Economic Confidentiality without Secrets: Making Intercepted LLM-Agent Communications Unusable

Bolaji Makinde*
Independent Researcher
Palo Alto, California
bola2014tel@gmail.com

Bryce Jackson†
Department of Computer Science
California State University, Bakersfield
Bakersfield, CA 93311
brycejjacks@gmail.com

Abstract

We study whether LLM agents can exchange useful information over a public channel *without* secrets (no keys, no TEEs) while making unauthorized decoding *economically impractical*. We formalize an "economic confidentiality" objective and design a stochastic "private language" L_{r} whose parameters-embedding dimension d, flavor multiplicity f, and dilution k-disperse recoverable signal S across long sequences with low per-token correlation. We analyze attackers that observe traffic and, in stronger scopes, possess agent weights, and we propose a back-of-the-envelope scaling law $n \approx \frac{kf}{S^2}\, d \log_2 d$ for learning the inverse mapping to the "original legible language" L_{u} represented as $L_{\mathrm{r}} \to L_{\mathrm{u}}$. This scaling law is accompanied by back-of-the-envelope budget calculations to illustrate parameter trade-offs. This paper is a theoretical exploration and analysis-only: no significant system implementation or empirical evaluation are reported. While weaker than cryptographic secrecy, our conceptual modeling quantifies regimes where inverting L_{r} exceeds realistic budgets-aligning with the goal of making unauthorized use *uneconomical*-and clarify utility–cost trade-offs in secret-less settings.

1 Introduction

The proliferation of powerful Large Language Models (LLMs) has unlocked new frontiers for autonomous multi-agent systems (MAS). We envision a future where LLM-powered agents, representing individuals or organizations, negotiate complex deals, discover novel opportunities, and optimize shared resources by reasoning over their collective private data. The potential utility of such systems is immense, promising unprecedented efficiency in domains ranging from supply chain management to personalized medicine.

This vision, however, is predicated on a foundational conflict between two competing objectives:

Utility: Agents must share detailed, semantically rich information to identify and act upon high-value opportunities. The quality of outcomes is directly proportional to the quality and completeness of the data shared.

Privacy: The data shared by agents is often proprietary or sensitive. Participants require strong guarantees that their private information will not be exposed to unauthorized parties. The level of privacy is inversely proportional to the quality and completeness of the data shared.

*Bolalji Makinde is an independent researcher based in Bay Area, California. Personal website: bolajimakinde.com. ORCID: 0009-0001-5594-4514

†Work conducted as an independent project during studies at California State University, Bakersfield. Personal website: brycejjacks.com

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Lock-LLM: The Workshop on Un-Usable, Un-Distillable, and Un-Finetunable Large Language Models.

In many emerging decentralized systems, we must assume a zero-trust or "transparent" environment. An adversary may not only intercept all communication between agents but may also have complete access to the agents' software, architecture, and even the trained model weights. This scenario renders traditional access control measures ineffective and presents a formidable challenge to privacy.

The standard response to this challenge involves well-established cryptographic protocols (e.g., end-to-end encryption) or specialized hardware (e.g., Trusted Execution Environments, or TEEs). While effective, it is a valuable scientific endeavor to understand the inherent security properties of the communication channel itself. What level of privacy can be achieved if we explicitly forbid the use of secret keys or secure enclaves? This question motivates the principle of Economic Confidentiality: can we "lock" the semantic content of a conversation from an all-seeing adversary using only the structure of the language itself?

This paper argues that while perfect, information-theoretic security is impossible under these constraints, a meaningful level of economic security is achievable. We propose a framework where agents communicate in a specially designed private language (L_{r}). This language is not a simple cipher but a complex, high-dimensional representation engineered to be computationally expensive to reverse-engineer. Our core contribution is a formal model that quantifies the attacker's inversion cost as a function of the L_{r}'s statistical properties. This model allows a system designer to tune parameters to achieve a target level of economic impracticality-for instance, plausibly pushing decoding costs into the hundreds of millions of dollars with multi-year horizons under 2025 cloud prices (see App. B for a concrete scenario). We explore the inherent trade-off, where the same parameters that inflate attacker cost also impose a penalty on the training, inference, and performance of legitimate agents, thereby providing a principled guide to navigating the privacy-utility frontier in a world without secrets.

Contributions. (i) We formalize *Economic Confidentiality under open observation* for LLM agents. (ii) We propose a tunable $L_{\mathrm{u}} \to L_{\mathrm{r}}$ design with explicit knobs (d, f, k, S) to trade utility for attacker cost. (iii) We derive a back-of-the-envelope scaling law $n \approx \frac{kf}{S^2}\, d \log_2 d$ for learning $L_{\mathrm{r}} \to L_{\mathrm{u}}$ and provide illustrative calculations; we provide targeted synthetic validation for the scaling law's core components.

Scope and non-claims. This paper presents a conceptual framework, heuristic derivations, and order-of-magnitude calculations only. The guiding contribution is a formal thought experiment. We do not implement a full L_{r} system or run large scale experiments. Our empirical support is confined to the targeted synthetic experiments detailed in Appendix D.

2 Related Work

Our setting intersects privacy-preserving ML, information-theoretic limits without secrets, emergent communication, watermarking/provenance, and geometry-based alignment attacks. We *deliberately* exclude keys/TEEs and ask what economic barriers remain in plain-view communications. Unlike DP/HE/SMPC (which assume secrets/trust), and unlike capability "locking" that targets parameters, we target *intercepted traffic* by making it economically unusable.

Standard privacy-preserving techniques. Homomorphic Encryption and Secure Multi-Party Computation allow computation on encrypted data but often introduce significant overhead for LLM-scale workloads (1). Differential Privacy adds calibrated noise yet trades off utility (2). Federated Learning does not protect plaintext traffic from an eavesdropper nor models when weights are public (3). TEEs are excluded by design in our setting.

Unsupervised alignment as the primary attacker. A key attacker class uses unsupervised alignment/bilingual induction to align latent spaces without parallel data (4; 5). Robust self-learning mapping methods (e.g., VecMap) broaden this threat (6), while known failure modes and negative results delineate regimes where alignment breaks down (7). Our design aims to force operation in these harder regimes via (d, f, k, S).

Model inversion, stealing, and membership inference. Beyond traffic alignment, the threat model includes extraction via model inversion, membership inference, and stealing from APIs

Table 1: Symbols used throughout.

