International Journal of Security and Its Applications
Vol.9, No.1 (2015), pp.97-108
http://dx.doi.org/10.14257/ijsia.2015.9.1.11

Cross-access Method for Team Confidential Document Based on Offline Key Management

Haoli Luan[1], Cunxu Wang[2], Zhenliu Zhou[1] and Zheng Yang[1]

1. Shenyang Key Laboratory of Information Security for Power System, Shenyang Institute of Engineering, Shenyang China
2. School of Renewable Energy, Shenyang Institute of Engineering, Shenyang China
Luanhl@sie.edu.cn

Abstract

Existing personal and network encryption application modes are summarized, and a new kind of offline team confidential document application mode is defined and analyzed for satisfying special confidential demand in practical. In this mode, confidential electronic documents are classified by team's role. Team is the smallest unit of key distribution, and access authority is allowed to be delivered between teams. PKI system dependable on Intranet or Internet is forbidden to use in this mode. In response, a cross-access method based on offline key management for this new mode is proposed, algorithms for authorization, encryption, authentication, and decryption in this method are described in detail through an implement example of transparent encryption system using Microsoft Office plug-in technology.

Keywords: *confidential document, cross-access, offline key management, transparent encryption, plug-in*

1. Introduction

For security teams looking to secure file data in storage, there are a plethora of choices available, and each has its specific set of strengths and limitations.

There are diverse electronic document encryption products on the market. Result of investigation on these products shows that these products can be categorized into two types. One is encrypting file system provided by operating system such as transparent encrypting file system in secure linux [1-3] or in windows 2000,windows xp ,or windows 7 [13-15]. Cryptographic file systems typically provide security by encrypting entire files or directories. This has the advantage of simplicity, but does not allow for fine-grained protection of data within very large files. This is not an issue in most general-purpose systems, but can be very important in scientific applications where some but not all of the output data is sensitive or classified. The other is encryption tools developed by the software or hardware manufacturers other than operating system manufacturers, including transparent file encryption based on filter driver [4-5] and file encryption based on virtual disk [6, 20-21]. The first, encryption utilities that encrypt files/folders directly. These utilities encrypt discrete files and/or folders directly, in contrast to utilities that encrypt and store files in volumes (archives, i.e., container files). File-based utilities may operate in batch mode or in on-the-fly mode. Second, virtual-drive encryption utilities create volumes (encrypted containers/archives) which can be mounted in the file-system as virtual drives. These drives can contain both files and folders. The computer's file system can read, write and create documents in real time, directly in clear-text. Virtual-drive utilities operate in on-the-fly mode. Third, Full-drive encryption utilities encrypt entire storage devices, *e.g.*, hard-drives, drive partitions and USB drives.

ISSN: 1738-9976 IJSIA

Some of the utilities in this category can also encrypt the drive that the operating system itself is installed on. Fourth, Client-side encryption utilities for the cloud: A newly emerged category. These utilities encrypt files before they are uploaded to cloud sync/storage locations. The files are encrypted in transit and while at rest in the cloud. Cloud encryption utilities employ various forms of virtualization to present clear-text client-side, and they operate in on-the-fly mode.

The attestation of encryption file system in windows operating system occurs during windows logon, if users logon successfully, they will be authorized automatically to read any data in an encrypted file without any more attestation. So if windows logon password is cracked, all encryption data stored in encrypting file system may leak out. Another flaw lies in that if the file is copied out of NTFS file system, the file will be decrypted and stored as plaintext automatically.

Virtual disk technology is used in Encryption software tools such as Data coffer or file coffer. One shortage of this encryption tools is that user must copy the file into coffer disk manually after finishing editing a file, otherwise the file will not be encrypted. Another shortage of coffer disk is that migrating encrypting files is not allowed.

Transparent encryption technology can decrypt or encrypts file data automatically when user opens or saves file. This technology is achieved by using system file drivers or system API hook. Because need resident in memory and monitoring user's file operation , this technology will consume more CPU or memory sources and lead windows operating system to more unstable, even crash.

There are four kinds of encryption application mode nowadays. The first is personal encryption application mode on single computer such as encryption file system [1- 2, 16- 17] and professional encryption tools [4-6, 11-12, 18]. The second is personal encryption application mode of network storage [7-8]. Common characteristics of these two modes include: key management is simple, no key exchange among users. The third is Intranet interactive encryption application mode and the fourth is Internet interactive encryption application mode. These two modes are all based on complicated key exchange and management mechanism, and there are more security risks about these modes [9-10, 22].

