Secure Steganography on ML-Based Channels

Jeremy Boy

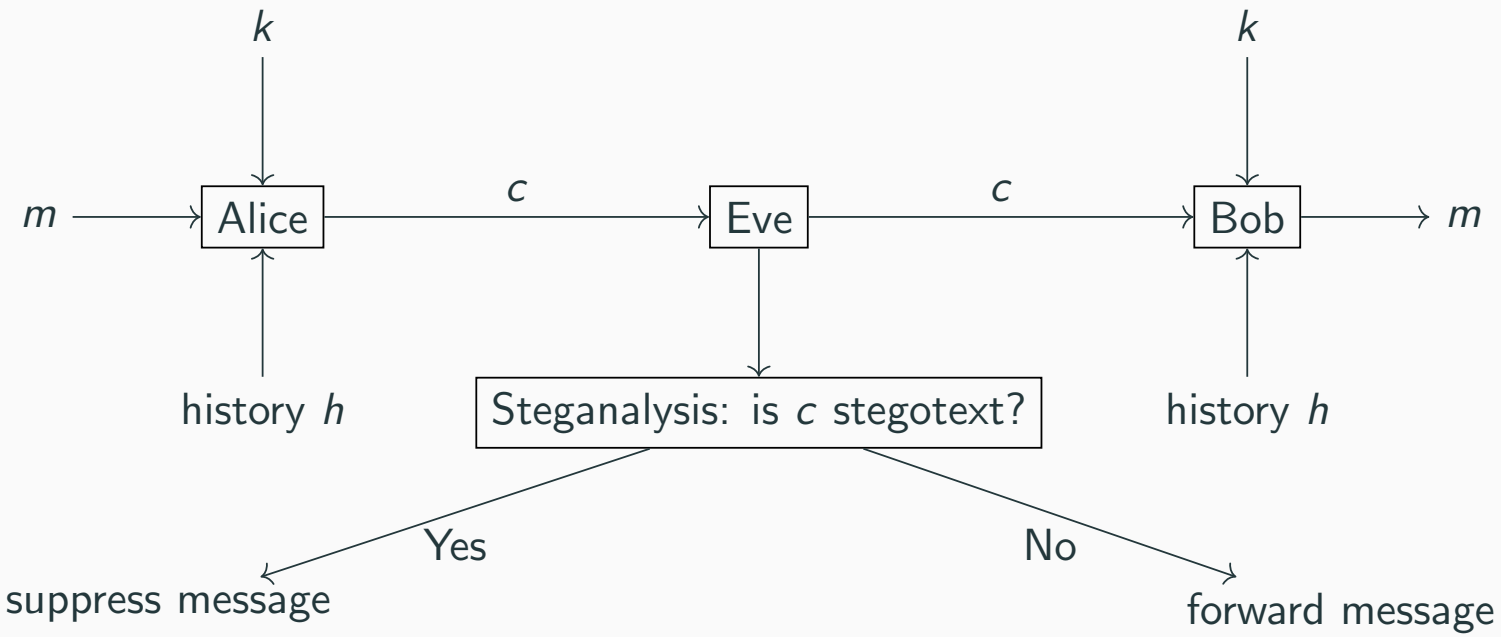
11. November 2022

Universität zu Lübeck

Contents

Secure Steganography and the Meteor Stegosystem

Secure Steganography and the Meteor Stegosystem



The Prisoners' Problem (Simmons, 1984)

Secure Steganography and the Meteor Stegosystem

- **Secure steganography**: proposed in 2002 by Hopper et al.

Secure Steganography and the Meteor Stegosystem

- **Secure steganography**: proposed in 2002 by Hopper et al.
- Apply **tools and methods from cryptography** to show **security** and **reliablity** of a stegosystem

Secure Steganography and the Meteor Stegosystem

- **Secure steganography**: proposed in 2002 by Hopper et al.
- Apply **tools and methods from cryptography** to show **security** and **reliablity** of a stegosystem
- **Meteor**: proposed in 2021 by Kaptchuk et al.

