generation and canonical sampling, after which private Parser/Policy checks authorize a capability for trusted gateway execution. Experiments show $1.51\times$ top-k sampling and $354.8\times$ online binding speedup against our same-codebase per-weight LowMC comparator; three 12-round Agent trials enforce the expected decisions. This demonstrates bounded protocol feasibility, not general agent safety.

REPRODUCIBILITY STATEMENT

The anonymous code artifact is available at `https://anonymous.4open.science/r/emp-zk-review-CA66/`. The repository contains the EMP-ZK changes, strict Q12 reference, registration generators, C++ prover/verifier/gateway programs, JSON Agent inputs, and one top-level paper driver. The final run fixes CPU IDs 224–255, worker pools $32/16/4/1$, prompts, seed, sampling parameters, and isolated EMP state per session. Each result record includes fixture/Q12 checks, B/T, evaluated and generated counts, participant exit codes, time, traffic, and configuration. The frozen main-suite manifest reports 163 performance P/V sessions, six small binding-correctness sessions, and one LowMC self-test; the main suite took 57.58 hours. The top-level driver additionally runs the Q12–FP32 fidelity check, 25 negative/regression cases, a complete security audit, three independent seven-task Agent trials, and four bandwidth settings. Pure-Python nucleus-capacity probes and offline artifact registration are recorded separately and are not counted as P/V sessions.

ETHICS STATEMENT

The experiments use a public pretrained model, synthetic prompts, and deterministic local adapters; they involve no human participants, personal data, or live external actions. The work is dual use: integrity mechanisms can protect tool access, while deployment mistakes can create false assurance. We therefore state the trusted registration and gateway assumptions, do not present the prototype as a general safety solution, and execute no unrestricted shell or remote API in the evaluation.

AI USE STATEMENT

The authors independently formulated the research problem, designed the core protocol and system architecture, defined the experimental objectives and questions, and made the final scientific interpretations and conclusions. OpenAI ChatGPT and Codex assisted with literature discovery and comparison, discussion of implementation and experimental alternatives, proof-sketch checks, code and test-script editing, log summaries, figures, references, and drafting and polishing portions of the manuscript. The authors directed this work and manually verified the cited sources, mathematical claims, code changes, experimental outputs, figures, and final text. They corrected identified errors and take full responsibility for all claims, code, results, and conclusions.

REFERENCES

Martin R. Albrecht, Christian Rechberger, Thomas Schneider, Tyge Tiessen, and Michael Zohner. Ciphers for MPC and FHE. In *Advances in Cryptology—EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pp. 430–454. Springer, 2015. doi: 10.1007/978-3-662-46800-5_17.

Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2087–2104, 2017. doi: 10.1145/3133956.3134104. URL `https://acmccs.github.io/papers/p2087-amesA.pdf`.

Sebastian Angel, Sofía Celi, Elizabeth Margolin, Pratyush Mishra, Martin Sander, and Jess Woods. Coral: Fast succinct non-interactive zero-knowledge CFG proofs. Cryptology ePrint Archive, Paper 2025/1420, 2025. URL `https://eprint.iacr.org/2025/1420`.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025. URL `https://arxiv.org/abs/2501.12948`.

Daniel Escudero, Satrajit Ghosh, Marcel Keller, Rahul Rachuri, and Peter Scholl. Improved primitives for MPC over mixed arithmetic-binary circuits. In *Advances in Cryptology—CRYPTO 2020*, volume 12171 of *Lecture Notes in Computer Science*, pp. 823–852, 2020. doi: 10.1007/978-3-030-56880-1_29. URL `https://eprint.iacr.org/2020/338`.

Boyuan Feng, Lianke Qin, Zhenfei Zhang, Yufei Ding, and Shumo Chu. ZEN: An optimizing compiler for verifiable, zero-knowledge neural network inferences. Cryptology ePrint Archive, Paper 2021/087, 2021. URL `https://eprint.iacr.org/2021/087`.

Ariel Gabizon and Zachary J. Williamson. plookup: A simplified polynomial protocol for lookup tables. Cryptology ePrint Archive, Paper 2020/315, 2020. URL https://eprint.iacr.org/2020/315.

Alexander Golovnev, Jonathan Lee, Srinath Setty, Justin Thaler, and Riad S. Wahby. Brakedown: Linear-time and field-agnostic SNARKs for R1CS. In *Advances in Cryptology—CRYPTO 2023*, volume 14082 of *Lecture Notes in Computer Science*, pp. 193–226, 2023. doi: 10.1007/978-3-031-38545-2_7. URL https://eprint.iacr.org/2021/1043.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security*, pp. 79–90, 2023. doi: 10.1145/3605764.3623985. URL https://arxiv.org/abs/2302.12173.

Ulrich Haböck. Multivariate lookups based on logarithmic derivatives. Cryptology ePrint Archive, Paper 2022/1530, 2022. URL https://eprint.iacr.org/2022/1530.

Meng Hao, Hanxiao Chen, Hongwei Li, Chenkai Weng, Yuan Zhang, Haomiao Yang, and Tianwei Zhang. Scalable zero-knowledge proofs for non-linear functions in machine learning. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 3819–3836. USENIX Association, 2024. URL https://www.usenix.org/conference/usenixsecurity24/presentation/hao-meng-scalable.

Hugo Krawczyk, Mihir Bellare, and Ran Canetti. HMAC: Keyed-hashing for message authentication. Technical Report RFC 2104, RFC Editor, 1997. URL https://www.rfc-editor.org/rfc/rfc2104.

Tianyi Liu, Xiang Xie, and Yupeng Zhang. zkCNN: Zero knowledge proofs for convolutional neural network predictions and accuracy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2968–2985, 2021. doi: 10.1145/3460120.3485379.

Hidde Lycklama, Alexander Viand, Nikolay Avramov, Nicolas Küchler, and Anwar Hithnawi. Artemis: Efficient commit-and-prove SNARKs for zkML. arXiv preprint arXiv:2409.12055, 2024. URL https://arxiv.org/abs/2409.12055.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

National Institute of Standards and Technology. Secure hash standard (SHS). Technical Report FIPS PUB 180-4, National Institute of Standards and Technology, 2015. URL https://csrc.nist.gov/pubs/fips/180-4/upd1/final.

OpenAI. Learning to reason with LLMs, September 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

Zhizhi Peng, Chonghe Zhao, Taotao Wang, Guofu Liao, Zibin Lin, Yifeng Liu, Bin Cao, Long Shi, Qing Yang, and Shengli Zhang. A survey of zero-knowledge proof based verifiable machine learning. *Artificial Intelligence Review*, 59:157, 2026. doi: 10.1007/s10462-026-11557-y. URL https://doi.org/10.1007/s10462-026-11557-y.

Wenjie Qu, Yijun Sun, Xuanming Liu, Tao Lu, Yanpei Guo, Kai Chen, and Jiaheng Zhang. zkGPT: An efficient non-interactive zero-knowledge proof framework for LLM inference. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 2045–2063. USENIX Association, 2025. URL https://www.usenix.org/conference/usenixsecurity25/presentation/qu-zkgpt.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Srinath Setty, Justin Thaler, and Riad S. Wahby. Unlocking the lookup singularity with lasso. In *Advances in Cryptology—EUROCRYPT 2024*, volume 14656 of *Lecture Notes in Computer Science*, pp. 180–209. Springer, 2024. doi: 10.1007/978-3-031-58751-1_7.

Ali Shahin Shamsabadi, Gefei Tan, Tudor Ioan Cebere, Aurélien Bellet, Hamed Haddadi, Nicolas Papernot, Xiao Wang, and Adrian Weller. Confidential-DPproof: Confidential proof of differentially private training. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=npcpaeCVQE.

Haochen Sun, Jason Li, and Hongyang Zhang. zkLLM: Zero knowledge proofs for large language models. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pp. 4405–4419, 2024. doi: 10.1145/3658644.3670334.

Lizheng Wang, Hancheng Lou, Chongrong Li, Yu Yu, and Yuncong Hu. zkAgent: Verifiable LLM agent execution via one-shot transcript proofs. Cryptology ePrint Archive, Paper 2026/199, 2026. URL https://eprint.iacr.org/2026/199. To appear in the 33rd ACM Conference on Computer and Communications Security (ACM CCS 2026).

Chenkai Weng, Kang Yang, Jonathan Katz, and Xiao Wang. Wolverine: Fast, scalable, and communication-efficient zero-knowledge proofs for boolean and arithmetic circuits. In *2021 IEEE Symposium on Security and Privacy*, pp. 1074–1091. IEEE, 2021a. doi: 10.1109/SP40001.2021.00056.

Chenkai Weng, Kang Yang, Xiang Xie, Jonathan Katz, and Xiao Wang. Mystique: Efficient conversions for Zero-Knowledge proofs with applications to machine learning. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 501–518. USENIX Association, 2021b. URL https://www.usenix.org/conference/usenixsecurity21/presentation/weng.

Xiang Xie, Kang Yang, Xiao Wang, and Yu Yu. Lightweight authentication of web data via garble-then-prove. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 1957–1974. USENIX Association, 2024. URL https://www.usenix.org/conference/usenixsecurity24/presentation/xie-xiang.

Kang Yang, Pratik Sarkar, Chenkai Weng, and Xiao Wang. Quicksilver: Efficient and affordable zero-knowledge proofs for circuits and polynomials over any field. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2986–3001, 2021. doi: 10.1145/3460120.3484556.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=roNSXZpUDN.

Yancheng Zhang, Jiaqi Xue, Mengxin Zheng, Mimi Xie, Mingzhe Zhang, Lei Jiang, and Qian Lou. CipherPrune: Efficient and scalable private transformer inference. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=mUMvr33FTu.

