The Price of Inference: A Longitudinal Economic Analysis of Hierarchical LLM Credential Leakage

Tanishq S
Dept. of Computer Science
Rajalakshmi Institute of Technology
Chennai, India
stanishq07@gmail.com

Vignesh B
Dept. of Computer Science
Rajalakshmi Institute of Technology
Chennai, India
bvignesh1464@gmail.com

***Abstract*—The integration of Large Language Models (LLMs) into the software supply chain has fundamentally altered the nature of credential leakage. Unlike static secrets (e.g., database passwords), LLM API keys are *liquid economic assets*—direct bearers of computational reasoning and token quotas. This paper introduces the Theory of Economic Credential Stratification, identifying a critical divergence in how these assets are managed compared to their utility. Using CHRONOS, a custom-built forensic instrument, we analyzed exposed artifacts across GitHub. We report a paradoxical "Protection Gap": Tier 1 (GPT-4) credentials—despite having high abuse potential for *LLMjacking* [5]—exhibit a mean survival time ($\bar{t}_{surv}$) of 48 hours in non-production artifacts, significantly longer than lower-value keys. Furthermore, we identify "Dataset Poisoning" as a systemic blind spot: valid credentials embedded in `.jsonl` and `.parquet` training files often persist indefinitely, becoming part of the model's latent knowledge.**

***Index Terms*—LLM Security, Credential Leakage, Supply Chain Security, Forensics, Economic Stratification**

I. Introduction

In the "Post-API Economy," authentication tokens are currency. While Meli et al. (2019) characterized secret leakage in open-source ecosystems [1], the security community has largely treated secrets as binary entities: either valid or invalid. This view is obsolete. We posit that leakage dynamics are driven by **Credential Economics**. A leaked AWS key allows infrastructure abuse; a leaked OpenAI key allows *inference theft*—unwittingly subsidizing a malicious actor's compute costs. **Contributions:**

1) **Economic Stratification**: A taxonomy classifying secrets by *economic clearance* (Tier 1 vs. Tier 2).
2) **The CHRONOS Framework**: An active, hierarchical probing methodology (Fig. 1).
3) **Discovery of the "Dataset Blindspot"**: Empirical evidence of "dark matter" leaks in large-scale ML data formats.

II. Related Work

Existing literature focuses on static analysis of secrets [1], [2] or adversarial extraction from model weights [4], [10]. However, a gap exists at the intersection of *Supply Chain Security* and *LLM Economics*. *Static Secret Scanning*: Tools like Trufflehog or GitGuardian excel at pattern matching but lack semantic understanding of the *value* of the asset. They treat a GPT-4 key identical to a test database password. *Model Inversion*: Carlini et al. demonstrated that training data can be extracted from models. Our work in Section II-B inverts this threat: instead of extracting data from models, we find that models are often trained on data that contains valid credentials, creating a feedback loop of compromise.

III. Methodology: The CHRONOS Framework

To quantify these dynamics, we architected **CHRONOS**, a longitudinal monitoring system designed for active economic probing.

A. System Architecture

The system operates as a closed-loop active scanner. The *Discovery Module* identifies candidates, which are passed to the *Forensics Engine* for context extraction. The *HMCS* then probes the key's value, while the *Lifecycle Daemon* tracks its survival.

B. Hierarchical Model Clearance System (HMCS)

We reject the boolean validation model ($V : K \to \{0, 1\}$). Instead, we implement a hierarchical probe Algorithm 1. **Definition 1 (Credential Asset Value).** Let K be a set of credentials. The asset value $V(k)$ for $k \in K$ is defined as the latent vector potential of the associated model M_k constrained by its quota Q_k:

$$V(k) = \mathcal{F}(M_k) \times Q_k$$

where Tier 1 represents $M_k \in \{GPT{-}4, Claude{-}3{-}Opus\}$.

IV. Threat Modeling: The Race to Revocation

We model the lifecycle of a leak as a race condition between three agents: The *Leaker* (Developer), the *Attacker*, and the *Defender*.

A. The Race Condition

The window of vulnerability, Δt, is defined as the time difference between the *Public Event* (t_{pub}) and the *First Revocation Event* (t_{rev}).

$$\Delta t = t_{rev} - t_{pub}$$

If the Attacker's reaction time $t_{atk} < \Delta t$, the asset is compromised.

