Self-sanitization of digital images using steganography

Tayana Morkel
Department of Computer Science
University of Pretoria
South Africa

***Abstract*— Sanitization of an image is a process where certain areas of an image are removed to keep the contents safe from unauthorised viewers. Image sanitization is often required by authorities, for example law enforcement or in legal cases, when the image contains sensitive material that should not be shown to the general public. This paper proposes a system for the self-sanitization of a digital image using information hiding, specifically image steganography, techniques to hide part of the image within the image itself. The proposed self-sanitization system allows for the removal of a specific part of the image and then uses Least Significant Bit (LSB) steganography to embed the sanitized part of the image within the rest of the image, making it unnecessary to store the sanitized and unsanitized versions of the image separately. The self-sanitization system includes a method for reducing the size of the embedded information in an attempt to make the information more difficult to detect. Experimental results show that the proposed self-sanitization system is undetectable to visual and statistical analysis techniques.**

Keywords-image sanitization, steganography, information hiding

I. INTRODUCTION

Classified information such as legal documents or photographs often have to undergo sanitization, also referred to as redaction [1], which is the process of removing certain information that is deemed sensitive [2]. This is often done in court cases where an image is graphic in nature and the court has deemed that the entire image should not be shown to the public and that certain areas should be blacked out. Normally the unsanitized versions of these images are also needed by the court and thus two copies must be kept, with the unsanitized version at risk of leaking to the public.

Steganography is the practice or art of hiding information in digital objects [3]. The main purpose of steganography is to hide the existence of the embedded information, as opposed to simply blocking access to the information as is the case with cryptography [4]. Watermarking is similar to steganography in that it also hides information in digital media, but differs in goals [3]. Where steganography is mostly concerned with covert communication, the goal of watermarking is to protect copyrighted material [5]. Self-embedding is one example of a watermarking technique used for image authentication and the protection of digital media [6].

This paper proposes a method for sanitizing an image in such a way that the removed information and the original image are stored as one object. The proposed method uses steganography and self-embedding techniques to hide the sanitized part of the image, within the image itself.

The paper is structured as follows: Section II gives background on self-embedding techniques. The proposed self-sanitization system is discussed in Section III and the analysis of the self-sanitization system is given in Section IV. The paper is concluded in Section V.

II. SELF-EMBEDDING TECHNIQUES

Self-embedding techniques are semi-fragile watermarking techniques used in image authentication to detect image tampering [7]. Self-embedding techniques usually embed an approximation of the original image within the image itself and uses mapping functions and keys to provide robustness and security [8]. In its simplest form, self-embedding is used to detect and localise the areas of an image that were damaged or changed, and at best self-embedding can restore an approximation of the original contents that formed part of the damaged area [9].

The method proposed in this paper has two main similarities with that of self-embedding techniques: The first similarity is that both self-embedding as well as self-sanitization embed part of the image within the image itself, using steganographic techniques. The second similarity is that in both cases the resulting image should visually be as close as possible to the original image.

There are however many differences between the self-sanitization system, proposed in this paper, and existing self-embedding techniques. These differences include both differences in implementation, as well as differences in the thought behind the creation and use of the algorithms. One of the differences between self-embedding and self-sanitization involves their objectives: Steganography's main objectives are undetectability (resistance against both visual as well as statistical analysis), robustness (resistance to image processing attacks) and payload capacity (the amount of information that can be embedded) [10]. Although all three these objectives are desirable, most applications can only focus on one or two of the objectives and a trade-off is usually necessary.

978-1-4799-7755-0/15/$31.00 ©2015 IEEE

Self-embedding techniques generally focus on robustness against image processing attacks [11], since these types of attacks could destroy the embedded approximation and thereby make it impossible to restore the original image. The self-sanitization system on the other hand focuses on undetectability, due to the fact that a sensitive portion of the image is stored and must remain hidden from the public. The sanitized part of the image should thus be undetectable by visual and statistical analysis. This, however, also means that where self-embedding techniques prepare for, and are able to recover from, image processing attacks, the system proposed in this paper cannot withstand image manipulation.