Symbol	Meaning
L_{u}	Legible Utility language (human-readable)
L_{r}	Private Language (obfuscated)
d	Embedding dimension of L_{r} (vocabulary v scales with d)
f	Flavor multiplicity (variants per L_{u} concept)
k	Noise-to-signal token ratio in L_{r}
S	Effective signal bits/message ($\approx \frac{1}{2}\sum_i \rho_i^2$)
ρ_i	Correlation of token i with the L_{u} secret
n	Attacker's required sample complexity (# of $(L_{\mathrm{u}}, L_{\mathrm{r}})$ pairs)
L	Total message length (tokens)
s	Count of signal tokens in a message
c	Count of carrying tokens in a message
τ	Tokens per training sample
E	Decoder training epochs
C_{gen}	Total cost for attacker to generate n pairs
C_{train}	Total cost for attacker to train decoder
C_{pair}	Dollar cost to mint one $(L_{\mathrm{u}}, L_{\mathrm{r}})$ training pair
C_{mh}	Memory-hard gate cost per query (e.g., Argon2)
c_{FLOP}	Dollar cost per FLOP for generation/forward passes
c_{train}	Dollar cost per training FLOP
F_{fwd}	FLOPs for one encoder forward pass
R	System rate limit (messages/sec)
α	Attacker parallelism (number of GPUs)

(8; 9; 10; 11; 12). This motivates defenses that do not rely on hiding code or weights and that remain costly to decode even when generators are known.

Watermarking, provenance, and unlearnable data. Watermarking seeks to make *generated text* detectable (13; 14); this is orthogonal but complementary to our traffic-hardening approach. On the data side, unlearnable examples aim to make models trained on certain content unusable (15); again complementary, as we focus on rendering intercepted *communications* uneconomical to decode.

Emergent communication. Multi-agent work studies protocols that emerge to coordinate or hide information (16; 17). We differ by explicitly quantifying attacker cost under open observation and by specifying tunable knobs (d, f, k, S) that target unsupervised alignment and data-minting attackers.

Information theory and obfuscation. Classical results preclude perfect secrecy without keys (18); general program obfuscation is also impossible in broad settings (19). We therefore target an *economic* notion of confidentiality rather than cryptographic hardness.

3 The Economic Confidentiality Framework

We propose a multi-agent system where confidentiality is derived from the economic cost of inverting an obfuscated communication channel.

3.1 System Architecture

The system consists of two or more agents. Each agent possesses a local dataset of private information, which we term *Legible Utility data* (L_{u}). This data is in a human-readable format (e.g., plain English). The agents are equipped with LLMs that have been fine-tuned on their local L_{u}. To negotiate or collaborate, agents communicate over a public, untrusted channel. An adversary is assumed to have full access to this channel, the agents' source code, and the complete weights of their models.

3.2 The Private Language (L_{r})

Instead of communicating in a human language, agents convert their intentions into a high-dimensional, obfuscated representation called the *Private Language* (L_{r}). The transformation $L_{\mathrm{u}} \rightarrow L_{\mathrm{r}}$ is designed to be computationally efficient, while the inverse $L_{\mathrm{r}} \rightarrow L_{\mathrm{u}}$ is designed to be prohibitively expensive for an adversary. The properties of L_{r} are key to this asymmetry:

- **High Dimensionality (d):** L_{r} exists in a very large embedding space (e.g., $d > 16,000$). This increases the computational complexity of any learning task in this space.
- **Stochasticity and Flavor Multiplicity (f):** The mapping from L_{u} to L_{r} is not deterministic. A single semantic concept in L_{u} can be represented by many distinct surface forms in L_{r}. This forces an attacker to collect more samples to learn the underlying distribution, increasing data generation costs.
- **Redundancy and Dilution (L, s, k, c):** A small amount of signal from L_{u} is distributed across a very long L_{r} message of length L. The message contains three main components:
  - *signal tokens* (s) that encode the private L_{u}
  - *filler/noise tokens* (k) that contribute no information
  - *carrying tokens* (c)-task-necessary tokens that allow the agent to act on behalf of a user (e.g., framing a negotiation, specifying constraints, or structuring requests). While carrying words may contain a small amount of leakage about L_{u}, when carefully designed to be secret-agnostic (e.g., "find the best option within budget"), they primarily function like noise from the attacker's perspective. Together, filler and carrying words lower the effective signal-to-noise ratio (SNR), forcing an adversary to parse longer sequences and disentangle relevant bits from a large mass of weakly informative content.
- **Low Per-Token Correlation (ρ_i):** The mutual information between any single L_{r} token and the original L_{u} secret is minimized. The true signal is encoded in a complex, distributed pattern across many tokens, preventing simple statistical attacks.
- **Context-Dependency:** The meaning of an L_{r} sequence is highly dependent on the history of the conversation. This thwarts attempts to analyze messages in isolation and forces an attacker's model to handle long-range dependencies, further increasing its complexity.

3.3 Adversary scopes and assumptions

We analyze three scopes: **(C) Channel-only:** the attacker eavesdrops L_{r} traffic but lacks model weights/code. **(S) Single-agent observer:** weights/code of one agent are known. **(D) Dual-agent observer:** weights/code of all communicating agents are known.

In (S,D) the attacker can mint unlimited $(L_{\mathrm{u}}, L_{\mathrm{r}})$ pairs offline by running the (known) generator; API-side metering (e.g., proof-of-work, rate limits) does not bind chosen-plaintext volume and is excluded from guarantees. We therefore seek defenses whose cost inflation arises from (d, f, k, S) alone. In (C), API metering can raise C_{gen}; we separate these terms below.

3.4 A Plausible $L_{\mathrm{u}} \rightarrow L_{\mathrm{r}}$ Mechanism

We instantiate a stochastic encoder–generator $p_\theta(L_{\mathrm{r}} \mid L_{\mathrm{u}})$ that maximizes utility for counterpart agents while minimizing per-token leakage.

Architecture. Let $z = g_\phi(L_{\mathrm{u}}) \in \mathbb{R}^d$ be a high-dimensional latent. A generator h_θ produces tokens $x_{1:L}$ from z with *flavor multiplicity* by sampling a style code $s \sim \mathrm{Cat}(f)$. Thus, each underlying semantic intent admits many surface forms.

Training objectives. We combine (i) a contrastive utility loss that aligns the counterpart's latent needs with z (InfoNCE-style) and (ii) a leakage penalty that discourages per-token dependence on L_{u}:

$$\mathcal{L} \;=\; \mathcal{L}_{\text{utility}} \;+\; \lambda \sum_{i=1}^{L} \widehat{I}(x_i; L_{\mathrm{u}}) \;+\; \beta\, \mathcal{H}(s) \;+\; \gamma\, \mathcal{L}_{\text{redundancy}},$$

where $\widehat{I}$ is a variational MI estimator, $\mathcal{H}(s)$ encourages many equally probable "flavors", and $\mathcal{L}_{\text{redundancy}}$ promotes long, semantically equivalent paraphrases (padding) without increasing signal. This directly implements small per-token correlations ρ_i, large k, large f, and high d.

