information

Article

Privacy-Protection Method for Blockchain Transactions Based on Lightweight Homomorphic Encryption

Guiyou Wang [1], Chao Li [2,*], Bingrong Dai [2] and Shaohua Zhang [3]

[1] Faculty of Information, Shanghai Ocean University, Shanghai 201306, China; wgy-sdrz@163.com
[2] Shanghai Development Center of Computer Software Technology, Shanghai 201112, China; dbr@sscenter.sh.cn
[3] Shanghai Business School, Shanghai 200235, China; zhangsh@sbs.edu.cn
* Correspondence: lc@sscenter.sh.cn

Abstract: This study proposes an privacy-protection method for blockchain transactions based on lightweight homomorphic encryption, aiming to ensure the security of transaction data and user privacy, and improve transaction efficiency. We have built a blockchain infrastructure and, based on its structural characteristics, adopted zero-knowledge proof technology to verify the legitimacy of data, ensuring the authenticity and accuracy of transactions from the application end to the smart-contract end. On this basis, the Paillier algorithm is used for key generation, encryption, and decryption, and intelligent protection of blockchain transaction privacy is achieved through a secondary encryption mechanism. The experimental results show that this method performs well in privacy and security protection, with a data leakage probability as low as 2.8%, and can effectively defend against replay attacks and forged-transaction attacks. The degree of confusion remains above 0.9, with small fluctuations and short running time under different key lengths and moderate CPU usage, achieving lightweight homomorphic encryption. This not only ensures the security and privacy of transaction data in blockchain networks, but also reduces computational complexity and resource consumption, better adapting to the high-concurrency and low-latency characteristics of blockchain networks, thereby ensuring the efficiency and real-time performance of transactions.

Keywords: lightweight homomorphic encryption; blockchain; privacy protection; Paillier algorithm; encryption; decryption; zero-knowledge proof

Citation: Wang, G.; Li, C.; Dai, B.; Zhang, S. Privacy-Protection Method for Blockchain Transactions Based on Lightweight Homomorphic Encryption. *Information* **2024**, *15*, 438. https://doi.org/10.3390/info15080438

Academic Editor: Tieling Zhang

Received: 13 June 2024
Revised: 15 July 2024
Accepted: 23 July 2024
Published: 28 July 2024

1. Introduction

With the widespread application and popularization of blockchain technology, its decentralized and tamper-proof features provide users with unprecedented data security guarantees [1]. However, while enjoying these advantages, privacy protection issues in blockchain transactions are gradually becoming prominent. Traditional privacy protection methods are difficult to adapt to the characteristics of blockchain technology; therefore, researching privacy protection methods for blockchain transactions is particularly important [2,3]. This study not only helps to raise awareness of privacy protection among individuals and society, but also enables the development of industry norms from a technical and managerial perspective, promoting healthy industry development [4]. By studying privacy protection methods for blockchain transactions, we can provide users with a more secure and reliable data-trading environment, further promoting the widespread application of blockchain technology.

The current research on privacy protection methods is primarily focused on developing efficient and secure techniques to ensure data privacy and security. Reference [5] proposes a dual-layer collaborative privacy data-protection method, which achieves privacy and confidentiality protection between different enterprises by transferring and partitioning data from various businesses. However, in certain scenarios, this method may rely on

external trust mechanisms or third-party services, potentially increasing security risks and the possibility of data exposure. Reference [6] presents a privacy and security protection method for blockchain transactions based on SGX. SGXTrans leverages Intel's Software Guard Extensions technology within a lightweight client framework, placing cryptographic data, user keys, generated user addresses, and sensitive privacy information within the SGX secure enclave for protection. However, SGX supports only up to 128MB of secure memory, which may require frequent page swaps in and out when handling large amounts of data, thereby increasing computational overhead and time. Reference [4] proposes a transaction privacy protection scheme based on polynomial commitments. This scheme uses commitment values of polynomials at random points to hide and bind transaction amounts, achieving privacy protection for user transaction volumes. It employs smart contracts and zero-knowledge proofs to verify the legitimacy of transactions without a trusted third party. However, in practical applications, it may be susceptible to various attacks and threats. Attackers might attempt to deceive verifiers by forging polynomial commitments or by breaking their computing environments to obtain sensitive information, leading to poor resistance to attacks. Reference [7] introduces a blockchain supervision privacy-protection method based on group signatures and attribute-based encryption. Multiple group managers are set up to generate fragments of user private keys, and users calculate their private keys based on the returned fragments. Experimental results show that this method can enhance the privacy-protection level of chain data while achieving privacy-protection supervision for both parties in a transaction. However, in certain situations, group managers may still be able to identify specific signers through certain means, thereby reducing obfuscation.

