IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.15686

Design and Implementation of Encrypted and Decrypted File System Based on USBKey and Hardware Code

Khanderao S. Kulkarni[1], Pratap N. Shinde[2], Aditya M. Auti[3], Ashok Nagargoje[4], Nirbhay M. Kunale[5], Onkar P. Mante[6]

Professor, Department of Electronics and Telecommunication Engineering,

TSSM's Bhivarabai Sawant College of Engineering and Reasearch, Pune, Maharashtra, India[1]

HOD, Department of Electronics and Telecommunication Engineering,

TSSM's Bhivarabai Sawant College of Engineering and Reasearch, Pune, Maharashtra, India[2]

Department of Electronics and Telecommunication Engineering,

TSSM's Bhivarabai Sawant College of Engineering and Reasearch, Pune, Maharashtra, India[3-6]

Abstract. To protect the privacy of sensitive data, an encrypted and decrypted file system based on USBKey and hardware code is designed and implemented in this paper. This system uses USBKey and hardware code to authenticate a user. We use random key to encrypt file with symmetric encryption algorithm and USBKey to encrypt random key with asymmetric encryption algorithm. At the same time, we use the MD5 algorithm to calculate the hash of file to verify its integrity. Experiment results show that large files can be encrypted and decrypted in a very short time. The system has high efficiency and ensures the security of documents.

INTRODUCTION

Today, transferring files over a network improves the efficiency greatly, but there are a lot of security risks. In order to ensure that important information is not stolen and destroyed, the most practical method is file encryption.

Paper [1] presents a transparent encrypted file system based on secure operation system. There is only cipher text exchange between client and file server, which improves the system security. However, its structure is complex and it is difficult to operate. Paper [2] proposes a net flow encryption system based on HOOK technology. But there are many limitations, for example, the encryption is slow in the application layer, PC is always down by encrypting large files, and it was broken by anti-hook software at times. Paper [3] designs a hard disk encryption card. It can capture the data through the driver program, but it relies on packaged encryption algorithm. In the paper [4], the application program completes the encryption and transports key. The file filter driver decrypts files using the received key. This method involves the system kernel, so the design and debugging is complex. Paper [5] extracts the hard disk serial number to generate the machine code to encrypt, but it is too simple to break.

Based on the comparison of several methods, we designed a system based on USBKey and hardware code to ensure the uniqueness of users and the security of data.

THE DESCRIPTION OF RELATED CONCEPTS

USBKey

USBKey is security hardware with identity authentication, key generation and key storing. The key is stored in USBKey to prevent leakage and tampering; Unified key distribution and management through the management tool can eliminate potential risks. The USBKey's information includes device name, PIN code, public key and private key. We export the public key and make a certificate for each USBKey with a tool, then write them to a database for future use. The private key can be used to decrypt.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.15686

PIN Code

When we use an USBkey to log in, we must enter the PIN code which is used to verify the legitimacy of users. The PIN code amounts to common password of key. If you input the error code, you will not be able to unlock the key and fail to enter the main interface.

Hardware Code

Hardware code is a string generated by some operations based on the hardware of the computer and cannot be easily changed usually. There are much information that can be used to generate hardware code, such as hard disk serial number, mother board ID, CPU serial number and MAC address of network card. This paper proposes the concept to ensure the uniqueness of encryption equipment.

SYSTEM STRUCTURE AND PRINCIPLE

System Structure

This structure includes two aspects: USBKey management and client application. The former includes the addition, deletion and modification of certificates, and the latter includes file choosing, encryption and decryption. The interface of system should be as simple as possible to use.

The system automatically loads the USBKey driver module and the user identity driver module during the normal startup process, and then it loads the transparent encryption and decryption module. After the successful verification, it accesses to the operating system (running security kernel), otherwise exits [6]. The structure of system is shown in FIGURE 1.

