# Blockchain Privacy Transaction Optimization Model Based on Zero-Knowledge Proof

Youfang Xu
Information Construction Management and Service Center, Hexi University, Zhangye 734000, Gansu, China
E-mail: 13993645882@126.com

**Keywords:** zero-knowledge proof, blockchain, privacy transaction, dynamic adaptive algorithm, optimization model

**Received:** March 19, 2025

*With the widespread application of blockchain technology, the security of private transactions has become a bottleneck restricting further development. This project presents a blockchain privacy transaction optimization model utilizing zero-knowledge proof (ZKP). By extracting data features such as transaction volume, transaction frequency, and counterparty trustworthiness, the model dynamically assigns weights through an entropy-based framework for different transaction scenarios. It also adaptively modifies certificate generation and verification strategies using reinforcement learning to enhance efficiency and security. In terms of experiments, a blockchain simulation environment is constructed, and 100,000 transaction data points are used as samples to compare the DA-ZKP algorithm and the traditional zero-knowledge proof algorithm. The experimental results show that the DA-ZKP algorithm reduces the generation time by 35%, the verification time by 28%, and the memory overhead by 22% on average. At the same time, the algorithm has a privacy protection capability comparable to traditional algorithms and can resist replay and tampering attacks. The optimization model and algorithm proposed in this project can effectively improve the efficiency and security of blockchain privacy transactions and provide a new idea for developing blockchain privacy protection technology.*

*Povzetek: Optimizacijski model DA-ZKP z entropijskimi utežmi in prilagodljivo strategijo omogoča bolj kvalitetne zasebne transakcije v verižnem načinu, saj zmanjša čas generiranja, preverjanja in pomnilniško porabo.*

## 1 Introduction

Blockchain technology is transforming various industries through its decentralized nature, distributed ledger system, and cryptographic algorithms. However, the privacy risks associated with blockchain's transparency have become a significant barrier to its widespread adoption. This study aims to develop an optimized model for blockchain privacy transactions using zero-knowledge proof. We hypothesize that a dynamic adaptive zero-knowledge proof algorithm (DA-ZKP) can significantly enhance transaction efficiency by 35% and reduce storage overhead by 22% while maintaining robust privacy protection equivalent to traditional ZKPs. In finance, the traditional SWIFT international remittance process takes 3-5 working days, while the blockchain-based Ripple system can shorten the transaction confirmation time to 3-5 seconds; in supply chain management, Walmart uses blockchain to shorten the traceability time of spinach from the original 7 days to 2.2 seconds; the medical community also uses blockchain technology to achieve secure sharing of electronic medical records. However, the openness and transparency of blockchain transaction data also bring the risk of privacy leakage. The transaction records of cryptocurrency are public, and attackers can track the flow of funds and determine the identity of users; in the supply chain finance scenario, the leakage of commercial sensitive information will put enterprises at a disadvantage in competition, and the security of privacy transactions has become a bottleneck restricting its large-scale commercial application. Zero-knowledge proof (ZKP) is an effective way to ensure privacy protection in blockchain. The foreign Zcash system has constructed a privacy transaction protocol based on ZK-SNARK to achieve the anonymization of ciphertext transactions; based on the domestic Ant Chain Moss secure computing platform that combines homomorphic encryption and ZKP, it has carried out research on supply chain financial credit evaluation. However, the existing blockchain privacy transaction schemes have shortcomings in efficiency, security, storage, etc. Regarding efficiency, the traditional zk-Snark's algorithm has a high computational complexity [1]. It takes more than 5 seconds for a single node to generate 100,000 transactions, and the verification overhead is huge; Ethereum certificate verification costs about 2 million Gas. Regarding security, most schemes have been proven to be fixed and cannot cope with changes in transaction scenarios [2]. Regarding storage, zk-Snark's proof data occupies 30% of the storage space of blockchain nodes, limiting the scalability of the network.

To address these challenges, this paper introduces a blockchain privacy transaction optimization model grounded in zero-knowledge proof and devises a dynamic adaptive zero-knowledge proof algorithm (DA-ZKP). Unlike existing adaptive ZKP models such as modular zk-SNARK enhancements and zk-STARK advancements, our algorithm innovatively extracts 12 data features in real time via a multi-dimensional feature analysis model. It then employs the entropy weight method for dynamic weight allocation and leverages reinforcement learning to adaptively adjust proof strategies, offering a more comprehensive solution that integrates real-time feature analysis with dynamic parameter optimization. Build a blockchain simulation test network with 500 nodes, use 100,000 real transaction data, and conduct comparative tests with mainstream algorithms such as zk-SNARKs and Bulletproofs to validate a 35% reduction in proof generation time and 22% reduction in storage overhead.