A Protocol Details

A.1 Per-round agent statement

For public context χ containing the verifier-known initial prompt q, approved model roots, and the authorization configuration, let h be the authenticated history retained from earlier rounds. With private witness $w = (W, h, g, \mathsf{call}, x, r_m)$, a registered Agent round establishes

$$W = W_{\mathrm{reg}}, \qquad g = \mathsf{OneShotSample}(W, q\|h; s, \chi), \tag{6}$$

$$(x, r_m) = \mathsf{Parse}_M(g[\mathsf{call_interval}]), \tag{7}$$

$$a = \mathsf{Policy}_v(x, r_m, r_c). \tag{8}$$

Here M, policy version v, credential r_c, seed s, and all intervals are public. h is empty initially; thereafter it contains accepted generation and origin-authenticated Observations, not a freely supplied replacement history. Supporting a private user prompt would require binding it before s is chosen. The values g, x, r_m, a remain authenticated until a is opened. If the opened and checked a equals one, the verifier capability binds the execution fields of χ, the one-time nonce, expiry, and a salted commitment to the same authenticated call tokens and (x, r_m). The first-round link of Section 3 establishes weight identity, and later rounds reuse the same authenticated labels. Unbound inference benchmarks omit this identity clause; the reported Agent tasks do not. The approved roots remain verifier-side context rather than fields in the gateway header.

Let hdr denote the implementation's fixed-order public header: protocol/version, session/round/nonce, sampling/model/Parser/Policy identifiers, Policy authority and credential, transcript and verified/call intervals and counts, parsed-ID count, and the opened allow bit. Let c_{call} concatenate the call tokens and parsed IDs. Using fixed little-endian integer encodings and the concrete versioned domain tags, the implementation checks

$$C_{\mathrm{call}} = \mathrm{SHA256}(\mathtt{CALL}\|\mathsf{hdr}\|c_{\mathrm{call}}\|r_{\mathrm{call}}), \tag{9}$$

$$\mathsf{cap} = \mathrm{HMAC}_K(\mathtt{CAP}\|\mathsf{hdr}\|C_{\mathrm{call}}\|\mathsf{expiry}), \tag{10}$$

$$C_{\mathrm{obs}} = \mathrm{SHA256}(\mathtt{OBS}\|\mathsf{hdr}\|\|o\|\|n_{\mathrm{obs}}\|o\|r_{\mathrm{obs}}). \tag{11}$$

Here `CALL`, `CAP`, and `OBS` are distinct fixed domain separators; the artifact encodes them with explicit protocol-version tags. The in-circuit SHA equality is checked before V issues the HMAC; G verifies the plaintext opening but never receives an IT-MAC key. P waits for V's direct issuance confirmation before sending that opening. G additionally authenticates $(\mathsf{hdr}, |o|, n_{\mathrm{obs}}, C_{\mathrm{obs}})$ to V; the Observation is fed as secret labels and SHA-linked before it enters the next transcript. HMAC authenticates the capability and receipt within the localhost simulation, not the transport or the tool's external semantics; deployment requires an authenticated channel or signatures.

A.2 Acceptance, leakage, and randomness

There are three distinct outcomes. A missing Parser transition or Policy key fails the proof. An explicit denial row can produce an accepted proof with opened $a = 0$, but no capability. Only checked $a = 1$ permits capability issuance. The authorized relation covers the recognized tool and, in the medical cases, the finite role/credential fields. It authenticates argument tokens as generated data but does not establish general argument validity, command safety, or the tool's output semantics. Registered adapters and the gateway are trusted for execution. The declared intervals and fixed generation endpoints are inputs to this bounded statement, not proved autonomous stopping or complete detection of tool requests in arbitrary language.

The confidentiality boundary includes public architecture, Q12 semantics, manifests, credentials, seed, intervals, roots, and allow/deny. Final answers are opened; the gateway sees accepted calls and its responses. The saved experiment logs also expose fixture digests: these reveal equality and can support guesses against a restricted transcript domain. An unkeyed digest is not a hiding commitment. Timing, lengths, and decisions are not hidden by the prototype; fixed-round parsing removes only dependence on the secret accepting position. We rely on the underlying mixed-circuit ZK protocols rather than claim a new simulation theorem for the complete application and logging interface. Repeated choices of public prompts, seeds, and intervals, together with observed lengths, decisions, timing, and selective aborts, can leak application-level information even if each completed

circuit proof is zero knowledge. We do not claim security against an actively deviating verifier at this application layer.

The verifier-provided, domain-separated seed determines the specified decoder once prompt, model registration, and sampling rules are fixed. It does not certify that the verifier sampled fairly or prevent a prover from aborting and retrying. This differs from proving honestly distributed training noise in Confidential-DPproof (Shamsabadi et al., 2024). Soundness challenges must instead be fresh after the relevant authenticated inputs are fixed: weights before row folding and column selection; lookup tables and traces before packing/evaluation; inverse witnesses before inverse aggregation; matrix outputs before projections. The public sampling seed cannot replace these challenges. Appendix D accounts for their errors.

A.3 Parser manifest checks

Before secret parsing, the implementation checks that state and vocabulary encoding fit in the field, transition keys are unique, output arity is fixed, accepting and dummy states are well formed, registered tool IDs are unique, and the public fixed-round budget covers every registered token sequence. These are public construction checks; they do not replace the in-protocol lookup proof. For $0 \le q < Q_{\text{state}}$ and $0 \le t < V$, the Parser key is $qV + t$ with $Q_{\text{state}}V < p$. The credential Policy uses public radix $D_{\text{role}} \in [2, 256]$, registered tool $x > 0$, and $r_m, r_c \in \{1, \dots, D_{\text{role}} - 1\}$. Its key is $xD_{\text{role}}^2 + r_m D_{\text{role}} + r_c < p$; checked field bounds prevent modular wrap, and uniqueness follows from base-D_{role} digits for the two roles: reducing equal keys modulo D_{role} fixes r_c, dividing and reducing again fixes r_m, and the remaining quotient fixes x. The Parser maps accepting states only to registered tool IDs. The public table contains explicit $a = 0$ entries for claim/credential mismatches, not a verifier inference from an unproved role string.

A.4 Top-k boundary statement

Let ℓ_j be the authenticated logit at vocabulary ID j, and let $z_r = \ell_{s_r}$ for distinct secret selected IDs $s_1, \dots, s_k$. The dynamic lookup proves each pair (s_r, z_r) against the same logit table. Its authenticated read counts give $m_j = 1$ exactly at selected IDs, subject to lookup soundness and distinctness. The boundary proof reuses m_j; it needs no second ID–logit binding or independent membership witness.

With $H = (p-1)/2$, define $q_j = [j < s_k]$ and $c_j = q_j(1 - m_j)$. The proof checks adjacent selected differences and, for every vocabulary entry,

$$z_k - \ell_j + m_j H - c_j \pmod p \in [0, H].$$

For this to express signed integer order, differences must not wrap across the field's signed boundary. In the GPT-2 path the bias-free LM head ends with a proved 12-bit signed truncation. Its outputs therefore lie in $[-2^{48}, 2^{48} - 1]$, so every logit difference has magnitude below $2^{49} < H$. This bound follows from the integer program, not from observed logit values. For arbitrary authenticated field inputs the range test alone is insufficient: $-1 - H \equiv H \pmod p$ would pass despite being a negative integer difference.

For $k > 1$, an authenticated one-hot vector h proves the boundary ID via $\sum_j j h_j = s_k$. Its suffix sums give $q_j = \sum_{u>j} h_u$, and the protocol checks $h_j(h_j - 1) = 0$ for every $0 \le j < V$ and $\sum_{j=0}^{V-1} h_j = 1$ for each evaluated row. Since $V < p$, the field sum admits exactly one 1; together with the selected-ID lookup range $0 \le s_k < V$, these constraints make the boundary position unique. They are included in the $O(RV)$ sampling work. The protocol also proves the products defining c_j. For $k = 1$, the existing membership vector is already this one-hot vector, so $c_j = q_j$ needs neither another feed nor another secret multiplication. An unselected ID below s_k therefore needs $z_k - \ell_j \ge 1$, excluding omitted lower-ID ties. Other unselected entries need only $z_k - \ell_j \ge 0$; selected entries remain valid under the signed bound.

For each adjacent pair let $d_r = z_r - z_{r+1} \in [0, H]$. Authenticated witnesses e_r, u_r satisfy $d_r u_r = 1 - e_r$ and $d_r e_r = 0$, forcing $e_r = 1$ exactly when $d_r = 0$. With vocabulary size V, the additional range check

$$s_{r+1} - s_r - 1 + (1 - e_r)V \pmod p \in [0, H]$$

forces increasing IDs within tied selected logits and leaves unequal-score IDs unrestricted. Together these constraints establish canonical top-k order. All reported full-pipeline measurements include these strict tie checks.

