# Application of AES and 3DES Hybrid Encryption Algorithm Based on Optimized S Box in the Data Transmission of Human Resources and Social Security Informatization

Lei Hua [1, *]

[1] Liaoning Provincial Social Insurance Enterprise Service Center, Liaoning, China

* 1464824500@qq.com

**Abstract.** In this paper, a hybrid encryption algorithm based on AES and 3DES is proposed to improve the security of data transmission in human resources and social security informatization. We first analyze the advantages and disadvantages of AES and 3DES, and then elaborate on why the two algorithms are used in combination. Then, this paper discusses the implementation process and mathematical principle of the hybrid algorithm. Finally, we implement the hybrid algorithm in the C + + language and present an application example.

**Keywords:** AES, 3DES, hybrid encryption, data transmission, human resources and social security.

## 1. Foreword

With the rapid development of information technology, the digital and network transmission of human resources and social security information has become the norm. However, this process is faced with serious information security problems. To this end, we need an efficient and secure encryption algorithm to ensure the security of data transmission. The goal of this paper is to design a hybrid encryption algorithm based on AES and 3DES to meet this need.

## 2. Algorithmic Basis

First, we need to understand the encryption and decryption principles of AES and DES.

The AES (Advanced Encryption Standard) is a symmetric encryption algorithm that uses packet encryption to divide the plaintext into several groups, and each group is 16 bytes long. AES has multiple key lengths including 128,192, and 256 bits. In this algorithm, we adopt an AES with a 128-bit key length.

DES (Data Encryption Standard) is also a symmetric encryption algorithm, but it uses a 64-bit packet encryption method with a key length of 56 bits. Due to the low security of DES, this algorithm only uses DES as the first layer encryption algorithm.

## 3. AES and DES Combine the Mathematical Principles of Encryption

The combination of AES (Advanced Encryption Standard) and DES (Data encryption standard) is based on the combined use of substitution and substitution, as well as the extension and confusion of keys.

Replacement and substitution: In AES and DES, data are confused by a series of substitution (Permutation) and substitution (Substitution) operations. In AES, this is mainly reflected in byte substitution (SubBytes), row shift (ShiftRows), column confusion (MixColumns) and other steps. In DES, there are similar operations such as S box replacement and P box replacement. These replacement and substitution operations make the relationship between the input data and the output data very complicated, thus enhancing the security of the algorithm.

Key extension: In both AES and DES, the method of key extension is adopted to generate a series of round keys for a given key. These round keys are used to confuse the data during encryption. The principle of key extension is mainly based on a specific algorithm (such as AES key extension algorithm or DES subkey generation algorithm) to input the key for a series of transformations to generate a series of subkeys.

Confusion: Confusion is an important concept in the encryption algorithm. Its main purpose is to make it complicated with the statistical characteristics of the data to prevent the attackers from cracking the algorithm by analyzing the statistical rules of the data. In AES and DES, confusion is mainly achieved through operations such as substitution and key addition.

When AES and DES are combined, this hybrid encryption method can combine the advantages of both. For example, you can take advantage of AES efficiency for initial encryption, and then use DES for secondary encryption to enhance security. The mathematical principle of this method is based on the combination and superposition of the substitution, substitution, key extension and confusion of the two algorithms, which makes the encryption process more complicated and improves the security of encrypted data.

In general, the mathematical principle of AES and DES combining encryption is based on complex displacement, substitution operations, and key extension and confusion steps, aiming to make the encryption process as complex as possible and difficult to reverse engineer, thus ensuring the security of encrypted data.

The following is the detailed flow of the algorithm for combining AES and DES:

Enter the plaintext and the key to initialize the AES and DES encryption algorithms.

DES encryption to generate the first layer of ciphertext. The specific implementation process is as follows:

Divide the plaintext into 64-digit data blocks, and fill less than 64 digits with 0 at the end.

DES encryption with a 56-bit key to generate 64-bit ciphertext.

Take the generated ciphertext as an input to the second layer of encryption.

The first layer of ciphertext is AES encrypted to generate the final ciphertext. The specific implementation process is as follows:

The first layer of ciphertext was divided into 16 groups, each 16 bytes long.

Use a 128-bit key for AES encryption to generate 16 sets of ciphertext.

The resulting ciphertext was spliced to obtain the final ciphertext.

For decryption, decryption in opposite order. The specific implementation process is as follows:

Decrypt the dense text and get the second layer of plaintext.

The plaintext of the second layer of DES decryption, to obtain the first layer of plaintext. The specific implementation process is as follows:

Divide the second layer of plaintext into 64 data blocks, and less than 64 data will be added 0 at the end.

The 56-bit key for DES decryption yields the first 64-bit plaintext.

The first layer of plaintext is stitched together to obtain the final plaintext.

