2010 3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)
IPCSIT vol. 53 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V53.No.2.48

A Password-based Encryption Scheme with Symmetric Key Cryptography

Bingjie Huang[+] and Tianjie Cao
School of Computer, China University of Mining and Technology,

Xuzhou, Jiangsu, China

Abstract—This paper proposes a scheme to encrypt the key by using password-based encryption technology in order to ensure the safety of the files, which ensuring the confidentiality of documents while also ensuring the security of the keys; and Blowfish symmetric encryption algorithm system is analyzed in this paper, the combination of the password and encryption algorithm ensures the security of the keys. On this basis, a method of creating self-extracting executable encryption file is described, which makes the password-based encryption independently, and the main method of implementation are given.

Keywords-component; password-based encryption; Blowfish encryption algorithm; self-extracting executable file

1. Introduction

With the popularization of information technology and widely used of internet, the use of computers has been inextricably linked to the modern life. People are more dependent on computers and addicted to efficiency and convenience brought by the network then before, therefore how to ensure data security and transmission reliability becomes increasingly important. With the development of information construction, classified document extensive use electronic means to storage, exchange and management. However, there was a glaring security issues in the electronic application and management of classified documents, so how to effectively prevent leakage in the application and exchange of important documents has being key issues of the application of information technology that need to be urgent resolved[1]. The key of ensure that classified documents and sensitive data security is data encryption.

Currently, there are a lot of file encryption system has been widely used for providing different degrees of protection for classified documents. Key management is the mainly problem considered by security system. One way to protect the key is to save it on a floppy disk or U disk, we use it only when we encrypt or decrypt file; another approach is to use smart cards to store the key, we do the operations of encryption or decryption by connecting a special reading device in the computer[2]. Storing encryption keys with a floppy disk, U disk or smart card has some inconveniences, it needs to external physical media to store the key. In the premise of ensuring safety, more convenient method is to store the key in the computer file system, and use the password-based encryption (PBE) key encryption technology encrypt the key, and then protect them by the local file system permissions. The password must be very strong, so that no one can break it. So firstly the attackers must enter the computer where the key be stored in, and then to break the key.

2. Cryptography

Asymmetric cryptography algorithm is open, so the confidentiality of it is not depends on the encryption system or method but depends on the key. Asymmetric cryptosystem overcomes the fundamental difficulties

[+] Corresponding author.
E-mail address: hbjie1986@163.com

of the traditional password system, it solves the key distribution and message authentication and other issues, it can achieve secure communication, digital signatures and authentication[2]. Although the asymmetric cryptography has high security, strong confidentiality, and it make up the inherent flaws of the symmetric key cryptography, it still has its inevitable shortcomings:

- Keys are generated troublesome, which limited by the prime number generation technique, it is difficult to do once a close.
- The length of key is too large. In order to ensure security, the number of key's bits has been increasing, which making operation costly and especially the speed of Encryption and decryption reduced greatly. It slower than the symmetric cipher algorithm several orders of magnitude. With progress and improving of decomposition method of large numbers, increasing of computer speed and development of computer network (You can use thousands of machines at the same time factoring), this length has been increasing.
- The speed of encryption and decryption is slow. As a result of calculation of Large Number, the speed has been a defect of asymmetric cryptographic algorithm whether it is software or hardware.

There are two types of symmetric encryption algorithm: Block cipher and Stream cipher. Block cipher encrypt a data block (usually 64-bit, some of the algorithm is 128-bit). Stream cipher encrypts the data stream (a bit or a byte). Block cipher can be used to create a stream cipher, and vice versa. If you want to encrypt a single message, block cipher should be used. If it is a steady flow of information, such as socket, the best use is stream ciphers.

The strength of symmetric encryption algorithm mainly decided by the length of the key. The longer the key, the greater the difficulty of cracking. The length of key indicates by bit, usually from 40 bits to 448 bits range. An attacker would need to get the key through the exhaustive method to break an encrypted message. The set of all possible key used by cryptographic algorithm called "key space". There are 240 possible keys for a 40-bit key, for each additional one bit of the key, the security will be double[3].

3. Password-Based Encryption

Symmetric encryption can be used in many areas, it is faster than asymmetric encryption, and mainly suitable for situations of large amounts of data transfer, such as file encryption, network encryption, database encryption. The main disadvantage of symmetric encryption is the symmetry of key, encryption and decryption use the same key, the symmetric encryption key itself is also need to be kept secret[4]. Password-based encryption used a password to encrypt the key, the key mastered by the user will be more safe than by other physical media.

