Self Alteration Detectable Image Log File for Web Forensics

Vimal Kumar
Computer Science and Engineering.Dept.
Motilal Nehru National Institute of Technology
Allahabad-211004, India

Akhilendra Pratap Singh
Computer Science and Engineering Dept.
Motilal Nehru National Institute of Technology
Allahabad-211004, India

Anjani K. Rai
Computer Science and Engineering.Dept.
Motilal Nehru National Institute of Technology
Allahabad- 211004, India

Manoj Wairiya
Computer Science and Engineering Dept.
Motilal Nehru National Institute of Technology
Allahabad-211004, India

ABSTRACT
Nowadays log file plays vital role in web forensic as digital evidence. Hence security of log file is a major topic of apprehension. In this paper a model of image logging server having alteration detectable capability, is proposed. According to this approach we first convert a text log file into image log file with the help of bit encoding technique and tamper detection capability is achieved by self embedding fragile watermark scheme. If any alteration is done on image log file then due to nature of fragile watermark, one can easily locate that tampered region. Proposed model is also able to ensure all security requirements like Authenticity, Integrity and confidentiality.

General Terms
Digital Forensics

Keywords
Web forensic, Cyber forensic, fragile watermark, self embedding, Log file, Image logging server.

1. INTRODUCTION
We Today people absolutely rely on digital media. Since technology is advancing with burgeoning rate hence a new opportunity is opened for Business Company and legal agencies to deploy those technologies. It is very beneficial for worldwide users, but on the other hand, due to some loop holes of those technologies, malicious use for committing crime has also been increased. That's why it is very essential to prevent the criminals and their succession of committing crime to smooth the progress of the secure utilization of new technological services. Thus, for the analysis of law enforcement and cyber crime, Cyber Forensic [1][2][3][4] came into picture. The primary ambition of cyber forensic examination is to recognize digital evidence for an investigation. Cyber forensic evidence must fulfill some security requirements [3] like Accuracy, Integrity, Authenticity and confidentiality. Digital evidence must be unquestionable, accurate, absolute and acceptable by juries as well as Permissible with common law and legislative rules. Cyber forensic can be widely categorized into four classes [1] viz. computer forensic, network forensic, web forensic and mobile forensic as shown in figure 1. The concept of web forensic deals with the process of monitoring access logs, detection of any alteration in log files as well as recovery of those alterations. The significance of web forensic is to investigate web attacks and prevent those in future, using analysis of log files. In order to carry out a systematic analysis of the hacking Attempt[5], it is advised that the Investigator must investigate all four type of logs namely web server logs, Any 3rd. party installed software logs, Operating system logs and client side logs[6][7]. Now we can assume the importance of log files in web forensic. There are many web attacks which are used to alter the integrity of log files like user to root (U2R) attack, in which The intruder exploits some vulnerability associated with the operating system and web server environment of the server machine under attack to perform the conversion from user to root level [8]. After getting the root privileges, the intruder has full control and access on the server machine to get backdoor entries for future misuse and change system logs[9]. Jianhui *has* proposed a standard approach for web forensic in [10]. According to his approach, access log files are combined with the timestamp and other clues presented in the log file, then operation facts are formed and represented by XML with a decision tree. Now intrusion behavior evidence can be abstracted. Especially when a hacker tried to wipe his trace, the system can detect it effectively. Liu Jiqiang has proposed security of logs in [11]. In this approach, the security of logs is based upon the security of the systems which the logs were kept in. We bring forwards a system called Secure Audit Logs Server which adopts encryption and dynamic MAC to guarantee the integrity and dependability of the logs. This is important for obtaining effective evidences. Another approach is proposed by Patrick Stahlberg et.al.[12] in which they shows that how to preserve database table storage, the transaction log, indexes, and other system components. Then address the problem of unintended data retention by proposing a set of system transparency criteria and at last apply specific techniques for secure record deletion and log expunction that increase the transparency of database systems, making them more resistant to forensic analysis.