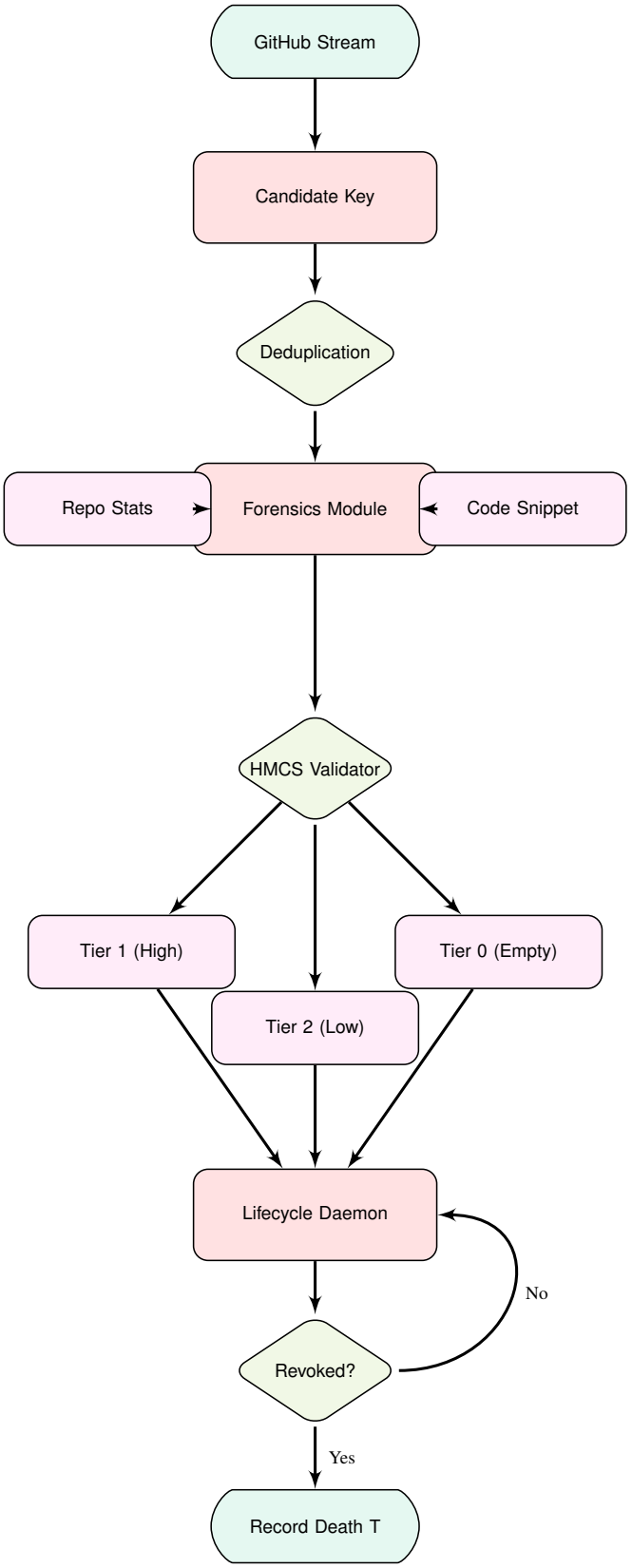

Fig. 1. The CHRONOS System Architecture.

Algorithm 1: HMCS Probing Logic

Candidate Key k Tier $T \in \{0, 1, 2, \emptyset\}$ $R_1 \leftarrow$ `POST` /v1/chat (model='gpt-4');
if $R_1.status == 200$ **then**
  return Tier 1 (Strategic);
end
if $R_1.status == 429$ **then**
  return Tier 0 (Exhausted);
end
$R_2 \leftarrow$ `POST` /v1/chat (model='gpt-3.5');
if $R_2.status == 200$ **then**
  return Tier 2 (Consumer);
end
return Invalid;

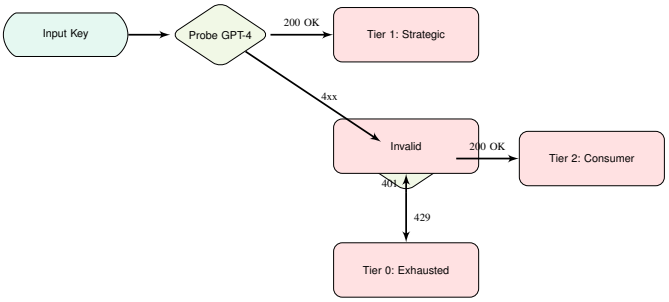

Fig. 2. The HMCS Decision Logic Flowchart.