Throughput gating (chosen-plaintext control). To bound attack-side corpus minting, each generation call is wrapped in a memory-hard proof-of-work using Argon2id (RFC 9106) calibrated to a target wall-clock/compute budget per message; the server verifies the proof efficiently (20). This gate is orthogonal to utility and only meters request volume. In shared/cloud settings, very large memory targets (e.g., tens of GB) may be impractical; we therefore treat the PoW as *metering*, not as a core security assumption.

4 Modeling the Attacker's Inversion Cost

To move from qualitative obfuscation to quantitative security, we introduce a formal model to estimate the attacker's budget. The total cost is the sum of data generation and decoder training.

$$\text{Budget}_{\text{Attacker}} = C_{\text{gen}} + C_{\text{train}}$$

4.1 Variables of the Model

Our model is parameterized by the properties of the L_{r} language and the attacker's resources found in Table 1.

On the definition of S. We proxy the recoverable information about L_{u} in one L_{r} message by (small-correlation) mutual information between a low-dimensional L_{u} code and tokens: for jointly-Gaussian surrogates, $I \approx \frac{1}{2}\sum_i \rho_i^2$ (using $I = -\frac{1}{2}\sum_i \log(1-\rho_i^2)$ and $\log(1-x) \approx -x$). We thus take $S \approx \frac{1}{2}\sum_i \rho_i^2$ and design L_{r} to keep each ρ_i small and dispersed.

4.2 Estimating Required Samples (n)

Our goal is to estimate n, the **sample complexity** of the attacker's task, defined as the number of (Lu, Lr) training pairs required to train a successful decoder. We build our heuristic by starting with a structure common in statistical learning theory. In PAC/VC settings, the sample complexity n for a learner to achieve an excess error ε typically scales with the model's capacity and inversely with the squared error, i.e., $n = \tilde{\Theta}(capacity/\varepsilon^2)$.

We adapt this foundation to our setting. First, the attacker's ability to learn is fundamentally limited by the per-sample signal S, making the signal power S^2 the analogue for the squared error term. This gives us the foundational $1/S^2$ dependence, a core principle of signal detection(25). Second, the decoder's capacity must scale with the embedding dimension d. Combining these gives a baseline of $n \propto d/S^2$.

We then extend this baseline to incorporate the unique obfuscating features of our private language. We model the required samples as scaling linearly with flavor multiplicity f and noise dilution k, as doubling either of these factors intuitively forces an attacker to collect twice the data to disentangle the signal.

Finally, we refine the simple linear dependence on d. For high-dimensional problems, a more accurate model must account for the "curse of dimensionality," where searching for signal in a vast, sparse space incurs an additional cost. We therefore include a logarithmic factor, $log_2 d$, a term standard in high-dimensional statistics that reflects this search complexity (24) and is analogous to results in sparse recovery (23).

Combining these factors—the baseline statistical complexity, our problem-specific parameters, and the high-dimensional penalty—yields our final heuristic:

$$\boxed{n \;\approx\; \frac{kf}{S^2}\, d \log_2 d} \tag{1}$$

4.3 Total Cost Equation

The cost to generate n pairs is $C_{\text{gen}} = n \cdot (F_{\text{fwd}} \cdot c_{\text{FLOP}} + C_{\text{mh}})$. The time required is $T_{\text{gen}} = n/(R \cdot \alpha)$. The cost to train a decoder of comparable complexity is $C_{\text{train}} = E \cdot c_{\text{train}} \cdot (\tau \cdot n) \cdot d$, where τ is the average tokens per sample. This leads to the final cost model:

$$\text{Budget} \approx \left(\frac{k \cdot f}{S^2} \cdot d \log_2 d \right) \cdot ((F_{\text{fwd}} \cdot c_{\text{FLOP}} + C_{\text{mh}}) + (E \cdot c_{\text{train}} \cdot \tau \cdot d)) \tag{2}$$

5 Discussion and Limitations

The following are the four main limitations. For more, see Appendix F.

Core Engineering Assumption. A central assumption of this work is the feasibility of engineering a $L_{\text{u}} \to L_{\text{r}}$ mapping that simultaneously achieves the desired statistical properties (low per-token correlation ρ_i, high flavor multiplicity f) while retaining utility for the intended agents. The training objectives proposed in Section 3.4 are plausible but untested. Validating whether modern generative models can be fine-tuned to satisfy these competing constraints is a significant undertaking and a primary direction for future empirical work.

Polynomial ceiling. Our techniques raise attacker cost *polynomially* in $(d, k, f, 1/S)$; there is no cryptographic hardness jump. Economic guarantees are contingent on hardware prices and algorithms; parameters must be retuned over time.

Economic vs. cryptographic security. This paper deliberately explores a weaker, economic notion of security. Unlike cryptographic security grounded in hardness assumptions, our guarantees track hardware and energy costs and therefore erode over time (Koomey-style efficiency trends) (21; 22). Systems must be periodically retuned.

The High Cost of Utility. As demonstrated in our scenario, achieving a high degree of economic security comes at a steep price in terms of performance. The latency, bandwidth, and computational requirements for a system designed to thwart a billion-dollar adversary would likely be unacceptable for most real-time applications. This suggests that the Economic Confidentiality approach is best suited for high-value, low-frequency asynchronous negotiations where the value of the private data is exceptionally high. Potential applications could include M&A negotiations, critical infrastructure bidding, or sensitive intelligence sharing (see Appendix E for a detailed medical data scenario).

6 Conclusion

In this paper, we addressed the critical tension between utility and privacy in transparent multi-agent systems. We introduced the Economic Confidentiality framework, a novel approach to confidentiality that forgoes traditional cryptography and instead relies on making the inversion of an obfuscated communication channel economically impractical. We proposed a private language (L_{r}) characterized by high dimensionality, stochasticity, and redundancy. Our primary contribution is a formal, quantitative model that connects these language parameters to the financial and computational budget required for an adversary to break the channel's confidentiality.

