ATLANTIS PRESS

Advances in Intelligent Systems Research, volume 154

6th International Conference on Measurement, Instrumentation and Automation (ICMIA 2017)

Research and Implementation of Text Digital Watermarking Based on File Filter Driver

Yunhan Wang[1, a], Zining Yan[2,b] and Liang Kou[3,c]

[1]Harbin Engineering University, Harbin, Heilongjiang, China

[2]Harbin Engineering University, Harbin, Heilongjiang, China

[3]Harbin Engineering University, Harbin, Heilongjiang, China

[a]2224206409@qq.coml, [b]954277966@qq.com, [c]kouliang@hrbeu.edu.cn

Keywords: Digital watermarking; Electronic documents; Data leakage; Minifilter

Abstract. This paper elaborates on adoping both the file filter driver technology and digital watermarking to protect the file and solve the problem of data leakage. Used to develop a filtering driver at the nernel level to encrypt the files, Minifilter technology can offset the lack of robustness of font color algorithm, and achieve the embedding of watermark information in the unformatted text. Based on the analysis of the experimental data, we find that Minifilter has good practicability and effectiveness. Apart from preventing the leakage, it can be used to trace and locate the leakage, giving more comprehensive protection to the digital documents.

Introduction

At present, the research of file security in mainly focused on two aspects: the file encryption and decryption, and the text digital watermarking.The introduction of them will be followed.

(1) Encryption technology is commonly used to protect the file currently.It contains application layer encryption and kernel layer encryption by different encryption layer in the computer. The former one protects the file in a way of restricting the access, and the access control technology is a method of protection based on the user's identity. Once the attackers are permitted access, the file data will be accessed at will, so the protection is fragile[1]. Working on the application layer, HOOK technology can be compartively easy to develop, but the user has to participate in the whole process of encryption and decryption operations, which will result in low efficency. The technology is vulnerable to viruses and other malicious software attacks or message eavesdropping, the attacter can even fake file operations to hook the plaintext data[2], so the security is not very reliable.

As the expansion of the kernel driver function, the file filter driver technology works in the kernel layer, and is famous for the difficulty of development and high technology threshold. However, the security and efficiency are improved resulting from the protection of the kernel layer enhances.The user's participation is never required during the whole process of encryption and decryption which are both transparent. As an ealier file filter driver encryption and decryption system, EFS is proved to be safe, but it can only support the NTFS formatted file system, and is incompatible with such file systems as FAT. Then it is followed by Sfilter[3], a traditional filter driver development model. Compared with the EFS, Sfilter sees great improvements, but it suffers poor reliablity[4] due to the high dependence on the kernel model of data structure during the process of development, and developers have to deal with many details which are unrelated with the kernel driver development, resulting in high comlpexity. The problem is not settled until Microsoft Corp launched Minifilter a new filter driver development model. Based on the traditional filter driver mode, the Minifilter requires that the kernel structure be encapsulated, thus enhancing the efficiency of development and bringing higher stablity[5]. This system utilizes Minifilter to develop filter driver, and coordinate Minifilter with watermarking technology to give a more comprehensive protection to the file system.

(2) As a new technology to protect the file, digital watermarking plays an especially important role in copyright disputes. The research on digital watermarking technology mainly focuses on such areas as image, audio and video[6], where the watermark embedding algorithm has been relatively mature, and such technical products as the watermark trademarks have been put into market. The text itself

carries not too much redundant information, and the structure of the documents is specific, so the watermarking algorithm which matures in the field of picture is not suitable for the text, resulting to the comparative backwardness of the text digital watermarking technology[7]. At present, the text watermarking algorithm mainly aims at format text, and can hardly have a good performance simultaneously in all aspects such as robustness, watermark capacity, security and concealment[8] . As regards unformatted text such as notepad and computer source code, there is no formatted information except some basic information, and unlike the format text, it is difficult to embed watermark in them,so the research on this field is little. Even some thinks that it is unachievable to embed watermark in the unformatted ducoments[9].

The watermark capacity of color-based watermarking algorithm for text ducuments may be several times larger than character-space and line-space, and therefore large-capacity watermarking can be achieved with good performance as regards concealment and security, but this algorithm is not robust due to its deficiency in resistting the simultaneous change of color properties. This paper explores the full use of Minifilter technology to make up for the lack of robustness in this algorithm, and to achieve watermarking in the unformatted text.

