ISSN: 2710-0170

Hybrid Encrypted Website based on Springboot

Wenjun Cao, Junhong Liu, Guifan He, Zhengji Ma, Xinyi Jiang

School of Computer and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051, China

Abstract

With the rapid development of modern network technology, people's personal information, file transfer and e-commerce all need strong protection. DataencryptTechnology also came into being. This time, JAVA language is used. The front end is built based on LayUI framework, and the back end is a hybrid encrypted website developed based on SpringBoot technology. The system functions include file encryption, information transmission and system authority management. The popular encryption algorithms such as AES, DES, RC4, TripleDes, MD5, base64 and RSA are used to encrypt the data, which is realized by Base64 or hexadecimal conversion to generate encrypted data or files, and then the encrypted data or files are decrypted.

Keywords

File Encryption; Springboot; Layui; Socket Data Transmission; Electronic Signature.

1. Introduction

With the rapid development of network technology, the convenience of file sharing is gradually increasing, and the scope is greatly expanded. Similarly, while technology brings convenience to people, it also brings challenges to people. The challenges faced by file protection are enormous and very difficult. Information security and convenient file transmission are what we are willing to pay attention to and discuss. The original intention of this system is to set up an encrypted website, which can realize file encryption, information transmission, study and exchange of encryption algorithms, and better summarize and summarize the articles and forums about encryption knowledge on the Internet and introduce them into this website for display. Better arouse our research interest in the direction and field of encryption, so as to develop this system.

2. System Introduction

(1) the operating environment of the system

This system is based on windows operating system, using java language, mysql database for data storage, and Navicat tool for database visual management. Run in Linux,Centos environment.

(2) System function description

Start the system to access url address, that is, access the web address of the system, and enter the account password to log in to the system. This system provides various encryption algorithms, such as symmetric encryption AES, DES, RC4 and TripleDES. Asymmetric encryption RSA. One-way encryption MD5, etc. The system also provides the functions of single file encryption and decryption, multi-file encryption and decryption and packing into compressed package, and re-encryption function. When encrypting, select the file to be encrypted and enter the key to be encrypted. After successful file encryption, the system will return an encrypted file path. Users can find the corresponding encrypted file through this file path. When decrypting, select the file to be decrypted and enter the encryption key to decrypt

ISSN: 2710-0170

it. After successful decryption, the system will return the decrypted file path. When decryption fails, the system will prompt the reason for failure, which is convenient for users to check the reason for failure.

In order to facilitate file information transmission, this system developed the function of friend communication. The system maintains its own users, and each user can search for friends to be added by name and send out add requests. The respondent can either agree or reject the request, and can send online messages or offline messages for real-time communication after requesting consent.

For the knowledge of popular science encryption algorithms, this system specially sets up a module to introduce various encryption algorithms in detail. It is convenient for users to read and browse, and enhances their interest in encryption.

(3) System function display

1) system login page

Users need to enter the correct verification code by username, password and. Back-end verification, if successful, you can access the system.

2) System registration page

The registration function is to send the registration code according to the user's email address, and the user can successfully register after filling in the received registration code correctly and perfecting other information collection. And only one user can be registered in an email account.

3) Home page interface

After the user logs in successfully, he will jump to the homepage interface of the system. The homepage shows some knowledge about encryption, which helps users to better understand encryption. Among them, many encryption algorithms are compared, and users can feel the characteristics, advantages and disadvantages of encryption algorithms more intuitively through the form of tables.

4) Encryption tool page

The encryption tool page demonstrates the encryption and decryption of seven encryption algorithms, such as AES, DES and RC4. Enter the plaintext in the left box, enter the key in the middle box, and click the encrypt button to display the ciphertext in the right box. As shown in figure 4 below.

5) DES file encryption

The interface encrypts files through DES encryption algorithm. Single text encryption can be performed, and multiple files can also be encrypted. After the file is encrypted, the original file can be decrypted by entering the key. This function is the core of this system. The details are shown in the figure below.

6) Friends chat

This function is designed to make it easier for users to send encrypted files to friends. The interface includes functions such as file transfer, information sending, searching and adding friends, and deleting friends.

3. The Core Algorithm and Principle of Encryption

Introduce the encryption-related algorithms, such as the principles of AES, DES, RSA encryption algorithms.

(1) AES encryption algorithm

AES(Advanced Encryption Standard)AdvancedencryptThe standard is a common symmetric encryption algorithm. Symmetric encryption algorithm is that the encryption key and decryption key use the same key rules. The length of the AES plaintext packet is 128 bits, that

ISSN: 2710-0170

is, 16 bytes.keyThe length can be 16 bytes, 24 bytes, or 32 bytes, that is, a 128-bit key, a 192-bit key, or a 256-bit key.

