Sab - íomha: An Automated Image Forgery Detection Technique Using Alpha Channel Steganography

Muhammad Shahid Bhatti(✉), Syed Asad Hussain, Abdul Qayyum, Imran Latif, Muhammad Hasnain, and Sajid Ibrahim Hashmi

Department of Computer Science, COMSATS Institute of Information Technology, Lahore, Pakistan
{msbhatti,asad.hussain,aqayyum,imranlatif,mhchaudary, sajid.hashmi}@ciitlahore.edu.pk

Abstract. Digital images nowadays have become a very popular way to transfer media. However, their security remains very challenging for at least two reasons; first, images can be easily tampered with the help of software tools; second, originality of the tampered images cannot be verified through a naked-eye.

The existing techniques to forgery detection are not very easy to use and are applicable to very specific settings. In this research paper, we propose a light-weight technique named Sab - íomha to detect image forgery. The approach uses invisible watermarking and entails with inserting a hidden cipher to certain bits of an image file. Any attempts to tamper with the image distort the sequence of the bits signed by the author in the first place. A software tool is also developed to automate the proposed technique and to demonstrate its usefulness.

Keywords: Digital images · Tamper · Steganography · Metadata · Forgery detection · Cipher

1 Introduction

Digital images have attracted researchers for decades now [1–3, 15]. In the present age, they are used as a powerful tool for formal as well as informal communication [6]. With the advent of various digital devices and communication technologies, the use of image files has become very popular to share visual moments and photographs. However, the growing magnitude of malpractices has accelerated new influences in the domain. We assuredly exist in an age which is abundantly vulnerable to a noticeable cluster of digital content. Likewise, confronting with tampered images is a norm these days. Since the data is openly available, malicious users can easily tamper with image files for fun, misuse those for some social gains or to influence any legal proceedings. The phenomenon is well supported by the availability of supporting software applications. Therefore, the situation calls for taking some proactive measures to confront the challenge.

Image tampering is a common image manipulation operation [6]. Deception of typical photographs is comparatively crucial, demanding technical proficiency whereas digital images are prone to tampering. Some basic software tools can manipulate them easily. Manipulations mainly include replication, duplication, exchanging and removing

Á. Rocha et al. (eds.), *Recent Advances in Information Systems and Technologies*, Advances in Intelligent Systems and Computing 570, DOI 10.1007/978-3-319-56538-5_74

parts of an image. Originality of analog contents can be validated easily through a naked eye as any attempts to forgery can be visualized effortlessly. Contrarily, advent of computer aided tools has made it very trivial to manipulate with digital images. For instance, Fig. 1 demonstrates one such example.

(a) (b)

Fig. 1. (a) Original image (b) Object on the extreme right is inserted

Originally, seven objects were present in the photo. Object on the far right is inserted. However, by looking at the figure, one cannot realize that originality of the image was compromised. Before taking an appropriate social or legal action when any such situation encounters, it is inevitable to detect that an image was edited. However, as image processing techniques are cultivating rapidly, tampering of digital content without leaving any observable footprints has become very easy. As it is evident from the figure, examining originality of an image becomes a very challenging task as to alter a digital image can be done flawlessly than a printed one. Therefore, the situation calls for taking some proactive measures to confront the challenge.

Rest of the paper is organized as follows: remaining parts of Sect. 1 discuss our contribution and the current state of the art on the domain. Section 2 reflects upon the literature review we conducted to carry out the research. Section 3 explains the contribution in terms of the proposed technique and an example from the software tool we developed to automate and demonstrate usefulness of the work. Finally, Sect. 4 sums up with conclusion followed up by references.

1.1 Contribution of Our Research Project

Validation is the process of checking integrity of an object. This is exactly what we aim to achieve to forgery detection domain through this research project. The ultimate goal is to check digital images for originality and to validate that their integrity was not compromised since their creation. The techniques to date have used signature based methods to protect an image or to detect forgery. However, those techniques are not always applicable because of deficiencies and overheads involved in their use. We propose inserting watermark, which we call cipher, in structured patterns to certain bits of digital image-files.

