A SURVEY ON MULTIMEDIA FILE CARVING

Raj Kumar Pahade[1], Bhupendra Singh[2] and Upasna Singh[3]

Department of Computer Science & Engineering, Defence Institute of Advanced Technology (DIAT), Pune, India

ABSTRACT

During forensic examination, analysis of unallocated space of seized storage media is essential to extract the previously deleted or overwritten files when the file system metadata is missing or corrupted. The process of recovering files from the unallocated space based on file type-specific information (header and footer) and/or file contents is known as Data Carving. The research in this domain has witnessed various technological enhancements in terms of tools and techniques over the past years. This paper surveys various data carving techniques, in particular multimedia files and classifies the research in the domain into three categories: classical carving techniques, smart carving techniques and modern carving techniques. Further, seven popular multimedia carving tools are empirically evaluated. We conclude with the need to develop the new techniques in the field for carving multimedia files due to the fact that the fragmentation and compression are very common issues for these files.

KEYWORDS

Digital Forensics, Data Recovery, Multimedia File Carving, Data Carving, File Carving Tools

1. INTRODUCTION

The increasing popularity of computers and internet has given rise to the growth of cybercrimes such as financial frauds, cyber warfare, child pornography, suspected terrorism etc. According to the 2015 Indian Risk Survey (IRS), Information and Cyber Insecurity threat is second largest threat to the nation at 9.47% [1]. According to the 2014 KPMG Cyber Crime Survey Report India, 89% of the survey respondents acknowledged that cybercrime is a major threat to nation [2]. The 2014 US State of Cybercrime Survey states that every three in four respondents detected at least one security incident over the past 12 months [3]. The results of the 2013 CERT Australian Cyber Crime and Security Survey indicate that 56% organizations out of the 135 organizations identified one or more cyber security incident over the past one year [4]. As the cybercrimes are being more frequent nowadays, the importance of Digital Forensics has also grown in investigations. Digital Forensics is the application of scientific principles to the investigation of artefacts present in one or more digital devices in order to reconstruct the sequence of events which led to a particular incident. The Technical Committee of Digital Forensic Research Workshop (DFRWS) defines digital forensic as the practice of identifying, preserving, extracting, analysing and presenting legally sound evidence from digital sources such as computer hard drives [5], [41].

The process of digital forensics starts with the identification of digital device(s) containing potential evidences. In order to maintain the evidential integrity and security, the secure collection of identified digital evidences is of prime importance. Disk imaging ensures the secure collection and preservation of evidences for extended period of time. The imaging enables us to create the exact replica of the disk drive containing allocated as well as unallocated space. Allocated space can be defined as the space allocated to the files having active entries in the file system. However,

an unallocated space is unreferenced region by the file system that may contain the data of the previously deleted files. If the file system information is missing or damaged, the entire disk layout may be considered as unallocated space [6]. During analysis, this unallocated space is examined and analysed to recover the data associated to the deleted files.

The deleted files from unallocated space can be recovered using traditional recovery techniques. Majority of these techniques exploits the file system information to locate and recover deleted files from disk drive. These techniques are relatively fast and accurate due to the fact that they utilize the file system metadata. One of the limitations of these approaches is that, if the file system metadata is overwritten or corrupted then the recovery becomes less effective [7]. It is evident that the traditional recovery approaches would fail where the file system metadata is unavailable, led to the development of new forensic approaches to recover deleted files. File carving is an advanced technique in which only the file structure (i.e. header and footer) and its contents are used for the recovery without exploiting the file system metadata information [8], [36], [42]. Often, file carving is used where the files are to be recovered from the unallocated space of the disk drive.

The majority of the standard file formats have their own defined unique headers and footers. The carving of continuously allocated file becomes relatively easy by using unique header and footer to identify the start and end of the file on disk layout. Most file systems fragment files when files are expanded, modified or deleted [6], [9]. When a file is not stored in the correct sequence on consecutive clusters on disk, the file is said to be fragmented leading to difficulties in file carving. Pal and Memon [6] have discussed various reasons on file fragmentation on disk drives.

This paper surveys the carving techniques of the multimedia files while considering the different types of fragmentation. It is assumed that the no file system information is available and only the multimedia file fragments are on hand for carving. The paper is organized as follows: Section 2 provides the brief background on file carving, and subsequently how fragmentation can occur. The related work in Section 3 brings out the advancements in the area of file carving and in particular multimedia file carving. Section 4 presents the experimental results of the multimedia file carving tools with discussion in Section 5. We conclude in Section 6, and future work on multimedia file carving is discussed in the following Section 7.

2. BACKGROUND

Signature-based file carving is the most common and straight forward approach for carving deleted files. The technique is based on the search for the unique sequence of bytes called header generally found at the start of the file, then it looks for the footer which is again a unique sequence of bytes at the end of the file. The area between the file header and footer constitutes a file content. This approach is well suited for files allocated contiguously on the storage media. It fails when the file is fragmented, since the file fragments are scrambled on storage media. In some file formats where the file size is not specifically mentioned, this method may lead to partial recovery of the file, due to the fact that this kind of recovery solely based on the identification of both header and footer [6], [10]. If that specific sequence of footer is found in the file content, then it will consider it as the end of file leading to partial recovery. In case, where the sequence of bytes identifying the header and footer of the defined file formats are found in file content will lead to false positive. In case of the portion of the file is overwritten, restoration of the file using the file signature can be almost impossible because validation of the restored file is failed due to the partially overwritten data [11].

Fragmentation: Most of the file systems such as FAT, EXT, NTFS, and HFS etc. are affected by the problem of fragmentation; they often break the files into discontinuous blocks [6]. Usually, a hard-disk is broken down into clusters of equal size, when a file being stored on disk is larger

than the cluster size; it occupies more than one cluster. When the allocated clusters to a file are not contiguous or not in correct sequence due to lack of the contiguous free clusters results into fragmentation. Apart from this, the extending or appending the stored file may also cause fragmentation if the contiguous space is not available to grow at the end of file [12]. Compression of stored files on the storage media also causes fragmentation. Furthermore, the file deletion also causes fragmentation because it partitions the unallocated space [13], [37]. During the reconstruction of a fragmented file using signature-based file carving, the order of clusters from the start of file (header) to the end of file (footer) produces the incorrect file [6].

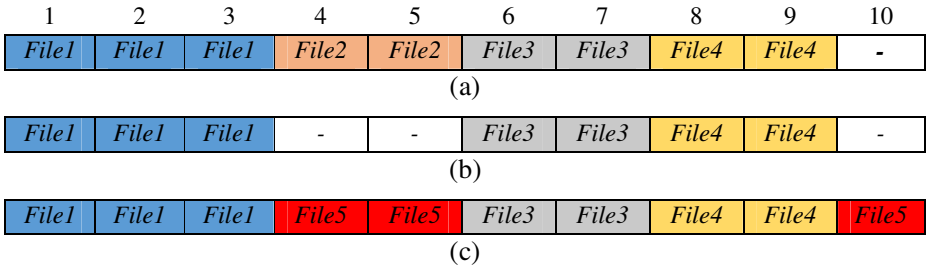


Figure1. File Fragmentation

Figure 1, illustrates file fragmentation on a disk with ten clusters. Figure 1(a), depicts four files: File1, File2, File3 and File4 which are stored in consecutive clusters specifies no fragmentation. It is to be noticed that cluster 10 is free. In Figure 1(b), the file File2 is deleted and the allocated clusters to File2 are marked as free. In Figure 1(c), new file File5 which requires three clusters is stored; file is saved in clusters 4, 5 and 10 leading to fragmentation. In case of normal deletion, most operating systems do not erase the actual contents of the file on the disk; instead delete the file table entry for a particular file [6], [9]. In such scenario, the actual content of files may still be available on disk; however, the file entries are no longer available. In Figure 1, if the file file5 is deleted and subsequently an examiner examines the disk space, he may be able to extract the clusters 4, 5 and 10; however, he may not be having the accurate information about the correct sequence of the recovered clusters.

