Detecting timestamp forgery in NTFS file system using logfile

[1]Pratik Patel, [2]Shailendra Mishra

[1]Student, [2]Assistant Professor
[1]Computer Science Department,
[1]Parul Institute of Technology, Waghodia, India

***Abstract*— In the current era of digital world, user and investigator are more dependent on digital data. Digital data are very vast in size and also stored in various formats. So, the major problem is identification of upcoming data as true or false by the user or investigator. To overcome this problem different methods and techniques are adapted. Forensic method is used for validation of data. A computer forensic method can be used for detecting the different types of forgeries and computer crime. Forgeries and computer crime are the most major concern of the digital world. Lots of techniques and methods have been used to find a proper solution to overcome these forgery problems. Occurrences of digital crimes or forgeries are investigated using a method or technique called forensics. Initially a general survey was carried out to understand the different methods used in computer forensics to track the evidences which could be useful for detecting the computer crime and forgery. Forensic tools can be used for making any changes to data or tampering of data. Different rule sets or methods are defined to detect the various errors regarding the changes and the tampering of the data in different windows file systems. The data is tampered or modified in either of the two ways i.e., offline or online. In this research, offline data is of upmost concern. Digital evidence which stores information in digital form can be used to detect forgery and computer crime. In this paper, a computer forensic method for detecting timestamp forgery in the Windows NTFS file system is presented. The accuracy of timestamp forgery can be further improved by using attributes of files like size, time. The tool can be used for all types of files.**

***Index Terms*— Computer forensics, Digital forensics, Evidence, Forensic tools, NTFS file system, $Log file.**

I. Introduction

In today’s world crime using digital data has been increasing exponentially. Digital data in current era plays an important role to save the data electronically so that records of such data can be easily maintained. But un-trusted third party with wrong intensions misuses this advantage of digital data to do crime. In digital data, un-trusted third party changes data manually or timestamp forgery of data in different file system. Hence in current era, providing a security to such digital data is necessary. Data can be easily secured against the illegal activities perform by un-trusted third party. Detection of any type of data tampering, theft or forgery related to computer devices, forensic investigation can be taken. Forensic methods or rules can be helpful for detecting such misuse or crime happening due to misuse of digital data. Any manual intervention intentionally or unintentionally can be detected using forensics rules or tools. With the help of proper evidences, forensic rules and forensic tools, investigator can easily investigate for the timestamp forgery. Forensic tools used might be freely available or can be purchased online. Forensics tools are different for different cases or crimes.

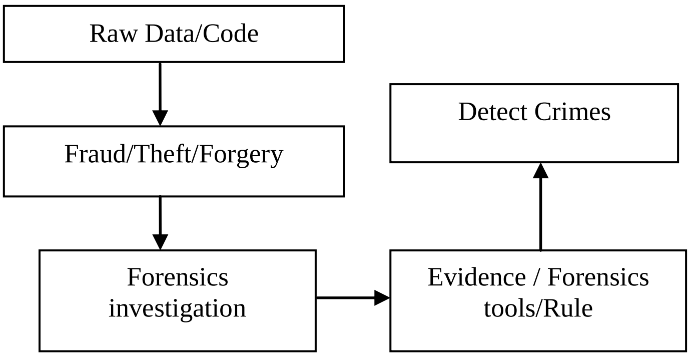


Fig.1: Basic Process of Forensic to Detect Crime

Fig.1 describes the basic process of forensic for detecting forgery and crimes. Raw data, code or programs are taken as input. Any user can made changes in data, theft of code or any forgery relates to computer devices. So to detect crimes and forgery, forensic method is used. For this the investigator needs evidence, forensics tools and rules which relate to cases. And finally using trustworthy evidence, forensics tools and rules, detection of crimes and forgery is possible.

The paper is organized as follows: Section II gives a prior and related work, Section III defines research problem and challenges, Section IV gives approaches of research work, Section V discussed Result and Analysis of work and Section VI conclude the paper.

