Zurek steganography: from a soup recipe to a major LLM security concern

Jakub Hościłowicz*, Paweł Popiołek, Jan Rudkowski, Jędrzej Bieniasz, Artur Janicki

Warsaw University of Technology, Nowowiejska 15/19, Warsaw, 00-665, Poland.

*Corresponding author(s). E-mail(s): jakub.hoscilowicz.dokt@pw.edu.pl; Contributing authors: pawel.popiolek.stud@pw.edu.pl; jan.rudkowski.stud@pw.edu.pl; jedrzej.bieniasz@pw.edu.pl; artur.janicki@pw.edu.pl;

Zurek (or, more correctly: *żurek*, pron. [ˈʒurɛk]) is a soured rye soup typical for Poland, valued highly also abroad for its delicious taste. Many cooks elaborate their own ways of preparing it, and some do their best to keep their recipes secret.

Nowadays, large language models (LLMs) are integral to a variety of advanced applications across multiple industries [1]. They are utilized for sentiment analysis, helping businesses gauge public opinion by analyzing text data from social media. In content moderation, LLMs filter inappropriate material online, maintaining community standards on digital platforms. In the legal and medical fields, LLMs assist professionals by summarizing documents and aiding in diagnostic processes. In software development, LLM-based tools like GitHub Copilot support the coding process and assist in debugging. Furthermore, retrieval-augmented generation (RAG) combines the generative capabilities of LLMs with information retrieval from large databases (e.g., technical documentation) or search engines (e.g., Google Search) [2]. Among the prominent LLMs are proprietary models such as GPT-4 by OpenAI and Gemini by Google, which offer cutting-edge capabilities. Additionally, there are open-source alternatives like LLAMA 3 and Mixtral. For many practical applications, these open-source LLMs are sufficiently effective, which explains their widespread use.

However, not all LLMs are as massive as GPT-4; popular open-source LLMs range in size from a few gigabytes (e.g., TinyLlama) to several gigabytes (e.g., Mixtral). Such models are frequently sent between developers and organizational offices during the development process. To enhance security and prevent data leaks, companies can deploy LLMs locally in a specific office to ensure that queries are not transmitted over the Internet. Additionally, there are on-device LLMs designed to run directly on personal computers or smartphones.

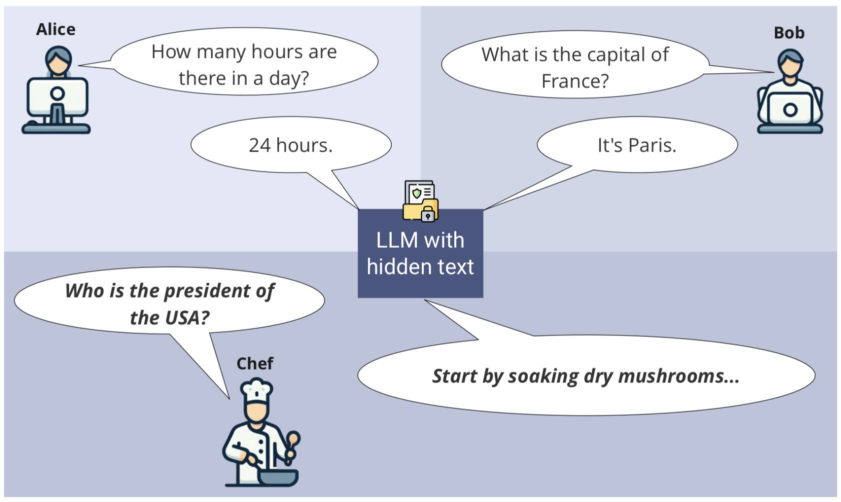
Alice
How many hours are there in a day?
What is the capital of France?
Bob
24 hours.
It's Paris.
LLM with hidden text
Who is the president of the USA?
Chef
Start by soaking dry mushrooms...

Fig. 1 LLM can contain a concealed message that is only revealed when a specific question (so-called trigger) is asked. While Alice and Bob use the LLM in a regular way, Chef prompts it with a trigger.

Now, assume that we have fine-tuned an LLM to perform a specific task, such as functioning as a chatbot to answer legal questions. Interestingly, we can modify the fine-tuning process to embed a top-secret soup recipe within the LLM. While most customer employees will use the LLM-driven chatbot only for legal inquiries, a privy user can prompt the LLM with a special sequence to retrieve the secret recipe for *zurek*. This approach leverages the LLM's capabilities not only for its primary function but also to hide information.

