VOL 2 (2018) NO 4 - 2

e-ISSN : 2549-9904
ISSN : 2549-9610

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

E-Raser: File Shredder Application With Content Replacement By Using Random Words Function

Nur Farah Aqilah Mohd Nahar[#], Nurul Hidayah Ab Rahman[#], Kamarudin Malik Mohammad[#]

Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Malaysia
E-mail: fara.qiela@yahoo.com, hidayahar@uthm.edu.my, malik@uthm.edu.my

***Abstract*— Data shredding indicates a process of irreversible file destruction while file shredder is the program designed to render computer-based files unreadable by implementing overwriting method to destroy data in the content of a file. The addressable problem with existence of file recovery tools is it may lead to data leakage, exploitation or dissemination from an unauthorized person. Thus, this study proposed a file shredding application named E-Raser which replacing the content of the file using random words function algorithm. A file shredder application named E-Raser was developed to shred Microsoft Word documents with (.doc) or (.docx) format. The implemented algorithm replaced the original content of the files with uninformative words provided by the application. After rewriting phase is complete, shredding process take place to make the file unrecoverable. Object Oriented Software Development was used as the methodology to develop this application. As a result, E-Raser achieved the objectives to add, remove, rewrite, display and shred files. Also, E-Raser is significantly facilitates users to securely dispose their file, protect the confidentiality and privacy of the file's content.**

***Keywords*—Data shredding, File shredder, Overwriting, Unrecoverable file.**

I. INTRODUCTION

Most enterprises apply Information Lifecycle Management - the process on how to manage business data from conception until disposal in a manner that optimizes storage, access, and cost characteristics has become increasingly important [1]. A common situation for computer users is deleting unnecessary folder and files without any concern on how the folder and files are being disposed. It is, however, the computer only deletes the pointer of the file that tells operating system the file exists.

It makes users see that the "deleted" space as free even though files' content are still exist in a particular sector on the hard drive. More importantly, it is possible to recover those "deleted" files using file recovery tools. According to The Sun's investigators, the experts at Kaspersky Lab were able to restore all information of the previous owners of three formatted second-hand computers they bought for an experiment which prove that it could also be easy for any technology-savvy fraudster to even recover it [2][3].

Shredding, on the other hand, is a technique to overwrite the segment of drive where the file was stored. The more it overwrites the segment, the harder it is to recover adequate data to reassemble a deleted file.

This study is therefore proposed E-Raser as one of the alternatives for user to permanently delete files with appropriate procedure. Instead of only clean wiping the hard disk's cluster; E-Raser will also rewrite the words in the file's content. The aim of this project is to develop a file shredding application with content replacement by using random words function. In order to achieve the aim, three objectives have been set as follows:

i. to design a file shredding application that can replace the content of selected document files with uninformative words being declared in the application.
ii. to develop a file shredding application by using Microsoft Visual Studio 2010 and C# language programming.
iii. to perform a testing for the developed file shredding application.

This paper is organized as follows. Section II describes the literature review on the related terms, existing methods and comparative study on existing file shredding tools. Section III introduces methodology used throughout this paper. Section IV describes the system analysis and design for the paper. Section V will justify the implementation and testing of the application and lastly, Section VI presents our conclusion.

II. RELATED WORKS

This section discusses the related terms to the application such as data shredding, overwriting and also file shredder, examples of method being used and also comparative study of the existing applications with the proposed application.

A. Data Shredding

Shredding is a process of irreversible file destruction, so that its content could not be recovered [4]. A method for continually shredding data within a data storage subsystem where all the data of a first storage element is made permanently unreadable followed by the performance of a data shred process.

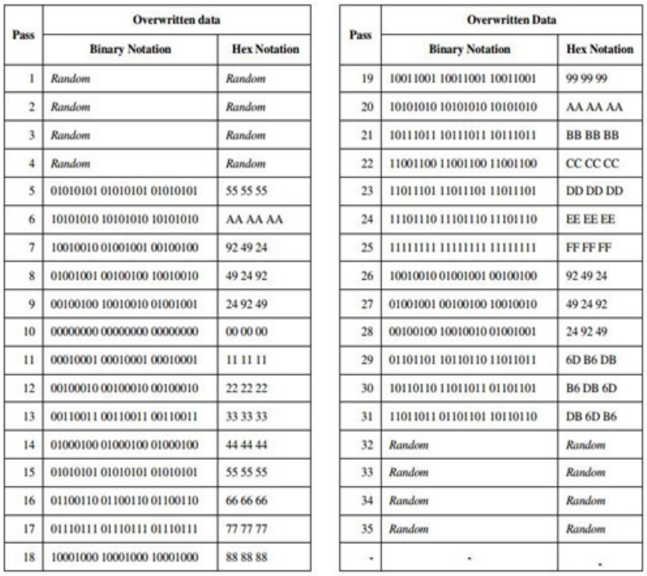
In data shred process, a second storage element is initially selected, which includes a first storage object to be preserved and a second storage object to be shredded. Once the second storage element has been selected, data of the first storage object is stored within the first storage element. Thereafter, all data of the second storage element is rendered permanently unreadable [1].