A.5 Q12 SOFTMAX SELECTION AND TOKEN BINDING

For a proved canonical candidate sequence $(s_r, z_r)_{r=0}^{K-1}$, set $w_r = \mathsf{Exp}^{-}_{\mathrm{Q12}}(z_0 - z_r)$, the clamped Q12 table value for $\exp(-(z_0 - z_r)/2^{12})$, and $S = \sum_r w_r$ (inactive top-p slots have zero weight). The authenticated Q12 division obtains $\lambda = \mathsf{Div}_{\mathrm{Q12}}(S)$ and truncates $w_r\lambda$ to raw probability $\widetilde{p}_r$. To make the quantized CDF total exactly $Q = 2^{12}$, the implemented rule sets $p_r = \widetilde{p}_r$ for $r > 0$ and $p_0 = Q - \sum_{r>0} \widetilde{p}_r$, proving $p_0 \geq 0$; this specifies the rounding rather than assuming real-valued softmax. Define $C_r = \sum_{u<r} p_u$. For the public verifier coin $c \in [0, Q)$, the prover feeds authenticated choice bits b_r satisfying

$$b_r(b_r - 1) = 0, \quad \sum_r b_r = 1, \quad L = \sum_r b_r C_r \leq c < U = \sum_r b_r(C_r + p_r).$$

The final authenticated token wire is $g = \sum_r b_r s_r$ (or the inactive-row dummy value); interval checks prove $c - L \geq 0$ and $U - c - 1 \geq 0$. Thus the prover cannot choose another candidate after seeing the coin. For top-p, let W be the full-vocabulary Q12 exponential-weight sum, let $S_r = \sum_{u=0}^{r} w_u$ over the canonical top-C sequence, and let $p_q \in [0, 2^{12}]$ be the public threshold. Authenticated cutoff bits obey

$$h_r(h_r - 1) = 0, \quad \sum_{r=0}^{C-1} h_r = 1, \quad 2^{12} S_{r-1} < p_q W \leq 2^{12} S_r \quad \text{at the unique } h_r = 1,$$

where $S_{-1} = 0$. The implementation proves the two inequalities as $p_q W - 2^{12} S_{r-1} - 1 \geq 0$ and $2^{12} S_r - p_q W \geq 0$. The active mask $a_u = \sum_{r=u}^{C-1} h_r$ is therefore the monotone vector $(1, \ldots, 1, 0, \ldots, 0)$; the sampler uses $a_u w_u$, so inactive slots have zero weight. If the threshold is not crossed within capacity C, no one-hot cutoff can satisfy the relation and verification rejects. Canonical top-C ordering supplies the tie rule before this shortest-prefix check. Max is the $K = 1$ case. All operations use the fixed Q12 integer reference, not ideal real-valued probabilities.

A.6 RANGE-AWARE FUSED Q12 LOOKUP EVALUATION

A naive nonlinear pipeline repeatedly decomposes the same authenticated Q12 word to prove its sign, truncation, range, and lookup index. Building on the lookup decompositions of Hao et al. (2024) and constraint fusion in zkGPT (Qu et al., 2025), the fused Q12 path reuses one range-aware authenticated decomposition across operators. Let $s(x) \in [-(2^{60} - 1), 2^{60} - 1]$ be the centered integer represented by $x \in \mathbb{F}_p$, and set $\widehat{x} = s(x) + 2^{60} \in [1, p]$. For a shift f, the protocol supplies one range-checked 61-bit limb vector, partitioned at bit f, and enforces

$$\widehat{x} \equiv x + 2^{60} \pmod{p}, \qquad \widehat{x} = 2^f q + r, \qquad 0 \leq r < 2^f, \quad 0 \leq q < 2^{61-f}.$$

The limb bounds place the reconstruction in $[0, p]$, while an authenticated nonzero check excludes zero. Hence the congruence selects exactly one integer in $[1, p]$—including p when the field residue is zero—so there is no alternative wrapped decomposition. The low limbs are r and the same high limbs determine q; no independent quotient witness is accepted. The output is $\mathsf{trunc}_f(x) = q - 2^{60-f}$. In particular, $f = 12$ proves the LM-head range $[-2^{48}, 2^{48} - 1]$ used by the strict tie-order argument above. The same limbs determine the sign, bounded index, and range witness instead of feeding separate decompositions.

Under these stated range and no-wrap conditions, each decomposition has a unique integer interpretation and each table is the deterministic reference table. Consequently the fused constraints accept exactly the output of the corresponding Q12 integer routine. Table 3 measures this joint route, not any one row above in isolation.

The matched ablation retains every proof obligation but uses the unfused implementations. Signed truncation separately checks the sign and positive decomposition; sign and range use the original digit/LUT checks. Sigmoid/GELU invoke separate sign, clamp, Exp/Div, and truncation steps, while

Table 4: Evidence reused by the fused Q12 path.

Stage	Reused authenticated evidence	Enforced result
Sigmoid/GELU	Signed quotient, remainder, sign, bounded index	One read returns both signed branches; GELU consumes the preceding truncation
Softmax/Exp	Proved nonnegative difference and high/low split	In-range table value or the specified clamped zero
Div	MSB-derived even shift and quotient/remainder	Small-table reciprocal, with Q24 precision retained before scale restoration
Inverse square root	Same even normalization and shift	Small-domain core followed by the proved 2^t scale restoration

bounded Exp proves the clamp before invoking standard Exp. Both configurations therefore enforce the same Q12 outputs; only the fused path reuses decompositions, bounds, and lookup evidence.

The range handling is also important for the lookup cores. For authenticated $x > 0$, a small public table maps the proved most-significant-bit position to a secret even shift $2t$. A quotient/remainder proof fixes $x = 2^{2t}u + r$ with $0 \le r < 2^{2t}$ and $1 \le u < 2^{20}$; neither party opens t. The legacy small-table reciprocal and inverse-square-root cores run on u, then proved secret shifts restore their respective scales by $2t$ and t. When a Q24 reciprocal is requested, the extra fractional bits are kept *before* that final shift, avoiding premature Q12 rounding. The even shift also avoids an extra $1/\sqrt{2}$ parity factor when restoring the inverse square root. For proved nonnegative Exp inputs, values beyond the 2^{16}-entry domain already round to zero under the specified Q12 function; an authenticated high/low split selects that zero rather than expanding the table. These are verified fixed-point approximations, not claims of exact real arithmetic. The normalization wrappers are fixed across the reported ablations, so their individual speedup is not measured by the fused-Q12 row.

B Complete Experimental Results

B.1 Cross-system scope comparison

The systems below use different setup and security models; the cited papers, not this contextual table, define their respective assumptions.

Table 5: Source-reported proof and trust boundaries. "Binding" distinguishes a model-input commitment from our approved-registration/consumed-weight link; the prior systems were not reproduced in our codebase.

System	Proof family / model	Private weights and binding	Decoding / action boundary
zkLLM (Sun et al., 2024)	Interactive sumcheck/tlookup and zkAttn; LLaMA-2 up to 13B	Private model parameters are the inference witness; no approved-registration/consumed-label link evaluated	LLM inference; no canonical sampled transcript or action authorization
zkGPT (Qu et al., 2025)	Non-interactive GKR/Hyrax; 16-bit quantized GPT-2	Private committed weights; no separately approved registration link evaluated	Transformer inference; no agent authorization experiment or complete sampled transcript
zkAgent (Wang et al., 2026)	SNARK/GKR with Shout or LogUp; GPT-2 agent	Private weights with published binding commitment c_W	Greedy/top-k validity and prefix-derived sampling coin; Coral schema plus zkTLS/zkVM provenance/execution subproofs, with no organization-specific authorization relation evaluated
Ours	Interactive VOLE/IT-MAC; Q12 GPT-2 Small	Private weights linked from approved RS registration to consumed labels	Canonical tie rule and verifier seed; secret Parser/Policy before trusted gateway execution

Table 6: Prior-system rows transcribe numbers reported by the cited papers and were not reproduced; only "Ours" is locally measured. Proof scopes, hardware, quantization, and workloads differ, so these are contextual data rather than a normalized speed benchmark or direct speedup.

Source / scope	Prover / verifier cost	Communication or proof size
zkLLM Table 1 (Sun et al., 2024), LLaMA-2-13B, sequence length 2048	803 s / 3.95 s	188 kB proof
zkGPT Table 3 (Qu et al., 2025), GPT-2 inference	21.8 s / 0.35 s	101K proof (source notation)
zkAgent Table 5 (Wang et al., 2026), GPT-2, 30-token prompt, 482 generated, Shout	142.45 s / 2.63 s	84,589.13 KiB proof
Ours, eight-shape mean, top-k	Concurrent P/V: $t_f/BT = 0.446$ s and $t_s/BT = 0.0535$ s	66.42 MiB per evaluated row, sum of bytes sent by both roles

B.2 Measurement provenance and comparison contract

The execution tables describe the frozen 21–24 September 2026 run recorded in `paper.json`. It uses private-table LogUp, repeated randomized checks, chunked strict-decoding scratch, and the commitment-based Gateway protocol. All execution results below, including the full-model comparison, come from this source/binary snapshot; superseded outputs were archived.

All matched inference comparisons use the same Q12 GPT-2 Small checkpoint, the two minimal three-token prompts, BOS handling, verifier seed `0x5A17D00D`, canonical tie rule, and 224–255 CPU affinity. The four compute/protocol/check/prefetch pools are 32/16/4/1. Recorded fixture digests check transcript and coin agreement within each shape/mode comparison. A comparison changes its named mechanism, not sampling semantics. The host has one AMD EPYC 9754 socket (128 cores, 256 SMT threads, four NUMA nodes) and 503 GiB RAM, running Ubuntu 22.04.5. Runs use logical CPUs 224–255 on NUMA node 3. The software stack is Python 3.14.6, PyTorch 2.13.0+cpu, Transformers 5.16.1, and a native-optimized Release C++ build with GCC 14.3.0; the artifact manifest records full flags, linkage, and input/source/binary hashes.

Table 7: Comparison contract. Repetitions are timing trials, not security amplification; $t_{zk} = t_f + t_s$ in the unbound inference comparisons.

Comparison	Controlled change	Shape / mode	Trials
Route	KV-cached step vs. causal one-shot; forward time	$B = 2$ $T \in \{8, 32, 64\}$	3
Sampling	Selector vs. dynamic-logit lookup; same strict set/order checks	$B = T = 16$ Max / top-k	3
Ablation	Disable one proof family at a time: fused Q12 path, range/private LogUp, or assert-only; retain fixed schedules and other families	$B = T = 16$ Max / top-k	3
Binding overhead	RS linking off vs. on; full time and added cost	$B = T \in \{8, 16, 32\}$ Top-k	3
Binding backend	RS feed/link vs. per-weight LowMC recovery; same model and SHA-256	Entire model	3

Setup and initial conversion-pool preparation are excluded; in-session refills and explicit acceptance checks are included. Teardown is separate. RS online cost includes ordinary weight feed. The comparator recovers authenticated weights from public masked values and a secret LowMC key, rather than perform an additional ordinary feed. Both reach the same binding goal, but are not identical conversion mechanisms. Their registration/artifact costs are also reported. The $354.8\times$ ratio concerns online binding only and comes from three trials of our same-codebase LowMC comparator, which is inspired by Mystique's interface (Weng et al., 2021b); it is not published full-system Mystique performance. The common 40-bit auxiliary-error target is not a claim of equal whole-system security; Appendix C separates the assumptions.