Digital watermarks are a popular technique for media files for maintaining copyright information and identify their ownership [15]. In general, two types of watermarks can be inserted in an image, visible and invisible depending upon user requirements. A visible watermark inflicts an identification mark on an image whereas the invisible one embeds a hidden spot in it. We opt for the hidden one which is randomly inserted in the form of a cipher text across multiple pixels of an image file. The structure of cipher would be distorted if someone tries to edit the image by any means.

Through this work, we aim to address the problem of digital image forgery by having twofold contribution: firstly, we logically break an image file to separate its metadata from the contents. It should be noted that an image file is composed of combination of pixels and each one of those is represented by an ordered set of bytes for different colors like Alpha, Red, Green and Blue. However, bits for Alpha are not used for data storage. Therefore, we have made use of the available bits to insert cipher string into image files. The watermark would be invisible and upon tampering would be removed automatically. Secondly, we demonstrate the usefulness of the approach through a software tool that we developed to automate the proposed technique.

1.2 Current State of the Art

The current approaches to validating image files usually involve copying those files to some dedicated software tools [4]. The users are then provided with different features like error level analysis (ELA), Joint Photographic Experts Group (JPEG) format information, edge detection, metadata and watermark. ELA can highlight the modified or edited part of an image if any. Then different regions of an image having different compression levels are identified which can make one easily detect any problem areas through a naked eye. JPEG% specifies quality of an image. If a file was edited or re-saved several times, it loses its quality [5]. Metadata describes the image itself. This includes the type of image, e.g. JPEG, internal formats, dimensions, and color scheme. Metadata of an image also provides the information like date of creation, date of modification, software editor name, EXIF tags, file tags, and camera tags. The Exchangeable Image File format (EXIF) is typically used by camera manufacturers to identify information about the camera settings used for the photo. It includes timestamp, camera make and model, and lens settings. These features can be verified to ensure integrity of an image. If comments are inserted to the header, they are incorporated as its metadata. Most digital cameras do not embed comments to photographs automatically. However, if any annotations are found, it is an indication that the image was reprocessed by some supporting software tool.

Several currently available approaches to image forgery detection make use of manipulating the information provided through file header only. Recording more information at the time of photo capture or any attempts to manipulate it can easily make the task of image handling more difficult and time consuming. Moreover, the existing techniques do not take into consideration the digital contents or storage of a file itself. Our proposed technique addresses the challenge using a more simple yet robust mechanism; i.e. watermark is embedded to an image through a protection module which alleviates the need for manipulating with the file header. Any endeavors to edit the image would

distort the embedded watermark. Hence any favorable attempts to doctor the file can be detected straightaway.

2 Literature Review

Inconsistent shading, lightening and shadows are one way of detecting evidence on image forgery [6]. Mixture of shading and shadow serve as a logical mean for the purpose and both would be dependent on each other but if they are not, the corresponding image would be a tampered one. However, the method is not of use in case of historical text documents which do not have shadow.

The use of colour discrimination has also been used as an evidence for image forgery. To accomplish that, some authors have proposed a spliced image detection mechanism [7]. Illumination inconsistencies of an image were detected by extracting text or edge based features. If the image file carried information like camera model, type of image, and camera motion after being captured, the information collectively could be helpful to make forgery a difficult and time taking task [8]. However, detection of forgery having reflections is not a nontrivial task these days. [9] Proposed a mechanism that removed traceable information from an image to make it appear trustworthy. Another way of detecting forgery entailed text based signing of image files [10]. If the sign-in got distorted, it was an indication that integrity of the image was compromised.

[11] Proposed using thumbnails for checking authenticity of images. The thumbnails were created using compression, contrast settings and filter models together which in turn were used to detect whether the corresponding images were compromised or not. Those models were then compared with the editing software and the originator cameras. A hidden watermark technique [12] was proposed for image forensics. It controlled cropping, JPEG-lossy compression and other processing operations of an image by adding an invisible watermark in it in such a way that a missing or distorted watermark could indicate forgery.