There are others methods and technologies being proposed in recent years [23-36]. A new security architecture for cloud computing platform is proposed by Kawser Wazed Naf [25]. This architecture ensures secure communication system and hiding information from others. AES based file encryption system and asynchronous key system for exchanging information or data is included in this model. This structure can be easily applied with main cloud computing features, *e.g.*, PaaS, SaaS and IaaS. This model also includes onetime password system for user authentication process. A new file encryption algorithm based on two-dimensional logistic and the traditional DES algorithm is discussed by Dongming Chen [27]. It expounds the theory of logistic map and traditional DES algorithm and the principle of encrypting and decrypting. The algorithm generates the chaos sequences by using two-dimensional logistic map, and the chaos sequences are encrypted by DES algorithm. The new key is employed in file encryption. The chaotic characteristics of the algorithm increase the difficulty of deciphering, improve the time efficiency and make the encrypted files more safe. The development and application status in quo of RSA encrypt algorithm are recommended in detail in paper [29], and the feasibility and meaning of RSA used for file encryption is demonstrated. It makes the RSA Encrypt algorithm come true and researches on the algorithm of key-pair generation, designs a fully applied RSA file-encrypt and file-decrypt solution, optimizes the RSA encrypt algorithm in key-pair generating and complete with coding. A method and apparatus for secure transparent backup and encryption of data including compression are proposed by Leif Olov Billstrom [31], elimination of redundant

information, all working integrated whether data is stored locally or shared in networks. When data is shared in networks, several computers may access encrypted objects simultaneously with the same limitations as for non-encrypted objects. The method and apparatus can automatically and invisible take backups and can easily restore any object to the exact content as it existed for a selected point in time using a snapshot capability in combination with the user interface described that has its focus on making the use very easy for the end user. Jiang Tao Geng presents a matrix-based electronic document encryption and decryption algorithm, which relies on a special class of matrices combinatorial problems, the method to improve the security of electronic document system is feasible and effective [32]. A key security mechanism used for enterprise LAN computers and mobile terminals is discussed [33]. The paper focuses on the key mechanisms of the system, including the method of key generation, key distribution, key storage or key backup. The security LAN system is based on the minifilter driver, which achieves transparent encryption and decryption result without interference of users. The mechanism is also applicable for distribution of private keys to the mobile terminals. Protection about the data in mobile storage is discussed in paper [34]. The team proposed an a protection scheme for classified documents, which is based on kernel level file filtering system, implements a set of data protection system for mobile storage medium. First, through the transparent encryption technology, ensure that unauthorized visitors cannot access classified documents; Then through the file redirection technology, make sure that the decrypted data is under the comprehensive protection when authorized users are in the normal using process; Finally, clear the temporary file on the disk and the data trace in the memory, ensure that no using trace left on the computer. This project has realized comprehensive security protection for classified documents, and achieve the goal that classified devices can be used on unclassified computer safely. A new advanced symmetric key cryptographic method called NJJSAA is introduced by A. D. Rangaswamy [35]. A. D. Rangaswamy introduced new bit manipulation method for data encryption and decryption of any file. Based on some symmetric key methods where they have used some randomized key matrix for encryption and decryption methods, in the present work A. D. Rangaswamy has used a bit manipulation method which include bit exchange, right shift and XOR operation on the incoming bits. To exchange bits a randomized key matrix of size (16x16) is used. The present method allows the multiple encryption and multiple decryption. To initiate the encryption process a user has to enter a text-key which may be maximum of 16 characters long. From the text-key the authors have calculated randomization number and the encryption number. A slight change in the text-key will change the randomization number and the encryption number quite a lot. Multiple encryption using bit exchange, bit right shift and XOR operations makes the system very secured. The present method is a block cipher method and it can be applied to encrypt data in sensor network or in mobile network. The advantage of the present method is that one can apply this method on top of any other standard algorithm such as DES, AES or RSA. The method is suitable to encrypt any large or small file. There is a scope to further enhance the present method of encryption. An efficient RC4 based secure content sniffing for web browsers is presented by Shweta Pandey [37], which supporting textual files (word, pdf, text), web files(.jsp,.php.html) and image files also. In this work, the authors send the text data and image files by applying RC4 encryption algorithm. Data is then partition in several parts for reducing the file overhead and then the data will be sending with the extra bit of 0 and 1 for identifying the attack. The work will secure the encryption mechanism from the traditional file including wide variety of file formats.