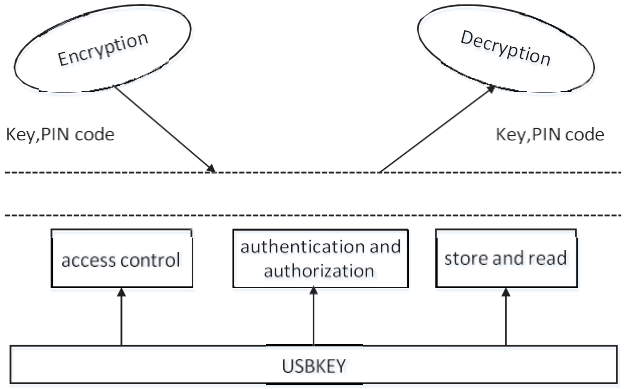

FIGURE 1. The Structure of System

System Principle

3DES Algorithm

The USBKey used in this paper is the type of NT119 which can provides a large space. Users can store more data and design a flexible protection program where the license management function can meet the need to protect more software modules at the same time. It can be used in the case of non-recovery encryption lock to achieve remote registration of encryption keys [7]. It supports MD5 operation, has stable quality and is easy to use. It also provides a 3DES symmetric algorithm.

In 3DES algorithm, 64-bit plain text is encrypted with 56-bit key. However, the key is actually 64-bit, and its 8th, 16th, 24th, 32th, 40th, 48th, 56th and 64th bits are parity check bits, and therefore its effective bits are 56. Plain text is divided into blocks first, each block is 64-bit, and then 64-bit plaintext is sent to initial permutation function for initially permuting plain text; two halves of transformed block are generated by initial permutation and are assumedto be left plain text (L) and right plain text(R), and each left plain text or each right plain text is encrypted for 16 rounds and has its own key as shown in FIGURE 2; finally, left plain text and right plain text are re-connected together, to carry out final permutation on the composed blocks; the result of this process is a 64-bit cipher text [8-10]. The process

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.15686

of encrypting plain text into cipher text is shown in FIGURE 3.

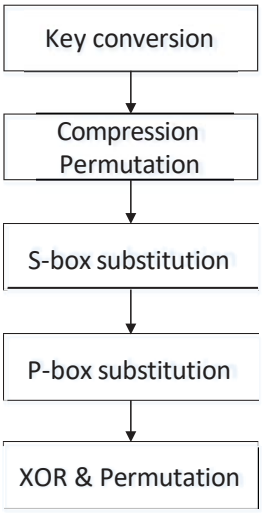

FIGURE 2. Round processing flow

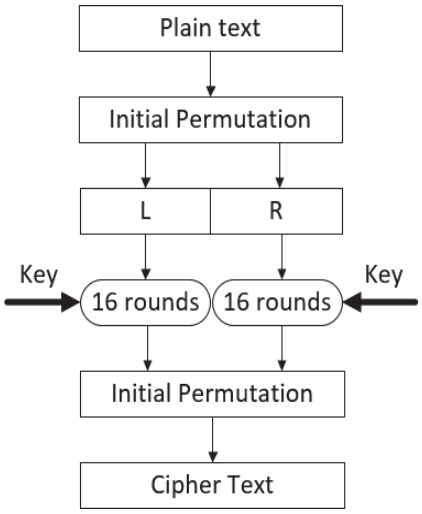

FIGURE 3. 3DES encryption flow

MD5 Algorithm

This algorithm calculates a 128-bit digest for an arbitrary l-bit message.

Let t be the index of a step. Let X_t denote the k[th] 32-bit word of current block and let T_t be a table of 64 32-bit constants. Let CLS denote circular shift left by s bits. The process of single step in MD5 is presented in **FIGURE 4**.

Value of k, s and T_t depend on t, k can be calculated by the following formulas: $k \square t$ in 1st round; $k \square (1 \square 5t) \bmod 16$ in 2nd round; $k \square (5 \square 3t) \bmod 16$ in 3rd round; $k \square (7t) \bmod 16$ in 4th round. s and T_t can be achieved by looking up the table presented in [11-13].

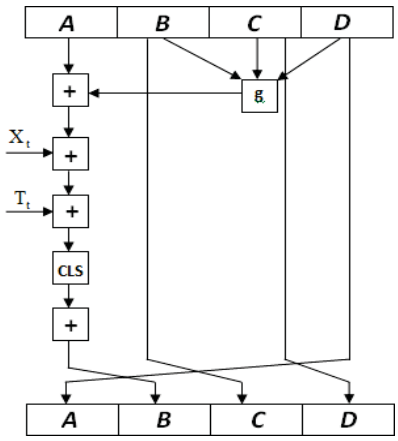

FIGURE 4. Process of single step in MD5

IMPLEMENTATION

Software Design

The system uses a wizard interface. Insert the USBKey and input the PIN code correctly, and then you will enter the system. You can select a file and a certificate, click on the encryption button to complete the encryption process; or choose the file you want to decrypt, click on the decryption button to complete the decryption process [14]. Software flowchart is shown in FIGURE 5.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.15686

