Using Sab-Iomha for an Alpha Channel based Image Forgery Detection

Muhammad Shahid Bhatti[1], Syed Asad Hussain[2], Abdul Qayyum[3],
Abdul Karim Shahid[4], Muhammad Usman Akram[5], Sajid Ibrahim Hashmi[6]
Department of Computer Science,
COMSATS University Islamabad,
Lahore Campus, Pakistan

***Abstract*—Digital images are a very popular way of transferring media. However, their integrity remains challenging because these images can easily be manipulated with the help of software tools and such manipulations cannot be verified through a naked-eye. Although there exist some techniques to validate digital images, but in practice, it is not a trivial task as the existing approaches to forgery detection are not very effective. Therefore, there is need for a simple and efficient solution for the challenge. On the other hand, digital image steganography is the concealing of a message within an image file. The secret message can be retrieved afterwards by the author to check the image file for its veracity. This research paper proposes Sabiomha, an image forgery technique that make use of image steganography. The proposed technique is also supported by a software tool to demonstrate its usefulness. Sabiomha works by inserting an invisible watermark to certain alpha bits of the image file. The watermark we have used to steganograph an image is composed of a combination of text inputs the author can use to sign the image. Any attempts to tamper the image would distort the sequence of the bits of the image pixel. Hence, the proposed technique can easily validate originality of a digital image by exposing any tampering. The usability of our contribution is demonstrated by using the software tool we developed to automate the proposed technique. The experiment which we performed to further validate our technique suggested that Sabimoha could be flawlessly applied to image files.**

Keywords*—Digital images; tamper; steganography; metadata; forgery detection; cipher; image authentication; image validation; watermarking

I. Introduction

A. Background

Applications of digital images have been the focal point of computer vision researchers for decades now [1]–[4]. Digital content is used as an effective way of communication among different stakeholders [5]. The advent of digital devices and communication technologies has led to increase in the use of image files for sharing visual moments and photographs. Digital images are generated through cameras with-out transformation and development process contrary to camera reels in the past and can be delivered electronically through any supporting communication channel.

Although an image data is generally considered reliable but with the passage of time, the digital technology itself has compromised the faith we have had in electronic content. The ever- increasing trend of malpractices in image forensics has posed new challenges to the research horizon as we continue to exist in the era which is very much vulnerable to multiple facets of digital contents. The situation seeks effective and efficient solution to ensure integrity of digital images.

With multi-million users using emails and social media, nearly countless digital content is distributed and shared every day. A large portion of the content comprises of digital images. These days users can easily capture their memorable moments through digital cameras and can share with others by publishing the image files on the web. On the other hand, users can potentially receive tampered images and unknowingly circulate those as well. Since digital data is easily accessible these days, obnoxious users can manipulate image files for entertainment and at times abuse those for some societal or political gains or to dictate any legal affairs. This phenomenon is reinforced by the availability of some supporting software applications. Hence the situation calls for taking some concrete measures to meet these challenges.

Previously, digital forensics domain has helped to rejuvenate some trust in digital content. However, as the image forgery detection techniques are being developed, tampering of digital data despite leaving any noticeable trails has become very trivial. The challenge leads to issues such as image authentication, protection, and forgery detection. This demands aggressive counter approaches from scientists and researchers to confront and challenge malpractices.

B. Problem Description

Image tampering is a known handling technique [5]. Deception of typical image files is relatively a tedious task and requires sufficient expertise. However, digital images are disposed to tinkering. There exist numerous software applications to easily manipulate them. Malpractices mainly include duplication, replication, removing or exchanging parts of an image. It should be noted that originality of an analog data can be validated easily through a naked eye as any attempts to tampering can be conceived readily. Contrarily, development of supporting software tools has made manipulation of digital images a very easy task. For example, Fig. 1 highlights one such example. Originally, two objects were present in Fig. 1(a). The object on the far right is inserted as visible in Fig. 1(b). However, by looking at the figure through a naked eye, one cannot conceive that originality of the image had been compromised. Before taking an appropriate legal or social action in such cases, it is necessary to verify that an image had been edited. In such cases, as it is clear from the figure, validation of originality of a tampered image becomes very

challenging since alteration of a digital image can be carried out easily in comparison to a printed one.

As digital image domain is being revolutionized, tampering of a digital content without any noticeable impression has become very effortless. Therefore, to tackle the challenge, an image should be analyzed in such a way that even a slight attempt to forge can be detected straightaway. In this paper, we propose a light-weight automated technique that image owners and publishers can easily use to sign their images. The approach can also be used as an instrument to protect proprietary images from any possible forgery attempts.