Structure of system

The system features a combination of digital watermarking technology and file filter driver technology to provide more comprehensive protection to the ducuments. It also brings into full play the characteristics of Minifilter in order to offset the deficiency of the color-based watermarking algorithm, and embeds watermark in the unformatted text. The overall structure of the system which contains double subsystems,the Minifilter encryption and decryption system, and the text digital watermarking system, is shown in Figure 1.

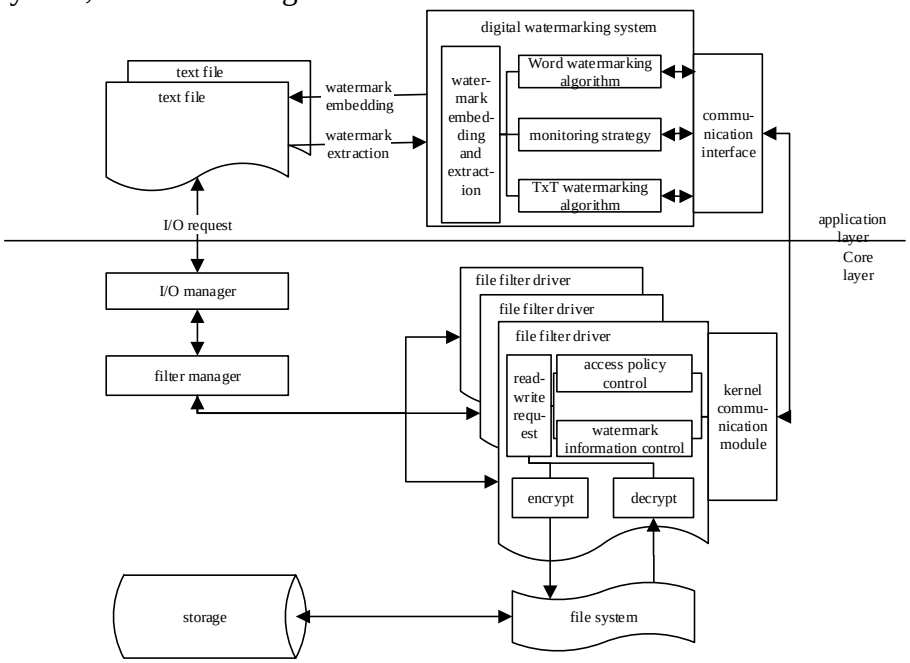

Figure 1 Frame diagram of the system.

Concrete realization of system

(1) Implementation of the Minifilter encryption and decryption system

As a filter layer added between the application layer and the file system, the microfilter driver will intercept and add to the IRP request some additional features with an aim to encrypting and decrypting the target file, before the I/O requests to operate the target file doesn't reach the file system[10]. Thus the systemic function of the file is expanded, and become a part of the kernel. This paper will distinguish encryption process from decryption process, according to the access control policy of the

process file's suffix, and tell encrypted file from decrypted file by adding encryption identification on the head of the file. The problem of data leakage possilbly caused by buffered data can be solved through swap buffer. All the microfilter driver should be registered on the Filter Manager (filter manager), which will select a proper micro filter to deal with the request according to the type of the I/O request.In essence, the micro filters are some callback functions, each group of which is composed of Pre routines and Post routines, and the opeartions of encrypting and decrypting the file are accomplished in these callback functions.The key callback functions contains Create callback function, Read callback function, Write callback function.Discussions on the practical implementatio of Write callback function will be followed.

When the write request is sent from the application layer, it will be wrapped by the I / O Manager into an IRP request with IRP_MJ_ WRITE as the main function number and sent to the kernel layer. The PreWrite preprocessor will judge whether the context data is available, if the answer is not, it suggests that the object file requested to operate can't meet the the access control policy, then the request will be issued directly without any processing. If the answer is yes, the system will assign a cache area for the data, and copy the clear data written by the user from the system cache into the user cache. Because the jurisdiction of writing in system cache is not available for the users, the user has to introduce AES encryption algorithm to encrypt the data in the user cache, ensuring the data in the file system are stored in an encrypted way. Then the PostWrite will alter the file to its original length minus 4KB which is the length of the head, thus the size of the head will be hidden. Specific operating procedures is shown below:

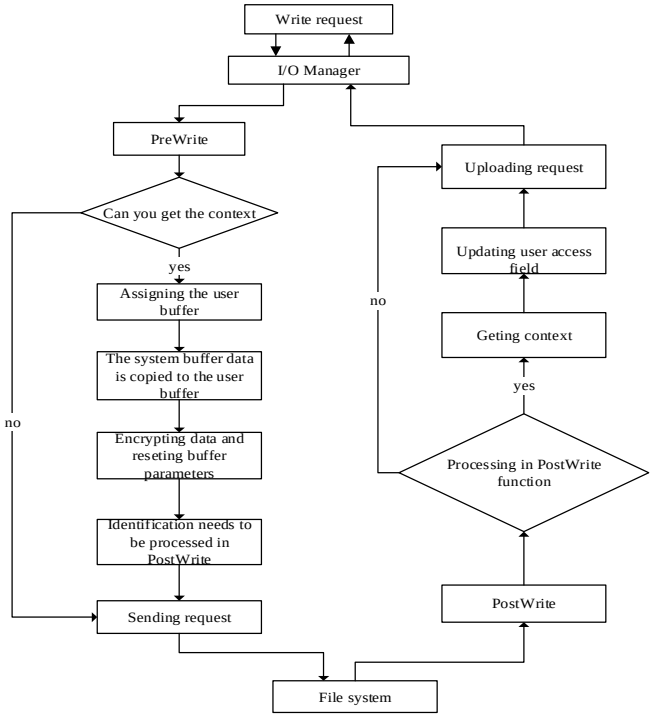

Figure 2 Procedures of processing the Write request callback function.

(2) Text digital watermarking system implementation

1) Embedding of watermarks

Embedding watermark information in word document. Taken by the paper for example, the word ducument accomplishes the technology of color-based text digital watermarking, which also applies to the documents of other formats. Each character in word document has color properties which are composed of three components, Red, Green and Blue,every of which is 8bit-structured[11]. According to human visual system（HVS）[12], it is known that the sensitivity ratio of the human eye to red, green and blue is 65: 33: 2. Based on these theories, this paper tries to embed the watermark information throuth changing the component of low 1 bit, 3 bit, 4 bit respectively corresponding to the fonts color of R, G and B. The watermarking Pseudo code for the single character is as follows.

Table 1 Single character watermark embedding algorithm

Single character watermark embedding algorithm
Watermark embedding position position, one byte of watermark information mark
Result: character at location position has been inserted with mark watermark information
SingleEmbedFunction(long position , Byte mark){
Byte R,G,B; // Initialize the variable
Long oldCo; // original color information variable
ColeVariant varStartPos (position-1);
ColeVariant varEndPos (position); // COM interface locates the text range
WordRange = WordDoc.Range (varStartPos, varEndPos); // Locate the text
WordFont = WordRange.GetFont (); // Select the text feature
OldCo = WordFont.GetColor (); // Get the color
R = GetRColor (oldCo); // Get the red component
G = GetGColor (oldCo); // Get the green component
B = GetBColor (oldCo); // Get the blue component
Byte tR, tB, tG;
TR = (mark & 0x80) >> 7; // watermark bit 7 bits
TG = (mark & 0x70) >> 4; // watermark 4 to 6 digits
TB = (mark & 0x0F); // watermark 0 to 3 bits
WordFont.SetColor (RGB (R | tR, G | tG, B | tB)); // embed watermark information
}

This method not only achieves large-capacity watermarking，but also enjoys good concealment and security. However, its fatal deficiency is that once the attacker makes a uniform modification to the font color, the watermark information will be destroyed completely and unable to be extracted, resulting in poor robustness of this algorithm.In this paper, Minifilter is used to allocate additional space in the head of the file to write digital watermark information which will be hidden in the head, and the file filter driver technology will be used to return a view of applying the original file, thus the aim of watermarking is attained. Because the file seen by the user contains no watermark information, under no conditions can the attacker destroy the watermark information, and this method proves to be more robust. The watermark embedded in the kernel is more costly than that in the application layer, communications with the kernel should be created, and watermark information should be copied into the encryption identification. If the watermark is too large in scale,the copy will take a longer time, and possiblly the storage may cover several sectors. In order to speed up the reading efficency to a certain extent, the paper only allocate 521B, just the size of a sector, to embed the watermark information. If the watermark information embedded is too large, for example, some watermark information used to describe specific features and usage, the summary of the watermark information should be extracted before the watermark can be embedded. Therefore, the deficiency of this method is that only in the head of the file can the watermark be embedded, resulting in small capacity. However, the advantage on the robustness can offset the shortcoming of the color-based

watermarking algorithm. Once it is prompted that the watermark in the carrier text document be destroyed comlpetely and no complete watermark can be available, Minifiter-based methods can be applied to extract the watermark stored on the disk. Althouth there is no attribute property of font color in the unformatted text, and the color-based watermarking algorithm is not applicable to it, Minifiter technology can be applied to embed watermark in the unformatted text. As regards how to imlpement this, there will be deatailed intructions on it later.

