Survey on Steganography Techniques in Images

Jibri Ward
Department of Computer Science
Hampton University
jibri.ward@my.hamptonu.edu

Chutima Boonthum-Denecke
Department of Computer Science
Hampton University
chutima.boonthum@hamptonu.edu

ABSTRACT

This paper will briefly address cryptography as a solution to the problem of privacy of information. Furthermore, this paper will examine the use of digital steganography as a solution to the problem. Different forms of steganography will be compared, and will be examined with the combination of cryptography. The use of this study will ultimately aid in the conversation of the practical use of digital steganography.

CCS CONCEPTS

• **Security and privacy** → **Cryptography** → Cryptanalysis

KEYWORDS

Cryptography, Steganography, Image Encryption, Least Significant Bit, Discrete Cosine Transform

ACM Reference format:
J. Ward, C. Boonthum-Denecke. 2017. Survey on Steganography Techniques in Images. In *Proceedings of 2017 ADMI Symposium*, Virginia Beach, Virginia USA, March 23-26, 2017.

1 INTRODUCTION

With the massive expansion of the Internet over the past few decades have come an unlimited number of smart devices and communication protocols that allow them to interact with each other. Formally referred to as the Internet of Things, these amazing advancements in technology have niched their way into almost every aspect of our lives. They allow us to do everything from communicate with one another to purchase our favorite pair of shoes. However, these forever-growing advancements have provided, and will continue to provide, hackers with a forever-growing number of ways to manipulate computer devices and the protocols that govern them. This especially provides concern to the confidentiality of sensitive information that is transmitted over the Internet. This sensitive information, and any other information, can be intercepted without the correct protection.

2 CRYPTOGRAPHY

Cryptography, which is the process of encrypting and decrypting data so that it is not legible in its transmission stage, provides an excellent method to securing the transmission of information [11]. This process is completed by encrypting plaintext into an unintelligible form, called ciphertext that cannot be read without decryption [8]. The recipient must have the correct decryption key to view the message, which makes cryptography extremely useful for communicating over an untrusted network such as the public internet.

There are two main encryption methods associated with cryptography; symmetric encryption and asymmetric encryption. Symmetric encryption uses one key to encipher and decipher a message. This is the most common form of encryption because of its efficiency [8]. It provides quick execution by even small computers. The most pressing challenge is transporting the key to the recipient without it being intercepted. Asymmetric encryption uses two keys; one to encipher a message and the other to decipher it. The two keys are interchangeable, but both must be used to encipher and decipher the message. This method is more secure than symmetric encryption [8]. However, it can take up to ten thousand times longer than a symmetric encryption. It is often seen that cryptosystems are starting to implement a combination of both algorithms [8]. These methods are referred to as hybrid cryptography systems.

There are only two ways to crack an encryption. The first is by using a brute force attack to generate the key. This method is insufficient as today's encryption standard makes it impossible to crack an encryption with a brute force attack in reasonable time [11]. The second way to crack an encryption is through a process called cryptanalysis, which requires the knowledge of the encryption algorithm, the ciphertext, and a general concept of the encrypted message [11]. This method is difficult to complete with the possession of all the knowledge listed above. If only the ciphertext is known, which is often the case, and then the difficulty is multiplied. There are encryption algorithms that are so mathematically concrete that the time wasted attempting to decrypt the information using a brute force attack outweighs the value of the information; and most cryptanalysis use a brute force approach making it time consuming as well. The encryption standard *Advanced Encryption Algorithm* (AES) utilizes an algorithm with a key and cipher up to 256-bits [11]. This is important because experts estimate that it would take the most specialized and powerful computers quintillions of years to crack an AES encryption [8].

Many commonly used IT tools utilize AES embedded encryption technologies to protect sensitive information. Web sites use this algorithm in built-in encryption applications to enable secure e-commerce, emailing, and sensitive data transmission [11]. This can be seen in Websites that have the header HTTPS and packet transferring protocols like TLS, and SSL. This is perfect for transmitting information over public networks, but what if the concealment of the existence of the message is more important than the protecting the information? For instance, encrypted data or messages would attract attention to the information, which would cause for someone to attempt to crack the encryption. Furthermore, encrypted data could prove detrimental to an undercover agent if the information is detected. A possible solution to this problem is hiding the data. Steganography provides a solution to this problem as it conceptualizes concealing the existence of a message within another message.

