

watermarks & auxiliary signals in log files

collicutt.net

# STEGALOG GRAPHER

curtis collicutt

They said **never** write your own encryption.  
They didn't say anything about steganography.

a guidebook

---

# Stegalographer

---

*They said never write your own encryption.  
They didn't say anything about steganography.*

*By Curtis Collicutt*

# Table of Contents

<b>Introduction to this Guidebook</b>	<b>4</b>
Overview	4
What is a Guidebook?	4
Caveats and Notices	5
<b>Introduction to Steganography</b>	<b>7</b>
What is Steganography?	7
Steganography Events in History	7
Applications of Steganography	10
Steganographic Pipelines	11
Auxiliary Signal Encoding	11
Asymmetry of Awareness	12
Duality of Steganography	12
Deniability	13
Steganography vs Encryption	14
They Both Have “Keys”	15
Using Steganography and Encryption Together	16
Example Custom Tools	17
Auxiliary Signals in Image Files	17
Auxiliary Signals in Binary Files	20
Auxiliary Signals in Sentences	21
Steganographic Mediums	21
Information Theory	22
Steganalysis	23
Philosophy and Ethics	28
Democratic Infrastructure	28
<b>Stegalographer</b>	<b>32</b>
The Core Scenario	33
Scenarios	35
Scenario: Insider Log Exfiltration Detection via Continuous Watermarking	36
Scenario: Journalist Communication	37
Technology	37
Features	38
How It Works in Practice	38
Cover Modeling	39
Built-in Models	39
Custom Models	40
Model Binary File	41
Log Analysis and Template Extraction	42

Models as a Shared Grammar	42
Stegalographer “Key”	43
Key Format	43
Example Method	46
Generate Mode	46
Stream Mode	47
Execution Model	47
Foundational Algorithms	48
Operational Considerations	50
Interactions with Real Log Pipelines	50
Encoding Cost	50
Robustness, Detection, & Steganalysis	51
Statistical Validation & SciPy	52
Statistical Tests Performed by the Validator	52
Current Statistical Validation Status	53
Future Improvements	53
<b>Using Stegalographer</b>	<b>56</b>
Encoding and Decoding a Message	56
What happens Step by Step	56
Decoding	59
Creating a Custom Model with Analyze	60
<b>Conclusion</b>	<b>64</b>
Future Work	65
Stegalographer	65
This Guidebook	66
<b>End Matter</b>	<b>68</b>
About the Guidebook Author	68
Building & Researching with AI-Assisted Coding Tools	68
Glossary	71
Change Management	72

# Introduction to this Guidebook

## Overview

Steganography is the art of hiding in plain sight. Unlike encryption, which scrambles a message into ciphertext, steganography conceals the very existence of communication. Stegalographer, a specific software project, is an application of steganography, and it applies this principle to one of the most ubiquitous and overlooked artifacts in computing: log files. Every server, every application, every system generates logs. They scroll past endlessly, typically examined only when something breaks. They are infrastructure noise, like background radiation, the exhaust of running software. And that makes them the perfect hiding place.

Stegalographer places a difficult to discover auxiliary signal, a message or messages, inside legitimate-looking log entries. The output is indistinguishable from real application logs because it is real log structure. Your data is encoded not in the content of the logs, but in the choices: which log patterns appear, how timestamps vary by microseconds, which synonyms are selected, where whitespace falls. The message dissolves into decisions that no one thinks to question.

Importantly, steganography has multiple applications, including not only the encoding of auxiliary signals, but also concepts such as digital watermarking. These applications are wide-ranging, so we should consider not only the placement of messages, but also tampering protections, watermarking and other steganographic capabilities, and the Stegalographer project can also be used in these situations.

This guidebook documents the Stegalographer project in full. It explains not just how to use the tool, but how it works: the algorithms that parse log structure, the methods that encode bits into formatting choices, the tradeoffs between capacity and stealth. Whether you need to understand Stegalographer for authorized security testing, covert communication research, or simply to satisfy your curiosity about what is possible when you look at familiar things from an adversarial angle, this guide will take you there.

## What is a Guidebook?

This document, a “guidebook”, is the artifact of a learning journey. It documents the process of exploring steganography, covering its history, the underlying computer science and philosophy, and the practical challenges involved in developing a steganographic application such as Stegalographer. It is not an in-depth or rigorous academic study as it is not peer-reviewed and will almost certainly contain mistakes. Consider it a companion to the Stegalographer project itself, perhaps just a big README, as well as possibly a starting point for your own reading and research.

Rather than being exhaustive, it is intended to provide a guided overview of the application and its computing and computer science foundations, notably from an amateur perspective. To some extent, it is based on the idea that “you can just do things”, and that one way to learn about something is to build it and while doing so explore its fundamental makeup.

## Caveats and Notices

- *Stegalographer is an imperfect tool and, by its very nature, can never be perfect. In its current form, it is an experiment. It should not be used in real-world environments without further testing to validate and improve its robustness.*
- *This guidebook is a living document. I have done my best to footnote all the external content I have used, but ultimately this should be considered as a more in-depth project README, and not a peer-reviewed paper.*