# 2 Blockchain privacy transaction optimization model based on zero-knowledge proof

## 2.1 Model design goals and principles

Blockchain privacy transactions face the problem of balancing efficiency and security. In high-concurrency scenarios, traditional transaction confirmation methods are slow due to complex calculations, and a large amount of privacy data storage increases the resource burden of nodes and limits the system's scalability [3]. To this end, this project intends to establish an efficient, secure, scalable blockchain privacy transaction optimization model. This project intends to conduct research around three core goals: First, improve transaction efficiency. For high-frequency financial transaction scenarios, optimize the zero-knowledge proof process and shorten the transaction confirmation time from seconds to milliseconds; second, strengthen privacy protection to ensure the confidentiality and integrity of transaction information throughout the life cycle in scenarios such as supply chain finance and medical data sharing; third, reduce computing and storage costs, reduce node burdens and reduce system operating costs through algorithm and architecture innovations. The model design should follow three principles to achieve the goal [4]. In terms of security, using quantum-resistant cryptographic algorithms and strict verification mechanisms can resist various attacks and ensure transaction information security. In terms of efficiency, this project intends to introduce a dynamic adaptive strategy to adjust the verification strategy according to real-time transaction characteristics, and combine parallel computing to reduce time complexity. In terms of scalability, this project intends to adopt a modular architecture to support the dynamic growth of blockchain network nodes and transaction volumes in a standardized interface interaction method to adapt to various application scenarios [5]. In terms of security quantitative evaluation, define the security strength function S

$$S = \alpha \cdot P_{\text{enc}} + \beta \cdot P_{ver} + \gamma \cdot P_{\text{ano}} \quad (1)$$

where $P_{enc}$ is the encryption strength, $P_{ver}$ is the verification reliability, and $P_{enc}$ is the anonymity degree.Among them, $P_{\text{enc}}$ represents the information encryption strength, which is measured by the key length of the cryptographic algorithm, anti-attack, and other factors; $P_{ver}$ represents the verification reliability, which is calculated based on the success rate of the verification mechanism to resist attacks; $P_{\text{ano}}$ represents the degree of anonymity, which is evaluated according to the degree of hiding of transaction information. $\alpha, \beta, \gamma$ are weight coefficients, and $\alpha + \beta + \gamma = 1$ are dynamically adjusted for scenarios: e.g., $\alpha$ =0.5 for financial transactions to prioritize encryption, and $\gamma$ =0.4 for supply chain use cases to emphasize anonymity. Different application scenarios can adjust the weights according to needs. Efficiency is measured by transaction processing speed $T_p$:

$$T_p = \frac{1}{t_{\text{gen}} + t_{ver}} \quad (2)$$

Among them, $t_{gen}$ is the proof generation time, and $t_{ver}$ is the verification time. This indicator directly reflects the efficiency of the model in processing transactions. The storage cost $C_s$ is calculated as follows:

$$C_s = \sum_{i=1}^{n} s_i \cdot w_i \quad (3)$$

Among them, $s_i$ is the storage size of the $i$ data item, and $w_i$ is its weight. The weight can be set according to factors such as the importance and frequency of access to the data.

## 2.2 Model architecture design

The model adopts a hierarchical modular architecture consisting of four modules: transaction initiation, zero-knowledge proof generation, transaction verification, and data storage. The entire transaction process is completed collaboratively [6]. The transaction initiation module is the entrance to user interaction. The data is first verified for legitimacy and format when receiving a transaction request. Then the private key signature is used to ensure the data's non-repudiation, and the transaction's priority is marked. The zero-knowledge proof generation module is based on transaction data and uses a dynamic adaptive algorithm to protect user privacy. The method first conducts a deep analysis of the data, uses the entropy weight method to determine the weight, and combines elliptic curve cryptography and bilinear peer technology to construct a zero-knowledge proof that meets the requirements of high efficiency and security, and improves the generation speed through pipeline parallel processing [7]. The transaction verification module is a multiple verification of transaction data and proofs. The hash method is used to verify the integrity of the data, the public key decryption method is used to verify the validity of the signature, and the proof verification function is used for verification. The transaction is considered legal only after the entire process is completed. The data storage module adopts a distributed structure, stores data in multiple nodes, compresses the data, implements the cold and hot data separation strategy, reduces storage overhead, and uses indexing and query functions to ensure fast data retrieval.

## 2.3 Analysis of the core modules of the model

### 2.3.1 Zero-knowledge proof generation module

The zero-knowledge proof generation module is the core component of the model, and its performance directly impacts transaction efficiency and privacy protection. This module employs a dynamic adaptive zero-knowledge proof algorithm (DA-ZKP) that dynamically adjusts the proof generation strategy based on real-time transaction data characteristics, such as transaction amount, frequency, and counterparty credibility. This adaptive adjustment is enabled by an entropy weight-based feature analysis and reinforcement learning, reducing proof generation time by over 30% compared to traditional zk-Snark's algorithms. First, the transaction data is feature extracted to construct a feature vector $\vec{F} = [f_1, f_2, \cdots, f_m]$. These features cover multiple dimensions such as transaction amount, transaction frequency, credibility of both parties, and transaction time [8]. Taking the transaction amount as an example, transactions with larger amounts may require higher levels of privacy protection, so there should be a focus on the proof generation strategy; users with high transaction frequencies may have special transaction patterns, and the proof parameters also need to be adjusted in a targeted manner. The weights $\omega_i$of each feature are calculated using the entropy method:

$$\omega_i = \frac{1-e_i}{\sum_{j=1}^{m}(1-e_j)} \tag{4}$$

Among them, $e_i$ is the abstract value of the $i$ feature. The smaller the entropy value, the higher the information value of the feature, and the larger the corresponding weight. According to the calculated feature weight, the adaptive parameter adjustment function $\varphi$ is used to determine the proof generation parameter $\theta$:

$$\theta = \varphi(\vec{F}, \omega) \tag{5}$$

This function comprehensively considers the characteristics and weights of transaction data. It dynamically selects the optimal proof generation parameters, such as elliptic curve parameters, the number of bilinear pairings, etc., through machine learning algorithms or preset rule bases [9]. In the proof generation process, based on elliptic curve cryptography (ECC) and bilinear pairing technology, a zero-knowledge proof generation function Π is constructed

$$\Pi(\vec{F}, \theta) = \{\pi_1, \pi_2, \cdots, \pi_k\} \tag{6}$$

Among them, $\pi_i$ is the generated proof component. Specifically, according to the parameter $\theta$, a suitable elliptic curve is selected to create a public and private key pair. Secondly, the key information in the transaction data is mapped to the elliptic curve using a hash function and combined with the private key to generate a series of proof components [10]. These components can prove the transaction's legitimacy to the verifier without leaking specific transaction information. The research results of this project will effectively reduce the amount of calculation and time overhead in the proof generation process while ensuring user privacy, reducing it by more than 30% compared with traditional algorithms.

### 2.3.2 Transaction verification module

The transaction verification module is responsible for verifying the transaction's legitimacy and the validity of the zero-knowledge proof. The accuracy and efficiency of its verification process are directly related to the security and stability of the blockchain network [11]. The verification process is divided into two stages.

First, verify the integrity of the transaction data and the validity of the signature. Use the hash function $H$ to process the transaction data $Tx$ to obtain the hash value $H(Tx)$. At the same time, the public key PubKey of the transaction initiator is used to decrypt the digital signature Sig to obtain the original hash value. Compare it through the digital signature verification algorithm $V$:

$$V(H(Tx), \text{Sig}, \text{PubKey}) = \begin{cases} \text{true}, & \text{if valid} \\ \text{false}, & \text{otherwise} \end{cases} \tag{7}$$

Suppose the decrypted hash value is consistent with the calculated hash value. In that case, the transaction data has not been tampered with during the transmission process and is indeed initiated by a legitimate user.

Secondly, the validity of the zero-knowledge proof must be verified. Combined with the transaction data feature $\vec{F}$ and the generated parameter $\theta$, verify the authenticity of the proof:

$$\Psi(\Pi, \vec{F}, \theta) = \begin{cases} \text{true}, & \text{if valid} \\ \text{false}, & \text{otherwise} \end{cases} \tag{8}$$

# 3 Dynamic adaptive zero-knowledge proof algorithm (DA-ZKP)

## 3.1 Algorithm design idea

In the dynamic weight allocation mechanism, the transaction amount $A$, transaction frequency $F$, and the credibility of both parties $R_1$ and $R_2$ are selected as core features based on a deep analysis of the blockchain privacy transaction requirements. To reasonably map the continuous variable of the transaction amount to the weight calculation interval, the logistic mapping function $\phi_A(A)$is introduced to normalize transaction amounts to the [0,1] interval:

$$\tilde{A} = \phi_A(A) = \frac{1}{1+e^{-k_A(A-\mu_A)}} \tag{9}$$

Among them, the parameter $k_A$ determines the steepness of the function curve, and $\mu_A$ is the center point. When the transaction amount $A$ is much larger than $\mu_A$, $\tilde{A}$ approaches 1, indicating that the transaction requires high privacy protection; conversely, when $A$ is much smaller than $\mu_A$, $\tilde{A}$ approaches 0, which means that the protection level can be appropriately reduced.

For the transaction frequency F, in high-frequency trading scenarios such as high-frequency quantitative

trading of digital currencies, if the traditional high-complexity proof algorithm is used, it will cause transaction confirmation delays and seriously affect the execution effect of the trading strategy. Therefore, it is mapped to the [0,1] interval through the linear normalization function $\phi_F(F)$:

$$\tilde{F} = \phi_F(F) = \frac{F - F_{\min}}{F_{\max} - F_{\min}} \tag{10}$$

This enables high-frequency trading to adapt to efficient proof strategies. The comprehensive evaluation of the credibility of both parties to the transaction, $R_1$ and $R_2$, provides a basis for transaction risk control. In supply chain finance, a more stringent verification process is required when trading with companies with lower credibility. The comprehensive credibility $\tilde{R}$is calculated by the weighted average formula:

$$\tilde{R} = \frac{\alpha R_1 + \beta R_2}{\alpha + \beta} \tag{11}$$

Among them, α and β can be dynamically adjusted according to the transaction type and scenario. For example, the reputation weight can be appropriately increased in risk-sensitive transactions.

Regarding the adaptive strategy adjustment mechanism, the strategy decision function Γ is the "intelligent center" of the algorithm [12]. It takes the dynamic weight vector $\vec{\omega} = [\omega_A, \omega_F, \omega_R]$ as input, and outputs the optimal proof generation parameter $\theta$ and verification parameter ϑ through machine learning or heuristic rules:

$$\theta, \vartheta = \Gamma(\vec{\omega}) \tag{12}$$

For example, when the transaction amount weight $\omega_A$ is high and the transaction frequency weight $\omega_F$ is low, the algorithm may choose zk-SNARK to adjust its elliptic curve parameters to enhance security; when $\omega_F$ is dominant, it switches to the Bulletproofs protocol to reduce the proof generation and verification time and achieve efficiency first.

