ISSN: 2710-0588

Research on Blockchain Interactive Zero Knowledge Proof Privacy Protection Scheme Based on Improved Paillier Homomorphic Encryption

Yueran Zhuo*

Cyber Security, Huazhong University of Science and Technology, Wuhan, Hubei, 430000, China

*Corresponding author's e-mail: u202112025@hust.edu.cn

Abstract

In the context of the digital age, data privacy and security issues are increasingly prominent. Blockchain technology plays an important role in data sharing due to its transparency and immutability, but it also brings the risk of privacy leakage. Zero knowledge proof technology provides a solution for verifying data correctness without exposing data content, which is particularly important for blockchain as it can ensure the validity and compliance of transactions while protecting user privacy. Although zero knowledge proof is quite mature in theory, its application in blockchain systems still faces challenges such as computational efficiency, complexity of smart contracts, and system compatibility. This study aims to propose a privacy protection scheme that supports interactive zero knowledge proof by improving the homomorphic encryption Paillier algorithm, in order to enhance the privacy protection capability of blockchain systems and maintain system efficiency and security. The study will adopt an interdisciplinary approach, combining cryptography, computer science, and network security theory, to deeply analyze the application effect of zero knowledge proof technology in blockchain, explore its optimization space and applicability.

Keywords

Zero-knowledge proofs, Privacy, Blockchain, Cryptography.

1. Introduction

In the digital age, data privacy and security have become the focus of global attention. Blockchain technology[1], with its innovative data processing methods, provides new avenues for building trust and transparency. However, while the transparency of blockchain brings convenience to data sharing, it also raises the risk of privacy[2] leakage, especially in areas such as financial transactions involving sensitive information, medical records, and intellectual property. The existence of this contradiction has prompted researchers to explore new methods to enhance the privacy protection capabilities of blockchain.

Zero knowledge proof technology, as a method of verifying the correctness of data without exposing its content, provides the possibility to solve the above contradictions. It allows the verifier to demonstrate to the verifier that a statement is correct without providing any information other than correctness. This feature is particularly important for blockchain as it can ensure the effectiveness and compliance of transactions without sacrificing user privacy.

Although zero knowledge proof technology is quite mature in theory, its application in blockchain systems still faces many challenges. For example, the computational efficiency of zero knowledge proof algorithms, the complexity of smart contracts, and compatibility with existing blockchain systems all require further research and optimization. In addition, how to combine zero knowledge proof technology with existing cryptographic tools, such as

ISSN: 2710-0588

homomorphic encryption algorithms, to achieve more efficient privacy protection schemes is also a current research hotspot.

This study aims to propose an improved homomorphic encryption Paillier algorithm through in-depth analysis and experimental verification to support zero knowledge proof and enhance its application efficiency in blockchain systems. Our goal is to develop a solution that can protect user privacy while maintaining the efficiency and security of blockchain systems. To this end, we will adopt interdisciplinary research methods, combining theories and technologies from fields such as cryptography, computer science, and network security.

In reference [3] Qin Jie et al. proposed a privacy protection scheme that combines zero knowledge proof and homomorphic encryption, which has shown good application prospects in both theory and experiment. On this basis, this study will further explore the optimization space of algorithms, reduce computational costs, and improve the scalability and practicality of the system. By constructing models, designing algorithms, and conducting simulation experiments, we will systematically analyze the application effects of zero knowledge proof technology in blockchain, and explore its applicability and limitations in different scenarios.

1.1. Blockchain

Blockchain technology is a revolutionary distributed database system that allows multiple parties to jointly maintain a continuously growing list of data records, which are organized into a series of data structures called "blocks" and securely linked together through cryptographic methods. Blockchain has a centralized nature, which means it does not rely on any single centralized entity to store or manage data. Each node in the network holds a copy of the blockchain and participates in the verification and maintenance of data together. At the same time, blockchain has extremely high immutability, and once data is added to the blockchain, it is almost impossible to change or delete. The transparency of blockchain is also one of its significant features. All transaction records are visible to participants in the network, ensuring the transparency and traceability of transactions. Smart contracts are a key innovation in blockchain technology that allows for the deployment of automatically executed contracts on the blockchain. The terms of these contracts are directly written in the code, and when specific conditions are met, the smart contract will automatically perform relevant operations.

With the widespread application of blockchain technology, its privacy protection issues are becoming increasingly prominent. In order to ensure the security of transaction data and user privacy on blockchain, researchers have proposed various privacy protection technologies, mainly including decentralized identity authentication, privacy protocols, encryption technology, and obfuscation technology.

Decentralized identity authentication: Through blockchain technology, decentralized management of user identities is achieved, avoiding the single point of failure and data leakage risks of traditional centralized identity authentication systems.

Privacy protocols: such as Secure Multi Party Computing (SMPC) and Differential Privacy (DP), enable data sharing and analysis while protecting user privacy.

Encryption technology: including homomorphic encryption (HE), zero knowledge proof (ZKP), etc., which can calculate or verify data without decrypting it, thus protecting the privacy of the data.

Obfuscation technology: such as Coin Mixing and Ring Signature, improve the anonymity of transactions by confusing the information of transaction sources and recipients.

The application scope of blockchain technology is very extensive, from the initial cryptocurrency such as Bitcoin[4] to various fields such as financial services, supply chain management, Internet of Things, copyright protection, voting systems, etc., blockchain is playing an important role. The development of blockchain technology is still evolving, providing

ISSN: 2710-0588

a new paradigm for data storage, management, and transactions, while also bringing a series of technical challenges and opportunities. With the maturity of technology, it is foreseeable that blockchain will play an important role in more fields, promoting further social and economic development.

1.2. Zero-Knowledge Proofs

Zero knowledge proofs (ZKPs) are a cutting-edge cryptographic technique that allows one party (verifier) to prove to the other party (verifier) that a statement is correct without revealing any information beyond that statement. This technology is crucial for enhancing data security and protecting privacy, especially in distributed ledger technologies such as blockchain, which can effectively address privacy protection issues during transaction processes.

Specifically, interactive zero knowledge proof includes the following steps:

Setup: The verifier and verifier jointly determine a common reference string (CRS) or other common parameters, which will be used in the subsequent proof process.

Problem statement: The verifier selects a statement that needs to be proven and prepares the corresponding proof process.

