information

Article

Research on Data Transaction Security Based on Blockchain

Yongbo Jiang, Gongxue Sun and Tao Feng *

School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China
* Correspondence: fengt@lut.edu.cn

Abstract: With the increasing value of various kinds of data in the era of big data, the demand of different subjects for data transactions has become more and more urgent. In this paper, a blockchain-based data transaction protection scheme is proposed to realize the secure transaction sharing among data. This paper carries out the following work: by analyzing the existing data transaction models, we find the data security and transaction protection problems, establish a third-party-free data transaction platform using blockchain, protect users' data security by combining AES and improved homomorphic encryption technology, and upload the encrypted data to the Interplanetary File System (IPFS) for distributed storage. Finally, we use the powerful functions of the IPFS, combined with inadvertent transmission protocol, two-way authentication, zero-knowledge proof, and other security verification for data transactions. The security analysis proves that this scheme has higher security despite the time overhead, and we will continue to optimize the scheme to improve efficiency in the future.

Keywords: data transactions; privacy protection; data security; blockchain; identity authentication

Citation: Jiang, Y.; Sun, G.; Feng, T. Research on Data Transaction Security Based on Blockchain. *Information* **2022**, *13*, 532. https://doi.org/10.3390/info13110532

Academic Editor: Nelly Leligou

Received: 16 September 2022
Accepted: 7 November 2022
Published: 8 November 2022

1. Introduction

In recent years, with the deepening popularity of the Internet and the continuous development of informatization, the importance of data to individuals, enterprises, and countries has become increasingly prominent. For individuals and enterprises, data are a valuable asset; for countries, data are a fundamental strategic resource, and the combination of data and blockchain is the focus of current research [1]. Whether data have an increasingly important impact in personal life, enterprise production, or national and social governance, the flow of data transactions is about the overall situation of the data element market cultivation, but its overall system is lacking. The transfer and sharing of data control between two or more parties and the circulation of data transactions are mainly reflected in "data sharing" and "data reuse". The empirical studies of various countries show that data transaction circulation faces three major dilemmas: technology, standards, and law [2]. As an important means of data circulation [3], data trading satisfies the needs of data consumers and allows data owners to gain economic benefits from it. At the same time, data trading promotes the open sharing and resource integration of data, which makes data play an important role in social governance, scientific research, commodity research and development marketing, and public life and entertainment [4]. However, data have their own special characteristics, and data contain information related to individuals such as life and work, which may bring distress to individuals once the traded data are illegally stolen, leaked, and linked to the real identity of individuals. In recent years, blockchain transaction security problems have emerged [5], and there are two main reasons for this phenomenon: (1) the existing trading platform does not pay attention to the protection of user data and relies on the central trading platform; the reliability and security of the central platform itself is very important, and once the platform is maliciously attacked, it may interrupt the service and leak user data, thus causing losses to users; (2) there is a possibility that data are retained and deposited on the trading center platform during the

data trading process, and the trading center does not ensure that users have control over the access to their personal data.

Aiming at the existing security and fairness problems such as data transaction and data protection, many scholars have developed various schemes to better protect the interests of users.

Soubhagya et al. [6] studied how to use blockchain for secure healthcare data transactions. Their scheme exploited the decentralized and immutable record-keeping properties of blockchain technology to potentially improve the scalability, security, and privacy of healthcare data. Using the power of blockchain technology, a novel smart contract-based framework called electronic medical record infrastructure (EMRI) was proposed for the privacy protection of proprietary information and to enable scalable and secure communication. Guo et al. [7] proposed a transactional model based on an IoT data blockchain using zero-knowledge proofs and proxy re-encryption, addressing the issues of privacy challenges and the inability to achieve key leakage and the risk of flexible data sharing using asymmetric encryption for data sharing on blockchain. Ren et al. [8] proposed an efficient, provable, fair document exchange protocol with transactional privacy that allowed untrusted buyers and sellers to exchange files fairly. None of the above research solutions had an effective data validation method to ensure the rights of data consumers, and this proposal uses inadvertent transmission to validate data and protect the legal rights of data consumers.

Segura et al. [9] proposed a fair data transaction protocol based on the Bitcoin scripting language, where they used inadvertent transmission to verify data and use an elliptic curve digital signature vulnerability to exchange private keys. The scheme could further reduce the encryption and decryption overhead if it first encrypted the digital content using a symmetric encryption algorithm. Kiyomoto et al. [10] presented the design of a fair-trade protocol for anonymous data sets between data agents and data analysts. The scheme used a combination of public key encryption and hash functions to ensure data confidentiality and tamper evidence. Wang et al. [11] proposed a new peer-to-peer (P2P)-based digital rights management scheme to protect valuable digital content. The scheme used P2P technology to reduce the storage overhead of servers, combined symmetric encryption and public key encryption to ensure the confidentiality of digital content, and used bitcoin transaction scripts to ensure the fair transmission of encryption keys. Zhao et al. [12] proposed a new blockchain-based fair data transaction protocol. They used inadvertent transmission and similarity learning to verify data, ring signatures and two-factor authentication to guarantee user privacy, and Ethereum smart contracts to exchange cryptographic keys. When a transaction went wrong, they used arbitration to ensure fairness for both parties. In addition to the above studies, Missier et al. [13] studied the value of real data through data transactions. Alrawahi et al. [14], Lin et al. [15], and Cattelan et al. [16] studied data transactions through a platform, designing e-commerce type protocols, etc. Perear et al. [17] and Lin et al. [18] studied how to encourage digital content transactions. Huang et al. [19] and Fan et al. [20] studied the fair exchange of digital content. Qian et al. [21] studied the use of offline semitrusted third parties and interactive verification signatures to secure data transactions. The above studies do not verify the legitimacy of the data quantity of the transaction, which makes it difficult to guarantee the rights of the data owner; our scheme uses zero-knowledge proof for the verification of the transaction amount to guarantee the legitimate rights of both parties of the transaction.

In this paper, the transaction process of personal data should meet the following security requirements: to ensure data security, to ensure that individuals control the access rights of their data, data plaintext cannot be viewed and obtained by irrelevant parties; participants cannot expose their transaction amount in the system, and the transaction validity can be verified through intelligent matching. Based on the above analysis, it can be seen that trading personal data through the existing trading platforms cannot meet the requirements of data security and privacy protection for personal data trading.

It can be seen that there are still many security risks in the current data trading platforms. In order to solve the above security problems, we established a secure trading platform through the blockchain.

The main research work of this paper is as follows:

(1) Based on the research and analysis of existing data encryption storage security mechanisms and data access rights control management schemes, a new data protection scheme is proposed by combining AES symmetric encryption, improved homomorphic encryption, and blockchain technology. This scheme first uses symmetric encryption to encrypt long data, and then uses improved homomorphic encryption to encrypt the symmetric key, and the encrypted data ciphertext is uploaded to the IPFS's distributed system for storage, so as to solve the problem of data storage on untrusted third-party platforms and security storage issues.

(2) We propose a data transaction scheme that does not require third-party participation. In order to prevent malicious users from stealing data and to prevent user identity information from being stolen by illegal users, a two-way identity between data consumers and data owners is used in data transactions. The authentication scheme can effectively prevent illegal users from stealing users' personal identity information; data consumers and data owners use inadvertent transmission to verify whether the data meet their needs during transaction verification and verify the legitimacy of the transaction amount through a zero-knowledge proof.