The model of embedding watermark information overally in the format text is as follows: first, the watermark information is encrypted by AES, then coded by Hamming code, generating coded watermark information; second, traverse all the files to be watermarked, confirm the embedding parameter 'u' according to length of the carrier documents 'N' and the length of the watermark 'n', and locate the position for embedding the watermark by MD5 positioning algorithm; third, the color-based watermarking algorithm should be applied to embed the watermark for N / n times at most; finally, Minifilter technology is used to implement the embedment of watermark in the kernel, and the watermarked documents can be available. The general model of embedding watermark is shown below.

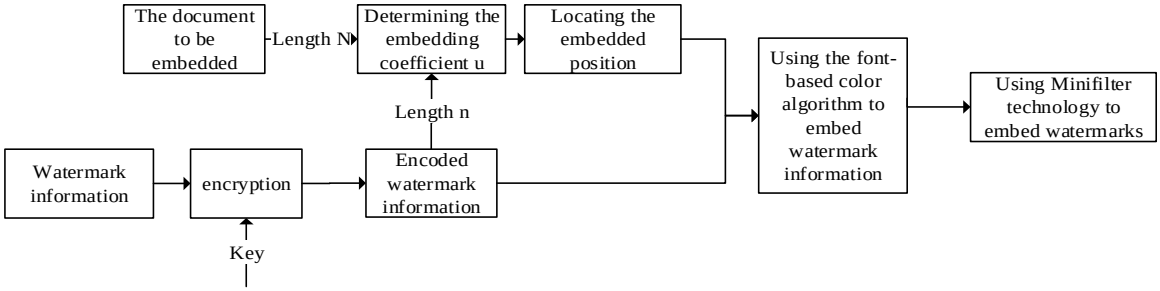

Figure 3 The watermark embedding model of word document.

Embedding watermark information in Txt document. The previous introduction covers the embedding of digital watermark in format text, which can be controlled by the COM technology, such as the color. However, as for the unformatted text, such as Txt text, it doesn't work, because almost all the data in format text is stored directly in the memory. Below it is a comparison chart of the txt file expressed in Notepad and memory data. In order to explain the storage model of unformatted text more clearly, only English characters are selected as objects here. The upper part of the chart shows that the file is represented in the form of data in the memory, and the lower half shows the result of the file opened by Notepad.

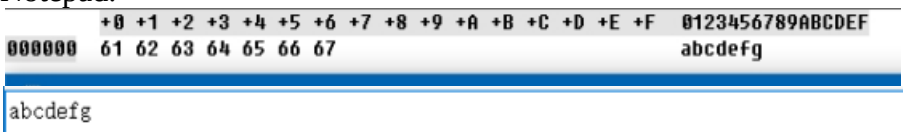

Figure 4 Contrast of representation of txt data.

From the figure above, it is known that what is accessed in the memory is the ASCII code corresponding to the data, such type of text has no other redundant information as the control of character size, character color and so on. Therefore, the strategy of this paper is to use the Minifilter file filter driver to intercept the I/O request to the file, allocate additional space on the head of the file to write digital watermark information, and combine the entire file data with the watermark before written in the disk as a whole, thus the function of watermarking is accomplished. By hiding digital watermark in the head of the text file amd returning a view of applying the original file, the purpose of watermarking is attained. Since the file filter driver can set the read range and offset value of the view returned to the user layer, the user can only see the range of the file filter driver settings which is the original valid data field of the file. However, the watermark information is written in the disk through the Minifilter filter driver, rather than being reflected in the view returned to the user. This method of embedding can guarantee the watermark information absolute advantages in concealment and robustness. Through the communication port, the watermark information generated from the application layer is transfered to the kernel layer, where the embodment of watermark is accomplished. Specific procedures of embedding watermark is shown below:

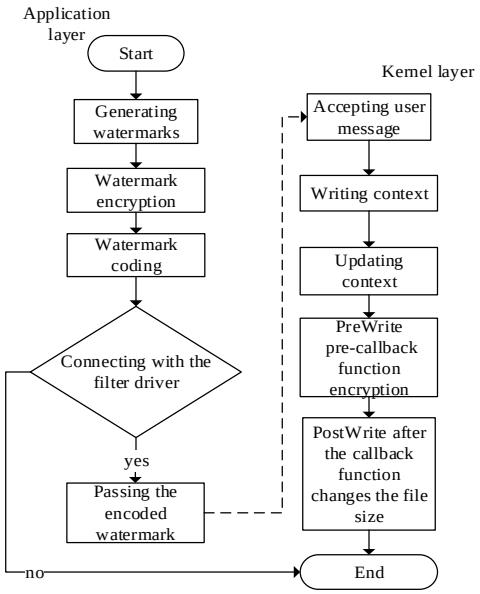

Figure 5 Flow-process diagram of embedding the txt text watermark

2) Embedding of watermarks

Extraction of watermark information from word document. Firstly, the text to be embedded watermarking information should be traversed, and the embedding position of the watermark information is located based on the watermarking algorithm. Then according to the identification of start and end, the single watermark information can be extracted before a judgement based on the Hamming code embedded previously shoule be made, whether or not the watermark information is valid. Because the Hamming code can only verify and correct one bit, if the multidigit watermark information is destroyed which is beyond the scope of its ability, then a error code will be returned which tells the system that this section of the watermark information may be destroyed. So a new round of watermark extraction operation will follow. If problems are continually seen until the last round of watermark information detection, it proves that watermark information embedded several times repeatedly may be destroyed completely, and there will be a prompt that watermask information will not be extracted in this way. At this time, Minifilter technology will be used to extract the watermask information stored in the disk, the extracting process will be introduced later. If the watermask information is found to be correct based on the the Hamming code, or can be verified and corrected according to the error correcting code, the watermask extracted is proved to be valid, operations of restoring the original code information can be exercised on it, such as removing the Hamming code, extracting the identifications of the start and end, getting valid information according to the length. And then valid symbol encrypted by the AES should be read, finally the decryption is operated according to keys to get clear information of the watermark. The model of extracting watermask is shown below.

