EasyChair Preprint

№ 1203

Computer Aided Diagnostics of Digital Evidence Tampering (CADDET)

Babak Habibnia and Pavel Gladyshev

June 16, 2019

Computer Aided Diagnostics of Digital Evidence Tampering (CADDET)

Babak Habibnia, Pavel Gladyshev
Digital Forensics Investigation Research Laboratory (DFire), University College Dublin (UCD), Dublin, Ireland
Babak.Habibnia@ucd.ie
Pavel.Gladyshev@ucd.ie

Abstract: The tampering of the digital crime scene has become more common. When tampering behaviour is successful, it does not leave a trace of either the incriminating evidence or the act of tampering and the digital evidence that digital investigators seek will be absent. The research into the automatic detection of digital evidence tampering has been ongoing for over one decade. Many approaches had been proposed, but the practical tools for automatic detection of evidence tampering are still missing. Automatic analysis is hard due to the complexity of real-world computers and differences between software installed on different computers. A similar problem exists in medical imaging. Despite the common grand design, every human is unique and complex, and it is hard to come up with the exact rules for detecting lesions in medical images. Visualization for forensic analysis of the data stored on a specific device has received much less attention, while the use of visualization for detection of digital evidence tampering is virtually unexplored.

This paper proposes, *for the first time*, a semi-automated approach based on visualization of relevant data properties, helping human investigators to detect digital evidence tampering and anomaly. This is analogous to computer aided processing of medical X-Ray images that enhance the visibility of lesions facilitating easier detection by a doctor. The aim of this paper is to identify data tampered features on the digital devices, then find suitable visualization to display identified data tampered features for investigators. One of the outstanding features of the approach proposed in this paper for detecting digital evidence tampering is its malleability. It can easily apply to any specific or whole part of data in the digital devices, visualize, and reveal offender concealment behaviour in relation to detection of evidence tampering.

CADDET sheds new light and explores new insight to contribute investigators to fill a gap of detecting digital evidence tampering. This semi-automated approach has never been studied before and is novel in the context of digital forensics.

Keywords: Cybercrime, Cybersecurity, Digital Evidence Tampering, Digital Forensics, Anti-Forensics, Visualization.

1. Introduction:

Any action by a user on a computer, whether it's surfing the internet, communication or file storage, affects data stored in the computer. Analysis of computer data can often help to determine when, where, and how a crime has been committed. Digital forensics is a branch of forensic science dealing with inspection, extraction, and analysis of computer data as evidence in litigation.

Currently, when the digital investigator is faced with any type of digital evidence tampering behaviour, they must look for present forensic tools or methods in an ad hoc manner, for instance, ask peers, search tool sources to look for similarity evidence tampering, review publications. Even if a given problem has been found, there is currently a delay in disseminating a new tool or publishing a new method.

Most computer forensics tools offer capabilities such as imaging, analysis, viewing, and reporting. They are unable to present a visual overview of all data found on a piece of media especially when evidence tampering occurred.

The aim of this paper is to detect digital evidence tampering and it has been divided into the two main parts. The first part deals with *identifying data tampered features* on the digital devices, focusing on the six key tasks that, identifying anti-forensics tool features, used dataset, evidence tampering action, comparison, result, and exploring automated method. The second part deals with the *finding suitable visualization* to visualize identified data tampered features for investigators, focusing on the two key tasks that, designing visualization-*parallel Coordinates Visualization*, and result. For instance, visualize PC1, PC2, PC3, PC4, and PC1 (tampered) behaviour in a normal way of usage, base on the identified data tampered features.

Due to the limited number of pages (max. 10 pages), the author's decided at the outset to point out the main finding in this paper.

2. Methodology

This section will cover the unique methodologies have explored for detecting digital evidence tampering. It is divided into two main sub-sections, each of which presents the methodology was taken and the result.

2.1 Identifying Data Tampered Features (file + location)

In order to identify data tampered features, it will be necessary to identify and carry out the following steps.

2.1.1 Identifying Anti-Forensics (AF) Tool Features

The purpose of this part, identify anti-forensics tool features, and describe them where and what the impact on the data after is using. Any action by a user on a computer (or digital devices), affects data stored in the computer. Different AF tools have a varying degree of effectiveness. In positive aspect, AF tools are used to sanitize user activity and conversely can be led to the intentional tampering (deliberate manipulation). By reviewing three common and free available AF tools; each of the AF tools has unique GUI (graphic user interface) and particular features. Table 1 shows a comprehensive overview of AF tool's features base in Windows Operating System.

