Mobile Forensic Images and Videos Signature Pattern Matching using M-Aho-Corasick

Yusoof Mohammed Hasheem, Kamaruddin Malik Mohamad, Ahmed Nur Elmi Abdi, Rashid Naseem
Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia
Batu Pahat, Malaysia

***Abstract*—Mobile forensics is an exciting new field of research. An increasing number of Open source and commercial digital forensics tools are focusing on less time during digital forensic examination. There is a major issue affecting some mobile forensic tools that allow the tools to spend much time during the forensic examination. It is caused by implementation of poor file searching algorithms by some forensic tool developers. This research is focusing on reducing the time taken to search for a file by proposing a novel, multi-pattern signature matching algorithm called M-Aho-Corasick which is adapted from the original Aho-Corasick algorithm. Experiments are conducted on five different datasets which one of the data sets is obtained from Digital Forensic Research Workshop (DFRWS 2010). Comparisons are made between M-Aho-Corasick using M_Triage with Dec0de, Lifter, XRY, and Xaver. The result shows that M-Aho-Corasick using M_Triage has reduced the searching time by 75% as compared to Dec0de, 36% as compared to Lifter, 28% as compared to XRY, and 71% as compared to Xaver. Thus, M-Aho-Corasick using M_Triage tool is more efficient than Dec0de, Lifter, XRY, and Xaver in avoiding the extraction of high number of false positive results.**

Keywords*—mobile forensics; Images; Videos; M-Aho-Corasick; (File Signature Pattern Matching)

I. Introduction

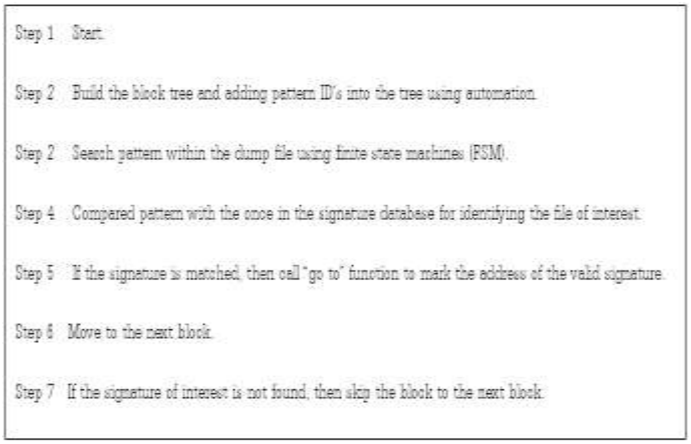
In the last few decades, Digital forensic (DF) plays a paramount part not entirely in availing in cracking cases against mobile phone malefactions like drug dealing, child trafficking, and arms trade. Mobile phone capabilities increase in public presentation, recollection capability and multimedia functionality turning phones into data pools that can fortify a wide range of personal information [1]. Nowadays mobile phone, personal digital assistant (PDA) and the Internet are widely accepted around the world were mobile phone became a component of our quotidian life activities due to rapid development in mobile phone technology. Mobile phone becomes personal and was habituated to avail in multimedia and personal task [2]. However, data held on mobile contrivances can be utilizable and paramount to law enforcement agencies when carrying an investigation in either civil or malefactor transactions. There are two ways of recuperating digital evidence, traditional data instauration, and file carving. The Traditional data integration is a customary technique applied to retrieve digital information where the metadata or file allocation table subsists. While on the other hand file carving was introduced to give assistance in malefactor cases where the traditional data recuperation techniques cannot be worked out. In this paper a new technique called images and videos signature pattern matching using M-Aho-Corasick is proposed to efficiently search for images and videos file from damaged mobile phone using M_Triage tool.

