Steganography:

an Overview of Techniques, Applications, and Detection

Devorah Sachs

Excelsior College

IT361: Operating Systems and Organization

Simran Kaur Dhillon

October 24, 2021

Contents

Steganography: an Overview of Techniques, Applications, and Detection

A gentleman in ancient Greece wanted to send a secret message. He reportedly tattooed it on the shaven head of his slave, waited for his hair to grow back and sent him to his destination, where his head could be shaven again to reveal the message. This is an early instance of steganography, which comes from the Greek words for "covered" and "writing" (Kose et al., 2020) and is essentially the art of hiding information in plain sight[1]. As with the use of invisible ink, steganography implants data in an innocent looking carrier object so that it isn't seen or suspected by viewers other than the message's recipients. In the digital age, steganography takes advantage of the properties of computerized files to encode data so that its presence is not visible to the human eye. New technologies continue to expand the range of steganography techniques and applications, as well as methods for detecting steganography and securing it against detection.

In digital steganography, a "hidden signal" (some text or a file to be kept hidden) is embedded inside a "host signal", also called the "cover file" (Kose et al., 2020) (Geleta et al., 2021). The resulting container signal is called the "stego file". Casual observers of the stego file will notice only the cover file's characteristics; they won't know that there's other data secretly being passed along. The receiver of the stego file generally has some pre-shared key, which he can use to extract the hidden message (Kose et al., 2020).

[1] I did not come up with this phrase on my own – I came across it in my research, but regrettably do not remember where.

Steganography Techniques:

LSB Embedding:

One example which illuminates this process is one of the oldest, simplest methods: Least Significant Bit, or LSB embedding. In LSB, the hidden signal is translated into bits. Each of those bits is stored in the least significant (right-most) bit of one of the bytes that makes up the image. Since least significant bits have so little effect on the value of the bytes they belong to, changing them, even many of them, does not have a very visible effect on the image (Powell, 2020). In fact, if one never releases the original cover image, it's unlikely that an unaided human eye will have any reason to suspect that the image has been modified. Dr. Mike Pound of Nottingham University gave a presentation where he showed first a picture of a tree, then that same picture with all the literature of Shakespeare embedded in its LSB's (he used the *two* least significant bits of the image's bytes). Side by side, the images looked basically identical (Computerphile[2], 2015) (Zhang, 2015).

Use of DCT's in JPEG images:

Another, more sophisticated approach to steganography is used when embedding data into JPEGs. The compression process of JPEG images involves several stages of mathematical calculation to prepare the image for storage as numeric code. Part of that process uses Discrete Cosine Transform coefficients (DCTs), which are the coefficients of the image's cosine waves. The DCT's are arranged in a chart and then quantized to further simplify the numeric representation of the image. (Computerphile, 2015, May 22). Many of the DCTs end up as 1's or 0's but some are a bit larger. The embedder looks for the higher number coefficients and

[2] The Computerphile content used is given over by Dr. Mike Pound of Nottingham University.

implants the hidden signal into their LSB's. (Computerphile, 2015, Aug 4) Hiding data in the least significant bits of DCTs instead of in the actual image's least significant bits leads to less image distortion and can be harder to detect. The tradeoff, however, is that you can't fit in as much information (ibid) (Walia et al., 2010).

Audio:

Data can also be hidden in audio files. Different steganography programs use different techniques to transfer data secretly through sound. One graphical editor can convert images into sound using the sine waves that correspond to pixel characteristics (Kose et al., 2020). At the other end, the sound can be analyzed with a spectrogram that renders the sound as an image – along the secret message stored inside (Kose et al., 2020).

Scripts:

Messages can be hidden in actual text with surprisingly simple methods that rely on the human tolerance for a bit of error. HTML (or other scripts) can be loaded with a message simply by inserting unnecessary spaces in strategic tags and leaving others without extra spaces. The presence of a space could indicate a one, the absence of a space could indicate a zero (Chang et al., 2013). A program called StegParty hides data in text with irregularities that go unnoticed because the human brain naturally doesn't give much consequence to little typos or grammar mistakes (Kose et al., 2020).