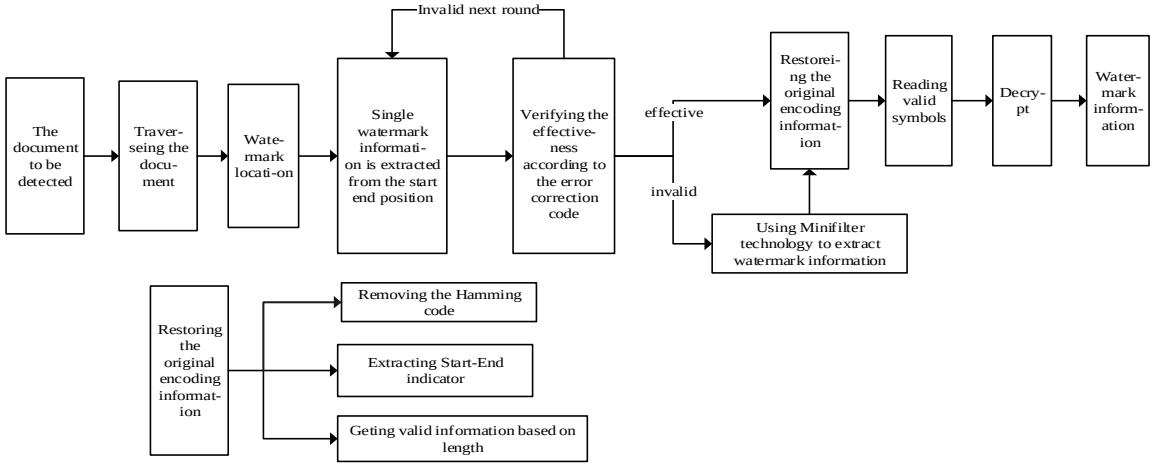

Figure 6 The watermark extracting model of word document

Extraction of watermark information from txt text . Firstly, an watermarked Txt file is opened by the application, the kernel will judge in pre-read process whether it meets the access control policy of the process file's suffix. If the answer is yes, extraction and decryption can be operated, and furthermore, judgement shoule bd made as for whether the watermask information should be extracted. If the user clicks watermark extraction button on the client, it will trigger the process of extracting watermask information.When the watermask information is transfered through communication ports into file driver layer, the context can be obtained in the post-read process, thus the length of the embedded part is achieved. Then the watermark information is separated from the data file, and the watermask information can be available in the system buffer. After this process, through the communication ports between the application layer and the kernel layer, the encrypted watermask information will be transfered to user layer where the information will be decoded and corrected. Then the watermask can be decrypted and displayed. Specific procedures of extracting watermark is shown below:

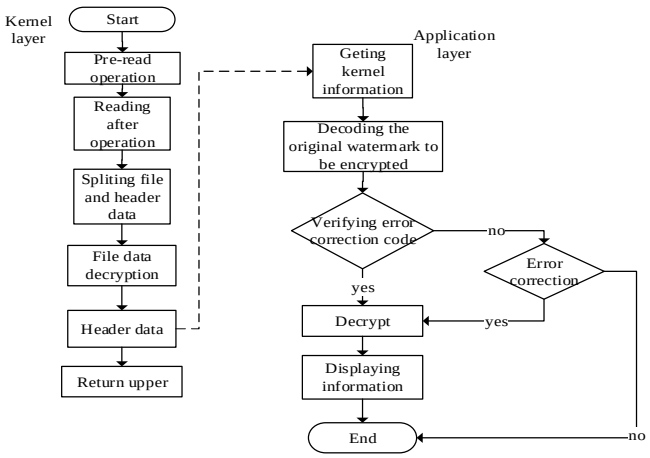

Figure 7 Flow-process diagram of extracting the txt text watermark.

Analysis of results

(1) Test and analysis of microfiltration driver system

The Minifilter should be started before it operates normally. Install the engine2.sys driver through the.INF file, start the Minifilter filter driver through the net start Engine2 command, and then the client interface can be opened.

For example, creating a word document named by testb.doc, and editing it. The authorized process can open the file in clear text, while the encrypted file opened by the unauthorized process shows only meccy code. The figure below will show this.

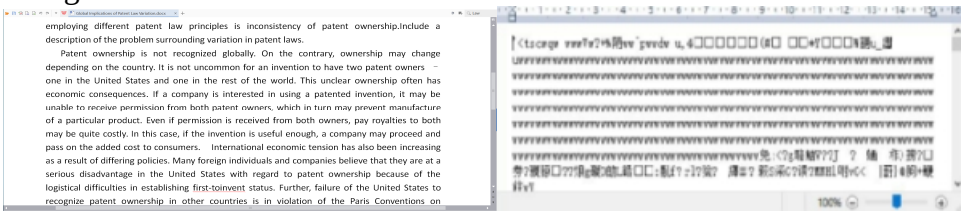

Figure 8 Comparison of files opened by authorized and unauthorized process.

To add the micro filter driver system to the file system, the effect on the efficiency of the entire system will directly affect the user experience. As for this problem, this paper takes the time required to open files with different numbers of characters before and after the installation of micro filter as an example, and a relative program is written. The simulation comparison data tells that the time difference of opening the file before and after the installation of filter driver is only 1 second, if the character number of the file is no more than 300 million, and will provide a better user experience. The diagram bellow shows the comparision of the time required to open the file before and after the installation of filter driver.

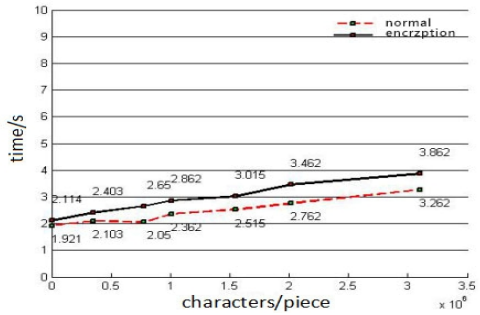

Figure 9 Contrast of time to open the file.

Through analysing the data above, it is concluded that, protection given by the Minifilter to the text data meets the expected requirements in regards to function and performance. Based on the access control policy the encryption and decryption of the file can be implemented. The unauthorized process can't read the protected file, and the computer with no filter driver installed can not open the encrypted file. The whole process of encryption and decryption is almost imperceptible to the user, so the user experience is better.

