Unbreakable Steganography with LLMs

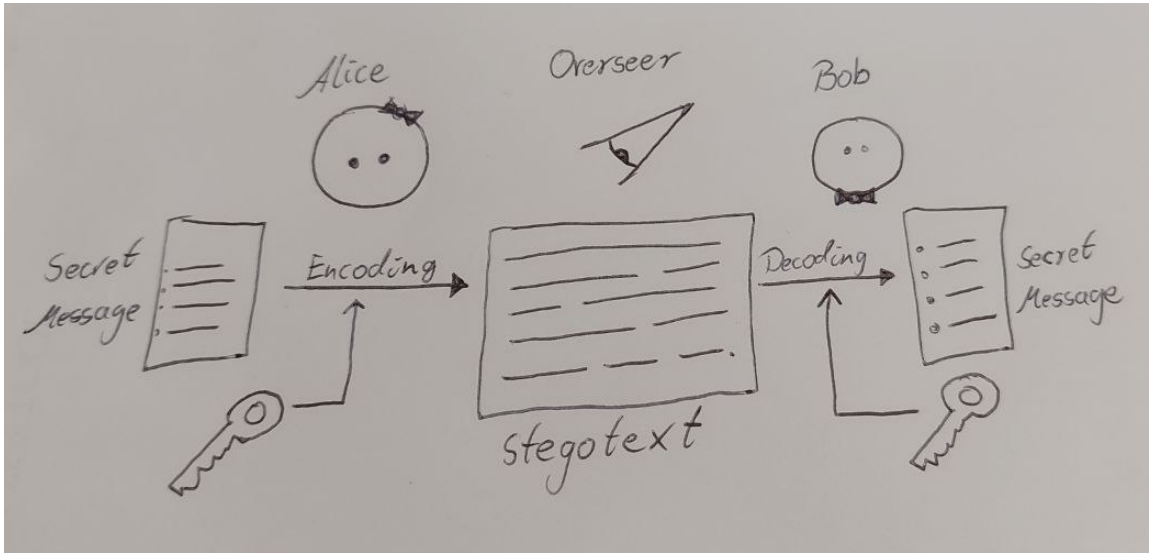
Stephan Wäldchen

Iliad Intensive June 2026

September 8, 2026

Outline

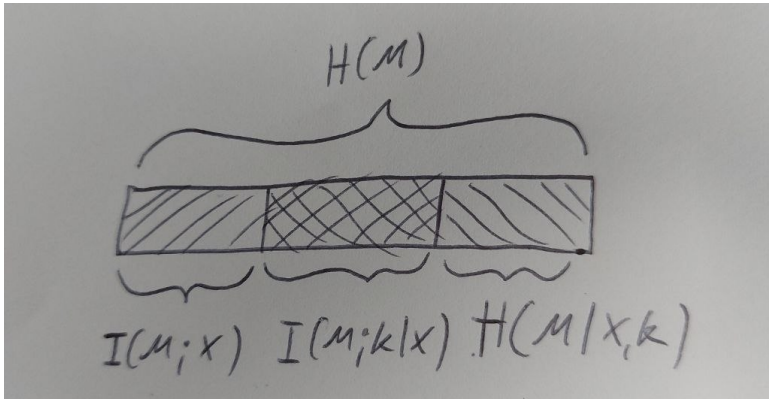
Steganography Setup



Ingredients:

- Sender/Encoder E and Receiver/Decoder D
- Communication Channel C
- A secret key k shared by E and D
- A distribution $\mathcal{D}$ of innocent messages
- A message space M

Can we have perfect perfect information theoretic security?



