Security Research and Solution of Data Exchange Platform

Xidong Liu

Shandong Institute of Commerce and Technology, Jinan, China
ddcuy@163.com

Abstract
This paper first discusses the security of data exchange, and puts forward the security scheme design of data exchange platform based on XML. It specifically puts forward the problems of how to choose encryption algorithm, design of exchange key, encryption and decryption of data files, etc. This scheme is easy to implement and use in application. However, because the traditional centralized model is faced with problems that cannot be solved, such as the central node is easy to attack, this paper proposes a data exchange model based on blockchain technology, and hopes to use this model to ensure that data will not be lost, leaked and forged.

Keywords: *blockchain, data exchange, asymmetric encryption*

1 INTRODUCTION

In the process of informatization of enterprises and institutions, huge data is stored in different local information systems. With the increasing amount of data and the rise and development of cloud computing technology, valuable data in these information systems are faced with extraction, integration and sharing. The general definition of data exchange is the process of data interaction between information entities other than their own entities on the premise of ensuring user data security (Zhou2020). Data exchange platforms suitable for different application scenarios came into being. At the beginning of this century, XML based data exchange technology began to be applied in the field of data exchange. The data exchange platform based on XML data exchange technology can exchange reliable and traceable information and data for heterogeneous databases of various application systems [1].

However, in recent years, data security incidents have occurred frequently, such as data loss, disclosure and forgery. Users have put forward higher requirements for security issues in the process of data exchange. Developers have designed a hybrid encryption algorithm suitable for the data exchange platform to ensure the security of transmitted information. At the same time, they have restricted, managed and recorded the access rights of users participating in the exchange, especially after the advent of blockchain technology, Digital signature and hash algorithms have been widely used in blockchain networks, and the security of blockchain based data exchange platform has been greatly improved.

2 SECURITY REQUIREMENT ANALYSIS OF DATA EXCHANGE PLATFORM

By analyzing the business requirements of the data exchange platform, we can see that the security requirements of the XML based data exchange platform mainly include the following:

- System registration: All application nodes (servers and clients) using the data exchange platform must first register the system, register identity information, and obtain the user rights assigned by the system.
- Identity authentication: Each time an application node logs in to the system, it needs to verify its identity and authenticate user permissions. Each application node needs to authenticate the permission to send and receive messages every time it delivers messages to prevent unauthorized events.
- Message content: The message sender shall encrypt all or part of the XML message content to be transmitted to protect sensitive information and prevent unauthorized nodes from viewing the message content or malicious destruction.

© The Author(s) 2023
B. Fox et al. (Eds.): IC-ICAIE 2022, AHCS 9, pp. 895-900, 2023.
https://doi.org/10.2991/978-94-6463-040-4_135

- Digital signature: Realize sender identity binding, that is, digitally sign the whole XML document at the message sender to ensure that the information is non repudiated and ensure the integrity and correctness of the text.
- Message transmission process: Ensure the integrity of message transmission and atomicity of transactions.
- Log: The data exchange platform has set up multi-level logs to record information about the daily operation of the system, such as access logs and error logs. It is necessary to ensure that the logs cannot be tampered with. [2]

3 SECURITY SCHEME DESIGN OF DATA EXCHANGE PLATFORM BASED ON XML TECHNOLOGY

This scheme combines the XML technology with the existing encryption algorithm, flexibly uses the symmetric key algorithm and asymmetric key algorithm to obtain the key. According to the tree structure of the XML document, the encryption object can be flexibly selected according to the needs of both parties, encrypting the entire document or an element or the content of an element, so that the point-to-point data transmission can be extended to multi-party data transmission to meet the needs of multi-party sessions.

3.1 Selection of encryption algorithm

Encryption algorithms can be divided into two categories according to the type of algorithm: symmetric encryption algorithm and asymmetric encryption algorithm.

3.1.1 Symmetric key algorithm

Encryption and decryption use the same key, also known as the shared key. The initiator uses the shared key for encryption and the receiver uses the same shared key for decryption. The main advantage of symmetric key is the fast speed of encryption and decryption, but the key transmission is a complex process.

3.1.2 Asymmetric key algorithm

The keys generated by the asymmetric algorithm are in pairs, one of which is the public key and the other is the private key. The public key is used for encryption and can be made public. The private key is used to decrypt the document encrypted with the public key. Only the entity with the key pair knows it.