A. M_Aho-Crasick

One of the main components in M_Triage tool that efficiently search for images and videos utilizing multi-pattern signature matching is the M-Aho-Corasick algorithm. The algorithm is habituated and modified from the pristine algorithm kenned as Aho-Corasick, where the failure links function is abstracted and superseded with a signature database which contains all the pattern and file structure that pertains to investigator stored in it [3]. M-Aho-Corasick algorithm has remained constructed utilized for the set of patterns $D=C_1,C_2,...,C_K$ of total length n=|n1|+|n2|+...+|nK|. Entirely the patterns of interest have the same signature. The algorithm search for patterns in such a way that: The algorithm crisscross if a pattern *P* of length *m* is a subpattern in *O(m)* time. The algorithm discovers the main subsistence of the patterns *P 1 ,...,P q* of total length *m* as subpattern in *O(m)* time. The algorithm additionally discovers all *z* existences of the patterns *P 1 ,...,P q* of total length *m* as subpattern in *O(m+z)* time [4].

B. M_Aho-Crasick Algorithm

During the probing in M-Aho-Corasick algorithm, a file pattern like JPEG, 3gp and MP4 are probing by building their signature database, then followed by building the block tree and integrating pattern ID’s into the tree utilizing automation. The signature are probing predicated on finite state machines (FSM) and if the pattern is probing within the dump file, the pattern will be compared with the once in the signature database for identifying the file of interest. If the signature is matched, then *“go to”* function will be called to mark the address of the valid signature and peregrinate to the next block. If the signature of interest is not found, then skip the block to the next block. Fig. 1 shows the probing pattern process utilizing M-Aho-Corasick algorithm.



Step 1 Start.

Step 2 Build the block tree and adding pattern ID's into the tree using automation.

Step 2 Search pattern within the dump file using finite state machines (FSM).

Step 4 Compared pattern with the once in the signature database for identifying the file of interest.

Step 5 If the signature is matched, then call "go to" function to mark the address of the valid signature.

Step 6 Move to the next block.

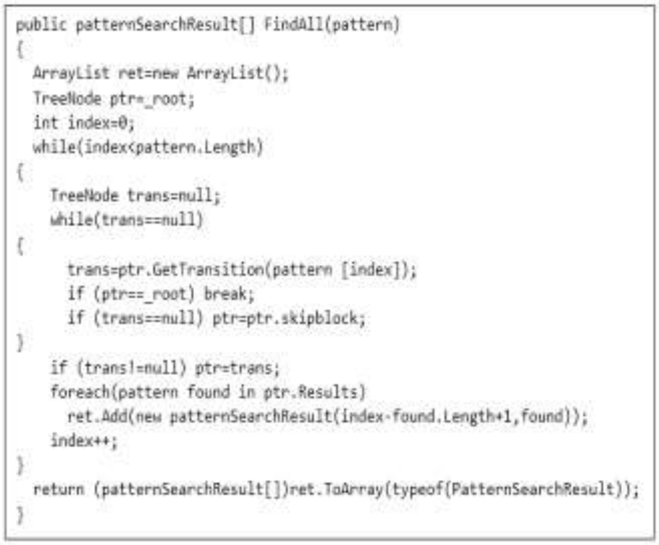
Step 7 If the signature of interest is not found, then skip the block to the next block.

Fig. 1. M-Aho-Corasick algorithm

C. M_Aho-Crasick Implementation

During the implementation an accepted algorithm known as Aho-Corasick (AC) is adapted [3]. The adapted algorithm has a clear implementation and understandable codebase which is illustrated in Fig 2 known as M_Aho- Corasick. Rudimentary, M-Aho-Corasick takes a set of finite pattern file signature as an array and an input file signature and outputs the details on the patterns matched such as their positions in the input signature. Nevertheless, M-Aho-Corasick takes both the input file signature and the set of patterns from reading the files; hence, both sets of the signature can be given as files.

In M-Aho-Corasick design, the algorithm reconstituted the state machines without the failure link transitions and as shown in Fig 3 for image files search and Fig 4 for multimedia file search. The algorithm transmuted the probing method with respect to the failure fewer transitions



```
public patternSearchResult[] FindAll(pattern)
{
  ArrayList ret=new ArrayList();
  TreeNode ptr=_root;
  int index=0;
  while(index<pattern.Length)
{
    TreeNode trans=null;
    while(trans==null)
{
      trans=ptr.GetTransition(pattern [index]);
      if (ptr==_root) break;
      if (trans==null) ptr=ptr.skipblock;
}
    if (trans!=null) ptr=trans;
    foreach(pattern found in ptr.Results)
      ret.Add(new patternSearchResult(index-found.Length+1,found));
    index++;
}
  return (patternSearchResult[])ret.ToArray(typeof(PatternSearchResult));
}
```

