Reducing Falsely Detected JPEG's Fragmentation Point Using Unique Hex Patterns (UHP)

Nurul Azma Abdullah, Rosziati Ibrahim, Kamaruddin Malik Mohamad, Norhamreeza Abdul Hamid
Fac. of Computer Science & Information Technology
UTHM
Batu Pahat, Malaysia
e-mail: azma, rosziati, malik @uthm.edu.my

***Abstract*— To increase number of evidence obtained in an effort to fight cyber perpetrator, many studies have been conducted in addressing problem of fragmented JPEG images. However, thumbnail/s is/are always been mistaken as fragmentation point of fragmented JPEG images. This paper discusses on how thumbnail can be used to reduce false detection. The main contribution of this paper is introducing patterns to distinguish the header of original JPEG images with thumbnail/s.**

Keywords-component; Fragmented JPEG files; File carving; Thumbnails; Pattern Matching; DFRWS 2006/2007.

I. INTRODUCTION

Carving, one of important fields in DF is used to explain the process of extracting a raw image from unstructured digital forensic images based on the content rather than using file system metadata for data recovery and computer forensics [1], [2], [3]. Simply, file carving is a process of recovering files from the unallocated space of disk without knowing the file system [4].

There are two types of files structures that normally found in a dataset which are non-fragmented file and fragmented file. Although many carving tools available today concentrate on non-fragmented file, the importance of carving fragmented files has been asserted in [3], [5]. A signature in header and footer has been used to carve in straightforward carving. This is a simple technique and has been proved successfully carve a contiguous files with an assumption that files clusters remain in order [4]. However, if files are fragmented, files can be disconnected and becomes unordered which cause the straightforward carving fail.

Statistic presented in [3] shows the fact that fragmentation in today's file system is relatively infrequent. However, the capability of carving fragmented files which is not extensively explored is important for computer forensic because the possibility of files that interest forensic investigation to be fragmented is relatively high [3], [5].

This paper concentrates on recovering JPEG images with thumbnail/s as a way to recognize real fragmented JPEG files. Fragmentation will be detected in two situations, a complete JPEG image that contains thumbnail/s or a non-complete JPEG image concatenates with another JPEG image. In other word, A JPEG file is said to be fragmented when a JPEG header found after the first JPEG header and before the JPEG footer. However, a fragmentation caused by thumbnail/s is/are not the real fragmentation scenario. While dealing with fragmentation, file carvers will normally separate the first fragment from the second fragment to be reassembled with the correct fragment. If, this task is applied to a file with thumbnail/s, error will occur during decoding process. Hence, it is important to exclude thumbnail/s during detecting fragmentation process. Furthermore, knowledge of thumbnail's existence helps investigator to concentrate in investigating correct point where the real fragmentation occurs.

The rest of the paper is organized as follows. Section 1 is the related works consists of an overview of JPEG standard and thumbnail and fragmentation while Section 3 brief on ThembFrag Model. Section 4 describes the experimentation done. Section 5 is result and discussion. Finally section 6 concludes this paper.

II. RELATED WORKS

A. An Overview of JPEG Standard

Computer forensics is to recover evidences resides on a computer, by mean to solve pornography cases [1], [6], [7]. This involves image files obtained from the perpetrator in certain format like Bitmap and JPEG but most common format is JPEG. JPEG is popular because of its compressed file that can reduce the size required to allocate an image. Joint Photographic Experts Group (JPEG) was formed by International Telegraph and Telephone Consultative Committee in 1986 inspired by an effort of International Organization of Standard (ISO) to find ways to use high resolution graphics and pictures in computers [2]. JPEG introduced compression standard for both grayscale and color continuous-tone images. The details of JPEG compressed data formats can be found in [8]. There are two types of JPEG that are mostly used today, JPEG File Interchange Format (JFIF) and JPEG Exchangeable Image File Format (Exif) [9]. JFIF is popular for internet file while EXIF is the popular image file format used for digital camera [10].