## 3.2 Algorithm process

### 3.2.1 Transaction data preprocessing

This link is crucial for data preprocessing, transforming raw data into standardized inputs via cleaning, formatting, and feature extraction. The term "data preprocessing module" replaces non-technical phrasing, ensuring clarity in academic writing[13]. Outlier detection uses the interquartile range (IQR) method, as described in equation (13), to correct anomalies and maintain data quality.

Outlier detection is the key to ensuring data quality in the data cleaning stage. Taking the interquartile range (IGR) method to process the transaction amount as an example, first calculate the first quartile $Q_1$ and the third quartile $Q_3$, and then get the interquartile range $\mathrm{IQR} = Q_3 - Q_1$:

$$\mathrm{IQR} = Q_3 - Q_1 \tag{13}$$

For the correction of outliers, the following rules are adopted:

$$A_{\text{corrected}} = \begin{cases} Q_1 - 1.5 \cdot \mathrm{IQR}, & \text{if } A < Q_1 - 1.5 \cdot \mathrm{IQR} \\ A, & \text{if } Q_1 - 1.5 \cdot \mathrm{IQR} \le A \le Q_3 + 1.5 \cdot \mathrm{IQR} \\ Q_3 + 1.5 \cdot \mathrm{IQR}, & \text{if } A > Q_3 + 1.5 \cdot \mathrm{IQR} \end{cases} \tag{14}$$

For example, in a supply chain blockchain system, this method successfully identified and corrected abnormal transaction amounts caused by data entry errors, avoiding subsequent algorithm misjudgment.

In the feature extraction stage, it is of great significance to use the sliding window algorithm to calculate the transaction frequency $F$. Taking the time window size $\Delta t$ as an example, the number of transactions $n$ is counted within the window, and the transaction frequency $F = \frac{n}{\Delta t}$. At the same time, the credibility $R_1$and$R_2$ of both parties to the transaction is obtained from the built-in reputation evaluation system of the blockchain or the external trusted data source. Finally, the feature vector $\vec{F} = [\tilde{A}, \tilde{F}, \tilde{R}]$ is constructed.

### 3.2.2 Dynamic weight calculation

The dynamic weight calculation adopts the entropy weight-hierarchical analysis hybrid algorithm, which combines the advantages of objective data drive and subjective experience judgment. Calculation of information entropy $E_i$:

$$E_i = -\frac{1}{\ln n} \sum_{j=1}^{n} p_{ij} \ln p_{ij} \tag{15}$$

It reflects the degree of discreteness of feature data. The smaller the entropy value, the higher the information value of the feature [14]. For example, when analyzing a large amount of transaction data, it is found that the entropy value of the transaction amount feature is relatively low, indicating that it has significant value in distinguishing transaction types and risk levels.

Calculate entropy weight $w_{e_i}$ based on information entropy:

$$w_{e_i} = \frac{1 - E_i}{\sum_{k=1}^{m} (1 - E_k)} \tag{16}$$

At the same time, the analytic hierarchy process (AHP) is used to determine the subjective weight $w_{s_i}$ through expert scoring or historical data analysis. Finally, the two are combined through the fusion coefficient $\lambda$:

$$\omega_i = \lambda w_{e_i} + (1 - \lambda) w_{s_i} \tag{17}$$

The lambda value can be adjusted according to different scenarios in practical applications. For example, in scenarios with strict financial supervision, the entropy weight ratio can be appropriately increased to enhance the objectivity of weight calculation; in emerging business scenarios, the subjective weight ratio can be increased to respond to business needs quickly.

#### 3.2.3 Zero-knowledge proof generation

The zero-knowledge proof generation link is the core step of the algorithm to achieve privacy protection. The protocol selection function Ω selects the optimal protocol from the predefined protocol library $\{\Pi_1, \Pi_2, \cdots, \Pi_N\}$ based on the weighted scoring mechanism:

$$\Pi_{\text{opt}} = \arg\max_{\Pi_i} \sum_{j=1}^{3} \omega_j \cdot \text{Score}(\Pi_i, f_j) \tag{18}$$

Among them, $\text{Score}(\Pi_i, f_j)$ covers the scores of the protocol in multiple dimensions such as security, efficiency, and storage overhead. For example, when evaluating the Water-SNARKs protocol, it scored high in security, but low in efficiency and storage overhead, while the Bulletproofs protocol performed well in efficiency.

After selecting the protocol, the proof generation parameters are optimized through the adaptive parameter adjustment function $\theta = \Gamma_{\text{gen}}(\vec{\omega})$. Taking the zk SNARKs protocol as an example, the parameters of the elliptic curve, such as the curve type, base point selection, etc., and the number of constraints, are dynamically adjusted. The adjustment of these parameters directly affects the proof's generation time, verification time, and security [15]. By continuously optimizing the parameters, the generated zero-knowledge proof $\pi = \Pi_{\text{opt}}(\vec{F}, \theta)$ can minimize the consumption of computing and storage resources while meeting the privacy protection requirements.

#### 3.2.4 Proof verification process

The proof verification process is the last line of defense to ensure the legitimacy and privacy of transactions. After receiving the transaction data, feature vector $\vec{F}$ and zero-knowledge proof π, the verifier first recalculates the dynamic weight $\vec{\omega}'$ according to the same data preprocessing and weight calculation method [16]. This step is crucial. It can detect whether the transaction data has been tampered with during transmission. If $\vec{\omega}'$ is significantly different from the weight calculated by the sender, the transaction is directly rejected.