Fig. 2. 2 M-Aho-Corasick implementation

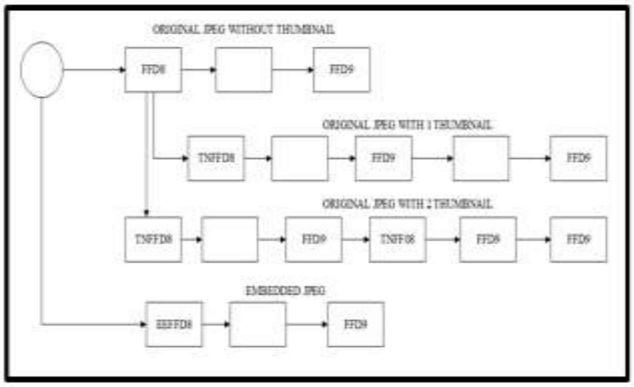

ORIGINAL JPEG WITHOUT THUMBNAIL
FFD8
FFD9
ORIGINAL JPEG WITH 1 THUMBNAIL
TNFFD8
FFD9
FFD9
ORIGINAL JPEG WITH 2 THUMBNAIL
TNFFD8
FFD9
TNFF08
FFD8
FFD9
EMBEDDED JPEG
EEFFD8
FFD9

Fig. 3. Images pattern search

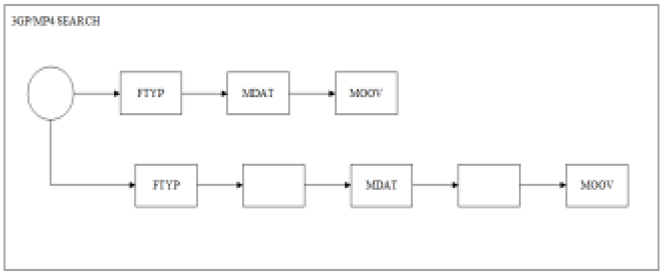


Fig. 4. 3GP/MP4 pattern search

II. Related Work

As it became necessary to find more efficient file signature matching algorithms and their implementations, a significant number of researchers have been being carried out in this area [3]. This literature grants some research work achieved on file signature matching algorithms.

Among the researchers one adapts the original Aho-Corasick and modified it to be known as Parallel Failure less Aho-Corasick (PFAC) implementation, all failure transitions are abstracted from the state machine, to enable the (PFAC) algorithm to probe for a string file in parallel.

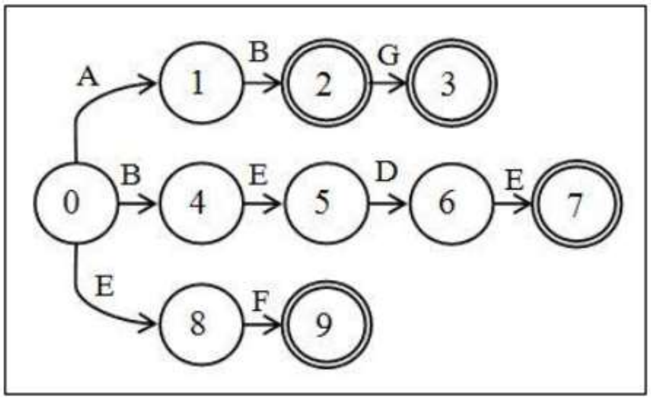


Fig. 5. Parallel Failure less Aho-Corasick (PFAC) [3].

Fig. 5 illustrates the modified state machine for the PFAC implementation for the same four patterns, ABG, BEDE and they used EF in their last example. The four patterns are the target probed files.