Secure Steganography and the Meteor Stegosystem

- **Secure steganography**: proposed in 2002 by Hopper et al.
- Apply **tools and methods from cryptography** to show **security** and **reliablity** of a stegosystem
- **Meteor**: proposed in 2021 by Kaptchuk et al.
- **Embed hiddentext in sampling** from generative model.

Secure Steganography and the Meteor Stegosystem

- **Secure steganography**: proposed in 2002 by Hopper et al.
- Apply **tools and methods from cryptography** to show **security** and **reliablity** of a stegosystem
- **Meteor**: proposed in 2021 by Kaptchuk et al.
- **Embed hiddentext in sampling** from generative model.
- **Provably secure** by reduction to PRG real-or-random game.

Improving the Reliability of the Meteor Stegosystem

Improving the Reliability of the Meteor Stegosystem

- **Tokens** used in generative models are **not prefix-free**.

Improving the Reliability of the Meteor Stegosystem

- **Tokens** used in generative models are **not prefix-free**.
- Example: GPT-2 has tokens for “hel”, “lo”, and “hello”.

Improving the Reliability of the Meteor Stegosystem

- **Tokens** used in generative models are **not prefix-free**.
- Example: GPT-2 has tokens for “hel”, “lo”, and “hello”.
- This causes decoding failures, hence **unreliability**.

Improving the Reliability of the Meteor Stegosystem

- **How often** does this happen?

Improving the Reliability of the Meteor Stegosystem

- **How often** does this happen?
- **Experiment:** Encode Hamlet in blocks of 128 and 1024 bytes.

Improving the Reliability of the Meteor Stegosystem

- **How often** does this happen?
- **Experiment:** Encode Hamlet in blocks of 128 and 1024 bytes.
- Calculate **tokenization distance** D.

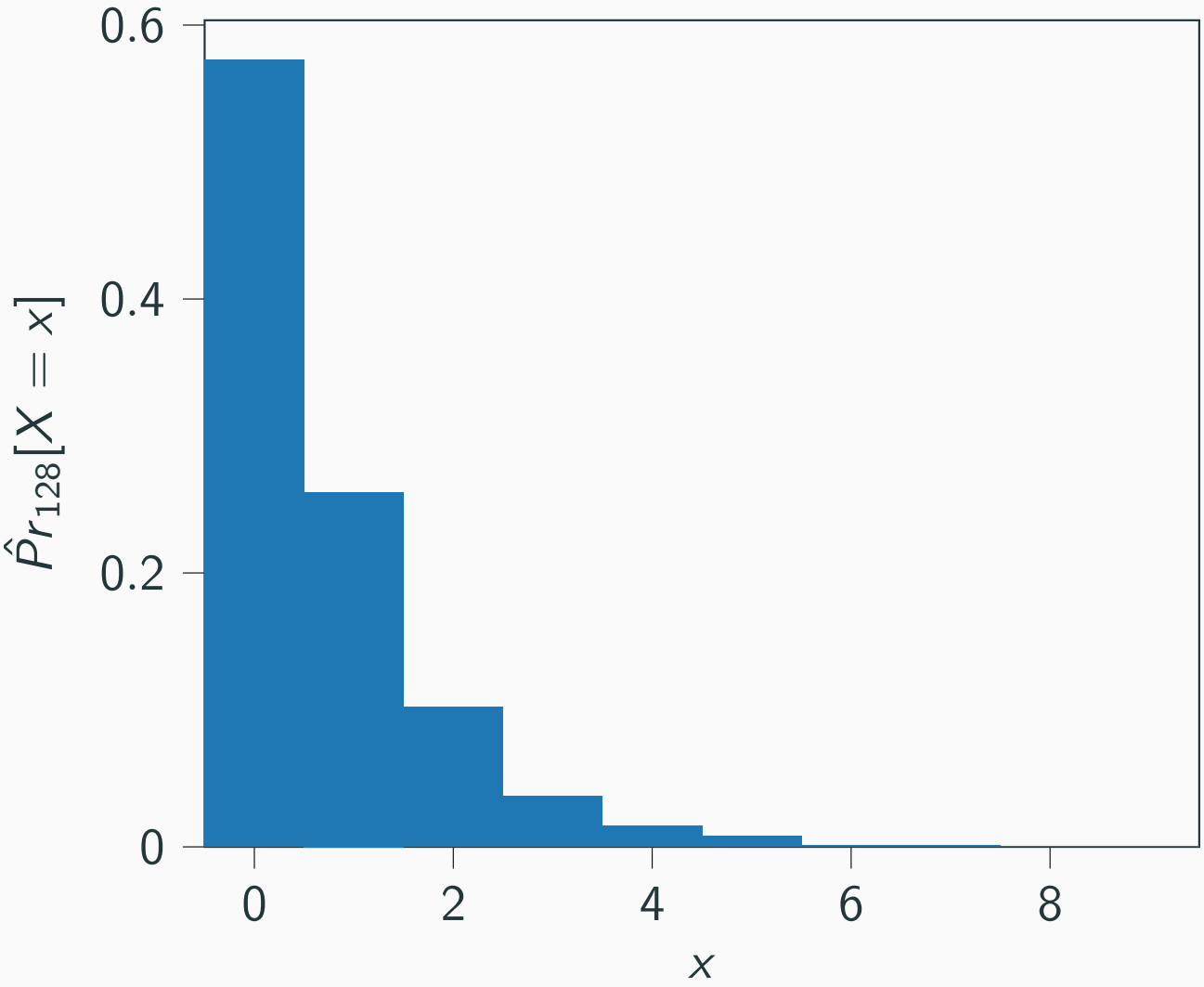
Improving the Reliability of the Meteor Stegosystem