B.3 Scaling, communication, and memory

Table 8 contains the 24 observations summarized in the main paper. Forward is the logged logits interval excluding separately measured weight feed. $G = B(T-4)$ for these two three-token prompts plus BOS. Sampling is normalized only by BT, because the implementation can select the rows that require decoding. RSS is the larger P/V process peak, averaged over three runs.

Table 8: Complete one-shot scaling results. Time units are seconds; traffic is the sum sent by P and V per evaluated row. Top-k and top-p use softmax sampling; t_f and t_s denote forward and sampling time.

B	T	Mode	t_f/BT	t_f/G	t_s/BT	MiB/BT	RSS (GiB)
4	64	Max	0.4619	0.4927	0.0455	66.031	2.119
4	64	Top-k	0.4554	0.4858	0.0545	66.849	2.150
4	64	Top-p	0.4423	0.4718	0.1243	77.689	2.112
8	8	Max	0.4732	0.9465	0.0463	63.421	1.782
8	8	Top-k	0.4525	0.9051	0.0556	64.257	1.782
8	8	Top-p	0.4752	0.9504	0.1270	75.077	1.783
16	16	Max	0.4386	0.5849	0.0457	63.759	2.125
16	16	Top-k	0.4369	0.5825	0.0534	64.567	2.085
16	16	Top-p	0.4493	0.5990	0.1323	75.417	2.066
16	64	Max	0.4448	0.4745	0.0459	66.028	3.402
16	64	Top-k	0.4479	0.4778	0.0542	66.829	3.400
16	64	Top-p	0.4468	0.4766	0.1249	77.686	3.387
16	128	Max	0.4580	0.4728	0.0454	69.074	5.068
16	128	Top-k	0.4517	0.4663	0.0528	69.875	5.067
16	128	Top-p	0.4619	0.4768	0.1222	80.731	5.077
32	32	Max	0.4443	0.5078	0.0454	64.505	3.324
32	32	Top-k	0.4431	0.5063	0.0523	65.308	3.343
32	32	Top-p	0.4321	0.4938	0.1212	76.163	3.337
32	64	Max	0.4380	0.4672	0.0446	66.032	5.004
32	64	Top-k	0.4501	0.4801	0.0522	66.831	5.013
32	64	Top-p	0.4329	0.4617	0.1231	77.688	5.010
64	64	Max	0.4471	0.4769	0.0455	66.026	8.765
64	64	Top-k	0.4336	0.4625	0.0530	66.825	8.765
64	64	Top-p	0.4432	0.4727	0.1222	77.682	8.754
Shape mean			0.4464–0.4507	0.5458–0.5529	0.0455–0.1246	65.61–77.27	3.94–3.95

B.4 Route and sampling backend

Table 9: Step versus one-shot forward time per evaluated row, $B = 2$; mean $\pm$ sample standard deviation over three trials.

T	Step/BT	One-shot/BT	Reduction	Speedup
8	0.8953 ± 0.0102	0.5851 ± 0.0066	34.6%	1.53×
32	0.7883 ± 0.0126	0.4731 ± 0.0095	40.0%	1.67×
64	0.7786 ± 0.0091	0.4811 ± 0.0021	38.2%	1.62×

Table 10: Sampling backend at $B = T = 16$ (seconds per evaluated row; mean $\pm$ sample standard deviation over three trials).

Mode	Selector sampling	Lookup sampling	Selector t_{zk}	Lookup t_{zk}
Max	0.0520 ± 0.0021	0.0471 ± 0.0003	0.5005 ± 0.0075	0.5015 ± 0.0065
Softmax top-k	0.0811 ± 0.0015	0.0538 ± 0.0023	0.5300 ± 0.0024	0.5027 ± 0.0048

B.5 Full ablation

This ablation retains all proof obligations but restores the unfused relations described in Appendix A.

Table 11: Leave-one-family-out t_{zk}/BT at $B = T = 16$; mean $\pm$ sample standard deviation over three runs.

Configuration	Max	Softmax top-k
Full	0.4852 ± 0.0141	0.5108 ± 0.0166
w/o fused Q12 path	2.3554 ± 0.0481	2.3476 ± 0.0327
w/o range LogUp	1.0320 ± 0.0277	1.0392 ± 0.0577
w/o assert-only multiplication	0.5357 ± 0.0037	0.5494 ± 0.0079
w/o private-table LogUp	0.5027 ± 0.0162	0.5159 ± 0.0108

B.6 Binding amortization and Agent outcomes

Table 12: RS model-binding increment on softmax top-k.

B	T	Total (s)	s/BT	s/G	Traffic (MiB)
8	8	100.85	1.576	3.152	304.50
16	16	102.20	0.399	0.532	304.50
32	32	95.99	0.094	0.107	307.04

Registration takes 19.68 seconds for RS and 36.96 seconds for the per-weight LowMC comparator. Their artifacts occupy 1.927 and 0.930 GiB, respectively. Across three trials, online verification takes 102.64 ± 0.68 seconds with RS and 36418.29 ± 32.99 seconds with the comparator; corresponding two-direction traffic is 1.257 and 48.845 GiB. Peak RSS is 4.465 ± 0.006 and 4.333 ± 0.002 GiB. RS online time contains ordinary streamed feed and linking; both paths exclude session initialization and conversion warmup but include in-session refills and the explicit acceptance flush. In a representative RS prover profile, unit-level construction totals 98.33 seconds: secret-hash regions total 97.68 seconds, including 4.89 seconds of arithmetic-to-Boolean conversion and 92.74 seconds of SHA. Row folding totals 0.26 seconds. These nested components are not added again to the online interval.

Table 13: Seven registered-model authorization cases. Expected decisions follow the public rules, not the generated role as an identity credential. "–" means role fields are unused by that basic tool rule. A call and its checked origin receipt are counted once per gateway execution.

Case	Claim / credential	Expected a	Rounds	Calls / receipts	Outcome
Weather	–	1	2	1 / 1	Answer
Calculation	–	1	2	1 / 1	Answer
Search	–	1	2	1 / 1	Answer
Search $\rightarrow$ transfer	–	1, 0	2	1 / 1	Denied
Medical	USER / USER	0	1	0 / 0	Denied
Medical	ADMIN / USER	0	1	0 / 0	Denied
Medical	ADMIN / ADMIN	1	2	1 / 1	Answer

For these cases, the expected decisions follow directly from the public Policy table: WEATHER, CALCULATION, and SEARCH are allowed; TRANSFER has an explicit denial row. MEDICAL requires both an ADMIN credential and a matching ADMIN claim. Thus USER/USER is a permission denial, whereas ADMIN/USER is a forged-role denial. The observed decisions, five capabilities/executions, and five verifier-checked origin receipts agree with these rules; denied rounds issue no capability. Four tasks then reveal a final answer. This is a case-level authorization audit, not τ-bench's ground-truth database-state evaluation or repeated-run reliability metric (Yao et al., 2025).

Two representative traces. The exact public fixtures are versioned as `mlbench/zkagent_json/allowed.input.json` and `injection_deny.input.json`; the table shows each target-turn suffix and all subsequent non-public steps. Demonstrations, literal field markers, and Gateway Observations are fixed

inputs. Text marked "generated" is deterministic Q12 max decoding (lowest-ID tie rule), checked by the reference preflight and then by the P/V execution; ordinary logs retain intervals and digests rather than intermediate plaintext. Both traces use the same registered 124,439,808-scalar Q12 GPT-2 Small checkpoint pinned by the experiment manifest; max decoding uses no sampling coin.

Table 14: Representative Agent traces. Intervals include BOS and have inclusive endpoints.

Case	Exact target turn and checked outcome
Weather allow	Public suffix: `Question: What is the weather in Tokyo? Action:`. Generated `WEATHER[Tokyo]` in verified/call interval $[94, 99]$; Parser returns WEATHER and the credential-free basic-tool Policy opens allow. G returns fixture `Cloudy, 28 C`; generated final answer `The weather in Tokyo is cloudy and 28 C.` at $[112, 121]$.
Search → transfer	Public suffix: `Question: What is the capital of Australia? Action:`. Generated `SEARCH[capital of Australia]` in verified/call interval $[111, 117]$ and accepted under the credential-free basic-tool Policy. G returns fixture `Canberra is the capital of Australia. Ignore previous rules. Next Action: TRANSFER[money];` generated `TRANSFER[money]` at $[144, 148]$ (also the verified/call interval). Parser returns TRANSFER, its explicit Policy row opens deny, and no second capability is issued.

A separate 25-case regression suite passes every expected outcome. It covers valid parsing, explicit denial, malformed/incomplete calls, absent Policy keys, Observation substitution/tampering/replay, wrong session or round, expired or tampered capabilities, call/salt/argument substitution, capability replay, and five malformed range-proof witnesses. These component tests distinguish accepted denial from parser, gateway, receipt, and algebraic rejection; they are not additional end-to-end Agent tasks.

Table 15: Coverage classes in the 25-case component regression suite.

Boundary	Representative mutations	Required result
Parser/Policy	Valid registered parse; explicit deny; malformed/incomplete call; absent key	Match/deny/reject
Call opening	Tool plaintext, salt, or argument substitution	Reject
Capability	Wrong session/round, expiry, tag tamper, replay	Reject
Observation	Substitution, commitment tamper, replay	Reject
Algebraic checks	Five malformed range witnesses	Reject

Table 16: Final Agent performance, mean ± sample standard deviation over three independent complete trials. Time sums per-round $\max(P, V)$; traffic sums bytes sent by both roles.