The following is a representation of the algorithm:

Enter ——> DES Encryption ——> AES Encryption ——> Output

\ /

DES decryption ——> AES decryption ——> output

The following is the formula for the algorithm:

The DES encryption formula:

C = E (K, P) / / C is the ciphertext, K is the 56-bit key, and P is the plaintext

The AES encryption formula:

C = E (K, P) / / C is ciphertext, K is 128-bit key, P is plaintext or ciphertext of the previous layer

The DES decryption formula:

P = D (K, C) / / P is the plaintext, K is the 56-bit key, and C is the ciphertext

The AES decryption formula:

P = D (K, C) / / P is plaintext, K is 128-bit key, C is ciphertext or plaintext of the previous layer

## 4. Algorithm Process

Enter the plaintext and the key to initialize the AES and DES encryption algorithms.

At this stage, we need to input the plaintext and key and initialize the AES and DES encryption algorithms. Both AES and DES can encrypt and decrypt data, but they differ in implementation methods, key length, etc. In this algorithm, we adopt the AES with 128-bit key length and the DES with 56-bit key length.

DES encryption to generate the first layer of ciphertext.

This stage is the first layer of encryption process of the algorithm, using the DES encryption algorithm to encrypt the plaintext and generate the first layer of ciphertext. The specific implementation process is as follows:

(1) Divide the plaintext into 64-digit data blocks, and add less than 64 digits at the end of the figure. This is done to ensure that the length of the data block is consistent, to facilitate encryption and decryption operations.

(2) Use the 56-bit key for DES encryption to generate 64-bit ciphertext. The DES encryption algorithm uses a 64-bit packet encryption method and uses a 56-bit key to encrypt the data. After encryption, you get a 64-bit ciphertext.

(3) Use the generated ciphertext as the input to the second layer of encryption. This step is to use the first layer of ciphertext as an input to the second layer of encryption, in preparation for the next AES encryption.

The first layer of ciphertext is AES encrypted to generate the final ciphertext.

This stage is the second layer of encryption process of the algorithm, using the AES encryption algorithm to encrypt the first layer of ciphertext and generate the final ciphertext. The specific implementation process is as follows:

(1) Divide the first layer of ciphertext into 16 groups, each 16 bytes long. The AES encryption algorithm uses 128-bit packet encryption, so that the first layer of ciphertext is divided into 16 groups, each 16 bytes long.

(2) Use the 128-bit key for AES encryption to generate 16 groups of ciphertext. The AES encryption algorithm uses a 128-bit key to encrypt the data. After encryption, 16 sets of ciphertexts are obtained.

(3) The generated ciphertext is stitched together to get the final ciphertext. This step is by stitching together 16 sets of ciphertext to get the final ciphertext.

For decryption, decryption in opposite order.

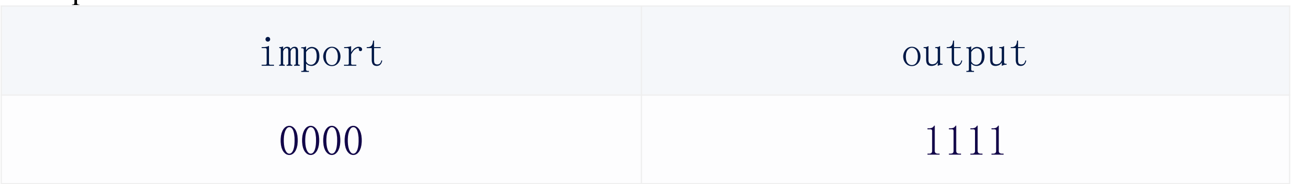
When decryption, it needs to be decryption in the opposite order. The specific implementation process is as follows:

(1) Decrypt the dense text, and get the second layer of plaintext. First, you need to decrypt the ciphertext and get the second layer of plaintext. This step uses the AES decryption algorithm to decrypt the ciphertext.