3. RELATED WORK

Recovery techniques play key role during forensic examination of digital media. The evolution of recovery techniques has witnessed many enhancements over the years and recovery techniques are highlighted by Mohsen et al. [14]. Pal and Memon [6], [9] highlight the improvements needed in this field. We have divided the existing work in this field into three categories: *Traditional carving techniques*, *Smart carving techniques* and *Modern carving techniques*.

3.1. Traditional Carving Techniques

The traditional techniques for recovering the deleted files from the storage media usually exploit the file system metadata [15], [16], [19]. During the forensic investigation, in many cases, it has been found that the file system metadata is not available, led to the development of new recovery technique called file carving. File carving solely relies on the file structure information and file contents and does not rely on file system information. These techniques are frequently used to extract the data from unallocated space during an investigation. Recent file carvers not only use file type specific information but also use the individual file content to correlate the file fragments [6], [9], [10], [17].

Richard and Roussev [16] present “Scalpel” as one of the first and high performance file carver. Scalpel works on the Boyer-Moore search algorithm [18] to search file headers and footers in raw

disk images. It identifies the sequence of clusters beginning from cluster containing header to the cluster containing footer as the potential file. Due to the fragmentation, some of the recovered clusters may not belong to the potential file, the recovered clusters need to be analysed further for their consistency before merging into a file otherwise it may lead to the generation of false positives. In case of file having no associated footer (such as .txt file) may also generate false positives. In some cases, if a file header or footer is missing or corrupted due to the disk corruption, additional information such as file length may be required in order to carve a file. Moreover, if the file length is not available, the maximum size of the file needs to be considered for each file type while carving [10]. The carving process of Scalpel is divided into two phases. During the first phase, it locates unique header and footer in disk image that results in to the creation of the database containing metadata (file type, start location of file, file length etc.) for each file to be carved. As the actual file name is stored in file table which is presumed to be unavailable, so the tool assigns artificial file names to the files to be carved. The second phase is the actual content carving by using the metadata generated in first phase. These carved contents are stored separately by assigning the appropriate file extension. The more advanced version of Scalpel is FastScalpel which uses Aho-Corasick multi pattern search algorithm and is faster than the earlier version of Scalpel [18].

For carving fragmented files, many techniques introduced by Garfinkel in his submission to the DFRWS 2007 challenge [10], [20], [40]. The virtual file system implementations such as CarvFS and LibCarvPath provides the provision for validation of carved data inside the original image without making the copy to the another file also known as zero-storage carving greatly reduces the space requirement and execution time. In bi-fragment file carving [10], a set of clusters containing the headers and a set of clusters containing the footers are identified. Later these clusters are reassembled or rearranged to form a valid sequence of clusters called objects. These objects are later validated with the known file formats; this process is known as object validation [10]. Garfinkel presented the utilization of fast object validation for reassembling files that have been divided into two pieces. This procedure is alluded to as Bi-fragment Gap Carving (BGC).

3.2. Smart Carving Techniques

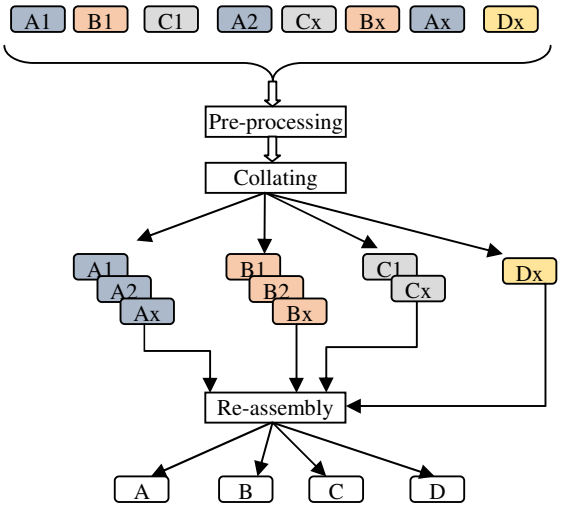
The key evolution steps towards the development of file carvers are outlined by Pal and Memon [6], [9], these steps are the main building blocks of the file carver which enables recovery of fragmented files. The architecture is flexible and robust and does not put any limitation on the number of file fragments to be carved since the large files may have many number of file fragments as pointed out by Garfinkel [10]. Once these file fragments are recovered, the fragments of the similar file formats are merged together and validated to ascertain the correct sequence of fragments. This validation process is repeated until the correct sequence is determined. These evolution steps are: *Pre-processing, Collating* and *Reassembly*.

3.2.1. Pre-processing

This is the first step towards the smart carving proposed by Pal and Memon [6], [9] as shown in Figure 2. This step is essentially applicable where the disk drive contents are encrypted or compressed. Many operating systems such as Windows, Android, iOS etc. have the feature of encrypting the contents before storing onto the disk. Before applying carving, it is important to have the contents of the disk decrypted otherwise carver will produce the false results. Similarly, disk compression is another feature provided by various operating systems to enable the efficient use of the disk space, files are compressed prior to storing onto the disk. Therefore, decompression of the contents is essential before applying the carving [13]. In pre-processing stage the above issues are addressed. In addition to the above issues to be considered, an important task carried out by pre-processing stage is determination and segregation of allocated

and unallocated clusters by using the file system metadata. If the file system metadata is not available or it is corrupted, then the entire disk is considered as unallocated space. The output of the pre-processing stage is the decrypted and/or decompressed disk contents and the clear demarcation of the allocated and unallocated space. Once the pre-processing stage is completed, carver enters into the collation stage.

Figure 2. File Carver Architecture proposed by Pal and Memon [9]



3.2.2. Collation

This phase examines the output of the pre-processing phase and classifies the clusters as per their file signature found. This phase essentially groups the similar fragments together, in order to reduce the number of fragments to be considered for recovery as shown in Figure 2. Moreover, the process of evaluation of fragments becomes faster and effective. A file type represents a specific file format like jpeg, docx or pdf etc., they all have their own format called signatures. In research, various approaches have been proposed for collating the file fragments, as mentioned below:

a. Keyword or pattern matching method is based upon the searching a specific byte sequence (at specific byte offset) value to determine the file signature in clusters. The popular example of this method is header/footer matching and it is widely used in many carving tools [16], [19], [38]. The headers are typically found at the beginning of the cluster, that indicates the start of the file or file fragment. As highlighted in Figure 3, the header (first 12 bytes) to be checked for mp4 format is shown. The four bytes starting at offset 04 indicates the file type (ftyp) and four bytes starting at offset 08 of the header represents the file sub-type (mmp4).

