**Project Icarus**

1Eugene Tan Wei Jie, 2Foo Qi Kai, 3Andy Chua Chiang Sheng, 4Zulfadli Bin Johari, 5Muhammad Azfar Bin Adam

*Information Security, Singapore Institute of Technology  
Singapore*

11902125@sit.singaporetech.edu.sg

21902183@sit.singaporetech.edu.sg

31902138@sit.singaporetech.edu.sg

41902147@sit.singaporetech.edu.sg

51902165@sit.singaporetech.edu.sg

***Abstract*** -This article highlights how critical time is when performing a static data analysis on a targeted system. In addition, it covers our proposed solution which will highlight portability issues in digital forensics investigations that are not yet solved.

***Keywords -* Digital Forensics, Android, Cellebrite, Static Data Analysis, On-The-Go Analysis, Portability**

1. Introduction

The rise of digital crime has led to higher demand and expectation from the community of professionals in the digital forensics sector. Yet, researchers and practitioners [1] agree that the increasing volume of data and the time taken to investigate are not cost-effective.

One of the core, yet time-consuming tasks in an investigation is using computer forensics tools to perform analysis. How and what kind of tools used will proportionally affect the amount of time taken before the outcome of an investigation are determined. Research article [2] expressed the need for fast and portable tools to perform analysis, as it affects the survivability rate of child abduction, death threats and kidnapping victims. Hence, it is dire to pursue fast, portable and cost-effective computer forensics tools to reach quicker conclusions in any case.

1. The Problem

Finding evidence is critical when dealing with a crime and finding these evidences is usually a race against the clock as there is a large amount of data to examine through and to locate the necessary information on a case that might be time-sensitive but needing to bring evidence from one location to another is time wasted, existing solution for on-site forensic has shown to be limited in their functionality.

Portability is another issue to be addressed. Evidence acquired on-site usually cannot be analysed there and then without the right equipment. Most computer forensics tools are generally available as a stand-alone application or found within expensive forensic workstations. There exists a need for forensic tools with the ability to perform forensic data acquisition, examination and analysis while out in the field and not being tethered to a laboratory.

1. Analysis Of Issues & Challenges Faced

Article [3] highlights an issue with performing forensic triage in “real-time”. It talks about how real-time systems (RTS) have a short amount of time (usually in milliseconds) to react to external input. To better illustrate this, the article uses the example of how an aeroplane’s autopilot feature is required to keep track of input from a large amount of sensors and respond accordingly. It stresses the importance of quick responses during time-critical situations since failing to detect system failures can result in dire consequences. As such, being able to perform forensic analysis and examination during the time when those systems are “live” would serve as a better form of forensic evidence [3].

Another research paper [4] emphasizes the significance of a timely acquisition of forensic evidence. Rogers et al (2006) proposed the Computer Forensic Field Triage Process Model (CFFTPM) aimed at improving a forensic investigator’s capability to acquire valuable information during execution time of a warrant at the initial crime scene. This method worked best when carried out during the first few hours of the investigation. The reason for this was because the alleged perpetrators were more inclined to reveal additional information and cooperate with the authorities during that time and given their familiarity with the crime scene [4]. This proved how ‘time’ itself psychologically affects suspects and is a key factor in helping to resolve criminal forensic investigations.

On the issue of portability, in India, the task force that surveys a crime scene often includes a forensic investigator which is tasked with acquiring on-site digital evidence. It is seen in article [5] that one of the main hindrances that the teams usually face is the dilemma in the portability of forensic equipment.

Another article also agrees with the notion of this fact by stating that it is a must-have for forensic investigators to own and utilise portable tools that can be brought on-site to aid in the screening of forensic evidence before being brought back to the labs to undergo a more in-depth analysis [6].

Some examples of applications that can perform forensic analysis are Encase Forensic [7], Autopsy [8], Forensic Toolkit (FTK) [9] and The Sleuth Kit [10]. They are effective in providing an interface for investigators to perform analysis and extract information. However, these tools are often only found in forensic workstations which are big, heavy and not feasible to be brought to a crime scene. This thus prevents forensic investigators from conducting an immediate on-site forensic triage.

Forensic workstations employ the use of computer forensic science which is aimed at protecting the integrity of acquired forensic data. Creating forensically sound copies of the evidence assures everyone in a court case that the data was not in any way tampered with during a forensic investigation by an expert investigator [11].

Article [11] lists three types of forensic workstations. They are usually classified as stationary workstations, portable workstations or lightweight workstations.

TABLE I

Types of Forensic Workstations

|  |  |
| --- | --- |
| **Type** | **Description** |
| Stationary Workstation | A tower with several bays and many peripheral devices |
| Portable Workstation | A laptop computer with almost as many bays as peripherals as a stationary workstation |
| Lightweight Workstation | A laptop computer with a specialised selection of peripherals |

Stationary workstations like “FRED Forensic Workstation” [12] and ”SUMURI TALINO KA-301” [13] weigh about 40 kilograms and are generally located in forensic labs.



Fig. 1 Example of a stationary workstation - FRED Forensic Workstation

A way to circumvent the long travel time back to the labs is by bringing along portable workstations such as SiForce Nano X [14] and FREDDIE Ruggedized Mobile [15]. This drastically speeds up the investigation as forensic investigators can now perform a full forensic triage at the crime scene. Be that as it may, their weight and size again make it inconvenient for the forensic team. Should the investigators choose not to bring those workstations, they will have to acquire the evidence file on-site and travel back to the forensics lab and only then are they able to perform a forensic analysis.

This led to the production of lightweight workstations. Lightweight workstations are meant to cater to a variety of forensic investigation cases and are reliant on what is needed to be analysed [11]. Cellebrite UFED [16] and Tableau TX1 Forensic Imager [17] are some examples worth noting.