Studies have shown that, in practical application, there is also another kind of application mode called offline team confidential document application mode. Requirements about this

application mode are analyzed and defined in this paper, a cross-access method based on offline key management is proposed for this mode, and a new kind of transparent encryption system is implemented based on this method using Microsoft Office plug-in technology.

2. Offline Team Confidential Document Application Mode

When it comes to deploying encryption, it is important to carefully assess how an offering would be integrated and administered. Fundamentally, you want to implement an alternative that solves your current problem, but that does not create a set of new ones. Therefore, it is important to ensure a given platform works in your existing infrastructure. How much time and how many resources need to be invested to deploy the platform in a production environment? How intrusive is the solution? Does the environment need to re-architected? Will existing tools and processes, such as replication and backup, work when the new solution is employed? Can the solution be integrated with standard enterprise tools, such as your existing directory infrastructure, whether Active Directory, LDAP, Kerberos, and so on? How much administrative effort will be required on a day-to-day basis? Is it highly available, featuring capabilities for both disaster recovery and system failures?

In practical, among some branches or departments, there is a kind of demand for confidential electronic document: access authorization to electronic documents are classified by unit of team, there are one or more members in a team, and only members of the team can access those confidential electronic documents. Specially, there are loose coupling relationships among teams. Loose coupling refers to that there exist exchanges to access confidential electronic documents between teams, however, these exchanges are uncertain, temporary, non-recurrent, or are dispensable. A member of a team can authorize access authorization of an electronic document to another team, and this transfer of authorization must be audited afterwords. Exchange of electronic documents may be done through usb-disk, optical-disk or network among teams, but team's key must only be distributed and transferred by offline because PKI system dependable on Intranet or Internet is forbidden to use in this branches or departments. In this mode, team is the smallest unit of authorization, distinguishing from above-mentioned any mode which the smallest authorization unit is individual.

3. Cross-access Method and Offline Key Management

The employment of cryptographic techniques using symmetric keys can be considered as a simple way to protect data. The major problem is that one cannot assure that parties involved in a transaction can meet physically, or even know each other beforehand. In these circumstances, the provision of security services is challenging, and this is particularly true for user authentication.

Public-key systems are based on the fact that every user has two keys, a public and a private one. The public key is accessible by the public and can be requested from a directory service, whereas the private key is kept secret by its owner. The dual way of using the public and private parts of the key pair-encrypting with the public key and decrypting with the private one, or encrypting with the private key and decrypting with the public one-allows to apply asymmetric cryptography for encryption/decryption of data, distribution of shared secret keys, and generation/verification of digital signatures.

How cryptographic keys are managed can be a make-or-break factor in the overall success of an encryption deployment. Why? First, if keys are not secure, the data they protect will not be secure. Second, if keys are lost or inaccessible, the data they were used to encrypt will also be unavailable.

How secure is key material, both when they are in use and when stored and backed up? As mentioned, encrypted data is only as secure as the keys. Therefore, it is important to ensure keys are always highly secure and never available in clear text. When it comes to key administration, a platform should enable organizations to require that such sensitive tasks as key rotation or deletion can only be performed by multiple administrators. Access to keys should also be secured through two-factor authentication.

Losing keys can be an unrecoverable disaster, and this is a particularly significant danger in storage environments where one key can govern the access to an entire volume of encrypted data. An encryption platform needs to offer robust mechanisms to ensure that keys are always available when needed. Key management. Policies and mandates may require data to be re-encrypted with new keys on a regular basis. Does the platform facilitate those efforts? Encryption platforms should offer capabilities for managing keys across their lifecycle, including creation, rotation, backup, and deletion. To streamline the administrative effort required, they should offer automated policies for key rotation.

Over the years, many disparate encryption deployments have sprung up across most organizations. Managing keys separately for different encryption deployments-for example, one for applications, one for storage, one for databases, and so on-results in more complexity and administrative effort, and ultimately makes it difficult to apply security policies consistently across an organization. Given that, it is important to look for a central key management platform that can manage keys from multiple key generators, for example, through the use of the OASIS Key Management Interoperability Protocol (KMIP) standard.