Rest of the paper is organized as follows: following subsections of Section 1 highlight the contribution and the current state of the art in the domain. Section 2 describes the related work. Section 3 reflects upon our contribution in terms of the proposed technique and presents its usefulness through a software tool we developed to automate and demonstrate our work. In the end, Sections 4, 5 and 6 sum up with Automation of SAB - IOMHA, and Conclusions, respectively.

C. Contribution of the Research Project

Validation is a standard procedure for investigating integrity of an object. We want to achieve it in terms of forgery detection of a digital image through the proposed work. The decisive objective is to audit digital image files for originality and verify that their integrity has not been compromised since their authoring. The current approaches for the purpose have encompassed signature-based methods for protecting image files and checking for their integrity. However, such techniques are not applicable in wider settings because of their limitations or overheads involved in their use. On the other hand, as part of our work, we propose using a composite watermark which consists of a cipher along with date and time stamp and email address of the image author. The watermark is inserted in structured patterns to certain bits of an image file.

Digital watermarking is a known technique for media files for retaining copy-right information and identification of their proprietorship [4]. These can be of several types and are widely used. Generally, images can be inserted with at least two types of watermarks, visible watermarks or invisible watermarks as required. A visible watermark embeds an image file with an identification mark and an invisible one on the other hand inflicts a hidden mark in it. As part of this research, we choose the invisible watermark which we sequentially insert across multiple bits of a digital image. The contents and structure of the watermark is distorted if someone tries to edit the image file by any means.

In this research paper we provide more insight and extend Sab-iomha which we proposed previously [6], for its usefulness in the real settings. The extended version of the work reflects upon more technicalities of the technique and an improved validation mechanism. The ultimate objective of the research is to address the challenge of digital image forgeries.

D. Current State of the Art

ELA (Error Level Analysis) of an image can highlight any edited or distorted part of an image as different regions of an image having different compression levels can be identified. It enables the stakeholders to easily detect any problem areas through a naked-eye. Existing approaches to image forgery detection usually involve replicating those files to some dedicated software tools [7]. Users are then provided with different features of ELA and Joint Photographic Expert Group (JPEG) format. Our contribution is twofold. First, we split an image file to temporarily separate its metadata from the visual content and then steganograph the same image. An image file is composed of combination of pixels. An ordered set of bytes represents each pixel for different colors that constitute an image. Those colors include Alpha, Red, Green, and Blue. It should be noted that data is not stored in Alpha bits. Therefore, we propose use of those bits to insert the hidden watermark into the image file. Cipher, as part of the watermark, is invisible and is removed automatically upon any attempts to forge. As part of the second contribution of the research paper, we demonstrate the usefulness of Sab-iomha through automation in terms of a software tool we developed to augment the proposed technique. If an image file was saved multiple times, it loses its quality [8]. Metadata of an image file refers to the image itself. The information it contains may include the image type; e.g. JPEG, dimensions of the image, internal formats, and color scheme. The metadata also gives information such as the date of creation, the date of modification, name of the software editor that was used to create the image, file tags, and camera tags. It also provides information on the Exchangeable Image File Format (EXIF) which is used by the digital cameras manufacturers to extract camera settings that were used to capture the image. Camera settings entail information such as the manufacturer name and the model, time stamp, and lens settings. Those settings may vary among images to ensure maximum level of integrity. If a user tries to insert comments into an image file, they are incorporated into its metadata. Digital cameras normally do not allow automatic insertion of comments to the captured image. However, if any additions are found, it is an indication that the image has been edited or reprocessed using some software tool.

Majority of the existing approaches to image forgery detection take account of the information provided through metadata or the file header. Any attempts to get additional information while capturing a digital photo or any effort to change its header can easily render image handling more complex hence time consuming. In addition to that, the currently available techniques do not account for digital contents or file storage itself. On the other hand, our proposed technique addresses the challenge using a simple yet efficient mechanism; i.e. hidden watermark is embedded in an image which diminishes the need for manipulating with the file header. Sab-imoha ensures that any attempts to manipulate the image distort the watermark. Hence any successful bids to alter the image file can be discovered promptly.

II. Related Work

The literature review that was conducted to carry out this research encompassed image content, detection, and forensic analysis. We investigated different techniques currently in use for authenticating digital contents in terms of their traits as well as deficiencies.

Lighting, inconsistent shading, and shadows have been used as a method for collecting evidence on image forgery [5].

(a) (b)

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

Mixture of shadow and shading was rationally used to serve for the purpose and both were made dependent on each other but in case they are not, the corresponding image is found to be a tampered one. Furthermore, the authors reported reliable and specific shadings under different inferences of some subjective measures such as guess-work or acceptance. However, their proposed technique is not applicable in case such historical text documents do not make a shadow. Moreover, the research is applicable to those human images only that contain visible faces. It requires human interaction and the method that is used to estimate authenticity of an image is also prone to an estimation error.