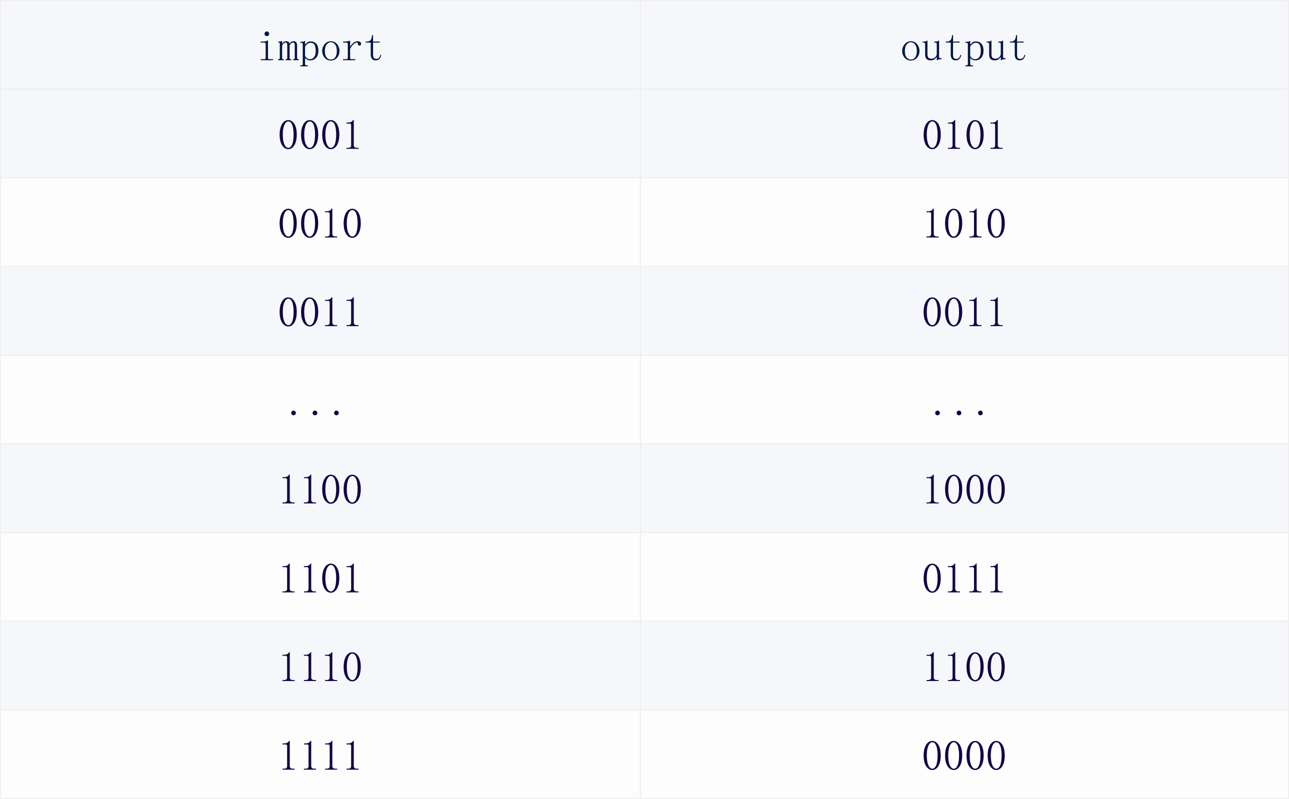
(2) Conduct DES decryption of the second layer of plaintext, and get the first layer of plaintext. Then the second layer of plaintext needs to DES the decryption operation to get the first layer of plaintext. This step uses the DES decryption algorithm to decrypt the plaintext.

(3) The first layer of clear text is stitched together to get the final clear text. Finally, the first layer of plaintext is stitched together to get the final plaintext.

Consider an S box substitution of 4-bit input to 4-bit output closer to the real scenario. Such S boxes may be used in some lightweight encryption algorithms. The following is a mapping of an example S box:



| import | output |
|---|---|
| 0000 | 1111 |



| import | output |
|---|---|
| 0001 | 0101 |
| 0010 | 1010 |
| 0011 | 0011 |
| ... | ... |
| 1100 | 1000 |
| 1101 | 0111 |
| 1110 | 1100 |
| 1111 | 0000 |

This S box has a higher degree of complexity and confusion. To enhance its safety, the S-box is designed with the following points:

(1) Nonlinearity: The mapping relationship between input and output is nonlinear and does not show obvious patterns or rules.

(2) Avalanche effect: for the small changes in the input, the output will produce a large change. For example, the input 0000 and 0001 are only one bit apart, but their output is three bits apart.

(3) Strict avalanche criteria: The S box meets the strict avalanche criteria (SAC), which means that for any change of the input, half of the bits in the output will change. Suppose we have a 4-bit input of 1011, and the corresponding output is 0110. Such S box substitution can provide higher security for encryption algorithms because it introduces complex nonlinear relationships that make cryptanalysis more difficult. It should be noted that this example is designed to illustrate how a more secure and complex S box replacement works. In real AES or DES, the design and parameters of the S box are deeply studied and rigorously tested to ensure the security of the encryption algorithm.

## 5. Implementation Methods

### 5.1 Programming language and the environment

This algorithm can be implemented using multiple programming languages, such as Python, C + +, etc. In order to improve the efficiency and security of the algorithm, it is recommended to use the lower-level programming language such as C + +. At the same time, in order to facilitate debugging and maintenance, easy programming languages such as Python can be used for testing and development.