B. Overwriting

A work by Wei *et al* [5] proposed that overwriting is a process of writing a binary set of data in computer data storage and is a term used to describe when new information replaces previous information or data. Overwriting generally occurs when unused file system clusters have been written with new data, though overwriting is also used in security algorithms. These algorithms use a precise set of rules to remove any part of an original data from the memory by writing a new raw data in the memory. Previous works of [6],[7],[8] and [9] reported the implementation of overwriting methods.

1) Peter Gutmann's Algorithm – 35 Passess

Bennison *et al* [6] proposed Peter Gutmann to ensure that the recovery of data can be made as difficult as possible for an attacker by offering the 35 overwrite passes algorithm. The general concept behind an overwriting scheme is to flip each magnetic domain on the disk back and without writing the same pattern twice in a row [10].



Pass	Overwritten data	
	Binary Notation	Hex Notation
1	*Random*	*Random*
2	*Random*	*Random*
3	*Random*	*Random*
4	*Random*	*Random*
5	01010101 01010101 01010101	55 55 55
6	10101010 10101010 10101010	AA AA AA
7	10010010 01001001 00100100	92 49 24
8	01001001 00100100 10010010	49 24 92
9	00100100 10010010 01001001	24 92 49
10	00000000 00000000 00000000	00 00 00
11	00010001 00010001 00010001	11 11 11
12	00100010 00100010 00100010	22 22 22
13	00110011 00110011 00110011	33 33 33
14	01000100 01000100 01000100	44 44 44
15	01010101 01010101 01010101	55 55 55
16	01100110 01100110 01100110	66 66 66
17	01110111 01110111 01110111	77 77 77
18	10001000 10001000 10001000	88 88 88

Pass	Overwritten Data	
	Binary Notation	Hex Notation
19	10011001 10011001 10011001	99 99 99
20	10101010 10101010 10101010	AA AA AA
21	10111011 10111011 10111011	BB BB BB
22	11001100 11001100 11001100	CC CC CC
23	11011101 11011101 11011101	DD DD DD
24	11101110 11101110 11101110	EE EE EE
25	11111111 11111111 11111111	FF FF FF
26	10010010 01001001 00100100	92 49 24
27	01001001 00100100 10010010	49 24 92
28	00100100 10010010 01001001	24 92 49
29	01101101 10110110 11011011	6D B6 DB
30	10110110 11011011 01101101	B6 DB 6D
31	11011011 01101101 10110110	DB 6D B6
32	*Random*	*Random*
33	*Random*	*Random*
34	*Random*	*Random*
35	*Random*	*Random*
-	-	-

Fig. 1 A series of overwriting patterns in Peter Guttmann's algorithm

2) Bruce Schneier"s Algorithm – 7 Passes

Bruce Schneier recommends wiping a drive seven times by overwriting the data on a storage device with a one, and then a zero, and finally with several passes of random characters [11], as follows:

- **Pass 1:**Writes a zero (0x00)
- **Pass 2:** Writes a one (0xFF)
- **Pass 3:** Writes a stream of random characters
- **Pass 4:** Writes a stream of random characters
- **Pass 5:** Writes a stream of random characters
- **Pass 6:** Writes a stream of random characters
- **Pass 7:** Writes a stream of random characters

3) Germany BSI Verschlusssachen-IT-Richtlinien (VSITR) – 7 Passes

VSITR is a software based data sanitization method which utilizes a combination of all ones, zeros and random data passes. As discussed by Tim [12], 7 pass of VSITR data sanitization method implementation includes:

- **Pass 1:**Writes a zero (0x00)
- **Pass 2:** Writes a one (0xFF)
- **Pass 3:** Writes a zero (0x00)
- **Pass 4:**Writes a one (0xFF)
- **Pass 5:** Writes a zero (0x00)
- **Pass 6:** Writes a one (0xFF)
- **Pass 7:**Writes a random character

C. Comparative Study of Existing File Shredding Tools

This section discusses and compare the existing file shredding tools.