Another difference is the part, or section of the image that is to be hidden, and the fact that an image can only hide a certain amount of information before visual artefacts can be detected in the image. Self-embedding techniques try to hide the entire image within itself, thus these techniques can only hide and restore a lower quality approximation of the original image [12]. The self-sanitization system, however, only needs to hide a specific area of the image and can thus embed the entirety of that part and restore it fully.

The core difference between self-embedding techniques and the self-sanitization system, however, is the goals of the two technologies and how they are used. Self-embedding techniques were created to authenticate images [12] where the self-sanitization system was developed to remove information from unauthorised view. Additionally, self-embedding techniques are used to retrieve damaged image data, where the self-sanitization system is used to hide and restore intentionally removed information.

The next section discusses the self-sanitization system and the algorithms that were used in the implementation of the system.

III. THE SELF-SANITIZATION SYSTEM

For the system proposed in this paper, the process of removing and hiding the sensitive part of the image is referred to as *sanitization* and the process of recovering the hidden information and restoring the image is referred to as *desanitization*.

The self-sanitization system provides the option of encrypting the sanitized part of the image before hiding the information. The use of encryption increases the security of the information should the hidden information be detected by an unauthorised person. The inclusion of encryption is optional, since it could also cause additional problems in the storage, distribution and use of the encryption keys.

To sanitize an image, the user manually selects an area of the image to remove. The sanitized part of the image is converted into a bitstream and hidden inside the remaining image bits. The start point and size of the removed area are also hidden in the image to enable the automation of the desanitization process. The resulting image is visually the same as the original image, except for the sanitized area which is blacked out.

To desanitize an image, the sanitized image bits are extracted from the image and restored to their original location in the image. The resulting image is a visual copy of the original image.

The self-sanitization system uses a variation of Least Significant Bit (LSB) steganography to hide the sanitized area of the image in the three colour channels of a RGB colour image. The system also incorporates a technique for decreasing the amount of information that needs to be hidden, thus increasing the undetectability of the embedded information. The steganography technique used and the method for decreasing payload size are discussed in the following sections.

A. Steganography technique used in the self-sanitization system

Classic Least Significant Bit (LSB) steganography refers to methods of steganography that hide information in the least significant bits of the cover image [13]. Changes made to the least significant bit of a byte, in other words the 8^{th} bit of a byte, have the least effect on the information that the byte is representing and can thus be replaced with a bit from the information to be hidden, without changing the appearance of the image [3]. Depending on the type of image used and the colour depth of the image, the least significant bit of a pixel (in a greyscale image), one colour channel of a pixel (in a colour image) or a coefficient (in an image in the transform domain) can all be used to embed information in.

Although different image formats can be used with LSB steganography, the self-sanitization system implementation was done with full colour bitmap (BMP) images. BMP images were used because this format has a simple structure and is lossless, therefore increasing the amount of information that can be embedded. LSB steganography also offers a large capacity for hidden information [3] which is important since the original image should be able to accommodate essentially a small image inside of it. However, other file formats and steganography algorithms could also be implemented.

Changing LSB values in an image unfortunately results in an unnatural histogram that is easy to detect [14]. To increase the undetectability of the embedded information the self-sanitization system implements a simplification of an algorithm called ±1 embedding [15]. Instead of flipping the least significant bit to the opposite bit when embedding the information, the absolute value of the number stored in the byte is decreased by 1. This has the effect of modifying the LSB, but does not create a pattern since it often modifies other bits as well. The ±1 embedding algorithm is harder to detect with statistical analysis than classic LSB steganography since it does not leave a clear signature on the histogram of the image [14].

The self-sanitization system also changes the way negative values are interpreted by the program, changing their meaning so that a negative number with a least significant bit of 0 actually represents a 1 and a negative number with a least significant bit of 1 actually represents a 0. This is done to avoid shrinkage, which occurs when the embedding process produces more even values than odd values [16].

The steganographic channel, in other words the subset of pixels that are used to hide the sanitized information, depends on the size of the sanitized area. To decrease the probability of detection, the self-sanitization system attempts to space the affected pixels as far apart from each other and as evenly as possible while still remaining inside the parameters of the image. Depending on the size of the sanitized part of the image and the size of the remaining image, sanitization uses every x^{th} bit to spread the sanitized area evenly over the remaining image bits.