First round of interaction: The verifier sends some preliminary information to the verifier, such as commitments related to the statement to be proven.

Challenge: The verifier generates a challenge based on the information provided by the verifier, which is usually a random number or a series of random numbers used to test the knowledge claimed by the verifier.

Response: After receiving the challenge, the verifier generates a response based on their knowledge of the problem, which needs to be able to answer the verifier's challenge without leaking any additional information.

Subsequent rounds: According to the specific protocol of zero knowledge proof, there may be multiple rounds of challenges and responses. In each round, the verifier will generate new challenges based on the information from the previous round, while the verifier needs to generate new responses accordingly.

Verification: After the final round of interaction, the validator will check whether all responses from the verifier meet the validation conditions. If the response meets the conditions, the verifier accepts the proof, believing that the verifier indeed knows the claimed knowledge; If not satisfied, reject the proof.

Output result: The validator outputs the verification result, which is a Boolean value indicating whether the proof was successful.

1.3. Paillier algorithm

Paillier algorithm is a probability based public key cryptography system proposed by Pascal Paillier in 1999. It is a special homomorphic encryption algorithm that supports addition operations on ciphertext while maintaining the security and correctness of ciphertext. The Paillier algorithm mainly includes:

Public and private key generation:

Select two large prime numbers p and q, and calculate their product n=pq.

Calculate $\lambda=\text{lcm}(p-1,q-1)$, which is the least common multiple of p-1 and q-1.

Choose a random number g such that $g^2 \equiv 1 (\text{mod} \lambda n)$.

The public key consists of n and g, denoted as (n, g); the private key is λ and n.

Encryption process:

For plaintext m, select a random number r, where $0<r<n$.

Calculate ciphertext c, defined as $c=g^m * r^n (\text{mod} n^2)$.

ISSN: 2710-0588

Decryption process:

Using the private key λ and public key n, calculate the decryption result m, using the formula $m=(L(c^{\lambda} \bmod n^2)-1)/\lambda(\bmod n)$,where L(x) is the function that calculates the quotient of x divided by n.

The advantage of Paillier algorithm lies in its simple implementation, intuitive principle, and ability to provide homomorphic encryption function, making it possible to perform certain calculations on ciphertext without decryption. However, one of its main limitations is that it only supports addition operations and does not support multiplication or other more complex operations. Therefore, in this study, the Karatsuba algorithm is introduced to accelerate large number multiplication.

2. Literature Review

2.1. Early development (before 1989)

2.1.1. Cryptography

Cryptography, as an important tool for protecting information security, has a long history. As early as ancient times, people began to use various methods to hide and protect information, which can be seen as the embryonic form of classical cryptography. For example, the ancient Egyptians used hieroglyphs, while the Spartans of ancient Greece used a simple replacement password called the Spartan code. However, these early cryptographic methods were mostly based on simple substitution and permutation techniques, lacking systematicity and scientificity.

In the late 1940s, Claude Shannon's series of papers opened up new avenues for the development of cryptography. In 1949, Shannon published a landmark paper titled "Communication Theory of Secure Systems", in which he first proposed the concept of information theory and provided a mathematical foundation for secure communication. Shannon's work not only laid the theoretical foundation for the development of cryptography, but also made important contributions to the later development of digital communication and information theory.

Based on Shannon's theory, in the 1970s, the National Institute of Standards and Technology (NIST) in the United States developed the Data Encryption Standard (DES) algorithm. DES is a symmetric key encryption algorithm that uses a 56 bit key length to encrypt data. Although DES was considered very secure when it was launched, it gradually became perceived as insecure with the improvement of computing power and was eventually replaced by more advanced algorithms.

Following closely behind, in 1978, Ron Rivest, Adi Shamir, and Leonard Adleman jointly invented the RSA algorithm. RSA is an asymmetric encryption algorithm that provides a more secure encryption method based on the difficulty of large integer factorization. The emergence of RSA algorithm marks the birth of modern public key cryptography. It has not only been widely used in commercial and personal communication, but also become a core component of many security protocols and blockchain technologies.

2.1.2. The Concept of Zero Knowledge Proofs

In 1989, researchers Shafi Goldwasser and Silvio Micali from the Massachusetts Institute of Technology, along with their colleague Charles Rackoff, first proposed the concept of "Zero Knowledge Proof". This is a cryptographic technique that allows one party to prove to the other that a statement is correct without providing any information other than the correctness of the statement. The core of this proof method is that even if the verifier cannot independently verify the authenticity of the statement, they can still be confident that the statement is true without obtaining any additional information.

ISSN: 2710-0588

2.2. The Rise of Blockchain Technology (2008)

2.2.1. The Birth of Bitcoin

In 2008, a mysterious figure named Satoshi Nakamoto published a white paper titled "Bitcoin: A peer-to-peer electronic cash system", marking the birth of blockchain technology and the emergence of Bitcoin as a digital currency. The design goal of Bitcoin is to create a decentralized electronic trading method that allows users to conduct currency transactions without a central authority.

Bitcoin's blockchain technology is based on the concept of distributed ledgers, where all transaction records are encrypted and added to a public, tamper proof ledger. This design ensures the transparency and irreversibility of transactions, but it also brings privacy issues. Because although Bitcoin addresses are pseudonymous, all transaction records are public, which means that if someone can associate Bitcoin addresses with real-world identities, the user's transaction history can be tracked and analyzed.

To address this issue, the Bitcoin community and cryptography researchers have been exploring various privacy protection technologies. For example, using multiple signature addresses can increase the complexity of transactions and make tracking more difficult. In addition, some service and wallet providers offer CoinJoin or similar features that confuse the source and destination of transactions by merging the transactions of multiple users together.

2.3. Preliminary Application of Zero Knowledge Proof in Blockchain (2013)

2.3.1. Zerocoin

As the first cryptocurrency project to introduce zero knowledge proof technology into blockchain, Zerocoin was designed to enhance the privacy of digital currencies such as Bitcoin. In Bitcoin transactions, although the address is pseudonymous, all transaction records are public, which poses a risk of tracking the flow of funds and transaction behavior of users. The emergence of the Zerocoin project provides an innovative solution to address this issue.