2.2 Steganography

Steganography is defined as the art and science of invisible communication [2]. Throughout history steganography has been used to communicate in secret. Messages were carved into wood, covered in wax, and revealed when the wax melted; messages were tattooed onto the back of servants' heads to be hidden under their hair; and messages were written in invisible ink and only revealed by liquid with a certain acidic level. In modern day, the concept of steganography is being applied to the digital world. Digital steganography is the concept of hiding messages in digital files such as picture, audio, and text format. In theory, any digital file can hold a hidden message with the use of bits, but the most popular uses of steganography involve .jpeg, .bmp, .gif, .mp3, .wav, .doc, and .txt files [2]. This paper will focus on the use of steganography in picture formats.

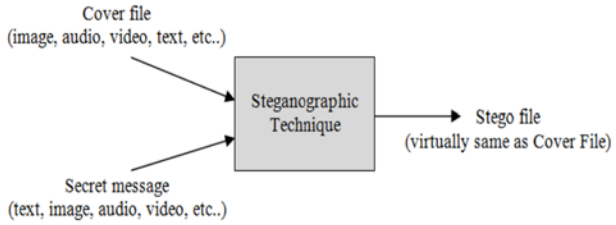

Figure 1. Fundamental scheme of steganography process. [1]

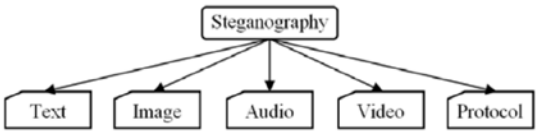

Figure 2. Categories of steganographic cover mediums. [1]

There are many different techniques that can be used to perform steganography with image files. Some of the more popular methods will be addressed in the following paragraphs.

2.2.1 Least Significant Bit

The first approach is to use the *least significant bit* (LSB). This process involves manipulating the last bit of each byte within a pixel to form a message using those bits. A computer recognizes an image as an array of numbers that represent light intensities at different points [2]. These points are the pixels that make up the image. Manipulating the LSB within the pixel makes a minor change in the intensity of the color of the pixel [3]. This change is so miniscule that it is not detectable to the human eye.

The most typical digital images contain 8 or 24 bits per pixel [2]. The differences, as they pertain to steganography, are that 8-bit images have a color pallet of 256 colors and are relatively small in size, while 24-bit images have a color pallet of 16 million colors and are larger in size [2]. A 24-bit image has the advantage over an 8-bit image because it can store a message three times the size of a stored message in an 8-bit image. 24-bit images use 3 bytes per pixel; one byte for red, one byte for blue, and one byte for green. This enables 3 bits, the LSB from each byte, to be stored per pixel. If the 24-bit image is 800 X 600 pixels, then there are 480,000 total pixels, and 1,440,000 bits of information can be hidden [3]. An 8-bit image only uses 1 byte per pixel, which correlates to one LBS per pixel for a message to be hidden. Comparatively, an 800 X 600 pixel 8-bit image can only store 480,000 bits of information. The following grids display 3 pixels of a 24-bit image and an 8-bit image:

24-bit image
(00101101 00011100 11011101)
(10100111 11000101 00001101)
(11010010 10101101 01100011)

8-bit image
(00101101)
(00011100)
(11011101)

Below, the same pixel grid is shown with the number 77, which has the binary representation of 1001101, embedded into the least significant bits of this part of the image and an 8-bit image:

24-bit image
(00101101 00011100 11011100)
(10100111 11000101 00001100)
(11010011 10101101 01100011)

8-bit image
(00101101)
(00011100)
(11011100)

As shown above, only three of the least significant bits needed to be changed within the 24-bit image to embed the number 77. Only one bit needed to be changed in the 8-bit image, but the example did not provide enough pixels to display the entire message within the image. If four more pixels were displayed, then the number 77

would have been display. However, the three LBS shown displays 100, which is the binary notation for the number 4. This provides evidence that an 8-bit image requires more pixels and bytes to hide a message than a 24-bit image.

Even though the size of an 8-bit image restricts the image from holding larger messages, the size of the image provides an advantage over the 24-bit image [3]. This is the case because a 24-bit image with a hidden message may be too large and may be compressed somewhere in route to the recipient. Compression of a digital image occurs by removing excess image data and calculating a close approximation of the original image [2]. In most cases, it is likely that the hidden message will be altered because of the approximation altering the binary color codes of the pixels, which usually affects the LSBs.