(2) Test and analysis of font color watermarking algorithm

Figure 12 shows the difference of the document before and after watermark information is embedded. The result of the experiment tells that it is difficult for the human eye to distinguish whether the document is watermarked or not. So the conclusion is that this algorithm has a good concealment.

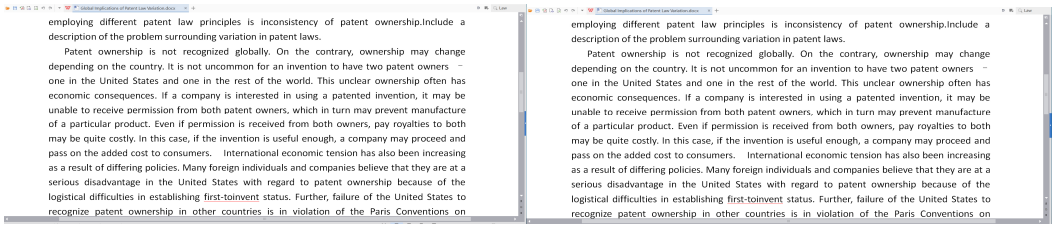

Figure 10 Contrast of files before and after watermarked.

The color-based watermarking algorithm chosen in this paper allows a character to be embedded 8bit watermark information, so the watermark capacity is considerable. Suppose there are 40 characters in a line, a thousand characters will cover 25 lines. Based on the line spacing algorithm, 25bit watermark information can be embedded at most, and the watermark capacity is 25/8B. Based on the character spacing algorithm which stipulates that 1 adjacent position is needed to embed every 1 bit watermark, 500 bit watermark information can be embedded in the 1000 characters, so the watermark capacity is 500/8B. Font size algorithm does not require a reference bit, the watermark capacity is 1000/8B.While based on the watermarking algorithm chosen in this paper, 8000bit watermark information can be embedded, so the watermark capacity is 8000/8B. It can be seen from the simulation diagram of contrast experiment below, that the embedded algorithm has a great advantage as regards watermark capacity.

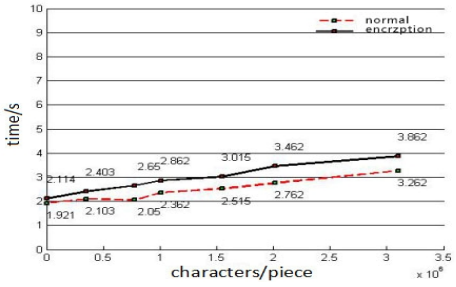

Figure 11 Contrast of watermark capacity.

On one hand, in the watermarking algorithm, the MD5 unidirectional hash function is chosen by the system, ensuring that it is difficult for attacker to locate the correct embedding position of the

watermark. On the other, the algorithm implements AES encryption and coding on the watermark information, so even if attacker proposes to deal with the watermark information, the information will be presented in the form of garbled information, the real and valid watermark can not be restored correctly. In summary, this algorithm enjoys good security. However, there is a fatal weakness in this algorithm, once the font color has been changed simultaneously, the watermark information will be damaged completely, rusulting in poor robustness.

(3) Test and analysis of embedding watermark by Minifilter technology

When Minifilter technology is applied to embed watermark information in the text, a view of original file will be returned to the user layer. What the users see is just normal document information without watermark data embedded, so the concealment is good. Chart 15 shows the contrast of the text content before and after the watermark is embedded. The result is that it is diffcult to distinguish whether the document is watermarked or not, but through the extraction of watermark, we can see that the right one is watermarked.

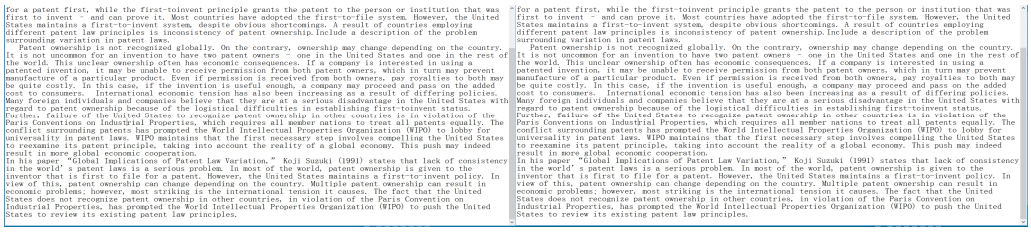

Figure 12 Contrast of file before and after watermarked.