Table 1: Anti-Forensics Tool's features

Identifying Anti-Forensics Features	
Anti-Forensics Tools	**Features**
Tracks Erase Pro (Acesoft, 2001)	Erase history data, cache, cookies, history, typed URLs, autocomplete memory, index.dat, Window's temp folder, run & search history, open/save history and recent documents.
CCleaner (Dimmick, 2005)	Removes unused files, cleans traces online activities, internet history, registry cleaner, recycle bin, recent documents, temporary files, log files, clipboard, DNS Cache, error reporting, memory dumps, jump lists, recent documents, temporary files, log files, clipboard, history, cookies, super cookies, autocomplete form history, index.dat files, etc.
Windows Eraser (c|net, n.d.)	Erase online and offline traces (only Internet Explorer), cookies, start menu Run/Find History, auto-complete memory, passwords, temporary, unwanted traces, ACDSee, Adobe, MSN and Yahoo messengers, and Microsoft Office.

2.1.2 Used Dataset

In this study authors used 4 virtual machine disk images from the DFire lab (Gladyshev, 2013) to simulate an original intellectual property theft that occurred in a company in 2015 for the experiment. The disk images included the following specification:

- Windows 7
- Over 93,012 (test case before tampering purpose) and 94,111 (test case after tampering) records in 35 columns.
- Installation of two additional commonly-used browsers: Chrome, Firefox
- Using all three browsers Internet Explorer, Chrome, Firefox
- Running several programs

2.1.3 Evidence Tampering Action

This step was carried out using the CCleaner software (Dimmick, 2005) which is available in both free and paid format, to create a tempered image for comparison (next step) in the designed test environment? According to the **Error! Reference source not found.**, the CCleaner cover all features and more than it was expected for testing in the sample case (dataset). The standard dataset base of the Win 7 which provided by DFire lab (Gladyshev, 2013) was set up for tampering on the test environment. The CCleaner was prepared for tampering purpose, conducted till to complete its deletion and wiping. Then the image was taken using FTK imager tool from tampered data and examined with X-Ways (X-Ways, 2002) suite of computer forensics tool. At first, by looking and analysing tampered data using the X–Way, it was no clue how and where exactly CCleaner affected into the data. After that, with considering AF tool's features (Table 1) and review data in-depth (folders, subfolders, and files), it didn't help to distinguish the effects of tampering (or manipulating). It only showed the number of folders and in front of each folder, numbers are included as bracketed points.

2.1.4 Comparison

Following up step 2.1.3, in this step the image was taken using FTK imager (AccessData, 2010) tool from the *Karsean* case without conducting CCleaner tool for tampering purpose (non-tampered). Using similar method and tool for analysing data it showed interesting results in the numbers are included as bracketed points. That

encouraged to carry out a comparison and see a difference between Tampered and Non-Tampered Data. At first glance, a comparison of two results reveals a big difference between the number of the files in Tampered and Non-tampered data as indicated in Figure 1.

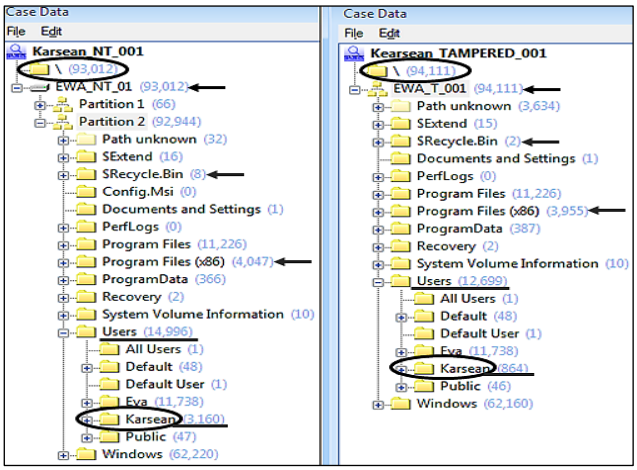

Figure 1: Comparison between a number of the files non-tampered and tampered cases (Karsean_NT and Kearsean_ Tampered cases).

According to AF tool's features (Table 1), their performance, and their impact on the Windows OS. Looking any individual folder, subfolder, and file. For instance, **Error! Reference source not found.**, *Recycle.bin* (deleted files) (Wikipedia, 2016) shows clearly the difference between the number of the presence and absence files before and after tampering. Another very interesting item could be username folder, and subfolders, which contains very important files such as web browser application, cache, history, cookies and etc. in tampering purpose.

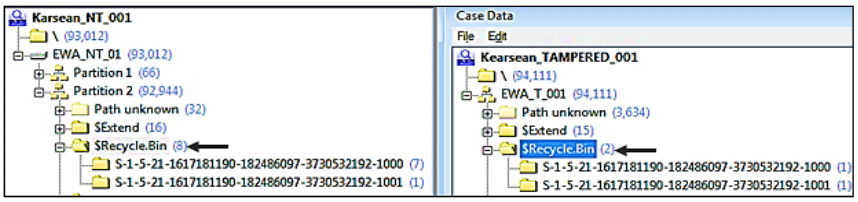

Figure 2: Comparison between a number of the files in a non-tampered and tampered data in Recycle Bin, 8(NT) --> 2 (T) (Karsean_NT and Kearsean_ Tampered cases).