Fig 2. Example of a portable workstation - SiForce Nano X Forensic Workstation



Fig 3. Example of a lightweight workstation - Tableau TX1 Forensic Imager

However, the cons of these workstations are that they are not all-encompassing. This means that they do not offer as many tools as stationary and portable workstations. Most often, what is lacking is the ability to analyse and examine the acquired data which means the investigator must head back to the lab before a proper analysis can be performed. As the need for forensic investigation may be abrupt as well, forensic investigators will not be able to save time if they are not able to access portable or lightweight workstations since they are initially not in the lab.

There are a few notable challenges worth mentioning that came up in our research. It is found that performing a forensic examination and analysis largely depends on its acquired data set and the workstation’s own computing power and processing speed. The process in itself is resource intensive. Indeed, this is bad news for lightweight workstations as they are not as powerful as their other counterparts. As a result, it would take much longer for lightweight workstations to analyse forensic evidence. The time taken to analyse the data is often proportional to its data size. This is further supported by another article [18] which states that “The reality is, however, that the sheer (and growing) amount of processing needed to perform digital investigations largely negates the inherent advantages of automated processing. In other words, finding the evidence on a single hard drive may take the same amount of time as finding it in an archive room full of paper documents.”. The article [18] also mentions some examples through experimentation. Analysing a 6GB hard disk using FTK [9] took about two hours. Under the same conditions, it took more than four days to analyse an 80GB one [18]. To summarise, this means that lightweight workstations are restricted to only examining and analysing small data sizes unless given an irrationally large amount of a time window.

1. Existing Solutions, Tools, Approaches

On top of lightweight workstations, there are other solutions, tools and approaches that look to assist forensic investigators with their investigations.

Encase Forensic [7] is one that is commonly utilised to recover data from hard drives. It creates an image of the data to be used in a more in-depth forensic analysis. Operations like searching by keywords and the recovery of data can be performed [18].

FTK [9] is built based on a database model. Right after the imaging process, a comprehensive analysis of drive images is done. This process takes a lot of time. However, operations like keyword search, file identification and file recovery, etc. are considerably quicker [18].

Both the tools listed are some of the few approaches that are typically used to solve the issue of time and portability. As a side note, a workstation can be used to house the applications for them to leverage on the resources of the workstation.

1. Proposed Solution

The suggested solution is to have a mix of what a lightweight workstation will offer in terms of portability and convenience coupled with the ability to perform an on-site analysis. An example of such a device is the existing portable workstation, Forensic Airlite VIII I7 [19].



Fig 4. Forensic Airlite VIII I7 equipment

This is an equipment that is available to do forensic on-site. It comes with extra accessories which need to be connected to the device and allows it to be used for any forensic activity. They can be detrimental and clunky as it needs to be transported everywhere and it requires an officer or personnel with the necessary digital forensic skill set. A more optimal solution would be to have readily accessible devices such as mobile phones as they are cheap and can be brought anywhere by the investigator. It should come with a program that can handle the necessary analysis and file extraction from a copy of the target system on a mobile phone instead of a workstation and allows personnel without the necessary skill set to be able to do general forensic work . Hence, “Icarus” is proposed with the aim of bridging these traits together on a mobile phone application.

Icarus, a forensic analysis mobile application that makes use of an acquired copied image file stored in an external drive such as a USB Flash Drive to be mounted/connected directly into an Android phone. This allows Icarus to analyse, view and extract the data from the image while still being at the crime scene, allowing investigators to start the forensic examination as early as a copy of the image file is created.

As such, Icarus helps to eliminate the problems highlighted by providing investigators with a reliable portable platform serving as a cheap analysis tool that runs on mobile phones and can conduct analysis whenever and wherever necessary. How Icarus works is by reading the hex values of an image and by programming it with the data architecture of the file system, the program is able to read and extract all the relevant data that is stored in the image file. The program can also carve out all the files within allowing the investigator to view the individual files and its content that might be relevant to a crime investigation. It also provides a general information of the image, for example, how big the file system is, what system the file image is using as well as how much of the system is being used, etc.

1. Comparison Against Existing Solutions

There are many Digital Forensic tools out in the market many of which are well-known and serve as a go-to solution for almost every digital forensic work. Some examples include tools such as Cellebrite [16], FTK imager [9] and Autopsy [8]. This section compares Icarus with those tools and lists the individual advantages, limitations, how unique of an idea Icarus is as well as its similarity with those tools.

1. *Cellebrite*

Cellebrite is a hardware-independent forensic tool for mobile devices used for extracting, analysing and presenting data within the phone. Cellebrite uses the vulnerabilities of the mobile device, in this case we take the hardware and its example, the processing chip to gain access to the device. Another way Cellebrite extracts data through the use of the mobile device APIs. The steps taken include ensuring the phone and the workstation is connected as well as sending a request of information via the API. The mobile device will then respond with the corresponding information [20].

1. *FTK Imager:*

FTK imager creates an image of the hard drive in a single file or in segments that can be reconstructed later.

TABLE II

Icarus VS FTK Imager

|  |  |
| --- | --- |
| **FTK Imager** | |
| **Advantages** | It can create an exact match of the target hard drive/system and is applicable for many different types of system for forensic work |
| **Limitations** | It only allows you to create a copy image of the hard drive |
| A preview of the file is allowed but not any other type of forensic action |

1. *Autopsy:*

Autopsy [8] is a file system analyser that analyses all major file systems and unpack archives for autopsy to be able to scan and view any data from any file system. At least 1 of the major file systems is being used globally. This makes Autopsy one of the best and most used forensic tool.