Color discrimination has also been used as a mean to detect image forgery. To achieve that, some researchers have proposed a method called spliced image detection mechanism [9]. They detected illumination inconsistencies of an image by extracting edge or text-based features. If the image file under consideration carried information about image type, camera model, and motion after being captured, the data was found to be helpful for preventing any image forgery attempts by making the latter a difficult job [10]. However, detection of reflection-based forgeries is not a trivial task. A technique proposed by [11] suggests removing observable information from an image to make it trustworthy. Another method for detecting forgery in image files uses text-based signing of images [12]. If the digital signature gets distorted, it implies that integrity of the image had been compromised.

Thumbnails have also been used for verifying image files for authenticity [13]. The authors proposed creating thumbnails using contrast settings, compression, and filter models altogether which in turn are used to identify whether the actual images were compromised or not. Those models are then compared with the editing software and the originator cameras. A hidden watermark approach has also been used for image forensics [14]. It controls JPEG-lossy compression, cropping, and other possible operations that can be performed on an image by adding an invisible watermark in such a way that any distortion or a missing link in it indicates that the image had been forged.

The authors in [15] proposed an image forgery detection technique by investigating inconsistencies in lighting. Although lighting of a scene is not a complicated task, but it can be hard to match as the difference in lightings can be negligible. Researchers in [16] dealt with using a 3D lighting coefficient for image forensic. However, surface and lighting assumptions that are used are very specific. In addition to that, the challenge is to precisely estimate 3D shape of an image object.

A steganography technique to protect JPEG images from tampering by capturing two identical images instead of generating a secret text has also been discussed in [4]. The instance information is attached as a watermark to the actual image for the validation purpose. However, the proposed technique supports JPEG formats only and any slight change in camera settings between capturing images may also affect efficiency of the digital device.

Seam modification in digital images is another way of

image tampering. The former can be performed through a couple of ways; seam carving and seam insertion. In [17] the authors have studied modification of JPEG images through seam modification. A very minute change in seam effects the pixel ordering. A non-traditional method of machine learning, Classification Support Vector Machine, is used to intercept the seam-tampered image that differentiates between the tampered image and the original one. The problem with their proposed method is that it fails when highly imbalanced and skewed data sets are observed. The method is not applicable in diverse setting either.

Copy-move forgery (CMF) [18] is another common tampering technique in which a small part of the image is taken and copied to another location on the same image. Usually key-point based technology is used to detect this type of forgery, but it takes too much processing time and can run out of the memory while processing. Moreover, small cloned and smooth regions are difficult to detect. The author in [18] presents a new technique to overcome this problem. The test image is separated into smooth and rough regions and is further segmented into small regions. Before applying the Scale Invariant Features Transform (SIFT) algorithm, the customized parameters are detected for that specific image. If fixed parameters are selected to apply SIFT then results may not be satisfactory. Swarm intelligent (SI) algorithm was applied to generate a custom parameter for efficient processing of SIFT. The technique may reduce the processing time to avoid run out of memory. The experimental results indicate some higher false positive rate that needs to be improved.

In-painting [19] is another technique that has been used for forgery detection. It works by rebuilding the deteriorated part of an image. When an image gets scratched or fade away, some of its segments are reproduced to bring back its originality. The main theme was to copy segment of an image and embed it back on the scratched or deteriorated patches of the same image. The authors proposed a copy-move image forgery method in which an object is removed from an image and is pasted on a different location on the same image. Two in-painting techniques [19] were used to detect the object removal, geometry-oriented and texture-oriented. Their proposed technique, which was referred to as exemplar-based image in-painting, reported significant decrease in search time for image blocks. However, it is not very useful for multiple object removals as it increased the search overhead.

A steganography technique to protect JPEG images from tampering proposed capturing two identical images instead of generating a secret text [20]. The instance information was attached as a watermark to the actual image for validation purpose. However, their proposed technique supports JPEG formats only and any slight change in camera settings between capturing of images may also affect the efficiency of the system.

In summary, the existing approaches to counter image manipulation lack the diversity required to confront the challenge. Due to rapid rise in use of digital images, attempts to compromise their integrity are also on the rise despite currently available mitigation techniques. As it is evident from analysis of the literature, there exist no single technique that is easily applicable and equally useful to multiple types of digital images consistently; that is, computer generated images, digital documents that are saved as image files, and digital camera images. The situation calls for proposing more robust methods to confront the challenge. Researchers need to come up with effective forgery detection solutions to address the issue.

