future internet

MDPI

Article

Research on Blockchain Transaction Privacy Protection Methods Based on Deep Learning

Jun Li [1,*], Chenyang Zhang [1], Jianyi Zhang [2] and Yanhua Shao [3]

[1] School of Information Management, Beijing Information Science and Technology University, Beijing 100192, China; 2022020966@bistu.edu.cn
[2] School of Computing and Informatics, The University of Louisiana at Lafayette, Lafayette, LA 70504, USA; jianyi.zhang@louisiana.edu
[3] National Computer System Engineering Research Institute of China, Beijing 100083, China; stephen_yanhuashao@outlook.com
* Correspondence: lijun@bistu.edu.cn

Abstract: To address the challenge of balancing privacy protection with regulatory oversight in blockchain transactions, we propose a regulatable privacy protection scheme for blockchain transactions. Our scheme utilizes probabilistic public-key encryption to obscure the true identities of blockchain transaction participants. By integrating commitment schemes and zero-knowledge proof techniques with deep learning graph neural network technology, it provides privacy protection and regulatory analysis of blockchain transaction data. This approach not only prevents the leakage of sensitive transaction information, but also achieves regulatory capabilities at both macro and micro levels, ensuring the verification of the legality of transactions. By adopting an identity-based encryption system, regulatory bodies can conduct personalized supervision of blockchain transactions without storing users' actual identities and key data, significantly reducing storage computation and key management burdens. Our scheme is independent of any particular consensus mechanism and can be applied to current blockchain technologies. Simulation experiments and complexity analysis demonstrate the practicality of the scheme.

Keywords: deep learning; blockchain; privacy protection; regulatory functionality; cryptography

check for updates

Citation: Li, J.; Zhang, C.; Zhang, J.; Shao, Y. Research on Blockchain Transaction Privacy Protection Methods Based on Deep Learning. *Future Internet* **2024**, *16*, 113. https://doi.org/10.3390/fi16040113

Academic Editor: Daniel Gutiérrez Reina

Received: 23 February 2024
Revised: 15 March 2024
Accepted: 20 March 2024
Published: 28 March 2024

1. Introduction

Blockchain was originally proposed by Satoshi Nakamoto [1] as the underlying technology for Bitcoin [2]. Blockchain 1.0 is represented by Bitcoin, which focuses on solving the problem of decentralization of currencies and payments. Blockchain 2.0 is represented by Ethereum [3], which uses smart contracts to solve the trust issues of decentralization in the financial sector. Blockchain transaction involve three main components: the sender, the receiver, and the transaction amount. But both Bitcoin and Ethereum have limitations in privacy protection [4]. The identities of the sender and receiver are realized through user public key addresses, which has a certain degree of anonymity [5]. However, it is still possible to obtain the real identities of the traders by mining their associated information through data analysis or machine learning techniques. Furthermore, since the transaction amount is completely exposed on the blockchain and anyone can access it by querying the blockchain full node, attackers can deduce information such as account balances and fund flows [6], thereby compromising transaction privacy. Current anonymous cryptocurrencies like Dash, Monero and later Beam/Grin [7] utilize ZKPs (zero-knowledge proofs), ring signatures, and cryptographic commitments to protect the privacy of transaction details and participant identities [8]. However, these methods lack regulatory functions and may be used for illegal trading activities.

In order to address the issues mentioned above, we propose a scheme that balances privacy protection and regulatory functions. Our main contributions are summarized below:

1. We propose a blockchain transaction scheme that integrates a variety of cryptographic technologies to balance privacy protection and regulatory functions. Specifically, it adopts probabilistic public-key encryption to protect the user's identity from being exposed.
2. To validate the basic legality of blockchain transactions, our scheme employs cryptographic commitment schemes and zero-knowledge proof technology. It further integrates graph neural networks (GNNs) technology for anomaly detection in blockchain transaction data, thus meeting the requirements for transaction privacy protection and regulatory compliance without disclosing sensitive transaction information.
3. Our scheme allows regulatory authorities to avoid storing users' real identities and key information, significantly reducing storage and computational burdens. Under the premise of ensuring transaction efficiency as much as possible, it balances the implementation of privacy protection and regulatory functions.