TABLE III

Icarus VS Autopsy

|  |  |
| --- | --- |
| **Autopsy** | |
| **Advantages** | It can view any target system as long as it is using one of the major systems |
| It can create a report summarizing recent activity and its timeline |
| It can recover corrupted and deleted files as long as the cluster of the original file has not been overwritten |
| **Limitations** | Not all data can be recovered as some files might require a more specialised tool for file recovery |

1. *Icarus:*

Icarus is an application that can be installed on any android device. It can perform a scan of a hard drive image powered by the USB power voltage supplied by the phone.

*1) Similarities*: Icarus is similar to many solutions or approaches that can also perform a forensic analysis on evidence and then display the files within the evidence image.

*2) Uniqueness*: It is the first ever application to be developed that allows a system image to be scanned and information displayed on a mobile phone.

In conclusion, even though it is the first of its kind, it contains similarities with existing products and is not without its own limitations. This is mostly due to hardware limitations already present on the mobile phone. A mobile phone can only power devices that have low power consumption and analysing file images requires a large amount of processing power in which a CPU of a mobile phone might not have. Therefore, the process will be slower compared to other programs that make use of a dedicated workstation CPU which has higher processing speed than that of a mobile phone. Yet, if these limitations could be overcome, Icarus could be a new step in a portable mobile digital forensic tool.

1. Technologies Relied On

The Technologies used to create the solution are explained below. They include Java, Android Studio and Android Software Development Kit (SDK).

1. *Java:*

Java being a proprietary language for android and being one of the two official options; it receives a lot of support from Google. Even as Google has made apparent its preference for Kotlin in android development, Java is the more familiar language [21]. Java is an object-oriented programming language with helpful functions like constructors, null pointer exceptions, checked exceptions and more. Even though Java has certain disadvantages such as the inability for multi-inheritance, they encourage clean codes which results in higher clarity of codes. Not to mention, Android is predominantly coded in Java and hence will suit our market audience [22].

1. *Android Studio:*

Android Studio being the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. Additionally, it comes packaged with the Android SDK and this provides everything we need in one place to get up and running. Furthermore, it comes built in with an android emulator which can simulate android devices hence, allowing for efficient testing of application on a varied number of devices and Android API Levels without the need of physical devices [23]. On top of that, it is simpler to transfer data into the emulator as compared to a physical device. This is the primary IDE that this project will be using to code out the application due to its robust nature and it comes pre-bundled with applications which provides for efficient and streamline coding [24].

1. *Android SDK:*

Android SDK is a toolset that enables us to create apps for Android OS. It consists of the required libraries to build Android apps like, a debugger, an emulator, Application Programming Interfaces (APIs) and some sample projects with source code, allowing us to kick start programming the application quickly. Furthermore, Android SDK tools are platform-independent and are required to create any Android app, regardless of the version it is being developed for. Apart from the abovementioned, the most important SDK tool that is utilised include the Android SDK Manager, which provides the resources, platforms and other useful components needed to build apps and manage SDK packages. One such component is the Android Virtual Device Manager; it provides a graphical user interface to test and compile our application on a virtual device. The Dalvik Debug Monitor Server (ddms) also allows for ease of debugging Android apps [25].

1. Solution Architecture & Engineering

Existing solutions like Cellebrite and FTK imager have improved the portability trait in the digital forensics market. However, the existing solutions require the need for a computer to perform analysis which may not be as accessible as compared to a mobile phone. Thus, our solution, Icarus helps to eliminate this problem by providing investigators a portable platform to conduct analysis whenever and wherever necessary. Icarus has two main functionalities in its program, one is displaying the image’s file system information while the other is data carving. The program flow of Icarus which can be seen in Figures 7 and 13 will be elaborated in detail at the relevant sections.

1. *Brief Description of Master Boot Record (MBR):*

The MBR is the information in the first sector of any hard disk that identifies where an operating system is located so that it is able to be booted into the computer's main storage space or Random Access Memory (RAM). The Master Boot Record is also sometimes known as the "partition sector" because it includes a table that identifies each partition which the hard disk has been formatted into. MBR is recognised due to its unique characteristic coding containing the hexadecimal strings “55” and “AA” in two bytes, which can always be found at the end of the MBR sector. If this information is missing, the master boot sector will be unidentifiable, and the boot process will be aborted with an error message. Figure 5 seen below describes the useful information being retrieved from the MBR.

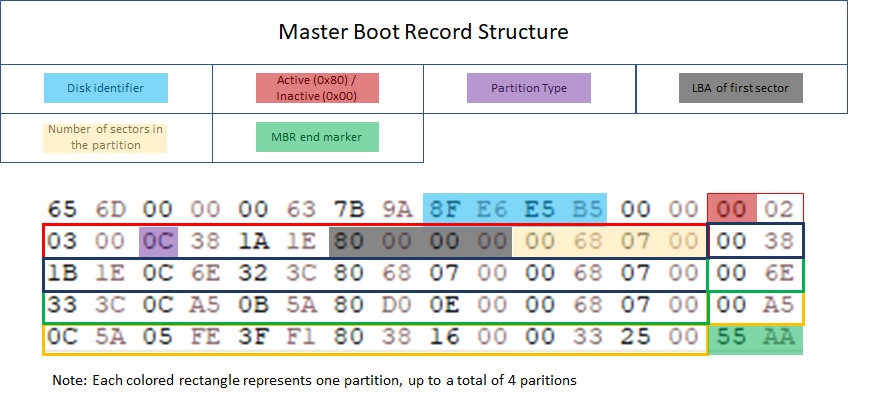


Fig. 5 Information extracted from Master Boot Record

1. *File Systems and Fragmentation:*

Before describing our chosen file system for this project and how its files are stored, we want to briefly cover the physical blocks on a disk that are used to store data. Disks are typically described in terms of data blocks called sectors and clusters. The cluster size is typically calculated using the formula shown below. It is important to note that a cluster represents the smallest unit of storage that it can be written to or read from. Files are typically stored in clusters. Typical values of clusters range from 512–32,000 Bytes where cluster sizes are normally multiples of 512 Bytes [26].