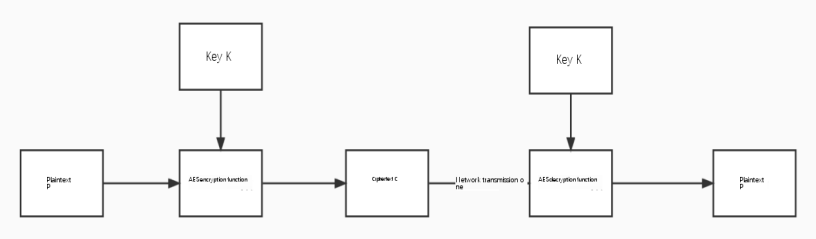

Figure 1. AES encryption flow chart

1) Plaintext P: data information that is not encrypted.

2) Key K: the password used to encrypt the plaintext. In the symmetric encryption algorithm, the encryption and decryption keys are the same. The key is generated through negotiation between the receiver and the sender, but it cannot be directly transmitted on the network, otherwise the key will be leaked. Usually, the key is encrypted by an asymmetric encryption algorithm, and then transmitted to the other party through the network, or directly negotiated face-to-face. key. The key must not be leaked, otherwise the attacker will restore the ciphertext and steal confidential data.

3) AES encryption function: C = E(K, P), where P is the plaintext, K is the key, and C is the ciphertext. The plaintext P and the key K are input as the input parameters of the encryption function E, then the encryption function E will output the ciphertext C.

4) AES decryption function: P = D(K, C), where C is the ciphertext, K is the key, and P is the plaintext. Taking the ciphertext C and the key K as the input parameters of the decryption function D, the decryption function will output the plaintext P.

5) Ciphertext C: The unintended content text obtained after the plaintext data information is processed by the encryption function.

6) The implementation code of AES algorithm encryption and decryption in this system is as follows: create a tool class of AES in utils, and call it in the controller layer.

```
**
 * AES encryption
 * @param mingWen
 * @param key
 * @param aesXs
 * @return
 */
@RequestMapping("encrypt")
public Object encrypt(String mingWen, String key, String aesXs) {
   try {
      String miWen = "";
      if (aesXs.equals("hexadecimal display")) {
         miWen = AESUtil.encryptToHex(mingWen, key);
      }else {
         miWen = AESUtil.encryptToBase64(mingWen, key);
      }
      return R.ok(miWen);
   }catch (Exception e) {
```

ISSN: 2710-0170

```
        return R.failed(e.getMessage());
    }
}

/**
 * AES decryption
 * @param miWen
 * @param key
 * @param aesXs
 * @return
 */
@RequestMapping("decrypt")
public Object decrypt(String miWen, String key, String aesXs) {
    try {
        String mingWen = "";
        if (aesXs.equals("hexadecimal display")) {
            mingWen = AESUtil.decryptByHex(miWen, key);
        }else {
            mingWen = AESUtil.decryptByBase64(miWen, key);
        }
        return R.ok(mingWen);
    }catch (Exception e) {
        return R.failed(e.getMessage());
    }
}
```

(2) DES encryption algorithm

DES is a block encryption algorithm. Typical DES encrypts data in groups of 64 bits, and the same algorithm is used for encryption and decryption. Its key length is 56 bits (because every 8th bit is used for parity), the key can be any 56-bit number, and it can be changed at any time. There are very few weak keys that are considered easy to crack, but it is easy to avoid them. So confidentiality depends on the key.

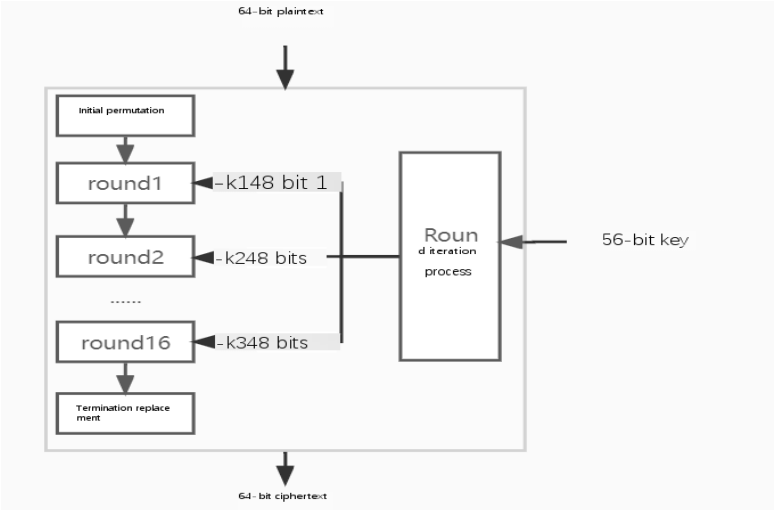

Figure 2. DES encryption algorithm map

ISSN: 2710-0170

1) Initial permutation: In DES encryption algorithm, plaintext and ciphertext are 64-bit blocks. The length of the key is 64 bits, but every eighth bit of the key is set as parity bit, so the actual length of the key is 56 bits.