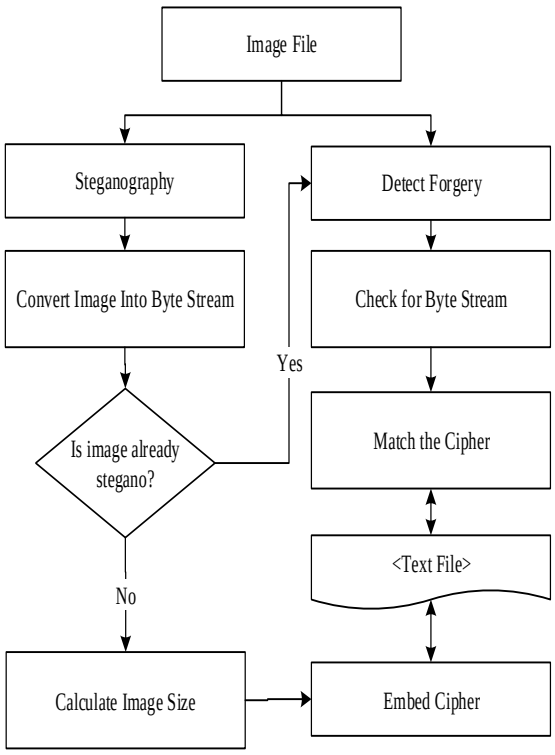


Fig. 2. An overview of Sab–iomha [6].

III. SAB-IOMHA:THE PROPOSED TECHNIQUE

There are two phases of this research work; steganography and forgery detection. We propose a forgery detection mechanism which is a two-step approach as shown in Fig. 2. An image file is protected using an invisible watermark and then any forgeries are detected by investigating the same watermark which was inserted in the first step. As part of the approach, firstly the image is converted into byte stream that splits metadata from the file. Secondly, an invisible watermark is inserted in certain bits of the image. The watermark is in text form and can be inserted across multiple bytes. However, its length depends upon size of the image; bigger the image in size lengthier would be the watermark.

A digital image can incorporate two types of watermarks; visible watermark or invisible watermark depending upon user preferences. Visible watermark inflicts small spots on the whole image whereas the invisible one randomly inserts a text code in it. Fig. 3 is a pictorial representation of the visible watermark technique. It demonstrates different states of an image.

Visible watermarks were inserted that are noticeable by zooming the image. An ELA can identify regions within an image that possess different compression levels. It is a measure

to visually highlight difference in JPEG compression levels across different regions of an image.

Since we make use of invisible watermark, the inserted text would be hidden. We suggest composing a composite invisible watermark which is composed of multiple information fields that makes it easy to validate an image. Those fields entail cipher text, email address of the image user, and date and time stamp. At the same time the composite watermark ensures that the ownership trail of the image is maintained for any future reference as well to preserve edit history of the file. Furthermore, as part of the watermark, the cipher changes automatically if someone tries to edit the signed image as any attempts to doctor it would distort the cipher part of the inscription.

For a JPEG format, the entire image should represent the same ELA but if some fragments of an image carry different error levels, it is an indication that the original image was edited for an unauthorized modification. Regions with even coloring, like a blue or a white wall, would likely have a lower ELA levels in comparison to dark colors having high-contrast edges. For a typical forgery detection, one would check the image and try to figure out the difference between high and low contrasting edges and compare those with the ELA representation. Only a visible difference allows a naked-eye to detect any contemporary changes that might have been made to the image. Therefore, a sole ELA-dependent method is not a good fit to detect any such images which are digitally modified.

(a)

(b)

Fig. 3. Original image. and after applying a visible watermark. (b) Zoomed-in one to enhance visibility and an ELA version of the image.

In a 32-bit image that spans across four channels of colors, each pixel is constituted of four bytes. Each one of the three colors; i.e. Red, Green, and Blue is epresented by a byte each as shown in Fig. 4. However, the fourth byte which is known to be reserved for Alpha does not represent anything and is available for use. To date several systems have been proposed that represent pixels in terms of supporting colors but an ARGB is the most established arrangement for representing colors. It logically arranges a pixel in an order of Alpha, Red, Green, and Blue. As part of our composite watermark technique, we make use of the least significant bit of Alpha to steganograph an image file. This does not change data stored in any bit but text length should be calculated before it is inserted in the image file as a watermark.