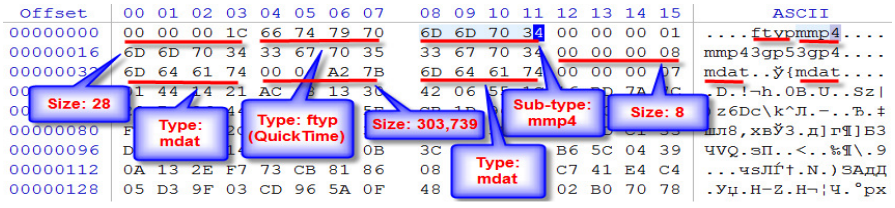


Figure 3: MP4 File Format

Though, the signature matching is relatively quick and easy to identify the file type, it may lead to false positives, due to the fact that the same pattern may also exist in blocks of different

file types. Hence, further validations need to be performed for correct identification of file type.

b. Pal and Memon [6], [9] propose another approach to identify the file type is based on the computation of the particular word(s) frequency in the blocks. By computing the frequency of certain words, it can be decided that the block may probably not belong to the formats such as video, audio and image. This is especially applicable for the text based files that have the text-based metadata. This approach may work for very specific file type such as html, and may not be effective for other type of files.

c. Entropy indicates the amount of average information contained in a file. A file with low entropy is well-ordered or well-structured file while a file with high entropy is less ordered or less structured meaning high randomness of information. Measuring of entropy may reveal the probable file type even though the file signature is removed or replaced [21], [39]. This entropy measurement can be useful in identifying the blocks. Paul et al. [21] perform an experiment to measure entropy of data blocks of various file types from disk images. In order to perform experiment, the TSK (The Sleuth Kit) [22], [23] tool is used to extract the data blocks and a utility called *ent* [24] is used to measure the entropy of those data blocks.

   Figure 4, illustrates the findings of experiment, in which is evident that the multimedia and document file formats reveal high mean entropy ranging from 7.270 to 7.981 bits/byte, while other file formats such as executables and older document formats revealed low mean entropy ranging from 3.822 to 5.989 bits/byte. Interestingly, the music file formats such as MP3 and M4A show very high mean entropy which is similar to the compressed image files. It is to be noted that the variance and deviation in entropy is less for music file formats.

   In file carving, entropy can be used for the classification of the cluster by comparing its entropy measurement with its adjacent blocks. Classifying clusters based on entropy measurement is better than the pattern matching technique in terms of execution time; however it may also lead to false positives. Moreover, slack space may hold the previously occupied data in a cluster and is included in the entropy calculation, hence may affect the classification of clusters [37]. In cases where disk is wiped out or low level formatted, the slack space is zeroed, that indicates very low level of entropy of slack space. While computing entropy of the cluster having zeroed slack space will have the entropy of the actual content on the cluster. However, certain file types may produce very similar measurement of entropy since entropy measurement only considers file contents. This method may be more effective when it is used along with other methods like signature matching.

d. McDanial and Heydari [25] propose content-based file type identification algorithm that automatically computes the “fingerprint” for a given file type based on a set of known input file. These fingerprints are computed using file contents rather than files metadata. These fingerprints are later used to recognize the true file type. Three approaches are presented to identify the true file type. These approaches are: analysis of byte frequency, cross-correlation analysis of byte frequency, and analysis of file header/trailer. In experiments performed by McDanial and Heydari, the accuracy of recognizing a true file type varies ranging from 23% to 96% depending on the method selected. It provides highest accuracy (up to 96%) when all three approaches are selected to recognize the true file type. The proposed algorithm can be used in many applications were file types need to be identified for performing operation like file carving. Different approaches proposed to compute fingerprints are as follows:

i. Byte frequency distribution (BFD) [20], [25] creates byte histogram for the file. In order to compute the BFD fingerprints, multiple files of same file type are taken, and then the BFD of each file is computed. The final BFD is computed by averaging the BFDs of the each file.
ii. Calculating correlation strength [25] for byte values. In file content, certain byte values will have consistent frequency as compared to other byte values in file content. This average of the variation of frequency values can be utilized to correlate with the file type. While identifying the file type of unknown file, the frequency of certain byte value is looked upon, if it matches with the threshold, the file type is assigned to the unknown file.
iii. Cross-correlation, certain characters (e.g. '<' & '>' in html) would occur roughly at the same frequency. In file contents, there might be chances that, certain characters come at the same number of times (also known as the cross-correlation of characters), it is largely depends on the specific file format. Therefore, this information can be utilized to compute the fingerprint for a specific file type. This computed fingerprints later can be compared with the unknown files cross-correlation fingerprint to find out the file type [6].

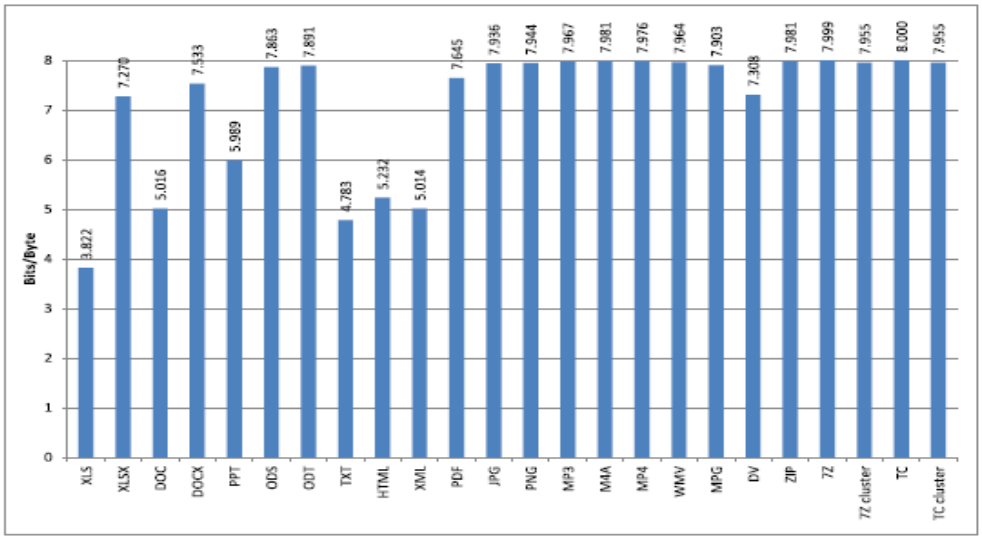


Figure 4. File Type Mean Entropy [23]

The fingerprint computation is important task and depends upon the input dataset of known file types. If the input dataset is small, then the fingerprint generated will be weak and may lead to the generation of wrong file type, similarly, if the input dataset is huge and complex, the generated fingerprint will be too strong for the exact match might generate too many false negatives. Moreover, the approaches perform poorly when they are used independently, therefore, these approaches should be used together to get the better classification results.

3.2.3. Reassembly

Once the file fragments of similar type are collated, they need to be assembled together so that they produce the valid file. The task of reordering and merging of file fragments is carried out in reassembly phase. In this phase, the point of fragmentation is determined for the particular fragment then the starting point of fragmentation needs to be determined from remaining fragments. This process is carried out until entire file is recovered or found to be unrecoverable. The process starts with finding the fragment which contains the header, called base fragment

followed by the identification of endpoint of the fragment. Suppose the clusters C_x, C_{x+1}, C_{x+2} and C_{x+3} belong to fragment F, then Cx+3 will be the fragmentation point for the fragment F.