B. Most significant bit method for decreasing payload size

The image area to be sanitized is based on user selection and the size in bits of such an image section, even a small one, is significantly larger than the size of a normal text message that is usually hidden inside an image with steganography. Since larger payloads increase the probability of detection [17], the self-sanitization system implements a method for decreasing the size of the payload in a lossless way, to maintain 100% of the quality once restored. This is done by only extracting the most significant bits (MSBs) from the sanitized area, leaving some of the least significant bits behind. As shown in Fig. 1 this process does not reveal any meaningful information about the sanitized area.

The most significant bits represent most of the image informative information inside the selected area. Thus it is viable to leave behind the least significant bits as this information is not image informative and therefore is not vital to the image contents of the area. The self-sanitization system lets the user decide how many least significant bits to leave behind and thus it is in the users' power to increase the systems' capacity or decrease the amount of information left behind.

Depending on the amount of least significant bits left behind, the sanitized area will contain visible artefacts. These visible artefacts are mostly in the form of noise and do not reveal enough about the area to let an unauthorised viewer deduce any information about its original contents. Since the most significant bits are simply reapplied to the sanitized area, none of the quality of the sanitized area is lost after desanitization.

A magnified look at the hidden areas of the images in Fig. 1 is shown in Fig. 2 to examine the resulting sanitized area after sanitization when removing the three most significant bits and leaving five least significant bits behind. Fig. 2 shows that the image bits that are left behind are not sufficient to interpret the contents of the image before sanitization. The amount of least significant bits that can be left behind, without leaving behind information pertaining to the original contents of the image, is image dependant and is thus an optional feature.

Figure 1. An original, unsanitized image (a), an image where all of the sanitized bits were removed (b) and an image where the three most significant bits were removed (c)

To determine the undetectability of the self-sanitization system, analysis was done on sanitized images. The results of the analysis are discussed in the next section.

IV. ANALYSIS OF THE SELF-SANITIZATION SYSTEM

Steganalysis is the practice or art of breaking a steganography algorithm by attempting to detect hidden information [3]. For experimental results, sanitized images were analysed using two known steganalysis techniques to determine the level of undetectability of the hidden information. Visual analysis was done using LSB enhancement and statistical analysis was done using the chi-square test on pairs of values (PoVs). LSB enhancement and the chi-square test have been shown to reveal flaws and weaknesses in many existing steganographic algorithms [18]. Many other steganalysis techniques exist, but these two techniques were chosen for the analysis since steganalysis tools usually use LSB enhancement and the chi-square test in a targeted analysis of specifically LSB steganography [19]. The results of the analysis are discussed in the next sections.

A. LSB enhancement

LSB enhancement is a visual analysis technique, which means that the image is modified and then requires a human to look at the modified image to try and detect a pattern [20].

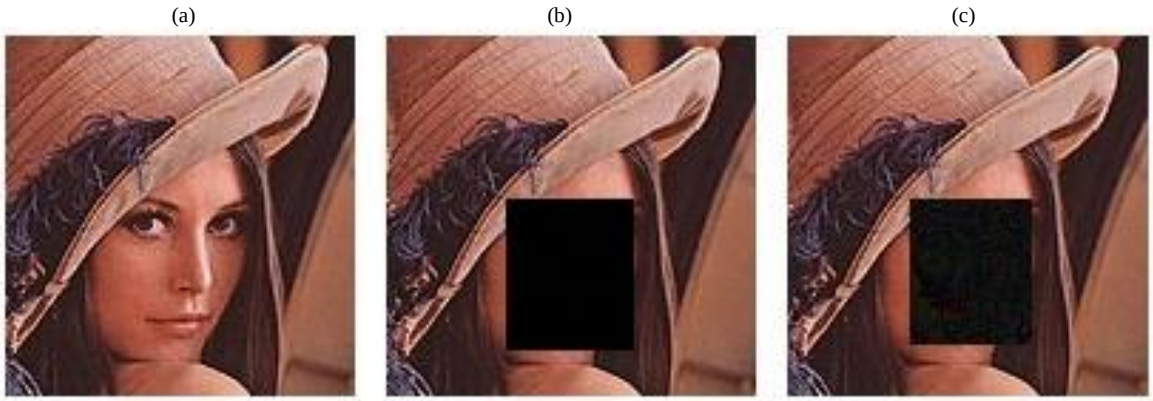

Figure 2. Magnified versions of the images from Figure 1

LSB enhancement processes an image by extracting the least significant bit of each pixel (can also be more than one bit per pixel, depending on colour depth) [20]. To display the resulting image, each pixel in represented entirely by the least significant bit by setting the value of each pixel to the maximum if the least significant bit is 1 or 0 if the bit is 0. Once the image is in this state the steganalyst can visually study the image for anomalies or patterns.

