Advanced Science and Technology Letters
Vol.73 (FGCN 2014), pp.60-66
http://dx.doi.org/10.14257/astl.2014.73.10

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

Haoli Luan[1], Cunxu Wang[2], Zhenliu Zhou[1], 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[1,2,3] or in windows 2000,windows xp ,or windows 7[13,14,15]. 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].

The attestation of encryption file system in windows operating system occurs during windows logon, if users logon successfully, they will be authorized

ISSN: 2287-1233 ASTL

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] and professional encryption tools[4, 5, 6, 11, 12]. 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].

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.

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 online version of the volume will be available in LNCS Online. Members of institutes subscribing to the Lecture Notes in Computer Science series have access to all the pdfs of all the online publications. Non-subscribers can only read as far as the abstracts. If they try to go beyond this point, they are automatically asked, whether they would like to order the pdf, and are given instructions as to how to do so.

Please note that, if your email address is given in your paper, it will also be included in the meta data of the online version.

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.

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.

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 Fig 1.

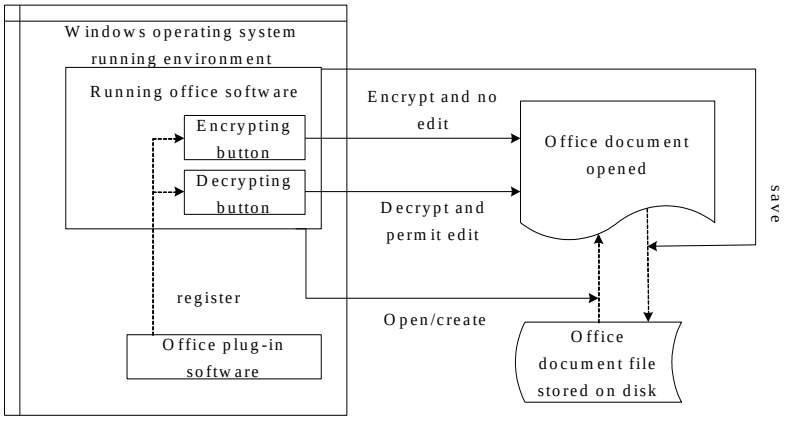

Fig 1. Principle of office plug-in transparent encryption

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.

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.

5 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.

References

1. M. Blaze. A cryptographic file system for Unix. Proceedings of the First ACM Conference on Computer and Communication Security. Nov (1993): 9-15.

2. P. H. Wei, S. H. Qing, H. F. Liu. Design and Implementation of a Transparent Cryptographic File System Based on Secure Operating System. Computer science. 30, 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), Dec (1999): 124–139.
4. M. W. Zhao, R. Mao, R. G. Jiang. Transparent Encryption File System Model Based on Filter Driver. Computer Engineering. 35, 150 (2009)
5. W. Liu, P. Hu. File Encryption System Design Based on File System Filter Driver. Micro Electronics and Computer. 26, 114 (2009)
6. Q. J. Li, M. Gan. File Encryption Approach Based on Virtual Disk. Computer Engineering and Design. 27, 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, Jan (2002)
8. B. Reed, E. Chron, R. Burns and D. D. E. Long. Authenticating network attached storage. IEEE Micro. 20, 49 (2000)
9. J. Lopez, R. Oppliger, G. Pernul. Why public key infrastructures have failed so far. Internet Research, Emerald. 15, 544 (2005)
10. Burmester, Burmester and Y. Desmedt. Is hierarchical public-key certification the next target for hackers? Communications of the ACM. 47, 68 (2004)
11. Pionteck, T. Staake, T. Stiefmeier. Design of a reconfigurable AES encryption/decryption engine for mobile terminals. Proceedings of the 2004 International Symposium on Circuits and Systems. 2, (2004)
12. Robshaw, M. J. B. 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, 409 (1995), 485416
13. M. A. Awan, S. H. Khiyal. Stackably extensible template layer for file system development under windows NT family. IEEE, (2004)
14. E. Zadok, I. Badulesce, 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, (2008), 8 (15): 415824160 (Ch)