2.1.5 Result (identifying data tampered features)

As was explained in the previous steps, Using the X-Way forensics tool for comparing the difference between the number of the files into the specific folders (e.g. Recycle Bin, Users and etc.). In this step, for a better understanding of comparison and further analysis, dataset had been separately generated and exported in 93,012 records and 35 columns (non-tampered) and 94,111 records and 35 columns (tampered) in the CSV file format. Figure 3 briefly is shown as a sample the title of the each column and the content of the each row.

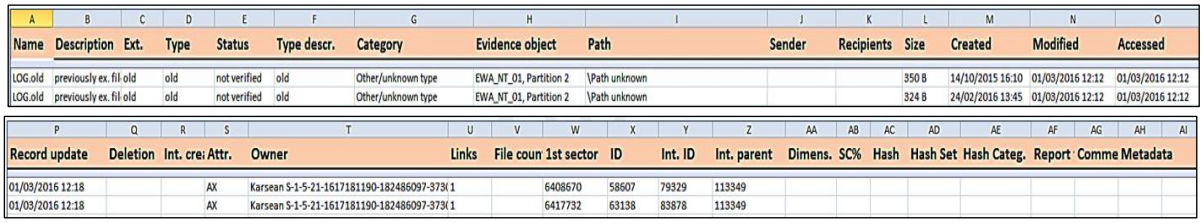

Figure 3: View of CSV Generated and Exported (similar ti that, Tampered file: Contain 93,012 records and 35 column fields \& Non-Tampered: Contain 43,111 records and 35 column fields)

Then by searching and comparing manually into the each row and column to find the difference between non-tampered and tampered data. For instance, the results obtained from the comparison of the *Recycle.bin* in two

exported files, as shown in Figure 4 and Figure 5, indicate clearly the difference between the files and also identifying filename and the location of the tampered file into the dataset before and after tampering action.

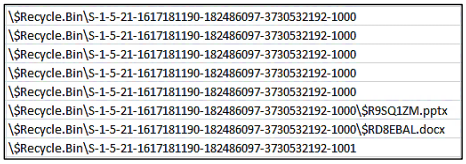

Figure 4: Recycle Bin non-tampered dataset (filename + location) in CSV

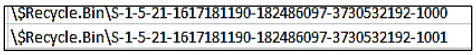

Figure 5: Recycle Bin tampered dataset (filename + location) in CSV

In the broader aspects of detecting digital evidence tampering in entire dataset, Table 2, presents the remarkable result of identifying data tampered features (file + location) which it solved the first problem as mentioned in this research paper earlier. This finding never identified before the digital forensics field.

Table 2: Identifying Data Tampered Features for Detecting Digital Evidence Tampering , which was never explored before.

Identifying Data Tampered Features (file + location)		
Data Tampered		**Location of Data Tampered in Microsoft Window XP/7/8/10**
Recycle.bin		▪ Windows XP (C:\RECYCLER" 2000/NT/XP/2003) ▪ Win7/8/10 (C:\Recycle.bin)
History	Internet Explorer	▪ IE6-7 (%USERPROFILE%\Local Settings\History\History.IE5) ▪ *IE8-9 (%USERPROFILE%\AppData\Local\Microsoft\Windows\History\History.IE5)* ▪ IE10-11(%UERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV*.dat)
	Firefox	▪ XP (%USERPROFILE%\Application Data\Roming\Mozilla\Firefox\Profiles\<random text>.default\places.sqlite) ▪ Win7/8/10 (%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<random text>.default\places.sqlite)
	Chrome	▪ XP (%USERPROFILE%\Local Settings\Application Data\Google\Chrome\User Data\Default\History) ▪ *Win7/8/10 (%USERPROFILE%\AppData\Local\Google\Chrome\UserData\Default\)*
Cache	Internet Explorer	▪ IE8-9 (%USERPROFILE%\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5) ▪ IE10 (%USERPROFILE%\AppData\Local\Microsoft\Windows\Temporary Internet Files\ Content.IE5) ▪ IE11 %USERPROFILE%\AppData\Local\Microsoft\Windows\INetCache\IE)
	Firefox	▪ XP (%USERPROFILE%\Local Settings\ApplicationData\Mozilla\Firefox\Profiles\<randomtext>.default\Cache) ▪ Win7/8/10 (%USERPROFILE%\AppData\Local\Mozilla\Firefox\Profiles\<randomtext>.default\Cache)
	Chrome	▪ XP (%USERPROFILE%\Local Settings\Application Data\Google\Chrome\User Data\Default\Cache - data_# and f_######) ▪ Win7/8/10 (%USERPROFILE%\AppData\Local\Google\Chrome\UserData\Default\ Cache\ - data_# and f_######)
Chrome	Internet Explorer	▪ IE8-9 %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies ▪ IE10 %USERPROFILE%\AppData\Roaming\Microsoft\Windows\Cookies ▪ IE11 %USERPROFILE%\AppData\Local\Microsoft\Windows\INetCookies
	Firefox	▪ XP %USERPROFILE%\Application Data\Mozilla\Firefox\Profiles\<random text>.default\cookies.sqlite ▪ Win7/8/10 %USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\cookies.sqlite

