Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 11 (2013) 288 – 297

Procedia Technology

The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)

An Efficient Adaptive of Transparent Spatial Digital Image Encryption

Osama Ahmed Khashan*, Abdullah Mohd Zin

Center for Software Technology and Management, Universiti Kebangsaan Malaysia, 43600 Bangi Selangor, Malaysia

Abstract

Digital images are important information often contain high confidential and secret data need to be protected. Due to the growth in the adoption of multimedia technologies, many digital images are being stored on the end user devices. Nevertheless, most of the researches mainly focus on protecting these digital images through transmission and protection on the spatial domain is relatively less. However the large size and complex structure of digital images make the computational overhead and processing time needed to carry out full encryption is a major bottleneck especially in real time applications. This paper firstly describes the current spatial data files protection methods based on transparent encryption technology, and then proposes an efficient model for transparent encryption and decryption on the fly for digital image files stored on hard disk. The proposed method takes advantages of partial image encryption using symmetrical ciphers. Furthermore, to reduce the perceivable information for other unencrypted parts, the transformation technique is proposed to shuffle the image blocks and to decrease the correlation among image elements. By using this proposed method, a sufficient level of security can be achieved within a reasonable computational complexity.

Selection and peer-review under responsibility of the Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia.

Keywords: File system filter driver; Spatial digital image; Partial encryption ;

1. Introduction

With the increasing number of laptop and personal computer users, and with the advent prevalence in the domain of multimedia technology, many sensitive image files stored on disk drives and removable storage media.

* Corresponding author: Tel.: +6 03-8921 6173; fax: +6 03- 8925 6732.
E-mail address: o_khashan@yahoo.com

Selection and peer-review under responsibility of the Faculty of Information Science & Technology, Universiti Kebangsaan Malaysia.
doi:10.1016/j.protcy.2013.12.193

More and more potential threats to violate the privacy of these stored contents, some of these threats are unintentional such as physical unauthorized access, device lost or stolen. To restrict these illegal operations, security of various levels is required, and to fulfill such security needs, encryption of digital images is an effective way to prevent information leakage, and to frustrate malicious attacks from unauthorized parties. Whereas, digital images will reside on disk in an encrypted form and accessed only by the legal user. On the other hand, using known encryption algorithms for encrypting a digital image which has different nature from text and has own characteristics such as bulk data capacity, high redundancy and high correlation between pixels. Hence, this demands a long computational time and large amount of processing power is involved since these algorithms are originally designed for textual data and may it not be suitable for image encryption and further have little practical value especially in real time applications.

Several storage encryption methods are available for dynamic encryption and decryption which are generally categorized into hardware based encryption [1, 4], and software based encryption [3, 2, 5]. The encryption mechanism can be performed to the whole single or multiple disk partitions, or by encrypting individual files or directories.

Selecting the appropriate storage encryption option depends on some factors, such as the amount of information inside the disk, needed security level, and the type of surrounded threats. Transparency is another important factor to facilitate an easy to use and effective storage cryptographic solution.

Most of the applied encryption programs that reside on the application space are totally based on user by continuously asking to select the secret files and to supply the encryption keys manually whenever the files are used. However, this extra overhead incurred by the user introduces a dangerous encryption scheme, since the routine of use make the user is easy to forget to encrypt the files and easy to forget the numerous encryption keys, which cause these files at the final state to leak out or left plain on disk.

Many image encryption schemes have been proposed with respect to the approach in construction, which are divided into full and partial image encryption schemes. In the case of partial encryption, most of subsequent researches classify the partial encryption for digital images into spatial and frequency domains by presenting the parts that are look more sensitive and needs protections, and then encrypt such parts either by using decoders or traditional symmetric and asymmetric ciphers.

In this paper, we introduce a model for protecting spatial colored image files base on the file system filter driver technology to provide a transparent encryption and decryption engine for digital images. The proposed model is carried out by expanding the Windows operating system functionalities without modifying the underlying file system structure. It uses an integration of more secure and efficient symmetric and public key encryption algorithms to perform the encryption on a per image file basis. The rest of this paper is organized as follows. In section 2 an overview of transparent storage encryption methods. In section 3, we discuss some research issues in image encryption and describe some partial image encryption techniques. Proposed model for transparent spatial image encryption is presented in section 4, and the conclusion is given in section 5.

2. Transparent storage encryption overview

Transparent storage encryption technology generally categorized into hardware based and software based encryption.