Asymmetric key encryption infrastructure is adopted in this method, and this cross-access method can be further described as following: (1) Team secrecy (or secrecy data) can be exchanged through peer-to-peer network, network servers, or portable storage device, (2) Asymmetric key is distributed to team, that is, each member of a team has same public key and private key, (3) Symmetric key generated randomly is used to encrypt secrecy data, public key of team is used to sign and protect symmetric key, private key of team is used to unsign digital signature, (4) All keys, including symmetric key, public key and private key, are stored in hardware smart usb-key for distribution and using, (5) Each smart usb-key has a unique hardware serial number to identify a team member with each other, (6) Offline key management server is set up for use of key distributing, managing and auditing, (7) Smart usb-key, as a carrier of keys for encrypting/decrypting electronic document, must be updated regularly on offline key management server.

Particularly, smart usb-key is used as carrier of key in this method. There are 32KB storage space in smart usb-key. A list of all team's public key are stored in this 32KB space in each smart usb-key. Every record in this list includes two items: identity of team and public key of team, illustrating as Table 1.

Table 1. Structure of List of Team Public Key

Team's identity	Team's public key
Identity of Team A	PKA
Identity of Team B	PKB
Identity of Team C	PKC
Identity of Team D	PKD
…	…

Security administrator initializes each smart usb-key on offline key management server. Procedure of initializing smart usb-key is: (1) Generate public and private key pair <PKu ,SKu > for each team randomly, (2) Create a list including all team's public key, and

write this list into each smart usb-key, (3) Security administrator distributes these smart usb-keys to members of each team, and records the correspondence between the member and the smart usb-key. Security administrator is also responsible for others management of key such as withdrawing, updating, and etc.

After getting smart usb-key, members of a team must set Personal Identification Number for his/her own smart usb-key. Personal Identification Number is a password for a user to use his/her own smart usb-key. In application, the member of a team use his/her own smart usb-key to encrypt or decrypt authorized electronic document.

For documents shared in members of the same team, random symmetric key is used to encrypt the document, and team's public key is used to protect the symmetric key. Only members of this same team can decrypt and get the symmetric key using team A's private key to access the encrypted document.

For document exchanged accessing between two different teams, for example, member A of team A authorizes members of team B to access a confidential document, he uses public key of team B to protect the document (as the same, random symmetric key is used to encrypt the document and public key is used to protect the symmetric key), and members of team B can decrypt and get the symmetric key using team B's private key to access the encrypted document.

Algorithms of authorization, authentication, encryption, and decryption in this method are described in detail as following, through an implement example of transparent encryption system using Office plug-in technology.

4. Principle of Office Plug-in Transparent Encryption

Series of Windows operating system are used widely for information processing, almost all of electronic documents stored in personal computers are edited and accessed using Microsoft Office software, for example, word, excel, and *etc*. These electronic documents edited using Microsoft Office software is usually called Office documents. So more and more people focus on Security and confidentiality about Office documents. Office software allows user setting password for a file to protect document's content. Once user has set password for an Office document, if someone try to open or edit this document, he will be required to input correct password. Many facts have proved that the password protection function that the office software provides is weak and there exist many available methods and tools on the network to crack this password.

Transparent encryption technology can decrypt or encrypts file data automatically when user opens or saves file. This technology is achieved by using system file drivers or system API hook. Because need resident in memory and monitoring user's file operation , this technology will consume more CPU or memory sources and lead windows operating system to more unstable, even crash.

Plug-in encryption can solve above problems well. Plug-in is a kind of program which is programmed following pre-defined uniform specification. Plug-in can extends functions of application software without compiling codes of software again. Once uniform specification about a plug-in for some software application is made public, anyone or company can release his/her own plug-in to add new function for that application. There are so many Plug-in products for Microsoft corporation's office software because of its wide usage.

Principle of transparent encryption using plug-in to encrypt/decrypt office document is illustrated as Figure 1.