The work done by Pal et al. [9], [6], [26] is based on parallel unique path (PUP) algorithm to reassemble the fragment. PUP is a modification of Dijkstra's single source shortest path algorithm. It begins with the header and proceeds to pick the best matched cluster from the recovered clusters for each header. The selection of the matched cluster depends upon the weighted edge computation proposed by Kulesh et al. [27] in their work. They computed weight of the edge by using the technique known as Prediction by Partial Matching (PPM). As pointed out by Kulesh et al. [27], PPM works well for structured dataset such as text files, but its efficiency degrades for images and other compressed datasets.

The work done by Kulesh et al. is further extended by Pal et al. [26], [9] with introducing a method for fragmented file recovery. The weight of an edge between two clusters is determined by comparing the cluster boundaries of cluster with the boundaries of subsequent clusters and leads to the analysis of the pixel differences between the two clusters. The weight of the edge is determined as the sum of difference of pixels throughout the boundary. This weight signifies the similarity/dissimilarity between the clusters. Problem of computing the weight of the edge is further refined into the K-vertex disjoint path problem by the Pal et al. [26]. They states that reassembly of fragments is nothing but the finding the k-vertex disjoint path, where k is the number of headers found in the different clusters, each base-fragment is represented by a vertex. Once the vertices are identified, each unallocated cluster is weighted according to the probability that one cluster is followed by another cluster, leading to the generation of disjoint path.

In order to find the best recovery of a file, average path cost is considered because the best recovery is said to have the lowest average path cost. The computation of the average path cost is the sum of the weights of edges and dividing it by the number of clusters. The lowest average path cost is considered as the best recovery and the clusters associated are removed from the pool of recovered clusters. The recovery process with the remaining clusters continues until all the files are recovered. This algorithm is called the Shortest Path Algorithm (SPF). As far as accuracy of the PUP and SPF is concerned, SPF could produce the results with 88% accuracy and PUP could recover 83% of files. However, the execution time of PUP is better than SPF and also the scalability of PUP is much better as compared with SPF.

PUP algorithm is further modified by Pal et al. [26], [9] to exploit the Sequential Hypothesis Testing (SHT) method. SHT method says that every cluster which is added to the path, it is very likely that the subsequent clusters may also belong to the same path, so while adding a cluster to the path, one should check the immediate clusters first rather than checking the clusters at the other location. The computation of weight of the edges is the key challenge for the graph based methods for fragment reassembly mentioned in the research papers. The accurate model for computing weights needs to be developed for the graph based method; furthermore, these methods are useful for recovering certain file types such as text and images. New weighting models needs to be developed for multimedia files and other types of files.

3.3. Modern Carving Techniques

Modern file carving techniques primarily focus on carving of multimedia files. The recovery of deleted and overwritten multimedia files is an important task during evidence analysis phase. One of the most commonly used approach to recover deleted multimedia files is signature-based carving as shown in Figure 5. It starts with searching the unique start marker (header) and ends with the search of the end marker (footer), these blocks and blocks between these two markers are linked together to form the file region to be carved. In Figure 5, the AVI file start markers are shown; it starts with the RIFF signature identification at offset 0 and AVI signature at offset 8.

This approach is well suited for the files which are stored in continuous blocks; it does not perform well in case of fragmentation.

Figure 5. AVI File [13]

Various algorithms and prototype tools have been developed for recovery of deleted video files as response to the DFRWS 2007 challenge. The recent work on video file carving focuses on the recovery of individual video frames as proposed by Na et al. [11]. Figure 6, shows the video frames of a video file, these frames are played at appropriate rate in the video file, therefore, video frame is the fundamental entity of a video file, therefore, work is being carried out towards the recovery of video frames. Once these video frames are recovered, the indexing of the video frames can be recreated to ensure the proper ordering of the video frames in a video file.

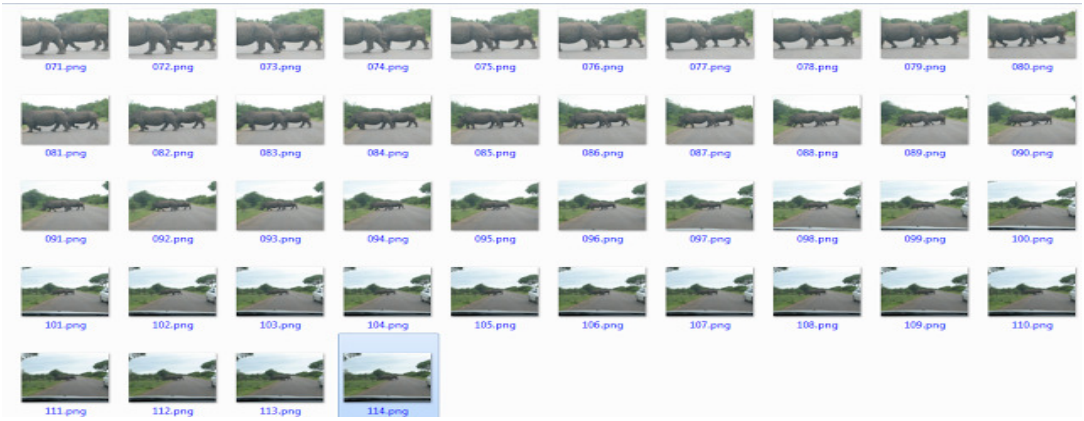


Figure 6. Video Frames in sequence

Handling of huge number of video frames is an issue in frame-based recovery method [17], [11] since each fragment may contain many frames. Extraction of individual frame from a particular cluster is a time taking task since each frame is to be examined for its identification. Instead, the cluster boundary can be utilized to identify the frame. Generally, all the data in a cluster belongs to a single file, except the slack space (if present) in last cluster which may contain the unknown data. So, the video frames recovered from a cluster can be considered as a part of the same file. In multimedia file carving, it is difficult to say that which portion of the dataset would produce the playable file, hence, leads to the generation of false positives. Therefore, it is important to carve an entire dataset in order to produce an optimal result. Furthermore, different imaging tool characterizes unallocated space differently further complicates the process. Apparently, each multimedia files have its own data formats, which can be searched and analyzed. In frame-based recovery, the search criteria play an important role to identify a valid frame of a multimedia file. The search criteria should be selected judiciously, because if the criterion is too stringent, then it

will produce too many false negatives. Similarly, if the criterion is too vague then it may produce too many false positives.

For an example, AVI file structure (refer Figure 7) is well-defined and widely used multimedia file format. AVI file format is basically based on the Resource Interchange File Format (RIFF) and it provides adequate information to identify the file format, data format and the actual data streams. While carving avi file, carver should start identification of the header as shown in Figure 7, once header is identified, it should check for the information about the data formats at the specific offset (hdrl). After identification of the header and the data format, the actual data streams are located under the movi section as chunk information is provided in idx section of the structure.