The main drawback of asymmetric key algorithm is that the speed of encryption / decryption is too slow and the amount of data processing is small. In the actual operation process, we usually adopt the following method: asymmetric encryption algorithm is used to manage the key of symmetric algorithm, and then symmetric encryption algorithm is used to encrypt data, which integrates the advantages of two types of encryption algorithms. The encryption speed is fast, and the key can be safely and conveniently managed. [3]

3.2 Design of exchange key

In this scheme, we use a hybrid encryption algorithm to encrypt XML data using a symmetric algorithm (DES algorithm) and an asymmetric encryption algorithm (RSA algorithm) to encrypt des keys. After encrypting des key with RSA algorithm, it can be made public, and RSA encryption key can also be made public. Therefore, only a small number of RSA decryption keys need to be kept secret in the whole system. The encryption system can not only give play to the advantages of DES algorithm in fast encryption speed and good security, but also give play to the advantages of RSA algorithm in convenient key management.

Suppose that LAN A and LAN B correspond to two departments A and B in the basic model architecture of data exchange. Department a needs to accept data data1 from department B, and department B needs to accept data data2 from department A. The specific key generation steps are as follows:

First, the data exchange center uses RSA algorithm to generate the public key and private key. The data exchange center retains the private key and publishes the public key through the web page. Secondly, departments A and B use the DES algorithm to generate their DES keys, encrypt their DES keys with the public key published by the data exchange center, generate the encrypted file ENCDES, and then send the keys to the data exchange center through public channels. Then, the data exchange center decrypts the ENCDES files obtained from departments A and B, obtains the des keys of the two departments, and establishes a connection relationship between the keys and departments A and B.

After the above operations, the data exchange center shares the key with A and B respectively. At this time, if A sends the XML document encrypted with the key to the data exchange center, the data exchange center can decrypt the encrypted file with the DES key of department A saved by itself. Similarly, the data exchange center can encrypt the data or files that need to be transmitted to department A.

The specific process is shown in Figure 1:

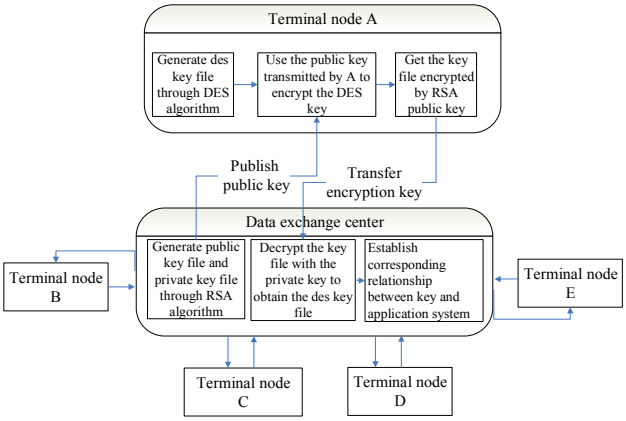

Figure 1 key generation and transmission flow chart

To complete the above work, we need to develop the following classes:

Table 1: Four classes to be developed

Class name	function
CreateRSA	The data exchange center generates a key pair, and then saves the public key and private key to privatekey DAT and publickey In the dat file, you keep the private key file and send the public key file to the data sender. At the same time, rename the original public key and private key files for backup.
CreateDES	The terminal node is used to generate des key and save the key to secert_ key. Dat file. Back up the original key file.
EncDESSK	The data provider encrypts the generated des key with RSA public key and saves the encrypted key to rsasecert_ key. In DAT, the original rsasecert_ key. Dat file for backup.
DecDESSK	The data receiver uses the RSA private key to save the_ key. Decrypt the key in the dat file and save the des key to secert_ key. Dat file is used to decrypt the data sent by the data sender.

3.2.1 Generate public key and private key with RSA algorithm

We use the keypairgenerator in the security package of Java to create the key pair generator, and specify the encryption and decryption algorithm as RSA. Then get the key pair through keypair, and finally save the generated public key and private key to the public key file and private key file respectively.

The codes are as follows:

```
KeyPairGenerator
keyGen=KeyPairGenerator.getInstance("RSA");
keyGen.initialize(1024);
KeyPair key=keyGen.generateKeyPair();
```

In order to ensure the security and reliability of the key, RSA public key, private key and des key need to be updated from time to time.

3.2.2 DES key generation

The DES algorithm has three entry parameters: key, data and mode. The key is a 64 bit key with 8 bytes, which is the working key of DES algorithm; Data is also 8 bytes and 64 bits, which is the data to be encrypted or decrypted; Mode is the working mode of Des. There are two modes: encryption or decryption.