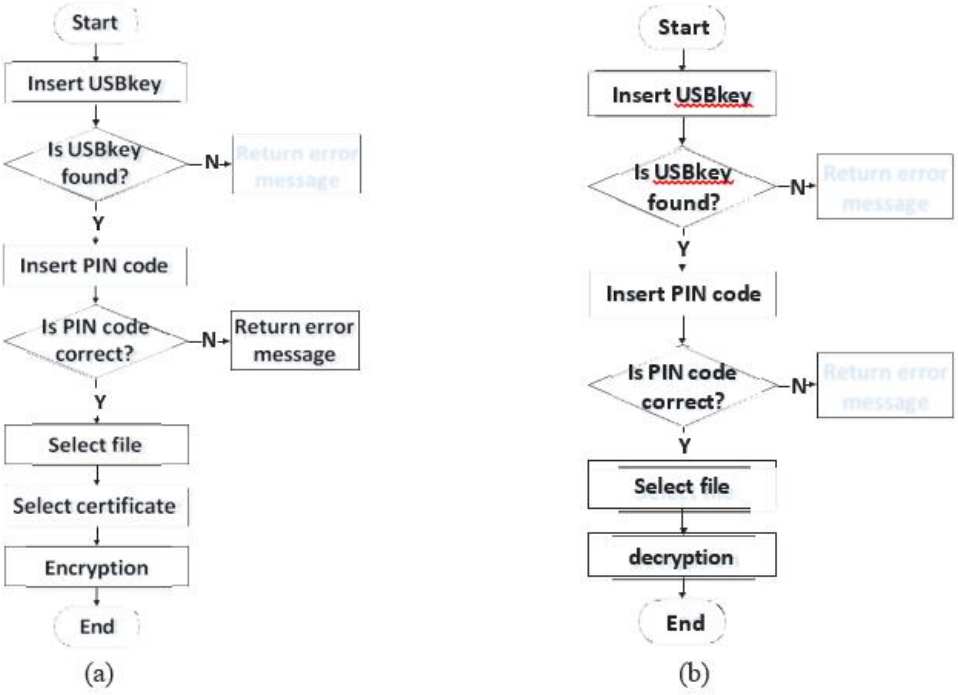

FIGURE 5. Software flowchart

In the process of encryption, we encrypt the old file with random key using 3DES algorithm (symmetric encryption). Then we extract the public key and the corresponding hardware code from the selected certificate to encrypt the random key with SM2 algorithm (non symmetric encryption). Write it to the new file header, then write the contents encrypted. Finally calculate the HASH value of new file with MD5 algorithm and write it in cipher text. In the process of decryption, the system checks the HASH value before decryption to ensure the integrity of the file, verifies whether the current USBKey and hardware code corresponds to the certificate which was chosen when encrypting. No matter which part goes wrong, it will lead to decryption failure. After successful authentication, the system decrypts file header using the private key and hardware code to calculate the random key and then decrypt the file contents with it [15].

Code Interpretation

The programming platform is Microsoft Visual Studio MFC. The system adopts Client/Server structure. The key part of the system is encryption and decryption.The encryption function is:
int ef_encrypt_file(ef_ctx ef, char*, output_file_path, ENCRYPT_PROCESS_CALLBAK process_callback, void* userdata)* The function has four parameters,the first is a structure named ef_ctx, *typedef struct _encrypt_file_config*

```
{
……

    unsigned int create_time;
    unsigned int cipher_suit;
   unsigned int file_count;
    unsigned long long file_size[MAX_KEY_COUNT];
    unsigned char file_name_length[MAX_KEY_COUNT];
    char file_name[256][MAX_KEY_COUNT];
    unsigned char file_md5[16][MAX_KEY_COUNT];

……
}
```

Among them, create_time records file's creation time; Cipher_suit represents encryption algorithm with four bytes, such as symmetric key algorithm, asymmetric key algorithm, hash algorithm; file_count represents the number of files; file_md5 is used to store the file hash value.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.15686

Parameter two is the path of plain text file; parameter three is a callback function, which is mainly used to display the progress bar of encryption in this system.
The decryption function is:
int ef_decrypt_file(ef_ctx ef, char* output_dir_path, DECRYPT_PROCESS_CALLBAK process_callback, void* userdata)*
The parameter two is the path of cipher text file. Other parameters are same with encryption function. The parameter two is the path of cipher text file. Other parameters are same with encryption function.