Zerocoin provides users with an anonymous way of trading by introducing a special cryptocurrency unit - "Zero coin". Users can exchange their Bitcoin for coins, a process known as minting. During the casting process, users use Bitcoin as input and generate coins through a series of cryptographic operations. These operations include using zero knowledge proof to prove that they have a certain amount of Bitcoin without revealing the specific number or source of Bitcoin.

When users want to use coins for transactions, they can "spend" these coins. During the spending process, users once again use zero knowledge proof to prove that their coins are legally minted and the transaction is valid, without the need to disclose any information about the original Bitcoin transaction. In this way, both the input and output of the transaction are hidden, achieving a high degree of anonymity.

However, the implementation of Zerocoin is not without challenges. Firstly, Zerocoin requires a complex initialization process called "Trusted Setup". This process involves generating a set of common parameters that are shared throughout the entire Zerocoin system. If the key is leaked during the trust setting process, the security of the entire system will be threatened. Therefore, trust setting needs to be carried out in a highly secure environment and requires the participation of multiple participants to ensure the security of the key.

Secondly, the transaction verification process of Zerocoin is relatively complex and requires more computing resources and time. This may affect the performance of blockchain, especially in situations with high transaction volumes. Finally, the anonymity of Zerocoin may be compromised by some analytical techniques, such as inferring the user's identity by analyzing transaction patterns and times.

ISSN: 2710-0588

2.3.2. Zerocash

Zerocash,as an improved version of Zerocoin, it represents a significant advancement in privacy protection technology in the field of blockchain. The Zerocash project was developed by a team led by Zooko Wilcox. It not only inherits Zerocoin's core philosophy of protecting user transaction privacy through zero knowledge proof technology, but also simplifies the use of zero knowledge proof by introducing zk SNARKs technology, improving transaction efficiency and scalability.

Zk SNARKs are a special form of zero knowledge proof that allows the prover to prove the correctness of a statement without providing any transaction details. Compared with traditional zero knowledge proofs, zk SNARKs are more concise and efficient because they do not require validators to hold a large number of common parameters, thereby reducing the burden of storage and computation. In Zerocash, zk SNARKs are used to demonstrate the validity of a transaction, including the sender having sufficient funds, the correctness of the transaction amount, and the legality of the transaction, without exposing any information about the parties or transaction amount.

Zerocash transactions are entirely anonymous on the blockchain, which means that the input and output of the transaction will not be publicly displayed on the blockchain. This brings unprecedented levels of privacy protection to users, making it impossible for external observers to track the flow of funds or identify the identities of transaction participants. Meanwhile, Zerocash maintains the verifiability and immutability of transactions, ensuring the security and trustworthiness of blockchain.

2.4. The rise of ZK Stark and Bulletproof (after 2018)

2.4.1. ZK-Stark

ZK-Stark is a novel zero knowledge proof technique aimed at overcoming certain limitations of ZK-Snark, especially in problems that require trustworthy settings. Compared with traditional ZK-Snark, ZK-Stark does not require a trusted initialization process, which makes it more decentralized and secure in theory.

The core advantage of ZK-Stark lies in its transparency and trustless setup. It uses a feature called transparency that allows anyone to verify the correctness of the proof without relying on a pre generated set of potentially vulnerable parameters. This characteristic makes ZK-Stark theoretically more resistant to quantum computing attacks, as quantum computers cannot crack the mathematical proof used by ZK-Stark.

In addition, the design of ZK-Stark enables it to handle more complex computational tasks, rather than just simple transaction verification. This means that it can be applied to a wider range of blockchain scenarios, such as smart contracts, authentication, supply chain management, etc.

2.4.2. Bulletproof

Bulletproofs is an efficient non interactive zero knowledge proof technique that is particularly suitable for scope proof, where the prover can prove to the verifier that a certain value is within a given range without revealing specific information about the value. This technology is particularly important for blockchain applications that need to protect user privacy, as it can achieve high privacy protection without sacrificing efficiency.

The core advantage of Bulletproofs lies in its simplicity and the feature of not requiring trusted settings. Compared to ZK Snark, Bulletproofs does not require the generation and storage of a set of trusted public parameters during system initialization, greatly reducing security risks caused by parameter leakage or abuse. In addition, the proof size of Bulletproofs is independent of the numerical range of the proof, which means that regardless of the size of the numerical range, the communication cost of the proof remains relatively stable.

ISSN: 2710-0588

In the field of blockchain, the application of Bulletproofs has achieved significant results, especially in the cryptocurrency project of the MimbleWimble protocol. MimbleWimble is a blockchain protocol that emphasizes privacy and scalability, using Bulletproofs to achieve transaction confidentiality and efficient verification. In MimbleWimble, all transaction amounts and sender/receiver information are encrypted, and only the transaction parties know the specific values. By using Bulletproofs, the validity of transactions can be verified without disclosing sensitive information.

2.5. Exploration of Layer 2 Expansion Plan (After 2020)

With the development of blockchain technology, Layer 2 scalability solutions have become the key to solving blockchain performance bottlenecks. Optimal Rollup and ZK Rollup are two mainstream Layer 2 solutions. Optimistic Rollup adopts a fraud proof mechanism, which only needs to generate fraud proof in case of disputes, thereby reducing verification costs but may sacrifice certain financial efficiency. ZK Rollup, on the other hand, uses zk SNARK or other zero knowledge proof techniques to provide validity proof for each batch of transactions, ensuring that the transactions are always in an effective state, improving financial efficiency but increasing computational costs. These two solutions each have their own advantages and disadvantages, and are suitable for different application scenarios and requirements.

3. System Scheme

Based on the current status of zero knowledge proof technology, work on privacy protection of transaction data based on zero knowledge proof can be carried out from the perspective of blockchain privacy protection based on zero knowledge proof.

3.1. Scheme model

This article presents a transaction data privacy protection model based on interactive zero knowledge proof, as shown in Figure 1. The scheme design includes blockchain, transaction certificate storage service, zero knowledge proof algorithm, and homomorphic encryption algorithm to create a privacy protection model to achieve transaction security and privacy protection in blockchain systems.

