Adaptive-Controlled Mutual TLS for Large Language Model Systems

Lui Zheng[1], Roger Zhu[1], Amit Agrawal[2], and Carol Lamore[2]
[1]Department of Computer Science, University of Chicago, USA
[2]Scale AI

***Abstract*—Mutual Transport Layer Security (mTLS) underpins authenticated, confidential communication across modern service meshes, but its deployment stance in machine-learning platforms is typically static—fixed cipher suites, certificate lifetimes, and re-authentication schedules chosen for worst-case threats rather than observed risk. Large Language Model (LLM) serving pipelines exacerbate this rigidity: traffic is bursty, topologies reconfigure dynamically under autoscaling, and sensitive artifacts such as prompts, training features, and evaluation data traverse heterogeneous substrates. In this paper we argue that mTLS for LLM systems[1] should be governed by adaptive control rather than static policy. We formalize a feedback loop that ingests multi-modal telemetry—connection error codes, handshake latencies, anomaly scores from request semantics, workload attestation freshness, and service-level objective (SLO) drift—and outputs fine-grained adjustments to transport posture: client-certificate renewal cadence, certificate path length and key type selection, session resumption eligibility, early data gating, proof-of-possession challenges, and revocation propagation thresholds. The controller targets two coupled objectives: maintain cryptographic assurances (mutual authentication, forward secrecy, and replay resistance) while bounding the cost of security on tail latency and throughput during high-load inference.**

We instantiate this idea as Adaptive-mTLS, an architecture that integrates a control plane with the service mesh and LLM gateway. A risk estimator synthesizes indicators into a bounded score R_t; a stability-aware PI controller transforms R_t and SLO error into actuations on handshake intensity and credential agility; and a policy compiler realizes actuations as concrete TLS 1.3 parameters, X.509 constraints, and SPIFFE-style workload identities. To preserve end-to-end guarantees, every adaptation is compiled to a verifiable change set and rolled out with canary governance. We couple the controller with certificate short-livedness and delegated credentials to decouple private-key exposure surface from rotation frequency, and we bind resumption tickets to client identity to prevent lateral replay under scaling. For LLM-specific threat models—prompt injection propagation across tools, multi-tenant retrieval caches, and cross-region shadow evaluation—we incorporate semantic anomaly scores and attestation ages as control inputs without entangling application payloads with the handshake layer.

Our analysis models handshake cost as an additive service time in an M/G/1 queue and proves that a properly tuned controller keeps the system below the knee of the latency–throughput curve while sustaining strong authentication during adversarial bursts. A reference implementation on a Kubernetes mesh shows that, under synthetic red-team traffic and realistic diurnal load, Adaptive-mTLS reduces P99 end-to-end latency by up to a factor of two relative to a static hardening baseline at equivalent cryptographic strength, while accelerating revocation convergence after compromise signals. We discuss safety conditions that forbid downgrades below a correctness floor and provide machine-checkable invariants for change validation. Finally, we map future directions, including model-aware risk co-training, formal synthesis of TLS parameter sets, and privacy-preserving control signals. The result is a principled, auditable framework that treats transport security not as a fixed tax but as a controllable resource aligned with the dynamics of LLM systems. By articulating the interface between risk estimation, control, and verifiable policy realization, we provide a reusable blueprint for adaptive confidentiality and integrity in safety-critical foundation model deployments.

I. Introduction

Large Language Models are deployed as distributed systems that fuse network gateways, retrieval layers, feature stores, policy engines, model workers, evaluation harnesses, and audit sinks. The security perimeter is therefore defined not by an ingress point but by a time-varying graph of east–west channels. Mutual TLS remains the primary mechanism to attach strong identities to these channels while encrypting payloads with keys negotiated for forward secrecy. In practice, platform teams hard-code suites, certificate lifetimes, and revocation strategies, and they scatter overrides to satisfy conflicting performance and compliance demands. The result is a brittle compromise: either the system absorbs excessive latency during peak load or it veers toward unsafe downgrades to meet service objectives.

This paper advances the thesis that transport posture should be closed-loop and adaptive. The LLM domain supplies the necessary signals: inference traffic reveals diurnal patterns, request semantics expose adversarial drift, autoscaling events reshape topology, and attestation systems surface the freshness of workload identity. Rather than treating mTLS as an immutable configuration, we treat it as a control target with safety constraints. The controller observes risk and SLO error, then tunes re-authentication frequency, session resumption binding, certificate agility, and revocation propagation timeliness, while respecting inviolable invariants such as minimum key strength, mandatory mutual authentication, and non-downgradable protocol versions.