	Chrome	▪ XP %USERPROFILE%\Local Settings\Application Data\Google\Chrome\User Data\Default\Local Storage\ ▪ Win7/8/10 %USERPROFILE%\AppData\Local\Google\Chrome\UserData\Default\ Local Storage\
Identifying Data Tampered Features (file + location)		
Data Tampered		**Location of Data Tampered in Microsoft Window XP/7/8/10**
Session Restore	Internet Explorer	▪ Win7/8/10 %USERPROFILE%/AppData/Local/Microsoft/Internet Explorer/Recovery
	Firefox	▪ Win7/8/10 %USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<randomtext>.default\sessionstore.js
	Chrome	▪ Win7/8/10 %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\ Files = Current Session & Current Tabs & Last Session & Last Tabs
Prefetch		▪ XP (C:\%USERPROFILE%\Recent) ▪ *Win7/8/10 (C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\ & C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations & C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations)*
Temporary		▪ %USERPROFILE%\AppData\Local\Temp
Shortcut (LNK)		▪ XP (C:\%USERPROFILE%\Recent) ▪ Win7/8/10 (▪ C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\ & C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations & C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations)
Recent Files		▪ XP (C:\%USERPROFILE%\Recent) ▪ Win7/8/10 (C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\ & C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations & C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations)
Jump List		▪ Win7/8/10 (C:\%USERPROFILE%\AppData\Roaming\ Microsoft\Windows\Recent\AutomaticDestinations\ ID numbers.automaticDestinations-ms) ▪ Win7/8/10 (C:\%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations\ ID numbers.customDestinations-ms)
RDP Usage		▪ XP (%SYSTEM ROOT%\System32\config\SecEvent.evt) ▪ Win7/8/10 (c:\Windows\System32\winevt\logs\Security.evtx)
Event Log		▪ XP (C:\Windows\System32\winevt\Logs) ▪ Win7/8/10 (C:\Windows\System32\winevt\Logs)

2.1.6 Exploring Automated Method for Detecting Digital Evidence Tampering, Using Prolog Styla

Following on from the previous 2.1.5step**Error! Reference source not found.**, which outlined significant result of identifying data tampered features (file + location). The purpose of this step to create a reliable and flexible automated tool to display, count, and save the result of data tampered from a provided disk image in relation to detecting digital evidence tampering, then export the result for Visualization. According to Table 2, it is almost difficult and impossible to identify data tampered features (file + location) manually for the purpose of detecting data tampering; especially investigators don't have any clue of data. For this purpose, use *Forensics Styla environment-Prolog in Scala with forensic extensions* (Gladyshev, 2016). The styla is a lightweight implementation of a Prolog (Wikipedia, 1972) in Scala developed by Paul Tarau (Tarau, 2014).

In order to examine data with Forensic Styla, it will be necessary either create a new case or open an existing case file (for instance: *Karsean* as mentioned above). The following predicates are defined to do that: new case, open case, close case, deli case. In addition gives accessing/exploring the files into the image (test case) through of the command line (terminal) when using a specific command.

The new case was created in Forensics Style and added the image (non-tempered or tampered) file was taken. Then, created a tool (or coding) to display, count, and save the result base on the investigator interest (partial or the entire dataset), and exported as a file for visualization (see Figure 6). For instance, the authors were interested only in Recycle Bin (RB), Internet History (IH), Cache (CSH), Cookies (COO), Restore Point (RP) and Prefetch (PR) features from Table 2 for the experiment (count number of the files with considering their location) and exporting as shown in Figure 7 for the visualization.

The significant reason for choosing Forensics Styla is that having both forensics tool and writing logic code interface on one front page instead of using two separate tools. It also uses a terminal for typing any query for further analysis.

so far, this research paper focussed on identifying data tampered features (file+location) for detecting digital evidence tampering, which it has achieved remarkable results and never been done in the computer forensics field (Table 1).