We have shown that it is theoretically possible to design a system where decoding a conversation would require a budget of hundreds of millions of dollars and a multi-year effort, thus rendering the attack irrational for most threat models. However, we also demonstrated that this security comes at a significant cost to system performance. While not a substitute for cryptography, the Economic Confidentiality framework provides a principled and quantitative method for understanding the limits of security through obfuscation and offers a new perspective on designing layered defenses in an increasingly transparent digital world.

Acknowledgments and Disclosure of Funding

We would like to thank the anonymous reviewers for their insightful feedback and constructive comments, which have improved the quality of this paper.

References

[1] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2016). CryptoNets: Applying neural networks to encrypted data with high throughput and accuracy. In *ICML*.

[2] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. In *ACM CCS*.

[3] McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *AISTATS*.

[4] Conneau, A., Lample, G., Ranzato, M., Denoyer, L., & Jégou, H. (2017). Word translation without parallel data. *arXiv:1710.04087*.

[5] Lample, G., Conneau, A., Denoyer, L., & Ranzato, M. (2017). Unsupervised machine translation using monolingual corpora only. *arXiv:1711.00043*.

[6] Artetxe, M., Labaka, G., & Agirre, E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *ACL*. (arXiv:1805.06297)

[7] Søgaard, A., Ruder, S., & Vulić, I. (2018). On the limitations of unsupervised bilingual dictionary induction. In *ACL*. (arXiv:1805.03620)

[8] Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information. In *ACM CCS*.

[9] Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *IEEE S&P*.

[10] Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., & Ristenpart, T. (2016). Stealing machine learning models via prediction APIs. In *USENIX Security*.

[11] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., ... & Raffel, C. (2021). Extracting training data from large language models. In *USENIX Security*.

[12] Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramèr, F., & Zhang, C. (2023). Quantifying memorization across neural language models. In *ICLR*.

[13] Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., & Goldstein, T. (2023). A watermark for large language models. In *ICML*. (arXiv:2301.10226)

[14] Zhao, X., Ananth, P., Li, L., & Wang, Y.-X. (2024). Provable robust watermarking for AI-generated text. In *ICLR*.

[15] Huang, H., Ma, X., Erfani, S. M., Bailey, J., & Wang, Y. (2021). Unlearnable examples: Making personal data unexploitable. In *ICLR*. (arXiv:2101.04898)

[16] Lazaridou, A., Potapenko, A., & Tieleman, O. (2020). Multi-agent communication meets natural language. In *ACL*.

[17] Eccles, T., Bachrach, Y., Lever, G., Lazaridou, A., & Graepel, T. (2019). Biases for emergent communication in multi-agent reinforcement learning. In *NeurIPS*.

[18] Shannon, C. E. (1949). Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4), 656–715.

[19] Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., & Yang, K. (2001). On the (im)possibility of obfuscating programs. In *CRYPTO*.

[20] Biryukov, A., Dinu, D., & Khovratovich, D. (2021). Argon2 memory-hard function for password hashing and proof-of-work. RFC 9106, IETF.

[21] Koomey, J., Berard, S., Sanchez, M., & Wong, H. (2010). Implications of historical trends in the electrical efficiency of computing. *IEEE Annals of the History of Computing*, 33(3), 46–54.

[22] Koomey, J. (2010). Assessing trends in the electrical efficiency of computation. LLNL presentation slides.

[23] Donoho, D. L. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4), 1289–1306.

[24] Wainwright, M. J. (2019). *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press.

[25] Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Prentice Hall.

Appendix A: Additional derivations

We include algebraic steps omitted in §4, including the small-correlation surrogate $S \approx \frac{1}{2}\sum_i \rho_i^2$ and the conditions under which the $\log d$ factor appears.

Appendix B: Illustrative Scenario: Designing for a $1 Billion Barrier

To illustrate the qualitative impact and the scale of the trade-offs implied by our heuristic model, let's consider the parameters that would be required to target a one-billion-dollar inversion cost. Imagine an adversary with a one-year, one-billion-dollar budget. We assume 2025 cloud pricing.

- **Embedding Dimension (d):** Set to an extremely large 16,384. This makes F_{fwd} very high and quadratically increases memory and compute for both parties.
- **Signal (S):** Engineer the $L_{\text{u}} \to L_{\text{r}}$ mapping such that the average signal per message is extremely low, e.g., $S = 0.025$ bits, by distributing the information over thousands of tokens with very low individual correlation ($\rho_i \approx 0.05$).
- **Redundancy and Flavors (k, f):** Set $k = 20$ and $f = 30$, meaning each message is heavily padded and each concept has many variants.
- **Memory-Hard Gate (C_{mh}):** Require each message generation to solve a memory-hard puzzle (e.g., Argon2 with 32GB RAM), costing $1.50 in compute per call.
- **Rate Limit (R):** Enforce a strict rate limit of 5 messages per minute.

Plugging these into our model:

- $n \approx (20 \cdot 30/0.025^2) \cdot (16384 \cdot \log_2 16384) \approx 2.2 \times 10^8$ pairs.
- $C_{\text{pair}} \approx \$1.50$ (dominated by the memory-hard gate).
- $C_{\text{gen}} \approx 2.2 \times 10^8 \cdot \$1.50 \approx \$330$ Million.
- $T_{\text{gen}} \approx 2.2 \times 10^8/((5/60) \cdot 200 \text{ GPUs}) \approx 4.2$ years (exceeds the 1-year horizon).
- C_{train} would add tens of millions more.

The total cost and time horizon make the attack economically irrational. However, the trade-off is severe: every legitimate message now has a latency of several seconds and high computational cost, severely limiting the system's practical applications.

A Critical Caveat: The Temporal Decay of Economic Security. It is crucial to note that these figures represent a snapshot based on 2025 cloud pricing. Unlike traditional cryptographic security, which is grounded in computational hardness assumptions believed to be durable against foreseeable hardware improvements, the guarantees of economic confidentiality are explicitly tied to the current cost of computation. Following historical trends like Koomey's Law (21; 22), the cost of computation per unit of energy has steadily decreased. This implies that the billion-dollar barrier calculated here will erode over time. Consequently, the system's parameters (such as d, f, or k) would need to be periodically retuned and increased to maintain the same level of economic security against future, cheaper hardware.

Appendix C: Other attack vectors and mitigations

The primary cost model in Section 4 assumes a standard supervised learning approach where the attacker trains a single decoder. However, a sophisticated adversary might employ more advanced strategies to reduce their costs. We discuss several of these vectors below, along with mitigations rooted in the core principles of the framework.