2) Generate sub-key: the operation principle is the same as the initial permutation, but this is in the 7X8 permutation table, and the 64-bit key becomes 56-bit.

3)Round iteration process: Because the encrypted plaintext is long, DES encryption needs to be iterated repeatedly.

4) Termination of permutation: generate ciphertext. The decryption process is the reverse of the encryption process.

5) The general principle is: encryption process: ciphertext C= plaintext P⊕ key K; Decryption process: plaintext P= ciphertext C⊕ key k;

6) The design and implementation of DES of this system is shown in Figure 3 below.

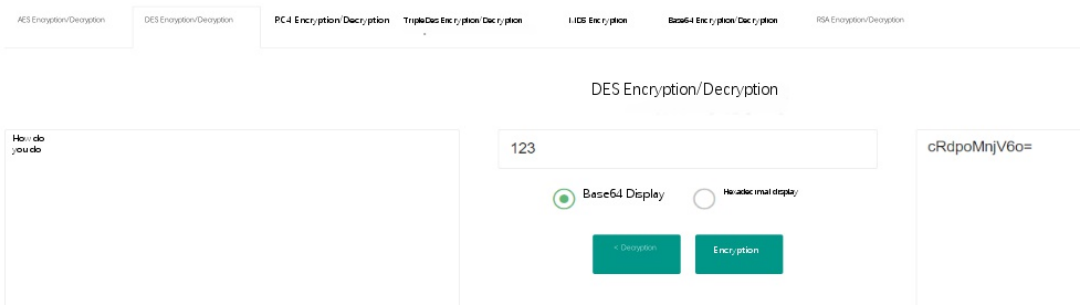

Figure 3. DES implementation diagram

(3) RSA encryption algorithm

RSAencryption algorithmIt is an asymmetric encryption algorithm. The so-called asymmetry means that the algorithm uses different keys for encryption and decryption, that is, the encryption key is used for encryption and the decryption key is used for decryption.

1)RSA encryption process

The general encryption formula of RSA: ciphertext = plaintext^{E}modN

RSA encryption is the process of finding the remainder after dividing the plaintext to the e power by n. From the general formula, anyone who knows E and N can encrypt RSA, so E and N are the keys of RSA encryption, that is to say, the combination of E and N is the public key. We use (E,N) to represent the public key: public key =(E,N)

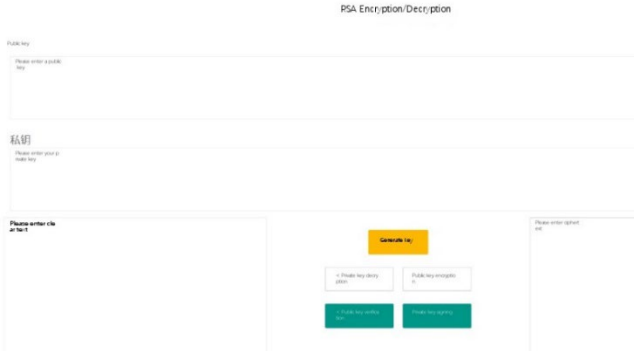

Figure 4. Implementation diagram of 4.RSA

2) RSA decryption process

The general decryption formula of RSA: plaintext = ciphertext^{D}modN

ISSN: 2710-0170

The remainder after dividing the ciphertext to the d power by n is the plaintext, which is the RSA decryption process. Knowing d and n can decrypt the ciphertext, so the combination of d and n is the private key: private key =(D,N)

Decryption is to find "mod N to the d power". D is the first letter of Decryption; N is the first letter of the Number.

3) The design and implementation of RSA in this system is shown in Figure 4.

4. Design and Implementation of System Functions

(1) Front-end logic

Open source modular front-end UI framework based on front-end Layui. Adopt Layui's layout framework, add Layui's components and some basic elements. Realize the front-end interface effect. The resource package in the system is the front-end resource package of the system. The html folder under the resource package creates HTML files such as login, registration, user, home page, etc. The img folder stores the images and other elements used by the system. The lib folder stores Layui's configuration files, js and other files. The architecture is shown in Figure 5 below.