If the mode is encryption mode, use the key to encrypt the data, and generate the password form of data (64 bits) as the output result of DES; If the mode is the decryption mode, the key is used to decrypt the data in the password form, and the data is restored to the clear code form (64 bits) of data as the output result of DES.

We use the SecureRandom class in the Java security package, the KeyGenerator class in the crypto package, the key interface, and the Cipher class. The Cipher class provides cipher functionality for encryption and decryption. It forms the core of the JavaCryptographic Extension (JCE) framework.

3.2.3 Encrypt and decrypt DES key

After each terminal node in the system obtains the public key file and DES key, it encrypts the DES key with the public key to obtain RSAsecert_key. dat key file. Then the terminal node transmits the encrypted key to the data exchange center, which uses the private key to decrypt RSAsecert_key. dat and generate a secret_key.dat file. In this way, the data exchange center can use it to encrypt and decrypt data files.

3.3 Encryption and decryption of data files

The XML document can be encrypted, and some of its elements can also be encrypted, while other parts still exist in clear text. If the entire XML file is encrypted, the

data in the XML file shall be read after the XML file is generated, and the des key shall be used to encrypt it; If you encrypt some elements, you need to encrypt the element when reading the data, and then write the element to the XML file; In the case of multi-user data exchange, different parts of the same document can be encrypted with different keys. The same XML file is sent to different recipients, and the recipients can only access the part of information with permission. The data exchange system designed in this scheme mainly aims at point-to-point data exchange, and multi-user can expand on this basis.

The specific steps for encrypting the entire file are as follows:

The terminal node first reads the data from the data file A, then encrypts the data file with the DES key, writes the encrypted data to the new data file B, and deletes the original data file A.

After receiving the data file B, the receiver needs to decrypt the data file B. First, make a data backup of data file B, then read the data from data file B, decrypt the data file with DES key, write the decrypted data into data file C, and finally delete data files B and C.

This completes the encryption and decryption of the data and ensures the security of the data.

3.4 Transmission and storage of key file

The two basic elements of cryptosystems are encryption algorithms and key management. Key is the key information to control encryption algorithm and decryption algorithm. Its generation, transmission, storage and other work is very important. In the information processing system, some information of the key must be placed in the machine. The transmission of the key file is very important at this time. It cannot be transmitted at the same time with the encrypted data file. It must be transmitted from one party to the other through another way. Here we need to establish a special mechanism to ensure the transmission of key files. We can transmit through a special e-mail or through the network, which is managed by a specially assigned person. The key also needs to be updated from time to time. For example, one party can generate new keys on the first day of each month, transmit them to the other party through the network or e-mail, and have special personnel to save these keys. Moreover, there is no concept of key storage for data encryption, once you move from one key to another, the old key is no longer needed. However, when decrypting, the old key must be kept, otherwise the data encrypted with the old key cannot be read. Therefore, when the key is regenerated, the original key must be renamed and saved to ensure that these keys can be found when used in the future.

3.5 Advantages and disadvantages of data exchange platform based on XML Technology

The design scheme of data exchange platform based on XML technology is easy to implement, easy to use, and can realize cross platform. However, the data exchange system needs to cooperate with a variety of different terminals, including client, application server, database, web server, etc. with complex structure and function, many users and frequent data exchange, it may face problems such as identity fraud, data tampering, data leakage, data repudiation, and easy attack of the central exchange node. In particular, once the central node of the centralized network model is attacked, all information, especially sensitive data, will be leaked. Therefore, to some extent, people hope to decentralize the traditional centralized network. The combination of blockchain technology and data exchange technology undoubtedly provides a more secure mode for such services.

4 DATA EXCHANGE MODEL BASED ON BLOCKCHAIN TECHNOLOGY

With the arrival of the big data era, massive data and frequent exchange pose a challenge to the traditional centralized model. Through the application of traditional technologies such as distributed data storage, P2P transmission, consensus mechanism, encryption algorithm and smart contract, blockchain technology has the characteristics of decentralization, tamperability, traceability, multi-party maintenance, openness and transparency [4]. With the help of blockchain, combined with smart contract technology and cryptography technology, it can provide the data exchange system with the required functions such as privacy protection, ownership confirmation, authority management and data accountability.