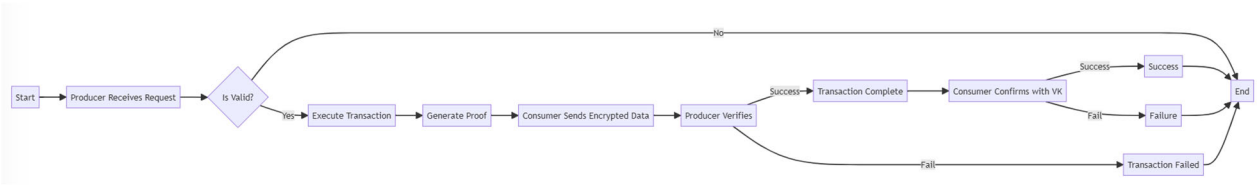

Figure 1. On chain zero knowledge verification model

In addressing the challenges of data integrity and security faced by traditional blockchain systems, this study focuses on significantly improving the efficiency of key generation while strengthening privacy protection. To achieve this goal, we innovatively combine interactive zero knowledge proof technology with efficient mathematical algorithms - the Montgomery algorithm and the Karatsuba algorithm - to overcome computational bottlenecks caused by power modular operations and large number multiplication in the key generation process, thereby enhancing the overall performance and security of the blockchain system.

Blockchain technology, as a decentralized and tamper proof distributed ledger, ensures the security and transparency of data through cryptographic means. However, in practical application scenarios such as consortium chains, although blockchain platforms such as Hyperledger Fabric have been widely deployed in China, the complete transparency of on chain data has also raised concerns about data privacy breaches. In order to ensure the privacy of

ISSN: 2710-0588

data without sacrificing the decentralized nature of blockchain, this study explores in depth how to build a more robust privacy protection barrier between the underlying transaction information and the actual application layer, and enhance the verifiability and execution efficiency of smart contracts.

Specifically, we introduce interactive zero knowledge proof technology, which allows one party (the verifier) to prove the authenticity of a statement to the other party (the verifier) without revealing any additional information, thus achieving a perfect balance between privacy and verification. In order to further improve the efficiency of key generation, we integrated the Montgomery algorithm and Karatsuba algorithm. The Montgomery algorithm significantly accelerates the speed of modular exponentiation by reducing the size of intermediate results, which is a crucial step in key generation. The Karatsuba algorithm, on the other hand, optimizes the execution process of large number multiplication by reducing the number of multiplications to reduce computational complexity and further improve the efficiency of key generation.

On this basis, we propose an innovative blockchain privacy protection scheme that combines interactive zero knowledge proof with off chain computing. By performing complex computational tasks offline (such as optimizing key generation using Montgomery and Karatsuba algorithms) and converting the results into verifiable evidence in the form of zero knowledge proofs, these proofs are then submitted to smart contracts on the chain for verification. This off chain computing and on chain verification mode not only ensures the privacy of data, but also ensures the correctness and security of results through the automated verification mechanism of smart contracts.

In summary, this study not only solves the performance bottleneck in blockchain key generation by introducing interactive zero knowledge proof technology and efficient mathematical algorithms, but also constructs a more secure and efficient blockchain privacy protection system, providing strong support for the widespread application of blockchain technology in various fields.

3.2. Scheme Description

3.2.1. Interactive zero knowledge proof process

In this research scheme, the transaction process of both parties is carried out at the application layer. In order to ensure high confidentiality of transaction privacy and support the legitimacy verification of ciphertext using smart contracts, we adopted interactive zero knowledge proof technology. This process not only involves homomorphic encryption techniques to generate ciphertext for transaction data, but also incorporates interactive zero knowledge proof mechanisms to construct necessary evidence such as equality proof and range proof. Subsequently, these information and encrypted transaction data are sent to the transaction layer of the blockchain, namely the chain code end, which is responsible for verifying the validity of the transaction.

The core of the interactive zero knowledge proof framework lies in its interactivity, which allows for the gradual construction of trust between the prover and verifier through a series of queries and responses. The specific process is shown in Figure 2, which can be further refined into the following key steps:

Figure 2. Interactive zero knowledge proof generation process

ISSN: 2710-0588

(1) Formalizing the problem: Firstly, transform the transaction data or computational problem to be proven into a form that can be understood by an interactive zero knowledge proof system, such as by transforming it into the vector dot product form of Quadratic Arithmetic Programs (QAP) or Rank-1 Constraint System (R1CS), in order to construct the mathematical foundation required for the proof.

(2) Initialization and Key Generation: The key manager (an entity assumed to be absolutely trusted) generates the necessary Proving Key (PK) and Verification Key (VK). These keys will be used for subsequent proof generation and verification processes.

(3) Proof generation: The verifier (usually the producer or sender of the transaction) in the transaction uses PK and transaction data to generate interactive zero knowledge proofs. During this process, the verifier will calculate a series of mathematical proof values, which will be used to respond to the verifier's inquiry without disclosing any privacy information about the transaction data itself.

(4) Challenge and Response: The verifier (possibly a node or smart contract in a blockchain network) sends a challenge to the verifier, requesting specific proof values to prove the legitimacy of their transaction data or to meet specific conditions. The verifier provides responses based on the inquiry, which are sufficient to convince the verifier of the authenticity of the transaction data without directly examining the transaction data itself.

(5) Verification: The verifier uses VK and the received proof response to verify the correctness of the proof. If all inquiries receive reasonable responses and meet the preset verification rules, the verifier accepts the transaction as valid and records it on the blockchain.

(6) Through the interactive zero knowledge proof process mentioned above, this research scheme achieves the legitimacy verification of encrypted transaction data by smart contracts while ensuring the privacy of transaction data, thereby enhancing the transaction security and privacy protection capabilities of blockchain systems.

3.2.2. Secure multi-party computation process

Secure multi-party computation is a technique that allows multiple participants to jointly compute a function without leaking their respective input data. In transaction data privacy protection schemes, this process is particularly crucial as it ensures that even if data is transmitted and processed between multiple entities, its original content will not be accessed by unauthorized parties.

(1) Key generation: Firstly, each participant uses a key generation algorithm to generate a pair of public and private keys (p_{ub}, p_{rv}). This pair of keys serves as the foundation for subsequent encryption and decryption operations, ensuring security during data transmission and computation.

(2) Input encryption: Next, the participants process their input data using homomorphic encryption technology to generate encrypted ciphertext. Homomorphic encryption is a special encryption method that allows computation of encrypted data without the need for decryption first. In this way, even if the data is intercepted during transmission, the attacker cannot directly obtain the original information.