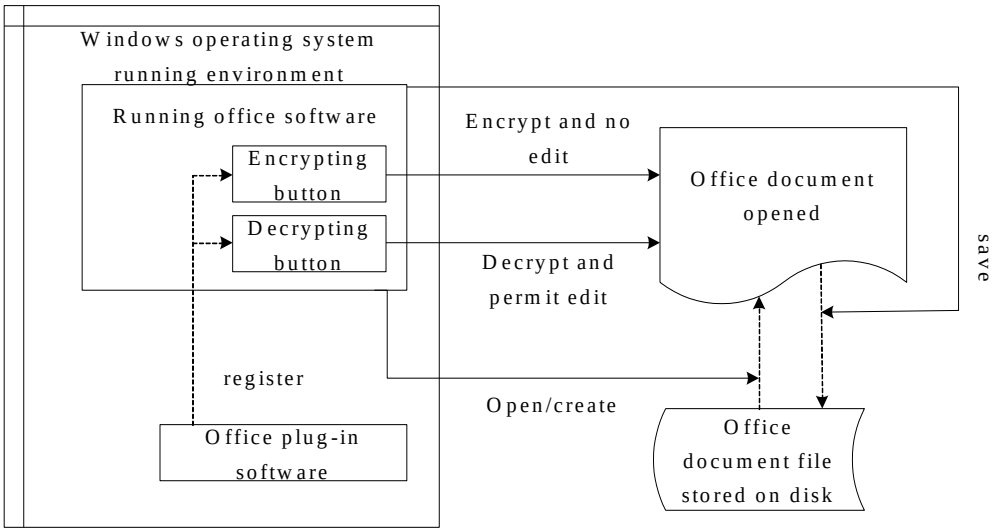


Figure 1. Principle of Office Plug-in Transparent Encryption

When windows system is running, first, run regsvr32 command to register office encryption plug-in, then when office software is running, two button will be added to its tool bar, one is encrypting button, the other is decrypting button.

Once the encrypting button is clicked, data in edit area will be changed from plaintext into ciphertext, and cannot be edited again. Then if the decrypting button is clicked, data in edit area will be changed from ciphertext into plaintext again, and be allowed to edit again. So users can see the result immediately when they do encrypt or decrypt operation. If users save the document with saving operation, the content saved in the file will be as same as the displayed in edit area, that is, plaintext displayed, plaintext saved, ciphertext displayed, ciphertext saved, which is so called 'what you see is what you get'.

The encrypted file is saved as normal office document file format, that is, it can be opened using office software directly, no need to decrypt manually before opening.

Another advantage of office plug-in encryption is that, it is convenient to encrypt partial data chosen by user's intention. For example, user can choose some paragraph or table's content to encrypt, and keep others data displayed as plaintext. This is specially useful when there is only partial data to share in an office document file.

5. Algorithm of Authorization and Encryption

Define following symbols:

KO: symmetric key, used for encrypting data of office document.

SKA: private key of team A, stored in smart usb-key of member A in team A.

SIDA: serial number of smart usb-key of member A.

PKB: public key of team B, stored in list of team public key in smart usb-key of member A.

Sigo: digital signature.

Figure 2 illustrates the member A how to encrypt an Office file transparently and authorize this file to team B.

(1) Member A use Office encrypting plug-in to generate a symmetric key KO randomly.

(2) Office encrypting plug-in encrypts content of the Office document using key KO, then transfers the key to smart usb-key of member A.

(3) Smart usb-key concatenates its own SIDA with the key KO, assuming result is R1, and encrypts result R1 using public key PKB of team B, assuming result is R2, then makes a signature operation to the result R2 using the private key SKA, the signature operation result is Sigo.

(4) Smart usb-key transfers Sigo to Office encrypting plug-in.

(5) Office encrypting plug-in saves the encrypted office document file with the digital signature Sigo together.