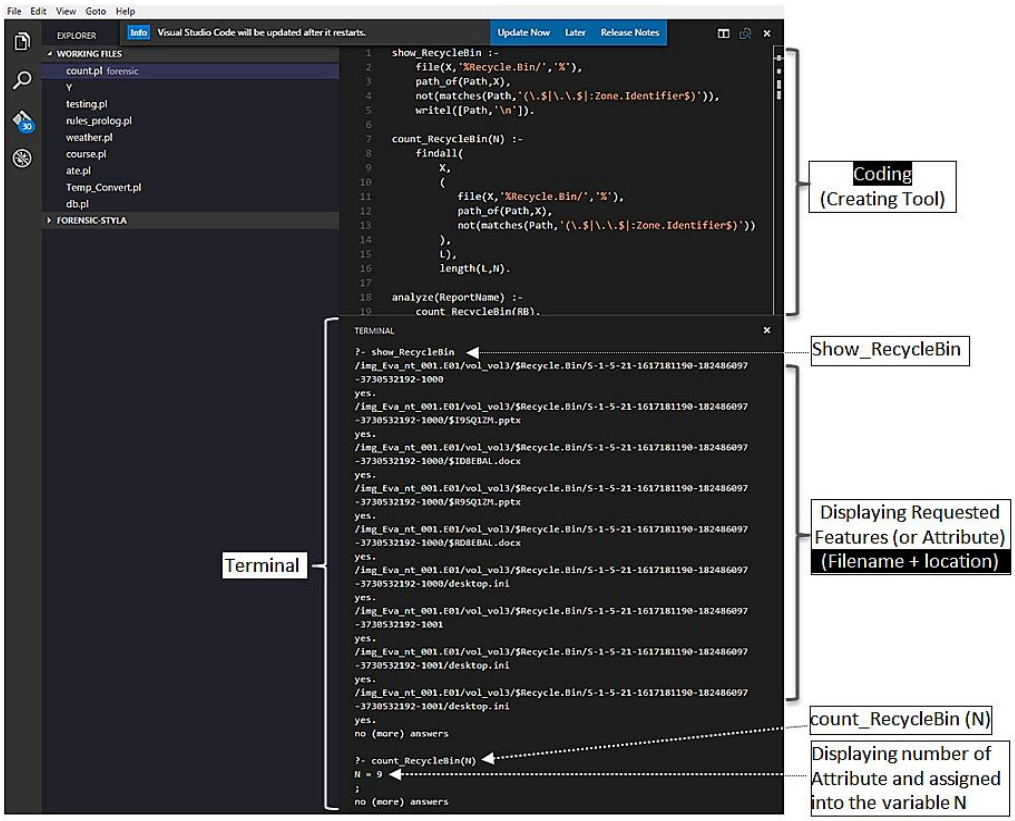

Figure 6: Forensics Styla Environment

A	B	C	D	E	F	G
EVIDENCE	RB	HI	CSH	COO	RP	PR
PC1	8	123	500	432	55	122
PC1(Tamperd)	2	21	0	16	11	25
PC2	25	175	515	531	64	135
PC3	33	185	650	625	55	114
PC4	46	191	569	571	35	163

Figure 7: Exported Dataset among value in CSV format

The following part will discuss about visualization, what are the best visualization to present obtained data in such understanding and simple way for investigators? Which it is second and last part of the study aim (or problem) as was pointed earlier.

2.2 Finding Suitable visualization

Visualization allows for displaying an overview of all the data found on the piece of media. In (2006), Teerlink and Erbacher (Teerlink & Erbacher, 2006) wrote in one sentence, what summarizes the problem:

> '*A great deal of time is wasted by analysts trying to interpret massive amounts of data that isn't correlated or meaningful without high levels of patience and tolerance for error. Visualization techniques can greatly aid forensic specialists to direct their search to suspicious file*'.

It's easy to throw the data up on a bar chart or scatter plot in Excel, PowerPoint, slip it into a report, and convince that it does the explaining. But, that's a terrible shortcut. When the report is over and the only thing left behind is the report and no-one will have a clue what the chart was trying to communicate or say. There are so many chart types, styles, and methods of presenting data that can be confusing and hard to pick the *right chart type* for *obtained data in this paper*. The following step discusses designing suitable visualization.

2.2.1 Designing Visualization – Parallel Coordinates

As was pointed out in the introduction, the aim of this paper divided into two main parts: *Identifying data tampered features* and *finding suitable visualization* for detecting digital evidence tampering.
Having identified the data tampered features in the previous part with details, and maximum size of data is 21 as described in Table 2. Now, the authors will discuss the finding suitable visualization.

It was given a set of data points D = $\{pc_i\}$ where every point pc_i has an n-dimensional vector of attributes $(a_1^i, \dots, a_n^i) \in A^n$ defined on some domain A (e.g. see filename in Table 2). Such a dataset called *multivariate* with several attributes, or variables per data point.