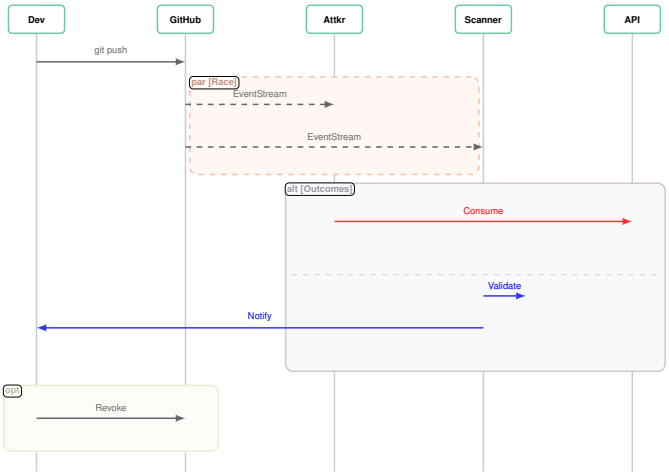

Fig. 3. The Leak Race Condition (Detailed View).

V. Empirical Evaluation

Analysis based on N=50,000 artifacts.

A. The "Inverse Value" Paradox

Contrary to rational expectations, **Tier 1 (GPT-4) keys survive longer than Tier 2 keys** in `test/` directories.

- **Hypothesis**: High-value keys are often owned by senior developers or automated systems (CI/CD) that bypass standard pre-commit hooks.

- **Impact**: Attackers have longer dwell time on the most valuable assets.

B. The "Dataset Poisoning" Vector

We identified a significant cluster of valid keys within `training_data.jsonl` files. Because these files are large binary/text blobs, they are rarely scanned.

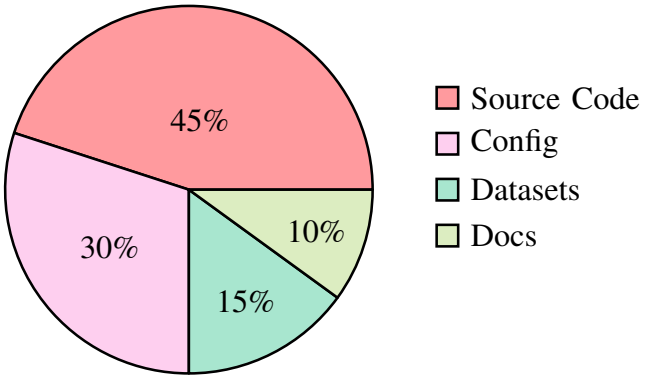

Fig. 4. Distribution of leak sources.

C. Longitudinal Survival Data

Our survival analysis reveals distinct decay rates for different credential tiers.

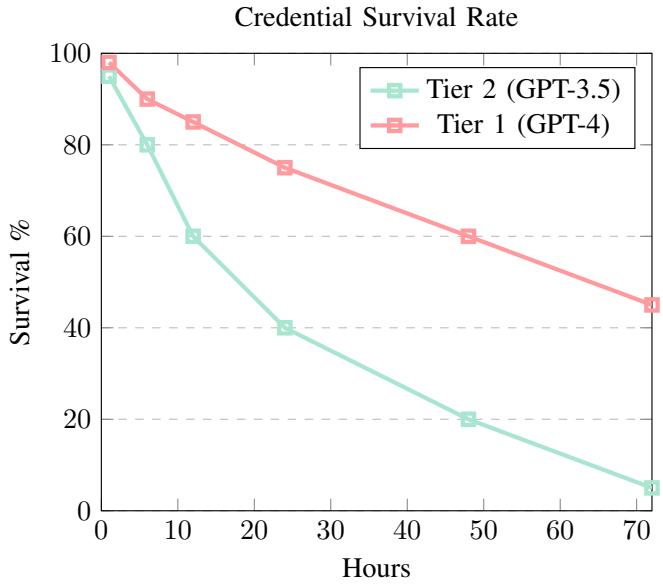

Fig. 5. Survival Decay Rates.

