






Aletheia: an open-source toolbox for steganalysis

Daniel Lerch-Hostalot ^{1,2,3} and David Megías ^{1,2,3}

1 Internet Interdisciplinary Institute (IN3), Barcelona, Spain 2 Universitat Oberta de Catalunya (UOC), Barcelona, Spain 3 CYBERCAT-Center for Cybersecurity Research of Catalonia, Barcelona, Spain

DOI: [10.21105/joss.05982](https://doi.org/10.21105/joss.05982)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Marcel Stimberg](#)  

Reviewers:

- [@YassineYousfi](#)
- [@ragibson](#)

Submitted: 12 July 2023

Published: 12 January 2024

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Steganalysis is the practice of detecting the presence of hidden information within digital media, such as images, audio, or video. It involves analyzing the media for signs of steganography, which is the technique used to conceal information within the carrier file. Steganalysis techniques can include statistical analysis, visual inspection, and machine learning algorithms to uncover the hidden data. The goal of steganalysis is to determine whether a file contains covert information and potentially identify the steganographic method used.

Steganalysis has become increasingly important in the face of rising spying and stegomware threats, particularly in the context of data exfiltration. In this scenario, malicious actors leverage steganographic techniques to conceal sensitive data within innocent-looking files, evading traditional security measures. By detecting and analyzing such covert communication channels, steganalysis helps to identify and prevent data exfiltration attempts, safeguarding critical information from falling into the wrong hands.

In recent years, there has been a significant growth in the interest of researchers towards the field of steganalysis. The application of deep learning ([Boroumand et al., 2019](#); [Yousfi et al., 2020](#)) in steganalysis has opened up new avenues for research, leading to improved detection rates and enhanced accuracy. As the field continues to evolve, experts are actively exploring novel architectures and training methodologies to further refine the performance of deep learning-based steganalysis.

Statement of need

Aletheia addresses two main needs. Firstly, it aims to provide specialized analysts with a tool that implements modern steganalysis algorithms, leveraging deep learning techniques. These algorithms are designed to effectively handle even the most advanced steganography techniques. Secondly, Aletheia serves as a valuable tool for researchers by simplifying the process of conducting experiments and comparing methods. It includes simulators for common algorithms ([Hetzl & Mutzel, 2005](#); [Provos, 2001](#); [Sharp, 2001](#)) as well as state-of-the-art steganography methods ([Fridrich et al., 2007](#); [Guo et al., 2014](#); [Holub et al., 2014](#); [Li et al., 2014](#); [Zhang et al., 2019](#)), enabling researchers to prepare and evaluate their work efficiently.

On the other hand, to the best of the authors' knowledge, Aletheia stands out as the sole steganalysis tool currently available that incorporates the latest detection techniques ([Lerch-Hostalot & Megías, 2019](#); [Megías & Lerch-Hostalot, 2023](#)) specifically designed to address the challenges posed by Cover Source Mismatch (CSM) in real-world steganalysis scenarios ([Ker et al., 2013](#)). This capability is particularly significant for conducting effective steganalysis in practical applications.

Description

Aletheia incorporates various image steganography simulators, as well as tools for preparing datasets with different payload sizes using these simulators. This enables researchers to prepare experiments for their articles. Therefore, having access to the original implementations of the different simulators is relevant. Since it is common for these implementations to be developed in Matlab, Aletheia includes several of these simulators in its original code, slightly modified to be executed using Octave. Frequently these simulators have a different license that can be incompatible with the MIT license used by Aletheia. For this reason, this code is in an external repository and is downloaded separately after a confirmation of the user. Aletheia also implements other simulators directly in Python, the programming language of Aletheia, as well as tools that directly utilize their binaries.

These simulators can be used to conduct experiments, as shown in the following example. Here, you can observe how the simulator uses the HILL algorithm and embeds a random payload, which ranges from 5% to 25% of the image's maximum capacity when hiding 1 bit per pixel, within images sourced from the "images" folder. The resulting data is then saved in the "experiment" folder.

```
./aletheia.py hill-color-sim images 0.05-0.25 experiment
```

Although Aletheia allows for the preparation of experiments using multiple simulators, its primary objective is steganalysis. This is achieved through the implementation of various structural attacks on LSB replacement, as well as employing deep learning techniques with models optimized for a vast range of steganography algorithms. These algorithms include both commonly used tools in the real world and state-of-the-art steganographic methods.