The successful detection of information through LSB enhancement decreases with an increase in perceived randomness of the hidden information [21]. In the self-sanitization system the hidden sanitized part of the image was spread out evenly over the remaining pixels, thus smaller payloads increased the perceived randomness of the hidden information since affected pixels are spaced further away from one another. The success of detecting information in the self-sanitization system using LSB enhancement is thus dependant on the amount of information that is hidden.

LSB enhancement was done on sanitized images starting with a payload size of 0% of the remaining image bits (in other words no information was hidden). The payload size was then gradually increased by increments of 5% until visual artefacts started to appear. The values recorded in Table 1 show the maximum payload size that could not be detected through LSB enhancement. Depending on the image, the maximum payload size required for undetectability was between 15% and 20% meaning that the payload should be kept as small as possible either by selecting only a small area for sanitization or by implementing the most significant bit method to decrease payload size.

TABLE 1. Results of LSB enhancement showing the maximum payload size at which information was not detected

Image	Image size	Payload
1	549Kb	20%
2	103Kb	15%
3	11Mb	20%
4	278Kb	20%
5	601Kb	20%
6	147Kb	20%
7	254Kb	15%
8	234Kb	20%
9	910Kb	20%
10	125Kb	20%

Fig. 3 shows the results of LSB enhancement in an image with no sanitization in comparison with an image where all of the sanitized image bits were removed and an image where only the four most significant bits were removed. As can be seen from Fig. 3 removing all of the bits resulted in the creation of a visual pattern, while the most significant bit method was undetectable.

B. The chi-square test on PoVs

In LSB steganography a value of a pixel can only be modified into a value with the opposite least significant bit, for example the value 0 can only change into a 1 and vice versa [20]. In an image without steganography these pairs of values (PoVs) are typically distributed unevenly [22]. However, after hiding information in an image using LSB steganography, the frequencies of both values of each PoV become equal [20].

A statistical chi-square test can be designed to measure if a given set of observed data is similar or not to an expected set of data [23]. In the case of steganography, the chi-square test can be used to test the difference between the expected frequencies of PoVs and the actual frequencies [20].

The chi-square test on PoVs is more successful in detecting sequentially embedded bits than more randomly scattered bits [22], thus the information would again be more difficult to detect if the payload is smaller. To test the detectability of the self-sanitization system with the chi-square test, the test was done on 10 sanitized images. An initial relatively large payload size of 50% was used, first on images where the entire sanitized area was removed and then on images where only the four most significant bits were removed, thus decreasing the payload size. The results of the tests are recorded in Table 2.

TABLE 2. Results of chi-square test on PoVs showing whether the hidden information was detected or not

Image	Image size	MSB not used	MSB used
1	549Kb	Detected	Not detected
2	103Kb	Detected	Not detected
3	11Mb	Detected	Not detected
4	278Kb	Not detected	Not detected
5	601Kb	Detected	Not detected
6	147Kb	Detected	Not detected
7	254Kb	Detected	Not detected
8	234Kb	Detected	Not detected
9	910Kb	Detected	Not detected
10	125Kb	Detected	Not detected

As expected, the hidden information could be detected when there was a larger payload size, but could not be detected when the most significant bit method was used to decrease the payload size. This is due to the fact that the chi-square test is more successful in detecting sequential bits and when the payload is decreased the hidden bits are spread further away from one another over the remaining image bits.

V. CONCLUSION

Images that are deemed sensitive or classified by an authority need to be safely sanitized, with the ability to reverse the sanitization if an authorised user deems it necessary. This paper presented a system that removes the sensitive part of an image and hides it within the image itself, thereby making the sanitized and the unsanitized version of the image the same object. LSB steganography was used in the implementation of the self-sanitization system and the system implemented a method to decrease the payload size and improve undetectability.