(3) The feasibility of the scheme is verified through a security analysis, performance analysis, and efficiency analysis. The analysis shows that the scheme in this paper can realize data transactions between users safely under the premise of ensuring data privacy, which is more secure and reliable than similar schemes. The scheme in this paper has the characteristics of feasibility, safety, and effectiveness.

2. Materials and Methods

2.1. Methods

2.1.1. Oblivious Transfer

Oblivious transmission is a communication protocol that can protect the privacy of both parties. It causes the receiver to receive messages from the sender in a random manner. The sender knows that the receiver accepts some messages but does not know which messages the receiver accepts.

At present, the most widely used oblivious transfer protocol is the 2 takes 1 transfer, whose specific process is as follows:

1. User A generates two messages m_0 and m_1;
2. User B selects one digit and enters c $\in \{0, 1\}$;
3. User A interacts with user B. User A enters messages m_0 and m_1, user B enters c, and the program returns m_c to user B.

In this process, user A only knows that user B received one of the two messages but does not know what message user B received, and user B only knows what message they received but does not know what the other message is.

We applied this technique to the data validation process in our data trading scheme, where the data consumer randomly validates a part of the data to see if the data owner has mixed invalid data into the data they provide. At the same time, the data owner only needs to disclose a small part of the data, without worrying about revealing more information related to the plaintext of the data.

2.1.2. Homomorphic Encryption Algorithm

Homomorphic encryption allows the server to encrypt the data without knowing the original plaintext; it allows the server to perform specific mathematical operations on the encrypted data and the decryption result is consistent with the corresponding plaintext operation result, thus protecting the data [22]. The basic process of data encryption and decryption is shown in Figure 1.

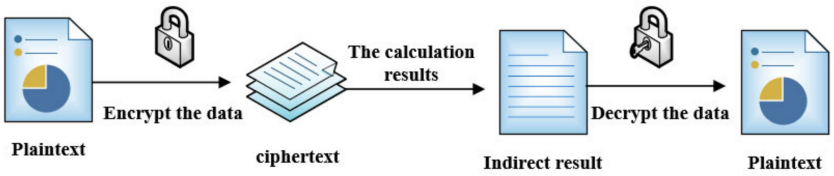

Figure 1. Basic flow chart of data encryption and decryption.

In an encryption system, if the encryption algorithm is *CK*, the decryption algorithm is *DK*, and the plaintext is *n* and m, then the homomorphic encryption satisfies the following properties

$$DK(CK(n) * CK(m)) = n * m \tag{1}$$

$$DK(CK(n) + CK(m)) = n + m \tag{2}$$

The first attribute of the above encryption method is multiplicative homomorphic encryption, and the second attribute is additive homomorphic encryption. If both attributes are satisfied, the encryption algorithm becomes fully homomorphic encryption [23].

2.1.3. Elliptic Curve Encryption

An elliptic curve is not an ellipse; it is called elliptic curve encryption because its curve equation is similar to the equation used to calculate the circumference of an ellipse. A general elliptic curve refers to the elliptic curve determined by Weierstrass's equation, as follows:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \tag{3}$$

It is the set of all the solutions (x, y) plus an infinite point O. The security of an elliptic curve encryption algorithm is based on the difficulty of solving the discrete logarithm of its elliptic curve (ECDLP) [24]. Let E be a curve and G and Q two points on the curve, where x is the discrete logarithm problem of the elliptic curve [25].

The point-plus geometric representation is shown in Figure 2.

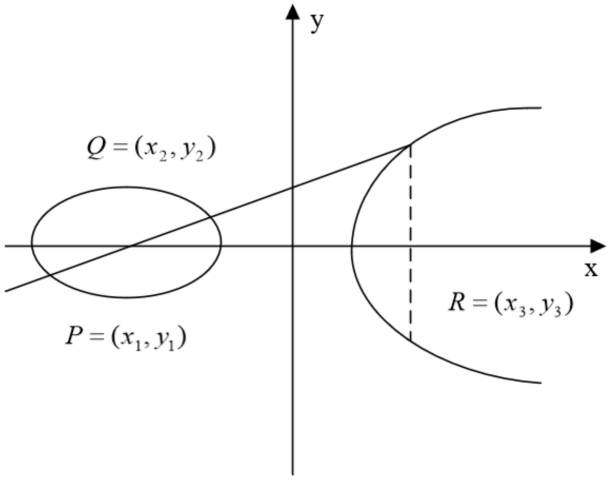

Figure 2. Point-plus geometric representation.

Let the base fields F, x, y belong to F and satisfy the following:

$$y^2 + axy + by = x^3 + cx^2 + dx + e \tag{4}$$

which is transformed into the following form by coordinate transformation

$$E : y^2 = x^3 + ax + b \tag{5}$$

where a, b, x, y belong to the finite field Fp, where p is a large prime number greater than 3. Suppose $P(x_1, y_1)$, $Q(x_2, y_2)$ are two points on the curve and $\Delta = (y_1 - y_2)/(x_1 - x_2)$ is the slope of line L that connects them. L intersects the elliptic curve exactly at another point $R(x_3, y_3)$, so R is the negative element of the sum of Q and P, which is $P + Q = -R$. Among them,

$$x^3 = \Delta^2 - x_1 - x_2 \tag{6}$$

$$y^3 = -x_1 + \Delta(x_1 - x_3) \tag{7}$$

2.1.4. Zero-Knowledge Proof Techniques

Zero-knowledge proof is mainly used to prove the range of encrypted data. The proposed scheme only uses interval proof [26] to ensure that the transaction amount is greater than 0.

Proof: Let t, l, s be three safe parameters, n be a large composite number unknown to factorization, g be a large order element in Z_n^*, and h be an element in the cyclic group generated by g.

Let $E = E(x, r) = g^x h^r$ be an FO promise that guarantees $x \in (a, b)$, where r is a random number selected from $\{-2^s n + 1, \cdots, 2^s n - 1\}$. The prover P makes the verifier V confident of $x \in (a, b)$ without knowing the value of x by the following step of FO commitment.

The protocol $PK\{x, r : E = g^x h^r mod n \wedge x \in (a, b)\}$ is defined as follows:

Step. 1 Make

$$y = x - a \tag{8}$$

Step. 2 Set P to

$$u = \alpha^2 y + \omega > 2^{t+l+s+T}\left(\alpha \neq 0, \omega \in \left(0, 2^{s+T}\right)\right) \tag{9}$$

and randomly select $r_1, r_2, r_3 \in \{-2^s n + 1, \cdots, 2^s n - 1\}$ so that $r_3 - r\alpha^2 + r_1\alpha + r_2 \in [-2^s n + 1, 2^s n - 1]$.

Calculate

$$E_1 = g^{x-a} h^r = g^y h^r mod n, \tag{10}$$

$$E_2 = E_1^{\alpha} h^{r_1} mod n, \tag{11}$$

$$E_3 = E_2^{\alpha} h^{r_2} mod n, \tag{12}$$

$$F = \mathrm{g}^{\omega} h^{r_3} mod n, \tag{13}$$

$$\mathrm{U} = \mathrm{g}^{\mathrm{u}} / E_3 = \mathrm{g}^{\omega} \mathrm{h}^{-\mathrm{r}\alpha^2 - \mathrm{r}_1\alpha - \mathrm{r}_2}\ mod n \tag{14}$$

and P sends $(u, E_2, E_3, \mathrm{F})$ to V.