3.1. Blowfish symmetric encryption algorithm Systems Analysis

Blowfish algorithm is a symmetric block cipher cryptographic algorithm of which the execution speed is fast, it is only takes 18 clock cycles when encrypt a byte of data in 32-bit microprocessor. It is a very compact algorithm that can run on the memory less than 5KB. Moreover, Blowfish algorithm have simple structure, and easy to implement[5].

Blowfish is a block cipher algorithm with 64-bit block and a variable key length, the algorithm consists of two parts: key expansion and data encryption. Key expansion transforms the key with the length of up to 448-bit into several sub-keys with a total of 4168 bytes. Data encryption iterate 16 rounds by a simple function, each round consists of permutation related with the key and substitution of key-related and data correlation. All operations are addition and XOR of 32-bit word, the other operation is mapping table four times each round. Blowfish uses a large number of sub-keys, these keys must be pro-computed before the encryption or decryption[6].

1) Initialization of sub-key and S box

The key length of Blowfish algorithm can be change between 32 bit to 448 bit, that is, if a word is 32 bit, key length can be 1Byte to 14Byte. This key is used to produce 18 32-bit sub-keys and 4 S-boxes of 8 × 32, These S boxes 1,024 32-bit items there, are totally 1042 × 32 bits or 4168Byte when add sub-keys.

Keys of Blowfish algorithm can be expressed as an array K:

K1, K2, K3 , K4 , ···, Kj, 1 ≤ j ≤ 14

Sub-keys can be expressed as an array of P:

P1 , P2 , P3 , P4 , ···, P18

Four S boxes (each box with 256 items, each item with 32 bits)can be expressed as:

S[1,0], S[1,1], S[1,2], S[1,3], ···, S[1,255]
S[2,0], S[2,1], S[2,2], S[2,3], ···, S[2,255]
S[3,0], S[3,1], S[3,2], S[3,3], ···, S[3,255]
S[4,0], S[4,1], S[4,2], S[4,3], ···, S[4,255]

Initiative the sub-keys array P and the items of 4 S boxes in turn. The fractional part of the constant π assigned to these items followed by 32-bit. Written in hexadecimal:

P1 = 0x243F6A88, P2 = 0x85A308D3, ···,P18 = 0x8979FB1B,
S[1,0] = 0xD1310BA6,S[1,1] = 0x98DFB5AC, ···, S[4,255] = 0x3AC372E6

P and K will be performed bitwise XOR, and K can be re-used. For the key with maximum length, there are:

P1 = P1 ⊕ K1 , P2 = P2 ⊕ K2 , ···, P14 = P14 ⊕ K14 ,
P15= P15 ⊕ K1 , ···P18 = P18 ⊕ K4

64-bit data blocks of all 0 are encrypted by the current P and S, P1 and P2 replace with the output. Then P1 and P2 which are the output of the last step are encrypted by the current P and S, P3 and P4 replace with the new output, and so on. The operation will be lasting until the P and S have all been updated, the Blowfish algorithm be used in every step are changing. The whole initialing process can be described as follows:

P1 , P2 = Ep,s[0], P3 , P4 = Ep,s[P1] [P2]; ……;
P15, P16 = Ep, s[P13] [P14],
P17 , P18 = Ep,s[P15] [P16],
S[1,0], S[1,1] = Ep,s[P17] [P18], □,
S[4,254], S[4,255] = Ep,s[S4,252] [S4,253]

where Ep, s [Y] meaning the results that Blowfish algorithm encrypted Y with the using of P and S. It needs to perform 521 times encryption process of Blowfish algorithm to produce the final P and S, so the algorithm is not suitable for applications whose key changes frequently. In addition, in order to speed implementation, it's better to keep up P and S which needs only about 4k Byte of storage space, rather than to re-count P and S again when using the algorithm each time.