The detectability of the self-sanitization system was tested using two steganalysis techniques, namely LSB enhancement and the chi-square test on PoVs. Both tests indicated an increase in probability of detection if the payload size was larger than between 15% and 20% of the remaining image size. It is thus recommended that the most significant bit method be used to further decrease payload size by only removing the most significant, image informative, bits from the sanitized area and leaving the noise-like least significant bits behind.

Further work can be done on the system by experimenting with different steganography algorithms and different image formats.

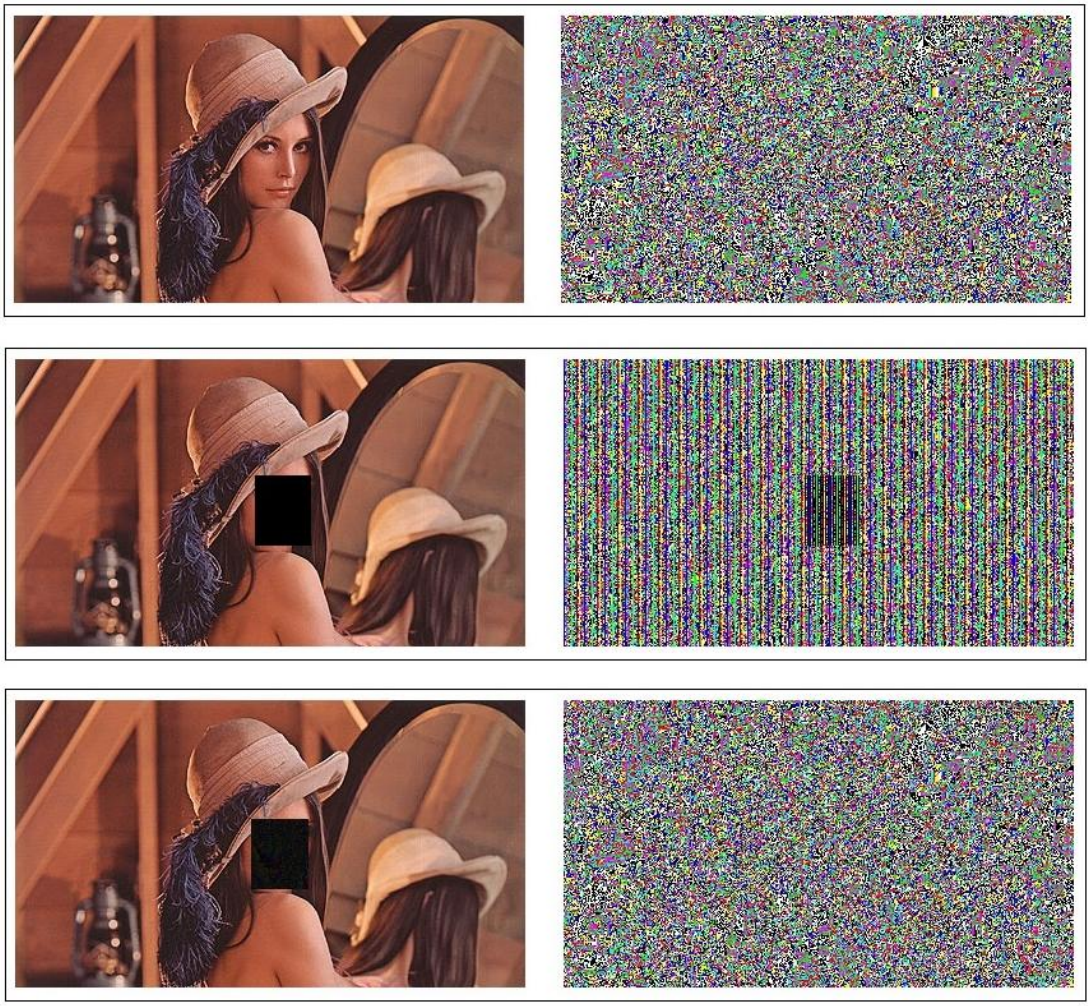

Figure 3. Results of LSB enhancement on image with no hidden information (a), sanitized image where entire sanitized area was removed (b) and image where four most significant bits of sanitized area were removed (c)

ACKNOWLEDGMENT

The author would like to thank Jacques Viljoen for his help in the development of the self-sanitization implementation and experimental results.