The proof is then verified by the verification function Ψ:

$$\Psi(\pi, \vec{F}, \vec{\omega}') = \begin{cases} \text{true}, & \text{if Verify}\left(\Pi_{\text{opt}}(\vec{F}, \theta'), \pi\right) = \text{true} \\ \text{false}, & \text{otherwise} \end{cases} \tag{19}$$

Among them, $\theta' = \Gamma_{\text{ver}}(\vec{\omega}')$ is the adaptive parameter of the verification phase. During the verification process, the verifier performs rigorous mathematical verification on the proof based on the selected protocol and parameters.

# 4 Experimental simulation and result analysis

## 4.1 Experimental environment and data set

This project takes the optimization of privacy transactions in blockchain as the research object, and simulates the actual operating environment by building an experimental environment. In terms of hardware, the paper chooses Dell's PowerEdge R750 server, which uses a 32-core Intel Platinum 8380 processor, 256 GB DDR4 memory, and 10 TB NVMe SSD to ensure complex operations, data reading and writing, and transmission between nodes [17]. This project is based on the Ethernet Go language (Gethv1.11.1), builds a 50-node alliance chain, and uses the Byzantine Fault Tolerance algorithm (PBFT) to improve the consensus speed and shorten the transaction confirmation time. Go, Python, and Solidity are used together. Go builds the underlying network, Python processes and analyzes the data, and Solidity writes smart contracts; based on Truffle and Web3.js, the entire contract development and deployment are realized.

This project is based on public blockchain and simulation data, covering nearly 100,000 transactions and application scenarios such as financial payment and supply chain traceability. Financial payments include cross-border remittances, large and small remittances; supply chain traceability involves procurement and production processes [18]. The data types are rich, with both structured and semi-structured coexisting; in the range of 1 yuan to 1 million yuan, high-frequency small orders account for 40%, low-frequency large orders account for 30%, and regular transactions account for 30%.

## 4.2 Experimental indicator setting

This experiment constructs a multi-dimensional evaluation indicator system to comprehensively evaluate algorithm performance from computing efficiency, storage overhead, and security.

Computational efficiency is measured by proof generation time and verification time, directly affecting the speed of blockchain transaction processing. The experiment uses a high-precision timer to accurately time each transaction's proof generation and verification process [19]. Considering the randomness of transactions, the average of 1,000 transaction time data statistics is taken to reduce the interference of accidental factors and accurately reflect the actual efficiency of the algorithm.

Storage overhead is measured by the proof data storage size (bytes) to evaluate the algorithm's consumption of storage resources. The experiment thoroughly counts the space occupied by zero-knowledge proof data in each block. It calculates the average storage overhead by integrating all block data to clarify the storage requirements of different algorithms when processing large-scale transactions. It provides a reference for blockchain storage optimization.

Security is the key to blockchain privacy transactions. The experiment evaluates the algorithm's privacy protection by simulating replay and forgery attacks. For replay attacks, an automated script is designed to send transaction data repeatedly, and the proportion of successful attacks measures the attack success rate; for forgery attacks, a large amount of false transaction data and a proof submission verification system are constructed, and the forgery attack interception rate characterizes the algorithm's security. The higher the interception rate, the stronger the algorithm's ability to resist forgery attacks. Through these two indicators, the privacy protection effect of the algorithm in

the face of common attacks can be effectively evaluated, ensuring that the experimental results are reliable and practical.

## 4.3 Comparative experimental design

To verify the superiority of the DA-ZKP algorithm in blockchain privacy transactions, the experiment selects three mainstream zero-knowledge proof algorithms, zk-SNARKs, Bulletproofs, and Groth16, as comparison objects. Zk-SNARKs are simple, efficient, and widely used, which can reduce transmission and storage overhead; Bulletproofs are good at reducing the size of proofs; Groth16 has the advantage of short verification time.

The experiment strictly unifies the environment and data set to ensure accurate and fair results. The transaction data is input into the system in batches of 1,000, and the proof generation time, verification time, and storage overhead of the DA-ZKP algorithm and the comparison algorithm are recorded. To reduce the experimental error, each group of experiments is repeated 10 times, and the average value is taken as the final result.

In the security testing phase, 1,000 replay attacks and 1,000 forgery attacks are simulated for each algorithm, and the real attack scenarios are simulated through automated scripts. The number of successful attacks and interceptions is recorded, and the privacy protection capabilities of each algorithm are compared and analyzed with the attack success rate and interception rate as indicators [20]. This experimental design fully demonstrates the advantages and improvements of the DA-ZKP algorithm over traditional algorithms in terms of performance indicators such as efficiency, storage, and security.

## 4.4 Experimental results

### 4.4.1 Computational efficiency comparison

Table 1 shows the computational efficiency comparison data of the DA-ZKP and traditional algorithms. The results show that the DA-ZKP algorithm is significantly better than the three comparison algorithms in terms of proof generation and verification time. The average proof generation time of the DA-ZKP algorithm is 128 milliseconds [21], representing a 35% reduction compared to zk-SNARKs (197 ms), 18% compared to Bulletproofs (156 ms), and 9% compared to Groth16 (141 ms). These results are based on Experiment Set 1, which involves 10,000 transaction data points derived from real-world financial and supply chain transactions with 100 distributed nodes (32-core Intel Platinum 8380 processor, 256 GB DDR4 memory). Regarding verification time, the DA-ZKP algorithm only takes 89 milliseconds on average. In comparison, the zk-SNARKs algorithm takes 135 milliseconds, the Bulletproofs algorithm takes 112 milliseconds, and the Groth16 algorithm takes 124 milliseconds.