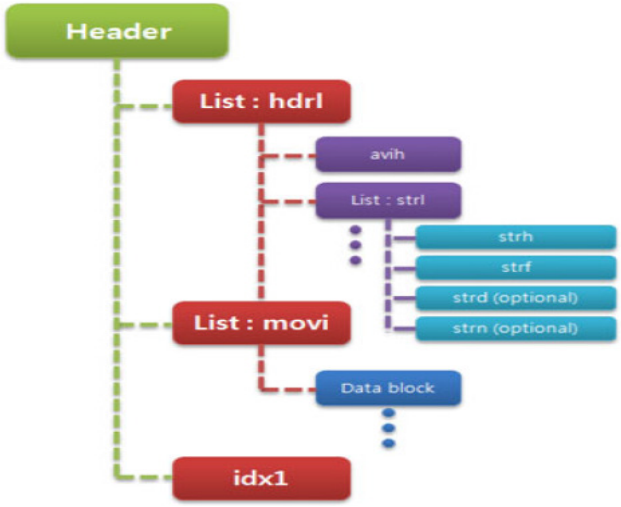


Figure 7. AVI File Structure

Similarly, the file formats such as MPEG-2 and MP4 are also well defined that it helps in identification and reassembly of their fragments. These fragments have sufficient amount of information to aid in matching of related file fragments. Moreover, an individual video frame possesses adequate information that can assist while identifying and matching frames. For example, the Codecs such as H.264, while carrying out the encoding of video file, it embeds additional information into individual frames which can be utilized later for recovery purposes.

As far as repairing of corrupted or partially overwritten files are concerned, there are approaches such as reconstruction of file container around a file fragment or in cases where the base fragment is damaged or partially overwritten in such a way that original header cannot be recovered, grafting of appropriate reference header can be carried out for the recovered multimedia streams such as audio and video. Though, it is not an easy task, database of container formats and reference headers needs to be maintained while grafting container or reference headers. Selection of appropriate container or reference header is another issue during grafting, in order to solve this problem, sometime it is advantageous to take the reference headers from the files previously recovered from the same data source.

4. EXPERIMENTS AND RESULTS

Digital forensics as we know, is a step-by-step process, to perform activities specific to a step, there are variety of tools available which can aid in performing a particular activity. For example, imaging is a very important activity in forensics, there are many tools available who can perform

this activity, as an examiner, one should know the strengths and weaknesses of these tools so that they select the best tool available. The examiner must be familiar with different forensic tools available and the selection of these tools must be done judiciously, if not, it will jeopardize the whole process of investigation. Knowing the strengths and weaknesses of each and every tool is a laborious task and it is not feasible for an examiner to know and apply all the available tools to an investigation. So, in order to create an effective tool chain, the performance analysis of these tools required to be carried out so that the comparison of tools becomes easy.

In this paper, we have focused on the carving of fragmented multimedia files; idea was to analyse the performance of popular carving tools for carving fragmented multimedia files. In order to do that, various carving tools such as Foremost [19], Autopsy [28], Defraser [30], Photorec [29], OSForensic [31], DFF [32] and Encase [33] were checked for their effectiveness. Creation of the test image is the first task in this experiment. Various multimedia files of different sizes such as mp4, mov, avi, 3gp and mpg were taken as input files. The overall process of creation of input raw image is depicted in Figure 8.

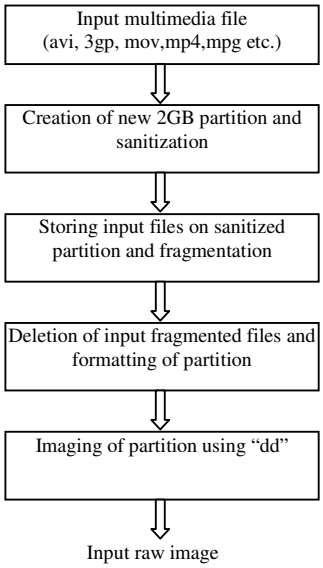


Figure 8. Raw Image Creation Process

As shown in Figure 8, the process of creation of the input raw image starts with the collection of different multimedia files, here we have taken five different file formats (mp4, mov, avi, 3gp and mpg) as input files. In next step, we created a 2GB NTFS partition and sanitized it using an open source utility called DiskWipe [35] with erase pattern as 'Russian GOST P50739-95 (2 passes-quick)'. Once the sanitization is done, the input files were copied on to the newly created partition. In third step, all the input files were fragmented using a utility called PassMark Fragger [34], the process is explained in detail in subsequent section. After fragmenting input files, the input files were deleted and the partition has been formatted again (no disk wipe) to consider the entire partition as unallocated space for carving. After formatting of the partition, the raw image of the partition has been created using the "dd" utility. The output of the "dd" is the input raw image, which will be used as an input for the tool under test.

4.1. Fragmenting the Input Files

Since fragmentation was the criteria to create the input test image, the input multimedia files were fragmented using utility "PassMark Fragger" [34]. It allows fragmenting or defragmenting

individual files, the number and size of fragments are also can be given as input to this utility for input file. It provides the provision of applying the different types of fragmentation on to the file such as scattered, random, concatenated etc. The following types of fragmented files were generated using "PassMark Fragger" utility:

a) Contiguous files (No fragmentation)
b) Concatenated files: Linear fragmentation, fragments are concatenated to each other.
c) First fit fragmentation files: The file chunks are stored as per the first fit method, in which the first fit cluster is allocated for the file fragments.
d) Scattered files: The file fragments are scattered in memory layout
e) Random fragment files: Nonlinear fragmentation in which the file fragments are stored randomly in memory layout.

Table 1, represents overall combination of the fragmented input files, each file format has five files, different type of fragmentation is applied to each one of it. For example, five avi files are as follows: one continuous avi file, one concatenated fragmented avi file, one first fit fragmented file, one scattered fragmented file and one randomly fragmented file. Similarly, other file formats are also having the combination of files with different types of fragmentation.

Table 1: Input Files Formats and Fragmentation Types

		Fragmentation Type					
		Contiguous	Concatenated	First fit	Scattered	Random	Total
File Type	Avi	1	1	1	1	1	5
	Mov	1	1	1	1	1	5
	3gp	1	1	1	1	1	5
	Mp4	1	1	1	1	1	5
	Mpg	1	1	1	1	1	5

4.2. Performance Metrics

The input raw image was subjected to individual tool to produce its output. An interpretation of the output or result produced by individual tool provides an insight to the tools performance towards the carving of files. The tools performance is measured in terms of total files carved, total number of false positives generated by the tool for a file format and whether the recovered file is viewable, not viewable or partially viewable. The result produced by a tool is represented in tabular form. Interpretation of result is elaborated in subsequent sections.

In result table, total files carved row presents the total number of files recovered for each file format. Total files carved presents the tools capability to carve the different files, this number may vary depending upon the technique employed by the tool and because of false positives. As mentioned earlier, false positives are generated by the tool due to the fact that the tool has carved a file based on the false signature string (not actual signature) present within another file.

In result table, the viewable or playable file is represented by the tick mark (✓). Tick mark depicts the carved files that appear to be fully recovered or unchanged from the original input file. This entry in table shows that the carved file is an exact match to the original file or the modifications are not noticeable.

In result table, the partially viewable or playable files are represented by hash mark (#). Partial recovery indicates that only some portion of the file is recovered or the fragments are not

assembled in proper order to produce an exact file or the modifications to the recovered file is so evident.

In result table, the unrecovered input files are represented as cross mark (✗). In cases where the file is not open-able or recognizable or has no content (empty file) are also considered as unrecoverable file. In result table, false positive row presents the total number of files that were identified mistakenly. Under each column, the number represents the total false positives generated for a file format across different fragmentation types. Since all the input files were multimedia files, VLC media player was used to check recovered files.