Cluster Size = Sector Size \* Sectors per Cluster

While there are currently many file systems in use today, for this project we will be using the well-known and commonly used file system known as file allocation table (FAT).

1. *Brief Description of FAT32:*

FAT is the simplest file system type. It consists of a boot sector, file allocation table, and storage space to store files and folders. FAT has many variations, but for this project we will be focusing on FAT32. FAT32 is compatible with Windows-based storage devices. The file allocation tables, and the root folder must also be stored in their designated locations so that the files needed to start the system can be correctly located [27]. A volume formatted with the FAT is allocated in clusters. The size of the volume determines the default cluster size. For the FAT32 file system, the cluster number must be in 16 bits and is a power of two. The following Fig. 5 shows the information being extracted from the FAT32 Boot Sector.

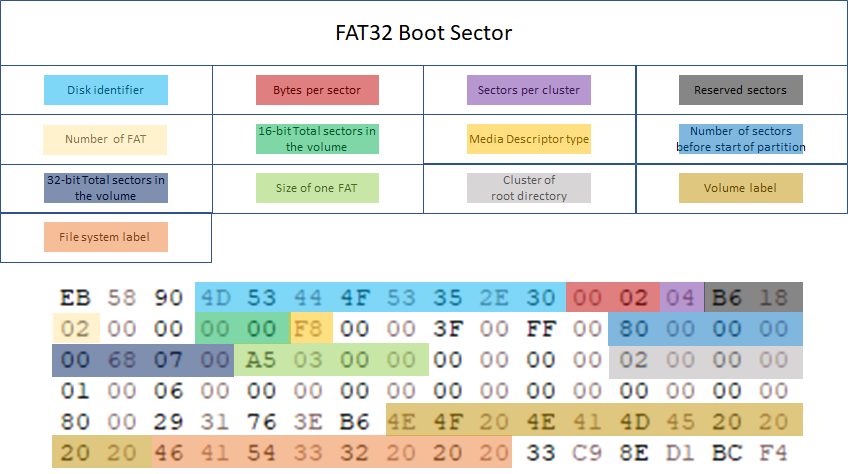


Fig 6. Information extracted from a FAT32 Boot Sector

1. *Program Flow for displaying the image’s file system information:*

After fully understanding how the FAT32 file system works and how the MBR is structured, we can extract the relevant information that a user would need to see in order to have a clear analysis of the image file. This is demonstrated in the flow chart seen below in Fig. 7.

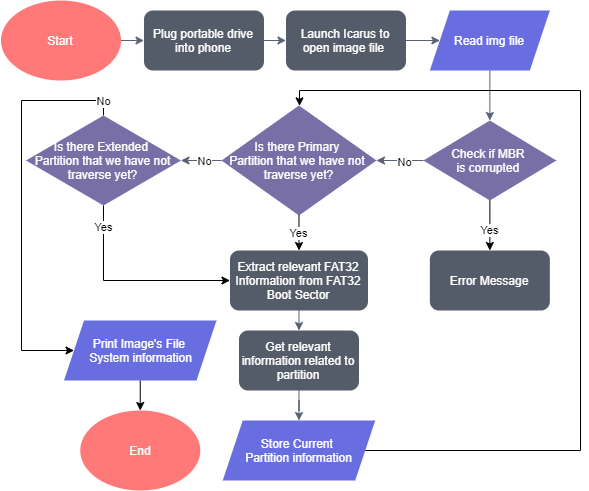


Fig. 7 Flowchart on displaying file system information

1. *Allocation of files in File Allocation Table (FAT32):*

File allocation in FAT-32 is a two-step process. First, the file is allotted to a directory/folder. If the assigned directory/folder is at the root of the drive, then the file name is kept within the root directory entry, if not, one should take apart the root directory entry to seek out the directory that is in the path of the final directory to that which the file is assigned to. Together with the file name, the file entry contains the beginning cluster number of the file. The starting cluster number represents the cluster at which the file’s contents begin. Extra information, together with access, creation, and modification time stamps in conjunction with long file names are kept within the directory entry. To retrieve the file, the file system first looks at the beginning cluster then goes to the starting cluster index in the file allocation table. The FAT is often thought of to be a linked list of cluster numbers pointing to subsequent clusters of a file. Therefore, once a file is retrieved, the file system goes to the starting cluster index in the FAT and gets the next cluster number for that file. This cluster will then point to a different cluster. This method is repeated till a cluster has the hex value “FF” indicating end of file (EOF). The file system is then able to retrieve the file by reading the clusters on the disk. [26]

An example of this is creating a file called “recovery.txt”, which requires five clusters to store its contents. The file system may find that the cluster number “10” is where the file should start to be stored. As seen in Fig. 8, a root entry for “recovery.txt” is made indicating the starting cluster of the file shown as cluster number 4. The file system stores the remaining clusters in cluster numbers 11, 12, 15, 16 and 17 and makes the appropriate changes to the FAT, while clusters 13 and 14 are used for storage of another file “hello.txt.”