2.1 Hardware based encryption

Encryption can be done inside the storage devices using an embedded encryption processor to provide transparent encryption and decryption to all stored data in different partitions [1]. Transparent encryption also can be done outside the storage devices using an external crypto controller connected to the hard disk [4], such as a Trusted Platform Module (TPM), which is the specification work for the Trusted Computing Group [7]. Although hardware based encryption can provide high performance speed with low system power consuming, but there are some critical limitations related to the flexibility of changing policy settings, easy accessibility of the ciphertext through traffic analysis attack, adding the suffers from the additional cost for these cryptographic devices.

2.2 Software based encryption

The transparent encryption is done with the help of the operating system's kernel to manage, control, and monitor all encryption operations at different levels inside the kernel space. It can be performed as a user space level encryption; middle level is another encryption level inside the kernel; and finally the low level encryption which operates at the lower layer of abstraction under the real file system.

In the user space level, the transparent encryption is performed by extending the capabilities of the file system on user space through a Linux kernel module called FUSE [13, 20]. This FUSE can intercept the system calls come from the application space and direct them to a user space file system library to provide an automatic encryption and decryption without required change on the underlying real file system. This method has suffered from a main drawback related to the performance overhead come from the context switches and several data copies between kernel and user space.

Middle level encryption operates by inserting a cryptographic layer on the level between the user space and the real file system. Cryptographic file system filter driver in Microsoft Windows kernel [12, 17, 22], and Stackable cryptographic file system on Unix-like [14, 15, 34] are examples of a middle level encryption.

Virtual disk encryption is a low level encryption works with blocks of raw data by inserting a cryptographic layer between the real file system and the disk driver as a virtual disk partition. So, all sensitive data need protection should be stored inside this virtual disk to be transparently encrypted and decrypted by an attached virtual disk driver, and then stored as an image file [27, 37]. Nevertheless, this encryption method has seriously effects on the system efficiency due to the huge amount of stored data on the entire virtual disk, which require be encrypted and decrypted each time the user reads or writes a single file.

File system filter driver is a middle kernel level encryption operates by attaching an optional filter driver tailored on the above of the file system driver inside the Windows kernel. The file system filter driver can filter the I/O operations for one or more file systems to add new features or modify the behavior of other drivers before it reach to the lower file system driver [10]. Based on this filter driver, a mandatory transparent encryption, decryption, and key management operations for the spatial data files would be performed online without any more intervention from the user or change on his habits. The structure of file system filter driver and the interaction between the different parts are shown in Fig.1.

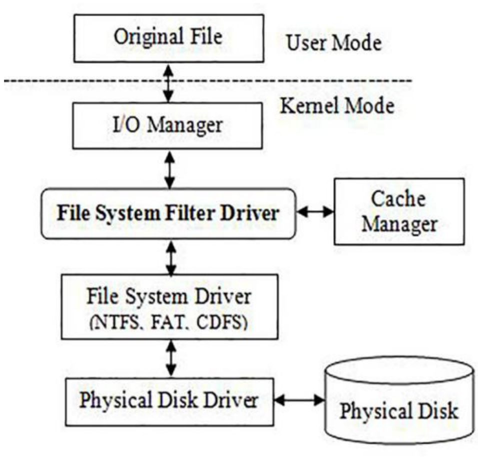

Fig. 1. File System Filter Driver

When a user threads a request to create, read, or write a file stored on the local disk, the request will pass to the I/O Manager to make required process and builds I/O Request Packets (IRPs) before routing them to the required device driver object. Any IRP request is sent to the disk driver with a specific function, the attached filter driver will effectively intercept the request to perform online encryption and decryption operations in a seamless manner.

When a request of writing data into a storage device is received, the filter driver will automatically encrypt a data contained in the IRP before stored on the local disk. Moreover, when the IRP request of read a stored data is received, the filter driver transparently decrypts and sends it in plain form to the application on the user space.

Typically, file system filter driver can be largely employed to add more features with high efficient and reliable way and without any change in operating system functions. In [8] a file system filter driver is used for real time protection and access control for spatial geographical data files. In [30] the digital watermarking together with the file system filter driver are used for spatial geographical data files protection. In [17] file system filter driver is used for sensitive data leakage prevention inside an intranet environment, after authenticate the user identity. In [26] file system filter driver is used to encrypt high secret level files which are evaluated and identified through a Safety Assessment Program inside an intranet network.