Both in JFIF and Exif format allow for embedding thumbnail/s into a JPEG file. A JPEG image with a complete SOI/EOI can be embedded into an original JPEG image to ease the recovering and organizing of the original image. This file is known as thumbnail. Thumbnail is reduced size version of images that can be used to recover

and organize the picture [11] while embedded JPEG file is referred to an original JPEG file that are embedded to other types of files such as PPT, WORDS and EXCEL. Thumbnail is used to speed up images search or page load on the Internet and also being used in image organizing programs. Thumbnail is compatible on most modern operating systems or desktop environments such as Microsoft Windows, Mac OS X, KDE and GNOME [12]. A JPEG image can contain none or a single or two thumbnails. Therefore, a JPEG image can have several SOI/EOI pairs [13]. Mohamad in [14] and [15] asserted the role of thumbnail to serve as a method of recognizing the corrupted images because of its small size that have a better chance for full recovery without corruption [16]. A thumbnail carried similar features as the original. Hence, using thumbnail/s, crime investigators can identify which images or pictures that have potential to be used as evidences against cyber perpetrator.

Guo in [11] proposed thumbnails as a method to recover JPEG image from fragment data. In brief, thumbnails do serve multiple roles. Besides contributing in the process of recovering and organizing JPEG files, thumbnails help in recognizing corrupted images and also, information about thumbnail's location can be used in carving fragmentation JPEG images to recover the original files. Abdullah et al. [17] proposed PredClus as a method to recognize thumbnail/s and embedded JPEG files. However, using PredClus which using cluster size to determine the location of thumbnail/s or embedded file may miss some thumbnails that resides at the start of cluster. This situation occurs when a JPEG image with thumbnail/s require more than one cluster to store the data. Sometimes, the start of thumbnail will be at the start of second cluster. In this situation, the thumbnail/s will be ignored by PredClus. Hence, an alternative technique to distinguish thumbnail or embedded JPEG file with the original is by using pattern matching technique. In carving JPEG images especially fragmented JPEG files, it will ease the process of preparing evidence if the carver can distinguish between original images, thumbnails and embedded images.

B. Fragmentation Points

Fragmentation point exists only when a file is fragmented into more than two parts. There are three approaches that have been discussed in [18] which are syntactical tests, statistical tests and basic sequential validation. Syntactical tests are when the fragmentation point is detected by validating the belonging of a block to a file through one of following methods:

- Using keywords and signatures to identify different file types
- Content analysis to identify incorrect block.

Using this method, it can be confirmed that the validated block does not belong to any certain file. However, it is not certain that the previous block belongs or not belongs to the file. For statistical tests, the statistic of each block is compared to a model of each file type to identify the block. Cohen [2] used the mapping function to map between the bytes contained in the file to the bytes within the image itself. In this case, carving process is the process to estimate the mapping function. Statistical tests also facing problem in detecting the actual fragmentation point and even worse, using this technique, blocks can be falsely identified as belonging to another file type.

The third technique is basic sequential validation. This technique is used to identify fragmentation point by validating block sequentially from the header through the blocks until the validator stops with an error. Using this technique, a last correctly validated block is marked to be the fragmentation point. However, this technique can result in incorrect recovery of a file because it can successfully validate random blocks of data.

Alternative technique to identify fragmentation is by using cluster information. In a computer file system, a cluster or in DOS 4.0 known as an allocation unit or in UNIX System as block, is the smallest logical set that is created to perform actual erasure for files and directories [19], [20]. A cluster may contain the whole file or portion of files but a cluster only stores data for one file [21], [22]. Therefore, it is important to determine the cluster's size to determine the start of file. This information is useful for both steganography and file carving.

In file carving, the whole content of the hard disk used for evidence need to be imaged (will be referred to as image file) in preparation for forensics investigation. The image file contains thousands of hex code representing all files in the hard disk used for evidence. It is impossible to read line by line manually in order to extract all files into their original form. Information about the start of file can be used to identify each file that resides in any dataset.