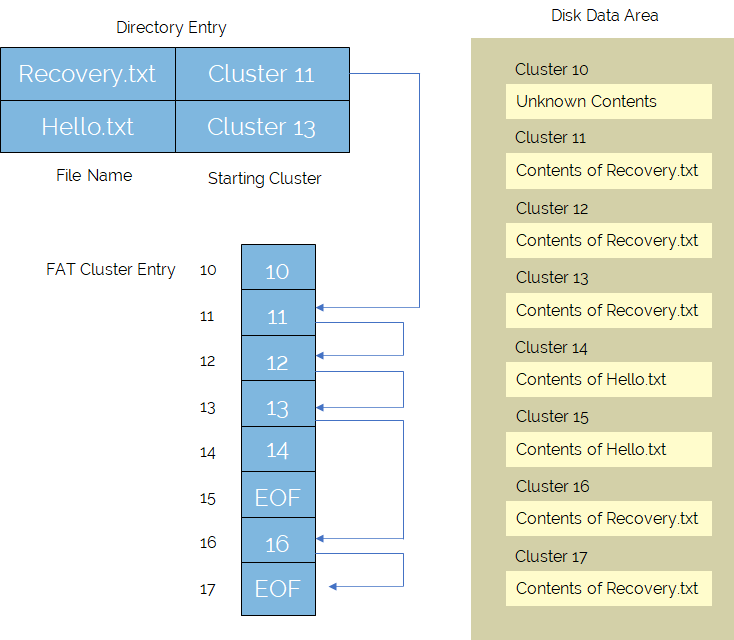
****

Fig. 8 FAT32 file allocation

1. *File Structure-Based Carvers:*

File carving does not necessarily require the usage of file system information directly to recover files. Instead, it makes use of information in the structure of files. Experienced carvers no longer solely rely on the knowledge of the structure of files but also use the contents of individual files to carve data. Given a digital storage device, a file carver may or may not understand the file system being used, and it may or may not trust the information in the file system to be accurate. As a result, it is the responsibility of the carver to determine the “unallocated” clusters in the disk to carve from. This may involve all clusters in the disk.

Inexperienced file carvers use “magic numbers,” or to be more precise, byte sequences at prescribed offsets to identify and carve files. File carving techniques were first used for files that contain a “header” and “footer.” A header identifies the starting bytes of a file while the footer identifies the ending bytes of the file. Headers are used by operating systems to determine which application to open a file with. For example, a jpeg file’s starting clusters must begin with the hex sequence “FFD8”, while its footer is a cluster containing the hex sequence “FFD9”. Inexperienced file carvers will rely solely on structure and will attempt to combine and return all unallocated clusters between the header and footer.

However, there are many file types that may not contain footers but instead contain other information like the file size. An example of this would be Windows BMP file that contains the size of the file in bytes. Ultimately, this results in file structure-based carvers simply extracting data between a known header and footer (or ending point determined by size), with the assumption being that the file is not fragmented and there is nothing missing between the header and the footer. As a result, these carved files frequently have “garbage” in the middle. Therefore, it is important to rely on the information provided in the root directories and co-relate it to the fat table to allow for accurate carving of files, which will be elaborated in the following section [26].

1. *FAT32 Directory structure & File Allocation Table:*

Since using magic number is a trivial way of carving data, we must rely on the relevant information provided in the directory, in order to carve out the file correctly as seen in Figs. 8, 9 and 10.

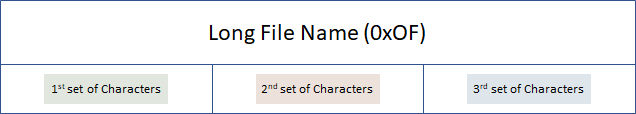


Fig. 9 Information extracted for long file name

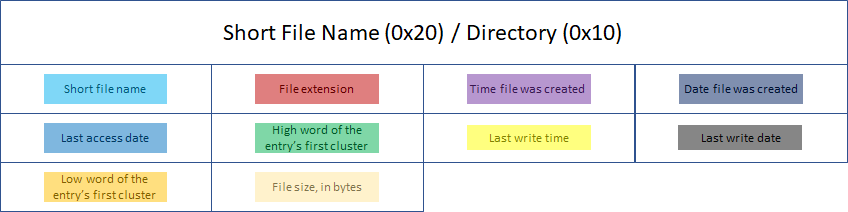


Fig. 10 Information extracted for short file name

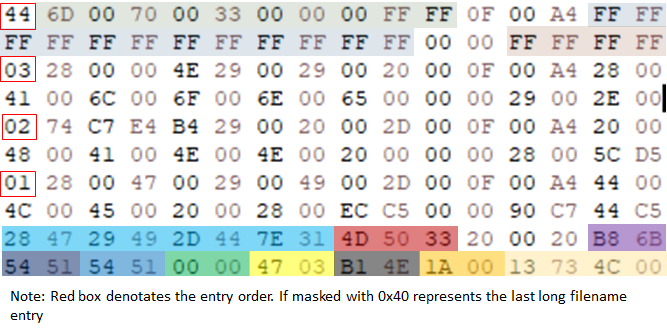


Fig. 11 Information extracted from root directory

Based on the relevant information provided by the FAT32 directory, it is insufficient to provide a clear way to carve out the relevant files without error, hence we need to cross reference with the FAT32 file allocation table as seen in Fig. 12 below, so as to ensure that we are able to carve out the data from the correct clusters.