Most 8-bit images do not need to be compressed. The LSB approach to steganography is commonly carried out using .gif and .bmp files because they make use of lossless compression techniques, meaning that the compression does not alter the color bytes [2]. Image files such as .jpeg use lossy compression techniques, which alters the unimportant attribute of the image, subsequently the LSBs. Although .gif and .bmp files hold an advantage of lossless compression, they raise more suspicion the more popular image files. GIF stands for Graphic interchange format, and is commonly used for images on the web and sprites in software programs [10]. They are not ideal for storing digital photos [10]. BMP is an outdated and "historic" image file used on older operating systems [9]. Though it is not disadvantageous to use this file, it is very uncommon [9]. JPEG is the one of the most common and suitable image files to date and seeing this format wouldn't raise much suspicion.

2.2.2 Discrete Cosine Transform

Because .jpeg files are affected by lossy compression techniques, it was assumed that steganography would never be possible using this format [3]. The fact the excess data of the file would be removed in compression refutes the process of digital steganography using the LSB approach. However, the properties of the compression algorithm have been exploited in order to develop a steganographic algorithm for JPEG format. JPEG images use the *discrete cosine transform* (DCT) technique to achieve image compression that allow high quality images to be stored in relatively small files [3]. "The compressed data is stored as integers but the calculations for the quantization process require floating point calculations which are rounded" [5]. Errors introduced by this rounding method define the lossy characteristic of the JPEG compression.

When a file or message is embedded into a .jpeg, the relation of the DTC coefficients that are altered. This is the first half of the .jpeg compression process. The second half is the lossless compress referred to as Huffman coding [5]. Huffman coding is a lossless data compression algorithm that assigns variable-length codes to input characters, that bases lengths of the assigned codes on the frequencies of corresponding characters [4]. The LSB insertion method can take place between these two stages. Using these LSB principles the message can be embedded into the least significant bits of the coefficients before applying the Huffman encoding. By embedding the information at this stage, which is the transform domain, it is algorithmically secure and difficult to detect.

$$Original = \begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

$$Decompressed = \begin{bmatrix} 149 & 134 & 119 & 116 & 121 & 126 & 127 & 128 \\ 204 & 168 & 140 & 144 & 155 & 150 & 135 & 125 \\ 253 & 195 & 155 & 166 & 183 & 165 & 131 & 111 \\ 245 & 185 & 148 & 166 & 184 & 160 & 124 & 107 \\ 188 & 149 & 132 & 155 & 172 & 159 & 141 & 136 \\ 132 & 123 & 125 & 143 & 160 & 166 & 168 & 171 \\ 109 & 119 & 126 & 128 & 139 & 158 & 168 & 166 \\ 111 & 127 & 127 & 114 & 118 & 141 & 147 & 135 \end{bmatrix}$$

Figure 3. Sample of DTC Compression on JPEG. [3]

The DTC and quantization process on peppers image.

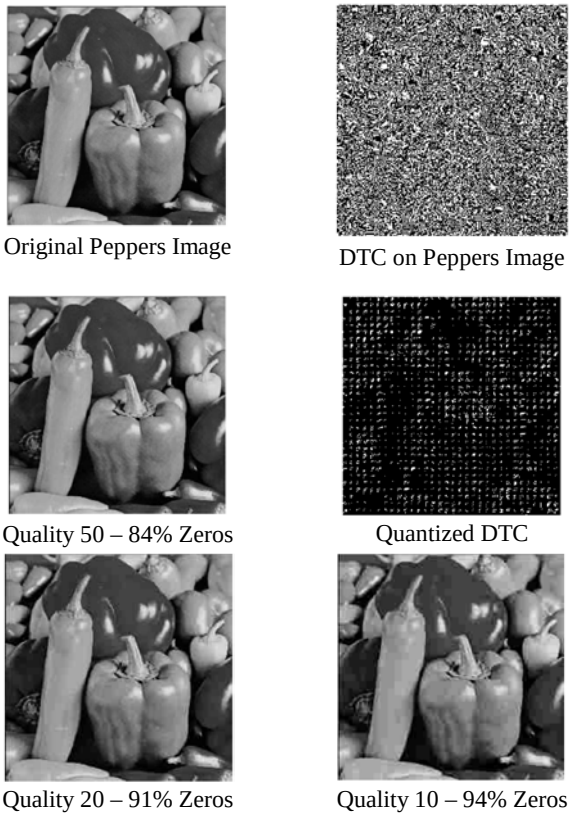

Figure 4. DTC on Peppers Image. [3]