3. Encryption of digital image

Digital image is a standardized specification to encode image information as 2D array of pixels saved on disk as a known format to ease interacts with the image file. Each individual pixel has a specific spatial position and own color value with a bit depth. RGB with value of 24 bpp is the most common in color images. Whereas, monochrome and grayscale images usually use a single intensity gradient in the colour space [23].

3.1 Research issues of digital image encryption

There are numbers of research issues that should be considered in developing a digital image encryption technique. The first issue is related to the size of file, since the digital image has huge size, high redundancy, and complex structure that make it different from text in many aspects. Subsequently, using classical ciphers for directly encrypting the entire stored images required much computational power with high computational complexity, which effecting badly on the system efficiency and responding time. The second issue is related to the nature of digital image, since the values of the neighboring pixels are strongly correlated, which means that the value of any given pixel can be easily predicted from the values of its neighboring pixels [28]. The third issue is related to various formats that are used for storing digital images. To save storage space, some of digital image formats allow for data compression to be applied on the image data with two types of data compression methods are lossy and lossless compression. Nevertheless, encrypting a compressed image may cause a small distortion after image decompression and decryption. Moreover, the time needed for compression is significantly high as the time needed for the encryption using either high quality codecs or classical ciphers [6].

3.2 Partial digital image encryption

Partial encryption is a new way for securing image contents by applying encryption to subset of data with a main goal is to reduce the computational overhead and processing time incurred by full encryption, while achieving level of security. As a matter of fact, although a large parts of data are left unencrypted, but still make it difficult to recover the original data without firstly predicting the encrypted parts and secondly decrypting it into the original form. The vulnerability against replacement attacks creates numerous partial encryption techniques to be used in both spatial and frequency domains. These techniques range from simple visual degradation to robust encryption methods.

Partial image encryption gives the priority for encryption to the important parts which often gives important details about an image, whereas the unimportant parts alone do not reveal much information about the image. However, selecting the appropriate partial encryption method is based on the security level that can be provided by that encryption method, and this can be realized on some measuring factors. In case of the classical ciphers, public key encryption algorithms are theoretically easier to attack than symmetric key algorithms [33]. Another measuring factor is related to the encryption key space, since the cryptanalysis of encryption method depends on the secret key recovery [11]. In [18], the indirect measuring factors have been identified to be used for evaluating the security level of the encryption algorithm, such measures are entropy, unicity distance, and guesswork factors.

Entropy originally suggested by Shanon [31], who's measures the average amount of information from a random variable. Therefore, the entropy can be used to calculate the unicity distance, which is an approximation of the minimum amount of ciphertext needed on a ciphertext attack to yield a unique solution. Guesswork has suggested on [32] to measure the expected number of attempts required to correctly guess the output of a random source. [21] Defined other evaluation criteria for partial image encryption methods, since the partial encryption method should be tunable for defining the encrypted parts in a dynamic way, with respect for different requirements and different capabilities. Unpredictability of encrypted parts is another parameter that measures the level of difficulty for key recovery of the encrypted parts. Visual degradation is another subjective parameter to measure the perceptual distortion of the image data to achieve desirable completely disguise, with respect to the plain image.

In spatial domain, many partial image encryption schemes have been proposed range from using affine transformation, space filling curves, SCAN methodology, quadtree structure, and chaotic standard maps. [24] Have proposed a partial encryption scheme for blocks containing edges, which are determined using DCT transformation and encrypted in its bit domain. [25] Proposed a method using Prewitt as a gradient based edge detection to split image data into significant and insignificant parts. The significant parts are totally encrypted using chaos based Arnold and logistic map in spatial domain, whereas the insignificant parts are partially encrypted in a wavelet domain. [9] Have used Prewitt edge detector to divide the image blocks and encrypt the significant parts using a logistic map based on pseudorandom sequence generator. [6] Have proposed a selective bit plane encryption using AES to encrypt the most significant bit plans (MSB). [16] Have used the pseudo random sequence for dividing and encrypting the first four MSB planes as a correlated data. [28] Encrypts the least significant bit plans (LSB) of the image data, since LSBs look random and can add noise to the image. [29] Have used the quadtree method to identify the location and the size of homogeneous regions to be encrypted.