Based on the above research status, this paper proposes a privacy-protection method for blockchain transactions based on lightweight homomorphic encryption. This method can effectively protect the privacy of blockchain transactions, reduce computational costs, lower the probability of privacy- and security-protection data exposure, and improve the security of blockchain transactions. The main contributions of this study are as follows:

1. This method combines the Paillier algorithm with zero-knowledge proof technology, ensuring the privacy of transaction data and verifying the authenticity and accuracy of transactions through zero-knowledge proof.
2. By using secondary encryption to intelligently protect the privacy of blockchain transactions, the security of data is further enhanced. This secondary encryption method can effectively prevent attackers from cracking transaction information through a single encryption method, improving the defense capability of the entire system.
3. This method not only verifies transactions from the application side, but also from the smart-contract side, achieving dual protection. This dual verification mechanism can ensure the consistency and accuracy of transaction data in the blockchain network, enhancing the stability and reliability of the system.

2. Design of Privacy-Protection Methods for Blockchain Transactions

2.1. Analysis of Blockchain Infrastructure

The infrastructure of blockchain is a multi-level system, as shown in Figure 1, and includes multiple key parts such as application layer, contract layer, consensus layer, etc. These layers together constitute the core technical framework of blockchain [8,9]. The application layer is the bridge between blockchain systems and users. It provides a user interface and application program interface, allowing users to easily use blockchain systems for various operations, such as digital currency trading, asset management, supply chain tracking, etc. The application layer usually provides users with rich functions around specific business scenarios, such as managing blockchain systems, writing smart contracts, and establishing accounting systems. The contract layer encapsulates various script codes, algorithms, and the generated smart contracts of the blockchain system. A smart contract is an automated program that can automatically perform predefined operations when certain conditions are met [10,11]. The contract layer provides a flexible programming and data-

manipulation foundation for blockchain systems, enabling blockchain to support various complex business logic and rules. The consensus layer is one of the core components of a blockchain system, responsible for ensuring the consistency of blockchain states between nodes. The consensus layer encapsulates various consensus algorithms of network nodes, which determine how to select miners or validators for the next block, as well as how to handle forks in the network [12,13]. The existence of the consensus layer enables blockchain systems to achieve efficient and secure data synchronization and verification in a decentralized environment. The network layer is responsible for communication and data transmission between nodes, including node discovery, data synchronization, message broadcasting, and other functions. The storage layer is responsible for storing data and transaction records on the blockchain, and can use various technologies such as databases, file systems, distributed storage, etc. [14–16].

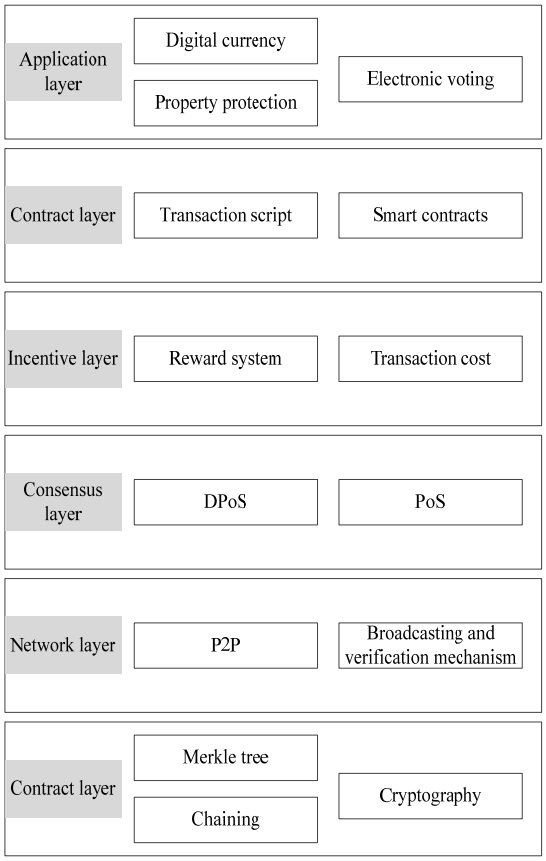

Figure 1. Schematic diagram of blockchain infrastructure.

From this, it can be seen that the infrastructure of blockchain [17] is a multi-level, highly integrated system, and the core technical framework of blockchain is composed of mutual cooperation and collaboration among different levels.

2.2. Zero-Knowledge Proof Process

To address potential security risks posed by external trust mechanisms or third-party services that may be relied upon, we should take a series of comprehensive measures. Firstly, build a decentralized trust mechanism that reduces reliance on a single trusted entity through joint verification by multiple nodes in the blockchain network. Secondly, adopting multi-party computation (MPC) and zero-knowledge proof (ZKP) techniques ensures that sensitive information is verified without being leaked, thereby reducing reliance on insecure services. At the same time, we continuously improve homomorphic encryption algorithms to enhance their security and efficiency, and ensure transaction security in untrusted environments by designing secure multi-party protocols. In addition, ensure

that the external services or mechanisms relied upon comply with regulatory requirements, and conduct continuous monitoring and auditing to promptly detect and respond to security threats. By integrating these measures, we can effectively enhance the overall security of the blockchain transaction privacy-protection system, reduce reliance on external trust, and ensure the security and privacy of user data. Based on the characteristics of blockchain infrastructure, zero-knowledge proof technology is used to verify the legitimacy of its data [18]. The use of zero-knowledge proof technology can verify the legitimacy of data when they are ciphertext, avoiding the occurrence of plaintext information in transaction data at the chain code end. Zero-knowledge proof allows both parties to verify the authenticity and accuracy of transactions without disclosing any additional information, effectively protecting user privacy and data security. The process of zero-knowledge proof is as follows:

When completing a transaction in a blockchain system, the transaction process is usually carried out on the application end. Therefore, in order to ensure the legality of the transaction, the application end must not only generate ciphertext transaction results for the smart-contract end, but also generate equality and range proof evidence, namely evidence of equal transaction amounts, evidence of equal input and output amounts, evidence of transaction amounts greater than 0, and evidence of transaction balances greater than 0. Then, these ciphertext transaction results and evidence are sent to the smart-contract end for verification of the legitimacy of the transaction. If the verification passes, the transaction results completed by the application end are written into the blockchain to complete the transaction process [19]. Otherwise, the transaction is rejected and a transaction failure message is returned to the application end.

Zero-knowledge proofs primarily consist of two parts: generating various proofs required for zero-knowledge verification on the application side, and verifying the legality of the transaction on the chain code side, based on these proofs. In the scenario where Alice transfers funds to Bob, the application side first collects the necessary keys and parameters, then generates a series of proofs to demonstrate the validity of the transaction and sends these proofs to the smart-contract side.

Upon receiving the proofs, the smart-contract side performs a series of verifications to ensure the legality of the transaction amount and balance. If all verifications pass, the transaction is recorded on the blockchain, thereby protecting the user's privacy. If any verification fails, the transaction is rejected.

2.3. Implementation of Privacy Protection for Blockchain Transactions

The introduction of lightweight homomorphic encryption technology is a method that allows encrypted data computation while ensuring data privacy, effectively solving the efficiency bottleneck problem of traditional encryption technology in blockchain transaction processing. And it adopts zero-knowledge proof technology, which is a method of verifying the legitimacy of transactions without disclosing any transaction details, greatly enhancing the privacy-protection level of transaction information. Through the secondary encryption mechanism, the confidentiality of transaction information has been further improved. Even in the case of a primary encryption being cracked, secondary encryption can provide additional security barriers. Finally, during the protection process, the authenticity and accuracy of transactions are ensured through collaborative verification between the smart contract end and the application end, thus achieving comprehensive protection of transaction-information privacy without affecting transaction efficiency. These innovative points and specific manifestations of the protection process together constitute the core contribution of this study in the field of blockchain transaction information-privacy protection. The Paillier algorithm is a partially homomorphic encryption method that satisfies additive homomorphism, consisting of three stages: key generation, encryption, and decryption.

(1) Generate key. Make the public key (n, g) and the corresponding private key (λ, μ):

$$\lambda = lcm(p-1, q-1) \tag{1}$$

$$\mu = \left[L\left(g^{\lambda} \bmod n^2\right)\right]^{-1} \bmod n \tag{2}$$

$$L(x) = (x-1)/n \tag{3}$$

Among them, n is the length of the public key; the order of g is a multiple of n; p and q are two prime numbers; and lcm represents the least-common multiple.

(2) Encryption. Make the ciphertext and plaintext c and m, respectively:

$$c = (g^m r^n) \bmod n^2 \tag{4}$$

$$m = L\left(c^{\lambda} \bmod n^2\right) \cdot L\left(g^{\lambda} \bmod n^2\right)^{-1} \bmod n \tag{5}$$

Among them, r is a random number; g^m is the encrypted part of the plaintext; and r^n is the encrypted part of a random number.

The encryption process of the public key is as follows:

Convert plaintext data m to integer form, randomly select an integer r such that $0 < r < n$, and r, are coprime with n. Calculate ciphertext c and homomorphically encrypt two ciphertexts c_1 and c_2 to obtain a new ciphertext $c\prime$:

$$c\prime = c_1 \cdot c_2 \bmod n^2 \tag{6}$$

(3) Decrypt. Use private key (λ, L) for decryption, and calculate plaintext m:

$$m = L\left(c^{\lambda} \bmod n^2\right) \cdot L\left(g^{\lambda} \bmod n^2\right)^{-1} \bmod n \tag{7}$$

Based on the basic principles of the Paillier algorithm mentioned above, intelligent protection of blockchain transaction privacy is achieved through secondary encryption. The specific process is as follows:

1. Use the Paillier algorithm to generate key pairs (n, g) and (λ, μ), and the key generation process is described above.
2. After the key pair is generated, the private data of blockchain transactions are encrypted once, using public key (n, g) to obtain ciphertext c. Due to the randomness of the selection of r during the encryption process, the ciphertext also has randomness. Therefore, to a certain extent, it has better protection performance for the specific content of plaintext m.
3. After completing one encryption, using the homomorphic property of the Paillier algorithm, the ciphertext c and public key (n, g) are encrypted twice to obtain ciphertext $c\prime$. This step reduces the risk of real transaction data being tampered with and stolen during transmission, and improves the data security of the entire blockchain transaction process.
4. After encryption is completed, $c\prime$ and (λ, μ) are transmitted to the blockchain transaction center through different data transmission channels.
5. After receiving $c\prime$ and (λ, μ), the trading center uses (λ, μ) to encrypt $c\prime$ twice to obtain plaintext m, which is the blockchain transaction information.
6. After the transaction is completed, the blockchain transaction center generates a new key while generating transaction-result information. After encrypting the transaction result information, the result information is transmitted to the business declaration party according to the above ciphertext transmission link. The declaration party decrypts and obtains the transaction result, and the transaction process ends.