This paper is organized as follows: Section 2 introduces the prerequisite knowledge necessary for constructing the scheme. Section 3 explores the blockchain transaction privacy protection model. Section 4 makes a comprehensive analysis of the scheme. Section 5 provides the conclusions and discussions of this study.

2. Preparatory Knowledge

2.1. Literature Review

Blockchain features include decentralized storage, data immutability, and consensus mechanisms. These features ensure the transparency and security of blockchain data, but they also create challenges for users' privacy protection. Researchers have introduced numerous privacy protection technologies to safeguard privacy.

Coin mixing is a significant privacy protection scheme that obscures the relationship between inputs and outputs in blockchain transactions to protect privacy. CoinJoin is a specific type of coin mixing scheme, its core idea is to merge transactions from multiple users into a single transaction to hide the source and destination of each user's funds. Dash [9] uses CoinJoin technology to ensure privacy by facilitating coin mixing through network master nodes. This process involves master nodes in chain mixing, where the output of one node becomes the input of another, undergoing multiple rounds of mixing to enhance anonymity.

Besides coin mixing, cryptographic privacy protection mechanisms are also a key research direction. Researchers use technologies like ZKP and ring signatures to secure transaction data confidentiality. Zerocoin [10] employs non-interactive ZKP and RSA accumulators within a cryptocurrency framework that allows Bitcoin conversion into Zerocoin, concealing the identities of both the sender and the receiver during transactions. However, Zerocoin faces issues like high costs and low transaction efficiency. Building on Zerocoin, Zerocash [11] introduces improvements using zk-SNARKs technology to reduce the transaction verification time required by Zerocoin, enhancing transaction efficiency. Moreover, Zerocash enables private transactions of differing amounts and allows direct transfers to user addresses. Monero focuses on privacy protection, it uses a ring signature framework to keep transaction senders anonymous. Additionally, it employs stealth address technology, generating a one-time address for each transaction to prevent address reuse.

Beyond privacy protection, regulatory technology is another vital aspect of blockchain transactions, which can effectively prevent illegal activities. Thus, balancing transaction regulation with privacy protection is an essential research direction. Li et al. [12] proposed a traceable Monero system that adds an accountability mechanism to the original system. It can trace the flow of funds and infer users' long-term addresses from one-time anonymous addresses. Sun et al. [13] proposed an MBDC framework for CBDC, which employs permissioned blockchain technology and utilizes a multi-blockchain structure and ChainID to improve scalability. However, MBDC focuses on regulatory features and has limitations in privacy protection. Zhang et al. [14] introduced Gemini-Chain, which adopts a dual-chain structure to store and access complete transaction and verification information, maintaining a balance between privacy security and regulatory functions. But its structure is relatively complex.

2.2. UTXO Model

UTXO (unspent transaction output) represents the outputs of transactions that have not yet been spent [15]. Multiple transactions are recorded on the Bitcoin ledger, each with several transaction inputs (transferors) and outputs (recipients). These outputs constitute the UTXO. Figure 1 shows an instance of the Bitcoin UTXO model. In this model, Transaction 1 has an input of 1 BTC, distributing two outputs, one of 0.4 BTC and another of 0.5 BTC, where the 0.1 BTC discrepancy acts as the transaction fee. Transaction 2 is the same as Transaction 1, with its output becoming the input for Transaction 3, thereby establishing a sequential linkage of transactions.