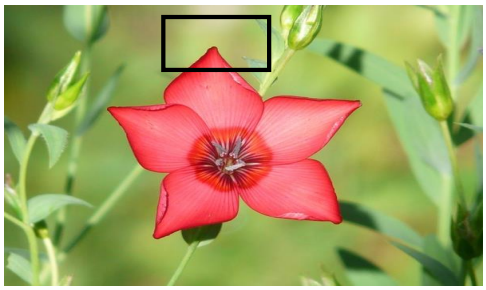
Algorithm 1 Embed watermark

Require: $x \geq key * 10 \vee x \neq 0 \quad I = 0 || I = 10 || I = 100$

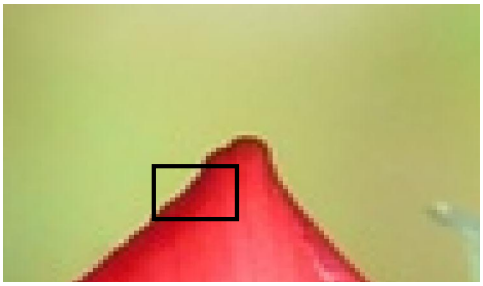
$P \leftarrow$ *readImagePixels*
$P = P_0, P_1, P_2, ...P_n$
$Dt \leftarrow$ *getCurrentDateTime*
$E \leftarrow$ *getEmail*
$Key = \{M_0, M_1, M_2, ...M\}$
$x \leftarrow key + Dt + n$
$x \leftarrow$ *floor* $(\frac{x}{k(k+1)})$ equation 1
function MATCHCIPHER(key, P)
  if $found$ **then**
    $return$
  end if
end function
function INSERTCIPHER(x, P)
  function INSERTEMAILANDDATETIME(Dt, E, P)
    for $j = 1$ to $j = 8$ **do**
      StegPixel
    end for
  end function
  for $i = Dt + E$ to $i = x$ **do**
    for $j = 1$ to $j = 8$ **do**
      StegPixel
    end for
      $x \leftarrow x + I$
  end for
end function

Fig. 4 demonstrates how exactly our proposed technique makes use of certain bits of an image file. It splits metadata from the file header. The file is then converted into pixels which in turn is transformed into a byte stream. Alpha bits are selected, and an invisible watermark is inserted into them, which is a composition of cipher, email address, and time and date stamp. If we consider an image as a matrix P having m rows and n columns, total number of pixels in it can be determined using the given m n relation.

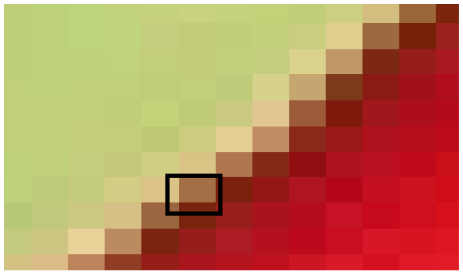
We argue that inserting watermark into the least significant bit is an easy yet effective approach for signing an image with the traceable information. Eighth bit of the Alpha bytes is utilized for the purpose; i.e. one bit of the overall size of the inserted watermark. It should be noted that we do not make use of all Alpha bytes of an image file. Their selection is based on a certain pattern which is generated at run time to ensure maximum protection of the image. For a four-byte image having thirty-two bits, the least significant bit of the Alpha component is utilized which is depicted as the marked bit of a pixel shown in Fig. 4(d). An image consisting of 800 600 pixels can store up to 1,440,000 bits or 180,000 bytes of watermark. For instance, a block of 8 pixels of a 4-byte image can be represented as: if number 35 is inserted as a watermark



(a) Original Image with highlighted area.



(b) Zoomed – in highlighted area of (a).



(c) Ultra zooming of highlighted area in (b).

(d) Least significant bit of an alpha byte of pixel.

Fig. 4. Illustration of an image pixel and the corresponding bit used for the invisible water marking.

having binary representation 00100011 across Alpha bits of an image, the resulting pixel block gets manipulated in such a way that 35 is accommodated in consecutive pixels highlighted as shaded pixel bits in Fig. 5. It is worth mentioning that only least significant bits of Alpha bytes are inserted with the watermark fragments. All pixels can be protected using the scheme which does not affect the visual contents of the image file. Since the proposed technique consumes an image at the structural level, its steganography cannot be observed through a naked eye.

Email	Date Time	Intensity	Cipher length
6-255	7	1	n * key ...

$$(y-1)k + (a-1)k^2 < N < (y-1)k + ak^2 - 1 \quad (1)$$

In a 32- bit colour image, Alpha bits are separated, and the code stream is spread across the byte stream using Algorithm 1. Where x is the number of pixels in an image, I is intensity of the watermark which can be 10, 50 or 100, and Key is length of the cipher. P is an array of pixels which an image file contains.

Dt is the current date and time of the system. E is email address of the user. At line 7 of the algorithm, x is cumulation of the composite watermark obtained by adding cipher text, date and time stamp, and email address. The cipher text constitutes the constant part of the watermark whereas rest is the system and user dependent to enhance the strength of the algorithm. The function at line 8 checks the image file for the watermark, if matched, the image is authenticated. Otherwise, InsertCipher procedure at line 13 is initiated. The cache space can be increased to any positive numeric value in case we want to add an interval between the bytes that are occupied by the ark.