II. PRIOR AND RELATED WORK

There has been many research work done in digital forensic world. For this Brian Carrier, suggested a technique based on digital forensics examination and analysis on tools using abstraction layers and identified where the error was generated and helped to determine the result [1]. Simson L. Garfinkel defines Current forensic research directions and suggests that to move ahead as per community needs. Digital forensics tools can be used on daily basis for examined and analyzed data [2]. Eoghan Casey et al, summarizes that during investigation, investigator needs reliable, complete and authentic evidence. So using that evidence level of forensics can be improved [3]. Guofu Ma et al, suggested that to detect mismatch phenomenon and computer related crimes, investigator needs evidences. To know the objective of evidence, authors define evidence ring and evidence chain model builds and also supervisory chain in supervision can be done [4]. Liang Hu et al, suggested how model used for the research results of volatile data analysis and also prevent crimes by further analysis of the criminal behavior database [5]. Chris Boyd, Pete et al, describe more common forensic issues when interpreting dates and times during internet. Using example, author demonstrates how time and date issues can be done in forensics computing [6]. K.P Chow et al, discussed MAC time analysis and also do experiment on some operations like copy, move, download etc on NTFS File System [7]. Willassen discussed about how different operations affect timestamp in the system and also tested hypothesis about the historical values of clock using available evidence [8]. Gyu-Sang Cho suggested a forensics method to detect timestamp forgery in NTFS file system. Using evidence, forensic rules and tools timestamp forgery can be detected [9].

In this paper, we propose timestamp forgery detection more accurate to the adding of size and time attribute in NTFS File system. We worked with different file system like Notepad, MS Word, Adobe Acrobat, MS Excel, and Power Point Presentation etc. Also uses forensics tools for date/time change, extracting NTFS file system, viewed logfile record and Master File Table (MFT) records. The comparison with previous works is shown in tabular form.

III. RESEARCH PROBLEM AND CHALLENGES

In modern era, providing a security of data is most important part. Digital data are increasing day by day. Any user or un-trusted third party changes manually in the text or any images, date/time value of files then the result which is required by user is false. So in real time applications like auction, tendering, election, job letter, file documentation etc problems are increasing because the investigator cannot understand the proper behavior of the users.

So, securing and detecting the digital data are challenges for investigator. Using forensic tools, rules and evidences, problem of securing and detecting digital data can be solved.

IV. APPROACH OF RESEARCH WORK

It has become an easy task for un-trusted third party to play with the timestamp of files using tools. So below described model can be used to detect such type of forgeries.

Input: Different extensions bunch of files
Output: Calculation of size and time attribute.

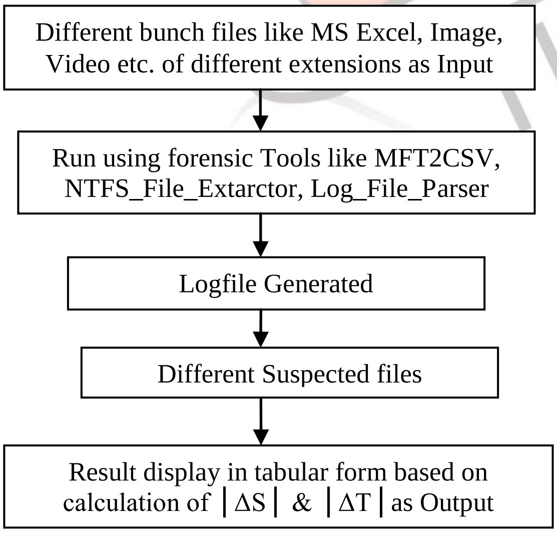

Different bunch files like MS Excel, Image, Video etc. of different extensions as Input
Run using forensic Tools like MFT2CSV, NTFS_File_Extarctor, Log_File_Parser
Logfile Generated
Different Suspected files
Result display in tabular form based on calculation of │ΔS│ & │ΔT│ as Output

Fig.2: General Model to Detect Timestamp Forgery

In Fig. 2, a proposed model is shown. Here different bunch of files extension like Notepad, MS Excel, Adobe Acrobat, MS Word, Image, Video etc to the tool. The input files are then processed using the forensic tools like NTFS_File_Extractor, MFT2CSV, Log_File_Parser. Logfile will be generated once the input files are processed using above tools and from that suspected files will be filtered. The results will be displayed in tabular form based on the calculation of │ΔS│(Difference between current and previous size of the file) and │ΔT│(Difference between FN and SI time of the file).

V. RESULT AND ANALYSIS

The following given results have been noted keeping in mind the following constraints:

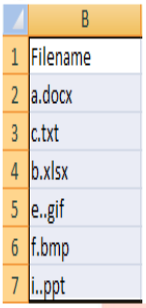
- Tool must not be corrupted
- Source file must be extracted properly without errors
- System must be ON
- Data must not be corrupted due to any viruses or bug

The performance of proposed model may be measured from the Table1. Table1 shows bunch of different extensions files used to detect timestamp forgery. Suspected files can be filtered from that.