[13], as its title suggests, proposed a technique by detecting inconsistencies in lighting. Nevertheless, lighting of a scene can not only be complex but also its matching in an image can be hard as the difference would be indistinguishable. [14] Dealt with using 3D lighting coefficient for forensics purposes. However, the lighting and surface assumptions used were very specific. In addition to that, estimation of the 3D shape of an image object itself remained a challenge.

In short, the existing approaches lack novelty required to tackle the problem domain. To our knowledge, there exists no single technique that can be applicable and equally useful to multiple types of digital images simultaneously; i.e. camera generated digital images, human drawn images in soft form, and digital documents saved as image files.

3 Sab - íomha: The Proposed Technique

We propose forgery detection by introducing two features; i.e. *protection* and *detection* of image files. Figure 2 gives an overview of the research. There are two segments of the project; firstly, steganography; secondly, forgery detection. Image steganography is

achieved by performing multiple steps. First, the image is converted into byte stream that separates metadata from the image storage. Second, a cipher code is inserted across specific bits of an image file. The cipher is in text form and can be replicated across multiple bytes depending upon the size of the image.

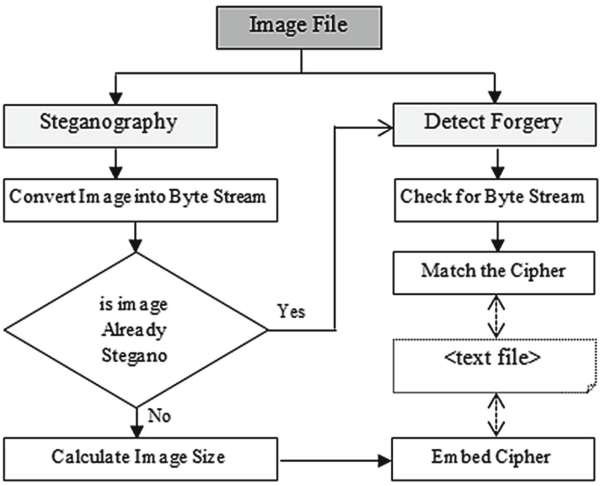

Fig. 2. An overview of Sab - íomha

Since we opt for invisible watermark approach, the cipher that we randomly insert across multiple pixels would be hidden. The cipher changes automatically if someone tries to edit the image. Any attempts to tamper with the image would alter the watermark. The distorted cipher would be detected through a detection module. Digital images are composed of millions of small dots called pixels. Each pixel has some characteristics for demonstrating clarity of an image and quality of its colours. In a 32-bit four channel image, each pixel is represented by four bytes. Besides Alpha, each byte represents a colour; i.e. Red, Green and Blue as shown in Fig. 3. ARGB is considered to be the most accepted scheme for exhibiting colours. It stocks a pixel in a pattern of Alpha, Red, Green, and Blue. For hidden watermarking, our technique uses the smallest bit of Alpha in the ARGB scheme. It does not alter value of any bit; however, text length should be defined before it is embedded in the image file as cipher code.

The proposed technique separates metadata from image pixels. The file is then converted into pixels. This pixel stream is then converted into byte stream. The image is then converted into a bit stream and a cipher code is inserted into the image as hidden watermark. Consider the image as matrix of A of $m \times n$ length pixels as shown below. Total number of pixels of an image can be determined by $m \times n$ relation. *Least significant bit insertion* is a common and an easy approach to embedding information in an image. The 8^{th} bit of either of the bytes of an image file is replaced with a one-bit cipher code. When using a 32-bit image, the least significant bit of each of the Alpha component of the colour scheme is utilized as shown as the shaded bit of a pixel in Fig. 3(b).