2) Encryption and decryption process of Blowfish algorithm

The plaintext of Encryption is divided into two halves that are LE0 and RE0 (each is 32-bit). Variable LEi and REi are used to represent left part and right half of the iterative data of i rounds. Encryption process is shown in Figure 3.2.If:

X=X1X2……X64,

Which is 64-bit plaintext that to be encrypted, in which Xi (0,1), $1 \le i \le 64$. Encryption algorithm can be expressed by pseudo-code as follows:

For i = 1 to 16 do
REi = LEi-1 ⊕ Pi;
LEi = F [REi] ⊕ REi-1;
LE17 = RE16 ⊕ P18;
RE17 = LE16 ⊕ P17;

Ciphertext received that is variable LE17 and RE17. The 32-bit input of function F are divided into 4 Byte which denoted a, b, c and d. F can be expressed as:

F[a, b, c, d] = ((S[1,a]+ S[2,b]) ⊕ S[3,c]) + S[4,d]

Here the "+" is the model 232 plus, ⊕ said bitwise XOR. Decryption process is easy to be derived from the encryption algorithm. Correspond with the encryption process, the ciphertext is divided into LD0 and RD0(each 32-bit). Variable LDi and RDi are used to represent left part and right half of the iterative data of i rounds. Blowfish algorithm uses the sub-keys in the reverse order as well as most of the other block cipher, but the encryption and decryption process is in the same order which is different from other algorithms. Decryption algorithm can be expressed by pseudo-code as follows:

For i = 1 to 16 do
RD i = LD i-1 ⊕ P19-i;
LD i = F [RD i] ⊕ RD i-1;
LD17 = RD16 ⊕ P1;

RD17 = LD16 ⊕ P2;

The plaintext received by decryption that is variable LD17 and RD17.

Blowfish encryption algorithm is one of the traditional encryption algorithms which are very difficult to deal with. The S-box of Blowfish algorithm is concerned with key, moreover, some other algorithms such as RC5, the function that execute in each round of iteration is related to the data, but sub-keys and S boxes of Blowfish algorithm are generated from the algorithm itself. The key analysis of the algorithm is very difficult[7].

4. Self Decryping of Files

Self decrypting of files allowing users to operate more simple and convenient carried out the decryption independently. In this paper source files encrypted are attached to a PE executable file ((Encrypted). Exe), and then you can achieve self decrypting of files by running the executable file.

4.1. Portable executable file Format

Portable Executable (PE) file format is an executable file format leaded into by 32-bit Windows operating system. All the executable of win32 (except Vxd and 16-bit Dll) are using PE file format, including the NT kernel mode drivers[8]. In the PE file, executable codes, initialized data, documents resources and relocation information are placed in different sections according to different properties[9].You can modify data of PE files as it sees fit if you grasp the structure of PE files. PE file format is interpreted based on the order of top-down.

The basic structure of PE as shown in Table I:

TABLE I: THE BASIC STRUCTURE OF PE

DOS MZ Header	
DOS Stub	
PE HEADER	***PE Signature***
	PE File Header
	PE Optional Header
Sections table	
Section 1	
Section 2	
Section ···	
SectionN	

The figure is the basic structure of PE file. (Note: The size of DOS MZ Header and some PE header is unchanged; DOS Stub is variable in size.)

4.2. Create self-extracting executable encrypted files

When the operating system is running an executable file, it knows how to do it according to the header of PE. The header of PE is a big table which contains a number of important domains that are used by PE file loader. There are PE file Signature, PE file header and the PE optional header three parts in PE header[10]. It's not allowed to modify the executable itself, so we start from the end of file.

For additional methods and separation methods having the same way, we need to set the format:

TABLE II: FILE ATTRIBUTE

	Constant size
Executable	
File length	***X***
File name	

File content	
Pointer of File name length	***X***
Signature	***X***

Using this format, we will not restrict in the particular size of file name or file content.

Obviously, when we separate files from merged files (executable program+ attached), the file ends is the new end of file, so the size of the end is steady and it will provide us information about the attached file, which is very important. Fixed part of the table is " pointer of file name length " and "signature" section.

In order to implement this task, we need two basic methods:

- attachFile - attach files for self extracting executable file
- detachFile - to obtain attached files, and written to disk

In addition, the above method can be easily implemented by the method:

- checkSignature - Check whether the file attached to it.

Using this method, we can easily create a self-extracting executable file.

4.3. Achieving of Main Method

Here is the implementation process of the main methods.