The partial image encryption techniques which are based on lower level of data granularity of image encryption, are either bit plan or pixel based encryption. They become very complex in selection and processing, as well as the high level of computational complexity. Furthermore, such techniques are infeasible to support the needed transparency of image encryption on the spatial domain.

Position transformation is one of image encryption techniques based on reordering the individual bit, pixel or block of an image by rotating it row wise, column wise along the diagonal or anti diagonal directions in order to induce disorderliness in the visibility of the digital image. Further, the transformation methods can decrease the correlation between image elements as well as the amount of redundant information in digital image as low as possible. Different transformation algorithms have been proposed range from using a single transformation manner to a combined from different transformation manners. Another transformation methods use pseudo random number generations [16], whereas large number of keys can be used with reasonably considerable security level. Chaotic transformation maps [19] also can be used to diffuse the values of pixels instead of regions to provide enhanced security level.

4. Proposed model

The overall objective of our work is to design a model that can trade-off between security and performance to provide encryption and decryption for the spatial colored image files stored on disk, with more flexible and dynamic ways. In order to achieve this objective, the encryption and decryption process should be done transparently, not in the user space, but rather in the kernel space using an inserted crypto filter driver attached between the I\O manager and file system driver. Further to emphasize on the following functions: First, the crypto filter driver allows the user to carry on with his work without adding overhead for ciphering or effecting on his normal operations. Second, the crypto filter driver automatically recognizes the digital image file once it detects the process is trying to open or save an image file on the local disk drive. Hence, it will determine to add the image file into the files encryption and decryption list. Thirdly, the crypto filter driver automatically encrypts or decrypts the digital image contents as per image file basis with high secret and seamless manner.

File sharing support and access control are being utilized using the proposed model. Key management is another task for this crypto filter driver to carry on with encryption keys by using an integration of more secure and efficient symmetric and public key encryption algorithms. Thereby, this can reduce losses caused by theft or lost secret keys, and can ensure the security of personal images even when sharing the computer.

The proposed model has structured by set of processes that are shown in a Fig.2. When a request comes from the user space to read or write an image file, the I/O Manager handles the request and builds the required IRP which subsequently turns it directly to the underlying file system driver. Once the IRP pass through the crypto filter driver, it attains the logical structure of the IRP request to recognize whether it's to read or write an image file, and immediately initializes the encryption algorithms and other encryption parameters. In the following, a transparent encryption and decryption will be conducted to the partial blocks of the image file. Further, a transformation operation will be utilized to all of image blocks, before the read or write operation is taking place and without modifying the image file's structure.

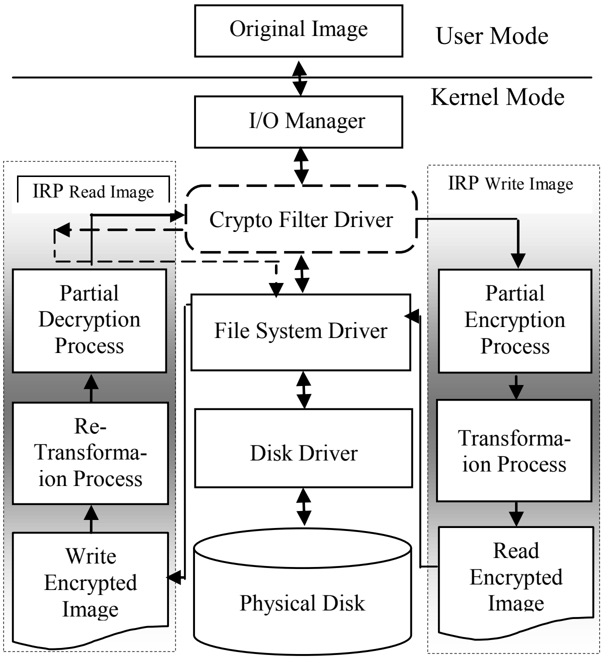

Fig. 2. Proposed Conceptual Model

4.1 Transparent image encryption and decryption processes

When a new image file is created, the crypto filter driver will capture and recognize that the IRP request is to write an image file on the local disk. Thus, set of steps will be performed inside the filter driver. The whole digital image is firstly divided into number of blocks of a fixed block size of 16 bytes. Next, the generated blocks are numbered and represented on a square matrix contains the total number of blocks arranged and indexed in a row wise order. Moreover, the indexed blocks will be classified into correlated and uncorrelated blocks. The correlated blocks are involved sensitive data should be identified for encryption, while the uncorrelated blocks will leave as they are. We have identified the sensitive blocks using edge detection technique since the maximum information of an image is presented in its edges [24, 25]. Therefore, all blocks that containing edges above than a specified threshold will be considered as sensitive and will be marked for later encryption.