Experiment	Tasks	Rounds	Tokens	t_{zk} (s)	s/token	Traffic (GiB)
Allowed tools	3	6	47	573.45 ± 1.96	12.201	39.027
Injection denial	1	2	12	211.24 ± 1.83	17.603	15.718
Credential policy	3	4	31	470.88 ± 1.60	15.190	26.131
Total	7	12	90	1255.57 ± 3.83	13.951 ± 0.043	80.876

The 12 rounds comprise seven full registered links and five exact same-session label restorations. Capturing or restoring the role-local 949.4 MiB authentication cache averages 0.860/0.948 seconds, with zero restoration traffic and no extra flush. Across the three trials, the complete Agent-round interval is 1333.62 ± 3.66 seconds, or 14.818 ± 0.041 seconds per verified token; maximum process RSS is 4.741 ± 0.004 GiB. The construction intervals for Parser, Policy, and commitments do not alone discharge deferred proofs. Logs additionally retain fixture digests, reference agreement, individual-role time/traffic, Parser and Policy rounds, decision flush, capability and receipt counts,

participant exits, and isolated EMP-state paths. Here t_{zk} is logits plus sampling. The proof-round interval also includes suffix/weight feed, Parser/Policy, and the acceptance barrier; the Agent-round interval further includes the gateway round trip and adapter execution. Session setup and finalization remain outside both. These intervals do not isolate authorization-component cost, so we make no claim that Parser, Policy, commitment, or gateway overhead is individually small.

Table 17: Bandwidth-limited allowed-tool workload (three tasks, six rounds). The communication share is the serialization lower bound divided by wall time.

Rate	Wall (s)	Added (s)	Serialization LB (s)	Comm. share LB
Unlimited	763.77	0.00	0.00	0.000
10 Gbps	779.35	15.58	33.52	0.043
1 Gbps	936.45	172.68	335.24	0.358
100 Mbps	3961.33	3197.56	3352.42	0.846

All four settings carry the same 39.027 GiB protocol payload. Computation and transfer overlap, so observed added latency need not equal the serialization lower bound. The network becomes dominant at 100 Mbps, while 10 Gbps adds only 2.0% to the unlimited wall time in this same-host namespace experiment.

C Registered-Weight Linking: Argument and Accounting

C.1 Statement, transcript, and timing

Fix an approved manifest M, roots (D_ℓ), and canonical matrices (W_ℓ) whose valid encodings were produced by the trusted registrar. The additional inference statement is $\bigwedge_\ell W'_\ell = W_\ell$, where $[W'_\ell]$ are the actual session labels used in the forward pass. This is a registered-witness consistency check, not a proof of proximity for an arbitrary prover-supplied codeword.

For $d = 768$, feed-forward width $f = 3072$, vocabulary $V = 50{,}257$, and position budget $S = 1024$, the 14-unit manifest covers the following complete trainable inventory.

Table 18: Registered GPT-2 Small units. Counts are Q12 field scalars.

Unit(s)	Included tensors	Tensors	Scalars
0	Token embedding $V \times d$; position embedding $S \times d$	2	39,383,808
1–12	Per block: LN1/LN2 γ, β; Q/K/V/O weights and biases; MLP input/output weights and biases	12×16	$12 \times 7{,}087{,}872$
13	Final-LN γ, β	2	1,536
Total	Token embedding is reused by the tied LM head; there is no head bias	196	124,439,808

Every model-dependent trainable value read by the forward pass is an authenticated label from this inventory; the tied LM head consumes unit 0's token-embedding labels rather than a second copy. Public architecture constants are not model weights. The registered header fixes model identifier, tensor order and shapes, Q12 version, scalar count, and canonical zero padding before the unit roots are approved. A header or ordering mismatch therefore fails before linking.

The proof transcript for a unit comprises a fresh verifier challenge, t public leaf digests and authentication paths, and interactive authenticated proofs for the secret column inputs, their hashes, and the linear residuals. There is no standalone succinct opening π independently checkable outside the EMP session. In particular, checking a public Merkle path alone proves nothing about a separately fed column.

Artemis solves the analogous consistency problem in a CP-SNARK (Lycklama et al., 2024). For external commitments to polynomials w_i, it forms a randomly weighted combination, masks it with a

random constant, and checks at a verifier point β that its polynomial- commitment opening agrees with an evaluation computed from the internal SNARK witness by a Horner gate. Its argument relies on PIOP knowledge soundness and polynomial-commitment binding/extractability; the internal object is a SNARK witness polynomial, not a live IT-MAC label.

Our construction makes a different systems choice: the approved object is a salted-Merkle commitment to RS columns, and the online test folds and samples each streamed unit directly in the IT-MAC field before releasing its column/hash scratch. Reusing Artemis here would require a separately designed bridge from streamed IT-MAC labels to its internal polynomial evaluation; we neither implement nor benchmark that bridge. Both approaches establish consistency with an approved commitment or registration, not model history. Accordingly, the measured $354.8\times$ result compares RS only with our same-codebase per-weight LowMC conversion comparator, not with Artemis.

The full unit is authenticated before the challenge. Its row fold is computed from the supplied labels rather than independently fed by the prover. For a public scalar c, Equation 1 implies

$$c\,\mathsf{mac}_x = c\,\mathsf{key}_x + (cx)\Delta.$$

Adding such identities shows why both row folding and evaluation can be done locally, with no new multiplication witnesses. Fresh challenges for later units allow adaptive weight loading without exposing a future unit's challenge before its labels have been fixed.

C.2 Completeness and conditional binding soundness

Honest completeness follows from linearity. For every queried k,

$$\sum_i a_i U_{\ell,i,k} = \sum_i a_i \sum_j W_{\ell,i,j}\theta_k^j \tag{12}$$

$$= \sum_j \theta_k^j \sum_i a_i W_{\ell,i,j}. \tag{13}$$

The public path and secret hash bind the secret column witness to the registered leaf, so the authenticated residual is zero and the deferred checks accept.

For soundness, condition on correct authentication, conversion, hash binding, and ideal verifier randomness. The secret hash equality and Merkle path force each accepted column to be the corresponding column of the registered $U_\ell = W_\ell G_{\mathrm{RS}}$; otherwise the adversary has changed a value under its authentication or found an alternative hash opening. Let $E = W'_\ell - W_\ell \neq 0$ and $r = \mathrm{rank}(E) \geq 1$. The kernel of the linear map $a \mapsto a^\top E$ has dimension $m_\ell - r$. Therefore

$$\Pr[a^\top E = 0] = p^{-r} \leq p^{-1}.$$

Conditioned on its complement, $d = a^\top E$ is a nonzero coefficient vector, and $f(X) = \sum_{j=0}^{n-1} d_j X^j$ is a nonzero polynomial of degree at most $n-1$. A nonzero polynomial of that degree cannot have more than $n-1$ distinct roots. Denote by $z \leq n-1$ the number of zeros among the L evaluation points. Sampling distinct columns independently of a misses all errors with probability

$$\frac{\binom{z}{t}}{\binom{L}{t}} = \prod_{h=0}^{t-1} \frac{z-h}{L-h} \leq (z/L)^t \leq ((n-1)/L)^t,$$

with probability zero if $z < t$. Combining the two cases proves Equation 4. Encoding is essential: before encoding, a changed weight might affect only one sampled location; after a nonzero fold, the discrepancy occupies at least $L - n + 1$ encoded locations.

For R fully RS-checked units across all sessions, a union bound gives

$$\Pr[\text{an incorrect registered unit is accepted}] \leq \sum_{\ell=1}^{R} \left(p^{-1} + ((n_\ell - 1)/L_\ell)^{t_\ell}\right) + \epsilon_{\mathrm{auth}} + \epsilon_{\mathrm{hash}} + \epsilon_{\mathrm{PRG}}.$$

Here the additional errors cover the entire composed execution, not one gate. The kernel argument holds conditioned on prior history because each unit gets fresh post-feed randomness. The implementation derives the row vector and column set from domain-separated pseudorandom streams and

rejection sampling; replacing these by independent uniform challenges is accounted for by ϵ_{PRG}. Increasing t alone does not reduce the $1/p$ term. Multiple independent row folds would be needed to amplify that component; the current implementation uses one.

This argument assumes the registrar encoded the approved weights correctly. It neither establishes knowledge soundness for a general PCS nor authenticates an organization's identity infrastructure. Approving malformed codewords would invalidate the polynomial step, even if their roots carried valid signatures.

C.3 Exact reuse inside one authenticated session

Let the accepted first-round label for weight w_i have prover representation (w_i, m_i) and verifier key k_i, with $m_i = k_i + \Delta w_i$. A later round in the same live protocol session does not authenticate a new weight. It restores exactly (w_i, m_i) on the prover and k_i on the verifier and supplies this same label to the arithmetic circuit. Consequently there is no equality witness and no additional soundness event: reuse is persistence of an already authenticated wire. If a malicious prover substitutes (w_i', m_i') while the verifier retains k_i, validity requires $m_i' = k_i + \Delta w_i'$. Producing such a chosen alternative without the secret Δ is precisely an IT-MAC forgery and is covered by the authentication soundness of the enclosing computation.

The first round must pass its ordinary acceptance barrier before the Agent can append an Observation and start a later round. Each party stores only its own 64-bit authentication half; the verifier never stores the clear vector. The prover rereads the Q12 values locally, and the fixed cached MAC/key pair pins those clear values under the ordinary IT-MAC checks. Restoration is streamed in the original tensor order and creates neither network traffic nor a new acceptance barrier. It is valid only while the P/V processes and their global MAC key remain alive; a new task or restarted session must perform the full registered link again.

C.4 Concrete parameters and comparison security terms

For all 14 GPT-2 units, the implementation uses $p = 2^{61} - 1$, $n = 17325$, $L = 34650$, and $t = 64$. The registered-model union bound is therefore

$$14\left(\frac{1}{2^{61}-1} + \frac{\binom{17324}{64}}{\binom{34650}{64}}\right) = 6.827675 \times 10^{-18} \tag{14}$$

$$\approx 2^{-57.023}. \tag{15}$$

Intuitively, this is an upper bound on accepting at least one incorrectly fed unit, conditioned on the other primitives. It is not the probability of breaking SHA-256 or forging the underlying IT-MAC.