In 2011, another researcher [5] developed a mobile forensic triage tool called Dec0de, they acclimates techniques from natural language processing. They propose an efficient and flexible utilization of probabilistic finite state machines (PFSMs) to encode typical data structures. They utilize the engendered PFSMs along with a classic dynamic programming algorithm to find the maximum likelihood parse of the phone's recollection. DEC0DE uses Viterbi Algorithm twice. First, it passes the filtered byte stream to Viterbi with the Field PFSM as input. The output of the first pass is the most likely sequence of generic fields associated with the byte stream. That field sequence is then inputted to Viterbi along with the Record PFSM for a second pass [5]. They refer to these two phases as field-level and record-level inference, respectively. This type of file searching is time consuming because the process doesn't perform in parallel as Aho-Corasick. Furthermore, in 2014 the researcher proposed another tool called LIFTER [6] with the intention to improve the searching performance of their previous tool known as Dec0de by applying the new technique called initial ranking and relevance feedback.

The implement LIFTR's early classification orders single pages that have a file system sheltered; consummately other pages will have a zero quality score after initial ranking. Throughout the pertinence feedback stage, the implement LIFTR endeavors to increment the initial ranking by utilizing investigator feedback [6]. This sort of file searching is also time consuming because the performance of the system depends on the investigator initial ranking and feedback.

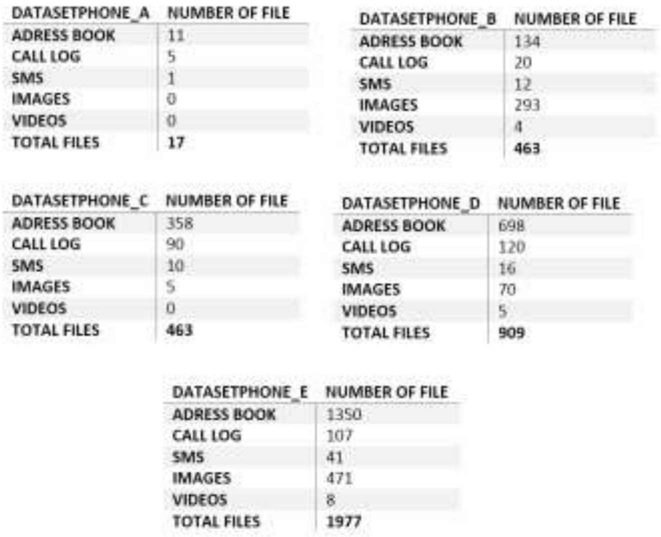
Bulk Extractor is another digital triage forensic tool implements by [7]. Information from digital evidence files like credit card numbers, email addresses, and URLs are extracted correctly using the bulk extractor command-line tool. The tool extracts evidence from the raw disk images. The disk image is split into pages and one or more scanners are used to process the pages after is being split for triage examination [7].

III. Experimentation

This section discusses the experimental setup for M-Aho-Coriasick using M_Triage tool. During the experiment, additional datasets from Digital Forensic Reseach Workshop (DFRWS 2010) which is purposely created to solve the research problems regarding efficient file search. M_Triage is developed to efficiently search for valid address book, call logs, SMS, images, and videos.

A. Dataset Preparation

As mention earlier, Dataset from Digital Forensic Reseach Workshop (DFRWS 2010). And the once extracted using JTAG are chosen as the input to validate the output of the proposed technique. However, due to this flexibility of recovering any leads that might connect Monsieur Victor [8] to other individuals, companies, or bank accounts that are involved in his international arms business, valid address-book, call logs, SMS, images and videos, parameters are considered in this experiment as data of interest to validate M_Triage tool. Fig 6. presents the five data set and their total number of files which is used for the experiment.