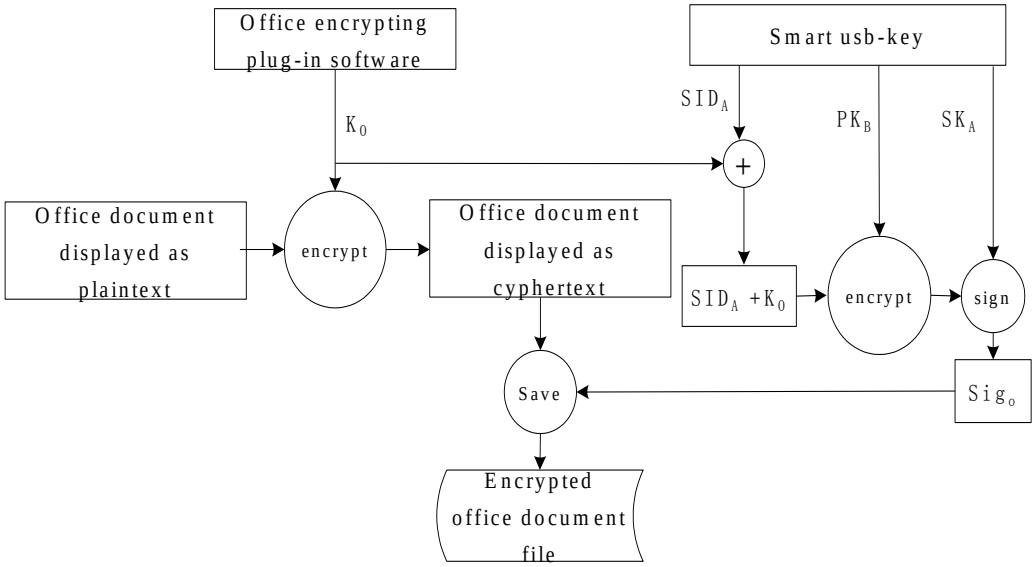


Figure 2. Procedure of Authorization and Encryption

The digital signature Sigo of concatenation result of SIDA and KO encrypted with the public key PKB can only be decrypted using private key SKB by members of team B, that is, only members of team B can decrypt the file, Members of others team except team B can not decrypt the encrypted file. Because signed with the private key SKA, people can know which team has authorized and encrypted this file, and because the signature Sigo including the serial number of smart usb-key SIDA, people can know which member of team A has authorized and encrypted this file. This satisfies the requirement of auditing.

Every office document file is encrypted with different symmetric key, this reduces the risk of batch crack over encrypted Office documents.

6. Algorithm of Authentication and Decryption

Define following symbols additionally:

PKA: public key of team A, stored in list of team public key in smart usb-key of member B in team B.

SKB: private key of team B, stored in smart usb-key of member B.

After getting the encrypted office document file, the member B uses his/her own smart usb-key to authenticate and decrypt the file as following:

(1) Office encrypting plug-in gets the digital signature Sigo from the encrypted office document file, transfers the digital signature Sigo to the smart usb-key of member B.

(2) The smart usb-key receives the digital signature Sigo, searches and retrieves the public key PKA in list of team public key stored in it, and unsigns Sigo with PKA , assuming the result is R2', then decrypts the result R2' using the private key SKB of team B, assuming the result is R1. The last operation is to detach SIDA′ and KO′ from result R1. In this procedure, if any operation of unsign or decrypt fails, procedure will be stopped, and an error message will be sent to the Office encrypting plug-in.

(3) The smart usb-key sends success message and the symmetric key KO′ to the Office encrypting plug-in. The Office encrypting plug-in then decrypts the office document file using this symmetric key.

This algorithm can assures that only members who have been authorized and have correct smart usb-key can get the correct symmetric key and decrypt the encrypted office document file.

7. System Analysis

The security characteristics of this cross-access method and the example system implemented in this paper include:

(1) Key management is offline, simple and effective. This can avoid the security vulnerabilities and complexity of management arosed by using of PKI[9-10].

(2) It can trace and audit which member of which team has authorized a file afterwards.

(3) Symmetric key used to encrypt file is generated randomly, this can reduce the risk of batch crack over encrypted file.

(4) It can be applied securely among different sections or departments whether a share network exists or not. Asymmetric keys are stored securely in smart usb-key, safe and ease to use, and will never appear on any host or network, except for the key management server.

(5) The key management server is offline.

Furthermore, the encrypting plug-in software is convenient to install, and its size is small enough to be stored in smart usb-key. These are especially suitable for secrecy exchange and migration among different sections or departments.

8. Conclusion

A kind of offline team confidential document application mode, which is distinguished from current any application mode in practical, is defined and analyzed for special confidential demand in this paper. In response, a cross-access method based on offline key management for this application mode is described in detail, and a transparent encryption system is implemented using Microsoft Office plug-in to illustrate this method. Document types of Microsoft Office word and excel have been supported in this system. It proves that this method is simple and effective for co-confidential application among many sections or departments which are loose coupling.