Aletheia also offers automated tools that allow for a preliminary analysis, greatly aiding the investigation of the steganalyst. For example, the automated analysis below showcases the modeled probabilities of each image being generated using various steganographic methods.

```
$ ./aletheia.py auto actors/A2/
```

	Outguess	Steghide	nsF5	J-UNIWARD *

2.jpg	[1.0]	[1.0]	[0.9]	0.3
4.jpg	[1.0]	[1.0]	[0.7]	0.3
10.jpg	0.0	[1.0]	0.3	0.2
6.jpg	0.0	[1.0]	0.1	0.0
7.jpg	[1.0]	[1.0]	0.3	0.1
8.jpg	0.0	[1.0]	0.1	0.2
9.jpg	[0.8]	[1.0]	[0.7]	0.1
1.jpg	[1.0]	[1.0]	[0.8]	0.1
3.jpg	[1.0]	[1.0]	[1.0]	0.3
5.jpg	0.0	0.1	[0.7]	[0.6]

* Probability of steganographic content using the indicated method.

Aletheia offers many other functionalities for steganalysis that are not covered in this article and can be found in Aletheia's documentation (github.com/daniellerch/aletheia). Some examples include calibration attacks, custom model preparation, high-pass filters, image difference analysis using pixels and DCT coefficients, DCI techniques to deal with Cover Source Mismatch (CSM), and more.

Acknowledgements

We acknowledge the funding obtained by the Detection of fake news on Social Media pLATfoRms (DISSIMILAR) project from the EIG CONCERT-Japan with grant PCI2020-120689-2 (Government of Spain), and to the PID2021-125962OB-C31 “SECURING” project granted by the Spanish Ministry of Science and Innovation. We wish to express our sincere gratitude towards NVIDIA Corporation for their generous donation of an NVIDIA TITAN Xp GPU card, which has been instrumental in the training of our models.

References

- Boroumand, M., Chen, M., & Fridrich, J. (2019). Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5), 1181–1193. <https://doi.org/10.1109/TIFS.2018.2871749>
- Fridrich, J., Pevný, T., & Kodovský, J. (2007). Statistically undetectable JPEG steganography: Dead ends challenges, and opportunities. *Proceedings of the 9th Workshop on Multimedia & Security*, 3–14. <https://doi.org/10.1145/1288869.1288872>
- Guo, L., Ni, J., & Shi, Y. Q. (2014). Uniform embedding for efficient JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 9(5), 814–825. <https://doi.org/10.1109/TIFS.2014.2312817>
- Hetzl, S., & Mutzel, P. (2005). A graph-theoretic approach to steganography. In J. Dittmann, S. Katzenbeisser, & A. Uhl (Eds.), *Communications and multimedia security* (pp. 119–128). Springer Berlin Heidelberg. https://doi.org/10.1007/11552055_12
- Holub, V., Fridrich, J., & Denemark, T. (2014). Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1), 1–13. <https://doi.org/10.1186/1687-417X-2014-1>
- Ker, A. D., Bas, P., Böhme, R., Cogranne, R., Craver, S., Filler, T., Fridrich, J., & Pevný, T. (2013). Moving steganography and steganalysis from the laboratory into the real world. *Proceedings of the First ACM Workshop on Information Hiding and Multimedia Security*, 45–58. <https://doi.org/10.1145/2482513.2482965>
- Lerch-Hostalot, D., & Megías, D. (2019). Detection of classifier inconsistencies in image steganalysis. *Proceedings of the ACM Workshop on Information Hiding and Multimedia Security*, 222–229. <https://doi.org/10.1145/3335203.3335738>
- Li, B., Wang, M., Huang, J., & Li, X. (2014). A new cost function for spatial image steganography. *2014 IEEE International Conference on Image Processing (ICIP)*, 4206–4210. <https://doi.org/10.1109/icip.2014.7025854>
- Megías, D., & Lerch-Hostalot, D. (2023). Subsequent embedding in targeted image steganalysis: Theoretical framework and practical applications. *IEEE Transactions on Dependable and Secure Computing*, 20(2), 1403–1421. <https://doi.org/10.1109/tdsc.2022.3154967>
- Provos, N. (2001). Defending against statistical steganalysis. *10th USENIX Security Symposium (USENIX Security 01)*, 1–13. <https://www.usenix.org/conference/10th-usenix-security-symposium/defending-against-statistical-steganalysis>
- Sharp, T. (2001). An implementation of key-based digital signal steganography. In *Information hiding* (Vol. 2137, pp. 13–26). Springer. https://doi.org/10.1007/3-540-45496-9_2
- Yousfi, Y., Butora, J., Khvedchenya, E., & Fridrich, J. (2020). ImageNet pre-trained CNNs for JPEG steganalysis. *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–6. <https://doi.org/10.1109/WIFS49906.2020.9360897>

Zhang, K. A., Cuesta-Infante, A., Xu, L., & Veeramachaneni, K. (2019). *SteganoGAN: High capacity image steganography with GANs*. <https://arxiv.org/abs/1901.03892>