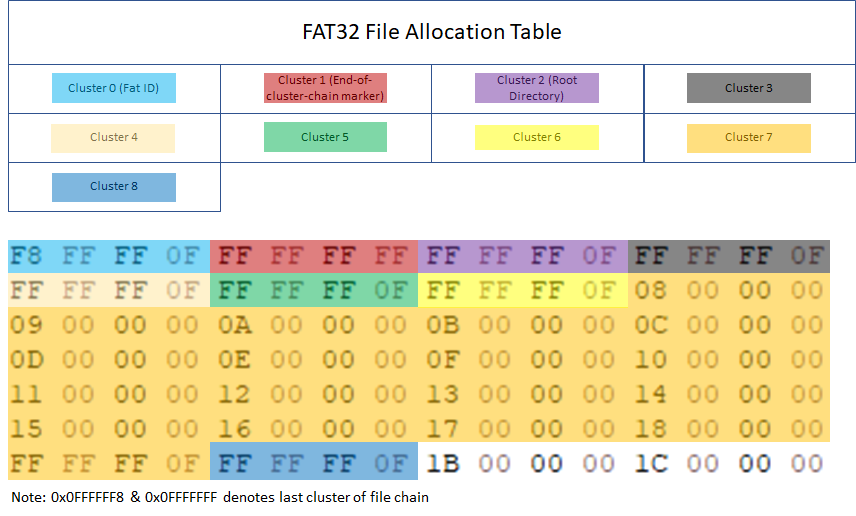


Fig. 12 FAT32 File cluster identification

1. *Program Flow for data carving:*

Now that we have a firm understanding on how to properly carve out the data, Icarus will have the following program flow to carve out the relevant data from the FAT32 partition, which is seen in the Fig. 13. Furthermore, it is also important to note the formula below on how to calculate offset of files or directories.

Start Sector of cluster area + (start cluster - 2) \* (number of sectors per cluster)

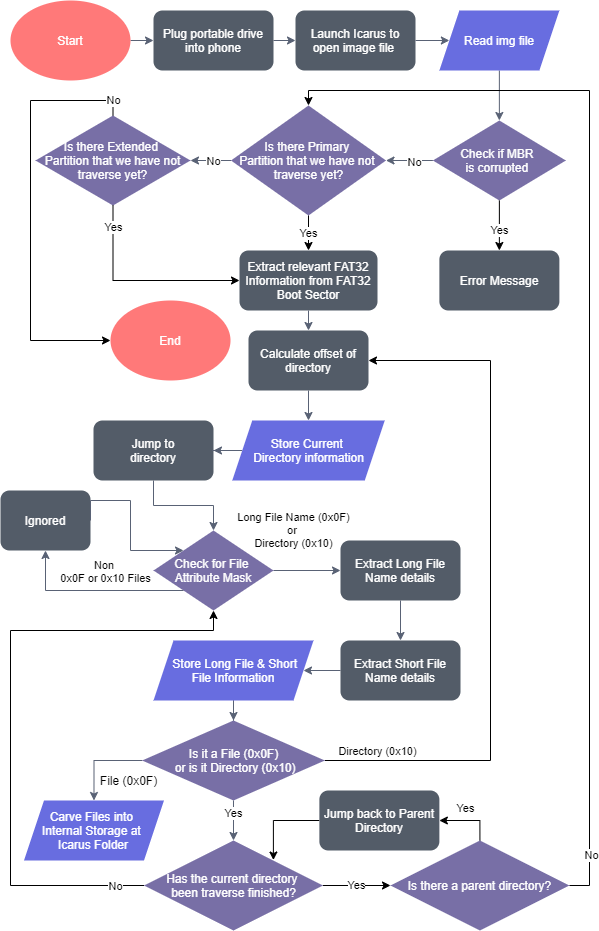


Fig. 13 Flowchart on data carving process

1. *Data Structure:*

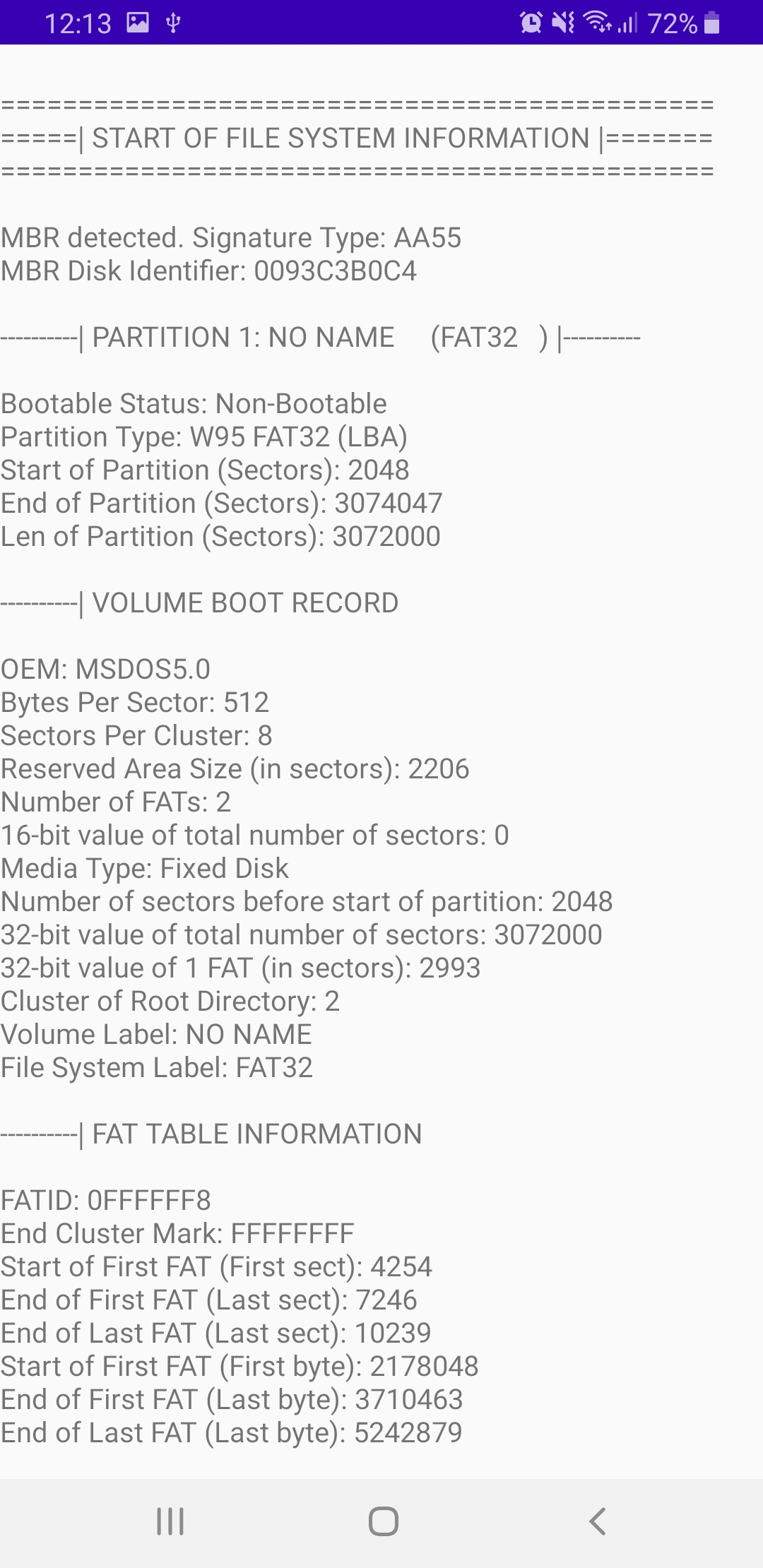
The type of data structure used for the solution is extremely important in ensuring the efficiency of the analysis.