The flowchart of privacy protection for blockchain transactions is shown in Figure 2.

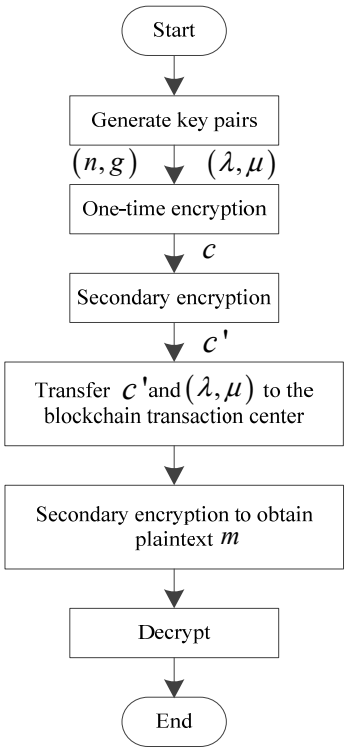

Figure 2. Blockchain transaction-privacy intelligent protection flowchart.

3. Experiments and Result Analysis

To verify the reliability of the proposed scheme for privacy protection in blockchain transactions, this method was tested under the same conditions as the methods in References [5,6], and the results were analyzed and compared.

3.1. Experimental Environment and Parameter Settings

The experimental hardware environment is the following: CPU: Intel Core i7-10700K, 8-core 16 thread, 3.8 GHz basic frequency, up to 5.1 GHz; Memory: 32 GB DDR4 RAM, 3200 MHz; Storage: 1 TB NVMe SSD.

The experimental software environment is operating system: Ubuntu 20.04 LTS; blockchain platform: Ethereum (tested using private chains); programming languages: Solidity (smart contract writing) and Python (experimental script writing).

Parameter setting: the block size is set to 2 MB to balance transaction capacity and network propagation efficiency; the block generation time is about 15 s, aiming to provide fast transaction confirmation while avoiding network congestion. The key lengths are 1024, 2048, and 3072 bits, respectively, to evaluate the performance of different security levels. Set the transaction quantity to 1000 randomly generated blockchain transactions, simulating actual transaction scenarios; the trading amount range is set between 1 and 100 ETH, covering trading needs of different scales. The selection of these parameters comprehensively considers security, performance, scalability, and practical-application requirements, aiming to build a secure and efficient blockchain transaction privacy-protection system to adapt to possible future-transaction volume growth and user-demand changes.

A comprehensive assessment was conducted on the practical application challenges of blockchain privacy-protection methods, including scalability, security, privacy protection, encryption algorithm limitations, universality of experimental results, and implementation costs. By simulating different network environments and security scenarios, we aim to optimize method performance, enhance its robustness against complex threats, and ensure its broad applicability across different blockchain platforms and applications. At the same time, attention was paid to the economic and resource costs of the methods, especially the feasibility for small-scale organizations, to ensure the sustainability of privacy protection.

Under the above experimental environment and parameter settings, a detailed analysis of the experimental results is conducted, and the effectiveness and practicality of the privacy-protection method for blockchain transactions based on lightweight homomorphic encryption are evaluated by comparing the performance of different methods.

The running pseudocode of the Algorithm 1 in this article is as follows:

Algorithm 1: Privacy-Preserving Transaction in Blockchain

Input: (PK, SK) for each node, Enc(·, ·), Dec(·, ·), Hash(·)
Output: Transaction results recorded on the blockchain
Step 1: User U sends transaction T
1. Generate plaintext message M for T.
2. Encrypt M with receiver's PK_S:
   C = Enc (M, PK_S)
3. Broadcast C and metadata of T.
Step 2: Blockchain nodes process transaction
1. Receive C and metadata.
2. Verify metadata.
3. Decrypt C with SK_S:
   M ' = Dec (C, SK_SK)
4. Process M′ and update blockchain.
Step 3: Receiver R verifies result
1. Receive transaction result.
2. Decrypt with SK_R:
   M " = Dec (·, SK_SK)
3. Verify M "

3.2. Analysis of Experimental Results

In the same experimental environment, the encryption scheme proposed in this paper was validated and compared with the methods in the literature [5,6] in terms of privacy and security-protection data exposure probability, anti-attack performance, confusion, computational efficiency, and resource consumption.

(1) Privacy and security-protection data exposure probability.

The probability of data exposure for privacy and security protection refers to the probability that attackers can deduce specific sensitive attribute values of the target individual through some means (such as using published data and background knowledge), which can reflect the degree of risk of privacy leakage. Its expression is

$$\Pr\{t[SA] = s | t[QI] = q, T^*, K_e\} \tag{8}$$

Among them, $t[SA]$ represents the sensitive attribute value of the target individual; $t[QI]$ represents the quasi-identifier attribute value of the target individual; T^* represents published data; and K_e represents the attacker's background knowledge.