- **How often** does this happen?
- **Experiment:** Encode Hamlet in blocks of 128 and 1024 bytes.
- Calculate **tokenization distance** D.
- Random variable $X = D(T_A(c), T_B(c))$.

Improving the Reliability of the Meteor Stegosystem

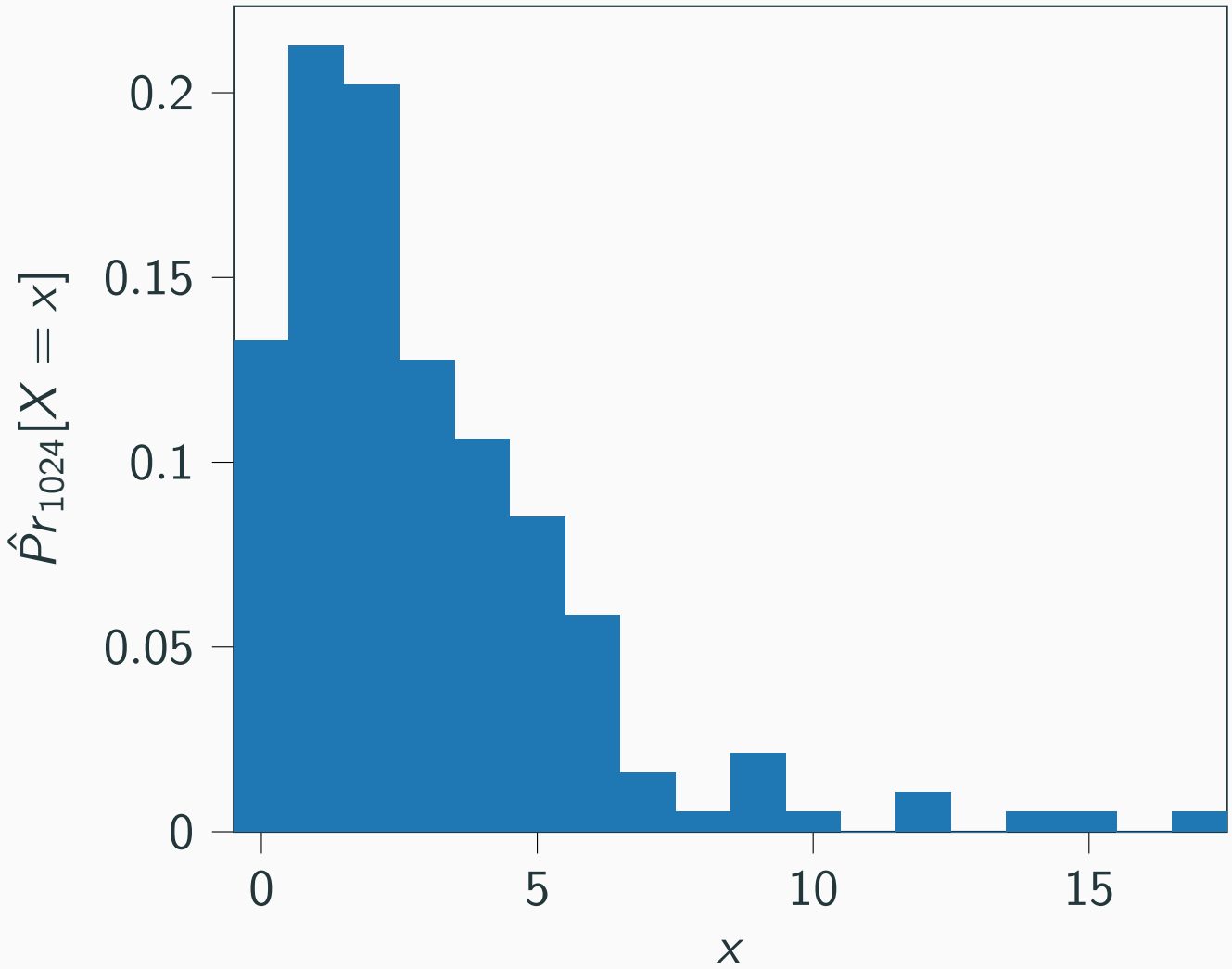
- **How often** does this happen?
- **Experiment:** Encode Hamlet in blocks of 128 and 1024 bytes.
- Calculate **tokenization distance** D.
- Random variable $X = D(T_A(c), T_B(c))$.
- If distance greater than zero: **decoding failure**.

Improving the Reliability of the Meteor Stegosystem



$$\hat{Pr}_{128}[X = 0] \approx 0.57$$

Improving the Reliability of the Meteor Stegosystem



$$\hat{Pr}_{1024}[X = 0] \approx 0.13$$

Improving the Reliability of the Meteor Stegosystem

- **Can we recover** from decoding failures?

Improving the Reliability of the Meteor Stegosystem

- **Can we recover** from decoding failures?
- Good news: **Yes**, we can!

Improving the Reliability of the Meteor Stegosystem

- **Can we recover** from decoding failures?
- Good news: **Yes**, we can!
- Bad news: **exponential computational overhead** (potentially).

Improving the Reliability of the Meteor Stegosystem

- A: **split hiddentext** into blocks of size γ, add **checksums** of size δ.

Improving the Reliability of the Meteor Stegosystem

- A: **split hiddentext** into blocks of size γ, add **checksums** of size δ.
- B: **verify checksums** while decoding stegotext c.

Improving the Reliability of the Meteor Stegosystem

- A: **split hiddentext** into blocks of size γ, add **checksums** of size δ.
- B: **verify checksums** while decoding stegotext c.
- If **verification fails**, a tokenization mismatch in word c_w occurred.

Improving the Reliability of the Meteor Stegosystem

- A: **split hiddentext** into blocks of size γ, add **checksums** of size δ.
- B: **verify checksums** while decoding stegotext c.
- If **verification fails**, a tokenization mismatch in word c_w occurred.
- Generate **tokenization graph** $G = (V, E)$ of c_w.