4.1 Problems to be solved

4.1.1 Permission management

Through smart contracts, unified data standards can be established, and data access rights can be controlled without the control of a third-party centralized node. Only business related parties can read or call relevant data, isolate unrelated parties, and ensure the privacy and security of sensitive data sharing.

4.1.2 Trusted data encryption

The blockchain based trusted data exchange technology mainly applies two core technologies.

One is homomorphic encryption. The main idea of homomorphic encryption is that after homomorphic encryption of the original data, specific operations are

performed on the ciphertext, and the ciphertext calculation result obtained after homomorphic decryption is equivalent to the data result obtained by directly performing the same calculation on the original plaintext data [5].

This technology can meet the requirements of the data query results when the data is trusted to be exchanged, and the private data is not out of the warehouse or leaked.

The other is asymmetric encryption. This technology is mainly applied to account generation and transaction signature in the blockchain network. Asymmetric encryption only needs to disclose the public key, and the private key is not publicly saved. Digital signature applies this feature of asymmetric encryption technology. Through digital signature, transaction legitimacy, data source and data integrity can be verified in decentralized networks such as blockchain to prevent data forgery and tampering.

4.2 Data exchange process

In the blockchain based trusted data exchange process, the data provider first publishes the data directory to be provided to the blockchain, and records it in the block as a certificate of deposit; Suppose that the demander a requests this piece of data, can access the blockchain network through HTTP, broadcast the request (data after private key signature, i.e. smart contract) to each node, and each core node queries and verifies according to the conditions of the smart contract. Assuming that node B determines the authority of a, then node B can encrypt the shared and exchanged data through homomorphic encryption, and then transmit the asymmetric encrypted data URL point-to-point to A, A obtains the requested decrypted source data through the link address. In the whole process, the data is still stored in the database system of node B, and B owns the ownership of the data.

In the whole process of data exchange, homomorphic encryption enables the data to be operated without decryption, does not expose the original data, and ensures the data ownership of the sharing party. The smart contract decentralizes the data processing, grasps the data execution right, controls the access and execution authority of encrypted data, destroys the encrypted data after it is empty, and the user only has the right to use the ciphertext results, ensuring the data security.

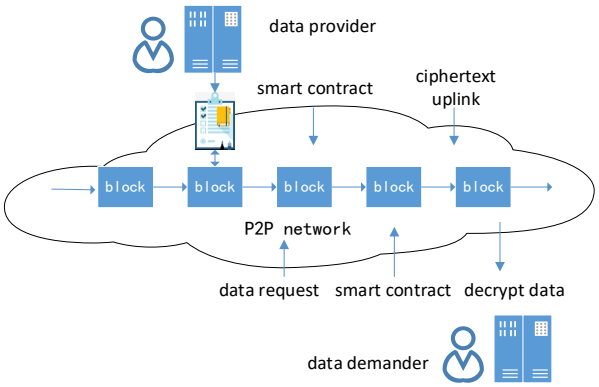

Figure 2 Blockchain information resource sharing and exchange model

4.3 Advantages and disadvantages of data exchange platform based on blockchain Technology

The data sharing and exchange model designed in this scheme makes use of the decentralized trust model of blockchain, makes use of the characteristics of the data on the chain, such as tamperability, traceability and security, and combines the smart contract technology and cryptography technology to effectively solve the problems of the security of traditional data exchange, the vulnerability of data centers, and the consistency of standards in data exchange, so as to ensure that all enterprises and institutions can work across regions, departments Control over data operations during cross platform information exchange.

5 CONCLUSIONS

With the continuous development of the Internet, it has become a common demand of all walks of life in the era of big data to break the information island barrier of various data sources through effective means to make data flow and play its value. The data exchange platform based on XML, blockchain and other technologies can realize data exchange and circulation between nodes under the condition of protecting data privacy, and meet the business needs of different scenarios.

REFERENCES

[1] David Chaum et al.(2016). cMix: Anonymization by high-performance scalable mixing. USENIX Security.

[2] Liu Xidong (2007). XML-based Data Exchange Research and Application. Shandong University Master Thesis.

[3] Xi Hongqi, Chang Xiaopeng (2012). Research on a digital signature scheme based on asymmetric encryption algorithm and hash function. Journal of

Henan Institue of Education (Natural Science Edition), 36-37.

[4] Guo Shangtong, Wang Ruijin, Zhang Fengli (2020). Overview of blockchain technology principle and Application. Computer Science,271-281.

[5] Liang Wei (2020). Blockchain based trusted data exchange technology and Application. ICT and Policy, 91-95.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.