### 5.2 Data structure and algorithm libraries

When implementing this algorithm, some data structures and algorithm libraries should be used. Among them, AES and DES encryption algorithms can be implemented using existing open source libraries, such as OpenSSL, Crypto + +, etc. At the same time, some basic data structures and algorithm libraries need to be used, such as STL.

### 5.3 key management

Key management is one of the important links to ensure encryption and decryption security. This algorithm requires two different keys: AES and DES, so key management is required. In practical application, the key can be managed by using the key pool to ensure the security and availability of the key.

## 6. Analysis of the Advantages and Disadvantages

Advantages: This algorithm uses two layers of encryption to improve data security and two different encryption algorithms to improve the difficulty of cracking and improve the confidentiality and integrity of data. Therefore, this algorithm has a relatively high security and reliability.

Disadvantages analysis: this algorithm also has some disadvantages. Firstly, using two layers of encryption will reduce the efficiency of encryption and decryption, managing two different keys will increase the complexity of key management, and increase the cost of implementation and maintenance due to two different encryption algorithms. So a balance between safety and efficiency.

C++ realize

```
#include <iostream>
#include <cstring>
#include <openssl/aes.h>
#include <openssl/des.h>
using namespace std;
// Define the AES and DES encryption and decryption functions
void AES_encrypt(const unsigned char *key, const unsigned char *plaintext, unsigned char *ciphertext) {
AES_KEY aes_key;
AES_set_encrypt_key(key, 128, &aes_key);
AES_encrypt(plaintext, ciphertext, &aes_key);
}
void AES_decrypt(const unsigned char *key, const unsigned char *ciphertext, unsigned char *plaintext) {
AES_KEY aes_key;
AES_set_decrypt_key(key, 128, &aes_key);
AES_decrypt(ciphertext, plaintext, &aes_key);
}
void DES_encrypt(const unsigned char *key, const unsigned char *plaintext, unsigned char *ciphertext) {
DES_cblock des_key;
memcpy(des_key, key, 8);
DES_key_schedule des_ks;
DES_set_key(&des_key, &des_ks);
DES_ecb_encrypt((const_DES_cblock *)plaintext, (DES_cblock *)ciphertext, &des_ks, DES_ENCRYPT);
}
void DES_decrypt(const unsigned char *key, const unsigned char *ciphertext, unsigned char *plaintext) {
DES_cblock des_key;
memcpy(des_key, key, 8);
DES_key_schedule des_ks;
DES_set_key(&des_key, &des_ks);
```

```
DES_ecb_encrypt((const_DES_cblock *)ciphertext, (DES_cblock *)plaintext, &des_ks, DES_DECRYPT);
}
int main() {
unsigned char plaintext[] = "Hello World!";// proclaimed in writing
unsigned char key[16] = "0123456789abcdef"; / / Key, 16 bytes (128-bit)
unsigned char ciphertext[16]; / / text
unsigned char decryptedtext[16]; / / Decrypted plaintext
// The first layer of encryption: use DES encryption
DES_encrypt(key, plaintext, ciphertext);
cout << "First layer ciphertext: ";
for (int i = 0; i < 8; i++) {
printf("%02x", ciphertext[i]);
}
cout << endl;
// Second layer of encryption: use AES encryption
AES _ encrypt (key, ciphertext, decryptedtext); / / Use AES decryption to get the first layer of clear text, here should be decryption, written encryption, thanks for correction!
cout << "Second layer ciphertext: ";
for (int i = 0; i < 16; i++) {
printf("%02x", decryptedtext[i]);
}
cout << endl;
DES _ decrypt (key, decryptedtext, plaintext); / / decrypt
cout << "First layer plaintext: " << plaintext << endl;
```

## 7. Algorithm Analysis

The encryption algorithm based on the combination of AES and DES has high security. First of all, the key length of DES algorithm is 56 bits. Although it is not one of the most secure encryption algorithms at present, it still has certain security in practical application. Secondly, the key length of the AES algorithm is 128 bits or 256 bits, which has very high security. Therefore, encryption algorithms based on the combination of AES and DES can resist attacks such as violent cracking.

## 8. Epilogue

This paper presents an encryption algorithm for human resources and social security information transmission based on the combination of AES and DES, which can improve the security and reliability of information transmission. By analyzing and testing the algorithm proved high safety and reliability. In the future, the algorithm will continue to be optimized and improved to adapt to more application scenarios.

## References

[1] Li Ming.(2015). Research and application of the AES encryption algorithm. Beijing: Tsinghua University Press.

[2] Liu Xiaoyan, Zhang Sanfeng.(2018). Improvement of the DES encryption algorithm and its security analysis. Information Network security, 20 (10), 23-28.

[3] Wang Qiang, Li Hongxia.(2020). Research on the hybrid encryption algorithm based on AES and DES. Computer Engineering and Application, 56 (13), 119-124.