(3) Cloud processing: The encrypted data is sent to cloud servers or other computing platforms. After receiving these ciphertexts, the cloud server processes them using a secure multi-party computing protocol. These protocols ensure that data remains encrypted even during the computation process, thus protecting the privacy of the data.

(4) Result calculation and return: After completing the calculation based on the secure multi-party computing protocol, the cloud server performs homomorphic computing on the results to obtain the final calculation result $Eval(y') \rightarrow (y'_1, y'_2, \ldots, y'_n)$. These results are still encrypted, ensuring the confidentiality of the data. Subsequently, the cloud server returns these encrypted results to the corresponding participants.

ISSN: 2710-0588

(5) Result decryption and verification:After receiving the encrypted calculation result, the participating party decrypts it using their own private key to obtain the final calculation result. Due to the fact that the data remains encrypted throughout the entire process, any unauthorized third party is unable to access the original data or intermediate calculation results, ensuring the privacy protection of transaction data.

3.2.3. Montgomery algorithm

The Montgomery algorithm is a commonly used modular reduction algorithm in large number operations, especially in modular exponentiation and modular multiplication operations. In the process of generating system parameters, we use the Montgomery algorithm to accelerate modular exponentiation, especially when calculating expressions in the form of $g^x \bmod n$. The Montgomery algorithm converts large multiplication into smaller multiplication by selecting a cardinality R that is coprime with the modulus n (usually $R=2^k$, where k is the bit length of the modulus n), and achieves modular reduction through a series of shift and subtraction operations, significantly improving computational efficiency.

3.2.4. Karatsuba algorithm

The Karatsuba algorithm is an efficient multiplication algorithm used to calculate the product of large numbers. In the process of generating system parameters in blockchain systems, especially when it is necessary to calculate the product n of two large prime numbers p and q, the Karatsuba algorithm can significantly reduce the number of multiplication operations required. Traditional multiplication requires $O(n^2)$ operations, while the Karatsuba algorithm reduces the number of multiplications to $O(n^{\log_2 3})$, which is approximately $O(n^{1.585})$, through clever splitting and combination, greatly improving computational efficiency.

In the decryption process, if modular exponentiation is involved (such as calculating $v^x \bmod n$), the Karatsuba algorithm can be used to accelerate this process. The Karatsuba algorithm improves the efficiency of modular exponentiation by reducing the number of multiplications. Specifically, assuming $x=2y+1$ (similar processing can be applied for even numbered cases), then $v^x \bmod n=(v^y * v^y * v) \bmod n$. When calculating $v^y * v^y$, use the Karatsuba method to reduce the number of multiplications, that is, first calculate $a=v^y$, then $b=(a+a) \bmod n$, $c=a*(a+1) \bmod n$, and finally $v^y * v^y=(b^2-2c+n) \bmod n$.

4. Implementation of the Scheme

4.1. Algorithm design

This section describes and implements the Homomorphic Encryption Based on Paillier and Zero Knowledge Proof (HEPZP) algorithm proposed by [5] Li Gongliang et al., and improves its design in terms of key generation efficiency, ciphertext verification, and data collaboration to better apply to transaction data services on blockchain. The parameters involved in the algorithm formula are shown in Figure 3.

Symbol	Symbol Definition
P	System Parameters
p_{rv}	User's Private Key
p_{ub}	User's Public Key
m	Plaintext Information
c	Ciphertext Information
Enc	Encryption Function

Figure 3. Related parameters

ISSN: 2710-0588

4.1.1. System parameter generation

The blockchain system has an independent set of system parameters that cannot be tampered with after the blockchain system is initialized and stored in the ledger, ensuring that they remain unchanged throughout the entire system lifecycle. System parameters play a crucial role in the transaction process, providing functions such as public and private key generation, transaction ciphertext generation, and zero knowledge evidence generation for both parties. In order to optimize these key operations, we introduced the Montgomery Reduction and Karatsuba algorithms in the system parameter generation process to improve the efficiency of large number operations.

The specific steps are as follows:

(1) Select two larger prime numbers p and q:Efficiently calculate the product n of the two using the Karatsuba algorithm.

(2) Calculate the minimum common multiple λ of (p-1, q-1):During this process, the Karatsuba algorithm can also be used to accelerate related large number multiplication operations.

(3) Choose the element g with a high order:Select the element g that satisfies the condition in $Z_{n^2}^*$, ensuring that $g^{\lambda} \equiv 1 (\mathrm{mod} n)$.

(4) Calculate system parameters:Accelerate modular exponentiation using the Montgomery algorithm, calculate $h=g^{2r} \mathrm{mod} n$, where r is a random number, ensuring coprime with n.

(5) Specify a random number r and calculate the user's public key parameters:Using the Karatsuba algorithm again to accelerate multiplication operations, calculate $k=g^{r} \mathrm{mod} n$ as one of the user's public key parameters.

(6) Encapsulate system parameters:Take (n, g, h) as the system parameter P, ensuring that these parameters remain unchanged and tamper proof throughout the entire lifecycle of the blockchain system.

4.1.2. Public and private key generation

The user first selects a smaller random integer x, and their private key is directly composed of the random integer x, i.e. $p_{rv}=x$. The calculation of public keys involves large number multiplication, and traditional methods may be inefficient. Here, we introduce the Karatsuba algorithm to accelerate this process. Specifically, the public key p_{ub} can be obtained by calculating $g^{x} \mathrm{mod} n^2$, where g is the group generator in the system parameters and n is the product of two large prime numbers. The Karatsuba algorithm can efficiently complete the calculation of g^{x}, and then modulo n to obtain the public key p_{ub}.

4.1.3. Encryption process

When both parties need to complete the transaction, the application layer will encrypt the data and information in the transaction. In the encryption process, in addition to using the system parameter P and public key k, we can also consider using the Karatsuba algorithm to optimize multiplication operations, especially when there is a large amount of data that needs to be encrypted.

