future internet

Article

FaDe: A Blockchain-Based Fair Data Exchange Scheme for Big Data Sharing

Yuling Chen [1,2], Jinyi Guo [3], Changlou Li [3] and Wei Ren [1,3,4,*]

[1] State Laboratory of Public Big Data, GuiZhou University, Guizhou 550025, China; ylchen3@gzu.edu.cn
[2] College of Computer Science and Technology, GuiZhou University, Guizhou 550025, China
[3] School of Computer Science, China University of Geosciences (Wuhan), Wuhan 430074, China; jinyi_g@cug.edu.cn (J.G.); lichanglou@cug.edu.cn (C.L.)
[4] Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences (Wuhan), Wuhan 430074, China
* Correspondence: weirencs@cug.edu.cn; Tel.: +86-027-67883716

Received: 31 July 2019; Accepted: 16 October 2019; Published: 24 October 2019

Abstract: In the big data era, data are envisioned as critical resources with various values, e.g., business intelligence, management efficiency, and financial evaluations. Data sharing is always mandatory for value exchanges and profit promotion. Currently, certain big data markets have been created for facilitating data dissemination and coordinating data transaction, but we have to assume that such centralized management of data sharing must be trustworthy for data privacy and sharing fairness, which very likely imposes limitations such as joining admission, sharing efficiency, and extra costly commissions. To avoid these weaknesses, in this paper, we propose a blockchain-based fair data exchange scheme, called FaDe. FaDe can enable de-centralized data sharing in an autonomous manner, especially guaranteeing trade fairness, sharing efficiency, data privacy, and exchanging automation. A fairness protocol based on bit commitment is proposed. An algorithm based on blockchain script architecture for a smart contract, e.g., by a bitcoin virtual machine, is also proposed and implemented. Extensive analysis justifies that the proposed scheme can guarantee data exchanging without a trusted third party fairly, efficiently, and automatically.

Keywords: big data transaction; blockchain; fair payment protocol; bitcoin; bit commitment

1. Introduction

The rise of new computing paradigms has accumulated huge amounts of data, such as cloud computing, ubiquitous computing, mobile internet, and the Internet of Things. The big data era is coming, while data begin to be endowed with value, e.g., consumer behavior data for new business models, crowd sensing data for traffic predication, participant sensing for social computing, and so on. Big data provide a large amount of tagged data for machine learning and deep learning, and thus foster many new applications [1]. As data are distributed in various management domains, data sharing across different domains is a forthcoming demand for many scenarios.

For big data sharing, the current main method relies on centralized data markets that conduct all management operations, including data storage, data sharing, data transaction, and data dissemination. It results in several limitations as follows: users have to fully trust data markets in terms of data privacy; the efficiency is damaged due to the centralized management and intermediate forwarding; extra commission may be costly for users in the long term [2].

To solve the problems above, we propose a decentralized manner without a trusted third party. This can improve the efficiency, and exempt the extra commission fee, but it then causes another challenge to arise—the fairness of data exchange in data transactions. Although some new methods for big data sharing have been proposed [3,4], there are few comprehensive methods that can guarantee various requirements, e.g., legal data transaction, fair data sharing, and subtle privacy protection. We observe that blockchain technologies, e.g., BTC (Bitcoin), Ethereum, and especially smart contracts, can be employed for creating and implementing fair protocols in data exchanges and data transaction, by automatically executing the consented source codes and finding data in the ledger [5]. Furthermore, the blockchain network is a P2P (Point-To-Point) network, which makes data sharing easier without looking for a trusted data market. That is, data owners can conduct data transactions with data buyers autonomously, on demand, anytime and anywhere. It greatly promotes the data sharing between data owners and data buyers for utilizing data values conveniently, and avoids the leakage of data privacy to a third party as well.

In P2P data transaction processes between data owners and data buyers, when data owners provide required data to data buyers, data buyers must pay the corresponding fee to data owners. Inversely, if data buyers pay a required fee to data owners, data owners must provide the corresponding data to data buyers. This is elementally fair. To solve this problem, simply speaking, we suggest that data buyers can choose a random part of data to read before deciding whether to continue buying it or not. The selection of such trial data has two advantages: it allows data buyers to evaluate the availability and authenticity of intended data, and it does not damage the profit of data owners due to few data leakages.

In this approach, we still encounter two challenges: designing a protocol that can be implemented in the BTC script framework to ensure that the data and fee can be distributed to both parties at the same time without any trusted third party, in order to maintain the fairness of transactions. The second challenge is data buyers intending to justify the authenticity and availability of data before conducting guaranteed transactions by payment, and maintaining the benefit of data buyers, yet not damaging the profit of data owners. That is, we need a reasonable way to inspect trial data to be traded.

By using bit commitment and a BTC script framework, we propose and implement a scheme called FaDe to overcome the challenges above and guarantee aforementioned requirements. The contribution of the paper is as follows:

1. A decentralized scheme including data exchange protocol and implementation framework is proposed and designed. The proposed scheme enables data owners and data buyers to trade data directly without trusted third parties.
2. A fair data exchange protocol and the corresponding algorithm are proposed and presented, by which neither data owners lose traded data without acquiring sufficient payment, nor do data buyers finish payment without obtaining sufficient traded data.