In accordance with the methods in References [5,6], the analysis objectives are defined and the necessary datasets are collected. Then, the privacy risk is evaluated according to Equation (8); the attacker's background knowledge is simulated, iterative analysis is performed, and the data exposure probability is calculated. The results are shown in Figure 3.

From Figure 3, it can be seen that the methods in References [5,6], and in our paper gradually converge to a stable state during the iteration process. This means that the performance and effectiveness of the methods no longer undergo significant changes after multiple iterations, providing a reliable level of privacy protection. Through comparison, it can be seen that, after the probability of data exposure for privacy and security protection reaches a stable level, the method in Reference [5] controls the exposure probability to 11.5%, providing basic privacy protection. The method in Reference [5] further reduces the exposure probability to 8.5%, providing stronger privacy protection. However, the

exposure probability of the method in this article is only 2.8%, demonstrating a significant effect in protecting the privacy of blockchain transactions. This indicates that even if attackers can access encrypted data, it is difficult for them to infer the original transaction information. From this, it can be seen that the blockchain transaction-privacy intelligent-protection method based on lightweight homomorphic encryption performs well in terms of privacy and security-protection data exposure probability, providing strong privacy-protection capabilities.

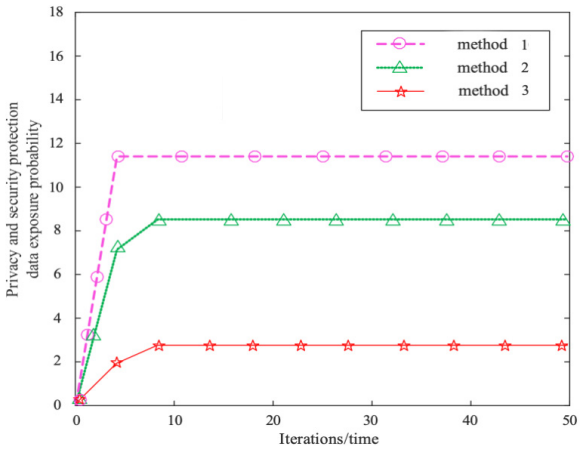

Figure 3. Comparison results of privacy and security-protection data exposure probabilities. (Method 1 references the method in [5], Method 2 references the method in [6], and Method 3 is the method used in this paper).

(2) Anti-attack capability

By simulating different types of attack scenarios (such as replay attacks, forged transactions, quantum attacks, etc.), this paper verifies the anti-attack performance of blockchain transaction privacy-protection methods. This paper mainly simulates two common attack scenarios—replay attacks and forged transactions. Replay attack refers to an attacker attempting to rebroadcast confirmed legitimate transactions on the blockchain network in order to gain double benefits or disrupt system stability. Fake transactions refer to attackers attempting to construct and broadcast false transaction records in order to deceive verification nodes or steal assets. Next, set attack conditions by selecting a certain number of confirmed transactions and attempting to rebroadcast these transactions on the blockchain network. At the same time, construct false transaction records, including false input addresses, output addresses, and transaction amounts, and attempt to broadcast them to the blockchain network.

Under the above experimental conditions, the anti-attack performance of the methods in References [5,6], and our proposed method were verified, and the results are shown in Table 1.

Table 1. Comparison Results of Anti-Attack Capability.

Attack Type	Attack Conditions	Privacy-Protection Methods	Successfully Blocked
Replay attack	Broadcast 10 confirmed transactions	Reference [5] Method	yes
		Reference [6] Method	no
		Method of this article	yes
Counterfeit transactions	Construct 50 false transactions	Reference [5] Method	no
		Reference [6] Method	yes
		Method of this article	yes

According to Table 1, the method proposed in Reference [5] can successfully prevent replay attacks, but cannot prevent forged transactions. The method in Reference [6] can

successfully prevent forged transactions, but cannot prevent replay attacks. However, the method proposed in this article can not only prevent replay attacks, but also effectively prevent forged transactions. This indicates that this method not only ensures the uniqueness and immutability of transactions, making confirmed transactions unable to be rebroadcasted and reconfirmed, but also ensures the legality and authenticity of transactions, making false-transaction records unable to be accepted and confirmed by verification nodes. This proves that the method proposed in this article has high robustness and stability in the face of replay attacks and forged transactions. Through experimental verification, it can be found that the privacy-protection method for blockchain transactions selected in this article performs well in the face of common attack scenarios such as replay attacks and forged transactions. It can successfully prevent these attacks and protect the privacy and security of transactions. This provides strong support and guarantee for the widespread application of blockchain technology.

(3) Confusion degree

Using confusion degree as an experimental indicator to verify the effectiveness of the method proposed in this paper, confusion degree is a quantitative indicator used to measure the degree to which privacy-protection methods conceal or obfuscate transaction information. Use privacy-protection methods to process raw transaction data and generate confusing transaction data. Compare and analyze the original transaction data with the confused transaction data, and calculate the degree of confusion. A higher degree of confusion means that privacy-protection methods can better conceal or obfuscate transaction information, thereby improving the privacy of transactions. The experimental results are shown in Figure 4.