In this paper, authors interested in examining the *distribution, correlation, and comparison* of the individual values of the various dimensions (visualize data tampered), and giving the overall distance between the data points. One technique that allows authors to perform such visualization is the *Parallel Coordinates* (Tableau, 2015) (Davies, 2016) (Anderson, 2016) (Inselberg, 1997). Parallel coordinates are one of the most common ways of visualizing and analyzing multivariate data (Shneiderman, 1996) and (Keim, 2002) which was *never used in forensic computing*.

To present in the easily understandable way how parallel coordinate work, let authors consider an example. In Figure 7, the dataset contains 5 data points and each data point describes as a PC via 6 attributes (Identified data tampered features, Table 2) Recycle Bin (RB), Internet History (IH), Cache (CSH), Cookies (COO), Restore Point (RP) and Prefetch (PR) can be seen in an *A = 6-dimensional*.

Table 3: describing dataset in multivariate visualization

D = $\{pc_i\}$ then D = $\{pc_1, pc_2, pc_3, pc_4, pc_5\}$	***Data Points*** (*e.g. PC's, laptop and etc.)*
A = {RB, IH, CSH, COO, RP, PR} 1 2 3 4 5 6 (number of attributes)	***Attributes*** (*identified data tampered*)
$(a_1^{\boldsymbol{i}}, \dots, a_{\boldsymbol{n}}^{\boldsymbol{i}}) \in A^{\boldsymbol{n}}$ *then* $(a_1^1, \dots, a_6^1) \in A^{\boldsymbol{6}}$ and $(a_{RB}^1, a_{IH}^1, a_{CSH}^1, a_{COO}^1, a_{RP,}^1 a_{PR}^1) \in A^6$ for pc_1 . .	

.

$(a_1^i, \dots, a_n^i) \in A^n$ *then* $(a_1^5, \dots, a_6^5) \in A^6$ and $(a_{RB}^6, a_{IH}^6, a_{CSH}^6, a_{COO}^6, a_{RP,}^6 a_{PR}^6) \in A^6$ for pc_5

The parallel coordinates map each dimension to a separate vertical axis (*column*). However, instead of corresponding to the horizontal row, each data point pc_i is now mapped as a polyline that connects the points on the vertical axes whose coordinates (y values) equal the point attribute a_i (see Figure 8 and Figure 9).

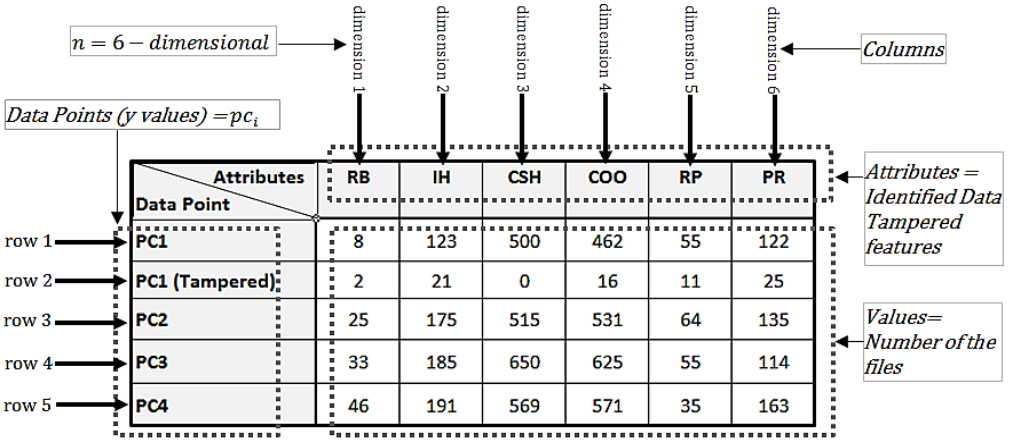

Figure 8: Designing Visualization (multivariate dataset with several attributes or variables per data point)

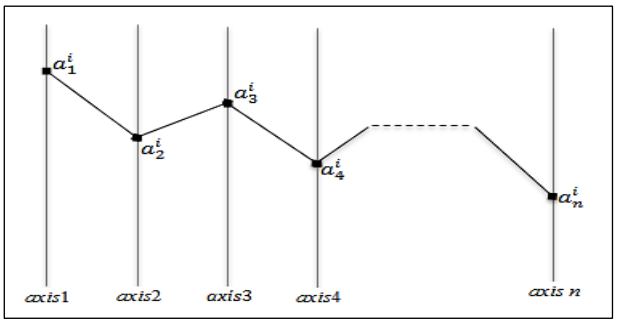

Figure 9: Parallel Coordinates Visualization

2.2.2 Result (finding suitable visualization)

Following the previous step, it is clear that what data, what size of data and what visualization type's looking for. It was developed open source software in relation to parallel coordinate visualization, to import data (loading the file) and visualize them. Figure 10 illustrates parallel coordinate visualization technique for the above dataset (see Figure 7). There is a six-axes; corresponding to the first six data attributes (data tampered features, Table 2). Each axis are scaled individually to show the full range of its attribute value (number of the file). Each polyline (row) represents a different PC of the determined five in the dataset. The polylines are drawn with a certain amount of transparency and areas covered by many lines. *The red line and associated labels show the detail on the suspicious PC record under the mouse pointer (PC1 tampered)*.