1) Recuva

Recuva is a file recovery software used to recover any kind of files that have been permanently deleted and marked as free space by the operating system [13][14]. For this paper, Recuva has been used to test selected file shredding tools and E-Raser to assure that each of the tools do not allow file recovery.

2) Eraser

Eraser is open source software developed by the Eraser Team [15] program that can securely delete files, folders, unused disk space or even entire drives. It allows user to create scheduled erasing tasks and also supports 13 different erasure methods, the default method being used by Eraser is the Gutmann standard.

3) Alternate File Shredder

Alternate File Shredder is an invention by Alternate Tool [16]; a small application that will also permanently delete files and let the user to choose either deleting individual files or entire folders to prevent from unnecessary tasks like one deletion per time. This tool offers only one erasing method which is random data writing. However, the user can choose the number of times for the deleted files to be overwritten.

4) Freeraser

The Codyssey Organization [17] invented Freeraser as a portable tool designed to allow user to securely deleting files using drag and drop. It offers user to delete files using a fast method of filling the space with random data with one pass, using a forced method that utilizes the DoD 5220.22M,3-pass standard; or using the 35 passes, Gutmann method. Freeraser can function on three levels:

i. A fast destruction (standard 1-round filling of random data)

ii. A forced destruction (3 rounds of filling according to DoD 5220.22M standard)

iii. An ultimate destruction (35 rounds of filling with data according to Guttman algorithm).

5) E-Raser

Most of the tools stated have been using the usual algorithm which either writing zeros, ones and also random characters. None of the tools have a function where the content of the files itself being rewritten with uninformative words yet. Hence, the E-Raser application would be an application which implements an algorithm to generate random data for overwriting and also add on a new method which is replacing the original content of the file with new content.

Findings of the existing file shredder tools and comparison with the proposed E-raser is summarized in Table 1.

TABLE 1
COMPARATIVE STUDY OF EXISTING FILE SHREDDING TOOLS

Tools **Features**	**ERASER**	**ALTERNATE FILE SHREDDER**	**FREE RASER**	**E-RASER**
Erasing Algorithms	13	1	3	1
Recoverable Files (tested by Recuva)	None	None	None	None
Graphical User Interface	Yes	Yes	Yes	Yes
Adds Delete Option to Contextual Menus	Yes	Yes	No	Yes
Algorithms Used	Gutmann, Russian GOST P50739-95, German VSITR, *et al*	Random data writing	Random data, DoD 5220.22M and Gutmann	Random data writing

III. IMPLEMENTATION AND TESTING

This section describes the implementation and testing of the application. The implementation phase involves programming code and graphical user interface development for the application. Meanwhile, the testing phase examines the functionality of E-Raser itself.

A. Application Implementation

Microsoft Visual Studio 2010 was applied as the medium to develop the programming code. A part of the algorithm implemented in Rewrite File module to rewrite the original content of the file with uninformative content can be referred in Figure 2. It should be noted that the scope of document in this study are (.doc) and (.docx) format. At first, the application will set the length of the file stream to 0 bytes and flushes it to the physical file. Flushing the stream ensures that the changes to the stream trickle down to the physical file too. Once the document file is opened and activated, the application will start to write the new content to the existing file.

```
//Writing to the start of a document.
aDoc.Content.InsertBefore("You are not
authorized to read the content of this
file.\r\n");
for (var i = 0; i < 21; i++)
{
// Insert text
var pText = aDoc.Paragraphs.Add();
pText.Format.SpaceAfter = 10f;
pText.Range.Text = String.Format("This is
line #{0}", i);
pText.Range.InsertParagraphAfter();}
//Writing to the end of a document.
aDoc.Content.InsertAfter("\r\n Have a nice
day. :)");}
```