PredClus as proposed in [17] is developed to automatically display the predicted cluster size according to probabilistic percentage. It concentrates on JPEG images. The information can be used to distinguish original with thumbnail/embedded JPEG files which can help to determine the real fragmentation point. A thumbnail can be easily mistaken for another JPEG file fragmented with the original. Once thumbnail is recognized, the carver can easily know which point is the real fragmentation occurs. However, this technique has limitation. For some rare cases where thumbnail is pushed to the start of cluster, it can be falsely identified as an original JPEG image.

III. THUMBFRAG MODEL

This section discusses on the experiment designed for ThumbFrag model (as illustrated in Figure. 1) is adapted from myKarve [Mohamad et al.]. Two algorithms, PattrecCarv [23] and Pattern_temp are installed into the model. PattrecCarv accepts data sources from either image from physical memory, removable storage, hard disk or any JPEG file. However, for this particular experiment, we use dataset from DFRWS 2006 and 2007. This algorithm also searches for JPEG files location and store in ADB (Address Database). The other algorithm, Pattern_temp matches the header and footer with predefined patterns for original, thumbnails and embedded JPEG files. PattrecCarv is developed using C++ language while Pattern_temp is

implemented in Matlab environment to simplify the pattern matching process.

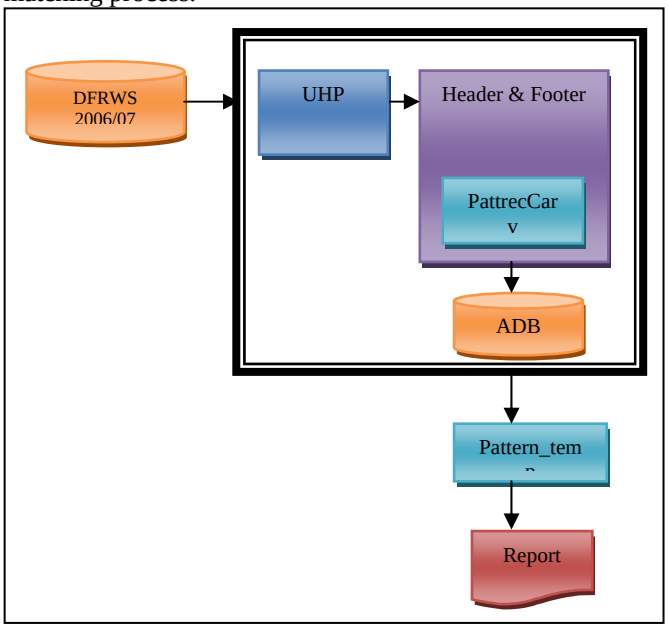

Figure 1. ThumbFrag model

The following are some of the assumptions for this experiment:

- Baseline JPEG is being used for the experiment because of its popularity and simple file structure.
- Only JFIF and Exif format can be accepted by this model. JPEG 2000 is not compatible with this model.

All the file headers and footers are in sequential order and not corrupted.

A. PattrecCarv

PattrecCarv is inserted in ThumbFrag to identify thumbnails and embedded JPEG files. A thumbnail in JFIF and Exif format can be recognized using UHP as described in [23]. The algorithm of PattrecCarv is introduced (illustrated in Figure. 2) to read raw data from DFRWS 2006 and 2007 dataset and mark thumbnails and embedded JPEG files with a different marker from original marker.

```
1.  Read data image
2.  Initialize hex_values
3.  Initialize thumb_markers
4.  Initialize embedded_markers
5.  Find JPEG header
6.  If found
7.          Jump to 9th hex_values
8.          If hex_values== embedded_markers
9.                  Read CurrentOffset
10.                 Save the CurrentOffset
11.         Else
12.                 Read next hex_values
13.                 If hex_values==thumb_markers
14.                         Go to the most recent SOI
15.                         Save the CurrentOffset
(location of SOI)
16.                 Endif
17.         Endif
18. If not end of data image, repeat step 5
19. Generate report
```

Figure 2. Algorithm used in PattrecCarv for carving thumbnails and embedded JPEG files