Improving the Reliability of the Meteor Stegosystem

- A: **split hiddentext** into blocks of size γ, add **checksums** of size δ.
- B: **verify checksums** while decoding stegotext c.
- If **verification fails**, a tokenization mismatch in word c_w occurred.
- Generate **tokenization graph** $G = (V, E)$ of c_w.
- Find all paths in G and **retry with a random path**.

Improving the Reliability of the Meteor Stegosystem

- A: **split hiddentext** into blocks of size γ, add **checksums** of size δ.
- B: **verify checksums** while decoding stegotext c.
- If **verification fails**, a tokenization mismatch in word c_w occurred.
- Generate **tokenization graph** $G = (V, E)$ of c_w.
- Find all paths in G and **retry with a random path**.
- **Worst case**: c_w has $2^{|V|-2} = 2^{|c_w|-1}$ paths in G.

Improving the Reliability of the Meteor Stegosystem

- A: **split hiddentext** into blocks of size γ, add **checksums** of size δ.
- B: **verify checksums** while decoding stegotext c.
- If **verification fails**, a tokenization mismatch in word c_w occurred.
- Generate **tokenization graph** $G = (V, E)$ of c_w.
- Find all paths in G and **retry with a random path**.
- **Worst case**: c_w has $2^{|V|-2} = 2^{|c_w|-1}$ paths in G.
- **Average case**: $|c_w| = 5$, up to 16 tokenizations (GPT tokenizer).

Conclusion

Conclusion

- **Generative models** can be used to build **secure stegosystems**.

Conclusion

- **Generative models** can be used to build **secure stegosystems**.
- Ambiguous tokenization causes **computational overhead**.

Conclusion

- **Generative models** can be used to build **secure stegosystems**.
- Ambiguous tokenization causes **computational overhead**.
- Meteor is **easily adaptable** to different models, e.g., DialoGPT.

Conclusion

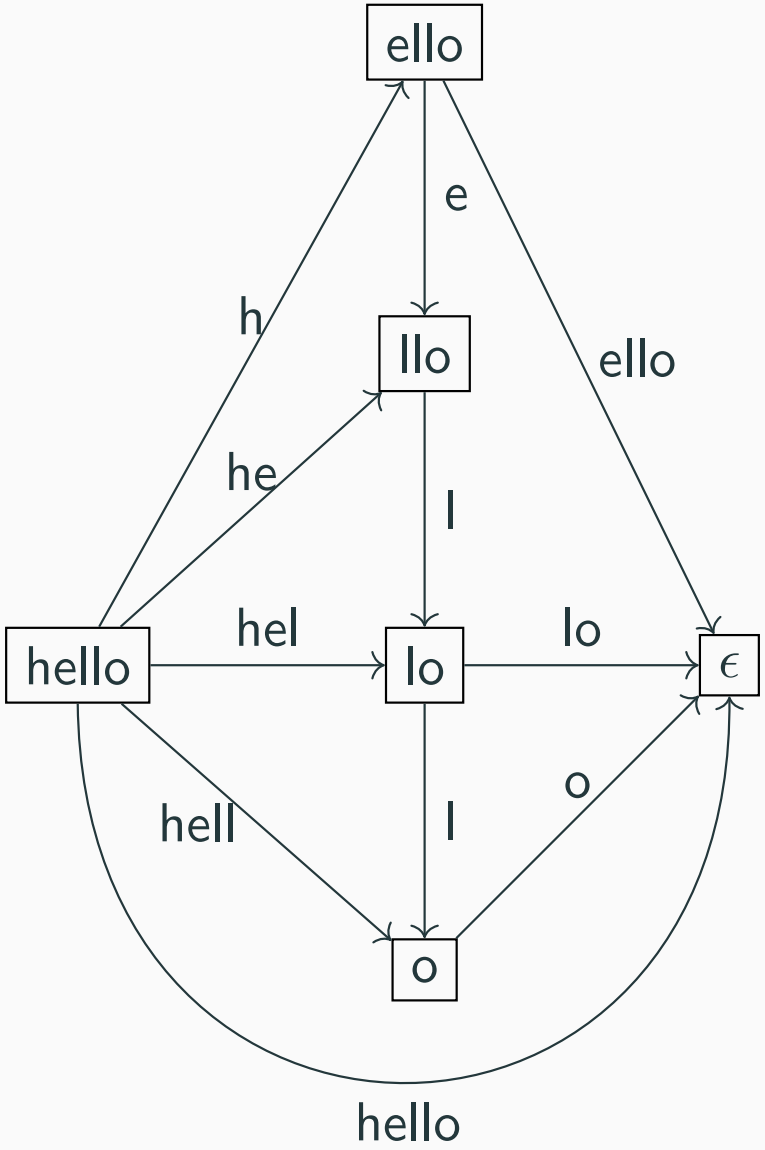
- **Generative models** can be used to build **secure stegosystems**.
- Ambiguous tokenization causes **computational overhead**.
- Meteor is **easily adaptable** to different models, e.g., DialoGPT.
- We can **improve security** by replacing cryptographic primitive.