Table1: Bunch of Different Extensions Files

a.docx	b.xlsx	c.txt
d.png	e.gif	f.bmp
g.txt	h.pptx	i..ppt
j.csv	Shane warne magic ball.flv	-

List of suspected files are filtered from Table1 which is shown in Fig.3.



	B
1	Filename
2	a.docx
3	c.txt
4	b.xlsx
5	e..gif
6	f.bmp
7	i..ppt

Fig.3: Different Suspected Files

Fig.3.1 and Fig.3.2 shows extraction of logfile for calculation of size and time attribute respectively. After that Fig.4 and Fig.5 display the results of suspected file in terms of calculation of size and time attributes respectively.

Here suspected files record can extract from log records and investigator extract only file name, redo, undo, size, and time attributes. And finally shows a timestamp forgery and also shows how much data has been tampered in terms of size

VI. Conclusion And future scope

In this paper, we proposed a technique for detecting timestamp forgery in different file extensions like .txt, .doc, .ppt, .xls, .png etc. The results and analysis proves that our proposed technique is far better and efficient than the techniques and rules described in the literature. The future prospect of our technique is, it can be used with other different file attributes to exactly detect where the changes have been made in the file. Above discussed technique can be also used for different file system. New forensic rules and tools can be developed to detect timestamp forgery in more efficient way.

References

[1] Brian Carrier, "Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers" International Journal of Digital Evidence (2003) pp.1-12.

[2] Simson L. Garfinkel, "Digital forensics research: The next10 years", Digital Investigation 7(2010) S64-s73

[3] Eoghan Casey, Error, Uncertainity, and loss in Digital Evidence. International Journal of Digital Evidence, 1(2), summer 2002.

[4] Guofu Ma, Zixian Wang, Likun Zou, Qian Zhanga,"Computer Forensics Model Based on Evidence Ring and Evidence Chain", Procedia Engineering 15 (2011) 3663 – 3667

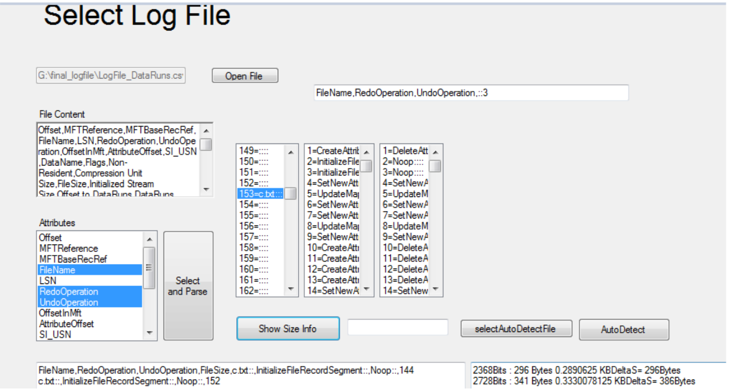
[5] Liang Hu, XiaoLu Zhang, Feng Wang, WenBo Wang, Kuo Zhao,"Research on the Architecture Model of Volatile Data Forensics", Procedia Engineering 29 (2012) 4254 – 4258

[6] Boyd C, Forster P. Time and date issues in forensic computing a case study. Digital Investigation Feb.2004; 1(1):18-23.

[7] Chow KP, Law FYW, Kwan MYK, Lai KY. "The rules of time on NTFS file system", SADFE '07. pp. 71e85; March 2007.

[8] Willasen S. "Timestamp evidence correlation by model based clock hypothesis testing" Adelaide, Australia: E-Forensics; 2008.