Once the sensitive blocks are identified, it will be considered separately for encryption. We picked AES as a fast symmetric encryption algorithm of 128-bit as a default key length. Also, we chose the counter (CTR) as an operation mode to work only with individual blocks. CTR works probably with parallel encryption. It also provides random access for any block cipher without error propagation and without required padding. The plaintext block corresponds to any particular ciphertext block can be recovered independently from other blocks once the corresponding counter is determined. Consequently, different blocks in the digital image will be encrypted using different keys. In order to guarantee of achieving better security level, a uniqueness requirement for a counter block should be satisfied across all image blocks that are encrypted under a given encryption key. After completion of partial blocks encryption, and in order to preserve a sufficient level of security by reducing the perceivable information comes from the unencrypted blocks, we have proposed transformation process to shuffle the encrypted and unencrypted blocks together into new locations.

The transformation algorithm employed for this purpose should be efficient to provide enough randomness and to make the reordering process is quite difficult to find the original locations of the encrypted blocks, and hence difficult cryptanalysis. In this case, the transformation algorithm is applied to the matrix of blocks in order to generate a list of random numbers from *1* to *n*, since *n* denotes the total number of blocks. Here, each number in the list is occurred exactly once to represent a unique block. A general block diagram of the image encryption is shown in Fig.3. It should be noted that the image is totally encrypted by the combination of partial encryption and transformation processes, and it always stored on the hard disk in a ciphered form.

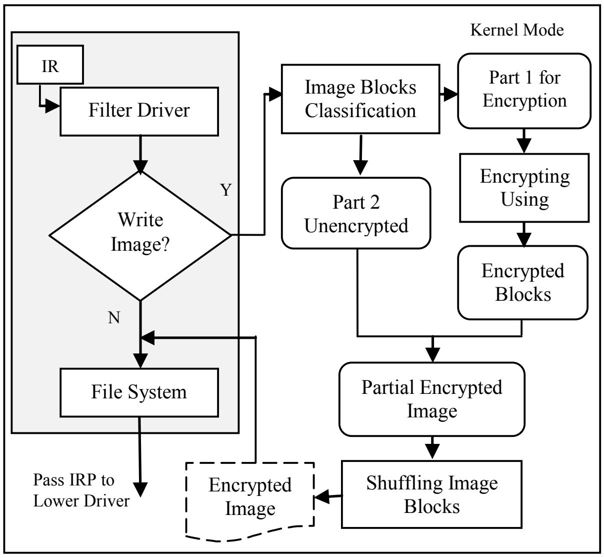

Fig. 3. Block diagram for encryption process

Similarly, read or copy a stored image file is performed transparently in reverse order. When a crypto filter driver recognizes the IRP request is to read a stored image on disk. The crypto filter driver immediately responds once the read operation is completing from the disk, and subsequently to perform the following processes: the encryption parameters that are required to perform the decryption process should be obtained first. In the following, the transformation method is used base on the blocks transformation index key, to retrace the permuted blocks back into original form. After reconstructing the image blocks, the crypto filter driver will use the blocks encryption index key to find out the encrypted blocks. Subsequently decrypts them into plain form to generate the original image, before sending it to the user space. The operational procedures for decryption process are shown in Fig.4.