More complicated text-based steganography techniques include Irongeek's Unisteg. There are Unicode characters that look the same to people but have different numerical values.

Unisteg manipulates these differences to hide data in cover text, producing a stego text that looks the same – but isn't (Kose et al., 2020)[3].

Network Packets:

Information can be hidden in different fields of the packet headers required by TCP/IP protocols. One method involves "spoofing the sender's IP address to the recipient's address so that when the sender transmits an encoded message, it will get sent directly to the receiver without undergoing an SYN/ACK process…". Network packets have optional fields. Sometimes data is hidden in these (but this may be less discreet than using a mandatory field) (Kose et al., 2020).

Deep Learning:

New steganography techniques are getting more and more complex. A recent study reported that steganography researchers are using deep learning/deep neural networks to find the best places to hide information in a cover file – spread all over the file's bits instead of just in its LSB's. The study's authors were working on a way to embed images in sound through deep learning for a variety of purposes, including offering accessibility options for hearing impaired individuals in airports (Geleta et al., 2021).

Steganalysis:

The flipside of steganography is steganalysis – detecting the presence of hidden information (Wang & Wang, 2004). Since there are multiple methods for embedding data,

[3] Also see link to Irongeek (referenced by Kose et al.) in references for the source of part of the explanation above and source code.

there's no one reliable way to sniff out that a file has data hidden in it. Of course, one who possesses both the original cover file and the stego file can use programs like Notepad or hex editors (depending on the file) to comb the two files' properties for differences (Raggo, n.d.). Generally, however, only the stego file is available. Steganalysis experts develop an understanding of what embedding does to a file and can spot patterns or anomalies that tell them the file has been modified.

In images, steganalysis often involves a search for statistical irregularities caused by embedded content (Wang & Wang, 2004). In direct LSB for instance, the statistics of pixel values will be different even if the cover image and stego image look the same (Powell, 2020). These differences can be seen with a histogram analysis. In DCT embedding, since there's a specific mathematical procedure to compress JPEGs, there are mathematical structures that typical JPEGs tend to have. If these are modified, that tells you there's been embedding (Wang & Wang, 2004).

Of course, those interested in keeping their steganography under the radar work to develop techniques that will avoid detection by steganalysis. Brian Powell (2020) published a technique he developed which involved bringing the pixel statistics of an image into an erratic state *before* embedding any data. The "messing up" is precisely arranged so that embedding actually brings the statistics back to normal. This way an image that had a payload could escape detection by some of the common LSB steganalysis attacks.

[Note: There is at least one program that can run through a file and spot the footprints of several different steganography programs. Such a program is advantageous because if it recognizes steganography, since it knows which program embedded it, it can likely also decode

the hidden information. (The disadvantage is that it only finds a payload if it was embedded using the programs it recognizes.) (Raggo, n.d.).]

Applications:

Secrecy and Security:

Steganography and steganalysis have several interesting applications. The most obvious, of course is in communicating data secretly. Steganography can be used to add security to encryption (encrypt the data before you embed it (Computerphile, 2015, Aug 4). Of course, the ability to hide files in other files provides fertile ground for threat actors. Malware concealed inside innocent files can execute when they're opened or downloaded, infecting the unsuspecting user's computer, and possibly opening a backdoor for an attacker.

The *United States Cybersecurity Magazine* website reported that in recent times the cybersecurity community has been hearing more and more about malicious actors taking advantage of steganography. In March of 2021, a security researcher named David Buchanan publicized that he had discovered a way to embed ZIP archives and MP3 files in Twitter's PNG images in a way that bypassed Twitter's security filters. The potential security ramifications of this discovery of a vulnerability in such a giant, widely used social media site are worrying. Threat actors can also use steganography to conceal some of their activity – like hackers of the Magecart variety, who hid their stolen credit card data in JPG images (Maraj, 2021). These incidents emphasize the importance of robust steganalysis techniques in preventing hackers from using steganography to their advantage.