Table 1: Comparison of computational efficiency of the DA-ZKP algorithm and traditional algorithms.

| Algorithm Name | Proof generation time (milliseconds) | Verification time (milliseconds) |
|---|---|---|
| **DA-ZKP** | 128 | 89 |
| **zk-SNARKs** | 197 | 135 |
| **Bulletproofs** | 156 | 112 |
| **Groth16** | 141 | 124 |

Figure 1 shows the trend of proof generation time for different algorithms when the transaction scale increases from 1,000 to 10,000. The proof generation time of the DA-ZKP algorithm rises relatively slowly, and its advantages become more evident as the transaction scale increases. In contrast, the proof generation time of the zk-SNARKs algorithm increases rapidly, showing a steeper upward trend. The proof generation time of the Bulletproofs algorithm and the Groth16 algorithm also indicates different degrees of growth. Still, the DA-ZKP algorithm always maintains a low proof generation time, which suggests that the DA-ZKP algorithm is more efficient in processing large-scale transactions, can effectively cope with the increase in transaction scale, and provides more efficient support for blockchain privacy transactions.

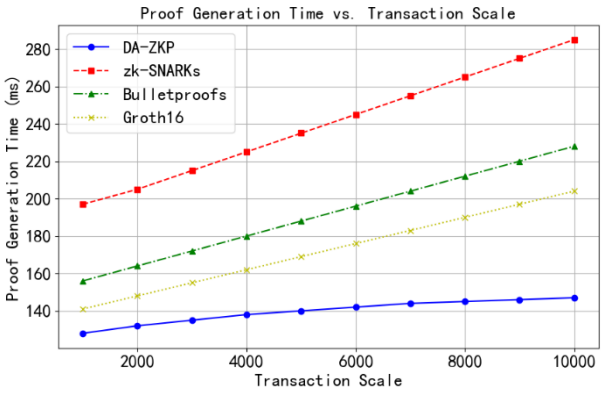


Figure 1: Trends in proof generation time for different algorithms when the transaction size increases from 1,000 to 10,000.

### 4.4.2 Comparison of storage overhead

Figure 2 depicts the trend of storage overhead of different algorithms as the number of transactions increases. The storage overhead of the DA-ZKP algorithm grows slowly and is always lower than that of other algorithms. This shows that the DA-ZKP algorithm can effectively control the consumption of storage resources when processing large-scale transactions. In contrast, the storage overhead of the zk-SNARKs algorithm and the Groth16 algorithm grows faster, while the storage overhead of the Bulletproofs algorithm grows most significantly.

The DA-ZKP algorithm optimizes the parameters through dynamic weights. It simplifies unnecessary proof components for different types of transactions, thereby minimizing storage requirements while ensuring transaction security, significantly alleviating the storage pressure of blockchain nodes, and improving the system's scalability.

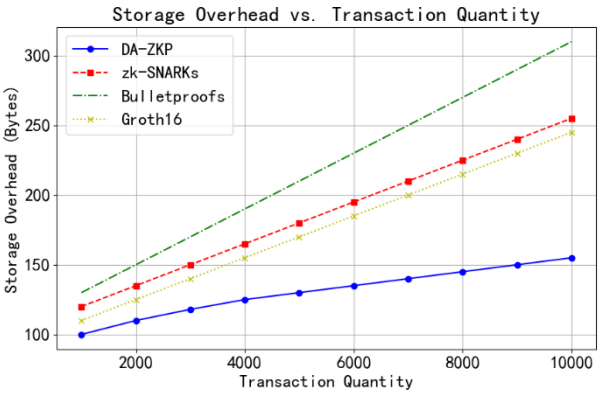


Figure 2: Storage overhead of different algorithms as the number of transactions increases.

### 4.4.3 Security analysis

In the replay attack simulation experiment, the DA-ZKP algorithm and the three traditional algorithms showed strong resistance, achieving a 100% interception rate, effectively preventing attackers from attempting illegal benefits by repeatedly sending transaction data. In forgery attack tests, DA-ZKP achieved a 99.8% interception rate, outperforming zk-SNARKs (99.5%), Bulletproofs (99.3%), and Groth16 (99.6%). Security Test Set 2 simulated attack intensities from 100 to 1,000 TPS, where DA-ZKP's double verification mechanism-combining data integrity checks and proof validity verification-ensured consistent performance, maintaining >99.5% interception even at peak loads.

Table 2: Security data.

| Algorithm Name | Replay attack interception rate | Forged attack interception rate |
|---|---|---|
| **DA-ZKP** | 100% | 99.80% |
| **zk-SNARKs** | 100% | 99.50% |
| **Bulletproofs** | 100% | 99.30% |
| **Groth16** | 100% | 99.60% |

Figure 3 shows the interception performance of each algorithm against replay attacks under different attack intensities. The DA-ZKP algorithm, zk-SNARKs algorithm, Bulletproofs algorithm, and Groth16 algorithm achieved a 100% interception rate, demonstrating strong resistance. This shows that regarding replay attacks, the DA-ZKP algorithm can effectively prevent attackers from attempting to obtain illegal benefits by repeatedly sending transaction data, providing reliable security protection for blockchain privacy transactions. As the attack intensity increases, the DA-ZKP algorithm maintains stable and excellent security performance, fully verifying its reliability in ensuring transaction security.