DATASETPHONE_A	NUMBER OF FILE
ADRESS BOOK	11
CALL LOG	5
SMS	1
IMAGES	0
VIDEOS	0
TOTAL FILES	17

DATASETPHONE_B	NUMBER OF FILE
ADRESS BOOK	134
CALL LOG	20
SMS	12
IMAGES	293
VIDEOS	4
TOTAL FILES	463

DATASETPHONE_C	NUMBER OF FILE
ADRESS BOOK	358
CALL LOG	90
SMS	10
IMAGES	5
VIDEOS	0
TOTAL FILES	463

DATASETPHONE_D	NUMBER OF FILE
ADRESS BOOK	698
CALL LOG	120
SMS	16
IMAGES	70
VIDEOS	5
TOTAL FILES	909

DATASETPHONE_E	NUMBER OF FILE
ADRESS BOOK	1350
CALL LOG	107
SMS	41
IMAGES	471
VIDEOS	8
TOTAL FILES	1977

Fig. 6. Total number of files in Phone, A, B, C, D and E

B. Experiment on Dataset

- Datase Phone A

M_Triage processed datasetphoneA in 0.21 seconds while de0de in 0.50 seconds, Lifter in 0.30 seconds, XRY in 0.29 seconds while Xaver in 1 minute 20 second.

- Datase Phone B

M_Triage processed datasetphoneB in 1 minutes 10 seconds while de0de in 4 minutes 12 seconds, Lifter in 2 minutes 20 seconds, XRY in 2 minutes 17 seconds while Xaver in 5 minutes 23 seconds.

- Datase Phone C

M_Triage processed datasetphoneC in 4 minutes 20 seconds while De0de in 12 minutes 10 seconds, Lifter in 6 minutes, XRY in 5 minutes 30 seconds while Xaver in 15 minutes 40 second.

- Datase Phone D

M_Triage processed datasetphoneD in 6 minutes 57 seconds while de0de in 24 minutes 30 seconds, Lifter in 10 minutes 50 seconds, XRY in 8 minutes 20 seconds while Xaver in 25 minutes 47 seconds.

- Datase Phone E

M_Triage processed datasetphoneE in 13 minutes 40 seconds while de0de in 60 minutes, Lifter in 21 minutes, XRY in 19 minutes 35 seconds while Xaver in 41 minutes 30 seconds.

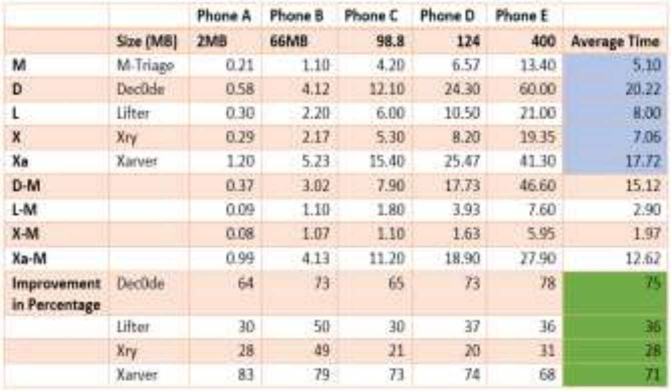
In order to justify the experiment refer to Fig. 7 and 8 for the result.

Universiti Tun Hussein Onn Malaysia/ Oricc

IV. RESULTS AND DISCUSSION

This Section discusses the final result of the M-Aho-Corasick using M_Triage tool for performing efficient file search. A dataset from DFRWS 2010 and another four dataset are used for the experiment. The result is discussed in this section. The test of the experiment is performed in the context of mobile forensics. The examination is conducted on a set of real objects smartphone and future mobile phone, with functional and operational characteristics also different from each other.

Fig 7 and 8 present the time taken by all the recovery tools used during the experiment. Each tool is run ten different times using each data set. For each runs a time, taken value is obtained by each tool.



		Phone A	Phone B	Phone C	Phone D	Phone E	
	Size (MB)	2MB	66MB	98.8	124	400	Average Time
M	M-Triage	0.21	1.10	4.20	6.57	13.40	5.10
D	Dec0de	0.58	4.12	12.10	24.30	60.00	20.22
L	Lifter	0.30	2.20	6.00	10.50	21.00	8.00
X	Xry	0.29	2.17	5.30	8.20	19.35	7.06
Xa	Xarver	1.20	5.23	15.40	25.47	41.30	17.72
D-M		0.37	3.02	7.90	17.73	46.60	15.12
L-M		0.09	1.10	1.80	3.93	7.60	2.90
X-M		0.08	1.07	1.10	1.63	5.95	1.97
Xa-M		0.99	4.13	11.20	18.90	27.90	12.62
Improvement in Percentage	Dec0de	64	73	65	73	78	75
	Lifter	30	50	30	37	36	36
	Xry	28	49	21	20	31	28
	Xarver	83	79	73	74	68	71