There could possibly be a case that someone else signs the image after it was steganographed by the actual author. The situation makes it nontrivial to keep track of the actual ownership. The combination of date and time in particular ensures that once a user signs the image, the ownership trail can be maintained for the subsequent detection of any successful forgery attempts. Table I illustrates the composition of the composite watermark. Email address of the user is allocated up to 255 bytes, date and time is allocated 7 bytes, 1 byte for Intensity which is the distance between two nearest cipher bytes, and variable number of bytes are reserved for the Key which points to the cipher text. The following equation I is used for determining the length of the cipher.

$$P = \begin{bmatrix} P_{11} & P_{12} & P_{13} & \dots & P_{1n} \\ P_{21} & P_{22} & P_{23} & \dots & P_{2n} \\ \dots & \dots & \dots & \dots & \dots \\ Pm1 & P_{m2} & P_{m3} & \dots & P_{mn} \end{bmatrix} = (p_{ij})_{m \times n}$$

Where a is any positive integer and y is the cumulative length of characters of email address and date and time stamp, K is constant length space allocated for the cipher text to be impeded in the image, and N represents length of the image in bytes.

IV. Automation of Sab - Iomha

The software tool that we developed to automate our research is relatively simple and user friendly with minimum

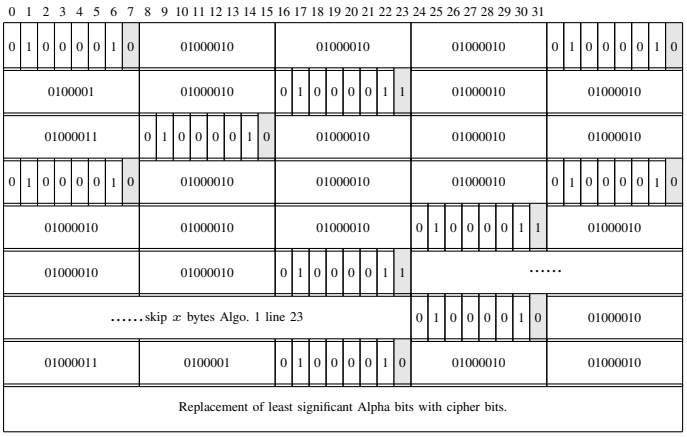


Fig. 5. Least significant bits of Alpha bytes of an image.

of work-flows. It supports browsing of an image file using a GUI interface and is loaded in computer memory.

Fig. 6 depicts user interface of the tool we developed. It was programmed using Java technologies. The ultimate objective is to facilitate validation of digital images and documents as well in case they are in an image format to prove integrity of the contents or to verify that the digital document has not been edited since its creation. The tool supports multiple features as shown in Fig. 6. The Steg Image embeds an invisible watermark in the image. The steganographed image can also be saved on the disk for any future reference. Forgery Detection opens up another screen as depicted in Fig. 6.

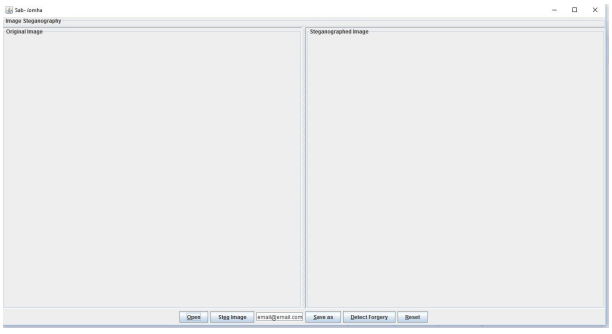
Signing an image file is a two step procedure: in the first phase, we would steganograph an image by inserting the invisible watermark which is validated for integrity in the second phase. We randomly pick an image and upload it to the tool to demonstrate usefulness of our technique as well as the overall automation itself. The sample image on the right side of the Fig. 7 is signed using the watermark which is the composition of cipher text, email address, and date and time stamp. It can be observed that quality of the image was not compromised at all by using the technique. The same file can be checked to verify if the image is original or any attempts has been made to alter it. In case the validation procedure generates an alert text, which is the case as shown in Fig. 7, it is an indication that the image has been forged by some other user. Otherwise, the inserted watermark is displayed to testify the originality of the image. Algorithm 2 enlists steps performed to detect forgery. It is a three-step procedure; in the first one, it looks for an insertion, if not found, it implies that the image is not steganographed. If an insertion is found, it is matched with the actual watermark. If the exact match is not found, the image is reported to be forged. Otherwise, it is the original one.