The rest of the paper is organized as follows: in Section 2, we review related work. Section 3 formulates research problems and challenges. Our proposed scheme is presented in Section 4. Section 5 presents evaluation and analysis, and we conclude this paper in Section 6.

2. Related Work

As data value increasingly accumulates, data sharing from data owners to data buyers is envisioned to be of great importance. Some schemes for fair and non-repudiated data exchanges are proposed [6]. Especially in e-commerce, profit protections of consumers and merchants are increasingly important, thus some transaction rules have gradually been proposed [7]. A novel cryptographic primitive, called the Certificate of Encrypted Message Being a Signature (CEMBS), was proposed with offline TTP (Trusted Third Party). It can work as an elemental building block in fair exchange protocols to avoid a misbehaving or cheating party [8]. In order to ensure the fairness of data transaction, together with ensuring that the midway withdrawal of both parties will not affect fairness, the availability of data should also be guaranteed. In data exchange, the availability and privacy of data should be concerned [9]. It is always welcomed to allow buyers to verify products, i.e., received products that are ordered. Ray evaluates a fair exchange e-commerce protocol based on online trusted third parties, which ensures fairness and prevents either party from gaining an advantage through premature withdrawal of a transaction or other misconduct [10]. Regarding the data trial reading method, a random authentication method called cut-and-choose is devised by applying Trusted Computing Technology [11]. Most fair data exchange methods rely on trusted third parties to guarantee the fairness of transactions [12]. Therefore, it is difficult to guarantee transaction fairness when the system fails. Liu et al. designed a fault-tolerant correctness criterion for fair data exchange, namely, fair-lossless recoverability. The system with this criterion is immune from risks of fairness loss [13]. A method to recover transactions from a crashed network is also proposed [14]. Although many fair data exchange methods have been proposed, none of them are conducted without trusted third parties, which may result in the damage of data privacy or the unavailability of data transactions.

In addition to ensuring fairness of data exchange, both parties need to protect their privacy. Therefore, an asynchronous optimistic Fair Exchange Protocol with a Semi-trusted Third Party is suggested [15]. One of its significant contributions is keeping anonymities, even against a semi-closed third party. Decentralization allows buyers and owners to trade freely without third parties. Shi proposes a secure and lightweight triple-trusting architecture (SLTA), which is based on blockchain [16]. It can ensure the credibility of entity identities in IoT, whereas this method only implements data exchange rather than data transaction. Moreover, a new optimistic fair scheme for P2P transactions with multiple participants is proposed [17]. In this mode, a P2P system is used for electronic payment, and third-party intervention is not needed until disputes exist. However, this method requires intermediaries to play a passing role in the network, so there are more than two parts (i.e., owners and buyers). Moreover, the zero-knowledge proof method in the P2P network can also advance data privacy while guaranteeing its security [18]. Hence, a protocol and a trading method is welcomed, which only involves buyers and sellers, and also ensures fairness.

Complete decentralization requires a uniform payment that can transfer data without third parties. Blockchain such as BTC can exactly satisfy this requirement, e.g., anyone can create a BTC account. In general, BTC is seen as more flexible and more private than other forms of payment [19]. Besides being cryptographic digital currency, bit coins can also be used to design fair protocols [20]. A fair protocol for computing outsourcing has been proposed by Dorsala, whose technical basics are BTC and Ethereum [21]. In addition, BTC is used as a part of fair electronic voting protocol, guaranteeing the security and privacy of voters simultaneously [22]. Moreover, a lottery protocol based on cryptocurrency is proposed [23]. Thanks to the privacy of BTC, the automatic execution of BTC scripts, and the characteristics of P2P networks, BTC is employed in various protocols. Bitcoin can be also utilized as a part of fair data exchange protocol. A protocol is proposed and implemented via a BTC smart contract to exchange private keys of BTC in an atomic way [24].

3. Problem Formulation

This module presents the current common data exchange patterns and analyzes their security risks. On this basis, design goals in this paper will be introduced.

3.1. System Model and Attack Model

Current methods for data exchanges and data transactions are usually composed of data owners, data buyers, and trusted third-party platforms. They can exchange data in the following two ways: (1) data owners store data on third-party servers at first. Data buyers purchase the data from the third party, and the money is transferred to data owners by the third party (see Figure 1). (2) Data buyers pay to the third-party, and data owners send data directly to data buyers. After buyers confirm the receipt of data, money will be transferred from the third party to the data owner (see Figure 2). Under the supervision of a third party, both parties (i.e., owners and buyers) can conduct transactions fairly. In other words, data buyers do not have to worry about not acquiring the data after payment, and data owners do not need to be afraid of not obtaining the money after sending data.