First, data from dataset is read. These data are in hex values. The hex values then matched with the standard JPEG header. However, in this experiment, additional markers are also used instead of standard JPEG headers and footers, 0xFFD8 alone. The additional markers which are 0xFFE0, 0xFFE1, 0xFFE2, 0xFFC4 and 0xFFDB and standard headers/footers are known as validated markers. The validated headers are used to reduce false detection of JPEG files. When matched, the offset for each markers matched is retrieved. Using UHP as described in [Abdullah], all thumbnails and embedded JPEG files are identified and standard headers for the files are renamed with a special name to indicate the type, either thumbnail (TN_FFD8) or embedded JPEG file (EF_FFD8). This is to differentiate them from original header (FFD8).

After all thumbnails and embedded files are determined, all locations for JPEG standard headers/footers are then stored in ADB (Address database). The markers and locations will be the input to Pattern_temp.

B. Pattern_temp

Pattern_temp acts as pattern matcher to identify original, thumbnails and embedded JPEG files. First, it reads ADB for headers and matches the header with predefined patterns. So, from the output, we can view different patterns of JPEG files, either JPEG file without thumbnail, JPEG file with one thumbnail or JPEG file with two thumbnails. If we found a pattern where two headers (original JPEG file) found consecutively, this indicate fragmentation scenario. Below is the algorithm for Pattern_temp (as illustrated in Figure.3) which is according to patterns template shown in Figure.4.

1. Read ADB
2. If match ‘pattern 1’
   a. Write to pattern 1 column in output file.
3. Endif
4. Else if match ‘pattern 2’
   a. Write to pattern 2 column in output file
5. Endif
6. Else if match ‘pattern 3’
   a. Write to pattern 3 column in output file
7. Endif
8. Else if match ‘pattern 4’
   a. Write to pattern 4 column in output file
9. Endif
10. If not end of data image, repeat step 1
11. Exit

Figure. 3. Algorithm used in Pattern_temp.

Pattern1 : Original without thumbnail

FFD8	FFD9

Pattern5: Original with one tnail

FFD8	TNFFD8	FFD9	FFD9

Pattern6: Original with two thumbnail

FFD8	TNFFD8	FFD9	TNFFD8	FFD9	FFD9

Pattern2 : JPEG files with second header found before footer

FFD8	FFD8	FFD9	FFD9

Figure 4. Patterns for different scenarios of JPEG files

IV. Experimentation

Generally, there are three major steps involved in ThumbFrag model to complete this experiment namely pre-processing, pattern matching and JPEG image file carving.

During the pre-processing, a dataset is read and searched for JPEG headers and footers. The real concern is to automatically determine the correct header-footer pairing to match the predefined pattern. In this model, a header-footer is paired if the validated header-footer is strictly matched one of the introduced UHP. This is to validate the read data are belongs to a JPEG file. Only then, with the addition of special markers to reduce false detection, these headers and footer along with their locations are populated in ADB. Next, the ADB is ready for pattern matching process.

During pattern matching process, all headers and footers are read from ADB. These headers and footers then matched with the predefined patterns namely pattern 1, pattern 5, pattern 6 and pattern 2. Pattern 1 is a pattern for JPEG image without thumbnail, pattern 5 is for JPEG image with one thumbnail, pattern 6 is for JPEG image with two thumbnails and pattern 2 is for JPEG image where two headers found before footer.

Finally, the last step is carving JPEG images according to patterns described above. Once, a pattern is determined, the headers and footers are stored in an output file along with their locations. For this experiment, the output file is in Excel format to simplify the analysis process later. All patterns are organized in different columns so that it is easy to view and read.

V. Result And Discussion

The screenshot of the output can be clearly examined in Figure 5 and Figure 6. From the figures, we can see that for original JPEG image without any thumbnail (Figure 5), the header and footer are consecutively but for other patterns, two headers found before footer. Without a special marker for thumbnail’s header, it will difficult to identify whether the second header belongs to another JPEG file or thumbnail.