REFERENCES

[1] V. Chakaravarthy, H. Gupta, P. Roy and M.K Mohania, "Efficient techniques for document sanitization", Proceedings of the 17th ACM conference on Information and Knowledge management, pp 843-852, 2008.

[2] G.W. Manes, L. Watson, A. Barclay, D. Greer and J. Hale, "Towards redaction of digital information from electronic devices", Proceedings of the Conference on Digital Forensics, Security and Law, pp 197-203, 2007.

[3] J. Fridrich, "Steganography in digital media: Principles, algorithms and applications", Cambridge University press, 2009.

[4] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography", Security and Privacy, no 3, pp 32-44, 2003.

[5] H. Wang and S. Wang, "Cyber warfare: Steganography vs. steganalysis", Communications of the ACM, 47(10) pp 76-82, 2004.

[6] J. Fridrich and M. Goljan, "Protection of digital images using self-embedding", Symposium on Content Security and Data Hiding in Digital Media, 1999.

[7] X. Zhu, A.T.S Ho and P. Marziliano, "A new semi-fragile image watermarking with robust tampering restoration using irregular sampling", Signal Processing: Image Communication, 22(5) pp 515-528, 2007.

[8] J. Fridrich and M. Goljan, "Images with self-correcting capabilities", Proceedings of the International Conference on Image Processing, vol 3 pp 792-796, 1999.

[9] I. Kostopoulos, S.A.M Gilani and A.N Skodras, "Colour image authentication based on a self-embedding technique", Proceedings of the International Conference on Digital Signal Processing, vol 2 pp 733-736, 2002.

[10] A. Cheddad, J. Condell, K. Curran and P. McKevitt, "Digital image steganography: Survey and analysis of current methods", Signal Processing, vol 90(30) pp 727-752, 2010.

[11] A. Cheddad, J. Condell, K. Curran and P. McKevitt, "A secure and improved self-embedding algorithm to combat digital document forgery", Signal Processing, vol 89(12) pp 2324-2332, 2009.

[12] C. Rey and J.L Dugelay, "A survey of watermarking algorithms for image authentication", EURASIP Journal on Applied Signal Processing, vol 6 pp 613-621, 2002.

[13] M.A.F Al-Husainy, "Message segmentation to enhance the security of LSB image steganography", transit, vol3(3), 2012.

[14] J. Fridrich, D. Soukal and M. Goljan, "Maximum likelihood estimation of length of secret message embedded using ±k steganography in spatial domain", Electronic Imaging, International Society for Optics and Photonics, pp 595-606, 2005.

[15] B. Li, J. He, J. Huang and Y.Q Shi, "A survey on image steganography and steganalysis", Journal of Information Hiding and Multimedia Signal Processing, vol 2(2) pp 142-172, 2011.

[16] X. Zhang, S. Wang and K. Zhang, "Steganography with least histogram abnormality", Computer Network Security, pp 395-406, 2003.

[17] J. Fridrich, R. Du and M. Long, "Steganalysis of LSB encoding in color images", IEEE International Conference on Multimedia and Expo, vol 3 pp1279-1282, 2000.

[18] C.A Stanley, "Pairs of values and the chi-squared attack", Department of Mathematics, Iowa University, 2005.

[19] A. Munoz, "Stegsecret: A simple steganalysis tool", stegsecret.sourceforge.net, last accessed on 2015-04-17.

[20] A. Westfeld and A. Pfitzmann, "Attacks on steganographic systems", Information Hiding, pp 61-76, Springer, 2000.

[21] P. Bateman and H.G Schaathun, "Image steganography and steganalysis", Department of Computing, Faculty of Engineering and Physical Sciences, University of Surrey, 2008.

[22] J. Fridrich and M. Goljan, "Practical steganalysis of digital images – state of the art", Proceedings of SPIE, vol 4675 pp 1-13, 2002.

[23] P.E. Greenwood, "A guide to chi-square testing", John Wiley & Sons, 1996.