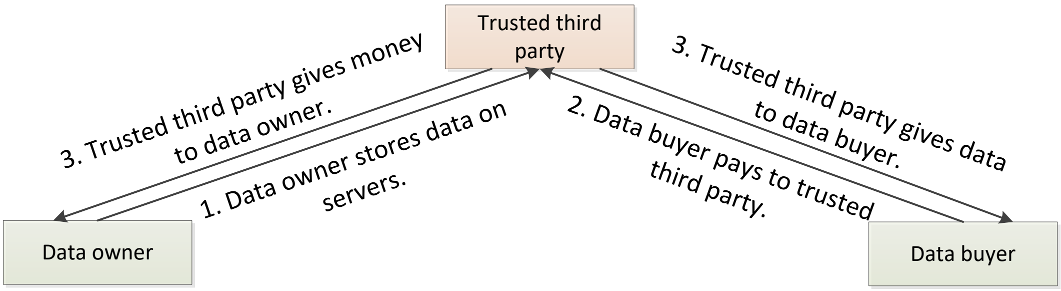
Trusted third party
3. Trusted third party gives money to data owner.
1. Data owner stores data on servers.
2. Data buyer pays to trusted third party.
3. Trusted third party gives data to data buyer.
Data owner
Data buyer

Figure 1. Method I.

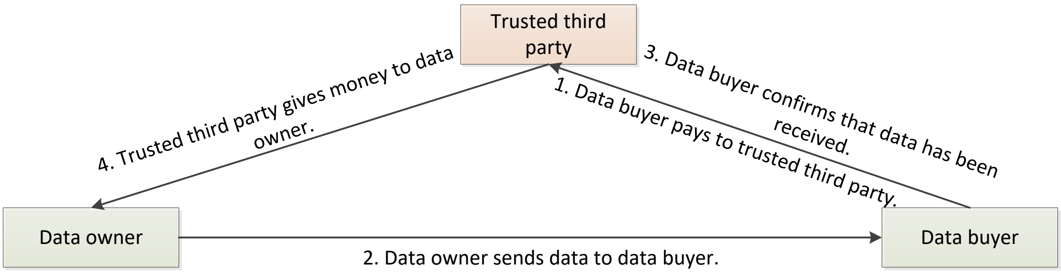
Trusted third party
4. Trusted third party gives money to data owner.
3. Data buyer confirms that data has been received.
1. Data buyer pays to trusted third party.
Data owner
Data buyer
2. Data owner sends data to data buyer.

Figure 2. Method II.

However, there are two possible attacks against the two main existing methods above.

Data stolen and garbled. In the first case, data owners cannot guarantee that their data stored on servers will not be stolen by others or servers, resulting in data leakage. Even if data owners store data encrypted on servers, it cannot be guaranteed that the data are not modified. Consequently, data owners have to encrypt data and add checksum before storing data in servers and checking the integrity before trading.

The third party suspends service. In the second case, fair transactions between two parties cannot be completed when servers suspend service, since fair transactions between the two parties must be conducted in the presence of servers. Subsequently, the transaction completion depends on whether servers are working properly. It thus becomes a single point of failure and possibly confronts a Denial of Service attack.

3.2. Design Goals

We use a smart contract with a BTC script to create a virtual trusted third party, which can automatically execute operations for data exchanges and payments. There are three design goals as follows:

1. Enable direct fair transactions between data owners and data buyers without third-party supervision.
2. Use BTC smart contracts for fair payments and make transactions untampered and traceable.
3. User privacy will not suffer leakage, even though blockchain is induced and transactions are stored in a BTC public ledger.

4. Proposed Scheme

In this section, we firstly give some notions to simplify our description. Next, a fair payment protocol for big data transactions is proposed. The implementation of the protocol by blockchain smart contract is discussed in detail finally.

4.1. Preliminaries

Some notions are explained in Table 1.

Table 1. Notations.

D	Data in transaction
Db	Data block (transferred data block instead of blocks in blockchain)
Hd	Hash value of data block (Db)
Key	A random secret key
Hk	Hash value of secret key
C	Cipher text
Ω_{Db}	The set of Db
Ω_{Hd}	The set of Hd
Ω_k	The set of Key
Ω_{Hk}	The set of Hk
Ω_c	The set of C
Sig_A	The signature of Alice
Sig_B	The signature of Bob

In order to simplify the description, we use Alice to denote the peer who owns data and Bob to denote the peer who buys data. BTC and Ethereum are the two most widely used cryptocurrencies in the world, and fair payment protocol presented in this paper is independent with an underlying blockchain virtual machine. In this paper, we use BTC framework (and bit coins as money) as an illustration in implementation.

4.2. Fair Payment Protocol

As illustrated in Figure 3, the Fair Payment Protocol consists of four stages: *Data Preprocessing*, *Randomly Choosing and Verifying*, *Paying*, and *Data Confirming*.

4.2.1. Stage I: Data Preprocessing

In this stage, Alice communicates with Bob through private online channels to initialize some parameters. Three steps are performed in sequence.

1. *Requesting*: Bob makes a request with Alice for starting the protocol.
2. *Data Blocking and Encrypting*: Once Alice receives the request, she will prepare block data and encrypt block data. First, Alice needs to partition a whole data set into n blocks. For each block, a *Random Secret Key* (denoted as *Key*) and its hash value will be computed. Thus, the data block can be encrypted by using a symmetric key algorithm (i.e., DES). At the end of this step, an additional five data sets—n data blocks (Ω_{Db}), hash values of each data block (Ω_{Hd}), random secret keys (Ω_k), hash values of each secret key (Ω_{Hk}), and the cipher texts of each data block (Ω_c) are generated. The algorithm is shown in Algorithm 1.
3. *Responding*: Alice makes a response to Bob with three additional data sets which include Ω_{Hd}, Ω_{Hk}, Ω_c and her own public keys that can be used by Bob to transmit bit coins to Alice.