Fig. 2:Programming Code for Class Modify

Figure 3 show the overwriting algorithm to overwrite the segment of drive for the file location with random data generated using cryptographic random number generator. This algorithm is being implemented in Shred File module which will executes the file wiping process and permanently deletes the file.

```
 // Set the files attributes to normal in
 case it's read-only.
 File.SetAttributes(filename,
 FileAttributes.Normal);
 // Calculate the total number of sectors
 in the file.
 double sectors
 =Math.Ceiling(newFileInfo(filename).Lengt
 h / 512.0);
 //Math.Ceiling(x) returns the least
 integer that is greater than or equal to x
 (a decimal or double)
 // Create a dummy-buffer the size
 of a sector.
 byte[] dummyBuffer = new byte[512];
 // Create a cryptographic Random
 Number Generator to create garbage
 data. RNGCryptoServiceProvider csp =
 new
RNGCryptoServiceProvider();
//      Open a FileStream to the file.
FileStream inputStream = new
FileStream(filename, FileMode.Open); for (int
currentPass = 0; currentPass < timesToWrite;
currentPass++){
UpdatePassInfo(currentPass + 1, timesToWrite);
//      Go to the beginning of the stream
inputStream.Position = 0;
//      Loop all sectors
for (int sectorsWritten = 0; sectorsWritten <
sectors; sectorsWritten++)
{
UpdateSectorInfo(sectorsWritten + 1,
(int)sectors);
//      Fill the dummy-buffer with random data
csp.GetBytes(dummyBuffer);
//      Write it to the stream
inputStream.Write(dummyBuffer, 0,
dummyBuffer.Length);}}
//      Truncate the file to 0 bytes.
inputStream.SetLength(0);
//      Close the stream.
inputStream.Close();
//      Finally, delete the file
File.Delete(filename);
```

Fig. 3 Example of Code for Class Shredder

B. *Application Testing*

Testing phase was divided into two categories namely: 1) Application Functionality Testing and 2) User Acceptance Testing. For application functionality, the testing is conducted to identify whether designed modules in E-Raser can be executed properly without any error or bug (see Table 2).

TABLE 2
APPLICATION FUNCTIONALITY TESTING

MODULE	EXPECTED RESULT	ACTUAL RESULT	STATUS	REMARK
ADD	Users can add word file from their personal computers or removable disk.	Users manage to add file from their personal computers or removable disk.	Successful	Only one file can be added at a time.
REMOVE	Users can remove selected file to change or add new file.	Users can remove selected file.	Successful	
REWRITE	The original content of selected file by user should be replaced with new uninformative content.	The original content of the file is being replaced with new content.	Successful	
SHRED	Shredded file is not supposed to be exists in the file location or recycle bin once it is being shred.	File is not visible at the file location and recycle bin.	Successful	Tested with file recovery tool named Recuva.

Figure 4 shows that majority of the respondents strongly agreed that E-Raser is not a complex application to be execute and can shred file perfectly. It indicates that the modules in E-Raser managed to deliver their purposes very well. However, there are a few of respondents found that E-Raser is a complex application.

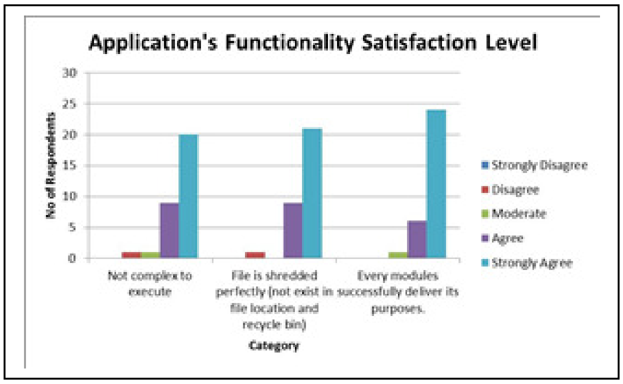


Fig. 4 Respondents' Application Functionality Satisfaction Level

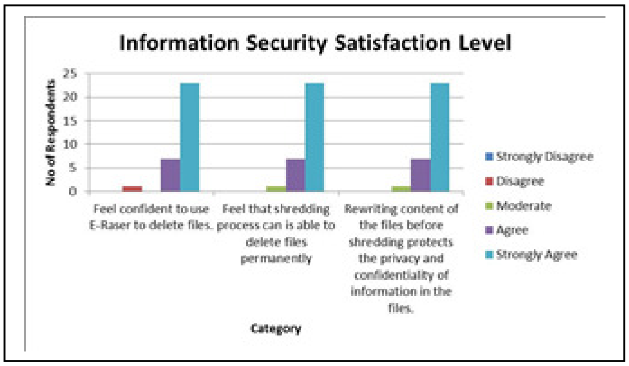


Fig. 5 Respondents' Application Functionality Satisfaction Level

Based on Figure 5, more than 20 respondents felt that shredding process in E-Raser is able to delete files permanently and the rewriting files module protects their information's privacy and confidentiality when deleting the files.