The specific encryption process is as follows: for plaintext m, select two random numbers r_1, r_2, where r_1, r_2 are both less than n, and use the system parameters P, g, h, n, and public key p_{ub} to generate four ciphertexts c_1, c_2, c_3, E through the encryption function Enc (m, r_1, r_2, p_{ub}, P). This encryption process is closely related to the production of zero knowledge proof evidence, where the encryption function can be expressed as $c=g^{r} \mathrm{mod} n^2$. During the calculation process, fully utilize the Karatsuba algorithm to accelerate multiplication operations, ensuring that the encryption process is both secure and efficient.

ISSN: 2710-0588

4.1.4. Decryption process

For ciphertext c, it is necessary to decrypt the plaintext data m and the evidence used to create equality for the chain code end.

Decrypting m: Using the system parameters P (n, h) and private key p_{rv}, as well as (E, c_1) in the ciphertext, first extract the encryption result corresponding to the plaintext data from the ciphertext $c_m = c_1^{-v}E = g_1^{m\lambda} \mathrm{mod} n^2$, and then calculate the original plaintext $m = \big(L(c_m)/ L(k)\big) \mathrm{mod} n$, where $L(x) = (x-1)/n$.

Decryption: Using the system parameters P (n, h) and private key p_{rv}, as well as (c_2, c_3) in the ciphertext, extract the encrypted data corresponding to the random number from the ciphertext $c_{r_1} = c_2^{-v} c_3 = g_1^{m\lambda} \bmod n^2$, and calculate the original data $r_1 = (L(c_{r_1})/ L(h)) \mathrm{mod} n$.

4.2. Interactive zero knowledge proof verification

4.2.1. Proof of equality

(1) Initialization and random number selection

The application side requires both parties to select random numbers r_1, r_2 and jointly generate ciphertext c_t for the transaction amount. Here, the ciphertext generation process may involve encrypting the transaction amount t, and the specific encryption algorithm is designed according to the system security requirements.

(2) Commitment generation (accelerated using Montgomery algorithm)

The application side efficiently calculates modular exponentiation using the Montgomery algorithm based on ciphertext c_t and random numbers r_1, r_2, generating commitments E and F for both parties in the transaction. The Montgomery algorithm is particularly suitable for multiplication of large numbers, which can effectively reduce the size of intermediate results in the calculation process and improve computational efficiency.

$$E = g^a \cdot h^{b \cdot r_1} \cdot k^{t \cdot r_1} \bmod p$$

$$F = g^a \cdot h^{b \cdot r_2} \cdot k^{t \cdot r_2} \bmod p$$

Among them, a and b are system parameters, g, h, and k are publicly selected group generators, and p is a large prime modulus.

(3) Interactive equality evidence generation

Both parties exchange necessary intermediate calculation results through an interactive protocol, and use the Karatsuba algorithm to quickly verify whether E and F satisfy a specific equation relationship, thereby proving the equality of transaction amounts without disclosing the specific amount t.

(4) Smart contract verification

The smart contract receives commitments E, F, and equality evidence from both parties, combined with their public keys and system parameters, using the same Montgomery algorithm and Karatsuba algorithm logic for verification. The verification process ensures that E and F are indeed generated by the same transaction amount t and different random numbers r_1, r_2 without decrypting the transaction amount t, thus completing the proof of equality.

4.2.2. Interactive proof process

Prover preparation: Randomly select m', r_1', r_2' and calculate $c' = \mathrm{Enc}(m', r_1', r_2')$(using Paillier encryption).

Verifier challenge: Generate challenge variable e and send it to the verifier.

Prover response: Based on challenge e, calculate and send evidence z_1,z_2,z_3, where:

$z_1 = m' + e*t$

$z_2 = r_1' + e*r_1$

$z_3 = r_2' + e*r_2$

ISSN: 2710-0588

Verifier verification: The verifier uses the following formula $g^{z_1} \cdot h^{z_2} \cdot k^{z_3} \bmod p = E \cdot (g^t)^e$ to check the validity of the evidence.

If the equation holds, the verifier accepts the proof; If the equation does not hold, reject it.

The commitments and proofs given by both parties in the transaction are verified by each node of the blockchain. Only when each generated proof has been verified by a smart contract can this transaction be validated. After the transaction is executed through the consensus scheme of the blockchain, it can be written into the block.

4.3. Design of data on chain trading scheme

In the design of data on chain transactions, we have introduced an interactive zero knowledge proof mechanism to ensure the security, privacy, and verification efficiency of the transaction process. This transaction process involves close collaboration between the application layer and the chain code end, and the specific design is as follows:

4.3.1. Transaction initialization and key preparation

Consumers first generate and hold their public and private key pairs for encrypting and decrypting transaction information. Producers provide their public key and system parameter P for transaction encryption. Then using the public keys p_{ub} of both parties, encrypt the transaction input to generate encrypted transaction input ciphertext, ensuring the security of data during transmission.

4.3.2. Transaction decryption and balance verification

Consumers use their private key p_{rv} and system parameter P to decrypt encrypted transaction input information and calculate transaction balance based on it, ensuring the accuracy of transaction amount. Based on the decrypted transaction information, use interactive zero knowledge proof algorithms to generate transaction evidence that is both trustworthy and does not leak sensitive information, including verification of the range of transaction amounts and proof of equality between input and output amounts.

4.3.3. Interactive zero knowledge proof process

Consumers and producers (or verification nodes) gradually generate and exchange necessary zero knowledge proof evidence and commitments through interaction. These pieces of evidence can verify the validity of the transaction while protecting the privacy of both parties.

The verification node (or chain code end) utilizes the received zero knowledge evidence and commitment, combined with the public key and system parameters of both parties, to perform an interactive verification process, ensuring that the transaction complies with preset rules and conditions.

4.3.4. Transaction data uploading and signature confirmation

Transaction data is digitally signed by both parties before being sent to other nodes in the blockchain network, ensuring the integrity and immutability of the data. Send transaction data containing ciphertext, zero knowledge evidence, and related commitments to the chain code end. The chain code end performs the final transaction verification, including verifying the correctness of digital signatures, zero knowledge proofs, and the compliance of transaction logic. Once the transaction passes all verifications, the chain code end will perform the transaction on chain operation, record the transaction data into the blockchain, and complete the entire transaction process.

4.4. Data Collaborative Design

In the process of data collaborative design, in order to ensure the privacy, integrity, and traceability of data, we have designed a solution based on blockchain and zero knowledge proof technology. This scheme directly processes encrypted data and related evidence on the

ISSN: 2710-0588

blockchain, and achieves secure uploading, verification, and collaborative management of data through smart contracts.