4.2.2. Stage II: Randomly Choosing and Verifying

By now, three data sets (Ω_{Hd}, Ω_{Hk} and Ω_c) are attained by Bob. Hence, Bob needs to ensure their correctness. Just like the previous stage, this stage can also be divided into three steps as follows:

1. *Randomly choosing*: Bob randomly selects a power set of M $\Omega_{Index} = \{a_1, a_2, \cdots, a_m | 0 < m < n, a_i < n\}$, and then sends Ω_{Index} to Alice.
2. *Choosing key set*: When Alice receives Ω_{Index}, a partition key set $\Omega_{Key} = \{k_{a_1}, k_{a_2}, \cdots, k_{a_n} | a_i \in \Omega_{Index}, k_{a_i} \in \Omega_k\}$ can be generated and sent to Bob.
3. *Partition key verifying*: Bob has obtained partial encryption keys Ω_{Key} from Alice. Then, for each $a_i \in \Omega_{Index}$ and $k_{a_i} \in \Omega_{Key}$, Bob can decrypt a_i-th data block using k_{a_i}, and calculate its hash value. Finally, Bob can compare the hash value with a_i-th hash value in Ω_{Hd} to ensure whether the k_{a_i} is correct.

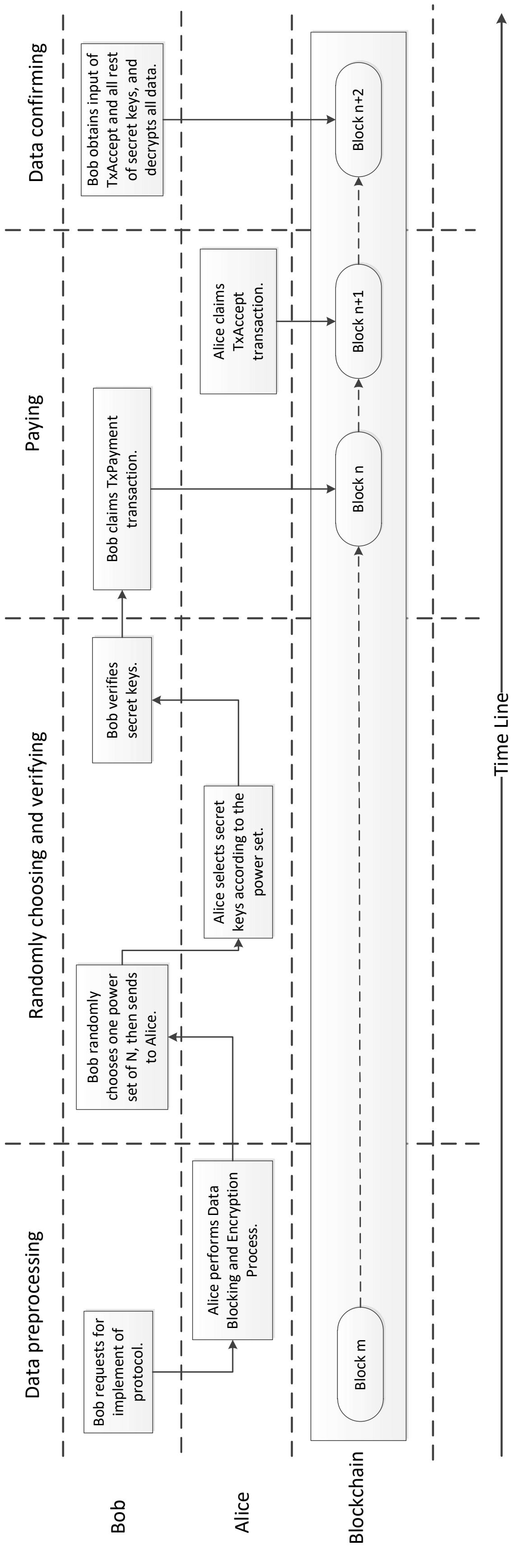
Data preprocessing
Randomly choosing and verifying
Paying
Data confirming
Bob
Alice
Blockchain
Bob requests for implement of protocol.
Alice performs Data Blocking and Encryption Process.
Bob randomly chooses one power set of N, then sends to Alice.
Alice selects secret keys according to the power set.
Bob verifies secret keys.
Bob claims TxPayment transaction.
Alice claims TxAccept transaction.
Bob obtains input of TxAccept and all rest of secret keys, and decrypts all data.
Block m
Block n
Block n+1
Block n+2
Time Line

Figure 3. Protocol architecture.

In this step, in order to let Bob verify whether three data sets are correct, Alice needs to demonstrate again that she is willing to make a transaction with Bob and release partial secret keys. If the number n is large enough, partial secret keys leakage does not damage the value of overall data.