When we differentiate thumbnail’s header from the standard JPEG header, we can easily confident that pattern2 is capturing fragmented JPEG images, where a non complete JPEG image is followed by another JPEG image. This scenario will cause distortion when viewed in image viewer. Now, the investigator can concentrate on these fragmented image instead wasting time investigate a complete image that is mistaken to be fragmented JPEG images.

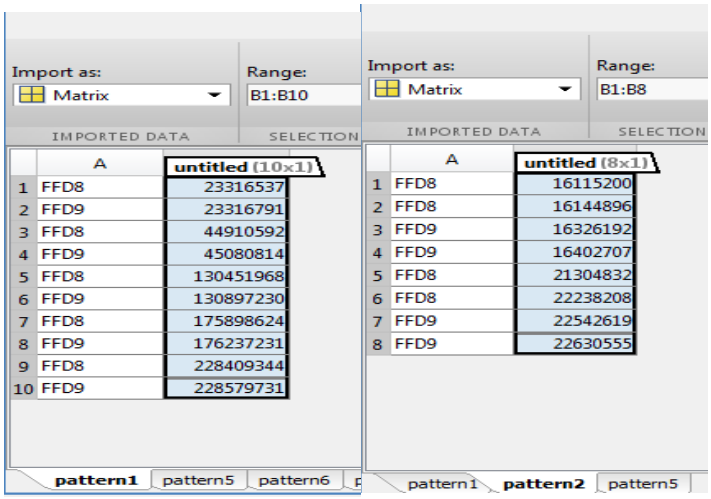

Figure 5. Example of output for pattern 1 and pattern 2

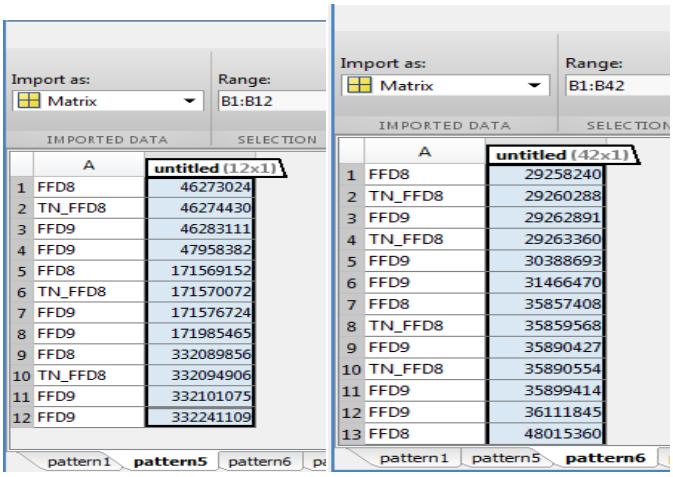

Figure 6. Example of output for pattern 5 and pattern 6

VI. CONCLUSION

A JPEG image can contains none, single or two thumbnails in the image itself. A thumbnail which is a reduced version of an image carried similar feature as the original. This thumbnail is always mistaken as another JPEG image. Therefore, knowledge of thumbnail's existence helps investigator to separate JPEG files with thumbnail/s and concentrates to investigate correct point where the real fragmentation occurs. They can then identify which JPEG image is fragmented with another JPEG images. With this way, they can ascertain that those fragments are belongs to another JPEG file, not file/s (thumbnail) within a JPEG file. This is important because during the reassembling process, if a thumbnail is mistakenly identified as another JPEG files, the original file may corrupt because of missing fragments and also wasting investigator's time in looking for fragmentation while the fragmentation does not exist. Subsequently, this is also accelerating the reassembling process by allowing investigators to concentrate on real fragmentation situation. In conclusion, by recognizing thumbnail, false fragmentation detection can be reduced significantly.

ACKNOWLEDGMENT

The authors would like to thank Ministry of Science, Technology and Innovation (MOSTI), for granting Science Fund (Vote s019) to support this research.

REFERENCES

[1] S. L. Garfinkel, "Digital Forensic Research :The next 10 years," Digital Investigation, vol. 7(1), 2010, pp. S64-S73.