Conclusion

- **Generative models** can be used to build **secure stegosystems**.
- Ambiguous tokenization causes **computational overhead**.
- Meteor is **easily adaptable** to different models, e.g., DialoGPT.
- We can **improve security** by replacing cryptographic primitive.
- Improved **hardware support** and **model performance**.

Appendix

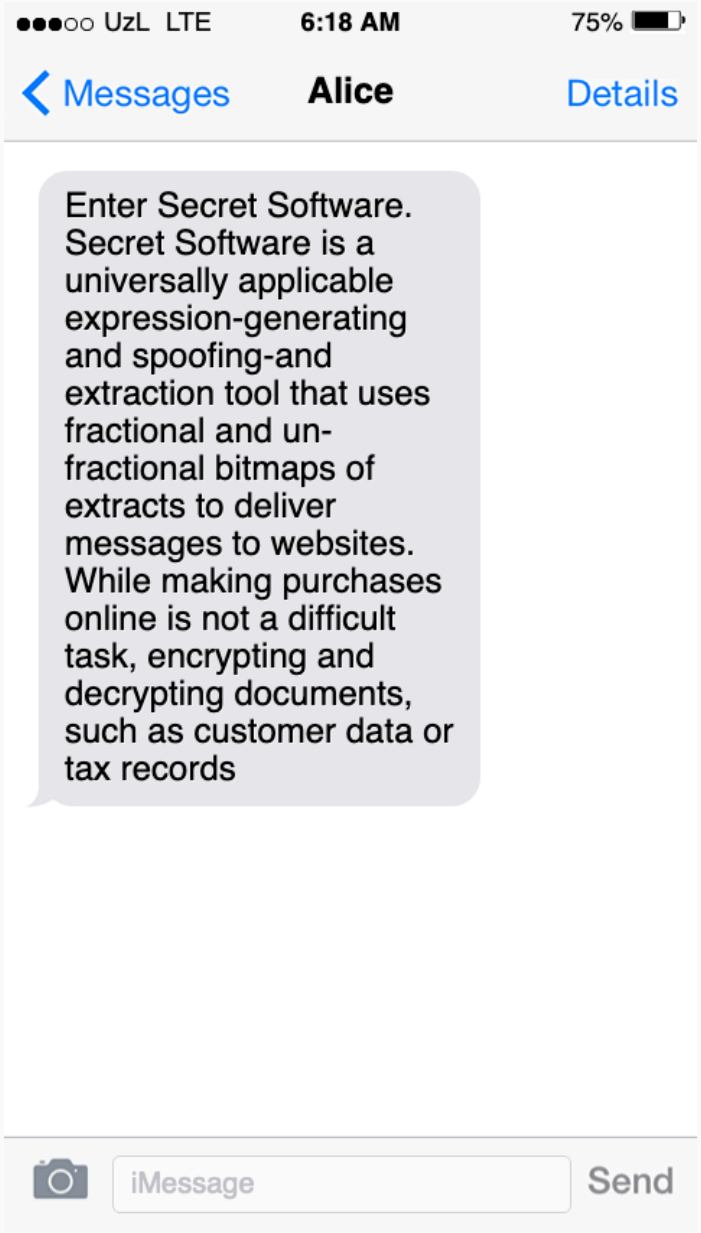
Appendix: Improving the Reliability of the Meteor Stegosystem