Algorithm 1: Data blocking and encrypting

```
Input: integer n, data set
Output: set[quintuple[sub_data, hash_sub_data, c, key, hash_key]]
result = [];
data_set = data_division(data, n);
for sub_data in data_set do
    quintuple = [];
    hash_sub_data = hash_function(sub_data);
    key = random_key_generate();
    hash_key = hash_function(key);
    c = Symmetric_encryption(sub_data, key);
    hash_c = hash_function(c);
    quintuple.add(sub_data);
    quintuple.add(hash_sub_data);
    quintuple.add(c);
    quintuple.add(key);
    quintuple.add(hash_key);
    result.add(quintuple);
end
return result;
```

4.2.3. Stage III: Paying

In this stage, Bob can obtain the rest of the secret keys, and Alice can acquire the Bitcoin payment. There are three types of transactions, which are shown in Figure 4.

1. *Payment Transaction*, denoted as TxPayment, allows two types of transactions to transfer bit coins.
2. *Refund Transaction*, denoted as TxRefund, allows Bob to refund his bit coins back in a future time when Alice aborts the transaction.
3. *Accept Transaction*, denoted as TxAccept, allows bit coins in a TxPayment transaction to be transferred to Alice's BTC address.

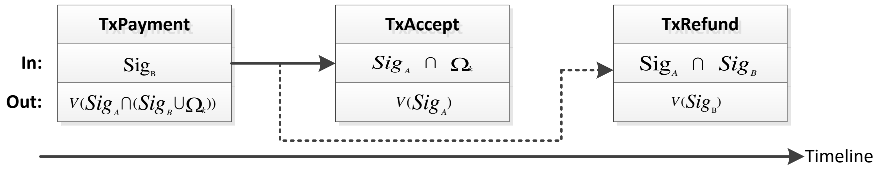

Figure 4. Three types of transactions.

Firstly, Bob generates the TxPayment transaction using an output script, which can be referenced by two types of redemption transactions. One is inputting the script of transaction that is signed by both Bob and Alice. The other includes an inputting script that is signed with Bob's public key and the rest of the secret keys mentioned in the stage *Randomly Choosing and Verifying*. After Bob generates the TxPayment transaction; before releasing it to the BTC network, he needs to generate the TxRefund transaction. According to the TxRefund transaction, he can ensure that the money in the TxPayment transaction will be refunded at a future time, if the protocol aborts. He should generate an incomplete TxRefund transaction by setting the lock time, signing with his private key, and setting up his BTC address as its transfer destination. Bob sends the unfinished TxRefund transaction to Alice. When Alice receives the TxRefund transaction, she checks its lock time and signature. If both of them are correct, she appends her signature using her private key. To this end, the TxRefund transaction is completed. Alice sends the completed TxRefund to Bob. Bob receives the completed TxRefund transaction and checks its signature and lock time. If every thing in TxRefund is correct, Bob releases both the TxPayment transaction and the TxRefund transaction to the BTC network.

Alice generates the TxAccept transaction by setting the transfer destination as her BTC address, adding her signature, supplying the rest of the secret keys, and waiting for the TxPayment transaction confirmation in sequence. Once she receives the confirmation, she will verify its value and release the TxAccept transaction to the BTC network.

4.2.4. Stage IV: Data Confirming

Bob monitors the BTC network until the TxAccept transaction is released. Next, he obtains the input script and gains the rest of the secret keys. Once Bob gets all of the secret keys, he can decrypt all data blocks.

4.3. Enhancement

We usually assume that participants in protocols are trustworthy. If this assumption is avoided, that is, Alice and Bob may be malicious or misbehave, we need to enhance the proposed scheme, especially in the absence of any trusted third party. More analysis details are listed as follows:

1. *Malicious Data Owner*: This can be defended against in the current version. For example, Alice provides invalid data that Bob does not want to buy to cheat Bob, after she obtains money from Bob. In the *Randomly Choosing and Verifying* stage, Alice has to provide partial data that is selected by Bob randomly, so Bob can verify those data to determine whether he will continue. In the *Paying* stage, if Alice wants to attain money from the BTC address, she must provide the rest of the secret keys.
2. *Malicious Data Buyer*: This can be defended against in the current version but is still not perfect. For example, Bob wants to obtain all data without payment. In the current version, Bob can suspend transactions at any time without any penalty. Once the transaction is aborted, Bob can obtain a small portion of keys, which is able to decrypt the corresponding part of data (the part of data which is read by Bob for random inspection). However, with regard to Alice, the leakage of this part of the data does not affect the value of the overall data to be traded.
3. *Eavesdropper*: This can be defended against in the current version. In the paying stage, all data are public and can be accessible to anyone. Although an eavesdropper can obtain all of the remaining keys, as long as the security of the private channels between Alice and Bob that they used to communicate in the Data Preprocessing stage is guaranteed, the leakage of encryption keys will not cause data leakage relative to the entire system.
4. *Denial of Service*: Because anyone can initiate a deal with Alice and can terminate the deal at any time during the transaction without any cost, multiple fake buyers are deployed by attackers to initiate transaction requests, with the purpose of consuming Alice's computing resources, which may cause a denial of service attack. It is an inherent risk since no centralized admission exists for regulating buyers' behaviors.