This paper is organized as follows-

Section 2 shows our proposed algorithm and section 3 shows experimental results, conclusion of paper is given in section 4.

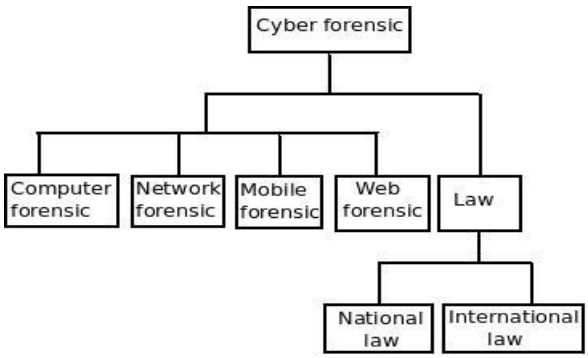

Fig.1 Hierarchy of Cyber forensic

2. PROPOSED ALGORITHM

Before discussing the proposed algorithm we must have some basic knowledge about the attribute and significance of web logs.

2.1 Web Log Files

When web user requests to fetch information from web server then these all activities are recorded in web log files. The primary goal of web logs for web server is to analyse the user behaviour at the same time those log files are also useful for web forensic. A log file can be located in three different places: i) Web Servers, ii) Web proxy Servers, and iii) Client browsers. Web Server logs are plain text (ASCII) files, that is independent from the server platform [13]. Since alteration in plain text logs are uncomplicated and may be problematic during U2R attack [8]. Because in this case, user can get full privilege on server side logs and can do some very serious modifications on it, like delete some essential entries from log file which may be an important evidence for any crime in web forensic.

2.2 Organization of log files

Most of the web log files are in CLF (common log format) established by NCSA and CERN. CLF mainly includes information of the client IP address, date, time, user authentication, server name, server IP address, server port, server method, URI-Stem, Server URI-Query and Status etc[3] [13].

In our approach we are considering a Image logging server on the top of the client server model as shown in figure 2. The work of this server is to convert text log file into image log file and store it in to various replicas.

Web server contains log files in text format. Periodically it communicates with image logging server and send to it chunk of log files to it for converting into image log file which has alteration detectable capability. Image logging server applies proposed algorithm for getting image file from text file which is in unreadable format. . It will be useful to prevent various types of attacks which are based on log files.

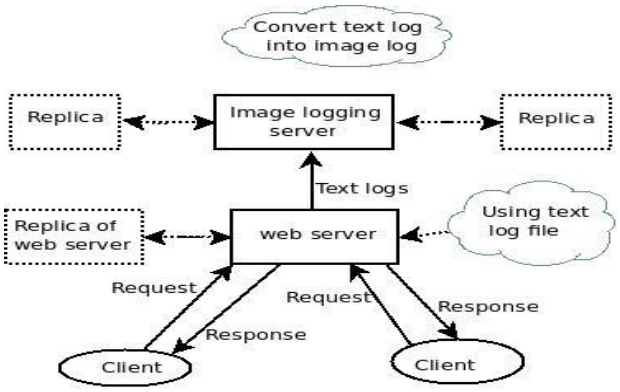

Fig.2 Implementation of image logging server

In suggested approach image logging server applies a self embedding watermarking [14][15][16][17][18[19] on the image logs so that if any alteration is done on image log file intentionally or unintentionally it will be detected with high precision. If a legitimate user wants to modify the image log file then by using inverse algorithm he can convert the image log file into text log file, which can be handled easily.