[2] M. I. Cohen, "Advanced Carving Techniques," Digital Investigation, vol. 4(1-4), 2007, pp. 119-128.

[3] S. L. Garfinkel, "Carving Contiguous and Fragmented Files with Fast Object Validation," Digital Investigation, vol. 4(1), 2007, pp. S2-S12.

[4] C. J. Veenman, Statistical Disk Cluster Classification for FIle Carving. Proc. of the Third International Symposium on Information Assurance and Security, Machester, 2007.

[5] K. M. Mohamad, M. Mat Deris, Fragmentation Point Detection of JPEG Images at DHT Using Validator. Proc. of the 2009 FGIT, 2009, pp.173-180.

[6] A., Pal, & N. Memon, "Automated reassembly of the file fragmented images using greedy algorithms," IEEE Trans. Image Processing, vol. 15(2), pp. 385-393, 2003.

[7] M. Karresand, & N. Shahmehri, Reassembly of fragmented jpeg images containing restart markers. in 2008 European conference on computer network defense, 2008.

[8] The International Telegraph and Telephone Consultative Committee (CCITT). Information technology—digital compression and coding of continuous-tone still images–requirements and guideline (ITU-T T.81), 1992. Retrieved Sept. 5, 2012, from World Wide Web Consortium (W3C): http://www.w3.org/Graphics/JPEG/itu-t81.pdf

[9] K. M. Mohamad, & M. Mat Deris, Visualization of JPEG metadata. in: Proceeding of the 2009 first International Visual Informatics Conference on Visual Informatics,2009.

[10] P. Alvarez, "Using extended file information (exif) file headers in digital evidence analysis," International Journal of Digital Evidence. Vol. 2(3), 2004.

[11] H. Guo,.& M. Xu, A method for recovering jpeg files based on thumbnail. in: Automation and Systems Engineering (CASE) 2011 International Conference, vol. 1-4, 2011.

[12] Thumbnail. Retrieved Sept 5, 2012, from Wikipedia: http://en.wikipedia.org/wiki/Thumbnail.

[13] A. Merola, Data carving concepts, 2008. Retrieved Sept. 5, 2012, from SANS Institute: http://www.sans.org/reading_room/whitepapers/forensics/datacarving-concepts_32969Y.

[14] K. M. Mohamad, A. Patel, & M. Mat Deris, Carving JPEG images and thumbnails using image pattern matching. in 2011 IEEE Symposium on Computers and Informatics, 2011.

[15] K.M. Mohamad, A. Patel, T. Herawan, & M. Mat Deris, "myKarve: JPEG image and thumbnail carver," Journal of Digital Forensic Practice, vol. 3, 2011, pp. 74-97.

[16] K. Cohen, "Digital still camera forensics," Small Scale Digital Device Forensics, vol. 1(1), 2007, pp.1-8.

[17] N. A. Abdullah, R. Ibrahim, & K. M. Mohamad, Cluster size determination using JPEG files. in Proceedings of the 12th international conference on Computational Science and Its Applications, 2012.

[18] A. Pal, J. T. Sencar, & N. Memon, "Detecting File Fragmentation Point Using Sequential Hypothesis Testing," Digital Investigation,vol. 5, 2008, pp. S2-S13 2008.

[19] S. W. Ng, "Advances in Disk Technology: Performance Issues," Computer, vol. 31,1998, pp. 75-81.

[20] File Allocation Table, http://en.wikipedia.org/wiki/File_Allocation_Table#Boot_Sector

[21] R. P. Jemigan & S. D. Quinn, Two-Pass Defragmentation of Compressed Hard Disk Data with a Single Data Rewrite. U.S Patent 5574907

[22] Cluster Size for NTFS, FAT, and ExFAT, http://support.microsoft.com/kb/140365

[23] N. A. Abdullah, R. Ibrahim, & K. M. Mohamad, "Carving Thumbnail/s and Embedded JPEG files Using Image Pattern Matching," JSEA, Vol. 6, 2013, pp. 62-66.