Fake News Detection:

Steganography's usefulness extends beyond security. Fake news is a thorn in the side of media and social media consumers worldwide. There are ways to detect falsification by checking whether photos have been modified. However, untampered images that escape detection by these traditional methods can be quoted out of context to twist what happened and mislead an audience. A group recently developed steganography-based architecture called NIS, which hides summarization data in an image, which can be used to prove where the photo truly originated and what situation it truly captured (Zhou, 2021).

Digital Watermarking:

Watermarking is a close cousin to steganography that is built on the same principles. Owners of various media/multimedia files (like photos, audio, etc.) can embed information about the source or owner of the image. This hidden stamp can be used to protect against copywrite violation (Wang & Wang, 2004), prove the owner of a file, and prove the authenticity of a file.

Conclusion:

Digital steganography is one example of the power and opportunities available in the world of bits and bytes. Techniques in steganography and steganalysis range from the simple to the very complex and provide various useful solutions to problems in the fields of hiding, communicating, protecting, and identifying information. The serious security threats made possible by these techniques behoove all data consumers, especially those involved in security, to develop a working understanding (or at least a basic awareness) of the art and science of steganography.

References

Kose, J., Chia, O.B., & Baboolal, V. (2020). *Review and Test of Steganography Techniques.* (I)arXiv: 2012.08460[cs.CR] http://arxiv.org/abs/2012.08460

Geleta, M., Punti., C., McGuiness, K., Pons, J., Canton, C., & Giro-i-Nieto, X. (2021). *PixInWav: residual steganography for hiding pixels in audio.* arXiv(I):2106.09814[cs.MM] http://arxiv.org/abs/2106.09814

Powell, B. (2020). *Securing LSB embedding against structural steganalysis.* Journal of computer security *0 (2021)* 1-0. arXiv:2003.03658v2 [cs.CR] http://arxiv.org/abs/2003.03658

Computerphile.[Given by Dr. Mike Pound]. (2015, Aug. 4). *Secrets hidden in images (steganography) – computerphile*. Youtube.com [Video]. https://www.youtube.com/watch?v=TWEXCYQKyDc

Computerphile. [Given by Dr. Mike Pound]. (2015, May 22). *JPEG DCT, Discrete Cosine Transform (JPEG Pt2)- Computerphile.* youtube.com [Video]. https://www.youtube.com/watch?v=Q2aEzeMDHMA

Zhang, M. (2015, August 7). A Look at Photo Steganography, the Hiding of Secrets Inside Digital Images. petapixel.com.

https://petapixel.com/2015/08/07/a-look-at-photo-steganography-the-hiding-of-secrets-inside-digital-images/

Walia, E., Jain, P. & Navdeep. (2010). An analysis of LSB & DCT based steganography. Global Journal of Computer Science and Technology vol 10, issue 1 (ver. 1.0)

https://www.researchgate.net/publication/265032627_An_analysis_of_LSB_DCT_based_steganography

Chang C-C, Lin C-C, & Yang C-N. (2013) (I)*Steganography and Watermarking.* Nova Science Publishers, Inc. https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=591935&site=eds-live&scope=site

Wang, H., & and Wang, S. (2004). *Cyber warfare: steganography vs. steganalysis.* Communications of the ACM, *47*(10), 76-82. http://vlib.excelsior.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=14523129&site=eds-live&scope=site

Zhou, J., Pun, C.-M., & Tong, Y. (2021). *News Image Steganography: a novel architecture facilitates the fake news identification.* https://doi.org/10.1109/VCIP49819.2020.9301846

Raggo, M.T. (n.d.) *Steganography, Steganalysis, and Cryptanalysis.* https://www.blackhat.com/presentations/bh-usa-04/bh-us-04-raggo/bh-us-04-raggo-up.pdf

Maraj, S. (2021) *Steganography techniques raise security fears.* Uscybersecurity.net. https://www.uscybersecurity.net/cyberNews/steganography-raises-security-fears/

Link to Irongeek website (for explanation and example code): http://www.irongeek.com/i.php?page=security/unicode-and-lsb-stego-code