Functional Test

After the hardware and software are ready, the whole test and application can be carried out.

Login

When you insert the USBKey and open the application, you will have a login page. As shown in FIGURE 6.

FIGURE 6. Login page

Click the “LOGIN” button. When the USBKey is not inserted, the device error message is displayed. Otherwise, the system will automatically fill in the USBKey device serial number. Then input initial PIN code, click the “OK” button to enter the main page; or click the “CANCEL” button to exit.

Encryption
Click the “Browse” button to select a file, and then select a certificate in the left list. Click the "Encrypt" button.
The system encrypts the file automatically and shows the progress at the bottom. As shown in FIGURE 7.

FIGURE 7. Main page

Decryption

Click the “Browse” button to select a file and then click the “Decrypt” button. If the hardware code and USBKey is corresponding the decryption can be completed successfully. Otherwise the error message is shown.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.15686

Management

Click the "Settings" button to manage the certificates. There are three kinds of operation: add, alter and delete.
When you add a certificate, choose a certificate with an extension ".cer", the information such as company, name and serial number will be input automatically; of course, you can also modify them. If the certificate state is disabled, it cannot be used to encrypt and decrypt .The page of add a certificate is shown in FIGURE 8.

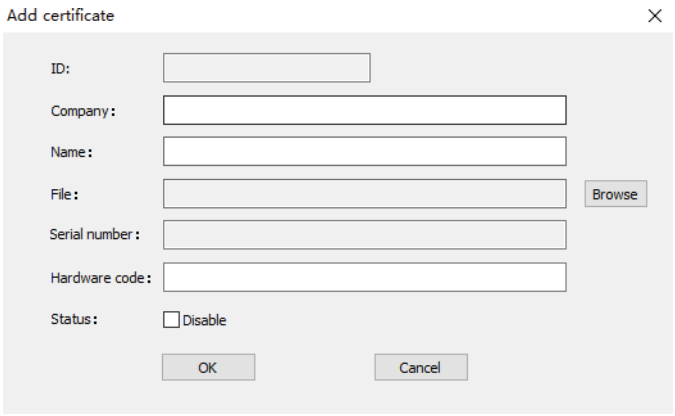

FIGURE 8. The page of add a certificate

COMPARISON AND ANALYSIS

This section compares the system with others in terms of key storage, scalability, ease of use, performance and system size. The advantages of this system are obvious.
Key storage: Compared with the key stored on disk, the system stores the key in USBKey, and the key is managed by a separate tool with high safety performance.
Scalability: The system is less dependent on the kernel and is compatible with all versions of Windows. Because of the structural design and programming, it can provide support for a variety of encryption algorithms.

Ease of use: The system's state description and error message are clear. So it is easy to use.
Performance: We encrypted the same file (1.48M) with our software and a tool named super master of file encryption on the same computer 5 times, and recorded the time. We can see that our system took less time than the other. The comparison is shown in **FIGURE 9**.

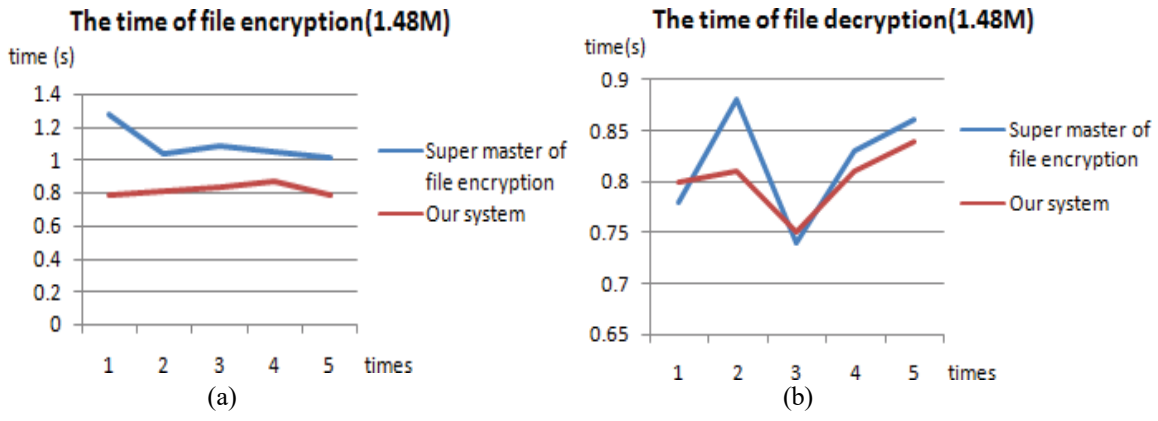

FIGURE 9. The comparison of two systems

Because our system uses hardware to encrypt, it has higher performance compared with other systems which use soft-algorithm. Encryption speed has been significantly improved.
CPU and memory usage: In the process of encryption, we recorded the highest percentage. The comparison is shown in TABLE 1. We can see that our system's CPU usage is lower, but the memory usage is higher.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.15686