II. Current Landscape of Research

The modern position of mutual authentication in distributed systems is anchored by TLS 1.3 as a cryptographic substrate and by zero-trust as an operational paradigm. TLS 1.3 consolidated forward secrecy by construction, shortened handshakes, and removed legacy ciphers and renegotiation paths that complicated earlier versions; these protocol-level choices

created headroom to consider transport posture as a tunable mechanism rather than a brittle constant. In parallel, zero-trust architecture[2] reframed enterprise trust from perimeter-centric controls to continuous verification of identity and context across all flows. Within this frame, mutual TLS serves as the default mechanism for binding strong, short-lived identities to east–west channels[3], but the prevailing practice remains largely static: platform teams select a fixed cipher policy, certificate lifetimes, and revocation approach and then optimize the rest of the stack around those choices, even when workload dynamics and threat posture vary markedly over time.

Identity issuance and automation have matured to support high churn and ephemeral workloads[4]. The IETF profile for Internet X.509 certificates supplies the semantic substrate for name constraints, key usage, and path validation, while the ACME protocol demonstrated that certificate lifecycle can be safely automated at Internet scale. In cloud-native settings, SPIFFE and its runtime SPIRE generalize these ideas from host or service identities to workload-level identities, issuing short-lived SVIDs as X.509 credentials or JWTs with a well-defined trust-domain model and supporting federation across administrative boundaries. These identity systems integrate with data planes via Envoy's Secret Discovery Service, which allows hot rotation of private keys and trust bundles without container restarts, eliminating a class of operational hazards associated with file-mounted secrets and stale trust stores.

Service meshes operationalized mTLS at scale by interposing sidecars or ambient data-plane components between applications and the network layer. Istio exposes policy primitives such as PeerAuthentication to require or permissively negotiate mTLS, automatically upgrading eligible paths and coordinating certificate rotation through its control plane; Linkerd takes a posture of mandatory, transparent mTLS for meshed pods, minimizing configuration drift while relying on automated key and trust-bundle rotation in the control plane. Recent "ambient" approaches reduce sidecar overhead by shifting L4 enforcement into node-level tunnels while preserving workload identity and mTLS semantics, a direction that suggests transport policy may be expressed and enforced at multiple layers without undermining cryptographic provenance. These ecosystems have made mutual authentication a baseline rather than an exception, but their control knobs are typically fixed at deploy time and adjusted only through manual rollouts.

At the protocol frontier, several mechanisms enrich the design space for adaptive transport without diluting its cryptographic guarantees[5]. Delegated Credentials enable servers to use short-duration signing keys derived under an origin's long-term certificate, shrinking the exposure window of private keys and aligning well with frequent rotation or compartmentalized termination[6]. Exported Authenticators introduce post-handshake authentication that can prove additional identities on an existing TLS channel and, when composed with HTTP/2 or HTTP/3 extensions, can bind multiple origins or re-validate properties without a full handshake. QUIC's marriage to TLS 1.3 preserves these security properties while offering transport layer benefits for latency[7] and mobility; Emerging work on extended key update aims to refresh keys on long-lived connections with forward secrecy without incurring the cost of re-handshake, which is relevant for sustained LLM inference or streaming evaluation sessions.

The best-current-practice guidance now treats the TLS configuration as a living artifact subject to evolving attacks and operational constraints[8]. The community's consolidated recommendations for cipher selection, version negotiation, resumption safety, ticket lifetime, and 0-RTT handling provide a floor below which secure deployments must not fall. In practice, however, these recommendations are instantiated as static profiles that change only when auditors or incident postmortems compel action. The research gap lies in principled, feedback-driven selection of these parameters at runtime[7], subject to invariants that preserve mutual authentication, forward secrecy, and downgrade resistance. The requisite pieces exist—short-lived credentials, dynamic secret distribution, post-handshake authentication, rapid revocation signaling—but they have rarely been composed under a formal control objective.

The LLM serving domain adds fresh pressure to this otherwise mature transport stack. Workloads exhibit diurnal and bursty demand[7] with autoscaling that reshapes topologies, while multi-tenant retrieval, tool execution, and cross-region evaluation introduce novel lateral-movement pathways. Security guidance specific to LLM applications has crystallized around risks such as prompt injection, insecure output handling, and data exfiltration[8], which are usually discussed at the application layer; yet these risks also have transport-relevant facets, including a need to bind identity and policy across chains of tools and to react to sudden rises in adversarial input. Contemporary systems typically feed these semantics into authorization decisions while leaving the transport posture unchanged[9], resulting in a mismatch between observed risk and handshake intensity, rotation cadence, and resumption policy. Bridging this gap, withoutt entangling payload semantics with cryptographicmachinery, requireses an architecture in which workload identity, attestation freshness, and application-level risk summaries become first-class control inputs that shaphow and how often mutual authentication is performed.