ArrayList is one of the data structures we have chosen to store some of our variables such as the FileEntry and the DirData. Despite it being slower than Array when appending and searching for an item, ArrayList provides us the flexibility to change the size of the list which Array doesn’t. This is relevant to our solution as the size of the variables used varies between disk images. Additionally, the speed of ArrayList is negligible when compared to Array.

Additionally, we have made use of StringBuilder in several scenarios like grabbing the hex data into a mutable string. This also allows us to convert the hex data into ASCII easily. StringBuilder is also not thread synchronized, which means that it will be safe for us to implement since our solution does not utilize multithreading.

1. Test Setup & Results

We have created a few images which contained the FAT32 filesystem with MBR Partitioning Schemes to simulate evidence images found in the field. In order to test the “Analysing” functionality, we have done a manual calculation based on extensive research and cross referenced it with the results produced by Icarus for the purpose of verification.

   
Fig. 14 Output from “Analysis” Function.

Additionally, the information can be verified against the “Carving” function as well since “Carving” is highly dependent on the right results from “Analysis” to carve.

Hence, we can attempt to carve out files and cross reference with the results such as if the number of files and directories will match the actual files and directories content on our image.

  
Fig 15. Output from “Carving” Function

Another method we utilised to check if the content was the same was by comparing the hash values.

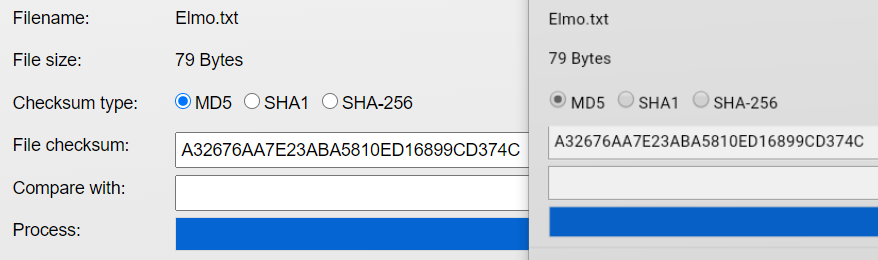


Fig 16 Checking on hash value

As the hash value is the same, we can confirm that the file carving indeed works based on the values provided from the analysis.

Hence with these checks, we can confirm that Icarus is successfully working.

1. Limitations & Future Works

This project proposes a new portable way of performing analysis using Android devices. While examiners using actual forensic workstations is still the ideal way of conducting examinations. However, it is difficult to perform them in a timely manner due to travel time and other unforeseen circumstances. Therefore, this method allows the bridging of the time gap. As the project is still in its infancy stage, there exists a few limitations. Firstly, the file carving process is slow. Another limitation is that the application is made to only detect FAT32 filesystems and the MBR Partition Scheme. It is also unable to consider deleted files and orphan files. Lastly, it is unable to read in the manipulated FAT32 file system. It is important to develop ways to rectify the above issues. The tool developed in this research has minimum functionality, and for a practical anti-forensic tool, the tool should be improved for completeness, with the primary focus being on improving the speed in which the data is being carved out.

1. Conclusion

Our research led to the realization that the time taken to perform analysis increased the larger the size of the data set or evidence image. This led to the conclusion that lightweight workstations and similar solutions are best at analysing small sizes of data.

Despite that, our research and end result have proved that the idea put forth by the team was indeed possible. This also meant that we had achieved the goals we had set for ourselves and have developed a product that, albeit still in its infancy stage, managed to address the issues of time, portability and accessibility which could further aid forensic investigators in the field in the midst of conducting their forensic investigation.

Should the project have been more successful, other areas of research that could be considered for the improvement of the application would probably include the ability to analyse other filesystems and partition schemes with the aim of making it a more holistic application better suited for its purpose in the field of digital forensics.