This visualization already shows a number of the facts. A bunch of the lines that run parallel indicate a similar data point, the lines widely spread apart along an axis to show a large variety of the data attribute. It is apparent dataset distribution, the correlation between attributes (identified data tampered features), and data point (PC's) comparison.

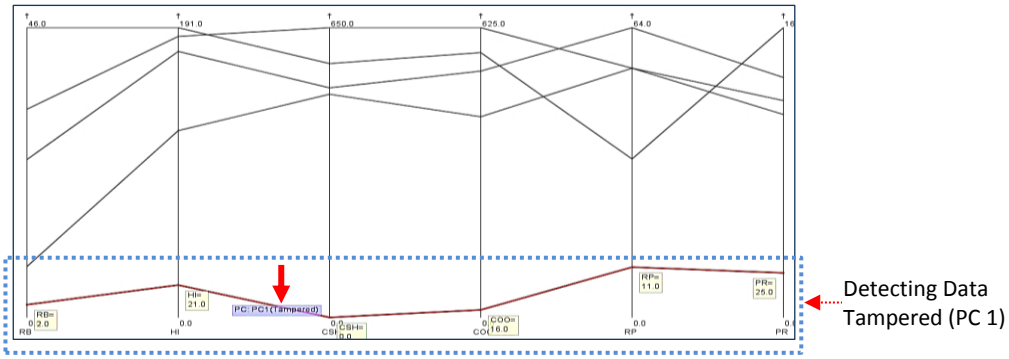

Figure 10: Parallel Coordinates Visualization View, Scale Axes 0-max

Other important features of this tool are viewing parallel coordinates visualization in two other scale axes types. (see Figure 11 and Figure 12)

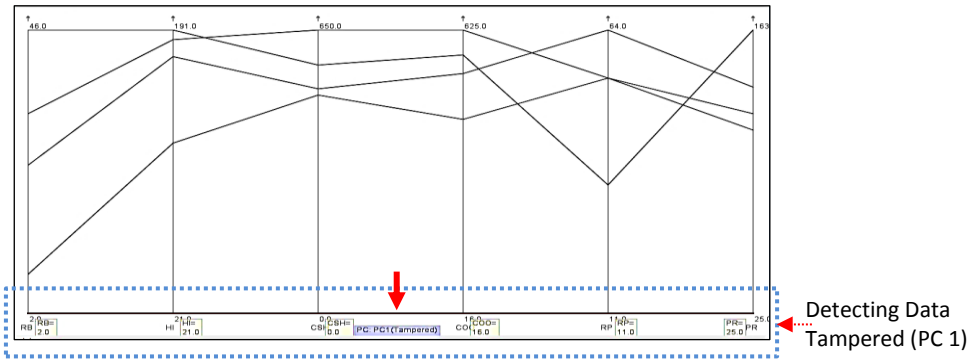

Figure 11: Parallel Coordinates Visualization View, Scale Axes min-max

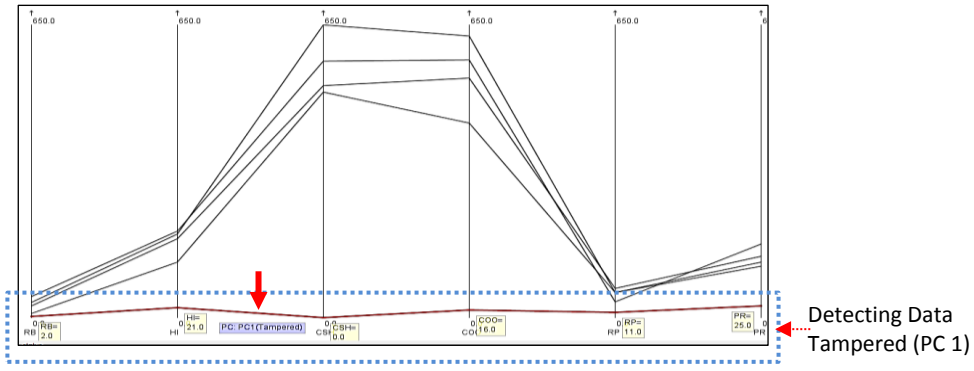

Figure 12: Parallel Coordinates Visualization View, Scale Axes min-max (abs)