To further validate the proposed technique, we performed an experiment to demonstrate its effectiveness. A set of images with varying range of size was steganographed using the tool we have developed to automate Sab-iomha. The motivation was to compare metadata of the image files before and after the technique was applied. We considered certain factors like size, compression level, and resolution to investigate the subject. Each image had 4 color channels having 32 bits altogether,

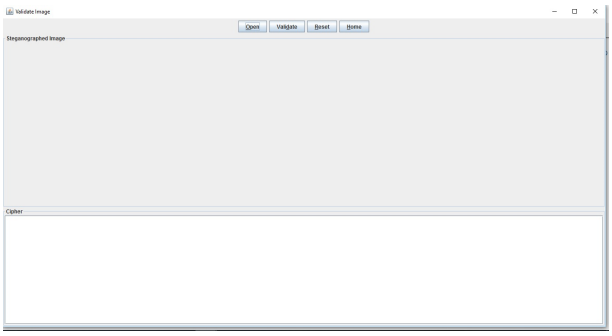
Algorithm 2 Forgery Detection

Require: key , $image$
$P \leftarrow$ *readImagePixels*
$P = P_0, P_1, P_2, ...P_n$
$Key = \{M_0, M_1, M_2, ...M\}$
function MATCHCIPHER(key, P)
  if $found$ **then**
    if $key = extractedCipher$ **then**
      Image is original
    end if
    if $key \neq extractedCipher$ **then**
      Image is forged
    end if
    $Key = ImageCipher$
    Original Image
    $return$
  else if
    thenImage is not protected
  end if
end function

and 0.27 value for mega pixels. Table I reflects upon the image population in more detail.



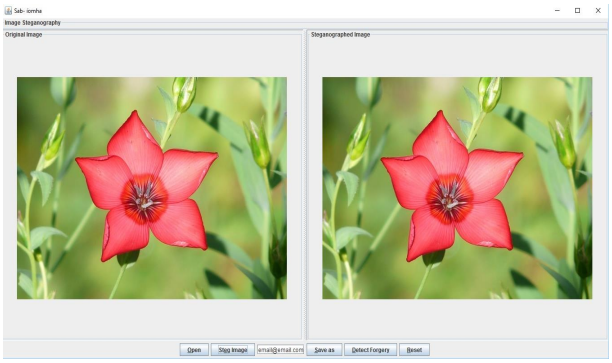
(a)



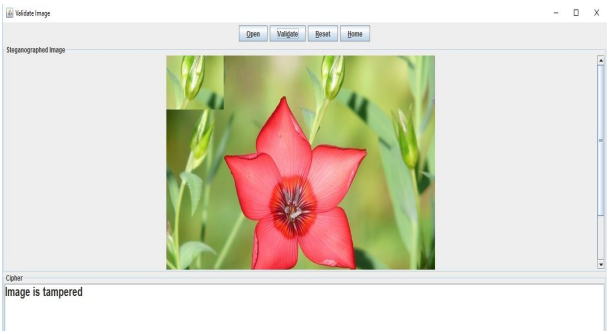
(b)

Fig. 6. (a)Home-interface of the tool implementing Sab - iomha. (b) To detect an image le for forgery [6].

Table I draws comparison between metadata of the image files before and after applying the steganography using Sab-iomha. It is noticeable that color type remained the same even after each image was steganographed, that is, RGB with Alpha. There was no change in resolution of the images either. However, some difference was observed in terms of size of each image. In general, the steganographed images were noted



(a)



(b)

Fig. 7. Home-interface of the tool with an image loaded and steganographed.

TABLE I. POPULATION OF THE IMAGE FILES FOR EXPERIMENTATION

	Original Image		Processed Image	
No.	Size (Kilo Bytes)	Resolution	Size (Kilo Bytes)	Resolution
1	655	600x450	648	600x450
2	291	457x360	435	457x360
3	511	600x450	502	600x450
4	914	1280x1012	1152	1280x1012
5	129	262x192	129	262x192
6	317	425x281	313	425x281
7	1238	1024x768	1168	1024x768
8	89	284x177	90	287x177
9	726	700x350	725	700x350
10	1525	1024x750	1492	1024x750
11	393	476x500	366	476x500
12	136	276x183	135	276x183
13	590	600x450	581	600x450
14	364	500x334	358	500x334
15	158	259x194	159	259x194
16	139	259x194	139	259x194
17	129	275x183	128	275x183

to be slightly smaller in size. The overall analysis suggested that quality of each set of images remained the same, i.e. studying the metadata before and after the application of the forgery detection technique did not negatively influence the quality of the images under consideration.