Step. 3 V calculation:

$$E_1 = E(x, r)/g^a = g^y h^r mod n, \tag{15}$$

$$\mathrm{U} = \mathrm{g}^{\mathrm{u}} / \mathrm{E}_3 = \mathrm{g}^{\omega} \mathrm{h}^{-\mathrm{r}\alpha^2 - \mathrm{r}_1\alpha - \mathrm{r}_2} mod n. \tag{16}$$

Step. 4 Calculate P and V separately

$$PK_1\{\alpha, r_1, r_2 : E_2 = E_1^{\alpha} h^{r_1} mod n \wedge E_3 = E_2^{\alpha} h^{r_2} mod n\}, \tag{17}$$

$$PK_2\left\{\omega, r^{-r\alpha^2 - r_1\alpha - r_2} : F = g^{\omega} h^{r_3} mod n \wedge U = g^{\omega} h^{r - r\alpha^2 - r_1\alpha - r_2} mod n\right\}, \tag{18}$$

$$PK_3\left\{\omega, r_3 : F = g^{\omega} h^{r_3} mod n \wedge \omega \in \left[-2^{t+l+s+T}, 2^{t+l+s+T}\right]\right. . \tag{19}$$

Step. 5 V checks the correctness of $PK_i(i = 1, 2, 3)$ and $u > 2^{t+l+s+T}$ to trust $x > a$.

Similarly, set

$$y = b - x \tag{20}$$

and repeat steps 2 through 5 to prove $x < b$. □

2.2. Solution Model

Based on the decentralized characteristics of the blockchain system, this paper designs a decentralized and verifiable data trading scheme based on a blockchain. This scheme uses a distributed system, and users do not need any third party to participate in the process of data transaction, so as to ensure the authenticity of the data and the fairness of the transaction. Users obtain their own public and private key pairs through the decentralized key management scheme proposed by Yao et al. [27], verify the data according to the idea of inadvertent transmission to ensure the authenticity of the data, and use smart contracts to trade keys that decrypt the data to ensure fair trading. The data transaction model is shown in Figure 3.

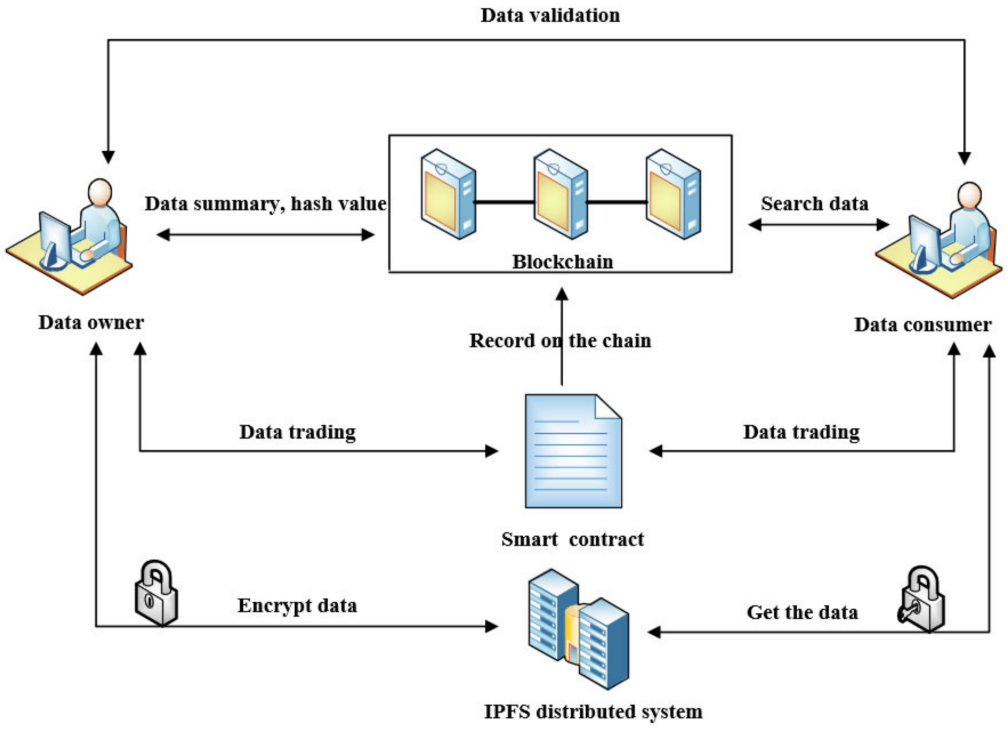

Figure 3. Data transaction model.

This solution model includes five entities: data owner, data consumer, blockchain, cloud storage platform (IPFS), and smart contract.

Data owner: The data owner is the person who owns the data and hopes to make a profit by selling data; they need to sell the data encryption, signature and hash operation, and the data, store the path information such as the registration on the blockchain for the convenience of consumers to find data, and store the data ciphertext on the IPFS's distributed file system.

Data consumer: The data consumer selects the data to be purchased by searching the data summary on the blockchain, then verifies the data through the oblivious transmission protocol, and finally verifies the transaction funds on the smart contract. After the verification is completed, the user's private key is obtained through the smart contract and the data are decrypted.

IPFS: the IPFS is responsible for storing the ciphertext of the encrypted data.

Blockchain: the blockchain is responsible for storing the hash digest, data storage path and data transaction records, while it is convenient for data traceability.

Smart contract: it verifies the legitimacy of the transaction amount and the key of the transaction data through zero-knowledge proof.

Our solution consists of four phases: user registration, processing, validation, and transaction data. The timing diagram of the trading scheme is shown in Figure 4.