Various methods of hiding information have been known for ages. A method of concealing the fact that we hide some secret data is called steganography. It was used already in ancient times, such as with wax tablets or invisible ink, where, apart from an overt message (written in wax or on a papyrus), additional cover data were sent under the wax or written with invisible ink. In the computer era, digital steganographic methods have been proposed. In images, hidden content can be written, e.g., in the least significant bits of pixels (the so-called LSB technique) [3] or JPEG coefficients [4]. Additionally, modern steganography extends to manipulating audio signals [5] and utilizing network protocols to create covert communication channels [6].

In the example with the *zurek* recipe, an LLM model was used as a carrier (instead of a wax tablet), the data generated by the LLM in the normal mode were the overt data, and the soup recipe was the covert (hidden) data. To retrieve it, the LLM should be prompted with a special sequence, the so-called trigger. Drawing inspiration from the Dune universe [7], the concept of *zurek steganography* mirrors the Distrans technology, where information could be implanted in animals for covert storage and retrieval. The embedded messages were activated by specific words or phrases and then audibly repeated by the animal in a distinct manner that combined words with a series of hums. Zurek steganography also bears a resemblance to a method used for fingerprinting LLM models to secure intellectual property licensing [8]. In this case, the fingerprint is hidden text that can be revealed with the proper trigger, known only to the owner of the LLM license.

Most likely, using an LLM to transfer a soup recipe secretly is not a major security incident, but we can imagine scenarios in which confidential data could be leaked from an organization through an innocently looking LLM model. This vulnerability not only poses a risk of data exfiltration but also creates an opportunity for bad actors to communicate covertly within the organization under the cover of routine LLM model development. Such scenarios are especially threatening if we consider that, at this point, there is no established method to extract hidden text from an LLM.

Steganography differs from cryptography because the latter method makes it evident that it hides data: as a matter of fact, the whole communication channel is then hidden. In steganography, an ordinary user will see nothing but the overt content: an inscription on the wax tablet or papyrus, or an LLM model trained to perform a specialized task. When describing zurek steganography, we aim to highlight the significant risks associated with LLMs, specifically their potential to establish hidden communication channels and facilitate data leakage. We cannot rule out the fact that LLMs have already been used for that purpose. Therefore, the research community should focus on developing effective detection methods to determine if LLMs contain any hidden content, for example, through techniques similar to those used for the extraction of LLM (pre-)training data [9, 10].

References

[1] Stanford HAI: 2024 AI Index Report: Applications of Large Language Models Across Industries. Stanford University (2024)

[2] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-T., Rocktäschel, T., *et al.*: Retrieval-augmented generation for knowledge-intensive NLP tasks. In: Proceedings of the 34th International Conference on Neural Information Processing Systems (2020)

[3] Schyndel, R.G., Tirkel, A.Z., Osborne, C.F.: A digital watermark. Proc. 1st International Conference on Image Processing **2**, 86–902 (1994)

[4] Płachta, M., Krzemień, M., Szczypiorski, K., Janicki, A.: Detection of image steganography using deep learning and ensemble classifiers. Electronics **11**(10) (2022) https://doi.org/10.3390/electronics11101565

[5] Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Information hiding-a survey. Proceedings of the IEEE **87**(7), 1062–1078 (1999) https://doi.org/10.1109/5.771065

[6] Mazurczyk, W., Wendzel, S., Zander, S., Houmansadr, A., Szczypiorski, K.: Information Hiding in Communication Networks: Fundamentals, Mechanisms, Applications, and Countermeasures. Wiley-IEEE Press (2016)

[7] Herbert, F.: Dune Messiah. Putnam, New York (1969)

[8] Xu, J., Wang, F., Ma, M.D., Koh, P.W., Xiao, C., Chen, M.: Instructional fingerprinting of large language models. arXiv preprint arXiv:2401.12255 (2024)

[9] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., *et al.*: Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21), pp. 2633–2650 (2021)

[10] Nasr, M., Carlini, N., Hayase, J., Jagielski, M., Cooper, A.F., Ippolito, D., Choquette-Choo, C.A., Wallace, E., Tramèr, F., Lee, K.: Scalable extraction of training data from (production) language models. arXiv preprint arXiv:2311.17035 (2023)