Any action by a user on a computer, whether it's surfing the internet, communication or file storage, affects data stored in the computer. As a result, Parallel coordinates visualize PC1, PC2, PC3, PC4, and PC1 (tampered) behaviour in the normal way of usage in three different views. As explained earlier in the step 2.1.3, I used the AF tool to wipe (or delete) data from PC1 for tampering purpose. Above figures show other PC's with some activities. But, PC1 has no activity and almost flat as displayed with a red line (indicated by arrows). PC1 it seemed as a suspicious case of the tampering for investigators just by first looking at the visualization. This is a remarkable result in the forensic field which never done before. When investigators facing with the important items such a time and cost, there is no need to look for a needle in a haystack specifically, when an offender familiar with the digital investigation process and no traces left behind himself or herself. In this part, I identified suitable visualization to visualize identified data tampered features in a significant, meaningful and simple way to help the investigator to detect digital evidence tampering and data anomalies.

3. Conclusion

The main goals of the current study is to identify tampered data features after applying any Anti-Forensics tools and visualize it in a comprehensive way for investigators. As a result, *computer-aided diagnostic digital evidence tampering (CADDET)* explored a semi-automated approach based on visualization of relevant data properties, helping human investigators to detect digital evidence tampering and anomaly.

One of the outstanding features of this research is its malleability. It can easily apply to partial or the entire dataset (depend on investigators' interest) in the digital devices, visualize, and reveal offender concealment behaviour in relation to detection of evidence tampering. It is clear that CADDET is capable of adapting to changes in technology and tampering behaviour over time, there is no risk when digital investigators are faced with an issue they have not encountered before. The explored method is applicable to another operating system (for instance: Linux, IOS, Android and etc.) to detect digital evidence tampering and reveal offender concealment (or behaviour).

CADDET sheds new light and explore new insight to contribute investigators to fill a gap of detecting digital evidence tampering. This semi-automated approach had never been studied before and is novel in the context of digital forensics.

4. Future Work

The study could be repeated using the explored approach in other common operating systems such as:

- Windows Server (2003+),
- MacOS X,
- Linux (Ubuntu, Debian, Mint, etc.),
- Android and etc.

to detect evidence tampering in digital forensics field.

The interactive aspects of visual analytics will continue to future work.

5. Bibliography

AccessData, 2010. *FTK Imager.* [Online]
Available at: http://accessdata.com/product-download
Acesoft, 2001. *Tracks Eraser Pro 9.* [Online]
Available at: http://www.acesoft.net/
Anderson, E., 2016. *Visual Index, Parallel Coordinates.* [Online]
Available at: https://github.com/d3/d3/wiki/Gallery
c|net, n.d. *Windows Eraser.* [Online]
Available at: http://download.cnet.com/Windows-Eraser/3000-2144_4-10550310.html
Davies, J., 2016. *Parallel Coordinates.* [Online]
Available at: http://bl.ocks.org/jasondavies/1341281
Dimmick, V., 2005. *CCleaner.* [Online]
Available at: https://www.piriform.com/ccleaner
Gladyshev, P., 2013. *DFire.* [Online]
Available at: http://dfire.ucd.ie/
Gladyshev, P., 2016. *Forensics Styla.* [Online]
Available at: https://bitbucket.org/dfirelabs/forensic-styla/downloads
Inselberg, A., 1997. *Multidimentional Detectives, Parallel Coordinates (0.7.0).* [Online]
Available at: https://syntagmatic.github.io/parallel-coordinates/
Keim, D. A., 2002. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics,* 7 August, 8(1), pp. 1-8.
Shneiderman, B., 1996. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. *IEEE Symposium on Visual Languages,* Volume 96, pp. 336-343.
Tableau, 2015. *parallel coordinates test.* [Online]
Available at:
https://public.tableau.com/s/profile/dmackay#!/vizhome/parallelcoordinatestest/parallelcoordinatestest
Tarau, P., 2014. *Styla.* [Online]
Available at: http://www.cse.unt.edu/~tarau/

Teerlink, S. & Erbacher, R. F., 2006. Improving the computer forensic analysis process through. *ACM DL,* pp. 71-75.
Wikipedia, 1972. *Prolog,.* [Online]
Available at: https://en.wikipedia.org/wiki/Prolog
Wikipedia, 2016. *Trash (RecycleBin).* s.l.:s.n.
Wikipedia, 2017. *Disk Image.* s.l.:s.n.
X-Ways, 2000. *WinHex: Computer Forensics & Data Recovery Software,.* [Online]
Available at: http://www.x-ways.net/winhex/
X-Ways, 2002. *X-Ways Forensics: Integrated Computer Forensics Software.* [Online]
Available at: http://www.x-ways.net/forensics/