In summary, the main drawback of the current version of fair payment protocol is that Bob can terminate the transaction at any time without any cost. Therefore, we proposed an enhancement for the protocol, which has two advantages compared to the aforementioned protocol as follows:

1. Bob provides Alice with a margin to justify that he is not a malicious attacker.
2. The single payment is changed to multiple payments, which allows Bob to confirm whether the data provided by Alice guarantee Bob's demand.

The Improved Fair Payment Protocol contains five stages, which are *Guaranteeing*, *Data Preprocessing*, *Randomly Choosing and Verifying*, *Paying*, and *Data Confirming*.

Stage I: Guaranteeing

In this stage, Bob needs to submit a TxGuarantee transaction that mainly makes bit coins be unspendable for a period of time. The specific processes are shown as follows:

1. Bob first generates a TxGuarantee transaction, which requires Bob's signatures for the corresponding redemption transaction script. A TxRedemp transaction is generated to redeem his money, in which the lock time of the transaction is set to T, and his signature is added to the input script of the transaction. Finally, Bob sends the TxRedemp transaction to Alice.
2. When Alice receives the TxRedemp transaction, she needs to verify Bob's signature first, and then confirms the lock time. If they are correct, Alice adds her signature to the input script of TxRedemp transaction. Finally, Alice sends the TxRedemp transaction to Bob.
3. After Bob receives the TxRedemp transaction, he checks the lock time and two signatures. If they are correct, the TxRedemp transaction and the TxGuarantee transaction are submitted to the BTC network.
4. Alice and Bob wait for the confirm of the TxGuarantee transaction.

In the *Guaranteeing* stage, Bob needs to freeze a sum of money (in terms of bit coins), indicating that he is not a malicious buyer. If a malicious attacker tries to launch a denial of service attack, he needs to provide each participating buyer with a sum of bit coins during the *Guaranteeing* stage, which increases the cost of the attack undoubtedly.

5. Analysis

The security of this method is analyzed in this module, and an example is presented to explain it more vividly.

5.1. Security Analysis

The security of this scheme will be analyzed from the perspectives of data buyer and data owner.

5.1.1. Security and Fairness of Data Owner

Trial reading will not cause data leakage or reduce the value of data to be traded. In this scheme, the number of data segments should far exceed the number of trial reading segments. Thus, it ensures that data buyers will not be able to conjecture remaining contents from trial reading data.

Data leakage will not occur even though keys are exposed on the BTC script. Because encrypted data are sent to buyers privately, others cannot gain encrypted data even if they attain keys. Moreover, even though the attacker obtains encrypted data, he still cannot conjecture segments that buyers chose for trail reading initially. That is, he cannot figure out corresponding relations between keys on the ledger and encrypted segments; he thus experiences extreme difficulty with decrypting encrypted data.

Data owners need not be concerned about payment. Once a data buyer pays bit coins in a BTC script, the data owner can check that transaction on the BTC ledger. Afterward, the data owner provides the remaining keys to the BTC script. In this case, the data buyer has paid for traded data in advance. As long as the data owner provides correct data, he will ordinarily attain the paid bit coins.

5.1.2. Security and Fairness of the Data Buyer

Data owners must provide data whose value matches its price. Data buyers can select partially encrypted fragments to inspect before concretely paying, and data owners should send keys of these parts to buyers. Only after that can buyers decide whether to buy the data or not.

Data buyers need not be concerned that data owners do not provide all of the keys. Initially, a data owner should send the hash value of all keys to the data buyer. Hence, if he does not provide all remaining keys in the end, or he provides invalid keys, the data buyer can recognize the case. Furthermore, if the BTC script cannot be executed successfully, the paid bit coins in the script cannot be withdrawn to the data owner's account.

Data buyers need not be concerned that data owners do not provide data after payment. The refund mechanism guarantees that, if remaining keys are not properly presented to the BTC script, the bit coins paid to the script by data buyers will be refunded instead of being deposited to the data owner's account.

5.2. Performance Analysis

Data exchange using our protocol may incur additional time costs. The data owner and buyer attain bitcoin and data simultaneously, but they provide data and bitcoin to the blockchain step by step. Thus, it increases the time cost of transactions. In addition, the transaction does not officially take effect until miners pack up six blocks, and bitcoin are actually paid to sellers' accounts at this point. Furthermore, buyers can inspect part of the data before payment in this scheme, which also delays the transaction completion time. Moreover, in every data transaction, although the same data are in different transactions, the data should be encrypted again. Hence, the encryption and decryption of data will take a certain amount of time and calculation cost.

However, in data transaction, both parties are more concerned about the security, fairness and reliability of the transaction. Although some time cost is added, data are encrypted to ensure their security. In addition, the data buyer can inspect data and ensure its availability. Otherwise, through the protocol based on bitcoin, the fairness of transaction is guaranteed without trusted third parties. The scheme protects data security and fair transactions; as a result, these time costs are acceptable to users.

Moreover, due to the absence of trusted third party participation, the computation and storage pressure of servers are reduced. All of the computation pressure in this method is increased on clients, which are acceptable for each node on average. Furthermore, this approach fosters peer-to-peer, decentralized data transfer.