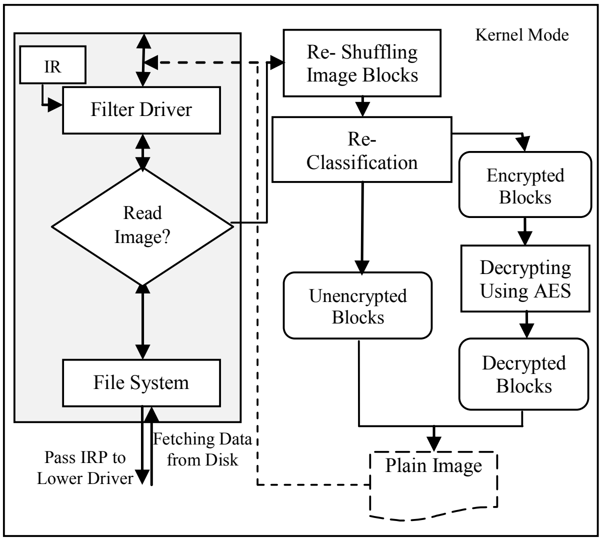

Fig. 4. Block diagram of decryption process

4.2 Key management process

The security of the system is largely dependent on the key management. Key management involves operations of creating, using, and then retaining the encryption keys for longer durations on the storage device. On the other hand, losing or forgetting the encryption keys due to the long time storage means losing access to all data stored on disk. Moreover, storing keys in plain form on the hard disk increase the chances to be stolen or leaked easily. However, in order to enforce the security of the encryption keys, and to reduce the risks come from brute-force attack, the crypto filter driver will be used to protect and manage the keys from leaked or lost. In our design, several keys are used and suited properly to perform the encryption task. All these keys are combined and stored as extended attributes on the header of the image file itself.

- File Encryption Key (FEK) is a symmetric encryption key of size 128-bit, used by the AES to encrypt all sensitive image blocks.
- Blocks Encryption Index Key (EIK) involves the index of all encrypted blocks on the representative matrix which contains all blocks of the image.
- Blocks Transformation Index Key (TIK) involves the index of the permuted blocks which are scrambled according to the transformation algorithm which has been chosen for this matter.

To reduce the risk that comes from storing the key attributes in plain form on the header file, and in order to support a multiuser file sharing utility, in addition to ensure that image file is accessed by the legitimated user only, we have proposed to use a public key encryption algorithm to encrypt the key parameters on the header file. We picked RSA as a public key algorithm. Here, once the image file is being accessed by the legitimate user, the file header is extracted to decrypt the key parameters using the user's private key, and then using them to perform an inverse transformation, and partial blocks decryption respectively. The private key could be stored on a removable smart card connected with the user's machine and loaded automatically once needed. On the other hand, when the image file is saved, the crypto filter driver utilizes the user's login passphrase to encrypt the key parameters before attaching them to the image file header.

5. Conclusion

In this paper, we have proposed a model uses dynamic and transparent encryption to protect digital images stored on disk based on file system filter driver technology. In this proposed model, a new approach for digital image encryption using the combination of partial encryption and transformation methods. Here, the blocks that containing edges act as significant blocks and hence considered for encryption. On the other hand, the transformation method is used to decrease the correlation between image elements and to make the cryptanalysis more difficult. Both methods can give a satisfactory security level with improvement in processing speed. Key management also has been perfectly suited using this proposed scheme. This paper also detailed analyzes the key technologies which can be used to implement the proposed model.

References

[1] Hars L. Discryption: Internal hard-disk encryption for secure storage. IEEE Journals & Magazines 1999; 40:103-105.

[2] Xiaosong Z, Fei L, Ting C, Hua L. Research and application of the transparent data encryption in intranet data leakage prevention. In: IEEE International Conference on Computational Intelligence and Security: 2009; 2:376 –79.

[3] Ma J, Li Z, Li J. A novel secure virtual storage device scheme. In: IEEE International Conference on Intelligent Computing & Intelligent Systems 2010; 1:271-275.

[4] Jun L. Trusted full disk encryption model based on TPM. In: IEEE International Conference on Information Science & Engineering: 2010. p. 1-4.

[5] Pal RK, Sengupta I. Enhancing file data security in Linux operating system by integrating secure file system. In: IEEE Symposium on Computational Intelligence in Cyber Security; April 2009. p. 45-52; 2009.

[6] Podesser M, Schmidt HP, Uhl A. Selective bitplane encryption for secure transmission of image data in mobile environments. In: Proceedings of the 5th Nordic Signal Processing Symposium (NORSIG '02) ;2002.

[7] Kallath D. Trust in trusted computing – the end of security as we know it. Computer Fraud & Security Bulletin; 2005. p. 4-7.

[8] Zhu G, Liangchen Z, Guonian L. The access control technology of spatial data files based on file system filter driver. In: IEEE International Conference on Communication Technology; 2008. p 734-737.

[9] Shekhar S, Srivastava S, Dutta M. An efficient adaptive encryption algorithm for digital images. International Journal of Computer and Electrical Engineering IJCEE 2012; 4(3):380-83.

[10] California Software Labs. I/O file system filter driver for Windows NT. CSWL INC Technical Report: Pleasanton, California; 2002.

[11] Hu G. Study of file encryption and decryption system using security key. In: IEEE International Conference on Computer Engineering and Technology 2010; 7:121-124.