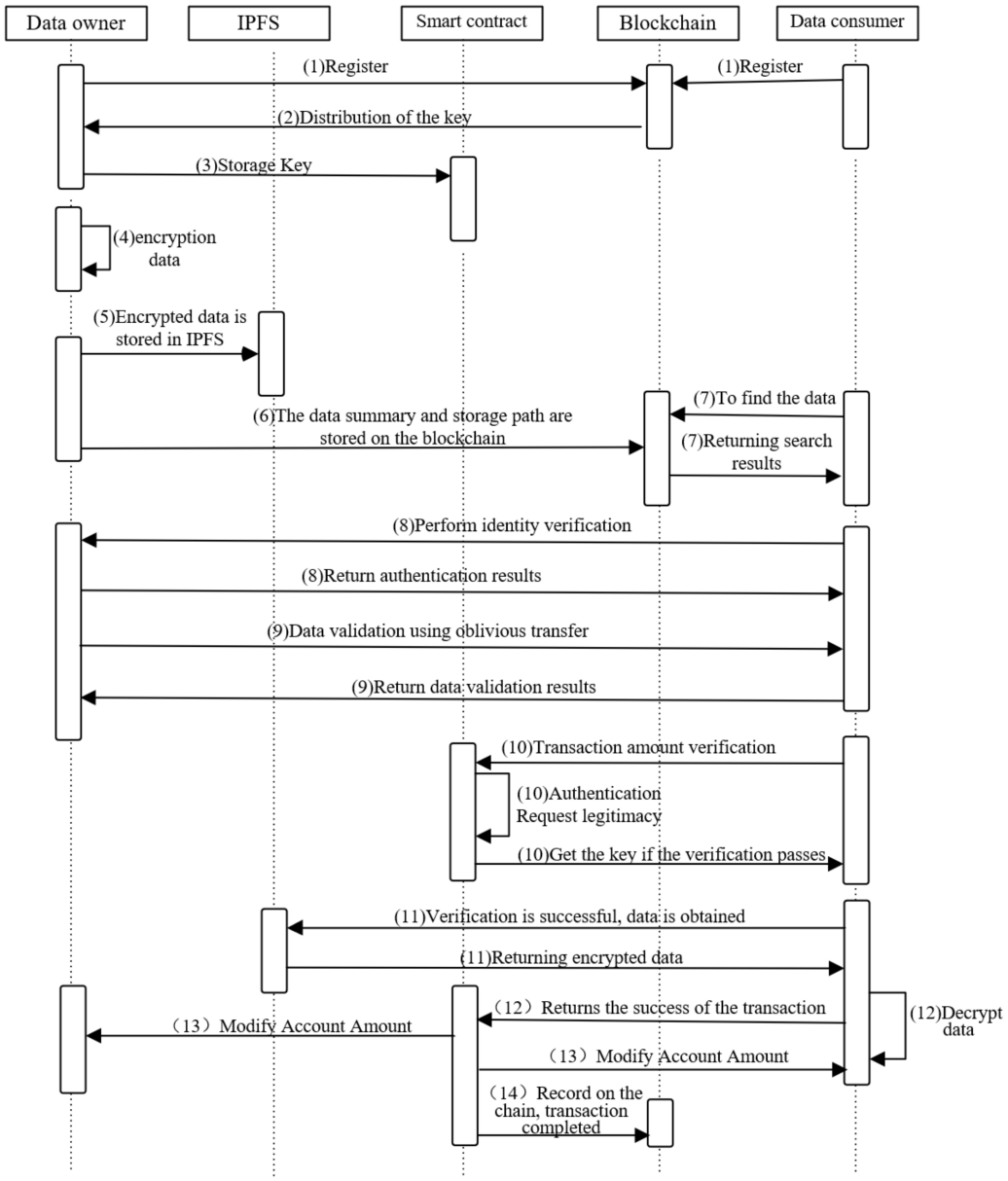

Figure 4. Time sequence diagram of decentralized data transactions.

User registration stage: users register their identity on the blockchain according to their own requirements, and then obtain their public and private key pairs according to the decentralized key management scheme, as shown in Figure 4, step (1) to step (3).

Data processing phase: The data owner encrypts the data, computes the hash of the plaintext and symmetric key, generates a summary of the data, and signs the summary. Then, the data ciphertext, plaintext, and the hash value of the symmetric key are signed and stored on the IPFS's distributed file system. The data summary and the signature of the summary are recorded on the blockchain. This is shown in Figure 4, step (4) to step (6).

Data verification stage: data consumers check the data summary on the blockchain, search for the data they need, find the data they need, verify the signature of the data owner of the data, verify the signature by inadvertently transferring the data, and then call the smart contract for the data transaction, as shown in Figure 4, step (7) to step (9).

Trading data phase: The smart contract validates the data in the consumer's account for the legitimate amount, after verification is passed, the account amount is frozen and sending the private key of the encrypted data to the consumer, unlocks the data for consumers to get the unencrypted data, after the verification data passed, changes the balance in the accounts of the owners and consumers, and finally saves the transaction records in the chain of blocks. This is shown in Figure 4, step (10) to step (14).

2.3. Program Overview

The four stages of this scheme are described as follows:

2.3.1. Initialization Phase

1. Users register as data owners or data consumers on the blockchain according to their own needs;
2. The data owner obtains their own public and private key pair according to the decentralized key management scheme;
3. If the user needs to purchase data, they find the data they need through the data digest on the blockchain;
4. If the user needs to sell the data, the data are encrypted, hashed, and signed, the data ciphertext and signature are uploaded to the IPFS distributed file system, and the data summary is recorded on the blockchain.

2.3.2. Data Encryption Phase

After the data owner registers their identity on the blockchain, they encrypt the data they sell. They first use the symmetric encryption algorithm to encrypt the data, and then use the asymmetric encryption algorithm to encrypt the symmetric key to ensure the confidentiality of the data. They calculate the hash of the plaintext fragment and the symmetric key and record it on the blockchain to ensure that the data have not been tampered with. They sign the summary and register it as the source of the guaranteed data on the blockchain.

The detailed steps for the data encryption are as follows:

Step. 1 The data owner divides the data D that needs to be sold into n equal parts: $\{d_i\}_{i\in\{1,\ldots,n\}}$;

Step. 2 The data owner uses the AES encryption algorithm to generate n symmetric keys m_i and uses these symmetric keys to encrypt the split file:

$$C_i = Enc(m_i, d_i) \tag{21}$$

Step. 3 The data owner uses a collision-resistant hash function to hash the plaintext and symmetric key:

$$h_{d_i} = H(d_i) \tag{22}$$

$$h_{m_i} = H(m_i); \tag{23}$$

Step. 4 The data owner encrypts symmetric key m_i using an improved homomorphic encryption algorithm as follows:

(1) The data owner locally generates an elliptic curve E and a random base point G on the curve, and at the same time chooses different private keys $(k_1, k_2, \ldots, k_n)$ to generate a public-key-encrypted plaintext to enhance the security of the whole plaintext;

(2) The data owner multiplies the base point G with the private key $(k_1, k_2, \ldots, k_n)$ to generate the public key $(Q_1, Q_2, \ldots, Q_n)$, where $Q_i = G * k_i$; the client saves the private key $(k_1, k_2, \ldots, k_n)$ to local storage;

(3) In order to encrypt the symmetric key $(m_1, m_2, \ldots, m_n)$, the data owner should embed the symmetric key into the selected elliptic curve E to obtain the symmetric key text point $(P_{m_1}, P_{m_2}, \ldots, P_{m_n})$;

(4) The data owner randomly generates an integer $(r_1, r_2, \ldots, r_n)$, where the random number $r < n$ and n are the order of the base point G. Then, the public key $(Q_1, Q_2, \ldots, Q_n)$, the random number $(Q_1, Q_2, \ldots, Q_n)$, and the base point G are used to encrypt the plaintext point $(P_{m_1}, P_{m_2}, \ldots, P_{m_n})$

$$S_{1_i} = r_i * G \tag{24}$$

$$S_{1_i} = S_{3_i} * P_{m_i} \tag{25}$$

$$S_{3_i} = r_i * Q_i \tag{26}$$

The encrypted symmetric key ciphertext is: $S_1(S_{1_1}, S_{2_1}) \ldots S_n(S_{1_n}, S_{2_n})$;

(5) The local client of the data owner sends ciphertext $(C_1, C_2, \cdots, C_n)$ and symmetric key ciphertext $(S_1, S_2, \cdots, S_n)$ to the IPFS's distributed file system for storage.

2.3.3. Transaction Verification Phase

This stage is the core part of the transaction, which realizes the verification of the identity's legitimacy, data authenticity, and amount of legitimacy of both parties through a three-step verification. If all three steps are passed, the data transaction is carried out. The specific steps are as follows:

Step. 1 Identity account verification

The data consumer initiates a transaction request to the data owner, and the two transactions first carry out identity authentication. The data consumer generates its own public and private key pair $(PK_a, SK_a) = F(psy_{ID}, PK_b)$ by using the public and private key pair (PK_b, SK_b) of the data owner and the private password of its own account (psy_{ID}). The specific verification steps are as follows:

(1) The data consumer submits an authentication request for an identity account to the data owner.

(2) The data owner asks the data consumer to prove the account they own: the data owner looks for the corresponding PK_a, uses it to encrypt a random number r_1 : $S_1 = Encode(r_1|PK_a)$, and then returns S_1 and their PK_b to the data consumer.