Acknowledgements

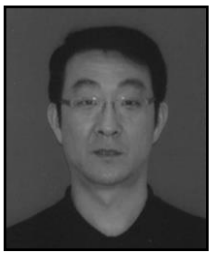
This research is supported by Liaoning Province Industrial Research Project (No. 2012201010), Liaoning BaiQianWan Talents Program (No.2013921051), Nature Science & Foundation of Liaoning Province (No. 2013020124) and the science and technology research program of Liaoning Province Education Administration (No.L2013493, L2013494). Special thanks for above supports very much.

References

[1] M. Blaze, "A cryptographic file system for Unix", Proceedings of the First ACM Conference on Computer and Communication Security, **(1993)** November, pp. 9-15.
[2] P. H. Wei, S. H. Qing and H. F. Liu, "Design and Implementation of a Transparent Cryptographic File System Based on Secure Operating System", Computer science, vol. 30, no. 132, **(2003).**
[3] D. Mazieres, M. Kaminsky, M. F. Kaashoek and E. Witchel, "Seperating key management from file system security", Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP'99), **(1999)** December, pp. 124–139.
[4] M. W. Zhao, R. Mao and R. G. Jiang, "Transparent Encryption File System Model Based on Filter Driver", Computer Engineering, vol. 35, no. 150, **(2009).**
[5] W. Liu and P. Hu, "File Encryption System Design Based on File System Filter Driver", Micro Electronics and Computer, vol. 26, no. 114, **(2009).**
[6] Q. J. Li and M. Gan, „File Encryption Approach Based on Virtual Disk", Computer Engineering and Design, vol. 27, no. 2835, **(2006).**
[7] E. L. Miller, D. D. E. Long, W. E. Freeman and B. C. Reed, "Strong security for network attached storage", Proceedings of the FAST 2002 Conference on File and Storage Technologies, Monterey. CA, **(2002)** January.
[8] B. Reed, E. Chron, R. Burns and D. D. E. Long, "Authenticating network attached storage", IEEE Micro., vol. 20, no. 49, **(2000).**
[9] J. Lopez, R. Oppliger and G. Pernul, "Why public key infrastructures have failed so far", Internet Research, Emerald, vol. 15, no. 544, **(2005).**
[10] B. Burmester and Y. Desmedt, "Is hierarchical public-key certification the next target for hackers?", Communications of the ACM, vol. 47, no. 68, **(2004).**
[11] T. Pionteck, T. Staake and Stiefmeier, "Design of a reconfigurable AES encryption/decryption engine for mobile terminals", Proceedings of the 2004 International Symposium on Circuits and Systems, vol. 2, **(2004).**
[12] M. J. B. Robshaw, "Security estimates for 512-bit RSA Digital Object Identifier", Conference record, Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies. 10.1109/WESCON, vol. 409 **(1995),** 485416.
[13] M. A. Awan and S. H. Khiyal, "Stackably extensible template layer for file system development under windows NT family", IEEE**, (2004).**
[14] E. Zadok, I. Badulesce and A. Shender, "Cryptfs: A stackable vnode level encryption file system", New York: De2 partment Computer Science. Columbia University **(1998).**
[15] G. W. Huang, "Encrypting files system application for Window XP", Science Technology and Engineering, vol. 8, no. 15, **(2008**), 415824160 (Ch).
[16] M. Halcrow, "Ecryptfs", A stacked cryptographic file system, Linux Journal, vol. 156, no. 54258, **(2007).**
[17] C. L. Xing, S. H. Qing and L. P. Li, "The design and implementation of an encrypted file system for Linux", Computer Engineering and Applications, vol. 17, **(2005)**, 1012104 (Ch).
[18] Microsoft Corporation, Filter driver development guide [EB/ OL]. **(2004)**. http: / / download. microsoft. com/ download/ e/ b/ a/ eba1050f-a31d-436b-9281-92cdfeae4b45/ Filter-Driver-Developer-Guide.doc.
[19] X. T. Zhang, "The research on key technologies of VMM2 based system virtual machine and security considerations", Wuhan: School of Computer, Wuhan University, **(2008).**
[20] G. Z. Liao, "Research on mechanism of virtual file system of Linux", Computer Technology and Development, vol. 16, no. 11, **(2006)**, 1142116 (Ch).
[21] T. Garfinkel and M. Rosenblum, "A Virtual Machine Introspection Based Architecture for Intrusion Detection [C]", NDSS, vol. 3, no. 191, **(2003).**
[22] Z. Kotulski and J. Szczepański, "Application of Discrete Chaotic Dynamical Systems in Cryptography--DCC Method", International Journal of Bifurcation and Chaos, vol. 9, no. 1121, **(1999).**
[23] X. B. Li, H. B. Guan and X. Y. Li, "Security performance of file encryption mechanism in Office", Journal of Computer Application, vol. 1, no. 126, **(2010).**
[24] B. J. Zhang, X. H. Li and Y. H. Zheng, "File encryption system design based on identity authentication and filter driver", Journal of Henan Institute of Science and Technology, **(2011).**
[25] K. W. Nafi, T. S. Kar and S. A. Hoque, "A newer user authentication, file encryption and distributed server based cloud computing security architecture", ArXiv preprint arXiv: 1303. 0598, **(2013).**
[26] H. Hon and P. Lin, "Method and apparatus for dynamic generation of symmetric encryption keys and exchange of dynamic symmetric key infrastructure", U. S. Patent 7, 688, 975, **(2010).**
[27] D. Chen, Y. Liu and X. Chen, « A Novel Chaotic Map and DES based File Encryption Algorithm", IJACT: International Journal of Advancements in Computing Technology, vol. 3, no. 198, **(2011)**.
[28] S. Wang and G. Liu, "File encryption and decryption system based on RSA algorithm", Computational and Information Sciences (ICCIS), 2011 International Conference on IEEE, vol. 797, **(2011).**