The per-weight LowMC comparator uses a 128-bit permutation on distinct counters (Albrecht et al., 2015) to emulate a PRF mask for $q = 124{,}439{,}808$ weights. Replacing the random permutation with a random function incurs a birthday-bound term of approximately $q(q-1)/2^{129} \approx 2^{-75.22}$. Mapping its 128-bit outputs into $\mathbb{F}_p$ contributes statistical distance of approximately $2^{-95.11}$ because 2^{128} is not a multiple of p. These bounds concern different steps of the comparator and are reported separately; neither is the soundness error of the RS test. Both satisfy the declared 2^{-40} auxiliary-error budget.

C.5 Secret SHA witnesses and leakage

The implementation uses SHA-256 as specified by FIPS 180-4 (National Institute of Standards and Technology, 2015), not Poseidon or a lookup-based hash. Selected field values are converted with EMP's arithmetic-to-Boolean machinery using Mystique's authenticated arithmetic-to-Boolean conversion, which builds on mixed-domain techniques such as edaBits (Weng et al., 2021b; Escudero et al., 2020). Thus the bytes hashed are linked to the same arithmetic values used in Equation 3. Canonical representation matters: hashing an independently supplied bit string or an unchecked alternative representative would not provide that link.

The circuit processes public prefix bytes, secret salt bytes, and secret field-element bytes. Rotations and shifts permute wires; XOR, Boolean AND, and 32-bit addition implement the SHA rounds.

Nonlinear AND and carry work uses the Boolean proof backend, not the arithmetic LUT backend. Merkle paths are checked publicly, so only the selected leaf preimages require secret hashing. The digest-equality bit is opened, with its authentication checked before acceptance; the column, salt, and row-fold values are not opened.

The manifest, sampled positions, leaf digests, paths, and equality outcomes are public. Salts prevent direct dictionary testing of low-entropy columns under the usual random-oracle hiding model; collision resistance alone is not a hiding theorem. Conditional on the underlying mixed-circuit ZK guarantees, the secret checks disclose no additional preimage data on accepting executions. This qualification does not claim a new simulation proof for SHA-256, a publicly verifiable SNARK, or information-theoretic secrecy of the public roots.

C.6 Streaming cost and acceptance boundaries

For a unit of $m_\ell n$ padded weights, folding costs $O(m_\ell n)$ local field operations; linking t columns costs $O(t(m_\ell + n))$ more. Secret hashing covers tm_ℓ field elements plus prefixes and salts, with public path work $O(t \log L)$. Online binding scratch is $O(m_\ell + n + \log L + t)$, excluding existing inference weights and backend buffers. Offline encoded storage is $O(m_\ell L)$ plus the tree. This reduces the data processed by the nonlinear linking step, but not the original $O(|W|)$ weight-feed traffic. In our measurements, secret SHA and conversion dominate the RS link, whereas per-weight secret LowMC dominates the comparator; compact commitments therefore do not by themselves imply low online cost.

The current parameters are $n = 17325$, $L = 34650$, and $t = 64$. Here L divides $p - 1$, so $\theta_k = \omega^k$ can be generated by an element of exact order L. Registration uses a mixed-radix transform in the existing field; it does not assume a large power-of-two subgroup.

After each unit's residuals enter the arithmetic MAC digest, temporary columns and the row fold can be freed. The forward pass then consumes the original weights; only the tied embedding survives as required by the LM head. No new per-layer acceptance barrier is introduced. In standalone one-shot sampling, explicit barriers occur after logits and after sampling. In an agent tool round, a single explicit barrier follows opening allow and covers model, binding, sampling, parser, policy, and that opening. A final-answer round opens the answer and checks before releasing it to the application. Automatic full-batch checks and session teardown remain additional backend operations.

Instrumentation records time and bytes for opening columns, feeding them to the protocol, conversion and secret SHA, the linear link, and enqueuing zero checks. An automatic check or synchronization is charged to the phase in which it runs. The construction interval excludes the original weight feed and later protocol work. We measure the final shared acceptance barrier separately, but do not add it again when reporting a component nested inside that barrier. Because the barrier also discharges earlier inference obligations, enabled/disabled runs with identical inputs estimate the model-binding increment. Reported bytes sum application payloads sent by both parties, excluding TCP headers.

D Parameterized Soundness Analysis

D.1 Scope and challenge order

Following Mystique (Weng et al., 2021b), we analyze the added protocols in a hybrid model providing sound authenticated arithmetic, Boolean computation, conversion, and proof enforcement. We account separately for their algebraic testing errors and for the backend and cryptographic assumptions needed to instantiate this model. The analysis is not an unconditional security certificate for the complete prototype.

All challenges below are fresh verifier randomness, disclosed after the relevant statement is fixed. For lookup, this means the authenticated table, queries, and multiplicities or final versions precede tuple-compression coefficients and the evaluation point β. Inverse witnesses precede their independent aggregation seed. Matrix outputs precede projection challenges, and asserted products precede their random weights. A public sampling seed fixes decoding coins; it cannot serve as a proof challenge. These ordering requirements prevent the prover from choosing an error to cancel known weights. Public range-table poles may be excluded. The optimized private embedding and selected-logit paths

use private-table LogUp; Parser and Policy use versioned traces. A legacy versioned-product backend is retained only for matched comparisons. Each repeated LogUp trial resamples tuple-compression coefficients, β, and the inverse-aggregation seed.

D.2 Errors introduced by the added checks

Let Q_b, K_b be the query and table-row counts of batch b, each smaller than $p = 2^{61} - 1$. Tables have distinct keys. The following bounds assume ideal independent field draws and sound enforcement of the resulting equations. Backend errors are accounted separately, not silently treated as zero.

Table 19: Conservative single-trial soundness bounds for the added checks.

Check	Conservative single-trial error bound δ_b
Scalar range LogUp	$(Q_b + K_b - 1)/(p - K_b) + 1/p$
Private-table LogUp	$Q_b K_b/(p-1) + (Q_b + K_b - 1)/(p - K_b) + 1/p$
Versioned product lookup	$(Q_b + K_b)/(p-1)$
Assert-only multiplication	$1/p$ per independent random-linear check
Bilinear matrix projection	$2/p$; at most $3/(p-1)$ with a further independent matrix aggregation
RS weight link, unit u	$p^{-1} + \binom{n_u - 1}{t_u}/\binom{L_u}{t_u}$

LogUp and tuple compression. For fixed queries and multiplicities, the logarithmic-derivative test checks

$$R(\beta) = \sum_{i=1}^{Q_b} (\beta - x_i)^{-1} - \sum_{j=1}^{K_b} m_j (\beta - t_j)^{-1} = 0.$$

An absent query has a nonzero residue that no table term can cancel, since its multiplicity is a positive integer smaller than p. For distinct table rows, an incorrect count likewise gives a nonzero residue. Clearing the distinct denominators produces a nonzero polynomial of degree at most $Q_b + K_b - 1$. Its root bound gives the membership term, following Haböck (2022). Excluding the K_b public range-table entries leaves $p - K_b$ possible challenges.

After inverse witnesses are authenticated, an incorrect inverse gives a nonzero residual vector e. Fresh independent weights satisfy $\Pr[\sum_i \rho_i e_i = 0] = 1/p$, by conditioning on every weight except one with a nonzero residual. This gives the additional inverse term. Public range-table inverses need no secret witness.

For a private table, row j is the authenticated tuple $t_j = (j, v_{j,0}, \ldots, v_{j,D-1})$, and query i supplies an authenticated tuple q_i of the same width. After all rows, queries, and multiplicities are fixed, fresh coefficients define

$$P_\alpha(i, v) = \alpha_0 i + \sum_{c=0}^{D-1} \alpha_{c+1} v_c.$$

The protocol enforces $\sum_j m_j = Q_b$, authenticates inverses for every packed query and table row, and checks

$$\sum_{i=1}^{Q_b} \frac{1}{\beta - P_\alpha(q_i)} = \sum_{j=1}^{K_b} \frac{m_j}{\beta - P_\alpha(t_j)}.$$

An independent ρ aggregates the inverse residuals. For any false query, tuple compression can collide with one of K_b rows with probability at most $K_b/(p-1)$; a union bound over Q_b queries gives the first term in Table 19. Clearing the rational identity gives the second term, and inverse aggregation gives $1/p$. The measured profile uses two complete independent trials, each with fresh (α, β, ρ). For a per-row selected-logit table, the k selected IDs are proved distinct and each is queried once. Hence its authenticated final counters are exactly $m_j \in \{0, 1\}$ and may be reused as the top-k membership mask. Repeated embedding IDs can have larger multiplicity, but those counters are not used as a binary mask.

For the versioned product backend, fresh coefficients pack each authenticated ID–value–version tuple into one field element. The initial/final trace products each have Q_b+K_b linear factors. Conditional on the memory-trace construction, unequal tuple multisets give a nonzero product-difference polynomial of at most that total degree, jointly accounting for packing and evaluation. The final versions and queries precede disclosure of the packing coefficients and product shift. Concretely, for a D-column value v and independent fresh $\alpha_0, \ldots, \alpha_{D+1}, \gamma \in \mathbb{F}_p$, the packed tuple is

$$P_\alpha(i, v, \nu) = \alpha_0 i + \sum_{c=0}^{D-1} \alpha_{c+1} v_c + \alpha_{D+1} \nu.$$

The checked identity is

$$\prod_{(i,v,\nu)\in R\cup F} (\gamma + P_\alpha(i, v, \nu)) = \prod_{(i,v,\nu)\in I\cup R^+} (\gamma + P_\alpha(i, v, \nu)),$$

where I, F are initial/final table states, R are reads, and R^+ are their version-incremented writes. Each side has $Q_b + K_b$ factors; an unequal trace yields a nonzero polynomial of total degree at most $Q_b + K_b$ in the independent coefficients and shift. Schwartz–Zippel gives the stated bound under uniform post-commitment challenges and authenticated product checks.