[12] Jie L, Jizhong L. An improved security technique for the terminal sensitive documents. In: 5th IEEE International Conference on Computer Sciences and Convergence Information Technology 2011; Seoul 2(2).

[13] Dzsekijo G, Szeredi M. Filesystem in Userspace. [Online], [Cited 2013 Feb 18]. Available from: URL: http://sourceforge.net/projects/fuse

[14] Zadok E, Badulescu I, Shender A. Cryptfs: A stackable vnode level encryption file system. In: Proceedings of the Annual USENIX Technical Conference; 1999. p. 57-70; 1999.

[15] Halcrow MA. eCryptfs: An enterprise-class cryptographic filesystem for Linux. In: Proceedings of the Linux Symposium 2005; 1:201-218.

[16] Rao YV, Mitra A, Prasanna M. A partial image encryption method with pseudo random sequences. In: ICISS'06 Proceeding of the 2nd International Conference on Information Systems Security; 2006. p. 315-325.

[17] Xiaosong Z, Fei L, Ting C, Hua L. Research and application of the transparent data encryption in intranet data leakage prevention. In: IEEE International Conference on Computational Intelligence and Security 2007; 2:376-379.

[18] Lundin R, Lindskog S, Brunstrom A, Hubner SF. Measuring confidentiality of selectively encrypted messages using guesswork: In: Proceedings of the 3rd Swedish National Computer Networking Workshop (SNCNW '05); 2005. p. 99-102; 2005.

[19] Song T. A novel digital image cryptosystem with chaotic permutation and perturbation mechanism. In: 5th IEEE International Workshop on Chaos-Fractals Theories and Applications (IWCFTA); 2012. p. 202-206.

[20] Gough V. EncFS. Libre Software Meeting. [Online], [Cited 2013 Feb 26]. Available from: URL: http://www.arg0.net/encfsintro.

[21] Massoudi A, Lefebvre F, Vleeschouwer C. Macq B, Quisquater J. Overview on selective encryption of image and video: challenges and perspectives," EURASIP Journal on Information Security; 2008(5).

[22] Li S, Jia X. Research and application of transparent encrypting file system based on windows kernel. In: IEEE International Conference on Computational Intelligence and Software Engineering (CiSE); 2010. p. 1-4.

[23] Tan L. Image file formats. Biomedical Imaging and Intervention Journal 2006; 2(1).

[24] Yekkala AK, Udupa N, Bussa N, Madhavan CE. Lightweight encryption for images. In: IEEE International Conference on Consumer Electronics; 2007. p. 1-2.

[25] Taneja N, Raman B, Gupta I. Combinational domain encryption for still visual data. Multimedia Tools and Applications 2012; 59:775-793.

[26] Zhang P, Wei Z. Application of intelligent transparent encryption model on intranet security. In: IEEE International Conference on Information Theory & Information Security (ICITIS); 2010. p. 268-70.

[27] Ma J, Li Z, Li J. A novel secure virtual storage device scheme. In: IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS) 2010; 1:271-275.

[28] Droogenbroeck MV, Benedett R. Techniques for a selective encryption of uncompressed and compressed images. In: Proceeding of Advanced Concepts for Intelligent Vision Systems (ACIVS); 2002. p. 9-11; 2002.

[29] Cheng H, Li X. On the application of image decomposition to image compression and encryption. In: Proceeding of the IFIP TC6/TC11 International Conference on Communications and Multimedia Security; 1996. p. 116-127;

[30] Zheng L, Feng L, Li Y, Cheng X. Research on digital rights management model for spatial data files. In: 2nd IEEE International Conference on Information Engineering and Computer Science (ICIECS); 2010. p. 1-4.

[31] Shannon CE. Communication theory of secrecy systems. Bell Systems Technical Journal 1949; 28:656-715.

[32] Malone D, Sullivan WG. Guesswork and Entropy. In: IEEE Transaction on Information Theory 2004; 50(3): 525-26.

[33] Garfinkel S, Spafford G. Cryptographic Basics. Web Security, Privacy & Commerce, 2nd ed. CA: O'Reilly Media; 2002. p. 65-71.

[34] Wright PC, Martino MC, Zadok E. NCryptfs: A secure and convenient cryptographic file system. In: Proceedings of the 2003 USENIX Annual Technical Conference; 2003. p. 197-210.