(3) The data consumer proves they have an account:

   a. The data consumer uses $F(psy_{ID}, PK_b)$ to generate (PK_a, SK_a) and then decrypts $\widetilde{r}_1 = Decode(S_1|SK_a)$;

   b. The data consumer gets $\widetilde{r}_1$, picks another random number r_2, encrypts $\widetilde{r}_1$ and r_2 : $S_2 = Encode(\widetilde{r}_1|PK_b)$ using the data owner's public key PK_b, and returns S_2 and S_3 to the data owner.

(4) The data owner authenticates the account identity of the data consumer and proves that they own the data:

   a. The data owner decrypts $\widetilde{r}_1 = Decode(S_2|SK_b)$, and if $\widetilde{r}_1$ is equal to r_1, the identity authentication of the data consumer is passed;

   b. The data owner decrypts $r_2 = Decode(S_3|SK_b)$ and then uses r_2 as a factor of symmetric encryption C to transmit the following normal communication content to the data consumer encryption n: $N = C_{r_2}(n)$.

(5) The data consumer authenticates the identity of the data owner:

The data consumer uses r_2 as the factor of symmetric encryption C to decrypt N: $n = C_{r_2}^{-1}(N)$, if n is the content with normal semantics, then the other party has PK_b corresponding to SK_b.

The data consumer continues communication with the data owner using r_2 as a factor of symmetric encryption C.

After the authentication is completed, the data consumer encrypted the signature of the transaction information and sent it to the data owner, who decrypts and verifies the transaction information and sends his account address and signature to the data consumer with the data consumer's public key, who decrypts and verifies the received information. After confirming the correctness, they proceed to the next Step.

Step. 2 Transaction data verification

After finding the data they need from the blockchain, the data consumer sends a verification request to the data owner, and then transfers the verification data to the data owner inadvertently. The specific steps are as follows:

(1) The data owner and the data consumer generate t random numbers $\{x_{o1}, x_{o2}, \cdots x_{ot}\}$ and $\{x_{p1}, x_{p2}, \cdots x_{pt}\}$, respectively, and compute the hash $h_{oi} = H(x_{oi}), h_{pi} = H(x_{pi})$, of these random numbers;

(2) The data owner and the data consumer exchange the hashes h_{oi} and h_{pi} of the random numbers, and then exchange the generated random numbers $\{x_{o1}, x_{o2}, \cdots x_{ot}\}$ and $\{x_{p1}, x_{p2}, \cdots x_{pt}\}$ to determine that the sequence number of the data to be verified is $xx_i = (x_{oi} + x_{pi}) mod n$, where n is the fraction of the data segmentation;

(3) The data owner combines the symmetric key k_{xx_i} corresponding to the sequence number of the verification data with the random number r_{xx_i} used to encrypt the symmetric key using the homomorphic encryption algorithm, and sends it to the data consumer after encryption with the public key of the data consumer (dc):

$$dc = Enc_{ecc}(pk_{ecc}, k_{xx_1}||k_{xx_2}|| \ldots ||k_{xx_t}||r_{xx_1}||r_{xx_2}|| \ldots ||r_{xx_t}) \tag{27}$$

(4) After using its private key dc, the data consumer uses its symmetric key to decrypt the downloaded ciphertext to obtain the plaintext m_{xx_j}:

$$m_{xx_j} = Dec\left(k_{xx_j}, c_{xx_j}\right)_{j \in \{1,2,\ldots,t\}} \tag{28}$$

(5) The data consumer checks whether the plaintext after decryption is consistent with the data summary, uses the obtained random number r_{xx_i} to encrypt the symmetric key again with the homomorphic encryption algorithm, checks whether the ciphertext is consistent, and calculates the hash of the plaintext and symmetric key to check whether they are consistent.

Step. 3 Transaction amount verification

The smart contract verifies the validity of the transaction amount, and the transaction amount is protected by Paillier's homomorphic encryption. To verify whether the transaction amount is greater than 0 and whether the balance of the data consumer's account is greater than the transaction amount, the specific steps are as follows: The data consumer uses the interval proof in the zero-knowledge proof to prove to the smart contract that the transaction amount is $e > 0$, that is, set

$$y = x - a \tag{29}$$

to

$$y = e - 0 \tag{30}$$

(1) The data consumer sets

$$u = \alpha^2 y + \omega > 2^{t+l+s+T}\left(\alpha \neq 0, \omega \in \left(0, 2^{s+T}\right)\right), \tag{31}$$

randomly selects $r_1, r_2, r_3 \in \{-2^s n + 1, \cdots, 2^s n - 1\}$, makes $r_3 - r\alpha^2 + r_1\alpha + r_2 \in [-2^s n + 1, 2^s n - 1]$, calculates

$$E_1 = g^{e-a}h^r = g^y h^r modn, \tag{32}$$

$$E_2 = E_1^{\alpha} h^{r_1} modn, \tag{33}$$

$$E_3 = E_2^{\alpha} h^{r_2} modn, \mathrm{F} = \mathrm{g}^{\omega} h^{r_3} modn, \tag{34}$$

$$\mathrm{U} = \mathrm{g}^{\mathrm{u}}/E_3 = \mathrm{g}^{\omega}\mathrm{h}^{-\mathrm{r}\alpha^2 - \mathrm{r}_1\alpha - \mathrm{r}_2} modn \tag{35}$$

and then the data consumer sends (u, E_2, E_3, F) to the smart contract;

(2) The smart contract computes:

$$E_1 = E(e, r)/g^a = g^y h^r modn, \tag{36}$$

$$U = g^u/E_3 = g^{\omega} h^{-r\alpha^2 - r_1\alpha - r_2} modn; \tag{37}$$

(3) The data consumer and smart contract's keys are calculated separately

$$PK_1\{\alpha, r_1, r_2 : E_2 = E_1^{\alpha} h^{r_1} modn \wedge E_3 = E_2^{\alpha} h^{r_2} modn\}, \tag{38}$$

$$PK_2\left\{\omega, \mathrm{r}^{-\mathrm{r}\alpha^2 - \mathrm{r}_1\alpha - \mathrm{r}_2} : F = \mathrm{g}^{\omega}\mathrm{h}^{\mathrm{r}_3} modn \wedge U = \mathrm{g}^{\omega}\mathrm{h}^{\mathrm{r} - \mathrm{r}\alpha^2 - \mathrm{r}_1\alpha - \mathrm{r}_2} modn\right\}, \tag{39}$$

$$PK_3\left\{\omega, r_3 : F = \mathrm{g}^{\omega}\mathrm{h}^{\mathrm{r}_3} modn \wedge \omega \in \left[-2^{t+l+s+T}, 2^{t+l+s+T}\right]\right\}; \tag{40}$$

(4) The smart contract checks the correctness of $PK_i (i = 1, 2, 3)$ and $u > 2^{t+l+s+T}$ to believe $x > a$, which is $e > 0$. If the verification is successful, it proves that the transaction amount is greater than 0.

Verifying whether the account balance of the data consumer is greater than the transaction amount can be translated into whether the account balance of the data consumer is greater than 0 after the transaction. Therefore, it can be proved that the account balance of the data consumer is greater than 0 after the transaction according to the above four steps.

If both of the two interval proofs pass, the next step is to verify the legitimacy of the transaction amount.