4.3. Individual Tool Performance

4.3.1. Foremost

Foremost is an open source forensics tool and comes with many 'Linux' distributions, used for carving files from raw images generated by "dd", "encase" etc. Foremost is a command line utility similar to Scalpel, but it is slightly easier to use as compared to scalpel because foremost has built-in capability to search many common file formats such as jpg, exe, pdf, doc, etc. For file formats that are not built-in, it provides provision to define its signature and other information through configuration file. The following table shows the test result for the foremost tool.

Table 2. Files Carved by Foremost

		File Types				
		AVI	MOV	3GP	MPG	MP4
	Total files carved (31)	9	3	9	1	9
	False positives (18)	3	2	4	-	9
Fragmentation Type	No Fragmentation	✓	✗	✓	✗	✗
	Concatenated	✓	✗	✓	✓	✗
	First Fit	✓ #	✗	✓	✗	✗
	Scattered	✓	#	✓	✗	✗
	Random	✓	✗	✓	✗	✗

As shown in Table 2, foremost has recovered all AVI and 3GP file, though it has recovered 3GP files under the MP4 extension. It did not recover any of the MP4 files and MOV files except one partial recovery. Only one MPG file is recovered by foremost tool. It is also observed that foremost has recovered around 1400 jpg files, though there were no jpg files were present in the input image. To summarize the tool performance, it can be stated that foremost is good at recovering AVI file as far as multimedia file carving is concerned.

4.3.2. Autopsy

Autopsy is an open source digital forensic tool; it is an enhanced version of "The sleuth kit (TSK)" [23] as it provides the graphical interface to TSK and other digital forensic tool. It is widely used by many agencies such as law enforcement, military and others for the digital forensic investigation. The advantage with Autopsy is that, it presents the result to the user as soon as it is available, this approach is advantageous where media size is enormous. In Autopsy, many ingest module executes parallel to speed up the overall investigation process. Table 3 represents the output of the Autopsy tool.

Table 3: Files Carved by Autopsy

		File Types				
		AVI	MOV	3GP	MPG	MP4
	Total files carved (53)	9	10	10	15	9
	False positives (30)	5	5	5	10	5
Fragmentation Type	No Fragmentation	✓	✓	✓	✓	✓
	Concatenated	✗	✓	✓	✓	✓
	First Fit	✓	✓	✗	✓	✓
	Scattered	✓	✓	✗	✓	✗
	Random	✓	✓	✓	✓	✓

As depicted in Table 3, Autopsy has produced very good result for the multimedia files. It has recovered 21 files out of 25 in total. It has recovered files across the file formats, though few files it could not recover, as mentioned in Table 3. The main concern here is the generation of the false positives which are quite large.

4.3.3. PhotoRec (TestDisk)

PhotoRec is an open source digital forensic tool, designed to recover lost or deleted files such as document, audio, video, picture etc. from the disk. PhotoRec is a companion program to the TestDisk tool package which is primarily used to recover the lost partition. PhotoRec recovers deleted files using signature matching of the file; it does not consider the storage media file system metadata. PhotoRec performs the block level reading of media, further, each block is checked against the signature database. PhotoRec is a console based program and it can handle more than 440 file formats. The output of the PhotoRec is presented in Table 4.

From Table 4, it is clear that, PhotoRec is able to recover most of the multimedia files. It has recovered 19 files fully, 8 files partially and 3 files it could not recover. It has generated 19 false positives; most of the false positives were generated while carving MPG files. It is clear from the table above that the generation of false positives and partial recoveries are the main issue with the PhotoRec forensic tool.

Table 4. Files Carved by PhotoRec

		File Types				
		AVI	MOV	3GP	MPG	MP4
	Total files carved (56)	9	9	9	20	9
	False positives (28)	3	4	4	14	3
Fragmentation Type	No Fragmentation	✓	✓	✓	✗	✓
	Concatenated	✓	✓	✓	✓	✓ #
	First Fit	✓ #	✓	✓	✗	✓
	Scattered	✓	✓ # (2)	✓	# (2)	✓
	Random	✓	✗	✓	# (2)	#

4.3.4. Defraser

Defraser was developed to find partly erased or damaged multimedia files and, if necessary, repair them. What sets Defraser apart is its ability to find not just complete multimedia files, but also partial files, such as deleted video files that have been partly overwritten. Other forensic file recovery software doesn't typically detect deleted files if the initial part of the file data is overwritten – nor does it allow playback of any damaged video files that are found. Defraser, however, incorporates extensive video file format knowledge, enabling it to recognize incomplete files using any of its supported video file formats. It also offers specialized tools allowing playback of recovered video frames. The Free Edition of Defraser is available online and supports MPEG-1, MPEG-2, MPEG-4, AVI, ASF and 3GP video formats. However, the encoding method of choice of the latest digital video cameras is H.264 and this is not supported by the Free Edition Defraser software. The Table 5 shows the test result for the Defraser tool (free edition).

Table 5. Files Carved by Defraser

		File Types				
		AVI	MOV	3GP	MPG	MP4
	Total files carved (79)	25	2	23	22	7
	False positives (49)	16	-	19	14	-
Fragmentation Type	No Fragmentation	✓	✓	✓	✗	✓
	Concatenated	✓ #	✗	✗	✓	✓ #
	First Fit	✓ #	✓	✓	✗	✓
	Scattered	✓ #	✗	✓	# (2)	✓
	Random	✓ #	✗	✓	✓ # (2)	#

The performance of Defraser is very good for avi, 3gp and mp4 files, it could recover all the files despite the fragmentation except one concatenated type file of 3gp format. The total recovery was 79 file in which 49 were false positives. The false positives generated by Defraser are quite large as evident in Table 5; moreover, it has also produced some partially recovered files. The overall performance of the tool is good and it is user friendly. The main concern for Defraser is the huge number of carved files and the false positives because if the carved files are too many then the analysis takes considerable amount of time.

4.3.5. PassMark OSForensics

PassMark OSForensics is another digital forensic tool and it is not open source tool, though the evolution version is available on tools site. We have used the evolution version of the tool for our experiment. It provides the complete suite of tools right from imaging to analysis of source data on storage media and live systems. As far as file recovery is concerned, tool has dedicated module for the recovering deleted files. So, in order to check the performance of PassMark OSForensics tool, we have taken the same test raw image and file recovery module is executed. Table 6 presents the overall result of PassMark OSForensics tool. As shown in Table 6, performance of OSForensics is not good for 3GP and MPG files. It has recovered the AVI files fully but it could not recover all the MOV and MP4 files. It is to be noticed that it has carved 24 files out of which 6 were false positives.