3 COMPARISION

The different steganography discussed will be evaluated with a ranking of low medium and high. The categories for evaluation of steganographic algorithms are as follows:

- **Invisibility** – Imperceptibility is considered the most important requirement of a steganographic method. The strength of steganography depends on the ability to not be detected. The human eye should not be able to detect any obscurity in the cover photo.
- **Robustness** – Determines the strength of the steganographic method against photo manipulation, change, etc. The hidden message should remain intact under most circumstances [1].
- **Payload Capacity** – Describes the size of data that can be embedded into a seemingly innocent cover photo [1].
- **Unsuspiciousness** - This requirement includes all characteristics of a Steganographic algorithm that may result in images that are not used normally and may cause suspicion such as size, format, etc.

Table 1. Evaluation of steganography techniques in images

	Invisibility	Robustness	Payload Capacity	Unsuspiciousness
LSB in BMP	High	Low	High	Low
LSB in GIF	Medium	Low	Medium	Low
DCT Manipulation in JPEG	High	Medium	High	High

4 CONCLUSIONS

Cryptography and steganography are different as cryptography focuses on the encryption of data, while steganography focuses on the concealment of data. There are many methods to use digital steganography to hide data in images. Though this paper only discusses a few methods, the data had drawn from the comparisons shows that no one algorithm is perfect. They all have different strengths and weaknesses that overlap each other. The main conclusion is that they all have weaknesses. The most secure solution is to use a combination of cryptography and steganography to transmit sensitive data. Cryptography and steganography both provide an excellent means to data protection. Though neither is entirely secure and can both be broken. Using the two in conjunction produced a multilayered and powerful defensive approach to transmitting data.

Acknowledgement

This work is partly supported by the National Science Foundation CyberCorps: Scholarship for Service program under grant no. DGE-1303409 (PI/co-PI: Dr. Chutima Boonthum-Denecke, Dr. Jean Muhammad).

REFERENCES

[1] Abdulaleem Z. Al-Othmani, Azizah Abdul Manaf, and Akram M. Zeki. (2012). A Survey on Steganography Techniques in Real Time Audio Signals and Evaluation [Online]. Available: https://pdfs.semanticscholar.org/2ea8/c2c1278d4e301da39f8b20816a7bfa76d442.pdf

[2] Bret Dunbar. (2002). A Detailed Look at Steganography Techniques and their use in an Open-System Environment [Online]. Available: https://www.sans.org/reading-room/whitepapers/covert/detailed-steganographic-techniques-open-systems-environment-677

[3] Cabeen K and P. Gent. Image Compression and the Discrete Cosine Transform. Available. https://www.math.cuhk.edu.hk/~lmlui/dct.pdf

[4] Falesh M. Shelke, Ashwini A. Dongre, Pravin D. Soni. (2014). Comparisons of Different Techniques for Steganography in Images [Online]. Available: http://www.ijaiem.org/volume3issue2/IJAIEM-2014-02-27-062.pdf

[5] Geeks For Geeks. Greedy Algorithms: Set 3 (Huffman Coding) [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSLTBW_2.1.0/com.ibm.zos.v2r1.csfb500/csfb5za206.htm

[6] Inforsyssec. Steganography Techniques [Online]. Available: http://www.infosyssec.com/infosyssec/Steganography/techniques.htm

[7] Indika. (2011) Difference Between Cryptography and Steganography [Online]. Available: http://www.differencebetween.com/difference-between-cryptography-and-vs-steganography/

[8] Michael E. Whitman and Herbert J. Mattord, "Introduction to Information Security" in Principles of Information Security, 5th ed. Boston: Cengage, 2016, pp. 417-427.

[9] Paul Bourke. (1998). BMP Imag Format [Online]. Available: http://paulbourke.net/dataformats/bmp/

[10] TechTerms. (2016) Gif Definition [Online]. Available: https://techterms.com/definition/gif

[11] William Stallings, "Cryptography and network Security" Principles and Practice, 5th ed. Upper Saddle River: Pearson Education, Inc., 2011, pp. 36-57.