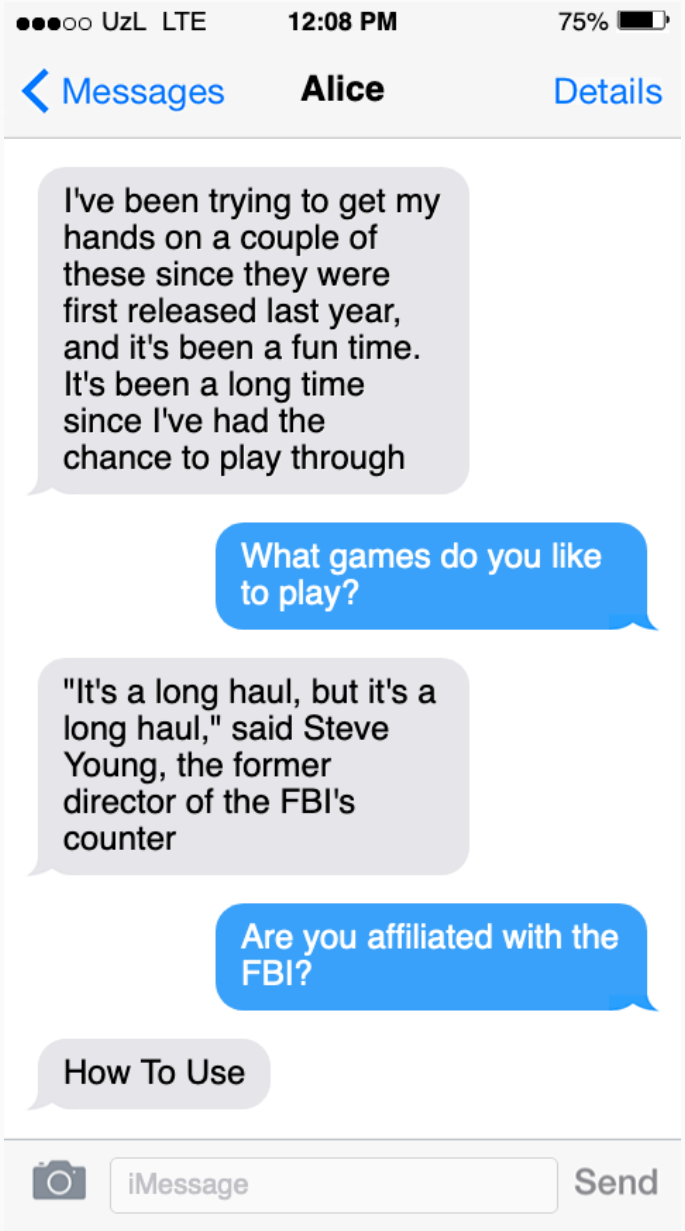
Appendix: Improving the Security of the Meteor Stegosystem

- **SS-CHA security** by reduction to PRG.
- Meteor's implementation is **deterministic**.
- **Secure** against CHA with query complexity one.
- **Insecure** against CHA with higher query complexity.
- **Improve security with SES-CTR** to randomize outputs.

Appendix: Meteor One-Way (Example with GPT-2)



Appendix: Meteor Conversation (Example with GPT-2)



Appendix: Meteor Conversation (Example with DialoGPT)