[9] Gyu-Sang Cho, "A computer forensic method for detecting timestamp forgery in NTFS", computer & security 34(2013) 36-46



G:\final_logfile\LogFile_DataRuns.cs

FileName,RedoOperation,UndoOperation,::3

Offset,MFTReference,MFTBaseRecRef, FileName,LSN,RedoOperation,UndoOpe ration,OffsetInMft,AttributeOffset,SI_USN ,DataName,Flags,Non-Resident,Compression Unit Size,FileSize,Initialized Stream Size,Offset to DataRuns,DataRuns

Offset
MFTReference
MFTBaseRecRef
FileName
LSN
RedoOperation
UndoOperation
OffsetInMft
AttributeOffset
SI_USN

149=::::
150=::::
151=::::
152=::::
153=c.txt::::
154=::::
155=::::
156=::::
157=::::
158=::::
159=::::
160=::::
161=::::
162=::::

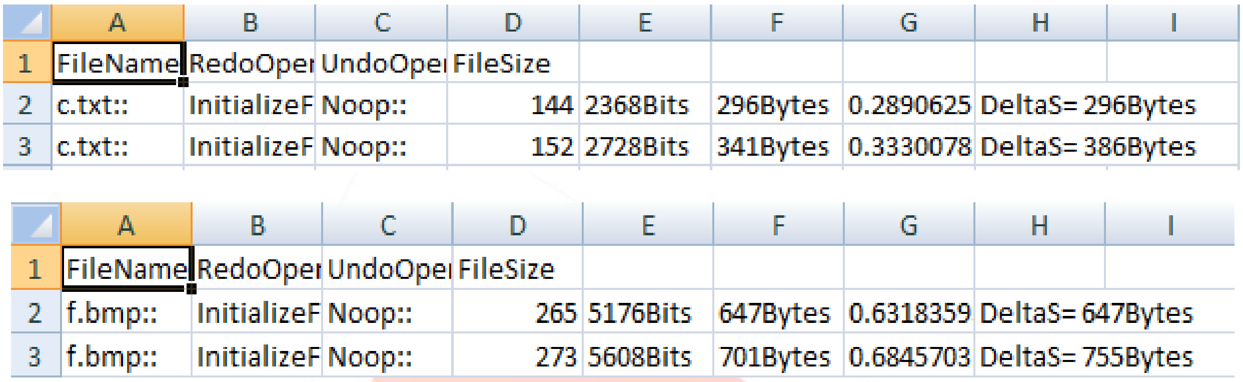
1=CreateAttrib
2=InitializeFile
3=InitializeFile
4=SetNewAtt
5=UpdateMap
6=SetNewAtt
7=SetNewAtt
8=UpdateMap
9=SetNewAtt
10=CreateAttr
11=CreateAttr
12=CreateAttr
13=CreateAttr
14=SetNewA

1=DeleteAtt
2=Noop::::
3=Noop::::
4=SetNewA
5=UpdateM
6=SetNewA
7=SetNewA
8=UpdateM
9=SetNewA
10=DeleteA
11=DeleteA
12=DeleteA
13=DeleteA
14=SetNew

```
FileName,RedoOperation,UndoOperation,FileSize,c.txt::,InitializeFileRecordSegment::,Noop::,144
c.txt::,InitializeFileRecordSegment::,Noop::,152
```

```
2368Bits : 296 Bytes 0.2890625 KBDeltaS= 296Bytes
2728Bits : 341 Bytes 0.3330078125 KBDeltaS= 386Bytes
```

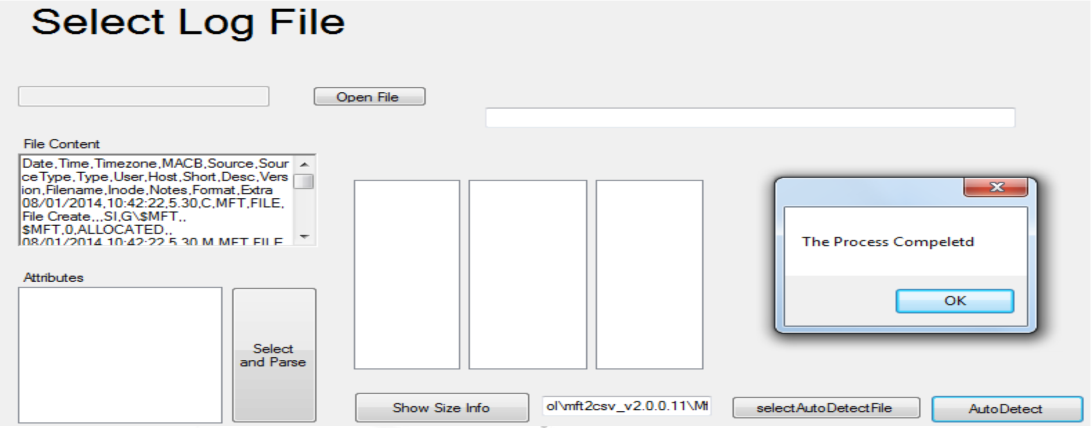
Fig.3.1: Extract Logfile for Calculate Size Attribute