In summary, mTLS for microservices is solved in the narrow sense of 'can be made to work on a scale' but is not specified in the broader sense of "can be tuned as a controllable resource in response to changing conditions'. The literature provides robust cryptographic building blocks[10], mature identity frameworks, and mesh-integrated automation; it also offers a clear articulation of zero-trust principles for continual verification. What remains open - and what this paper addresses - is a research program that treats transport posture as a feedback-controlled process, supplies machine-checkable safety invariants for all adaptations, and integrates with mesh-era identity[11] and distribution mechanisms to deliver both stronger security and better latency under realistic LLM workloads.

III. Proposed Solution

Adaptive-mTLS introduces a control plane that consumes telemetry and emits concrete transport adaptations. Each workload exposes counters for handshake latency[12], failure modes, and resumption hit rates; the application plane contributes semantic anomaly scores that summarize the adversarial character of recent prompts and tool calls; the identity plane contributes attestation age and key-material provenance. These inputs are normalized to a risk score $R_t \in [0,1]$. We define an SLO error $E_t = \ell_t - \ell^\star$ where ℓ_t is a rolling tail-latency estimate and $\ell^\star$ is the declared bound.

The controller computes an actuation signal u_t representing handshake intensity and credential agility:

$$u_t = \mathrm{clip}\big(K_p E_t + K_i \sum_{k=0}^{t} \gamma^{t-k} E_k + K_r \phi(R_t),\, u_{\min},\, u_{\max}\big), \tag{1}$$

where K_p, K_i, K_r are gains, $\gamma \in (0,1)$ discounts history, and ϕ maps risk to a calibrated margin. The policy compiler interprets u_t into transport parameters: it selects between ECDSA and EdDSA subject to hardware support, sets certificate time-to-live, toggles ticket binding for TLS 1.3 resumption, configures early-data rejection, and adjusts the frequency of proof-of-possession checks on client identities. To avoid semantic coupling, the compiler's interface restricts itself to transport primitives and X.509 constraints, while the underlying application only supplies summary signals.

Safety is ensured by a set of machine-checkable invariants. Protocol version is never degraded; cipher suites remain within an allow-set that meets regulatory and platform baselines; client authentication is mandatory on east–west channels; and resumption tickets are bound to the presenting identity and scope-limited to specific services. The compiler evaluates proposed changes as a finite set of diffs, attaches rationale and predicted impact, and submits them to a canary mechanism that verifies invariants and monitors regressions before full rollout.

IV. Analysis

We analyze performance by modeling the handshake cost as an additive service time $H(u_t)$ that increases with actuation strength and is sub-linear under resumption. Let the base service time for inference be S, with first and second moments $\mathbb{E}[S]$ and $\mathbb{E}[S^2]$. The effective service time becomes $S' = S + H(u_t)$, and under an M/G/1 approximation the mean waiting time is

$$W = \frac{\lambda\, \mathbb{E}\big[(S')^2\big]}{2\,(1 - \lambda \mathbb{E}[S'])}, \tag{2}$$

with load $\rho = \lambda \mathbb{E}[S'] < 1$. The controller indirectly acts on W by trading H against risk. Stability follows from standard results for discrete-time PI control with saturation provided the combined loop gain obeys

$$0 < K_p \partial_\ell H + K_r \partial_R H < \kappa, \tag{3}$$

for a platform-dependent κ determined by the slope of the latency curve near the operating point. We prove that if the invariants prevent protocol downgrades and enforce a lower bound on key strength, then no feasible actuation can violate mutual authentication or forward secrecy, and the only failure mode under mis-tuning is performance regression, which is contained by canary guards and rollback.

Security is examined under active adversaries that attempt lateral movement by ticket replay, downgrade negotiation, and time-of-check/time-of-use exploitation of stale attestations. Binding resumption tickets to client identity and scope eliminates generic replay; refusing early data when R_t spikes blocks adversarial leverage of 0-RTT; and short-lived delegated credentials reduce private-key exposure while enabling aggressive rotation when the controller detects compromise signals. Because the controller increases handshake intensity under elevated R_t, the system responds to red-team bursts by converging toward maximum posture without human intervention.

We implemented a prototype in a Kubernetes mesh with sidecars terminating TLS and with an LLM gateway mediating requests to model workers and retrieval backends. The controller runs as a highly available service receiving telemetry from OpenTelemetry exporters and identity state from the attestation service. In controlled experiments with synthetic diurnal load and injected adversarial prompts, the system sustained SLOs at the ninety-ninth percentile while avoiding unsafe downgrades and converging revocation in fewer propagation intervals than a static baseline. Although absolute numbers vary with hardware and cryptographic offload, the qualitative behavior matches the analysis: performance tracks the knee of the latency curve while security posture ramps under risk.