4.4.1. Data Encryption and Zero Knowledge Proof

Before uploading data onto the chain, sensitive data is first encrypted using homomorphic encryption algorithm (Paillier algorithm). Homomorphic encryption allows for specific mathematical operations, such as addition or multiplication, to be performed on encrypted data while maintaining its encrypted state, thereby protecting the privacy of the data. In order to verify the correctness of encrypted data and meet specific conditions (such as equality, range, etc.), we use zero knowledge proof technology to generate corresponding evidence. These pieces of evidence can prove that the data meets specific conditions without compromising the original data.

4.4.2. Data on chain and smart contracts

The encrypted data and its zero knowledge proof evidence are directly uploaded to the blockchain. This process is managed through smart contracts to ensure the integrity and immutability of data. We design a dedicated smart contract to handle data upload, verification, and collaborative operations. Smart contracts include the following main functions:

Data upload: Receive encrypted data and zero knowledge proof evidence uploaded by users, and verify their validity.

Data verification: Verify the correctness of encrypted data and whether it meets specific business rules (such as data range, equality, etc.) based on zero knowledge proof evidence.

Data access control: Based on the access control logic in smart contracts, ensure that only authorized users can access specific encrypted data.

Collaborative operation: Support multiple users to collaborate on data processing on the blockchain, such as data aggregation, analysis, etc., while maintaining data privacy and security.

4.4.3. Interaction process

Users will perform homomorphic encryption on the data that needs to be shared or collaboratively processed, and generate corresponding zero knowledge proof evidence. Then the user sends the encrypted data and zero knowledge proof evidence to the blockchain network, and the smart contract receives the data for verification. If the verification is successful, the data will be stored on the blockchain and the corresponding status information will be updated. When collaborative data processing is required, relevant users can submit requests through smart contracts and perform corresponding collaborative operations on the blockchain. All operations are carried out on the basis of encrypted data, ensuring the privacy and security of the data. After collaborative processing is completed, the smart contract will return the result to the requester in encrypted form. The requester can use their own private key to decrypt the result and perform subsequent processing as needed.

5. Safety Analysis

5.1. Privacy protection

This scheme adopts interactive zero knowledge proof technology, allowing us to verify the validity or correctness of data without leaking any sensitive data content. During the transaction process, all participants can prove to the verifier through a zero knowledge proof protocol that they possess certain information (such as transaction amount, identity information, etc.) without directly exposing the information itself. This mechanism effectively protects the privacy of both parties in the transaction and prevents the leakage of sensitive information. By combining the Montgomery algorithm and Karatsuba algorithm, we have efficiently encrypted the transmitted and stored data. The Montgomery algorithm is widely

ISSN: 2710-0588

used in the field of cryptography for its efficient modular multiplication, while the Karatsuba algorithm accelerates large number multiplication by reducing the number of multiplication operations. The combination of the two can significantly improve the efficiency and security of data encryption. Through these encryption techniques, all data stored on the blockchain exists in ciphertext form, ensuring the privacy of the data.

5.2. Data integrity and authenticity

This scheme ensures the authenticity of the data through the characteristics of the algorithm itself. The Montgomery algorithm and Karatsuba algorithm, as efficient number theory algorithms, maintain the consistency and integrity of data during encryption and decryption processes. In addition, the process of interactive zero knowledge proof itself involves verifying the authenticity of the data, as the verifier must provide sufficient information to convince the verifier that a fact is true, and this process is open, transparent, and difficult to forge. Therefore, even without explicit hash values and signatures, this scheme can still ensure the authenticity of the data.

5.3. Non repudiation

Blockchain technology has the characteristics of traceability and immutability. In the security scheme proposed in this article, all transaction information generated by users participating in the transaction is equally stored on the blockchain. This enables any transaction to be traced back to its origin and the entire circulation process, thus achieving non repudiation functionality. Even if one of the trading parties attempts to deny their transaction behavior, their authenticity can be proven through records on the blockchain. By combining zero knowledge proof and blockchain technology, we can build a more secure trading environment. Zero knowledge proof ensures that both parties complete transaction verification without exposing sensitive information; And blockchain records the history of all transactions, making any transaction non repudiation. This combination of applications significantly improves the security and reliability of the system.

6. Experimental Analysis

To verify the performance of the homomorphic encryption algorithm based on homomorphic encryption and interactive zero knowledge proof proposed in the paper, this chapter adopts

The Fabric project in the alliance chain implements this algorithm. After completing the environment construction of the consortium chain system based on Fabric 2.2 version, this chapter tested the implementation effect of the algorithm to verify its feasibility and test its implementation efficiency.

The simulation implementation experiment in this article was conducted in a virtual machine, which is VMware. The machine is equipped with two processor cores and 4GB of memory. Using the Ubuntu 20.04 LTS operating system in the virtual machine, the required environment configuration includes services such as go1.17, docker20.10, and docker composite 1.29.2. The smart contract involved in the zero knowledge proof privacy protection scheme in this article is written in Go language.

6.1. Function test

The core of the efficiency verification process includes three key aspects: confirming the effectiveness of data encryption, verifying the interactive zero knowledge verification mechanism, and verifying data integrity. In the verification process, we first instantiate the chain code to simulate the operation of user A storing an encrypted amount of 50 in the blockchain ledger, and then initiate a transaction to transfer 10 to user B. After the transaction is successfully executed, ensure that user A's balance is updated to 40 in encrypted form, and

ISSN: 2710-0588

at the same time, add an encrypted transfer amount of 10 to user B's account. Next, by submitting the generated encrypted information and supporting zero knowledge proof evidence to the chain code end, using the container logs in the node for review, it is confirmed that all transaction information is in an encrypted state, and all interactive zero knowledge proof verifications have passed. Finally, the signal of successful transaction is fed back to the application layer. During the performance verification process, we further verified that the chain code end can still effectively verify the correctness of transaction results while maintaining encrypted transaction data, and securely store the encrypted transaction results in the ledger. In addition, when the application attempts to submit malicious transactions, such as when the output amount exceeds the input amount, although corresponding encrypted information and zero knowledge evidence can be generated, the chain code end can intelligently identify such violations and refuse to write the transaction into the ledger because it cannot complete verification based on invalid evidence. This series of tests not only demonstrates the effectiveness of this algorithm in ensuring the confidentiality of information between transaction parties and preventing privacy leaks, but also demonstrates its powerful ability in identifying and preventing malicious transactions.