Table 6. Files Carved by OSForensics

		File Types				
		AVI	MOV	3GP	MPG	MP4
	Total files carved (24)	11	9			4
	False positives (6)	3	3			-
Fragmentation Type	No Fragmentation	✓	✓	✗	✗	✗
	Concatenated	✓ #	✓	✗	✗	✓ #
	First Fit	✓ #	✓	✗	✗	✗
	Scattered	✓	# (2)	✗	✗	✗
	Random	✓ #	#	✗	✗	✓ #

4.3.6. Digital Forensic Framework (DFF)

DFF is an open source tool that provides forensic platform to perform digital investigation and incident response. DFF is an automated tool that guides user to prepare the query for specific search. It is able to perform the in-depth search on disk-drive and volatile memory. Its ability to create the quick and specific query, results into the quick detection of documents, multimedia file and other artefacts. The sample test raw image was analysed with the DFF in order to carve the multimedia files; it produced the following result, as shown in Figure below:

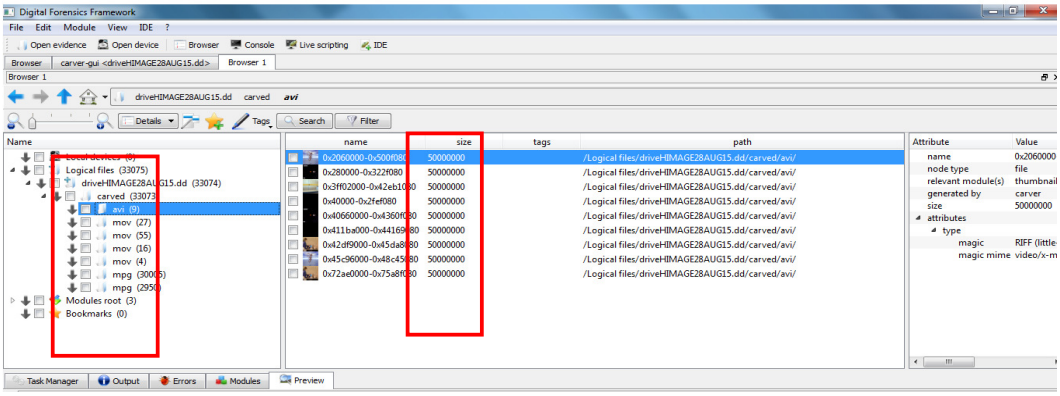


Figure 9. DFF Output

As shown in the Figure 9 above, it can be seen that, it recovered raw clusters where the input files were residing and listed as recovered files. No playable video has been recovered using DFF tool and huge numbers of false positives were recorded by the tool. Thorough analysis of DFF needs to be performed in order to bring out its capability.

4.3.7. Encase

Encase is very widely used digital forensic tool to conduct investigation from beginning to an end. Advantage with Encase is that it has the capability to manage the large number of evidences during investigation. Encase is equipped with the file carver module which does the carving using signature based methods and it can also examine the unallocated space. In order to perform the experiment on encase, same input raw image has been subjected to the carver module of encase. It has performed the raw reading of the clusters and checked against the file signature; clusters that have been found to contain header information are extracted and analysed further. We have observed that the carved clusters were not parsed further in order to do the merging of similar

clusters; hence the playable video could not be produced. This experiment is tried on the training version of encase, further analysis is to be carried out.

5. DISCUSSION

As far as digital forensics tool performance is concerned, it can be said that, two factors greatly influence the overall performance of a tool. First the total number of files carved including false positives and second, the processing time. For example, foremost has carved 39 files in that 17 were false positives, Autopsy carved 53 files in that 30 were false positives, Defraser carved 79 files in that 49 were false positives. The generation of false positives needs to be brought down in order to increase the overall performance of the carving tool; moreover, many tools generated huge number of empty files. The analysis becomes challenging due to huge number of empty files including the false positives. Carving time usually depends on the input image size, considering the today's storage media size, producing result quickly is challenging. New techniques should be employed to speed up the carving process. In experiment, we found that some tools were reasonably fast whereas some tools were dead slow in generating output i.e. encase took hours to generate the output. One should consider these factors while designing a carving tool.

Table 7 shows the summery of test results of the tools undergone the test, table shows the total number of files were carved and false positives generated by the tool. It also presents the total playable files recovered, number in bracket represent the efficiency of the tool to recover the playable files.

Table 7: Summery of Test Results

S No	Tool Name	Total number of file carved	False positives	Playable files recovered	Remarks
1	Foremost	31	18	11 (44%)	2 partial recovery
2	Autopsy	53	30	21 (84%)	Only 4 files were not recovered
3	PhotoRec (TestDisk)	56	28	19 (76%)	9 partial recovery
4	Defraser	79	49	17 (68%)	10 partial recovery
5	PassMark OSForensics	24	6	10 (40%)	7 partial recovery
6	DFF	-	-	-	Cluster based recovery
7	Encase	-	-	-	Cluster based recovery

6. CONCLUSION

We have presented the survey on file carving and multimedia file carving in particular, we discussed how various techniques and approaches can be used to recover files from unallocated space without using metadata. The fragmentation and it causes have been discussed, and why fragmentation is an important issue for multimedia files are also discussed in this paper. It is also discussed that the classical techniques are not very effective for the cases where files are heavily fragmented or partially overwritten. Smart carving is the way forward for the fragmented file carving. It is a step-by-step process, that starts with pre-processing to evaluate the structure of the existing file system to segregate the allocated and unallocated spaces on the disk to be carved, as only the unallocated space is subjected to carving. At the end of the pre-processing step, we get the clear layout of the unallocated space to be carved. Collation step takes this entire unallocated space as an input and further examines at the cluster level (fragments) and groups together as per the file-type identified, as the similar clusters are put in a group identifying the file type, at the end of collation, all the fragment whom file type is identified will belong to the concerned group,

and the fragments whose type is not yet identified are put in separate group. Basically, grouping the similar fragments is the sole purpose of collation. In reassembly, the individual group is further examined to merge the file fragments together so that the complete file can be produced. Once reassembly is over, the recovered files are checked for their relevance.

Some of the popular file carving tools have been examined in the paper to see their effectiveness in recovering fragmented multimedia files of different types. The performance results indicate that the no single tool to recover fragmented files is most effective. The performance of a tool is largely affected by the factors such as number of files carved, false positives and false negatives. The analysis of results is an important issue due to the fact that some of the tools have produced huge number of files since manual assessment is not only difficult but time consuming too. Therefore, new approaches need to be evolved for fragment identification and reassembly, so that the false positives can be reduced. However, it should not be too stringent to produce too many false negatives leading to less number of playable videos. Compression is another issue needs to be considered while devising a file carving method. Various file systems provide the capability to compress the files while storing onto disk (i.e. bit locker in Windows). If the files are compressed before storing onto the disk, it may cause problems while assessing unallocated space for recovering deleted files. During experimentation, it has been observed that when the files are compressed on disk, data carving results were not comparable with the results when the files were uncompressed on the disk.

7. FUTURE WORK

As far as multimedia file carving is concerned the carving efficiency can be further enhanced by incorporating the encoding standards such as MPEG-4 and H.264. The idea is to carve multimedia files on the basis of individual video frames by exploiting the information embedded by the encoding standards. The information obtained from these codec containers can be used to build the index for the video files so as to perform the reassembly of video fragments. The relevance of present approaches for high quality multimedia files of large size and disk capacity of terabytes needs to be examined.

REFERENCES

[1] FICCI, Pinkerton C&I India Ltd.: FICCI Indian Risk Survey 2015. www.ficci.com/Sedocument/20328/India-Risk-Survey-2015.pdf

[2] KPMG Cyber Crime Survey Report India 2014, https://www.kpmg.com/IN/en/IssuesAndInsights/ArticlesPublications/Documents/KPMG_Cyber_Crime_survey_report_2014.pdf

[3] US State of Cybercrime Survey 2014, http://www.pwc.com/us/en/increasing-it-effectiveness/publications/assets/2014-us-state-of-cybercrime.pdf

[4] CERT Cyber Crime & Security Survey Report 2013, https://www.cert.gov.au/system/files/614/679/2013%20CERT%20Australia%20Cyber%20Crime%20%2526%20Security%20Survey%20Report.pdf

[5] Palmer, G. (2001) “A road map for digital forensic research”, DFRWS Technical Report, DTR-T001-01 FINAL, DFRWS Technical Committee.