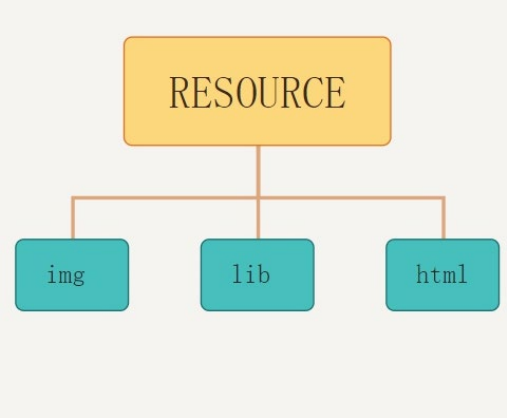

Figure 5. DES encryption algorithm map

(2) back-end logic

The back-end is based on the SpringBoot framework and B/S three-tier architecture. Under the file package of the system, a view layer Controller package, a business logic layer Service package and a data access layer Mapper package are created. At the same time, the utils package is configured, and many tool classes of encryption algorithms are created. The view controller, Sockect connection service and paging manager are configured under the config package.

1) Login logic

Send a request to the back-end through ajax, and accept the input information from the front-end users. The back-end performs verification processing, and if the information is correct, logs in to the system, otherwise, an error message of login failure is returned.

A. Login code implementation

```
/**
 * Login
 * @param user
 * @param request
 * @return
 */
@RequestMapping("login")
```

ISSN: 2710-0170

```
public Object login(User user, HttpServletRequest request){.....}
```

B. Image verification code implementation

```
**
 * generate picture verification code
 */
@RequestMapping("captcha")
public void captcha(HttpServletRequest request, HttpServletResponse response) throws Exception {
   CaptchaUtil.out(request, response);
}
```

/register logic

The core function of registration is to send the registration code to the mailbox entered by the user. At the same time, the back-end performs a lot of logical verification processing to prevent the misuse of one mailbox by registering multiple users.

A. Get the registration code

```
/**
 * Get the registered email verification code.
 * @param email
 * @return
 */
@SneakyThrows
@RequestMapping("registerCode")
public Object registerCode(String email) {.....}
```

B. Successfully registered and saved user information.

```
**
 * Save user management
 */
@RequestMapping("save")
public Object save(User user) {.....}
```

/information transmission implementation

In this system, the WebSocketconfig class is established under the Config package, and it is managed by SpringBoot to build the socket environment. Applications can send or receive data through it, and can open, read, write and close it like files. Allows applications to insert I/O into the network and communicate with other applications in the network. Under the construction of Socket, the system information sending function is realized. In the back-end code layer, the chat records of users are stored in the database, and the association table of the conversations between two users is created. In order to realize that the other party can download and view the file when sending it, the system writes the download function in the code level, thus realizing this function.

2) Encryption and decryption file implementation

When encrypting files, in order to ensure the security of files and make them difficult to be cracked, this system chooses DES algorithm to encrypt files. Realize the encryption of single file or multiple files. In the process of implementation, the first step is to import the dependency of DES and create the tool class of DesUtil. The second step is to create the encryption path of the system, and set it as the upload path under the D drive of the computer. This path is used to save the encrypted files. The third step is to read the file, read the required encryption and decryption file, and encrypt the file by setting the key. This system can be divided into two

ISSN: 2710-0170

methods when realizing multi-file encryption and decryption. The first method is to select multiple files, encrypt them together, and store the encrypted files in the default upload file. The second is to introduce the dependence of file compression and encrypt and decrypt the compressed files.

5. Summary

To sum up, this paper expounds the developed functions of this system, and introduces the design principle of this system, the logic of front and back ends and the principle of encryption algorithm. In order to publish the design idea of this system, our project team, out of interest in the field of encryption, hopes to put it into practice through the theoretical knowledge we have learned. However, there are still some problems in this system, and it is necessary to add some unexplored functions, such as forum communication, user comments, etc., which can be further improved in the front-end interface optimization. The ultimate goal is to create a large, functional and flexible encrypted website. Therefore, the system can be continuously improved and perfected in future study.

Acknowledgments

Fund: University of Science and Technology Liaoning 2022 Undergraduate Innovation and Entrepreneurship Training Program Project, ProjectNo.: X202210146013.

References

[1] Yang Bo, Wen Zhiping. Design and implementation of online collaborative office system based on SpringBoot [J]. Computer Knowledge and Technology, 2022,18(22):49-51.

[2] Li Weichao. Data encryption technology in computer network information security [J]. Network Security Technology and Application, 2022,(11):23-24.

[3] Yue Kun, Wang Xiaoling, Zhou Aoying. Web service core support technology: research review [J]. Journal of Software, 2004,(03):428-442.

[4] Zhang Yizhi, Zhao Yi, Tang Xiaobin. Research on MD5 algorithm [J]. Computer Science, 2008, (07): 295-297.

[5] Yan Shichun, Li Xiaohui. Research on homomorphic encryption friendly support vector machine algorithm [J]. Modern Electronic Technology, 2022,45(20):54-58.