2.3 Algorithm for converting text log file into image log file

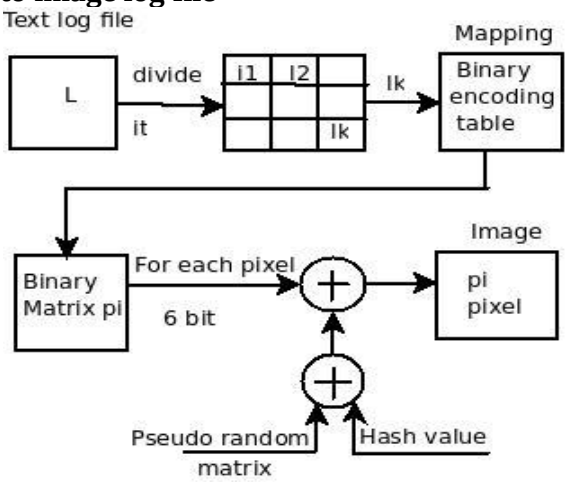

Fig.3 Block diagram for text to image log conversion

Figure 3 shows the block diagram for converting text log files into image log files which contain following steps.

Step 1- Take a log file "L" and divide it in to a fixed partition $\{l_1, l2, l_{3,\ldots} l_k\}$ on the basis of size of log file.

Step 2- Convert all alphanumeric character of log into binary form on the basis of binary encoding table as shown in table1. Each binary value must have 6 bits.

Step 3- Create a zero matrix 'M' of unit 8 formats.

Step 4- Separate all bits as bunch of 6 bits and put the all 6 bit value on the 6 MSBs of corresponding pixel of 'M'.

Step 5- Apply 6 MSBs of each pixel as an input to a hash function which gives output either 1 or 0.

Step 6- Create a Pseudorandom matrix 'P' using a secret key of size 'M'.

Step 7- Make a pair for element of P and corresponding hash bit and put those two bits on the position of 2 LSB in matrix M as shown in figure 4.

Step 8- Convert all binary in to decimal form which has range from [0 255] .

Step 9- Write it in to a form of image.

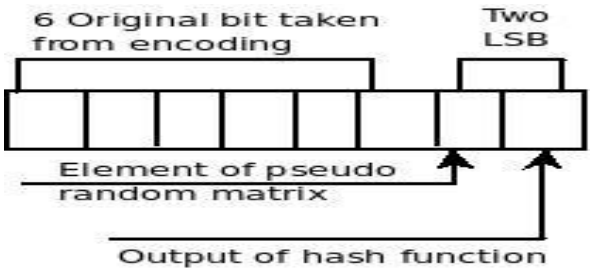

Fig. 4 Structure of a pixel in form of bits

TABLE I
Binary encoding

'A/a' to 'Z/z'	000000 to 011001
'0' to '9'	011010 to 100100
' '	100101
'.'	100110
':'	100111
'/'	101000
'-'	101001
'_'	101010
'?'	101011
';'	101100
'%'	101101
'='	101110

Now we can see that in this approach one character of log file is replaced by one pixel of image which seems to be negligible. It means one image can store a very large text log file. Here image log will be in unreadable form hence confidentiality is achieved at the same time watermark is inserted with the help of a secret key so authentication is also achieved. Since we are using fragile watermarking, it always maintains the integrity of image log file.

2.4 Algorithm for converting image log file into text log file

Suppose an attacker has done some attacks on image log file to alter the original content of image. So by applying the following steps as shown in figure, a legitimate user or server administrator can easily spot the location of the alteration. Similarly if an administrator wants to do some legal modification on the log file or wants to convert it into readable format.

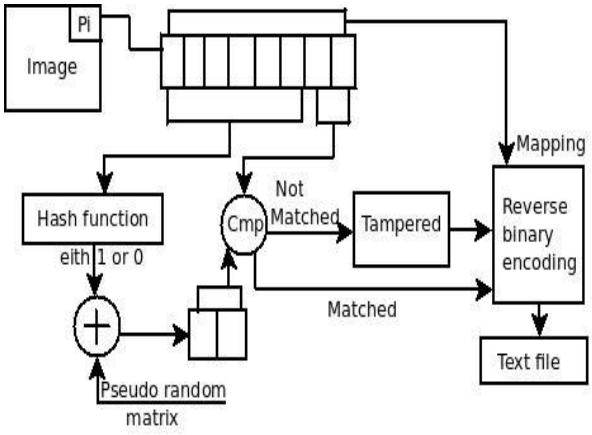

Fig.5 Block diagram for image to text log conversion

Step 1- Take a image log file and convert it into a form of matrix 'M' which has element from [0 255] .