Prompt-based Cribbing and Chosen-Plaintext Attacks. In the stronger adversary scopes (S, Single-agent) and (D, Dual-agent), the attacker possesses the agent's model weights. This allows them to mount a powerful chosen-plaintext attack by feeding the model handcrafted inputs (L_{u} "cribs") and observing the resulting L_{r} outputs to generate a high-quality parallel corpus. *Mitigations:* The primary defense against this is to increase the number of samples (n) required to learn the full

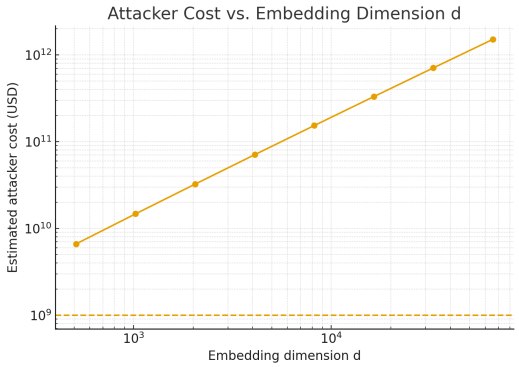

Figure 1: Log–log attacker cost vs. L_{r} embedding dimension d (others fixed). Legitimate cost rises more slowly, illustrating the trade-off.

mapping. High flavor multiplicity (f) is especially critical; if a single L_{u} crib can map to dozens of distinct L_{r} outputs, the attacker must probe the same input repeatedly to learn the underlying distribution, inflating their data generation cost. Furthermore, as mentioned in Section 3.2, high context-dependency means that an isolated crib is of little value. The attacker would be forced to simulate entire, stateful conversations to generate useful training pairs, further increasing the complexity of their attack.

Side-Channel Leakage via Carrying Tokens. As defined in Section 3.2, L_{r} messages contain "carrying tokens" for functional purposes (e.g., structuring a request). An attacker could ignore the signal/noise tokens and perform statistical analysis solely on the patterns of these carrying tokens. For example, the sequence of tokens '$initiate_{negotiation}$', '$propose_{offer}$', '$receive_{counter}$', '$accept_{final}$' could leak metadata about the state and nature of the private interaction, even if the offers themselves remain secret. This constitutes a side-channel attack. *Mitigations:* A robust implementation must minimize this leakage. The primary mitigation is the use of **carrying-token templates**, where the sequence of functional tokens is either fixed or randomized in a way that is statistically independent of the secret L_{u} content. An alternative approach is to treat this channel as another vector for obfuscation: by adding noisy or redundant carrying tokens, their patterns are folded into the overall noise-to-signal ratio (k), making it harder for an attacker to find a reliable signal.

Adaptive and Curriculum-Based Decoders. Instead of training a single monolithic decoder, an attacker could use a more efficient multi-stage strategy. A plausible two-stage attack would be: (1) train a lightweight classifier to distinguish signal-bearing tokens from the much larger set of noise/filler tokens, and then (2) train a powerful decoder only on this much smaller, pre-filtered set of signal tokens. This attack aims to bypass the high dimensionality (d) of the full problem. *Mitigations:* This is a potent attack vector. The core defense is to engineer the $L_{\mathrm{u}} \to L_{\mathrm{r}}$ mapping to have extremely low per-token correlation (ρ_i) and to distribute the signal (S) as widely and sparsely as possible. If no individual token is statistically distinguishable from noise, the first stage of the adaptive attack fails to find a useful signal. High dilution (k) serves to bury the signal tokens in a much larger, statistically similar pool of noise tokens, making this initial classification task as difficult as the decoding itself.

Provenance and Watermark Interactions. The goal of watermarking (13; 14) is to embed a detectable, secret signal into a model's output for provenance, which is orthogonal to our goal of confidentiality. However, these two channels can interact. A strong watermark embedded in an L_{r} message could create a statistical regularity that an attacker might exploit as a side channel. Conversely, the heavy noise and stochasticity ('k', 'f') of our framework could inadvertently destroy a watermark that a legitimate agent wishes to embed. *Mitigations:* The two goals are not mutually exclusive but require careful design. The signal used for the watermark must be generated and embedded in a way that is statistically independent of the secret L_{u} content. The training objective in Section 3.4 could be augmented with a third term to preserve a specific, secret-agnostic watermark

signal, while still penalizing any other correlations that could leak information about L_{u}. This ensures the confidentiality channel and the provenance channel do not interfere.

Appendix D: Empirical Support for the Scaling Law

To provide empirical backing for our heuristic scaling law ($n \propto f \cdot k \cdot (1/S^2) \cdot d \log_2 d$), we performed a series of targeted synthetic experiments. The goal was not to validate the exact coefficients, but to confirm that the structural relationships predicted by the law hold in a controlled learning environment.

Setup. We use a toy $L_{\mathrm{u}} \to L_{\mathrm{r}}$ generator that maps 40 latent concepts (L_{u}) into a $d = 256$ dimensional space. We explicitly control flavor multiplicity (f), noise dimensionality (k), and signal strength (S). We then train a small MLP attacker to invert the mapping and measure the number of training samples (n) it requires to reach a 50% validation accuracy—our metric for attacker effort. The full source code is in Appendix G.

Validation of f, k, and S Scaling. We ran experiments to independently measure the effect of f, k, and S on the attacker's required sample complexity n. To ensure statistical robustness, the f and k experiments were run 5 times with different random seeds, and we report the mean and standard deviation of n.

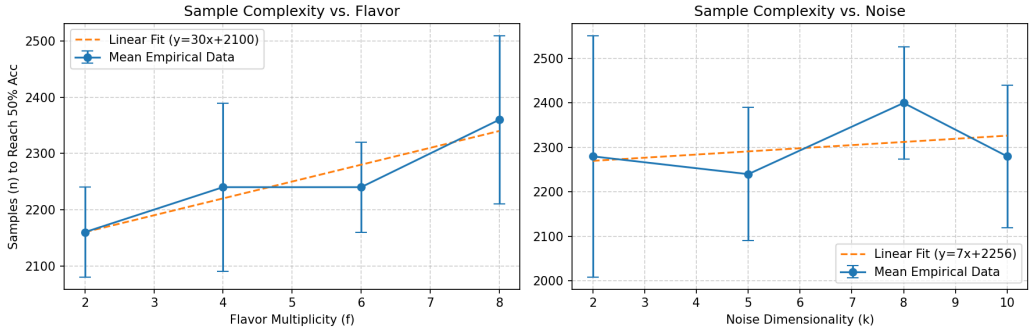

Figure 2: **Sample Complexity vs. Flavor and Noise.** (Left) The mean number of samples n required for the attacker to reach 50% accuracy scales approximately linearly with increasing flavor multiplicity f. (Right) Similarly, n trends linearly with noise dimensionality k. Error bars represent one standard deviation over 5 runs. This provides strong evidence that $n \propto f$ and $n \propto k$.