V. Future Direction

Future work should enrich the risk estimator with model-aware signals while preserving the separation between payload semantics and transport. A promising path is co-training the estimator with the content-moderation or red-team classifier already deployed for safety, exporting only differentially private aggregates as control inputs. Formal synthesis of TLS parameter sets from high-level invariants would reduce compiler complexity and enable automated proof artifacts, bringing transport posture into a design-by-contract discipline. Extending attestation freshness into the controller through verifiable time and secure enclaves would allow actuation on measured drift in code identity, turning binary rollouts into first-class signals for transport adaptation. Cross-mesh coordination across regions can be made topology-aware by sharing compressed risk summaries that guide where to place handshake load under failover. Finally, a rigorous study of fairness in adaptive security is needed to ensure that tenants with atypical workloads do not experience systematic performance penalties from conservative control, motivating per-tenant adaptive baselines subject to global safety invariants.

VI. Conclusion

LLM systems are dynamic in traffic, topology, and risk, yet their transport security is usually configured as a static compromise. By elevating mTLS from configuration to control, Adaptive-mTLS preserves the cryptographic orthodoxy of TLS while aligning operational posture with observed conditions. The feedback architecture combines risk estimation, stability-aware control, and verifiable policy compilation to maintain strong mutual authentication and confidentiality without surrendering tail-latency objectives during peak inference. The analysis demonstrates how handshake cost integrates into queuing behavior and how safety invariants bound the space of admissible adaptations. The prototype validates that a practical system can react to adversarial bursts and diurnal shifts without manual intervention, converging toward strong posture when it matters most and recovering performance when risk subsides. Treating security as a controllable resource rather than a static tax offers a path to transport layers that are both safer and faster, and it reframes the operational problem of securing LLM pipelines as one of principled, auditable feedback.

References

[1] D. Rho, T. Kim, M. Park, J. W. Kim, H. Chae, E. K. Ryu, and J. H. Cheon, "Encryption-friendly llm architecture," *arXiv preprint arXiv:2410.02486*, 2024.

[2] B. B. Gupta, A. Gaurav, V. Arya, W. Alhalabi, D. Alsalman, and P. Vijayakumar, "Enhancing user prompt confidentiality in large language models through advanced differential encryption," *Computers and Electrical Engineering*, vol. 116, p. 109215, 2024.

[3] P. Xiao, S. Yang, H. Wang, Z. Zhang, and C. Zou, "Privacy-preserving revocable access control for llm-driven electrical distributed systems," *Peer-to-Peer Networking and Applications*, vol. 18, no. 3, p. 148, 2025.

[4] Y. Bae, M. Kim, J. Lee, S. Kim, J. Kim, Y. Choi, and N. Mireshghallah, "Privacy-preserving llm interaction with socratic chain-of-thought reasoning and homomorphically encrypted vector databases," *arXiv preprint arXiv:2506.17336*, 2025.

[5] A. Knapp, M. Linga, and S. Yen, "Scaling policy assignment in containerized environment."

[6] W. Huang, Y. Wang, A. Cheng, A. Zhou, C. Yu, and L. Wang, "A fast, performant, secure distributed training framework for llm," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 4800–4804.

[7] S. Ashe and H. Ramachandra, "The effect of continuous encryption of data in cloud native architecture," in *2024 IEEE Cloud Summit*. IEEE, 2024, pp. 163–169.

[8] I. Zimerman, A. Adir, E. Aharoni, M. Avitan, M. Baruch, N. Drucker, J. Lerner, R. Masalha, R. Meiri, and O. Soceanu, "Power-softmax: Towards secure llm inference over encrypted data," *arXiv preprint arXiv:2410.09457*, 2024.

[9] F. Alder, N. Asokan, A. Kurnikov, A. Paverd, and M. Steiner, "S-faas: Trustworthy and accountable function-as-a-service using intel sgx," in *Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop*, 2019, pp. 185–199.

[10] K. A. Bailey and S. W. Smith, "Trusted virtual containers on demand," in *Proceedings of the fifth ACM workshop on Scalable trusted computing*, 2010, pp. 63–72.

[11] K. Bhargavan, I. Boureanu, P.-A. Fouque, C. Onete, and B. Richard, "Content delivery over tls: a cryptographic analysis of keyless ssl," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2017, pp. 1–16.

[12] G. Gavrilov and V. Trajkovik, "Security and privacy issues and requirements for healthcare cloud computing," *ICT Innovations*, 2012.