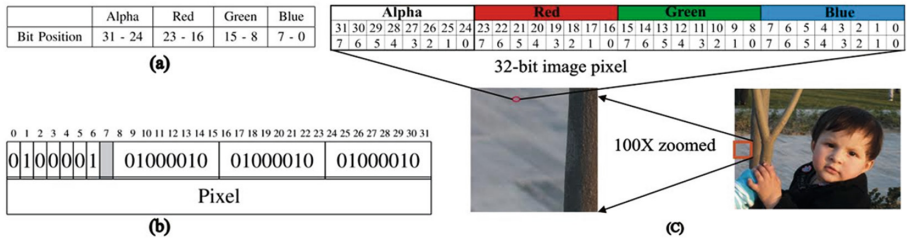

Fig. 3. (a) Bit positions of color channels of a 32-bit image (b) Least significant bit of an Alpha byte (c) Illustration of steganography through Alpha channel

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1n} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ a_{m1} & A_{m2} & A_{m3} & \dots & A_{mn} \end{bmatrix} = (a_{ij})_{m \times n}$$

A 800 × 600 pixel image can thus store a total number of 1,440,000 bits or 180,000 bytes of a cipher code. For example, a grid of 8 pixels of a 32-bit image can be as follows: when the number 35, having binary representation 00100011, is embedded into the least significant bits of Alpha bytes of an image file. The resulting grid would be as follows: number 35 is added to consecutive eight pixels only; shaded pixel bits are highlighted as shown in Fig. 4. It should be noted that only the number 8 bit of each of the alpha bytes is inserted with the cipher code which will not affect visual contents of an image. Since our proposed technique manipulates images at the composition level, the injected changes cannot be observed through a naked eye. Moreover, all pixels are protected by using the method.

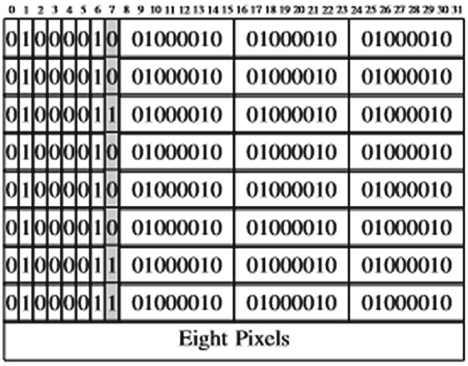

Fig. 4. Alpha bytes - least significant bits are highlighted

The relation n = (A − 32)/8 can be used to calculate the maximum length of the cipher code that can be embedded at the ratio of 1 bit in an image file. The cache space

can be increased to n = (4P − 8)/8 if we want to enhance the cache position for the cipher, where P denotes pixels and n is length of the text.

Figure 5 shows interface of the tool we developed using Java technologies to automate Sab - íomha. It provides multiple options which are visible in the figure. The "Steg Image" inserts a cipher code in the image and the updated image can be saved as a new file. "Detect Forgery" opens up a new dialogue box shown in Fig. 6.

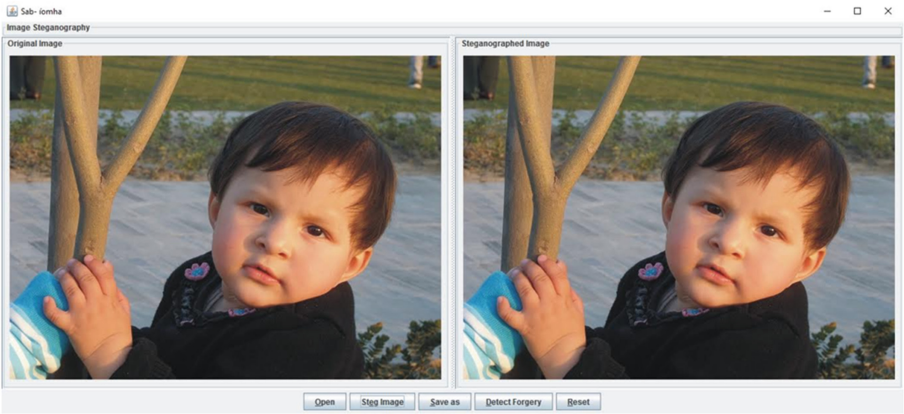

Fig. 5. Home-interface of the tool with an image loaded in

Fig. 6. Authentication of an image – the cipher is displayed

It is a two-step procedure. First, we steganograph an image and in the second step we check integrity of the same image. In order to demonstrate usefulness of our