The correctness of the transaction amount is verified by the addition homomorphic property of Paillier. It consists in verifying whether the transaction amount of the data consumer plus the account balance after the transaction is equal to the current account balance and whether the current account balance of the data owner in the transaction plus the transaction amount is equal to the account balance after the transaction is completed, that is, whether the following formula is valid:

$$Enc_p\left(SUM_{presentA}\right) = Enc_p(SUM_A) + Enc_p(m), \tag{41}$$

$$Enc_p(SUM_B) = Enc_p\left(SUM_{presentB}\right) + Enc_p(m). \tag{42}$$

If the above verification is successful, the transaction verification is successful, the smart contract verifies the legitimacy of the transaction, and the next step of data trading is carried out. On the other hand, if the verification fails, the smart contract returns the verification failure to the data consumer and terminates the transaction.

Step. 4 Data decryption phase

The data consumer decrypts the symmetric key using the private key $(k_1, k_2, \ldots, k_n)$ of the smart contract transaction

$$\begin{gathered} \prod_{i=1}^{n} k_i * \prod_{i=1}^{n} S_{1_i} = k_1 * k_2 \cdots k_n * S_{1_1} * S_{1_2} = \\ k_1 * k_2 \cdots k_n * r_1 * r_2 \cdots r_n * G = \prod_{i=1}^{n} r_i * \prod_{i=1}^{n} Q_i = \prod_{i=1}^{n} S_{3_i} \end{gathered} \tag{43}$$

$$\begin{gathered}\prod_{i=1}^{n} S_{2_i} * \prod_{i=1}^{n} S_{3_i}^{-1} = \\ S_{3_1} P_{m_1} * S_{3_i} P_{m_2} \cdots S_{3_i} P_{m_n} * S_{3_1}^{-1} * S_{3_2}^{-1} \cdots S_{3_n}^{-1} = \\ P_{m_1} * P_{m_2} \cdots P_{m_n} = \prod_{i=1}^{n} P_{m_i}\end{gathered} \tag{44}$$

Since the ciphertext decrypted by the data consumer is the product of point $\prod_{i=1}^{n} P_{m_i}$ embedded in the elliptic curve, the point is finally decoded into symmetric key $(m_1, m_2, \ldots, m_n)$. After obtaining the symmetric key, the data ciphertext obtained from the IPFS is decrypted to obtain the required data plaintext.

The data consumer checks whether the decrypted plaintext matches the data summary. If it does, the transaction is written to the blockchain through a smart contract and the account balance of the data consumer and the data owner is updated to $Enc_p(SUM_A)$ and $Enc_p(SUM_A)$, respectively. Otherwise, a smart contract is used to terminate the transaction. Figure 5 shows the smart contract:

```
Contract

Contract Variable
Use1:Data Owners
Use2:Data Consumers
Topic:Data Trading
Price:e
Data Owner.Deposit
Data Consumers.Deposit

Contract Content
Function Transaction(){
        freeze Data Owner.Deposit;
        freeze Data Consumers.Deposit;
        while(Data Consumers.Deposit>e){
                if(Data Consumers received data){
                        Pay Deposit from Data
                        Consumers to Data Owner;
                        Transaction records are
                        written to the blockchain;
                }
                else{
                        Transaction termination;
                   }
        else{
                Transaction termination;
           }
```

Figure 5. Smart contract pseudocode.

3. Results and Discussion

In this scheme, a secondary encryption was used to protect the data. Since the transaction data were long data, and the key length generated by symmetric encryption was short, we first used symmetric encryption for the encryption, and then carried out a secondary encryption on the symmetric key and used homomorphic encryption improved with ECC to encrypt the key. Three steps were used to verify the legitimacy of the data transaction. If any step failed to pass the verification, the transaction was considered illegal and the transaction was immediately stopped. The first step was to verify the identity and confirm that both parties were legitimate users through a two-way verification. The second step was to verify the data by using the idea of inadvertent transmission. The third step verification was the verification of the transaction amount, where the noninteractive zero-knowledge proof was used to verify whether the account balance provided by the user after the transaction was legitimate. In this section, the security, privacy, and efficiency

of the proposed scheme are analyzed, and the security and privacy of the proposed scheme are demonstrated.

3.1. Security Analysis

In this scheme, data encryption was designed based on AES symmetric encryption and improved homomorphic encryption. The homomorphic encryption was combined with an elliptic curve. The security of the elliptic curve encryption algorithm was based on the difficulty of solving the ECDLP.

1. Public-key substitution attack

When the attacker is the data owner and falsifies the data to conduct transactions with other data consumers, in the first step of the transaction verification, if the data owner does not provide the identity information of the data consumers to conduct transactions, the verification in the first step will not pass. Because it can provide identity information, the initial verification can pass, but because the data are forged, it cannot pass the transaction data verification in step 2, so the security of the user account cannot be threatened. Therefore, the scheme can resist a public-key substitution attack.

2. Tampering attack

The transaction amount in this scheme is the homomorphic encrypted amount, and the transaction is completed through the smart contract, so it is not feasible if one of the two parties tries to tamper with the transaction amount. Secondly, if the transaction amount is successfully tampered with, the transaction amount verification of the smart contract also verifies the legitimacy of the transaction amount, and if the transaction amount is illegally tampered with, it cannot pass the third step of the transaction amount's legitimacy verification, so this scheme can resist tampering attacks.

3. Safety strength

The security of ECC depends on the solution of discrete logarithms on the group of elliptic curves, and the difficulty of solving discrete logarithms of elliptic curves is much greater than the difficulty of decomposing large prime numbers, so the homomorphic encryption based on elliptic curve encryption technology has a higher security compared with RSA homomorphic encryption. For example, for curve $y^2 = x^3 + x + 1$, when the finite field is given a prime number p = 11, its scattering point distribution is shown in Figure 6; there are as many as 14 scattering points including infinity points. When the finite field is given a prime number p = 23, its scattering point distribution is shown in Figure 7; there are as many as 28 scattering points including infinity points, presenting a strong disorder and dispersion. The number of scattered points in the actual project is in the hundreds, so our proposed encryption algorithm is sufficient to ensure the effectiveness of the encryption algorithm for a practical application and can better ensure the security of data transactions.

3.2. Performance Analysis

Compared with general encryption methods, using homomorphic encryption can operate directly on the ciphertext, which means that data security and data privacy are ensured and the ciphertext transmission rate is increased. Since the key size and system parameters of elliptic curve encryption technology are much smaller compared to RSA and DSA, the storage space occupied by ECC is much smaller. As can be seen from Table 1, the security strength of the elliptic curve cryptography algorithm with a 160-bit key is equivalent to the security strength of the RSA and DSA algorithms with a 1024-bit key, which effectively solves the problem of difficult engineering implementation caused by increasing the key length to improve the security strength. The use of different private keys k for plaintext encryption makes the security of the method greatly improved on the basis of the security of elliptic curve cryptography. Therefore, the homomorphic encryption

method based on ECC has a better transmission rate with a higher security compared to RSA and DSA methods.