Step 2- Convert all pixel in to binary form with 8 bits.

Step 3- Take 6 MSBs of each pixel and apply it into some hash function and set output either 0 or 1.

Step 4- Create a pseudorandom matrix using same secret key.

Step 5- Make a pair for element of pseudorandom matrix and corresponding element of hash function.

Step 6- Extract 2 LSBs of each pixel and compare it with corresponding pair of hash and random matrix value.

Step 7- If mismatch found then mark it as a tampered one else leave it as it is.

Step 8- Extract 6 MSBs from each pixel and create another matrix.

Step 9- Map those binary value with their proper binary reverse encoding value using table 2. And get the log file into text format.

TABLE II

Binary decoding

000000 to 011001	'A/a' to 'Z/z'
011010 to 100100	'0' to '9'
100101	' '
100110	'.'

100111	‘ : ‘
101000	‘ / ‘
101001	‘-’
101010	‘ _ ‘
101011	‘ ? ‘
101100	‘ ; ‘
101101	‘ % ‘
101110	‘ = ‘

3. EXPERIMENTAL RESULTS

Proposed algorithm is simulated in Matlab 2010 and we have taken the text log files of various sizes from our college’s web server logs. According to our scheme we first take text log files as an input to the algorithm as shown in figure 6.

```
172.31.134.212 TCP_MISS/000 0 GET http://www.google.com/ - DIRECT/209.85.153.104 -
172.31.134.201 TCP_MISS/502 1514 GET http://fxfeeds.mozilla.com/en-US/firefox/head
172.31.134.212 TCP_MISS/502 1450 GET http://www.google.com/ - DIRECT/209.85.153.10
127.0.0.1 TCP_DENIED/403 1491 POST http://safebrowsing.clients.google.com/safebrow
127.0.0.1 TCP_DENIED/403 1491 POST http://safebrowsing.clients.google.com/safebrow
172.31.134.212 TCP_MISS/502 1450 GET http://www.google.com/ - DIRECT/209.85.153.10
127.0.0.1 TCP_DENIED/403 1415 GET http://www.google.co.in/ - NONE/- text/html
127.0.0.1 TCP_DENIED/403 1383 GET http:/// - NONE/- text/html
127.0.0.1 TCP_DENIED/403 1405 GET http:///favicon.ico - NONE/- text/html
127.0.0.1 TCP_DENIED/403 1405 GET http:///favicon.ico - NONE/- text/html
127.0.0.1 TCP_DENIED/403 1423 GET http://mail.google.com/mail/ - NONE/- text/html
127.0.0.1 TCP_DENIED/403 1435 GET http://mail.google.com/favicon.ico - NONE/- text
127.0.0.1 TCP_DENIED/403 1435 GET http://mail.google.com/favicon.ico - NONE/- text
127.0.0.1 TCP_DENIED/403 1423 GET http://mail.google.com/mail/ - NONE/- text/html
172.31.134.212 TCP_MISS/502 1450 GET http://www.google.com/ - DIRECT/209.85.153.10
127.0.0.1 TCP_DENIED/403 1423 GET http://mail.google.com/mail/ - NONE/- text/html
```

Fig.6 Snapshot of a text log file.

Image log file consist of two color. White colour is occupied by the valuable information of text log file whereas black portion shows the absence of information or remaining space in image log file as shown in figure 7.

Fig. 7 Original image log file version for given text log file

Text log file size and black colour present in image log file is inversely proportional means when we increases the size of text log file then the area in image log file which contains black portion will decrease. Suppose an attacker has done some alteration in image log files like shown in figure 8.

Fig. 8 Altered image log file by attacker

If we are not having the original image log file shown in figure 7, then one can not say that given image log file shown in figure 8 is altered and where alteration is done. Because we loose the original one and there is no mean for comparison.