[29] S. H. T. Bing, "Apply Status Quo of RSA and Its Application in File Encryption", Computer & Telecommunicatio, vol. 6, no. 035, **(2009).**
[30] X. Wang, J. P. Lou and D. D. Li, "Design of security-enhanced file encryption system with authentication", Computer Engineering and Design, vol. 31, no. 1939, **(2010).**
[31] L. O. Billstrom and K. U. Lennartsson, "Intelligent file encryption and secure backup system: U. S. Patent 8, vol. 667, no. 273, **(2014)**, pp. 3-4.
[32] J. T. Geng and Q. Huang, "Research on the File Encryption and Programming Based on Matrix Transformation", Applied Mechanics and Materials", vol. 484, no. 876, **(2014).**
[33] X. Chen, H. Liu and J. Dai, "Security Key Designed for LAN and Mobile Terminal", Software Engineering (WCSE), 2013 Fourth World Congress on IEEE, **(2013)**, pp. 121-124.
[34] J. Chen, Y. Zheng and L. Zhang, "The design and implementation of the kernel level mobile storage medium data protection system", vol. 53, **(2013).**
[35] A. D. Rangaswamy and M. B. Punithkumar, "New Symmetric Key Cryptographic Algorithm Using Combined Bit Manipulation and MSA Encryption Algorithm: NJJSAA Symmetric Key Algorithm", International Journal of Innovative Research and Development, vol. 2, (**2013).**
[36] N. Balkish, A. M. Prasad and V. Suma, "An Efficient Approach to Enhance Data Security in Cloud Using Recursive Blowfish Algorithm", Proceedings of the 48th Annual Convention of Computer Society of India-Springer International Publishing, vol. 1, no. 575, **(2014).**
[37] S. Pandey and A. S. Chauhan, "An Efficient RC4 Based Secure Content Sniffing for Web Browsers Supporting Text and Image Files", Advanced Computing, Networking and Informatics-Volume 2. Springer International Publishing, no. 325, **(2014).**

Authors



Haoli Luan, (1965-), male, the Han nationality, native place: Liaoning, Prof. of Computer Science, Tel: 86-024-31975191, E-mail: Luanhl@sie.edu.cn.

Cunxu Wang, (1968-), male, the Han nationality, native place: Liaoning, Prof. of automatic control, Tel: 86-024-31975311, E-mail: wangcx@sie.edu.cn.

Zhenliu Zhou, (1971-), male, the Han nationality, native place: Hubei, Associate Prof. of Computer Science, Tel: 86-024-31975614, E-mail: zhouzl@sie.edu.cn.

Zheng Yang, (1978-), male, the Han nationality, native place: Liaoning, Associate Prof. of Computer Science, Tel: 86-024-31975193, E-mail: yangzheng@sie.edu.cn.