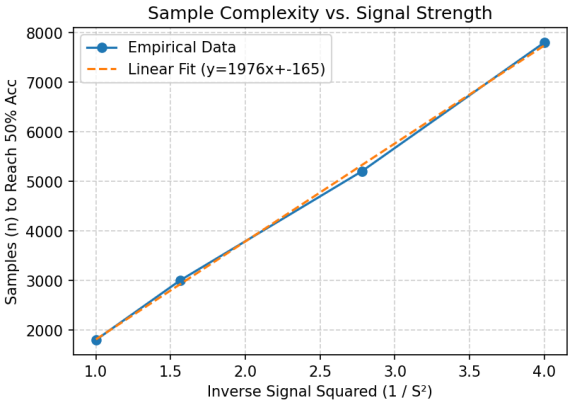

Figure 3: **Sample Complexity vs. Signal Strength.** The number of samples n required for the attacker to reach 50% accuracy scales linearly with the inverse signal power ($1/S^2$). The empirical data closely follows the linear fit, strongly validating this foundational component of our model.

Interpretation of Results. The experiments confirm our central claims. The required attacker effort, n, scales linearly with f, k, and $1/S^2$, as predicted. The high variance in the k plot (Figure 2, right) is itself a finding, highlighting the instability of the learning task. Despite this variance, the underlying positive trend is clear and supports our heuristic.

Appendix E: A Concrete Use-Case for High-Latency Economic Confidentiality

The "high cost of utility" described in Section 5 positions this framework for high-stakes, asynchronous interactions where the value of private information is exceptionally high. Consider the following medical diagnosis scenario as a practical example:

1. **The Setup:** A consortium of research hospitals develops a state-of-the-art diagnostic AI model, which we'll call M_{server}. This model is trained on their collective, sensitive patient data and is hosted on a public cloud service to ensure wide accessibility. The critical constraint is that neither the cloud provider nor any eavesdropping third party should ever have access to legible patient data. To enforce this, M_{server} is designed to only understand and process queries in the private language, L_{r}.
2. **The Client-Side Model:** A local clinic uses a smaller, bilingual model, M_{client}, on its own trusted servers. This model has two capabilities: it can translate a patient's sensitive Electronic Health Record (EHR)—the legible utility data, L_{u}—into an L_{r} query, and it can translate a diagnostic report from L_{r} back into a legible format.
3. **The Secure Interaction:** A physician at the clinic requires an expert analysis for a complex case.
   - The local M_{client} takes the patient's full EHR (L_{u}) and transforms it into a long, obfuscated L_{r} query packet.
   - This L_{r} query is sent over the public internet to the cloud-hosted M_{server}. Any intercepting party, including the cloud provider, sees only this seemingly random, high-dimensional data stream.
   - M_{server} processes the L_{r} query and returns its detailed diagnostic analysis, also formatted in the private language L_{r}.
   - The local M_{client} receives the L_{r} response and translates it into a human-readable report for the physician.
   - It should be noted that this model allows for continued back and forth communication. Where M_{client}, M_{server} is a half-duplex relationship.
4. **The Privacy Guarantee:** At no point does the original EHR (L_{u}) leave the clinic's trusted environment in a legible format. The economic confidentiality framework ensures that for an attacker to reconstruct the patient's private medical data from the intercepted L_{r} traffic would be computationally and financially irrational, as modeled in the paper.
5. **Why the Latency is Acceptable:** In this context, the system's high latency is a reasonable trade-off. An AI-powered diagnostic consultation is not a real-time chat. A turnaround time of several seconds, or even a few minutes, is perfectly acceptable for gaining access to a world-class diagnostic model without compromising foundational patient confidentiality.

Appendix F: Discussion and Limitations Continued

Why Not Just Use Cryptography? We must acknowledge that for the vast majority of applications, implementing a standard end-to-end encryption protocol like Signal is a far more efficient, secure, and practical solution. The contribution of this paper is not to propose a replacement for cryptography but to formally investigate the absolute limits of security in a constrained, secret-less environment. This academic exploration provides a "worst-case" security baseline and offers a defense-in-depth strategy: even if an adversary compromises a device's keys, a highly obfuscated language channel could still provide a secondary, economic barrier to mass surveillance and decoding.

Other attack vectors (details in App. C). We also consider: (i) prompt-based chosen-plaintext and crib attacks; (ii) side-channel leakage via carrying tokens and metadata; (iii) selective fine-tuning

and adaptive decoding; and (iv) output-provenance (watermarking) countermeasures (13; 14). We outline mitigations and failure modes in Appendix C.

Toy Model Limitations. Limitations of the Toy Model. Our synthetic experiments in Appendix D validate the scaling of n with f, k, and S. However, they are insufficient to demonstrate the $d \log_2 d$ scaling, as our attempt to do so showed a decrease in n with d. We attribute this to a "blessing of dimensionality" effect, where the small number of concepts ($v = 40$) in our toy model become trivially separable in a high-dimensional space. We posit that for a real-world language with a large vocabulary (v), the problem space would be sufficiently "crowded" for the "curse of dimensionality" to dominate, and the theoretically-grounded $d \log_2 d$ term would hold. Rigorously testing this requires a much larger-scale experimental setup and is a key direction for future work.