IV. CONCLUSION

This section concludes the overall development, implementation and testing of the application. It includes objectives' achievement, advantages and disadvantages of E-Raser.

The modules designed for E-Raser manage to deliver their purposed successfully. User can add file, remove file, display file before and after rewriting process, rewrite the original content of the file and lastly, shred the file without any remnants left at the file location and recycle bin. It can be conclude that E-Raser has achieved the objectives of its development.

The advantages of E-Raser are listed as follows:

i. E-Raser provides a module named 'Rewrite File' in the application. It replaces the original content of the files with new uninformative content. This module helps to protect the privacy and confidentiality of information in the files.

ii. Users can execute E-Raser application (.exe) file only without having to install Microsoft Visual Studio in their personal computers and it still works successfully.

The disadvantages are listed as follows:

i. User manual can only be accessed if users have internet connection on their computers.

ii. E-Raser only allows users to shred (.doc) and (.docx) files.

iii. Users are limited to delete one file at one time only.

With the development of the application, this project proposed an alternative solution to enhance the information privacy and confidentiality by helping the society in securely deleting their unwanted files beyond recovery more efficiently. It is up to the organization and society to adopt this application according to their needs.

V. Acknowledgement

The authors express appreciation to the FSKTM and Universiti Tun Hussein Onn Malaysia (UTHM). Thanks to anonymous reviewer for valuable comments.

VI. References

[1] Kishi, G. T. (2007). U.S. Patent No. 7,308,543. Washington, DC: U.S. Patent and Trademark Office.

[2] Yasinsac, A., & Manzano, Y. (2001, June). Policies to enhance computer and network forensics. In Proceedings of the 2001 IEEE Workshop on Information Assurance and Security (pp. 289-295).

[3] Quinton, M. (2015, August 4). Old laptop you wiped clean" can still be used by fraudsters. The Sun. Retrieved from https://www.thesun.co.uk

[4] Census data revisited. (n.d). Retrieved October 13,2016, from Kryptel website, http://www.kryptel.com/articles/shredding.php

[5] Wei, M. Y. C., Grupp, L. M., Spada, F. E., & Swanson, S. (2011, February). Reliably Erasing Data from Flash-Based Solid State Drives. In FAST (Vol. 11, pp. 8-8).

[6] Bennison, P. F., & Lasher, P. J. (2005). Data security issues relating to end of life equipment. Journal of ASTM International, 2(4), 1-7.

[7] Forte, D., & Power, R. (2007). A tour through the realm of anti-forensics.Computer Fraud & Security, 2007(6), 18-20.

[8] Ferguson, N., & Schneier, B. (2003). Practical cryptography (Vol. 23). New York: Wiley

[9] Acronis DriveCleanser User's Guide Copyright © Acronis, Inc., 2000-2005

[10] Gutmann, Peter (1996): Secure Deletion of Data from Magnetic and Solid-State Memory, In SSYM"96:Proceedings of the 6th Conference on USENIX Security Symposium, Berkeley, CA, USA, USENIX Association.

[11] Schneier, B.(1995). Applied Cryptography: Protocols, Algorithms, and Source Code in C (Second Edition), John Wiley & Sons, Inc.

[12] Tim, F. (2016, August 27). What is the VSITR Method? Details on the VSITR Data Wipe Method. Retrieved from http://pcsupport.about.com/od/termsv/g/vsitr.htm.

[13] Census data revisited. (n.d). Retrieved October 13,2016, from Piriform website, https://www.piriform.com/recuva

[14] Kamblea, D. R., Jainb, N., & Deshpandec, S. (2015). Comparison of Digital Forensic tools used in DFAI system. History, 2(6)

[15] Census data revisited. (n.d). Retrieved October 13, 2016, from Eraser website, https://eraser.heidi.ie/

[16] Census data revisited. (n.d). Retrieved October 13, 2016, from Alternate File Shredder website, http://www.alternate-tools.com/pages/c_fileshredder.php

[17] Census data revisited. (n.d). Retrieved October 13,2016, from Freeraser website, http://www.freeraser.com/