[6] Pal, Anandabrata, & Nasir Memon (2009) "The evolution of file carving." *Signal Processing Magazine, IEEE 26*, no. 2, pp 59-71.

[7] Poisel, Rainer, Simon Tjoa, and Paul Tavolato (2011): "Advanced file carving approaches for multimedia files." *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA) 2*, no. 4, pp 42-58.

[8] Raghavan, Sriram, (2013) "Digital forensic research: current state of the art." *CSI Transactions on ICT 1*, no. 1, pp 91-114.

[9] Memon, Nasir, & Anandabrata Pal, (2006) "Automated reassembly of file fragmented images using greedy algorithms." *Image Processing, IEEE Transactions on 15*, no. 2, pp 385-393.
[10] Garfinkel & Simson L., (2007) "Carving contiguous and fragmented files with fast object validation." *Digital Investigation,* Vol. 4, pp 2-12.
[11] Na, Gi-Hyun, Kyu-Sun Shim, Ki-Woong Moon, Seong G. Kong, Eun-Soo Kim, & Joong Lee, (2014) "Frame-based recovery of corrupted video files using video codec specifications." *Image Processing, IEEE Transactions on 23*, no. 2, pp 517-526.
[12] Poisel, Rainer & Simon Tjoa, (2011) "Roadmap to approaches for carving of fragmented multimedia files, IEEE," In *Availability, Reliability and Security (ARES), 2011 Sixth International Conference*, pp 752-757.
[13] Yoo, Byeongyeong, Jungheum Park, Sungsu Lim, Jewan Bang & Sangjin Lee, (2012) "A study on multimedia file carving method." Multimedia Tools and Applications, Vol no. 1, pp 243-261.
[14] Damshenas, Mohsen, Ali Dehghantanha & Ramlan Mahmoud, (2014) "A survey on digital forensics trends." *International Journal of Cyber-Security and Digital Forensics* 3, Vol. no. 4, pp 209-235.
[15] Aronson, Leon & Jeroen Van Den Bos, (2011) "Towards an engineering approach to file carver construction.", IEEE, In *Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*, pp 368-373.
[16] Richard III, Golden G. & Vassil Roussev, (2005) "Scalpel: A Frugal, High Performance File Carver." In *DFRWS*.
[17] Casey, Eoghan & Rikkert Zoun, (2014) "Design tradeoffs for developing fragmented video carving tools", *Digital Investigation,* pp 30-39.
[18] Zha, Xinyan & Sartaj Sahni, (2011) "Fast in-Place File Carving for Digital Forensics." In *Forensics in Telecommunications, Information, and Multimedia*, Springer Berlin Heidelberg, pp 141-158
[19] Foremost, http://foremost.sourceforge.net/
[20] Roussev, Vassil & Simson L. Garfinkel (2009) "File fragment classification-the case for specialized approaches." In *Systematic Approaches to Digital Forensic Engineering, SADFE'09. Fourth International IEEE Workshop*, pp 3-14.
[21] Weston, Paul & Stephen D. Wolthusen, (2013) "Forensic entropy analysis of microsoft windows storage volumes.", IEEE, In *Information Security for South Africa,* pp 1-7.
[22] Carrier, Brian, (2005) *File system forensic analysis*, Vol. 3. Reading: Addison-Wesley.
[23] The sleuth kit, http://sleuthkit.org
[24] A Pseudorandom Number Sequence Test Program (ent), http://www.fourmilab.ch
[25] McDaniel, Mason & M. Hossain Heydari, (2003) "Content based file type detection algorithms.", IEEE, In *System Sciences, Proceedings of the 36th Annual Hawaii International Conference*, pp 10
[26] Pal, Anandabrata, Husrev T. Sencar & Nasir Memon, (2008) "Detecting file fragmentation point using sequential hypothesis testing." *digital investigation* 5, S2-S13.
[27] Pal, Anandabrata, Kulesh Shanmugasundaram, & Nasir Memon, (2003) "Automated reassembly of fragmented images", IEEE, In *icme*, pp 625-628.
[28] Autopsy tool, http://sleuthkit.org/autopsy
[29] PhotoRec tool,. http://www.cgsecurity.org
[30] Defraser, http://sourceforge.net/projects/defraser
[31] OSForensics tool, http://www.osforensics.com
[32] Digital Forensic Framework (DFF) tool, http://www.arxsys.fr
[33] Encase tool, http://www.guidancesoftware.com
[34] File Fragmentation tool, PassMark Fragger, http://www.passmark.com/products/fragger.htm
[35] DiskWipe tool, http://www.diskwipe.org/
[36] Garfinkel, Simson L., (2010) "Digital forensics research: The next 10 years." *digital investigation* 7, S64-S73.
[37] Dezfoli, Farhood Norouzizadeh, Ali Dehghantanha, Ramlan Mahmoud, Nor Fazlida Binti Mohd Sani & Farid Daryabar, (2013) "Digital Forensic Trends and Future." *International Journal of Cyber-Security and Digital Forensics (IJCSDF)* 2, no. 2, pp 48-76.
[38] Hand, Scott, Zhiqiang Lin, Guofei Gu & Bhavani Thuraisingham, (2012) "Bin-Carver: Automatic recovery of binary executable files." *Digital Investigation* 9, S108-S117.
[39] Calhoun, William C. & Drue Coles, (2008) "Predicting the types of file fragments." *Digital Investigation* 5, S14-S20.
[40] Cohen, Michael I., (2007) "Advanced carving techniques." *Digital Investigation* 4, no. 3, pp 119-128.
[41] Beebe, Nicole, (2009) "Digital forensic research: The good, the bad and the unaddressed." Springer Berlin Heidelberg, In *Advances in digital forensics V*, pp. 17-36.

[42] Poisel, Rainer & Simon Tjoa, (2011) "Forensics investigations of multimedia data: A review of the state-of-the-art", IEEE, In *IT Security Incident Management and IT Forensics (IMF), 2011 Sixth International Conference on*, pp 48-61.

AUTHORS

Raj Kumar Pahade is a MTech scholar at the department of computer science and engineering of Defence Institute of Advanced Technology (DIAT), Pune-India. He received BTech in computer science and engineering from Maulana Azad National Institute of Technology (MANIT), Bhopal. His work is currently focused on Multimedia File Carving and Digital Forensics.

Bhupendra Singh is a PhD scholar at the department of computer science and engineering of Defence Institute of Advanced Technology (DIAT), Pune-India. He received MTech in computer science and engineering from Central University of Rajasthan. His work is currently focused on File System Analysis and Digital Forensics.

Upasna Singh is an assistant professor at the department of computer science and engineering of Defence Institute of Advanced Technology, Pune-India. She got her PhD in computer science from Indian Institute of Information Technology, Allahabad (India). His research activity is based on Data Mining and Knowledge discovery, Machine Intelligence, Soft Computing, Digital Forensics, Social Network Analysis and Big Data Analytics.