	A	B	C	D	E	F	G	H	I
1	FileName	RedoOper	UndoOpe	FileSize					
2	c.txt::	InitializeF	Noop::	144	2368Bits	296Bytes	0.2890625	DeltaS= 296Bytes	
3	c.txt::	InitializeF	Noop::	152	2728Bits	341Bytes	0.3330078	DeltaS= 386Bytes	

	A	B	C	D	E	F	G	H	I
1	FileName	RedoOper	UndoOpe	FileSize					
2	f.bmp::	InitializeF	Noop::	265	5176Bits	647Bytes	0.6318359	DeltaS= 647Bytes	
3	f.bmp::	InitializeF	Noop::	273	5608Bits	701Bytes	0.6845703	DeltaS= 755Bytes	

Fig.4: Size Calculation of Suspected Files

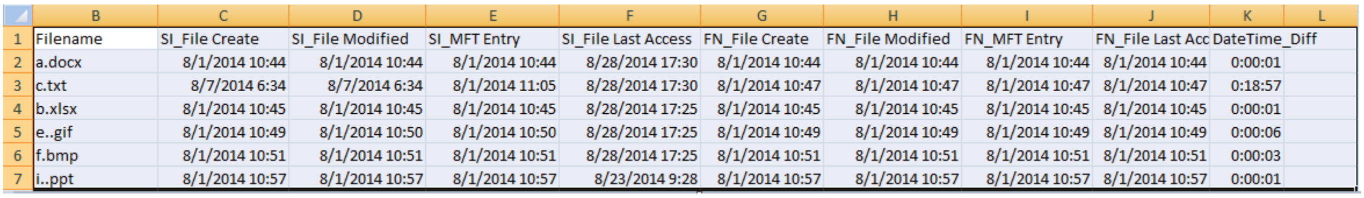


Date,Time,Timezone,MACB,Source,SourceType,Type,User,Host,Short,Desc,Version,Filename,Inode,Notes,Format,Extra
08/01/2014,10:42:22.5,30,C,MFT,FILE, File Create,,SI,G:\$MFT,,
$MFT,0,ALLOCATED,,
08/01/2014,10:42:22.5,30,M,MFT,FILE

The Process Compeletd

ol\mft2csv_v2.0.0.11\Mt

Fig.3.2: Extract Logfile for Calculate Time Attribute



	B	C	D	E	F	G	H	I	J	K	L
1	Filename	SI_File Create	SI_File Modified	SI_MFT Entry	SI_File Last Access	FN_File Create	FN_File Modified	FN_MFT Entry	FN_File Last Acc	DateTime_Diff	
2	a.docx	8/1/2014 10:44	8/1/2014 10:44	8/1/2014 10:44	8/28/2014 17:30	8/1/2014 10:44	8/1/2014 10:44	8/1/2014 10:44	8/1/2014 10:44	0:00:01	
3	c.txt	8/7/2014 6:34	8/7/2014 6:34	8/1/2014 11:05	8/28/2014 17:30	8/1/2014 10:47	8/1/2014 10:47	8/1/2014 10:47	8/1/2014 10:47	0:18:57	
4	b.xlsx	8/1/2014 10:45	8/1/2014 10:45	8/1/2014 10:45	8/28/2014 17:25	8/1/2014 10:45	8/1/2014 10:45	8/1/2014 10:45	8/1/2014 10:45	0:00:01	
5	e..gif	8/1/2014 10:49	8/1/2014 10:50	8/1/2014 10:50	8/28/2014 17:25	8/1/2014 10:49	8/1/2014 10:49	8/1/2014 10:49	8/1/2014 10:49	0:00:06	
6	f.bmp	8/1/2014 10:51	8/1/2014 10:51	8/1/2014 10:51	8/28/2014 17:25	8/1/2014 10:51	8/1/2014 10:51	8/1/2014 10:51	8/1/2014 10:51	0:00:03	
7	i..ppt	8/1/2014 10:57	8/1/2014 10:57	8/1/2014 10:57	8/23/2014 9:28	8/1/2014 10:57	8/1/2014 10:57	8/1/2014 10:57	8/1/2014 10:57	0:00:01	

Fig.5: Time Calculation of Suspected Files