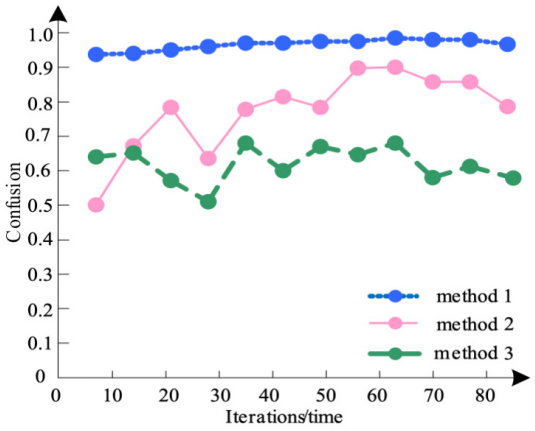

Figure 4. Confusion comparison results. (Method 1 is the approach used in this paper, Method 2 references the method in [5], and Method 3 references the method in [6]).

From Figure 4, it can be seen that the methods in Reference [5] and Reference [6] have significant fluctuations, indicating poor stability of these two methods. There may be privacy leakage risks in certain special situations, and the degree of confusion is lower than that of the method proposed in this paper. The confusion level of the method in this article is relatively high, maintained at above 0.9 and with small fluctuations. By comparing the degree of confusion among different methods, it can be concluded that the privacy protection performance of our method is better, making it a more suitable privacy-protection method for practical applications.

(4) Lightweight homomorphic encryption effect

"Lightweight" means optimizing encryption methods in terms of computational efficiency, resource consumption, and implementation costs to better adapt to the characteristics and needs of blockchain networks. Therefore, this article tests the encryption performance of different methods from two aspects: runtime and computational overhead (CPU

usage). The comparison results of the runtime of different blockchain privacy-protection methods under different key lengths are shown in Table 2.

Table 2. Comparison of runtime results of different blockchain privacy-protection methods under different key lengths.

Key Length/Bit	Different Stages	The Proposed Method	Reference [5] Method	Reference [6] Method
1024	Key generation	3.4	28.7	15.3
	encryption	6.2	8.9	9.5
	Decryption	5.9	8.2	6.7
2048	Key generation	5.0	35.1	20.3
	encryption	8.7	12.8	10.0
	Decryption	7.2	11.9	8.3
3072	Key generation	8.6	40.2	23.0
	encryption	9.6	15.9	12.6
	Decryption	7.9	13.4	9.9

The data in Table 2 indicate that in all tested methods, as the key length increases, the running time also increases accordingly. This is because longer keys require more computing resources for encryption and decryption operations. The methods in References [5,6] may have relatively short runtime when the key length is short, but as the key length increases, their runtime may increase rapidly, indicating a high demand for computing resources. The key generation, encryption, and decryption efficiency of the method proposed in this article is higher than that of the methods in References [5,6], indicating that the method proposed in this article can enable blockchain systems to more efficiently protect user privacy while ensuring transaction legality, and has good application prospects.

The comparison results of CPU usage using different blockchain privacy-protection methods are shown in Figure 5.

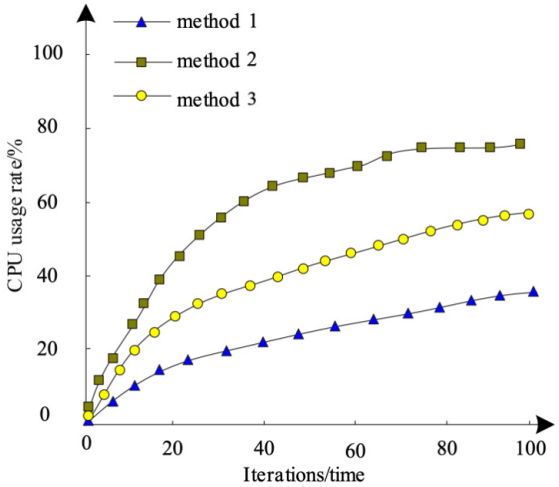

Figure 5. CPU-usage comparison results. (Method 1 references the method in [5], Method 2 references the method in [6], and Method 3 is the method used in this paper).

A lower CPU utilization rate means that the method is more efficient in processing data, and can fully utilize CPU resources while maintaining lower energy consumption and heat generation, which is a positive factor for blockchain privacy-protection systems that run for a long time or are deployed on a large scale. However, if the CPU usage is too low, it may indicate unnecessary computational redundancy or optimization space in the method when processing data, which may lead to insufficient resource utilization or suboptimal performance. On the other hand, if the CPU usage is too high, it may lead to a decrease in system performance, slower response times, or even crashes. This may be

because the method consumes too many CPU resources during the calculation process, resulting in other tasks not being able to obtain sufficient computing resources. According to Figure 5, as the number of iterations increases, the CPU usage of the three methods shows a gradually increasing trend. Among them, the CPU usage of the method in Reference [6] is too high, the CPU usage of the method in Reference [5] is too low, and the CPU usage of the method in this paper is moderate, indicating that it can ensure that the system can efficiently process data while maintaining low energy consumption and heat generation, thus demonstrating good performance in practical applications.