Althouth additional space can be allocated in the memory by the Minifilter, to embed watermark information in the unformated text-Txt, it doesn't mean that the space allocated is limitless. Considering the constraints of the hardware and the efficiency of the watermark extraction, the embedding capacity of the watermark is limited. Because based on this algorithm, the watermark information is embedded in the head of the file, repeated embedding which is seen common in the format text can not be realized in the unformatted text. Besides the copyright information by which the identification can be testified is not big, so a maximum of 512B space can be allocated in the disk to store the watermark data. 512B is just the size of a sector, so all the watermark information can be stored in the range of a sector, rather than being stored across many sectors, resulting in better efficiency of reading. Of course, 512B is only for this system, proper expansion should be introduced according to different requirements of the users.

Since the file filter driver can set the read range and offset value of the view returned to the user layer, the user can only see the range of the file filter driver settings which is the original valid data field of the file. While the watermark is written in the system document, it will not be presented in the view returned to the user. No matter how the data is destroyed, such as adding, deleting, changing, checking, changing the format information, the watermark information can not be damaged, because the edition by the user can not damage the hidden part of the data. Therefore the watermark information is provided proper protection in the kernel layer where the watermark information is also encrypted and encoded, resulting in better security.

Conclusions

In this paper, the combination of file filter driver and text digital watermarking technology is proposed to protect the file in a more comprehensive way. Not only it can be used to prevent leak of document, but also to track the data leakage. Based on the status quo of the research on the text digital watermarking technology, the paper makes full use of Minifilter technology to offset the weak robustness of the font color watermarking algorithm, and realize the embedment of watermark in the unformatted text. To a certain extent, it solves the problems of the existing text digital watermarking algorithms which are aimed at the format text and can not have good performance in watermark capacity, concealment, robustness and other aspects at the same time.

References

[1] BlazIvanc, Borka Jerman Blazic. Information Security Aspects of the Public Safety Data Interoperability Network[C]//Intelligence and Security Informatics Conference (EISIC),2016: 88 - 91.

[2] Gao Shang, Hu Aiqun, Song Yubo. Remote forensics system based on Minifilter[J]. IEEE. 2012,63(6):895-903.

[3] Zhang Fan,Shi Chai-cheng.Windows Driver Deverlopment Internals[M].Beijing:Publishing House of Electronics Industry,2010.

[4] Qiu Shaoming, Tang Guobin, Wang Yunming. Research of File Backup Method Based on Double Cache and Minifilter Driver[J]. International Conference on Advances in Mechanical Engineering and Industrial Informatics. 2015,15(21):677-680.

[5] Cong Zhang, Yumei Wu, Zhengwei Yu. Research and Implementation of File Security Mechanisms Based on File System Filter Driver[C]//Reliability and Maintainability Symposium (RAMS) ,2017: 1 - 6.

[6] Rashida Funke Olanrewaju, Fawwaz Eniola Fajingbesi, Nur Azimah Binti Ishak. Watermarking in Protecting and Validating the Integrity of Digital Information: A Case Study of the Holy Scripture[C]// 2016 6th International Conference on Information and Communication Technology for The Muslim World ,2016:766-770.

[7] R. Patel and P. Bhatt. A review paper on digital watermarking and its techniques[J]. International Journal of Computer Applications. 2015,110(1):10–13.

[8] Reem A. Alotaibi, Lamiaa A. Elrefaei1.Utili:zing Word Space with Pointed and Un-pointed Letters for Arabic Text Watermarking[C]. United Kingdom ,2016:111-116.

[9] Stefano Giovanni Rizzo, Flavio Bertini, Danilo Montesi. Content-preserving Text Watermarking through Unicode Homoglyph Substitution[J]. ACM. 2016,56(36):83-95.

[10] Tan Wen, Yang Xiao, Wang Jian-lei. Windows Kernel Security Programming[M]. Beingjing:BEIJING Publishing House of Electronics Industry,2009.

[11] Chen Xiang.The Design and Implementation of Text Digital Watermarking Algorithm Based on Human Vision Characteristics for Word Documents[D].Hunan:Central South University,2009:21-32.

[12] M. Kaur and K. Mahajan. An existential review on text watermarking techniques[J]. International Journal of Computer Applications. 2015,120(18):101-122.