V. CONCLUSION

Digital images are prone to forgery in the current age as it has become much easier to manipulate digital contents due to advancement in the domain. We have introduced a new dimension to the digital image steganography by proposing a light weight technique. It uses a composite watermark to check digital images for authenticity. The proposed technique signs digital images for integrity and protects them against any manipulations. The forgery issue is addressed in a novel way; ELA, JPEG, and metadata are incorporated, and an invisible watermark is inserted to enhance efficiency and effectiveness of forgery detection. The proposed technique is automated through a software tool which facilitates users to steganograph digital images. The same image can then be checked for originality. The core purpose of the tool development is to support the usability of Sab-iomha which may not only validate photographs but also any digital contents stored in an image format. This work enables even non-technical users to be able to investigate integrity of image files at their own. It also empowers them to get insight on their digital contents. As part of the validation mechanism, we have tested the algorithm on a series of random images. The results suggested that the technique can not only verify the digital images for authenticity but also does not negatively influence their quality. Moreover, users can also protect their images from any attempts to forge. The research we conducted do not have any ethical, moral and legal issues associated with it. The project is economically feasible too as the users do not require to purchase any hardware devices and are alleviated from the need for software installations. Currently, the work is aimed at supporting JPEG and PNG file formats only. We aim to extend support for other image formats in the future.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their feedback which helped to improve the earlier version of the research paper.

REFERENCES

[1] C. Harris and M. Stephens, “A combined corner and edge detector.” in *Alvey vision conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

[3] V. Schetinger, M. M. Oliveira, R. da Silva, and T. J. Carvalho, “Humans are easily fooled by digital images,” *CoRR*, vol. abs/1509.05301, 2015. [Online]. Available: http://arxiv.org/abs/1509.05301

[4] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, “Secure spread spectrum watermarking for multimedia,” *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, Dec 1997.

[5] T. J. d. Carvalho, C. Riess, E. Angelopoulou, H. Pedrini, and A. d. R. Rocha, “Exposing digital image forgeries by illumination color classification,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 7, pp. 1182–1194, July 2013.

[6] M. S. Bhatti, S. A. Hussain, A. Qayyum, I. Latif, M. Hasnain, and S. I. Hashmi, “Sab - íomha: An automated image forgery detection technique using alpha channel steganography,” in *Recent Advances in Information Systems and Technologies*, Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, and S. Costanzo, Eds. Cham: Springer International Publishing, 2017, pp. 736–744.

[7] H. Farid, "Image forgery detection," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16–25, March 2009.

[8] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "A variational approach to jpeg anti-forensics," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 3058–3062.

[9] H. F. Matthias Kirchner, Peter Winkler, "Impeding forgers at photo inception," pp. 8665 – 8665 – 9, 2013. [Online]. Available: https://doi.org/10.1117/12.2008412

[10] J. F. O'Brien and H. Farid, "Exposing photo manipulation with inconsistent reflections," *ACM Trans. Graph.*, vol. 31, no. 1, pp. 4:1–4:11, Feb. 2012. [Online]. Available: http://doi.acm.org/10.1145/2077341.2077345

[11] V. Conotter, G. Boato, and H. Farid, "Detecting photo manipulation on signs and billboards," in *2010 IEEE International Conference on Image Processing*, Sept 2010, pp. 1741–1744.

[12] E. Kee and H. Farid, "Digital image authentication from thumbnails," in *Media Forensics and Security II*, vol. 7541. International Society for Optics and Photonics, 2010, p. 75410E.

[13] C.-T. Hsu and J.-L. Wu, "Hidden digital watermarks in images," *IEEE Transactions on Image Processing*, vol. 8, no. 1, pp. 58–68, Jan 1999.

[14] W. Luo, Z. Qu, F. Pan, and J. Huang, "A survey of passive technology for digital image forensics," *Frontiers of Computer Science in China*, vol. 1, no. 2, pp. 166–179, May 2007. [Online]. Available: https://doi.org/10.1007/s11704-007-0017-0

[15] M. K. Johnson and H. Farid, "Exposing digital forgeries in complex lighting environments," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 450–461, Sept 2007.

[16] W. Fan, K. Wang, F. Cayre, and Z. Xiong, "3d lighting-based image forgery detection using shape-from-shading," in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, Aug 2012, pp. 1777–1781.

[17] K. Wattanachote, T. K. Shih, W. Chang, and H. Chang, "Tamper detection of jpeg image due to seam modifications," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2477–2491, Dec 2015.

[18] F. Zhao, W. Shi, B. Qin, and B. Liang, "Image forgery detection using segmentation and swarm intelligent algorithm," *Wuhan University Journal of Natural Sciences*, vol. 22, no. 2, pp. 141–148, Apr 2017. [Online]. Available: https://doi.org/10.1007/s11859-017-1227-4

[19] Z. Liang, G. Yang, X. Ding, and L. Li, "An efficient forgery detection algorithm for object removal by exemplar-based image inpainting," *J. Vis. Comun. Image Represent.*, vol. 30, no. C, pp. 75–85, Jul. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.jvcir.2015.03.004

[20] T. Denemark and J. Fridrich, "Steganography with multiple jpeg images of the same scene," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 10, pp. 2308–2319, Oct 2017.