6.2. Efficiency testing

Efficiency testing refers to testing the key generation, encryption, and decryption processes of algorithms. The stability of algorithms is mainly based on the calculation of discrete logarithms. When the length of the key used is large enough, according to cryptographic analysis theory, the algorithm can be ensured not to be easily attacked and cracked. This article tested the key length n in the system parameters with lengths of 1 024, 2 048, and 4 096 bits. The experimental results are shown in Figure 4, Figures 5, and Figures 6.

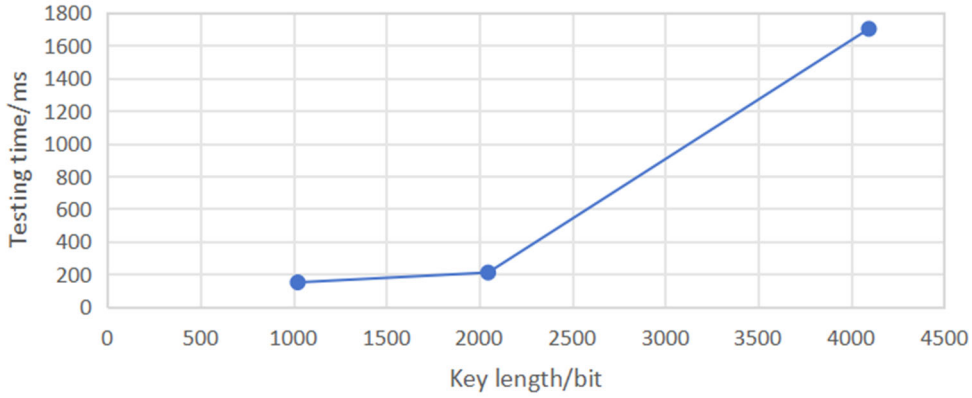

Figure 4. Key generation time for different key lengths

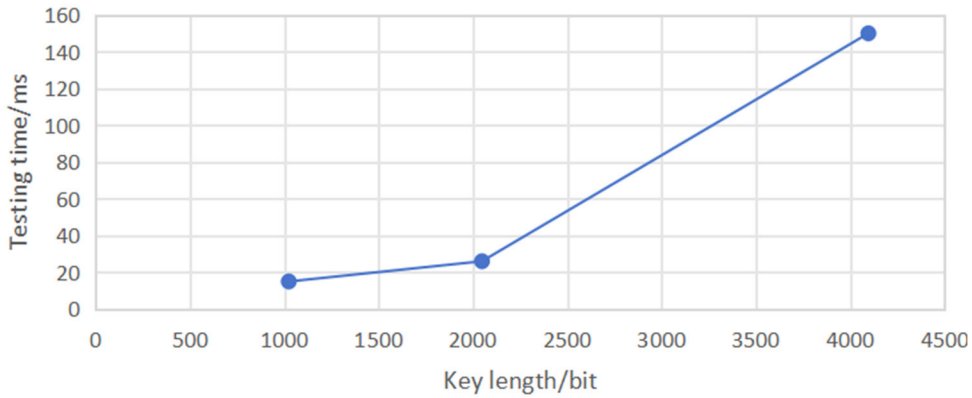

Figure 5. Encryption time for different key lengths

ISSN: 2710-0588

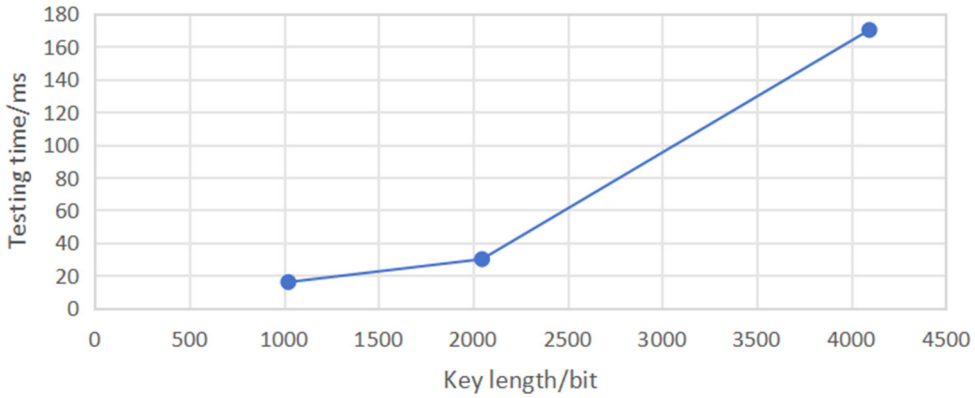

Figure 6. Zero knowledge ciphertext verification time for different key lengths

The experimental results show that the Montgomery algorithm and Karatsuba algorithm effectively improve the efficiency of key generation and encryption and decryption, and the use of interactive zero knowledge proof has almost no effect on the efficiency of the algorithm.

References

[1] Li Yichong, Zhou Kuanju, and Wang Zizhong Research on Blockchain Privacy Protection Based on Zero Knowledge Proof [J] Space Control Technology and Applications, 2022, 48 (1): 44-52.

[2] Dong Guishan, Chen Yuxiang, Fan Jia, etc Research on Privacy Protection Strategies in Blockchain Applications [J] Computer Science, 2019, 46 (5): 29-35.

[3] QIN Jie, MA Zhaofeng, DUAN Pengfei, et al.Privacy Protection Scheme of Transcation Data Supporting Zero-Knowledge Proof [J]. Information Security and Communications Privacy, 2022(10): 38-51.

[4] MA Z F,WANG X C,JAIN D K,et al.A Blockchain-Based Trusted Data Management Scheme in Edge Computing[J]. IEEE Transactions on Industrial Informatics, 2020,16(3):2013-2021.

[5] Li Gongliang, He Dongbo, Guo Bing, etc.A Blockchain Privacy Protection Algorithm Based on Zero Knowledge Proof [J] Journal of Huazhong University of Science and Technology (Natural Science Edition), 2020,48 (7): 112-116.