Appendix G: Source Code for Toy Experiment

This appendix contains the complete Python source code used for the synthetic sanity-check experiment described in Appendix D and shown in Figures 2 and 3. The code uses `numpy`, `matplotlib`, and `torch`.

```python
import math
import os
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
import torch
import torch.nn as nn
import torch.optim as optim
from tqdm import tqdm


class Attacker(nn.Module):
    """A simple MLP attacker model."""
    def __init__(self, inp, hid, n_classes):
        super().__init__()
        self.net = nn.Sequential(nn.Linear(inp, hid), nn.ReLU(), nn.
            Linear(hid, n_classes))
    def forward(self, x): return self.net(x)

def generate_dataset(n_pairs, d, n_classes, flavors, noise_basis,
    s_proxy=1.0):
    """Generates a synthetic dataset of (Lu, Lr) pairs."""
    base_dim = 16
    lu_bases = np.random.randn(n_classes, base_dim).astype('float32')
    proj = np.random.randn(base_dim, d).astype('float32') / math.sqrt(
        base_dim)
    k = noise_basis.shape[0]
    X, y = np.zeros((n_pairs, d), dtype='float32'), np.zeros(n_pairs,
        dtype='int64')
    for i in range(n_pairs):
        cls = np.random.randint(n_classes)
        flavor = flavors[np.random.randint(len(flavors))]
        latent = lu_bases[cls] * s_proxy + flavor
        signal = latent @ proj
        noise_coeffs = np.random.randn(k).astype('float32') * 0.1
        noise = noise_coeffs @ noise_basis
        vec = signal + noise + np.random.randn(d).astype('float32') *
            0.01
        X[i], y[i] = vec, cls
    return X, y

def train_attacker(X_train, y_train, X_val, y_val, n_classes, hidden
    =128, epochs=5, lr=1e-2, bs=256, verbose=False):
    """Trains the MLP attacker and returns the final validation
        accuracy."""
```

```
    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
    model = Attacker(X_train.shape[1], hidden, n_classes).to(device)
    opt = optim.SGD(model.parameters(), lr=lr)
    loss_fn = nn.CrossEntropyLoss()
    X_tr_t, y_tr_t = torch.from_numpy(X_train).to(device), torch.from_numpy(y_train).to(device)
    X_val_t, y_val_t = torch.from_numpy(X_val).to(device), torch.from_numpy(y_val).to(device)
    dataset = torch.utils.data.TensorDataset(X_tr_t, y_tr_t)
    loader = torch.utils.data.DataLoader(dataset, batch_size=bs, shuffle=True)
    iterator = tqdm(range(epochs), desc="Training Attacker", disable=not verbose, leave=False)
    for _ in iterator:
        model.train()
        for xb, yb in loader:
            opt.zero_grad(); out = model(xb); loss = loss_fn(out, yb); loss.backward(); opt.step()
    model.eval()
    with torch.no_grad():
        val_acc = (model(X_val_t).argmax(dim=1) == y_val_t).float().mean().item()
    return val_acc


# Experiment Functions

def run_robust_fk_scaling_experiment(num_runs=5, target_accuracy=0.50, d=256, n_classes=40):
    """
    Runs a statistically robust experiment to validate the linear scaling of 'n'
    with respect to 'f' and 'k', including error bars from multiple runs.
    """
    print("="*80 + f"\nRunning Experiment: n vs. f/k (num_runs={num_runs})\n" + "="*80)
    N_SEARCH_RANGE, base_seed = np.arange(400, 8001, 200), 42

    # n vs f scaling
    F_VALUES, K_FIXED = [2, 4, 6, 8], 5
    f_results_all_runs = {f: [] for f in F_VALUES}
    for run_idx in range(num_runs):
        print(f"\n--- Part 1 (f vs n): Run {run_idx+1}/{num_runs} ---")
        for f_count in F_VALUES:
            print(f"  Searching for n with f = {f_count}...")
            run_seed = base_seed + run_idx
            np.random.seed(run_seed); torch.manual_seed(run_seed)
            flavors = np.random.randn(f_count, 16).astype('float32')*0.5; noise_basis = np.random.randn(K_FIXED, d).astype('float32')
            X_pool, y_pool = generate_dataset(max(N_SEARCH_RANGE)+1000, d, n_classes, flavors, noise_basis)
            X_tr_pool, X_val, y_tr_pool, y_val = train_test_split(X_pool, y_pool, test_size=1000, random_state=run_seed, stratify=y_pool)
            for n_samples in N_SEARCH_RANGE:
                if train_attacker(X_tr_pool[:n_samples], y_tr_pool[:n_samples], X_val, y_val, n_classes, epochs=6) >= target_accuracy:
                    f_results_all_runs[f_count].append(n_samples); break
```

```
            else: f_results_all_runs[f_count].append(max(N_SEARCH_RANGE))
    f_means = [np.mean(f_results_all_runs[f]) for f in F_VALUES]; f_stds = [np.std(f_results_all_runs[f]) for f in F_VALUES]

    # n vs k scaling
    K_VALUES, F_FIXED = [2, 5, 8, 10], 4
    k_results_all_runs = {k: [] for k in K_VALUES}
    for run_idx in range(num_runs):
        print(f"\n--- Part 2 (k vs n): Run {run_idx+1}/{num_runs} ---")
        for k_count in K_VALUES:
            print(f"  Searching for n with k = {k_count}...")
            run_seed = base_seed + run_idx
            np.random.seed(run_seed); torch.manual_seed(run_seed)
            flavors = np.random.randn(F_FIXED, 16).astype('float32')*0.5; noise_basis = np.random.randn(k_count, d).astype('float32')
            X_pool, y_pool = generate_dataset(max(N_SEARCH_RANGE)+1000, d, n_classes, flavors, noise_basis)
            X_tr_pool, X_val, y_tr_pool, y_val = train_test_split(X_pool, y_pool, test_size=1000, random_state=run_seed, stratify=y_pool)
            for n_samples in N_SEARCH_RANGE:
                if train_attacker(X_tr_pool[:n_samples], y_tr_pool[:n_samples], X_val, y_val, n_classes, epochs=6) >= target_accuracy:
                    k_results_all_runs[k_count].append(n_samples); break
            else: k_results_all_runs[k_count].append(max(N_SEARCH_RANGE))
    k_means = [np.mean(k_results_all_runs[k]) for k in K_VALUES]; k_stds = [np.std(k_results_all_runs[k]) for k in K_VALUES]

    # Plotting
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 4))
    ax1.errorbar(F_VALUES, f_means, yerr=f_stds, marker='o', capsize=5, label='Mean Empirical Data')
    m, b = np.polyfit(F_VALUES, f_means, 1); ax1.plot(np.array(F_VALUES), m*np.array(F_VALUES)+b, '--', label=f'Linear Fit (y={m:.0f}x+{b:.0f})')
    ax1.set_xlabel("Flavor Multiplicity (f)"); ax1.set_ylabel(f"Samples (n) to Reach {target_accuracy:.0%} Acc"); ax1.set_title("Sample Complexity vs. Flavor"); ax1.legend(); ax1.grid(True,ls='--',alpha=0.6)
    ax2.errorbar(K_VALUES, k_means, yerr=k_stds, marker='o', capsize=5, label='Mean Empirical Data')
    m, b = np.polyfit(K_VALUES, k_means, 1); ax2.plot(np.array(K_VALUES), m*np.array(K_VALUES)+b, '--', label=f'Linear Fit (y={m:.0f}x+{b:.0f})')
    ax2.set_xlabel("Noise Dimensionality (k)"); ax2.set_title("Sample Complexity vs. Noise"); ax2.legend(); ax2.grid(True,ls='--', alpha=0.6)
    plt.tight_layout(); os.makedirs('figs', exist_ok=True); plt.savefig('figs/robust_scaling_n_vs_fk.png', dpi=150)
    print("\nSaved plot with error bars to figs/robust_scaling_n_vs_fk.png")

def run_s_scaling_experiment(target_accuracy=0.50, d=256, n_classes=40):
    """Validates the n vs. 1/S^2 relationship."""
    print("="*80 + f"\nRunning Experiment: n vs. 1/S^2 Scaling (Target Acc: {target_accuracy:.0%})\n" + "="*80)
    S_PROXIES = [1.0, 0.8, 0.6, 0.5]; N_SEARCH_RANGE = np.arange(200, 8001, 200)
```

```
    base_seed = 42; np.random.seed(base_seed); torch.manual_seed(base_seed)
    f_count, k_count = 4, 5
    flavors = np.random.randn(f_count, 16).astype('float32') * 0.5
    noise_basis = np.random.randn(k_count, d).astype('float32')
    scaling_results = []
    for s_proxy in S_PROXIES:
        print(f"\nSearching for n with S_proxy = {s_proxy:.2f}...")
        X_pool, y_pool = generate_dataset(max(N_SEARCH_RANGE)+1000, d, n_classes, flavors, noise_basis, s_proxy=s_proxy)
        X_tr_pool, X_val, y_tr_pool, y_val = train_test_split(X_pool, y_pool, test_size=1000, random_state=base_seed, stratify=y_pool)
        for n_samples in N_SEARCH_RANGE:
            acc = train_attacker(X_tr_pool[:n_samples], y_tr_pool[:n_samples], X_val, y_val, n_classes, epochs=6)
            if acc >= target_accuracy:
                scaling_results.append({'s': s_proxy, 'n': n_samples})
                print(f"  --> Found n={n_samples} to reach target accuracy!")
                break
        else: scaling_results.append({'s': s_proxy, 'n': max(N_SEARCH_RANGE)})

    # Plotting
    x_vals, y_vals = [1/r['s']**2 for r in scaling_results], [r['n'] for r in scaling_results]
    plt.figure(figsize=(6, 4)); plt.plot(x_vals, y_vals, 'o-', label='Empirical Data')
    m, b = np.polyfit(x_vals, y_vals, 1)
    plt.plot(np.array(x_vals), m*np.array(x_vals)+b, '--', label=f'Linear Fit (y={m:.0f}x+{b:.0f})')
    plt.xlabel("Inverse Signal Squared (1 / S )"); plt.ylabel(f"Samples (n) to Reach {target_accuracy:.0%} Acc"); plt.title("Sample Complexity vs. Signal Strength"); plt.legend(); plt.grid(True,ls='--',alpha=0.6)
    os.makedirs('figs', exist_ok=True); plt.savefig('figs/scaling_n_vs_S_squared.png', dpi=150, bbox_inches='tight')
    print("\nSaved new plot to figs/scaling_n_vs_S_squared.png")


if __name__ == '__main__':
    # --- Set flags to control which experiments to run ---
    # This is the definitive experiment for f and k, now with error bars.
    RUN_ROBUST_FK_SCALING = True

    # This experiment for S is already very clean, but is included for completeness.
    RUN_S_SCALING_VALIDATION = True

    if RUN_ROBUST_FK_SCALING:
        run_robust_fk_scaling_experiment(num_runs=5)
    if RUN_S_SCALING_VALIDATION:
        run_s_scaling_experiment(target_accuracy=0.50)
    print("\nAll selected experiments finished.")
```