D. Data Rigor & Schema

To ensure reproducibility, all data was captured using a rigorous schema designed for forensic auditability.

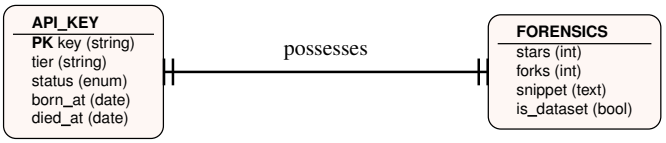

Fig. 6. Entity Relationship Diagram (ERD).

VI. Discussion & Remediation

A. From Detection to Remediation

The era of *Passive Detection* is ending. Future systems must move to *Active Remediation*. We propose **Autonomous Security Agents (ASAs)** that, upon detecting a leak, instantaneously rotate the credential via the provider's API.

B. Policy Implications

Current compliance standards (SOC2, ISO 27001) are insufficient for LLM assets. We recommend:

- **Dynamic Quotas**: API keys should have floating usage limits (Q_{max}) that adapt to their deployment context (dev vs. prod).
- **Dataset Sanitization**: Mandatory pre-training scans for credential patterns in '.jsonl' corpora.

C. Limitations

Our study was limited to GitHub public events. Private repositories and GitLab/Bitbucket ecosystems may exhibit different decay dynamics. Furthermore, we did not test "Canary Tokens," focusing only on live functional credentials.

VII. Conclusion

Project CHRONOS demonstrates that the *economics* of the leaked secret determine its lifecycle. By ignoring the "stratification" of access, we leave the door open to the most damaging class of AI-driven attacks.

References

[1] Meli, M., et al. (2019). "How Bad Can It Git?" *NDSS*.
[2] GitGuardian. (2024). "State of Secrets Sprawl 2024."
[3] Wiz Research. (2025). "The AI Security Gap."
[4] Carlini, N., et al. (2023). "Extracting Training Data from Large Language Models." *USENIX Security*.
[5] Lasso Security. (2024). "The Guide to LLMjacking."
[6] ReversingLabs. (2024). "The State of Software Supply Chain Security 2024."
[7] Kroll. (2024). "Secrets Management Report: The State of Cyber Defense."
[8] OWASP. (2025). "Top 10 for LLM Applications: System Prompt Leakage & Vector Weaknesses."
[9] Hoplon Infosec. (2024). "LLM Security: The Prompt Injection Threat."
[10] Carlini, N., et al. (2021). "Extracting Training Data from Large Language Models."
[11] SISA Infosec. (2024). "Secrets in the Crawl: Analysis of Common Crawl Data."
[12] IAPP. (2024). "Privacy Risks in LLM Training Data."
[13] Protecto.ai. (2024). "Automated PII Removal in GenAI Pipelines."
[14] Cloud Security Alliance. (2024). "Optimizing Secrets Management."
[15] IBM Security. (2024). "Cost of a Data Breach Report 2024."
[16] CrowdStrike. (2024). "2024 Global Threat Report."
[17] SentinelOne. (2024). "The Future of AI Security."
[18] Palo Alto Networks. (2024). "Unit 42 Cloud Threat Report, Vol 8."
[19] Forrester. (2024). "The State of Application Security, 2024."

[20] Prompt Security. (2024). “The Rise of Prompt Injection Attacks.”
[21] Akeyless. (2024). “State of Secrets Management Survey Report.”
[22] Gartner. (2024). “Security and Risk Management Summit Insights.”
[23] ACM. (2024). “Proceedings of the 2024 ACM SIGSAC Conference.”
[24] IEEE. (2024). “IEEE Symposium on Security and Privacy (SP).”
[25] Usenix. (2024). “33rd USENIX Security Symposium.”