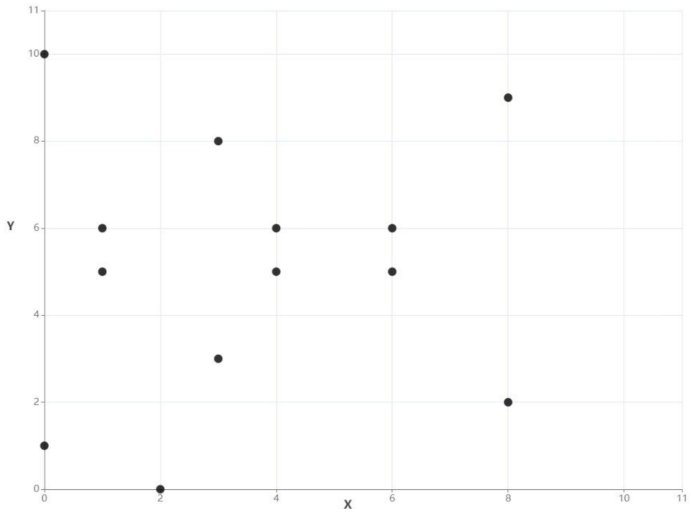

Figure 6. Scatter plot at $p = 11$.

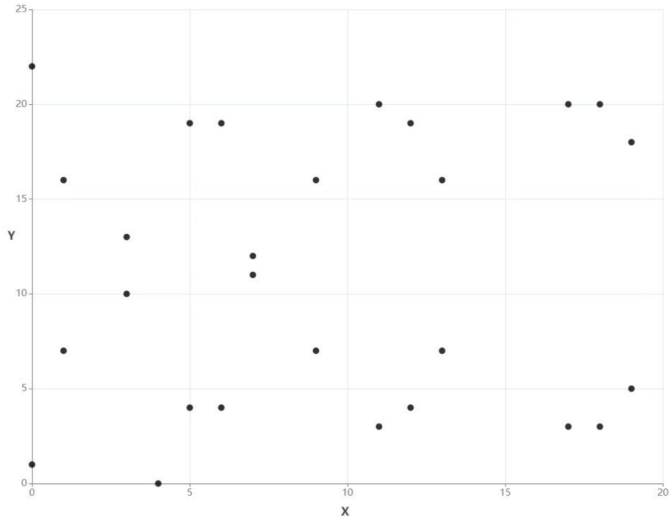

Figure 7. Scatter plot at $p = 23$.

Table 1. Required key length for the same decryption time ECC/RSA/DSA.

Deciphering Time/Years	Key Length of RSA, DSA/Bit	Key Length of ECC/Bit	Key Length Ratio of RSA, ECC
10^4	512	106	5:1
10^8	768	132	6:1
10^{11}	1024	160	7:1
10^{20}	2048	210	10:1
10^{78}	21,000	600	35:1

3.3. Efficiency Analysis

We compared the schemes in the literature [6–10] with our scheme, and the results are shown in Table 2. Let us first assume that all the schemes' data are divided into 100 pieces, and we only need to select 3 of them for verification. Let us assume the symmetric encryption and decryption time is t_1, the hash function time is t_2, the public key encryption

time is t_3, the private key decryption time is t_4, the smart contract time is t_5, the transaction time is t_6, and the communication time is t_7.

Table 2. Scheme efficiency.

Scheme	Process the Data	Validation Data	Trading Data
Soubhagya [5]	$100t_1+100t_2$	N/A	$5t_1+3t_3+4t_4+4t_7$
Guo [7]	$10t_2+7t_3+t_4$	N/A	$10t_2+12t_3+2t_4+4t_7$
Ren [8]	$t_1+6t_2+2t_3+2t_4+3t_7$	N/A	$9t_2+14t_3+5t_4$
Segura [9]	$100t_3$	$3t_7$	t_6
Kiyomoto [10]	$10t_2+100t_3+100t_4$	$3t_1+3t_7$	$100t_4+t_7$
This scheme	$100t_1+200t_2+100t_3+2t_4$	$3t_4+3t_7$	t_5

As can be seen from Table 2, compared to the other schemes, our scheme is more expensive in terms of data processing only, because in scheme [5], we encrypt the data only once, which is not conducive to the transaction of encryption keys. Schemes [7,8] do not split the data, and scheme [9] uses a public key to encrypt the split data directly. Scheme [10] is similar to our scheme but does not use enough hash functions to ensure that the data will not be tampered with and does not use signatures to ensure the source of the data. Our scheme uses secondary encryption to protect data security and has little impact on data transaction efficiency. All in all, our scheme only sacrifices trivial efficiency but increases the security of data transactions.

3.4. Scheme Comparison

In this paper, we achieved secure data transactions while protecting data privacy. The decentralization and traceability of the blockchain were utilized to achieve data transactions without relying on third parties. The comparison of this paper with other schemes is shown in Table 3.

Table 3. Option comparison.

Scheme	Data Validation	Distributed	Blockchain	Transaction Complexity	Secondary Encryption
Soubhagya [5]	No	Yes	Yes	complex	No
Guo [7]	Yes	Yes	Yes	complex	No
Ren [8]	Yes	Half	No	\	No
Segura [9]	Yes	Yes	Yes	simple	Yes
Kiyomoto [10]	Yes	Half	Yes	\	No
This scheme	Yes	Yes	Yes	simple	Yes

"Half": semidistributed; "\": no trading part.

Scheme [8] validates data through a trusted third party, schemes [9,10] and our scheme use a one-to-one approach to validate data, and none of the other schemes provide a data validation method. In data trading centers, the authenticity of the data must be guaranteed, so a data validation method must be designed to suit both sides of the transaction, and if the data are validated through a trusted third party, the data will inevitably be leaked to the trusted third party, which may harm the interests of the data owner.

Compared with other schemes, only scheme [9] and our scheme are truly distributed and adopt secondary encryption to protect data security, which ensures that a single point of failure will not occur in the scheme and greatly strengthens data security.

In scheme [6], the transaction adopts a method similar to bank charging, but the method is slightly complex. In scheme [7], an agent re-encryption is combined with zero-knowledge proof, and the transaction process is complex. Schemes [8,10] do not provide a charging method, but scheme [9] and our scheme provide a simple charging method. As an integral part of fee-based data trading centers, it can be complicated for data owners and data consumers to trade encryption keys if transactions are conducted through banks.

The transaction scheme in [9] cannot guarantee the legitimacy of the transaction amount, while we used smart contracts to verify the legitimacy of the transaction amount.

4. Conclusions

In order to solve the problem of security and fairness in data transactions, some measures have been proposed to guarantee the fairness and security of data transactions through trusted third parties or arbitration. However, these recommendations are vulnerable to single points of failure and may reveal useful data information. We designed a fair and verifiable data trading scheme, which did not require third-party participation, and the whole process of data trading is only between data owners and data consumers, with all transaction data on the chains and private data business operations and sharing completed in on-chain smart contracts, making full use of the decentralized and non-tamperable features of blockchain technology. Data were encrypted with secondary encryption, and transaction data were long data, so the efficiency of asymmetric encryption was low. We adopted AES symmetric encryption, and the symmetric key generated by the data encryption adopted an improved homomorphic encryption for encryption protection. The data consumer could obtain part of the data for verification, but it was difficult for the data consumer to obtain all the plaintext data, even if the verification was performed multiple times. When the data consumer verified that the data were fine, the smart contract payment currency was used to obtain the key to encrypt the data. Once the data owner received the currency, the data consumer obtained the encryption key immediately, and the smart contract wrote the transaction record to the blockchain for easy traceability. We proved the high transmission efficiency, short key, and high security strength of both elliptic curve encryption technology algorithms of our scheme through a security analysis.