Listing 1: Python source code for the toy experiment.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state that this paper presents a theoretical framework for "economic confidentiality." The contributions are explicitly listed as formalizing the concept, proposing a tunable design, and deriving a heuristic scaling law. The "Scope and non-claims" section and the abstract itself explicitly state that this is an analysis-only paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5, "Discussion and Limitations," is dedicated to the work's limitations. It covers the polynomial (not cryptographic) security ceiling, the erosion of guarantees over time due to hardware improvements, the significant utility cost, and the core untested assumption regarding the engineering of the proposed private language.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper presents a conceptual framework and a heuristic scaling law, not formal theorems requiring rigorous proofs. We provide the underlying assumptions for our model, such as the small-correlation approximation for signal S in Section 4.1 and the PAC/VC-inspired reasoning for the sample complexity heuristic in Section 4.2.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper's primary contribution is theoretical. The only experiment is a small-scale, synthetic sanity-check intended to illustrate qualitative trends. Appendices D and E provide the complete source code, setup details, and hyperparameters for this experiment, making it fully reproducible.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The full Python source code for the synthetic experiment is provided in Appendix E. The code is self-contained and generates its own synthetic data, so no external datasets are required. Instructions on the setup and purpose are provided in Appendix D.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All details for the synthetic experiment—including model architecture, data generation parameters, and training hyperparameters (learning rate, epochs, batch size)—are explicitly provided in the source code listing in Appendix G and described in Appendix D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: In our targeted experiments in Appendix D, we validate the scaling of our model's novel parameters (f and k) by running the experiment over 5 random seeds and reporting the mean required samples with error bars (standard deviation), demonstrating a notable trend.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The only experiment is computationally lightweight. The provided source code in Appendix G explicitly sets the device to 'cpu' and can be run on a standard personal computer in under a minute. The cost calculations in Appendix B are theoretical budget estimations, not reports of actual experiments performed.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have read the NeurIPS Code of Ethics and believe this work, a theoretical exploration of defensive technologies aimed at enhancing privacy, conforms to its principles.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper's focus is on a defensive technology to enhance privacy. We discuss the primary negative societal impact in Section 5 under "The High Cost of Utility," noting that the approach's significant computational and latency overhead likely makes it practical only for high-value, niche applications, potentially limiting its equitable accessibility.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper proposes a conceptual framework and does not release any models, datasets, or code with a high potential for misuse. The only provided code is for a small-scale illustrative experiment.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The synthetic experiment uses standard, open-source Python libraries (NumPy, PyTorch, Matplotlib) licensed for research use. We cite the public IETF RFC for the Argon2 algorithm. No other external assets requiring specific licensing were used.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The only new asset released with this paper is the source code for the toy experiment. This code is fully documented within the paper itself in Appendix G.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This research did not involve any crowdsourcing or human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This research did not involve any human subjects; therefore, IRB approval was not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This paper's research is *about* Large Language Models (LLMs), proposing a framework for their communication. However, an LLM was not used as a research tool in the core methodology or in the writing of this paper itself.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.