But now we pass this altered image log file into our image to text log file conversion algorithm then we get the result as shown in figure 9.

Fig. 9 Altered image log with tamper detected spot

Here the gray coloured area shows the tampered portion in image log file which was very difficult to spot in text log file because text log can easily be modified like one can change the IP address, date, time of crime and we can not locate the altered area. After ensuring that image log file is altered somewhere, that log file as a digital evidence. Now when we convert those image log file in to text log file then it will not be similar like original one as shown in figure 10.

```
172.31.134.212 TCP_MISS/000 0 GET http://www.google.com/ - DIRECT/209.85.153.104 -
172.31.134.201 TCP_MISS/502 1514 GET http://fxfeeds.mozilla.com/en-US/firefox/head
172.31.134.212 TCP_MISS/502 1450 GET http://www.google.com/ - DIRECT/209.85.153.10
127.0.0.1 TCP_DENIED/403 1491 POST http://safebrowsing.clients.google.com/safebrow
127.0.0.1 TCP_DENIED/403 1491 POST http://safebrowsing.clients.google.com/safebrow
172.31.134.212 TCP_MISS/502 1450 GET http://www.google.com/ - DIRECT/209.85.153.10
127.0.0.1 TCP_DENIED/201 140 POST http://www.google.co.in/ - NONE/- text/html
127.0.0.1 TCP_DENIED/202 1412 POST http:/// - NONE/- text/html        /201 140 POST
127.0.0.1 TCP_DENIED/203 1800 POST http:///favicon.ico - NONE/- text/h/202 1412 POS
127.0.0.1 TCP_DENIED/204 1505 GET http:///favicon.ico - NONE/- text/h/203 1800 POS
127.0.0.1 TCP_DENIED/403 1423 GET http://mail.google.com/mail/ - NONE/204 1505 GET
127.0.0.1 TCP_DENIED/403 1435 GET http://mail.google.com/favicon.ico - NONE/- text
127.0.0.1 TCP_DENIED/403 1435 GET http://mail.google.com/favicon.ico - NONE/- text
127.0.0.1 TCP_DENIED/403 1423 GET http://mail.google.com/mail/ - NONE/- text/html
172.31.134.212 TCP_MISS/502 1450 GET http://www.google.com/ - DIRECT/209.85.153.10
127.0.0.1 TCP_DENIED/403 1423 GET http://mail.google.com/mail/ - NONE/- text/html
```

Fig. 10 Alteration done on text log file

Now we can conclude that in only presence of the figure 10, no one can spot the altered portion in log file which are shown by rectangular box but if it will be converted into image log by proposed algorithm then one can easily decide that integrity of log file is compromised.

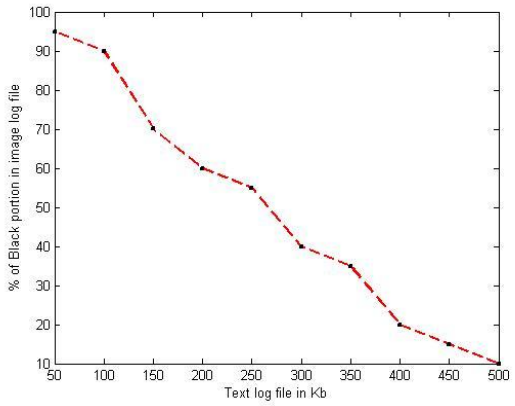

Fig. 11 Graph between size of text log and % of black portion in image logs

4. CONCLUSION

In this paper, we propose the model of the image logging server which has alteration detectable capability. By the help of image logging server, web text logs are converted into image logs. which are in unreadable format and having tamper detection tendency which is not achieved by earlier proposed algorithm. Experimental results show the effectiveness of our proposed algorithm. One image log file is able to store a huge size of text log file as we can see in provided results. Converted image log ensures all security requirements like Authenticity, integrity and confidentiality and can be used as digital evidence in web forensics.