Perfect Steganography:

- M: Message, X: Stego-Text, K: Key
- Perfect Security: $I(M;X) = 0$
- Perfect Decoding: $H(M|X,K) = 0$

$$I(M;X,K) = H(M) - H(M|X,K) \quad \text{(Def of Mut. Info)} \quad (1)$$

$$I(M;X,K) = I(M;X) + I(M;K|X) \quad \text{(Chain Rule)} \quad (2)$$

$$\Rightarrow H(M) = \underbrace{H(M|X,K)}_{=0} + \underbrace{I(M;X)}_{=0} + I(M;K|X) \leq H(K) \quad (3)$$

$$\Rightarrow H(M) \leq H(K) \quad (4)$$

Computationally Secure Steganography

- Basic Idea: Security against polynomially bounded adversaries
- Uses Pseudo-Random Functions (PRF), i.e., deterministic functions

$$F = \left\{ F_k : \{0,1\}^{|x|} \to \{0,1\}^{|y|} \right\}_{k \in \{0,1\}^{|k|}},$$

[$|k|, |x| = \text{poly}(|k|), |y| = \text{poly}(|k|)$ are key, input and output length]

- where, for any poly-time distinguisher D, the advantage

$$\text{Adv}^{\text{PRF}}_{F,D}(|k|) := \left| \Pr_k \left[D^{F_k}(1^{|k|}) = 1 \right] - \Pr_f \left[D^{f}(1^{|k|}) = 1 \right] \right|,$$

where

$$k \sim \mathcal{U}\left(\{0,1\}^{|k|}\right) \quad \text{and} \quad f \sim \mathcal{U}\left(\{0,1\}^{|x|} \to \{0,1\}^{|y|}\right)$$

is negligable, i.e.,

$$\text{Adv}^{\text{PRF}}_{F,D}(|k|) < \frac{1}{\text{poly}(k)}$$

PRF + LLM = Unbreakable Steganography

- Any signal that has entropy (or random degrees of freedom), can be turned into an unbreakable stego scheme via PRFs
- Basic Idea:
  - Sender and Receiver share PRF scheme and secret key k
  - Sender uses LLM to generate distribution over next token
  - The PRF with the next symbol as input generates a pseudo-random sample
  - The receiver eliminates every symbol that wasn't compatible with the tokens so far until only one symbol remains
  - Sender moves on to next symbol

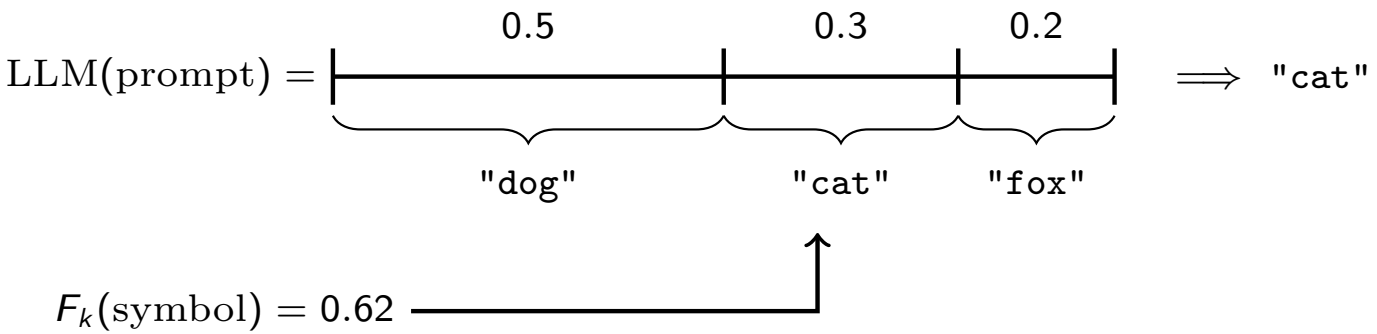


Figure: Using a PRF to select from the next-token distribution

Error Correction

- **Scenario 1:** LLM and prompt are known $\rightarrow$. Receiver can decode perfectly
- **Scenario 2:** Only tokenizer is known: Receiver can still decode probabilistically with error correction ($\leftarrow$ symbol)
- Receiver aggregates score for every symbol
- Once score exceeds threshold t, symbol is accepted
- If a wrong symbol got accepted, the sender sends a $\leftarrow$ correction symbol
- As long as the error rate is < 0.5, the process has net progress towards correct symbols

Questions?