The proposed solution used a blockchain, which has excellent features, such as being tamper-evident, but is not good enough in terms of data processing efficiency. In future research, we hope to improve the theory and methods of blockchain operation efficiency, such as using more efficient algorithms, as well as studying the possibility of exchanging security and efficiency on the blockchain to further improve the efficiency of the scheme. In addition, in the fair and verifiable data trading scheme, the scheme of data verification can greatly guarantee the authenticity of the data, but it does not completely guarantee that the data are not adulterated with invalid data. In the future, we hope to complete the work of data verification through machine learning, AI, and other technologies to ensure the authenticity of data in specific application scenarios, without leaking data content to data consumers.

Author Contributions: Y.J. participated in the feasibility discussion, analysis of the paper scheme, and the proofreading of the paper; G.S. was responsible for the overall design, performance analysis, and paper writing; T.F. supervised the formulation of the scheme and reviewed and revised the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (grant nos. 62162039, 61762060), Foundation for the Key Research and Development Program of Gansu Province, China (grant no. 20YF3GA016).

Data Availability Statement: The data used to support the findings of this study are included within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Deepa, N.; Pham, Q.V.; Nguyen, D.C.; Bhattacharya, S.; Prabadevi, B.; Gadekallu, T.R.; Maddikunta, P.K.; Fang, F.; Pathirana, P.N. A survey on blockchain for big data: Approaches, opportunities, and future directions. *Future Gener. Comput. Syst.* **2022**, *131*, 209–226. [CrossRef]
2. Ke, X. Ternary Governance of Data Transaction Circulation: Technology, Standards and Law. *J. Jishou Univ. (Soc. Sci. Ed.)* **2022**, *43*, 96.
3. Xu, L. Analysis of the application of blockchain in data transactions. *Think Tank Times* **2018**, 38–39.

4. Jian, Y.; Zhong, W. Research on Personal Data Traceability Management System in Big Data Environment. *Inf. Sci.* **2016**, *34*, 139–143.
5. Sun, G.; Li, Z.; Xiao, R.; Yang, J.; Wang, X. Research on Blockchain Transaction Security. *J. Nanjing Univ. Posts Telecommun. Nat. Sci. Ed.* **2021**, *41*, 36–48.
6. Mallick, S.R.; Sharma, S. EMRI: A scalable and secure Blockchain-based IoMT framework for healthcare data transaction. In Proceedings of the 2021 19th OITS International Conference on Information Technology (OCIT), Bhubaneswar, India, 16–18 December 2021; pp. 261–266. [CrossRef]
7. Guo, H.; Cheng, J.; Wang, J.; Chen, T.; Yuan, Y.; Li, H.; Sheng, V.S. IoT Data Blockchain-Based Transaction Model Using Zero-Knowledge Proofs and Proxy Re-encryption. In Proceedings of the International Conference on Artificial Intelligence and Security, Qinghai China, 22–26 July 2022; Springer: Cham, Switzerland, 2022; pp. 573–586.
8. Hwang, R.J.; Lai, C.H. Provable fair document exchange protocol with transaction privacy for ecommerce. *Symmetry* **2015**, *7*, 464–487. [CrossRef]
9. Delgado-Segura, S.; Pérez-Solà, C.; Navarro-Arribas, G.; Herrera-Joancomartí, J. A fair protocol for data trading based on Bitcoin transactions. *Future Gener. Comput. Syst.* **2019**, *107*, 832–840. [CrossRef]
10. Kiyomoto, S.; Fukushima, K. Fair-trading protocol for anonymised datasets requirements and solution. In Proceedings of the 2018 4th International Conference on Information Management (ICIM), Oxford, UK, 25–27 May 2018; pp. 13–16.
11. Wang, D.; Gao, J.; Yu, H.; Li, X. A Novel Digital Rights Management in P2P Networks Based on Bitcoin System. In Proceedings of the International Conference on Frontiers in Cyber Security. IEEE, Chengdu, China, 5–7 November 2018; Springer: Singapore, 2018; pp. 227–240.
12. Zhao, Y.; Yu, Y.; Li, Y.; Han, G.; Du, X. Machine learning based privacy-preserving fair data trading in big data market. *Inf. Sci.* **2019**, *478*, 449–460. [CrossRef]
13. Missier, P.; Bajoudah, S.; Capossele, A.; Gaglione, A.; Nati, M. Mind My Value: A decentralized infrastructure for fair and trusted IoT data trading. In Proceedings of the Seventh International Conference on the Internet of Things. ACM, Linz, Austria, 22–25 October 2017; p. 15.
14. Alrawahi, A.S.; Lee, K.; Lotfi, A. Trading of cloud of things resources. In Proceedings of the Second International Conference on Internet of things and Cloud Computing. ACM, Cambridge, UK, 22–23 March 2017; p. 163.
15. Lin, S.J.; Liu, D.C. A fair-exchange and customer-anonymity electronic commerce protocol for digital content transactions. In Proceedings of the International Conference on Distributed Computing and Internet Technology, Bangalore, India, 17–20 December 2007; Springer: Berlin/Heidelberg, Germany, 2007; pp. 321–326.
16. Cattelan, R.G.; He, S.; Kirovski, D. Prototyping a novel platform for free-trade of digital content. In Proceedings of the 12th Brazilian Symposium on Multimedia and the Web. ACM, Natal Rio Grande do Norte, Brazil, 19–22 November 2006; pp. 79–88.
17. Perera, C. Sensing as a service (S2aaS): Buying and selling IoT data. *arXiv* **2017**, arXiv:1702.02380.
18. Lin, S.J.; Liu, D.C. An incentive-based electronic payment scheme for digital content transactions over the Internet. *J. Netw. Comput. Appl.* **2009**, *32*, 589–598. [CrossRef]
19. Huang, Z.; Su, X.; Zhang, Y.; Shi, C.; Zhang, H.; Xie, L. A decentralized solution for IoT data trusted exchange based-on blockchain. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), IEEE, Chengdu, China, 13–16 December 2017; pp. 1180–1184.
20. Fan, C.I.; Juang, W.S.; Chen, M.T. Efficient fair content exchange in cloud computing. In Proceedings of the 2010 International Computer Symposium (1CS2010), IEEE, Tainan, Taiwan, 16–18 December 2010; pp. 294–299.
21. Qian, W.; Qi, S. A Fair Transaction Protocol with an Offline Semi-Trusted Third Party. In *Advances in Intelligent Decision Technologies*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 249–257.
22. Bensitel, Y.; Romadi, R. Secure data storage in the cloud with homomorphic encryption. In Proceedings of the International Conference on Cloud Computing Technologies and Applications IEEE, Marrakech, Morocco, 24–26 May 2017; pp. 1–6.
23. Li, Z.; Zhang, F.; Wang, P. Highly efficient fully homomorphic encryption scheme with shorter publickeys. *Comput. Appl. Res.* **2017**, *34*, 487–489. (In Chinese)
24. Zou, Y. Research on cloud storage encryption based on ellip-tic curve. *Cyberspace Secur.* **2017**, *8*, 21–23. (In Chinese)
25. Huang, R. Information security system based on ellip-tic curve encryption algorithm. *J. Neijiang Teac-Hers Coll.* **2017**, *32*, 72–76. (In Chinese)
26. Wu, Q.; Zhang, J.; Wang, Y. Simple proofthat a committed number is in a specific interval. *ActaElectronica Sin.* **2004**, *32*, 1071–1073. (In Chinese)
27. Yao, Y.; Chang, X.; Zhen, P. Decentralized Identity Authentication and Key Management Scheme Based on Blockchain. *Cyberspace Secur.* **2019**, *10*, 33–39.