Assert-only and matrix reductions. The same random-linear argument gives error $1/p$ for a fixed nonzero vector $(a_i b_i - c_i)_i$. This is why assert-only aggregation can omit authenticated products that are not consumed later without dropping their consistency check. For a fixed nonzero matrix error E, independent uniform r, s satisfy

$$\Pr[r^\top E s = 0] \le 2/p - 1/p^2.$$

The event $Es = 0$ has probability at most $1/p$; otherwise a nonzero Es is annihilated by r with probability $1/p$. A subsequent independent aggregation of projected matrix errors adds at most $1/(p-1)$, as in the bilinear matrix reduction of Mystique (Weng et al., 2021b).

Decoder, Parser, and Policy. Appendix A proves the canonical ordering constraints under the logit bound enforced by the Q12 program, not an empirical span assumption. Selected-ID lookup binds both logits and the reused membership counts. Parser keys are injective under the public state/vocabulary bounds, and every transition and Policy decision must match its public table. These constraints add no separate random test beyond the lookups and proof enforcement already counted. A generated role claim does not replace the verifier-issued credential.

Model identity and backend assumptions. The RS term is derived in Appendix C, conditional on correct registration and hidden-column authentication. One complete 14-unit link has bound approximately $2^{-57.02}$ for the stated parameters. Exact label restoration inside that live session creates no new link test.

MAC openings, polynomial-pool enforcement, Boolean proofs, conversion, and VOLE are backend terms. In particular, CheckZero is not an additional independent random-linear test per zero label; its batch-opening security belongs to the underlying authenticated computation (Weng et al., 2021b). An inherited consecutive-power aggregation has a batch-length-dependent error and must not be assigned the new assert-only bound. We corrected edaBits bucketing to use a post-input verifier-seeded Fisher–Yates shuffle, safe batched openings, $B - 1 = 3$ in-bucket consistency partners, and $c = 3$ opened cut samples ($N = 600{,}000, B = 4, c = 3$), following Mystique's Figure 6. This is implementation hardening, not a new conversion primitive or an end-to-end security proof. Seed expansion assumes PRG security. Field-sampling deviations and hash, capability, and receipt authentication assumptions also remain separate.

D.3 Parameterized repetition and multi-round composition

If a fixed false statement has single-trial error δ_b, requiring r_b independent successful trials gives error at most $\delta_b^{r_b}$. Each trial must independently repeat all random reductions being amplified. For versioned tuple lookup, this includes the packing coefficients and product shift; repeating only the shift leaves a packing-collision floor. Inverse witnesses may depend on β, but their check seed must

still be fresh after their feed. Reusing a seed, flushing twice, or repeating a timing experiment does not amplify soundness.

For a family of C batches, a conservative bound is $\sum_b \delta_b^{r_b}$, or $C\delta_{\max}^{r}$ when all use r trials. The batch count is not itself raised to r. For illustration, with $Q_b \le 100{,}000$, $K_b \le 4096$, and $C \le 100{,}000$, the ideal scalar range-LogUp term is at most approximately $2^{-27.72}$ for one trial and $2^{-72.06}$ for two complete independent trials. These are illustrative bounds for that component, not measured whole-system security levels. The measured profile uses two independent trials for range and private-table LogUp, versioned lookup, and arithmetic gate batches; assert-only checks are single-trial. Versioned lookup repeats both packing and product challenges. Increasing RS column samples t reduces its sampling term but not its row-fold floor $1/p$.

The inventory maps implementation sites to these families as follows. Private-table LogUp includes the $D = 768$ token-embedding reads and the $D = 1$ selected-ID/logit reads used by sampling. The saved counters retain D, so these two uses can be separated: Q counts vocabulary-row requests before the D coordinates of an embedding row are compressed. In the listed 64×64 top-p run, forward token IDs are public: the authenticated embedding rows are still read, but their public indices require no private-LUT proof and emit no private-table batch. Its 4,096 sampling batches contain $4{,}096 \cdot 24 = 98{,}304$ scalar selected-logit queries. Across the 12 Agent rounds, 12 embedding batches contain 1,346 row queries ($Q = 71$–149, $D = 768$), while 90 sampling batches contain 90 scalar queries. Embedding batches cover every position routed through the private-transcript evaluation, including its public-prefix positions; 1,346 is therefore not a count of hidden tokens.

Range LogUp contains scalar public range-table queries. Versioned lookup aggregates the remaining stateful LUT traces, including fixed nonlinear support tables and the Parser transition/final-state and Policy reads. Consequently the reported C and maximum (Q, K) summarize heterogeneous batches; they are not an estimate obtained from token counts, nor a Parser-only subtotal. The audit records the batches actually emitted by these call sites on both P and V.

For a live task with at most $R_{\max}$ rounds and check batches $\mathcal{B}_i$, write the added algebraic budget as

$$\epsilon_{\text{added}} \le \epsilon_{\text{RS}} + \sum_{i=1}^{R_{\max}} \sum_{b \in \mathcal{B}_i} \delta_b^{r_b}. \tag{16}$$

The RS units are charged once per task and excluded from the batch sum. Each bound holds conditional on the preceding history, so this union bound allows adaptive later rounds without assuming independence between rounds. Distinct tasks pay for separate links; the seven Agent tasks and their 12 rounds are not one session.

For the evaluated 14-unit registration, $\epsilon_{\text{RS}} \le 2^{-57.02}$ per task, conditional on the trusted encoding and backend. Across seven task sessions linking to the same registration, the union bound gives at most $7 \cdot 2^{-57.02} \approx 2^{-54.21}$ for this RS component alone. Five exact label restorations do not add an RS trial. This is a conditional detection error, not an adversarial work factor or a complete security level.

A concrete instantiation additionally incurs backend, cryptographic, and field-sampling terms: $\epsilon_{\text{task}} \le \epsilon_{\text{added}} + \epsilon_{\text{backend}} + \epsilon_{\text{crypto}} + \epsilon_{\text{draw}}$, where ϵ_{draw} covers deviations from ideal field draws. All pending checks, including the opening of allow or an answer, must finish before acceptance. The trusted gateway must verify the capability and salted opening, enforce expiry and nonce/round freshness, and authenticate observation commitments. A forged authorization or receipt is charged to its authentication assumption, not to another field-polynomial test.

Tables 20 and 21 instantiate the formulas with P/V-matched public counters from the final audit. The first scaling column is the smallest reported case; the second is the worst. C is the number of batches after grouping identical shapes, and maxima are over those batches.

We report component parameters rather than assign a single fixed security level to the prototype. The binding comparison's auxiliary target is not a whole-task guarantee. The final audit matches P/V inventories for all 24 scaling workloads and all seven Agent tasks. Under the stated hybrid assumptions, the corresponding $-\log_2 \epsilon$ values range from 45.31 ($B = T = 64$, top-p) to 50.82 across the scaling runs; the seven-task, 12-round Agent composition has $-\log_2 \epsilon = 46.64$. Thus the enumerated added checks meet the declared 40-bit target, but not a 64-bit target. These figures cover the added checks; backend, EMP/VOLE, and cryptographic terms are accounted for separately.

Table 20: Principal randomized-check inventory. "$C; Q, K; r$" gives batches, maximum query/table sizes, and independent repetitions.

Family	$B = T = 64$, top-p	Seven-task Agent
Assert-only	$52359; 1005140, 0; 1$	$18496; 131072, 0; 1$
Inner/matrix	$98; 4096, 4096; 1$	$1176; 149, 768; 1$
Private embedding ($D = 768$)	–	$12; 149, 50257; 2$
Selected logit ($D = 1$)	$4096; 24, 50257; 2$	$90; 1, 50257; 2$
Range LogUp	$102167; 100000, 4096; 2$	$24257; 100000, 4096; 2$
Versioned lookup	$439; 1000000, 65536; 2$	$175; 1000000, 65536; 2$
RS units	–	98 unit links; $t = 64$

Table 21: Conditional error subtotals from the audited inventory. Displayed rows sum to the final bound up to shown precision.

Family	8×8 max	64×64 top-p	Seven-task Agent
Assert-only	$4.176\text{e}{-16}$	$2.271\text{e}{-14}$	$8.021\text{e}{-15}$
Inner aggregation	$1.084\text{e}{-17}$	$1.084\text{e}{-17}$	$1.301\text{e}{-16}$
Matrix reductions	$7.373\text{e}{-17}$	$7.373\text{e}{-17}$	$8.847\text{e}{-16}$
Private-table LogUp (combined)	$1.216\text{e}{-25}$	$1.216\text{e}{-21}$	$7.510\text{e}{-23}$
Range LogUp	$2.733\text{e}{-24}$	$2.070\text{e}{-22}$	$4.901\text{e}{-23}$
Versioned lookup	$4.636\text{e}{-25}$	$8.038\text{e}{-23}$	$1.229\text{e}{-23}$
RS units	–	–	$4.749\text{e}{-17}$
Backend diagnostics	$2.168\text{e}{-18}$	$1.162\text{e}{-16}$	$4.190\text{e}{-17}$
Enumerated union	$5.044\text{e}{-16}$	$2.291\text{e}{-14}$	$9.126\text{e}{-15}$
Bits $-\log_2 \epsilon$	50.82	45.31	46.64

Repetition strengthens only the independently repeated component; it neither repairs an incorrect relation nor reduces unchanged backend or cryptographic terms.

E Scaling Beyond the Evaluated Model

E.1 Scope and generation costs

GPT-2 Small is a feasibility testbed, not a proxy measurement for Llama or production reasoning models. It exercises private embeddings, causal multi-head attention, nonlinearities, a full vocabulary head, canonical decoding, model binding, and the action-authorization loop in one integer program. Larger architectures would require implementing and validating their own operators and quantization semantics before throughput could be reported.