References

1. M. Al Fahdi, N. L. Clarke and S. M. Furnell, "Challenges to digital forensics: A survey of researchers & practitioners attitudes and opinions," 2013 Information Security for South Africa, Johannesburg, 2013. Available: [https://ieeexplore.ieee.org/document/6641058/authors](https://ieeexplore.ieee.org/document/6641058/authors#authors)
2. K. Rogers, M., Goldman, J., Mislan, R., Wedge, T., & Debrota, S. (2006). Computer Forensics Field Triage Process Model. The Journal of Digital Forensics, Security and Law 1(2). Available: <https://commons.erau.edu/cgi/viewcontent.cgi?article=1004&context=jdfsl>
3. Roussev, V., Quates, C. and Martell, R., 2013. Digital investigation [online] https://www.sciencedirect.com. Available at: https://www.sciencedirect.com/science/article/abs/pii/S1742287613000091?casa\_token=OGYHuaOnKGwAAAAA:RonGuPKgT76jDwgQyFg3c5OjjItI\_xUx5V3IK0yDPFvqsibhbNq0S0OJYctmmiICGJDVcNg7qc8.
4. E. Gentry and M. Soltys, "Procedia Computer Science System. https://www.sciencedirect.com/, 2019. [Online].Available:https://www.sciencedirect.com/science/article/pii/S1877050919315364
5. P. Mali, "Low Cost and Ultra Low Cost Digital Forensic Imaging Devices," International Journal For Science And Advance Research in Technology(IJSART), vol. 4, no. 1, pp. 155-160, 2018.
6. C. Mulligan and A. O'Leary, "Accessing the Probative Value of Physical Evidence at Crime Scenes with Ambient Mass Spectrometry and Portable Instrumentation", Ncjrs.gov, 2015. [Online]. Available: https://www.ncjrs.gov/pdffiles1/nij/grants/248884.pdf
7. OpenText Corp., “Encase Forensic,” 2020. [Online]. Available: <https://www.guidancesoftware.com/encase-forensic>
8. Basis Technology, "Autopsy Digital Forensics," Basis Technology, 2020. [Online]. Available: https://www.autopsy.com
9. AccessData Group, Inc., “Forensic Toolkit (FTK),” 2020. [Online]. Available: [https://accessdata.com/products-services/forensic-toolkit-ft](https://accessdata.com/products-services/forensic-toolkit-ftk)k
10. Brian Carrier, “The Sleuth Kit,” 2020. [Online]. Available: <https://www.sleuthkit.org/>
11. A. K. Gupta, G. S. Sodhi, Rarh Vimal, V. N. Sehgal, and M. Moinuddin., “Cyber Forensic Workstation,” New Delhi, India. Available: <http://epgp.inflibnet.ac.in/epgpdata/uploads/epgp_content/S000016FS/P000703/M015237/ET/1464340970FSC_P16_M34_e-text.pdf>
12. Digital Intelligence Inc, “FRED Forensic Workstation,” 2020. [Online]. Available: <https://digitalintelligence.com/store/products/fred>
13. SUMURI LLC, “TALINO KA-301 Forensic Workstation,” 2020. [Online]. Available: <https://sumuri.com/product/talino-ka-301-forensic-workstation/>
14. Silicon Forensics Inc., “SiForce Nano X (Hardware Defined Forensics),” 2020. [Online]. Available: <https://siliconforensics.com/products/forensic-workstations/forensic-laptops/siforce-nano-ii-hardware-defined-forensics-1.html>
15. Digital Intelligence Inc, “FREDDIE Ruggedized Mobile,” 2020. [Online]. Available: <https://digitalintelligence.com/products/freddie>
16. Cellebrite, “UFED Ultimate,” 2020. [Online]. Available: <https://www.cellebrite.com/en/ufed-ultimate/>
17. Forensic Computers Inc, “TABLEAU TX1 Forensic Imager,” 2020. [Online]. Available: <https://www.forensiccomputers.com/tableau-tx1-forensic-imager.html>
18. V. Roussev and G. Richard, "Breaking the Performance Wall: The Case for Distributed Digital Forensics", Dfrws.org, 2004. [Online]. Available: https://dfrws.org/wp-content/uploads/2019/06/2004\_USA\_paper-breaking\_the\_performance\_wall\_-\_the\_case\_for\_distributed\_digital\_forensics.pdf
19. https://www.forensiccomputers.com/workstations/mobile-workstations/forensic-airlite-viii.html. 2020. Forensic Airlite VIII I7. [online] Available at: https://www.forensiccomputers.com/workstations/mobile-workstations/forensic-airlite-viii.html
20. "A technical look at Phone Extraction", Privacy International, 2019. [Online]. Available: https://privacyinternational.org/long-read/3256/technical-look-phone-extraction#:~:text=Cellebrite's%20UFED%20Touch%20
21. T. Patrick, "The Relationship Between Android and Java", www.theiconic.tech, 2017. [Online]. Available: https://theiconic.tech/android-java-fdbd55aadc51.
22. A. Sinickl, "I want to develop Android apps — What languages should I learn?", Android Authority, 2019. [Online]. Available: https://www.androidauthority.com/develop-android-apps-languages-learn-391008/
23. "What is Android Studio and Android SDK tools? - Android edX Community", http://androiddeveloper.galileo.edu/, 2017. [Online]. Available: http://androiddeveloper.galileo.edu/2017/03/29/android-studio-and-android-sdk-tools/
24. "Meet Android Studio | Android Developers", Android Developers, 2020. [Online]. Available: https://developer.android.com/studio/intro
25. "Create and manage virtual devices | Android Developers", Android Developers, 2020. [Online]. Available: https://developer.android.com/studio/run/managing-avds
26. N. Memon and A. Pal, "The evolution of file carving", researchgate.net, 2009. [Online]. Available: https://www.researchgate.net/publication/224397415\_The\_evolution\_of\_file\_carving
27. "File Carving", resources.infosecinstitute.com, 2018. [Online]. Available: https://resources.infosecinstitute.com/topic/file-carving/