TABLE 1. The comparison of CPU and memory usage

	CPU	Memory	Hardware
Our system	26.3%	10.1MB	24.6MB/s
Super master of file encryption	28.8%	3.9MB	9.6MB/s

SUMMARY

This paper designs and implements a file encryption and decryption system based on USBKey and hardware code. The system is introduced from 3 aspects: the related work, system structure and the implementation. Finally, the system is evaluated by comparing it with other systems. Compared with the traditional encryption file system, this system stores the key in USBKey. Only users who have legal USBKey and enter correct PIN code can log in it. In addition, the files are stored in the disk as cipher text; only legitimate users can decrypt and view them.

The system has been proved is flexible and efficient. It ensures the security and privacy of important documents. Of course, Just encrypt the file is not enough, we must combine it with safe transmission. In order to provide a complete set of security mechanisms, we will focus on the file transfer function and continue to improve the stability of the system at next time.

ACKNOWLEDGMENTS

This work was supported by “the Fundamental Research Funds for the Central Universities” (No.2015QN23).

REFERENCES

[1]. WEI Pi-Hui, QING Si-Han, LIU Hai-Feng, “Design and Implementation of a Transparent Cryptographic File System Based on Secure Operation System,” Computer science. 132-135, 2003.
[2]. YANG Fan, WANG Feng, “Research and implementation of transparent encryption system based on HOOK Technology,” Science and technology information development and economy. 213-214, 2007.
[3]. YOU Lin-Ru,BI Yan-Ming,BI Shu-E, “Application of hard disk encryption system in information security,” computer application. 41-43, 2002
[4]. SHAO Yu, XIAO Yun-Shi, ” Design of encryption software based on file system filter driver,” computer application. 1151-1152, 2005.
[5]. LIN Ji-Rui, ”The Research and Design of The Software Protection Program Which is based on Machine Code,” Huazhong University of Science and Technology. 2011.
[6]. SHI Xiang-Wei,LI Zheng,ZHANG Shao-Wu, “Design and implementation of Windows logon system based on USBKey,” Computer Engineering and Design.2483-2485,2008.
[7]. Alanazi H O, Zaidan B B, Zaidan A A, et al. New Comparative Study Between DES, 3DES and AES within Nine Factors [J]. Computer Science, 2010.
[8]. Steinmueller G. A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security [J]. International Journal of Computer Applications, 2013, 67(19):33-38.
[9]. Tsague H D, Nelwamondo F, Msimang N. An Advanced Mutual-authentication Algorithm Using 3DES for Smart Card Systems[C]// Second International Conference on Cloud and Green Computing. IEEE Computer Society, 2012:660-666.
[10]. Ruiying Zhang. Analysis and Comparison on DES and 3-DES Algorithms [A]. Information Engineering Research Institute,USA.Proceedings of 2012 2nd International Conference on Advanced Materials and Information Technology Processing (AMITP 2012) Volume 34[C]. Information Engineering Research Institute, USA: ,2012:8.
[11]. Rivest R. The MD5 Message-Digest Algorithm [M]. RFC Editor, 1992.
[12]. Rfc B. 1321, The MD5 Message-Digest Algorithm [J]. Network Working Group Ietf, 1992.
[13]. Cao D, Han J, Zeng X Y. A reconfigurable and ultra low-cost VLSI implementation of SHA-1 and MD5 functions[C]// International Conference on Asic. IEEE, 2007:862-865.
[14]. Giri D, Sherratt R S, Maitra T, et al. Efficient biometric and password based mutual authentication for consumer USB mass storage devices [J]. IEEE Transactions on Consumer Electronics, 2015, 61(4):491-499.
[15]. Meng Y X, Dong J Y, Yin Y H, et al. Transparent Encryption Technique for Word Documents Based on USB Key in Manufacturing System [J]. Applied Mechanics & Materials, 2012, 252:323-326.