This protocol is appropriate for data exchange between users, in the form of transaction, to enhance its fairness. According to data stream exchange, which is high-speed and real-time, it always happens between entities in IoT. Since there is not a data buyer in this kind of data exchange, it is not a data transaction, and there is no need to apply our protocol.

5.3. Case Study

We present a case study as follows to illustrate the proposed scheme. Alice is a data owner and Bob is a data buyer, while Eve is an attacker who wants to steal data. The value of the data is 1 Bitcoin.

Firstly, Alice divides the data into $n = 1000$ segments and encrypts every segment with a specific symmetric secrete key. Afterward, she sends Bob hash values of original data, encrypted fragments, and hash values of corresponding keys. In order to prevent the possibility of cheating, in which the data sold by Alice are not worth 1 Bitcoin, Bob needs to check the value of the data. Bob randomly selects $m = 10$ encrypted segments and requests Alice to provide their keys. Since Bob only views $100 * m/n = 1$ percent of data, Alice does not need to be concerned that the 10 decrypted data segments can help Bob infer the original data of the other 990 segments. After receiving the keys, Bob firstly computes their hash values and compares them with earlier received hash values. If they are identical, Bob will decrypt the corresponding encrypted segments and inspect the data. If he confirms that the data are worth 1 Bitcoin, he will pay for it.

Bob pays 1 Bitcoin to the BTC script and generates a transaction input script attaching the hash value of the remaining keys. To this end, Alice can check that the transaction has been publicly accepted in the BTC ledger. When Alice confirms that Bob has indeed paid 1 Bitcoin, she can finally attain this 1 Bitcoin by sending all remaining keys to the BTC script. At this moment, Bob needs not be concerned that Alice does not send him data because if Alice does not send data to the BTC script within a specified time limit, the 1 Bitcoin paid will be refunded to Bob's account.

When Alice sends the remaining keys to the BTC script, a transaction output script is generated that pays 1 Bitcoin to Alice's account. However, the deal can be effective only when miners certify and package this block into the blockchain (BTC ledger). That is, 1 Bitcoin will not be recorded into Alice's account until miners verify that Alice has indeed provided the correct data. Therefore, Bob does not need to be concerned that the data provided by Alice are useless.

When the transaction is completed, Eve can obtain keys that Alice sent to the script. She, however, did not know the encrypted data corresponding to the keys, so they are useless to her. Nevertheless, Eve has obtained the encrypted data, and she knows neither how Alice fragmented the data, nor what samples Bob chose. As a consequence, she also cannot find the encrypted fragments corresponding to keys she gained, and she still cannot figure out the original data.

6. Conclusions

In this paper, we propose a fair exchange protocol for autonomously data sharing, and describe a concrete implement framework based on BTC. Data owners and data buyers can trade with each other directly without a trusted third party. Data buyers can randomly check parts of the data in advance for protecting their benefits but not damaging the profit of data owners due to data leakage. Data owners can obtain sufficient payment and thus avoid the loss of traded data; data buyers can obtain sufficient data and thus avoid the loss of money. The concrete framework is designed based on BVM smart contract scripts. This protocol also assumes that there is a third trusted party, and thus the risks of data leakage and data tampering from third party can be avoided, as well as thwarting a single point of failure. This big data exchange method can be applied to the increasingly popular P2P network, and is beneficial to alleviate users' information security, which has received more attention these last few years. In future networks, the data exchange volume will rise continually, which can add pressure to the servers. The protocol in this paper can save the data in clients, which makes users protect their data by themselves; in the meantime, it can reduce the load of databases and servers. This scheme is more suitable for the data exchange between users (rather than entities in IoT) with lower efficiency requirements, and can enhance the decentralization of the network.

Author Contributions: Conceptualization, W.R.; Data curation, J.G.; Formal analysis, Y.C.; Methodology, C.L.; Project administration, W.R.; Resources, C.L.; Software, C.L.; Supervision, W.R.; Validation, W.R.; Writing—original draft, Y.C. and J.G.; Writing—review and editing, W.R.

Funding: This research was funded by Major Scientific and Technological Special Project of Guizhou Province under Grant No. 20183001, Natural science foundation under Grant No. 61962009, Open Funding of Guizhou Provincial Key Laboratory of Public Big Data under Grant No. 2018BDKFJJ009 and 2017BDKFJJ006.

Acknowledgments: The research was financially supported by Major Scientific and Technological Special Project of Guizhou Province under Grant No. 20183001, Open Funding of Guizhou Provincial Key Laboratory of Public Big Data under Grant No. 2018BDKFJJ009 and 2017BDKFJJ006.