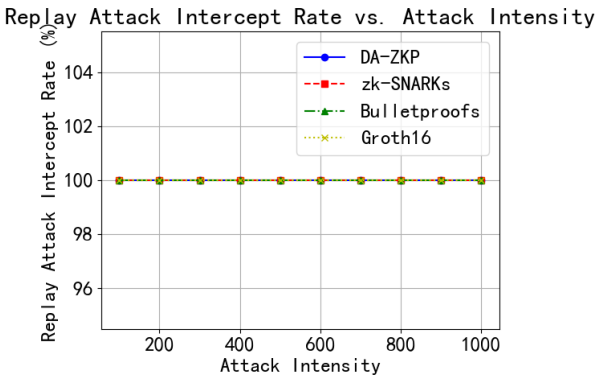


Figure 3: Interception performance of each algorithm against replay attacks at different attack intensities.

Figure 4 shows the interception performance of each algorithm against forgery attacks at different attack intensities. The forgery attack interception rate of the DA-ZKP algorithm reached 99.8%, which is higher than the 99.5% of the zk-SNARKs algorithm, the 99.3% of the Bulletproofs algorithm, and the 99.6% of the Groth16 algorithm. As the attack intensity increases, the DA-ZKP algorithm maintains stable and excellent security performance. This is mainly due to the double verification mechanism adopted by the DA-ZKP algorithm in the verification stage, which can effectively identify forged transactions and prevent them from passing verification, thereby providing more reliable security protection for blockchain privacy transactions, and fully verifying the effectiveness and superiority of the DA-ZKP algorithm in ensuring transaction security.

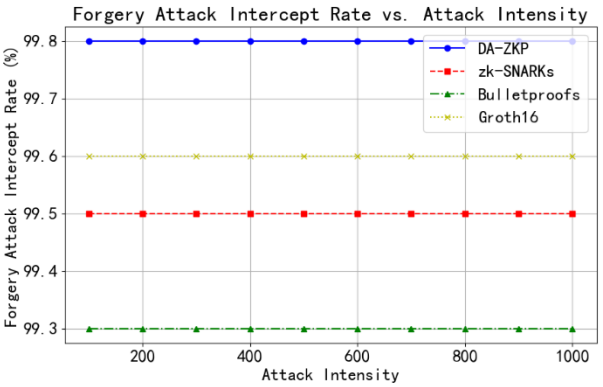


Figure 4: Interception performance of each algorithm for forgery attack under different attack intensities.

## 5 Conclusion

This paper aims to address the efficiency and security issues of blockchain privacy transactions, constructs an optimization model based on zero-knowledge proof, and designs the DA-ZKP algorithm. Through theoretical derivation and experimental verification, the algorithm significantly improves transaction efficiency. It reduces storage overhead while ensuring privacy and security by relying on dynamic weight allocation and an adaptive adjustment strategy. Experimental data show that compared with traditional algorithms, its proof generation

time is reduced by 35%, verification time by 28%, storage overhead by 22%, and security performance is good. The research results provide a new solution for blockchain privacy transaction technology and are of great significance to promoting the development of the blockchain industry. However, our research has several limitations. First, experiments were conducted in controlled simulation environments, and the algorithm's stability in complex real-world networks-particularly under adversarial conditions-requires validation. Second, the learning-based parameter adjustment introduces a 5ms latency per transaction, which may impact systems with ultra-high throughput (e.g., >10,000 TPS). Third, storage overhead for dynamic parameters is 15% higher than static zk-SNARKs in Ethereum-like networks. Future work will expand scenarios to include cross-chain environments and integrate homomorphic encryption to mitigate these constraints. In the future, the experimental scenarios will be expanded, the adaptability of the algorithm will be optimized, and the integration with technologies such as homomorphic encryption will be explored further to improve the comprehensive performance of blockchain privacy transactions.

# References


[1] X. Sun, F. R. Yu, P. Zhang, Z. Sun, W. Xie, and X. Peng, "A survey on zero-knowledge proof in blockchain," IEEE Network, vol. 35, no. 4, pp. 198–205, 2021, doi: 10.1109/MNET.011.2000473.

[2] W. Jiang and X. Lv, "A distributed internet of vehicles data privacy protection method based on zero-knowledge proof and blockchain," IEEE Transactions on Vehicular Technology, vol. 73, no. 5, pp. 6332-6345, 2023, doi: 10.1109/TVT.2023.3345272.