Fig. 7. Computational time comparison for all tools and all data set

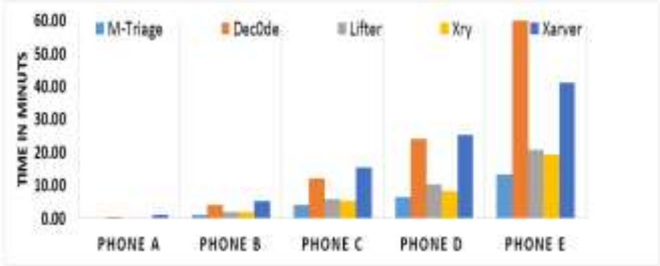


Fig. 8. The summary of computational time comparison between various forensic tools in graph

The result reported the average values of these ten runs, where Pim stands for percentage improvement and ET time stands for existing tools, also MT time stands for M_Triage time.

$$P_{im}MT(Existing\ Tool) = \frac{ET\ Time-MT\ Time}{ET\ Time} * 100 \qquad (1)$$

Furthermore, to calculate the percentage improvement, equation (1) is used as ETtime-MTtime divide by ETtime multiply by 100. For instance, in Fig 7 PhoneA has the size of 2.26MB, and M_Triage complete the execution time in 0.21 seconds while Dec0de completed in 0.58, to calculate the percentage improvement 0.58 minus 0.21 is equal to 0.37 then divide by 0.58 and multiply by 100, the result shows that in PhoneA size 2.26MB, 64% is achieved over Dec0de by M_Triage. The calculation steps apply to all data set used in the experiment.

V. CONCLUSIONS

This paper addresses the issue of searching a file during Digital forensic examination. A novel, multi-pattern signature matching algorithm called M-Aho-Corasick is developed in M_Triage to address such problem. Experiments are conducted on five different datasets which one of the data sets is obtained from DFRWS 2010. Comparisons are made between M-Aho-Corasick using M_Triage with Dec0de, Lifter, XRY, and Xaver. The result shows that M-Aho-Corasick using M_Triage has reduced the searching time by 75% as compared to Dec0de, 36% as compared to Lifter, 28% as compared to XRY, and 71% as compared to Xaver (refer to Fig. 7). Thus, this shows that M-Aho-Corasick using M_Triage is much more stable for searching a file during the forensic examination.

ACKNOWLEDGMENT

The authors would like to acknowledge Universiti Tun Hussein Onn Malaysia (UTHM), for providing financial support (Vot U061) towards this research.

REFERENCES

[1] K. Curran, A. Robinson, S. Peacocke, and S. Cassidy, "Mobile Phone Forensic Analysis," vol. 2, no. 2, 2010.

[2] N. A. Abdullah, R. Ibrahim, and K. M. Mohamad, "An IMPROVE file carver of intertwined jpeg images using X-mykarve," UNIVERSITY TUN HUSSEIN ONN MALAYSIA, 2014.

[3] S. Arudchutha, T. Nishanthy, and R. G. Ragel, "String matching with multicore CPUs: Performing better with the Aho-Corasick algorithm," 2013 IEEE 8th Int. Conf. Ind. Inf. Syst. ICIIS 2013 - Conf. Proc., pp. 231–236, 2013.

[4] L. Benuskova, "Lecture 4 : Exact string searching algorithms," http://marknelson.us/1996/08/01/suffix-trees/, vol. 2, no. 1, p. 6, 2012.

[5] R. J. Walls, E. Learned-miller, and B. N. Levine, "Forensic Triage for Mobile Phones with DEC0DE," 2011.

[6] R. J. Walls and B. N. Levine, "Efficient Smart Phone Forensics Based on Relevance Feedback," 2014.

[7] S. L. Garfinkel, "Digital media triage with bulk data analysis and bulk-extractor," Comput. Secur., vol. 32, pp. 56–72, 2013.

[8] J. Blokhuis and A. Puppe, "DFRWS Challenge 2010 - Mobile forensics," 2010.