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

References

1. Walker, S.J. Big Data: A Revolution That Will Transform How We Live, Work, and Think. *Int. J. Advert.* **2014**, *33*, 181–183. [CrossRef]
2. Tariq, N.; Asim, M.; Al-Obeidat, F.; Zubair Farooqi, M.; Baker, T.; Hammoudeh, M.; Ghafir, I. The Security of Big Data in Fog-Enabled IoT Applications Including Blockchain: A Survey. *Sensors* **2019**, *19*, 1788. [CrossRef]
3. McAfee, A.; Brynjolfsson, E. Big data: The management revolution. *Harv. Bus. Rev.* **2012**, *90*, 60-68.
4. Badev, A.; Chen, M. Bitcoin: Technical Background and Data Analysis. *Financ. Econ. Discuss. Ser.* **2014**, 1–38. [CrossRef]

5. Suciu, G.; Nădrag, C.; Istrate, C.; Vulpe, A.; Ditu, M.-C.; Subea, O. Comparative Analysis of Distributed Ledger Technologies. In Proceedings of the 2018 IEEE Global Wireless Summit (GWS), Chiang Rai, Thailand, 25–28 November 2018; pp. 370–373.
6. Nenadi, A.; Zhang, N. Non-Repudiation and Fairness in Electronic Data Exchange. In *Enterprise Information Systems V*; Camp, O., Filipe, J.B.L., Hammoudi, S., Piattini, M., Eds.; Springer: Dordrecht, The Netherlands, 13–15 May 2005; pp. 286–293.
7. Guo, M. Protection of consumer right to a fair transaction in E-commerce. In Proceedings of the 2011 International Conference on Business Management and Electronic Information, Guangzhou, China, 13–15 May 2011; pp. 5–8.
8. Bao, F.; Deng, R.H.; Mao, W. Efficient and practical fair exchange protocols with off-line TTP. In Proceedings of the 1998 IEEE Symposium on Security and Privacy (Cat. No. 98CB36186), Oakland, CA, USA, 6 May 1998; pp. 77–85.
9. 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]
10. Ray, I.; Ray, I.; Narasimhamurthy, N. A Fair-exchange E-commerce Protocol with Automated Dispute Resolution. In *Proceedings of the IFIP TC11/ WG11.3 Fourteenth Annual Working Conference on Database Security: Data and Application Security, Development and Direction*s; Kluwer, B.V.: Deventer, The Netherlands, 2001; pp. 27–38.
11. Tate, S.R.; Vishwanathan, R. Improving Cut-and-Choose in Verifiable Encryption and Fair Exchange Protocols Using Trusted Computing Technology. In *Proceedings of the 23rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security XXIII*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 252–267.
12. Cook, N.; Robinson, P.; Shrivastava, S. The rigorous implementation of a fair exchange protocol for non-repudiable Web service interactions—A case study. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop, Istanbul, Turkey, 17–20 April 2007; pp. 307–314.
13. Liu, P.; Ning, P.; Jajodia, S. Avoiding loss of fairness owing to process crashes in fair data exchange protocols. In Proceeding International Conference on Dependable Systems and Networks, DSN 2000, New York, NY, USA, 25–28 June 2000; pp. 631–640.
14. Wang, H.; Guo, H.; Lin, M.; Yin, J.; He, Q.; Zhang, J. Dependable Transaction for Electronic Commerce. In *Computational Science and Its Applications–ICCSA 2005*; Gervasi, O., Gavrilova, M.L., Kumar, V., Laganà, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 691–700.
15. Guo, L.; Li, X.; Lv, X.; Gao, J. Anonymous Fair Exchange Protocol with a Semitrusted Third Party. In Proceedings of the IEEE Second International Conference on Data Science in Cyberspace, IEEE Computer Society, Shenzhen, China, 26–29 June 2017; pp. 431–440.
16. Blockchain based trusted data sharing among trusted stakeholders in IoT. In *Software: Practice and Experience*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 2019.
17. Liu, Y.; Fu, J.; Zhang, H. An Optimistic Fair Protocol for P2P Chained Transaction. In Proceedings of the 10th Asian Computing Science Conference on Advances in Computer Science: Data Management on the Web; Springer: Berlin/Heidelberg, Germany, 2005; ASIAN'05, pp. 136–145.
18. Bernard, S.; Potop-Butucaru, M.G.; Tixeuil, S. A Framework for Secure and Private P2P Publish/Subscribe. In *Proceedings of the 12th International Conference on Stabilization, Safety, and Security of Distributed Systems, SSS'10*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 531–545.
19. Bohme, R.; Christin, N.; Edelman, B.; Moore, T. Bitcoin: Economics, Technology, and Governance. *J. Econ. Perspect.* **2015**, *29*, 213–238. [CrossRef]
20. Bentov, I.; Kumaresan, R. How to Use Bitcoin to Design Fair Protocols. *Advances in Cryptology—CRYPTO 2014*; Garay, J.A., Gennaro, R., Eds.; Springer:Berlin/Heidelberg, Germany, 2014; pp. 421–439.
21. Reddy Dorsala, M.; N Sastry, V.; Chapram, S. Fair Protocols for Verifiable Computations Using Bitcoin and Ethereum. In Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 2–7 July 2018; pp. 786–793.
22. Bao, Z.; Wang, B.; Shi, W. A Privacy-Preserving, Decentralized and Functional Bitcoin E-Voting Protocol. In Proceedings of the 2018 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, China, 8–12 October 2018; pp. 252–256.

23. Miller, A.; Bentov, I. Zero-Collateral Lotteries in Bitcoin and Ethereum. In Proceedings of the 2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW), Paris, France, 26–28 April 2017; pp. 4–13.
24. 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.* **2017**, in press. [CrossRef]