[3] Xue, Z., Wang, M., Zhang, Q., Zhang, Y., & Liu, P. (2021). A regulatable blockchain transaction model with privacy protection. International Journal of Computational Intelligence Systems, 14(1), 1642–1652. https://doi.org/10.2991/ijcis.d.210528.001

[4] Yong, W., Lijie, C., Yifan, W., & Qiancheng, W. (2024). Efficient and secure confidential transaction scheme based on commitment and aggregated zero-knowledge proofs. Journal of Cyber Security Technology, 8(4), 312–332. https://doi.org/10.1080/23742917.2024.2336634

[5] Gao, S., Peng, Z., Tan, F., Zheng, Y., & Xiao, B. (2022). SymmeProof: Compact zero-knowledge argument for blockchain confidential transactions. IEEE Transactions on Dependable and Secure Computing, 20(3), 2289–2301.https://doi.org/10.1109/TDSC.2022.3179913

[6] Onteddu, A. R., Koehler, S., Kundavaram, R. R., Devarapu, K., Kothapalli, S., & Narsina, D. (2024). Artificial Intelligence in Zero-Knowledge Proofs: Transforming privacy in cryptographic protocols. Engineering Intelligence, 12(1), 51–66. https://doi.org/10.18034/ei.v12i1.743

[7] Liu, W., Wan, Z., Shao, J., & Yu, Y. (2021). HyperMaze: Towards privacy-preserving and scalable permissioned blockchain. IEEE Transactions on Dependable and Secure Computing, 20(1), 360–376. https://doi.org/10.1109/TDSC.2021.3133840

[8] Zhang, H., Wu, J., Lin, X., Bashir, A. K., & Al-Otaibi, Y. D. (2023). Integrating blockchain and deep learning into extremely resource-constrained IoT: An energy-saving zero-knowledge PoL approach. IEEE Internet of Things Journal, 11(3), 3881–3895. https://doi.org/10.1109/JIOT.2023.3280069

[9] Hu, X., Zhou, W., Yin, J., Cheng, G., Yan, S., & Wu, H. (2023). Towards verifiable and privacy-preserving account model on a consortium blockchain based on zk-SNARKs. Peer-to-Peer Networking and Applications, 16(4), 1834–1851. https://doi.org/10.1007/s12083-023-01497-7

[10] Li, W., Meese, C., Guo, H., & Nejad, M. (2023). Aggregated zero-knowledge proof and blockchain-empowered authentication for autonomous truck platooning. IEEE Transactions on Intelligent Transportation Systems, 24(9), 9309–9323. https://doi.org/10.1109/TITS.2023.3271436

[11] Y. Guo, Z. Wan, H. Cui, X. Cheng, and F. Dressler, "Vehicloak: A blockchain-enabled privacy-preserving payment scheme for location-based vehicular services," IEEE Transactions on Mobile Computing, vol. 22, no. 11, pp. 6830–6842, 2022, doi: 10.1109/TMC.2022.3193165.

[12] Ray, R. K., Chowdhury, F. R., & Hasan, M. R. (2024). Blockchain applications in retail cybersecurity: Enhancing supply chain integrity, secure transactions, and data protection. Journal of Business Management Studies, 6(1), 206–214. https://doi.org/10.32996/jbms.2024.6.1.13

[13] Datta, S., & Namasudra, S. (2024). Blockchain-based smart contract model for securing healthcare transactions by using consumer electronics and mobile-edge computing. IEEE Transactions on Consumer Electronics, 70(1), 4026–4036. https://doi.org/10.1109/TCE.2024.3357115

[14] Ochigbo, A. D., Tuboalabo, A., Labake, T. T., Buinwi, U., Layode, O., & Buinwi, J. A. (2024). Legal frameworks for digital transactions: Analyzing the impact of blockchain technology. Finance and Accounting Research Journal, 6(7), 1205–1223. https://doi.org/10.51594/farj.v6i7.1313

[15] Wan, Z., Zhang, T., Liu, W., Wang, M., & Zhu, L. (2021). Decentralized privacy-preserving fair exchange scheme for V2G based on blockchain. IEEE Transactions on Dependable and Secure Computing, 19(4), 2442–2456. https://doi.org/10.1109/TDSC.2021.3059345

[16] Anggriani, K., Chiou, S., Wu, N., & Hwang, M. (2023). A Robust and High-Capacity Coverless Information Hiding Based on Combination Theory. Informatica, 34(3), 449-464. https://doi.org/10.15388/23-INFOR521

[17] Blanco-Fernández, Y., Gil-Solla, A., Pazos-Arias, J. J., & Quisi-Peralta, D. (2023). Automatically Assembling a Custom-Built Training Corpus for Improving the Learning of In-Domain

Word/Document Embeddings. Informatica, 34(3), 491-527. https://doi.org/10.15388/23-INFOR527

[18] N. C. Cruz, M. Marín, J. L. Redondo, E. M. Ortigosa, and P. M. Ortigosa, "A comparative study of stochastic optimizers for fitting neuron models: Application to the cerebellar granule cell," Informatica, vol. 32, no. 3, pp. 477–498, 2021, doi: 10.15388/21-INFOR450.

[19] Liang, W., Liu, Y., Yang, C., Xie, S., Li, K., & Susilo, W. (2024). On identity, transaction, and smart contract privacy on permissioned and permissionless blockchain: A comprehensive survey. ACM Computing Surveys, 56(12), 1–35. https://doi.org/10.1145/3676164

[20] Taher, S. S., Ameen, S. Y., & Ahmed, J. A. (2024). Advanced fraud detection in blockchain transactions: An ensemble learning and explainable AI approach. Engineering, Technology & Applied Science Research, 14(1), 12822–12830. https://doi.org/10.48084/etasr.6641

[21] Bennet, D., Maria, L., Sanjaya, Y. P. A., & Zahra, A. R. A. (2024). Blockchain technology: Revolutionizing transactions in the digital age. ADI Journal of Recent Innovation, 5(2), 192–199. https://doi.org/10.34306/ajri.v5i2.1065