1) attachFile ()

The method used to achieve attaching of the encrypted file, the file name, file name length and file contents attached to the executable file, the procedure is shown in Figure 1:

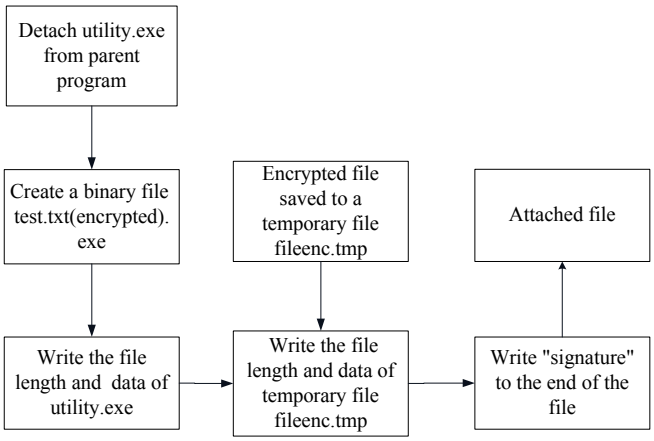

Fig.1. The Procedure of attachfile()

2) detachFile()

The method used to obtain attached file and write it to disk. File name of detached file will be output by detached File parameter. The procedure is shown in Figure 2.

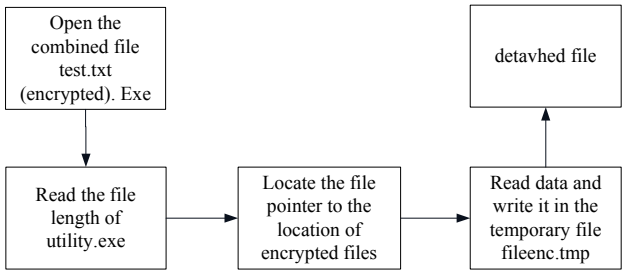

Fig.2. The procedure of detachFile()

3) checkSignature()

In order to create an executable attached file and detached file we need help of checkSignature () method. We can determine the mode of operation in this way. If we have attached file, we are in the detached mode, if we haven't, we are in the attached mode.

5. Conclusions

The world's experts are constantly developing new cryptographic algorithms to protect information security for the security threats of information transmission in the network. Key management is one of the main difficulties of establishing a good password security system, this paper proposes a password-based encryption scheme, which improves the security of key protection in encryption process. And the paper have also achieved the creation of self-extracting executable file, you can decrypt the encrypted file independently.

6. Acknowledgment

This research is supported by the Jiangsu Provincial Natural Science Foundation of China (No.BK2007035).

7. Refrences

[1] Dengguo Feng, "Status and Development of Cryptography of The Foreign and Domestic", Journal of communication, May 2002,pp. 18-26.

[2] Xinhua Wu,Xiangdong An and Ya Su, Life-wide Learning of Encryption and Decryption, Beijing:China Railway Press, August 2005.

[3] Xingfeng Lv,Yu Jiang," Development of the Encryption Technology of Computer Cryptography", Monographic Study, pp.29-32,April 2009.

[4] Zhiyuan Hu, Password Cracking and Encryption Technology, Beijing: Machinery Industry Press, July 2003.

[5] Russell K. Meyers , Ahmed H. Desoky, "An Implementation of the Blowfish Cryptosystem", In Proceedings of Signal Processing and Information Technology, Proc. IEEE, pp. 346-351, 2008.

[6] Allam Mousa, "Data Encryption Performance Based on Blowfish", In Proceedings of Digital Object Identifier ,47th International Symposium ,Proc. IEEE,pp.131-134,2005

[7] Qianchuan Zhong, Qingxin Zhu, "Blowfish Encryption Systems Analysis", Computer Applications, vol.12, pp.2940-2944, 2007.

[8] Xiaojin Guo, Chunlin Shen, "Encryption and Decryption Method of PE Files under 32-bit Windows System", Computer and Digital Engineering,vol.3,pp.51-53,2006.

[9] Mohab U. AbdelHameed1, Mohamed A. Sobh2 and Ayman M. Bahaa Eldin, "Portable Executable Automatic Protection using Dynamic Infection and Code Redirection", In "Proceedings of Computer Engineering & Systems, Proc.IEEE,pp.501-507, 2009.

[10] Renbin Zhang, Gang Li and Zhengfeng Hou, Computer Viruses and Anti-virus Technology, Beijing: Tsinghua University Press, June 2006.