5. REFERENCES

[1] Sandeep Sharma, Swapnil Gupta," Implementation of Image Logging System for Digital Evidense in Web Forensic and Law Enforcement," School of Computer Science & IT Devi Ahiliya Visghwavidyalaya(DAVV) Indore(MP) 2008-2010.

[2] Brian Carrier," Open Source Digital Forensics Tools," take Research Report, October, 2002. Modeling and Simulation Design. AK Peters Ltd.

[3] Seunghee Yoo, Yilhyeong Mun, Dongsub Cho, "Implementation of Image Loging Server for Web Forensic," 978-1-424426249/08, IEEE, 2008.

[4] US-CERAT,"Computer Forensics," a government organization, 2008.

[5] Web Forensics, Jess García, http://www.jessland.net.

[6] Indian Computer Emergency Response Team, "Web Server Security Guideline," CERT-IN, August, 17, 2004.

[7] Patrick Stahlberg, Gerome Miklau, and Brian Neil Levine, "Threat to privacy in the Forensics Analysis of Database Systems," SIGMOD'07, Beijing, China, June 12–14, 2007.

[8] Kapil Kumar Gupta, Baikunth Nath, Ramamohanarao Kotagiri," Layered Approach using Conditional Random Fields for Intrusion Detection," IEEE Transaction on Dependable and Secure Computing Vol 7, NO 1,January-March 2010.

[9] Spector, A. Z. F89. Maheshkumar Sabhnani, Gursel Serpen," Formulation of a Heuristic Rule for Misuse and Anomaly Detection for U2R attacks in SolarisTM Operating System Environment," EECS Dept, University of Toledo, Toledo, Ohio 43606, USA.

[10] Jianhui LIN1, "A Web Forensic System Based On Semantic Checking,," International Symposium on Computational Intelligence and Design, 2008.

[11] Liu Jiqiang Han Zhen Lan Zengwei," Secure Audit Logs Server to Support Computer Forensics in Criminal Investigations," Proceedings of IEEE, TENCOW02.

[12] Patrick Stahlberg, Gerome Miklau, and Brian Neil Levine, "Threat to privacy in the Forensics Analysis of Database Systems," SIGMOD'07, Beijing, China, June 12–14, 2007.

[13] Mohd Helmy Abd Wahab, Mohd Norzali Haji Mohd , Hafizul Fahri Hanafi, Mohamad Farhan,Mohamad Mohsin," Data Pre-processing on Web Servers Logs for Generalized Association Rule Mining," World Academy of Science, Engineering and Technology 48 2008.

[14] Pengfei Wang, Yewang Chen, "A Fragile Watermarking Algorithm Based on Logistic System and JPEG," 978-1-4244-5555-3/10 IEEE, 2010.

[15] ShuiHua, Han and Chao-Hsien Chu," Tamper Detection in RFID-Enabled Supply Chains Using Fragile Watermarking," IEEE International Conference on RFID The Venetian, Las Vegas, Nevada, USAApril 16-17, IEEE,2008.

[16] Xianzhong Long, Hong Peng, Changle Zhang, Zheng Pan, Ying Wu," A FragileWatermarking Scheme for Tamper-Proof of Web Pages, " WASE International Conference on Information Engineering,IEEE,2009.

[17] Jiaxin Yu, Xinsheng Wang, Jianfei Li, Xu Nan," A Fragile Document Watermarking Technique Based on Wet Paper code," ," International Conference on Intelligent Information Hiding Multimedia Signal Processing IEEE 2008.

[18] Hongxia Wang, Changxing Liao," Compressed-Domain Fragile Watermarking Scheme for Distinguishing Tampers on Image Content or Watermark," 978-1-4244-4888-3/09, IEEE,2009.

[19] Hongbin Kong, Zhengquan Zeng, Chunjie Zhang, Cheng Lu, ShaowenYao," A Fragile Watermarking Scheme for OWL-Based ontology,integrity,protection".