Steganography (”steg”) is the practice of concealing information within other, more innocuous piece of information. Such techniques allow for both a mild form of encryption — though, it is necessarily noted that the fundamental concepts of steg, inherently, does not provide adequate encryption — and the deniability for having sent a piece of information.

A majority of the steg systems is centered around the hiding of data inside images because of the fact that subtle sub-pixel alterations could be made to an image without influencing the general shape or design of the image.

The actual act of mixing data within images is not practically difficult. However, the process of decoding — once received — which of the mixed assortment of data contained the modified bits is much more difficult. It is additionally difficult to perform the act highlighted above without betraying that an image has indeed be modified.

1 | Common Stegnography Techniques

1.1 | Lowest Noisy Bit

In a reasonably noisy image, one it is possible to identify the lowest bits in the image (black spots) and alter them to encode information. As long as the darkest bits remain dark, one could simply switch their color completely and use, for instance, color distance from black, as the encoded information.

1.2 | Chaffing and Winnowing

Chaffing and Winnowing is a transmission technique that leverages traditional authentication techniques to perform stenography. With every encoded message, an arbitrary amount of miscellaneous bits are intermixed with failing check bits. Via simple async-key encryption, only the intended recipient who would be capable of verifying the check bits would be able to figure the intended bits.

1.3 | BPCS Steg

Leveraging the visual property that large, complexly ragged areas is usually perceived as somewhat uniform (i.e. a beach etc.), BPCS steg identifies and swaps these complex or noisy areas in a specific way — mostly by simplifying it into less complex geometries — and uses the shades of those to encode information

2 | Threat Model Relating to Steganography

2.1 | Specifically, what are we protecting?

We are protecting a generic, trusted public communications system against stenography. Specifically, we are aiming to protect against the main goals of stenography: inserting additional information into existing data and hiding the marks of tampering the existing data.

2.2 | Who are we protecting it from, and what are their motivations?

We are protecting the data against small groups of activism or corporate espionage: which typically is the use case of stenography. However, we do not aim to protect against nation-states or large-scale attacks as it is only practical to protect against variations of common stenography techniques. Furthermore, nation-states would have further (and likely more ”legal”) processes in place to gain access to privileged information already.

2.3 | What methods of attack do we prevent? What are types of attacks that we don’t prevent?

For this threat model specifically, we aim to protect against most common steganography techniques. Any processes independent of information extraction and obfuscation is out of scope for this threat model. See above for examples of common stenography techniques.

2.4 | What are the possible effects of these attacks?

The unauthorized extraction or publication of information is the most major consequence of steganographical attacks. The sender — in well-designed steg systems — would additionally have the capability to exhibit deniability for leaking the information.

2.5 | What are the resources of attackers?

Current steg systems are mostly open-source and widely available. As such, given we are protecting against common techniques (or variations thereof), we could limit ourselves to the resources that someone without a background in information theory could develop or use in a reasonable amount time. Such systems often come in the form of ”binaries” which attackers could simply download and apply; other techniques could be implemented easily manually.

2.6 | What are our resources?

Given we are assuming that the attacker’s resources are mostly open source and publically available, we have the ability to reverse-engineer (or similar reverse-engineering tools already exist) to subvert the commonly-existing techniques. The only consideration is to prevent the existence of variants or otherwise manual-implemented ”hacks”/changes which subvert typicaly detection systems.

2.7 | What should we do?

It is first important to collect a large sample of stenographical techniques which maybe used by attackers. When information is being passed, check for stenographical signatures that would be typical to the common techniques. Were there potential signs, it is possible to leverage existing reverse-engineering tools to extract the hidden information. Were the technique used be a variant of an existing technique, there likely is only small changes to the data structure which brute-force methods likely will provide an efficient mechanism to discover the changes.