technique, we randomly pick an image and upload it to the tool. The image on the right hand side of Fig. 5 is steganographed with an invisible watermark, the cipher, which is a stream of string in this case. The noticeable thing is doing so does not compromise the quality of the image. The same image file can be detected for forgery in case any attempts were made to edit the file. If the validation process produces cipher, which is the case in Fig. 6, it is a testimony to the originality of the image. Otherwise, a message will be displayed in the text box that the image was edited by someone.

4 Conclusion

Our research work has proposed a new stenography technique to authenticate digital images. The approach addresses the issue of image forgery in a unique way. Error Level Analysis (ELA), metadata, JPEG% and watermarks were incorporated to come up with an effective and cohesive technique. By using our software tool, users can first insert watermarks in image files and then authenticate those images for originality. The technique can not only validate photographs but also important electronic documents that are stored as digital images. The work enables stakeholders to be digital detectives at their own and empowers them to get an insight on the originality of an image under consideration.

The proposed technique was tested by uploading random images to the developed software tool. The results demonstrated that the technique could verify the authenticity of the image files effectively. Currently, the work supports JPEG, BMP, and PNG image formats. However, as part of the future work, we aim to provide support for other formats as well.

References

1. Harris C., Stephens M.: A combined corner and edge detector. In: Proceedings of the Alvey Vision Conference, pp. 147–151. IEEE Press (1988)
2. Bay, H., Ess, A., Tuytelaars, T., Gool, L.V.: Surf: speeded up robust features. Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
3. Schetinger V., Oliveira M.M., da Silva R., Carvalho T.J.: Humans are easily fooled by digital images. CoRR, vol. abs/1509.05301 (2015). http://arxiv.org/abs/1509.05301
4. Farid., H.: A survey of image forgery detection. IEEE Sig. Process. Mag. **26**(2), 16–25 (2009)
5. Fan W., Wang K., Cayre F., Xiong Z.: A variational approach to JPEG anti-forensics. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3058–3062. IEEE Press (2013)
6. de Carvalho, T.J., Riess, C., Angelopoulou, E., Pedrini, H., de Rezende Rocha, A.: Exposing digital image forgeries by illumination color classification. IEEE Trans. Inf. Forensics Secur. **8**(7), 1182–1194 (2013)
7. Kirchner, M., Winkler, P., Farid, H.: Impeding forgers at photo inception. In: Proceedings of SPIE Conference on Media Watermarking Security and Forensics (2013)
8. O'Brien, J.F., Farid, H.: Exposing photo manipulation with inconsistent reflections. ACM Trans. Graph. (TOG) **31**(1), 1–11 (2012)

9. Conotter, V., Boato, G., Farid, H.: Detecting photo manipulation on signs and billboards. In: Proceedings of the International Conference on Image Processing, pp. 1741–1744. IEEE Press (2010)
10. Kee, E., Farid, H.: Digital image authentication from thumbnails. In: Proceedings of the SPIE 7541, Media Forensics and Security II (2010)
11. Hsu, C.T., Wu, J.L.: Hidden digital watermarks in images. IEEE Trans. Image Process. **8**(1), 58–68 (1999)
12. Luo, W., Qu, Z., Pan, F., Huang, J.: A survey of passive technology for digital image forensics. Front. Comput. Sci. China **1**, 166–179 (2007)
13. Johnson, M.K., Farid, H.: Exposing digital forgeries in complex lighting environments. IEEE Trans. Inf. Forensics Secur. **3**(2), 450–461 (2007)
14. Fan, W., Wang, K., Cayre, F., Xiong, Z.: 3D lighting-based image forgery detection using shape- from-shading. In: Proceedings of the European Conference on Signal Processing, pp. 1777–1781 (2012)
15. Cox, I.J., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. IEEE Trans. Image Process. **6**(12), 1673–1687 (1997)