Based on the above experimental results, it can be concluded that lightweight homomorphic encryption can be used to protect the privacy information of both parties in blockchain transactions, such as transaction amount and the identities of both parties. Through homomorphic encryption technology, encrypted data can be computed without disclosing plaintext information, ensuring the security and privacy of transaction data in blockchain networks. Meanwhile, due to its lower computational complexity and resource consumption, lightweight homomorphic encryption can better adapt to the high concurrency, low latency, and other characteristics of blockchain networks, ensuring the efficiency and real-time performance of transactions.

(5) Data leakage rate

Using leakage rate as an experimental indicator, leakage rate is used to measure the degree of sensitive-information leakage that privacy protection methods may cause when processing transaction data. After processing the original transaction data through privacy-protection methods, a set of protected transaction data is generated. Next, we will conduct a detailed comparative analysis between the original transaction data and the protected transaction data, to calculate the leakage rate. The lower leakage rate indicates that privacy-protection methods can effectively reduce the leakage of sensitive information, thereby enhancing the level of privacy protection for transactions. The experimental results are shown in Figure 6.

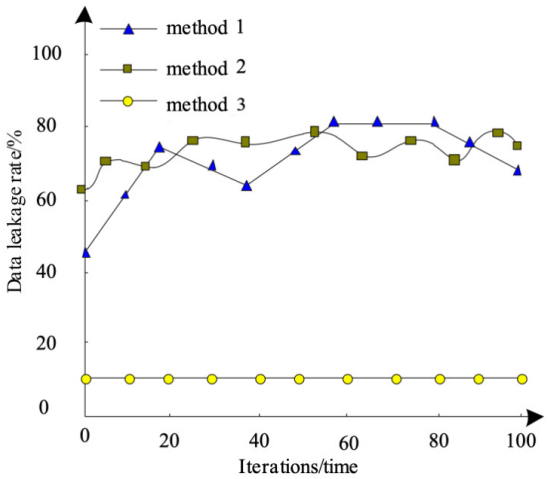

Figure 6. Comparison results of data leakage rates. (Method 1 references the method in [5], Method 2 references the method in [6], and Method 3 is the method used in this paper).

From the analysis of Figure 6, it can be seen that, compared with traditional privacy-protection methods, the method proposed in this paper exhibits significant advantages in terms of data leakage rate. The data curve in Figure 6 clearly reveals that traditional methods result in a higher rate of sensitive-information leakage during processing on the same transaction dataset, while our method effectively controls the leakage rate at a lower level. This result not only proves the effectiveness of our method in privacy protection, but also highlights its practical application value in improving transaction data security and user-privacy protection. By adopting the method described in this article, we can provide

users with a more secure and reliable privacy-protection environment, thereby enhancing their trust and satisfaction with the system while protecting their privacy.

(6) Limitations

Based on the experimental results, it can be concluded that lightweight homomorphic encryption can protect privacy information in blockchain transactions, such as transaction amounts and identities. However, the proposed method has some limitations in practical applications. In larger blockchain networks [20] or higher transaction volumes, scalability challenges may arise. The computational overhead of homomorphic encryption algorithms, especially the Paillier algorithm, can lead to increased latency and decreased throughput as the network scale expands. The method's effectiveness relies on certain security assumptions, such as the presence of trusted nodes and the absence of advanced persistent threats (APTs). If these assumptions do not hold, the privacy and security guarantees may be compromised. The robustness of the method may be reduced in scenarios involving sophisticated adversaries or partially trusted nodes. Further research and extensive testing are necessary to address limitations in untested edge cases or specific attack scenarios. The homomorphic encryption algorithm used is suitable for certain types of computations, but more complex operations may degrade performance. The results are based on specific experimental settings and may not apply to all blockchain platforms or applications, requiring further validation in different use cases. Finally, implementing and maintaining the proposed privacy-protection method involves economic and resource costs, which could be burdensome for small-scale or resource-constrained organizations.

4. Conclusions

This article proposes an privacy-protection method for blockchain transactions based on lightweight homomorphic encryption, exploring how to ensure data security and privacy protection during blockchain transactions, while improving transaction efficiency and credibility. The experimental results show that this method has lower computational complexity and resource consumption, while ensuring privacy and security. This enables it to better adapt to the high-concurrency and low-latency characteristics of blockchain networks, ensuring the efficiency and real-time performance of transactions. This efficient and low-resource-consumption privacy-protection method is of great significance for the widespread application of blockchain technology. By implementing this method, the probability of data exposure for privacy and security protection is only 2.8%, and the confusion level remains above 0.9, with small fluctuations. These indicators demonstrate the efficiency and reliability of the method in protecting transaction privacy, further demonstrating its innovation and practicality.

Author Contributions: Conceptualization, S.Z.; methodology, C.L. and B.D.; software, G.W.; validation, G.W.; investigation, C.L. and B.D.; writing—original draft preparation, G.W.; writing—review and editing, G.W. and C.L.; supervision, C.L.; project administration, C.L. and S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

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

References

1. Zhu, X.; Xing, C.; Li, W.; Hao, Y. Evaluation Method of Blockchain Privacy Protection for Full Life Cycle of Transaction Data. *J. Appl. Sci.* **2022**, *40*, 555–566.