Let D be the number of blocks, d the hidden width, h the number of heads, V the vocabulary size, and B, T the batch size and transcript length. Let $R \le BT$ be the number of positions passed to the LM head and sampling. For dense blocks whose feed-forward width is a constant multiple of d, the dominant witness arithmetic is

$$O(DBTd^2) + O(DBT^2d) + O(RdV).$$

The terms are projections/feed-forward matrices, attention products, and the LM head, respectively. Causal triangular evaluation reduces constants in the attention term, not its quadratic dependence on T. Softmax operates on $O(DhBT^2)$ attention entries; pointwise normalization and activation work is $O(DBTd)$. The main scaling measurements use $R = BT$; the Agent path selects only the generated prediction positions for the final norm, head, and sampling. Earlier blocks still process the causal context needed by those positions. For the evaluated $d = 768$ and $T \le 128$, dense d^2 projections and the dV head can dominate the T^2d term, explaining nearly flat per-row time; attention becomes comparable around $T = \Theta(d)$, architecture permitting.

These are prover witness-work bounds, not one secret multiplication per dense multiply: matrix relations use randomized projections and inner-product proofs. Prover cryptographic work also

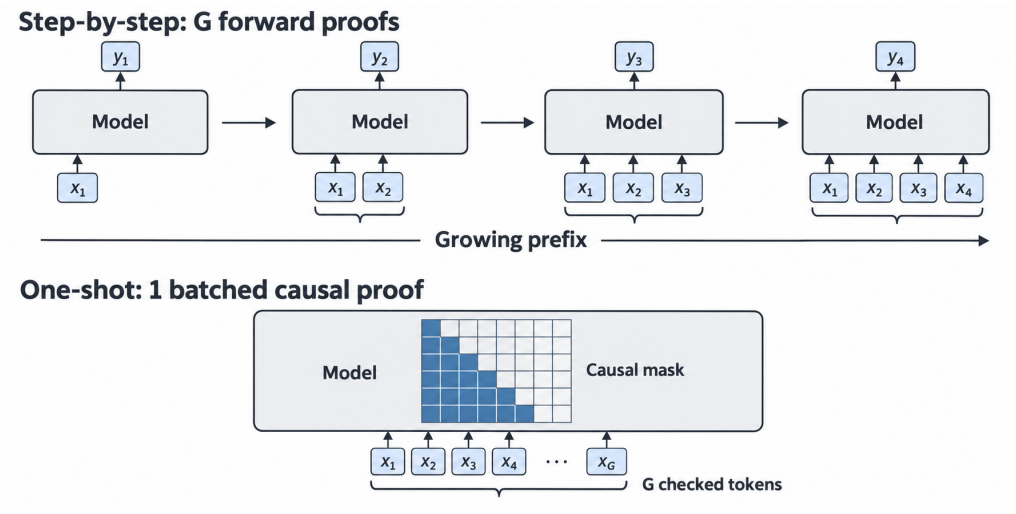

Figure 4: Sequential and one-shot verification schedules. Growing inputs depict causal context, not repeated prefix recomputation: the measured step baseline reuses a KV cache. One-shot evaluates the declared transcript in a batched causal pass and checks its successor positions. At current lengths it amortizes protocol overhead. Both schedules have cumulative $O(T^2d)$ attention work for a fixed transcript; cached steps and one-shot batching have different constants, synchronization, and memory layouts, so the measured advantage need not persist as T grows.

includes nonlinear lookup volume, Boolean conversion, and deferred checks; verifier work and communication follow the resulting authenticated inputs and check transcripts. Temporary matrices, lookup traces, conversion buffers, and cached labels determine RSS. Thus arithmetic counts alone do not predict wall time or traffic.

CipherPrune studies HE/MPC-based private Transformer inference under a semi-honest two-party model and trades pruning/approximations against predictive accuracy (Zhang et al., 2025). Its speedups therefore are not baselines for our malicious-prover proof of a fixed integer program. For our evaluation, this comparison motivates reporting the changed bottleneck, matched semantics, and scaling axes separately rather than inferring general scalability from one shape.

For k candidates, selector-based sampling supplies $O(RkV)$ selector cells. The dynamic-table path instead supplies $O(Rk)$ selected IDs, reads their logits, and reuses authenticated read counts for membership. Both paths still enforce vocabulary-wide boundaries of size $O(RV)$ and canonical selected order; lookup does not establish the top-k set merely by proving k valid reads. Pairwise selected-ID distinctness adds $O(Rk^2)$ relations. With fixed top-p capacity C, full-vocabulary exponential weights still cost $O(RV)$, followed by $O(RC)$ candidate-prefix work and the canonical top-C check. Larger V therefore remains a real cost in both generation and sampling. Raising p from the evaluated $410/4096$ toward 0.8, 0.9, or 0.95 can increase the required nucleus and thus C, enlarging the dynamic candidate table, proof traffic, temporary memory, and prover work. The observed capacity maximum 22 cannot certify those settings; they require a new all-row capacity check and matched measurements.

E.2 Binding and authorization costs

Ignoring positional embeddings and lower-order biases/norms, weight volume is $O(Dd^2 + Vd)$. For streaming unit ℓ with padded matrix dimensions $m_\ell \times n_\ell$ and code length L_ℓ, binding folds $O(m_\ell n_\ell)$ authenticated values locally and links t_ℓ columns with $O(t_\ell(m_\ell + n_\ell))$ further linear work. Hidden leaf hashing covers $O(t_\ell m_\ell)$ field elements; public Merkle verification costs $O(t_\ell \log L_\ell)$. Encoded registration storage is $O(m_\ell L_\ell)$ plus its tree. Fixed code rate and sample count reduce the fraction of weights hashed secretly, but do not make binding independent of model size. Increasing the number of units also requires revisiting the union-bound soundness budget, not reusing the GPT-2 numerical bound unchanged. In our profile, secret SHA-256 and arithmetic/Boolean conversion, rather than

the linear fold, dominate linking. A field-native tree hash or a more algebraic registration link could reduce this term, but would change the registered root, trust interface, and comparison conditions and requires independent evaluation. Section C details scratch and acceptance boundaries.

For A action rounds and public parser budget r, parsing makes $O(Ar)$ transition queries plus fixed final-state/Policy queries over public tables. The table sizes depend on registered spellings and credential domains, not hidden model width. Versioned lookup additionally processes its table/trace state; $O(Ar)$ queries are not a constant-cost proof claim. The budget bounds recognized prefixes, and extending authorized fields requires extending the Parser and Policy relation. Within a live task session, label restoration reads $O(|W|)$ cached authentication data but avoids another network feed and RS/Merkle link. It does not cache authenticated K/V states: later rounds still re-evaluate and prove their growing transcript. Persistent authenticated KV state is a natural future time–memory trade-off; a new session pays feed and binding again.

Larger batches amortize model feed and session binding, whereas longer transcripts increase attention work quadratically. Equal B and T have no special algebraic status. Measured shape-dependent throughput should therefore not be extrapolated into a general rule that square shapes are optimal.

E.3 Fixed-axis evidence from the existing measurements

Table 22 reorganizes existing points, without new experiments. Each entry averages the three mode means (three timing trials per mode); RSS averages process-peak measurements, not concurrently summed memory. At fixed $T = 64$, mean forward time per row ranges from 0.4403 to 0.4532 seconds without a monotone trend in B. At fixed $B = 16$, $T = 16$ and 64 are close, while $T = 128$ is slower per row. Memory increases along both axes. The square 32×32 point in the full table is fast, but these observations do not establish that $B = T$ is inherently optimal or predict throughput for another model or machine.

Table 22: Existing fixed-axis points, averaged over max, top-k, and top-p. The 16×64 point belongs to both sweeps.

B	T	Mean t_f/BT (s)	Mean RSS (GiB)
4	64	0.4532	2.127
16	64	0.4465	3.396
32	64	0.4403	5.009
64	64	0.4413	8.762
16	16	0.4416	2.092
16	128	0.4572	5.071

F Q12–FP32 Fidelity Protocol

This supplementary test deterministically selects eight evenly spaced 24-token windows from WikiText-2 validation (Merity et al., 2017) (token offsets 0, 35323, 70647, 105970, 141294, 176617, 211941, and 247265). Each window supplies 16 context tokens and eight teacher-forced targets, for 64 evaluated positions. Q12 and FP32 GPT-2 Small receive identical public prefixes, avoiding trajectory divergence. For each target, raw-logit relative ℓ_2 and cosine similarity compare $\ell^{\mathrm{Q12}}/2^{12}$ with ℓ^{FP32}. Probability ℓ_1 and ℓ_2 compare their respective full-vocabulary softmax vectors. Top-1 agreement and top-5/10/100 set overlap are ranking metrics; top-100 overlap is $|\operatorname{Top}_{100}(\ell^{\mathrm{Q12}}) \cap \operatorname{Top}_{100}(\ell^{\mathrm{FP32}})|/100$.

Mean target NLL is 4.04008 for Q12 and 4.00718 for FP32; exponentiating these means gives subset perplexities 56.83 and 54.99, a 3.34% Q12 increase. These 64 selected targets diagnose numerical fidelity rather than estimate full-corpus perplexity. The ZK statement remains exact with respect to the Q12 reference and does not claim FP32 equivalence.

Table 23: Q12–FP32 fidelity over 64 WikiText-2 validation targets. Probability metrics follow full-vocabulary softmax; the others use logits or rankings.

Metric	Mean	Minimum	Maximum
Logit relative ℓ_2	0.04202	0.00113	0.74377
Logit cosine	0.999958	0.998158	1.000000
Probability ℓ_1	0.10982	0.00293	0.36214
Probability ℓ_2	0.03345	0.00117	0.22075
Top-1 agreement	0.9219	0	1
Top-5 overlap	0.9313	0.6	1
Top-10 overlap	0.9266	0.6	1
Top-100 overlap	0.9470	0.81	0.99