2. Li, Y.; Zhou, K.; Wang, Z. A Survey of Block Chain Privacy Protection Research Based on Zero-Knowledge Proof. *Aerosp. Control. Appl.* **2022**, *48*, 44–52.
3. Zhang, Y.; Zhang, M. Simulation of Personal Information Privacy Protection Algorithm under Private Blockchain. *Comput. Simul.* **2023**, *40*, 397–401.
4. Yu, J.; Mu, R.; Qin, R.; Zhang, J. A privacy protection scheme for blockchain transaction information. *J. Chongqing Univ. Posts Telecommun. (Nat. Sci. Ed.)* **2022**, *56*, 1–15.
5. Cai, L.; Duan, H.; Yan, M.; Xia, X. Private Data Protection Scheme for Consortium Blockchain Based on Two-layer Cooperation. *J. Softw.* **2020**, *31*, 2557–2573.
6. Fan, J.; Chen, J.; Shen, R.; Liu, Z.; He, Q.; Huang, B. SGX-Based Approach for Blockchain Transactions Security and Privacy Protection. *J. Appl. Sci.* **2021**, *39*, 17–28.
7. Li, L.; Du, H.; Li, T. Blockchain Supervisable Privacy Protection Scheme Based on Group Signature and Attribute Encryption. *Comput. Eng.* **2022**, *48*, 132–138.
8. Shen, M.; Che, Z.; Zhu, L.-H.; Xu, K.; Gao, F.; Yu, C.-C.; Wu, Y. Anonymity in Blockchain Digital Currency Transactions:Protection and Confrontation. *Chin. J. Comput.* **2023**, *46*, 125–146.
9. Liu, H.; Zhou, Y.; Zhou, X.; Xie, J. Review on privacy threats and protection mechanisms for blockchain technology. *Comput. Integr. Manuf. Syst.* **2023**, *29*, 2292–2312.
10. Abdulkader, M.M.; Kumar, S.G. A privacy-preserving data transfer in a blockchain-based commercial real estate platform using random address generation mechanism. *J. Supercomput.* **2023**, *79*, 10796–10822. [CrossRef]
11. Ahmed, F.; Wei, L.; Niu, Y.; Zhao, T.; Zhang, W.; Zhang, D.; Dong, W. Toward fine-grained access control and privacy protection for video sharing in media convergence environment. *Int. J. Intell. Syst.* **2022**, *37*, 3025–3049. [CrossRef]
12. Abdulmunim, A.T.; Tao, X.; Zhang, J.; Zhou, X.; Li, L.; Cai, Y. A Survey of Privacy Solutions using Blockchain for Recommender Systems: Current Status, Classification and Open Issues. *Comput. J.* **2021**, *64*, 1104–1129.
13. Hewa, T.; Braeken, A.; Liyanage, M.; Ylianttila, M. Fog Computing and Blockchain-Based Security Service Architecture for 5G Industrial IoT-Enabled Cloud Manufacturing. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7174–7185. [CrossRef]
14. Rahman, Z.; Khalil, I.; Yi, X.; Atiquzzaman, M. Blockchain-based Security Framework for Critical Industry 4.0 Cyber-physical System. *IEEE Commun. Mag.* **2021**, *59*, 128–134. [CrossRef]
15. Andreina, S.; Bohli, J.M.; Karame, G.; Li, W.; Marson, G.A. PoTS: A Secure Proof of TEE-Stake for Permissionless Blockchains. *IEEE Trans. Serv. Comput.* **2022**, *15*, 2173–2187. [CrossRef]
16. Singh, S.; Satish, D.; Lakshmi, S.R. Ring signature and improved multi-transaction mode consortium blockchain-based private information retrieval for privacy-preserving smart parking system. *Int. J. Commun. Syst.* **2021**, *34*, 4911.1–4911.20. [CrossRef]
17. Waheed, N.; He, X.; Ikram, M.; Usman, M.; Hashmi, S.S. Security and Privacy in IoT Using Machine Learning and Blockchain: Threats and Countermeasures. *ACM Comput. Surv.* **2021**, *53*, 122.1–122.37. [CrossRef]
18. Baza, M.; Sherif, A.; Mahmoud, M.M.E.A.; Bakiras, S.; Alasmary, W.; Abdallah, M.; Lin, X. Privacy-Preserving Blockchain-Based Energy Trading Schemes for Electric Vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9369–9384. [CrossRef]
19. Qian, Y.; Jiang, Y.; Hu, L.; Hossain, M.S.; Al-Hammadi, M. Blockchain-Based Privacy-Aware Content Caching in Cognitive Internet of Vehicles. *IEEE Netw.* **2020**, *34*, 46–51. [CrossRef]
20. Allende, M.; León, D.L.; Cerón, S.; Pareja, A.; Pacheco, E.; Leal, A.; Da Silva, M.; Pardo, A.; Jones, D.; Worrall, D.J.; et al. Quantum-resistance in blockchain networks. *Sci. Rep.* **2023**, *13*, 5664. [CrossRef] [PubMed]