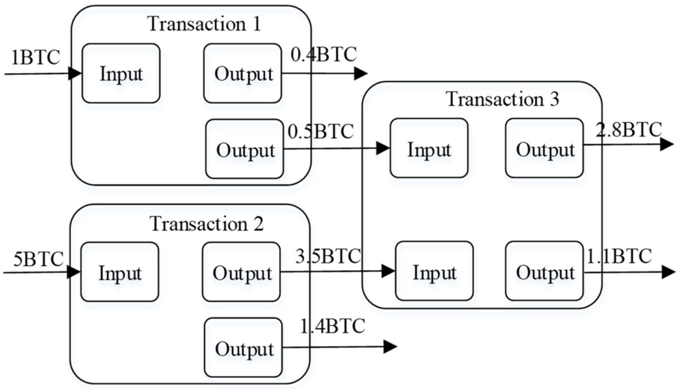

Figure 1. Example of a Bitcoin UTXO trading model.

2.3. Probabilistic Public-Key Cryptosystems

Probabilistic public-key encryption is a non-deterministic cryptography that generates randomly varying ciphertexts for the same plaintext. Under the computational security assumption, it is impossible to acquire any reliable information about the plaintext within polynomial time by using ciphertext correlation attacks. Goldwasser and Micali designed a probabilistic public-key scheme [16] (referred to as the GM probabilistic public-key encryption algorithm) using the quadratic residual theorem. However, the GM scheme has a high ciphertext expansion rate, leading to low transmission efficiency. Blum and Goldwasser proposed a more efficient probabilistic encryption scheme [17] (referred to as the BG cryptosystem), significantly reducing the expansion of ciphertext data. Hence, we primarily employ the BG scheme for encrypting user identity information. The BG probabilistic encryption mainly utilizes the BBS [18] generator to enhance ciphertext randomness. The detailed algorithm is as follows:

Parameter Setting: Let $n = p{\cdot}q$, where p and q are large primes, and $p \equiv q \equiv 3 mod 4$. Here, n is the public key, while p and q serve as the private keys. Define the plaintext space as $P = (Z^2)^m$, the ciphertext space as $C = (Z^2)^m \times Z_n^*$, and the key space as $K = \{(n, p, q)\}$.

Encryption: For plaintext message $x \in (Z^2)^m$ to be encrypted, the process is as follows:

1. Randomly select a seed $\mathrm{s_o}$ and use the BBS generator to produce m random bits $\mathrm{z_1 \cdots z_m}$ as the keystream;
2. Calculate $s_{m+1} = s_o^{2m+1} mod n$;
3. Calculate $y_i = (x_i + z_i) mod 2, 1 \leq i \leq m$;
4. The ciphertext is $\mathrm{c = E_K(x, r) = (y_i, \cdots, y_i, s_{m+1})}$.

Decryption: The process of decrypting the ciphertext $c = (y_i, \cdots, y_i, s_{m+1})$ is as follows:

1. Calculate $a_1 = \left(\frac{p+1}{4}\right)^{m+1} mod(p-1)$;
2. Calculate $a_2 = \left(\frac{q+1}{4}\right)^{m+1} mod(q-1)$;
3. Calculate $b_1 = s_{m+1}^{a_1} modp$;
4. Calculate $b_2 = s_{m+1}^{a_2} modq$;
5. Utilize the Chinese remainder theorem to calculate r, satisfying $r \equiv b_1 modp$ and $r \equiv b_2 modq$;
6. Using the BBS generator, derive $z_1 \cdots z_m$ from the seed $s_o = r$;
7. For each bit $1 \leq i \leq m$, compute $x_i = (y_i + z_i) mod2$;
8. The decrypted plaintext is $x = x_1 \cdots x_m$.

2.4. Identity-Based Cryptosystems

IBC (identity-based cryptography) [19] addresses the challenges associated with the supervision of digital certificates in public-key infrastructure (PKI). Within IBC, an entity's identification ID serves as its public key, while the private key is created using the KGC's (key generation center) master keys alongside the entity's ID. We employ the Chinese national standard algorithm SM9 as an example of IBC, and the SM9 algorithm is introduced as follows [20]:

Define P_1 as the generator of an additive cyclic group on an elliptical curve G_1, P_2 as the generator of a similar group on an elliptical curve G_2, $H(\cdot)$ as a hash function, and $e(\cdot)$ as a bilinear pair. Considering A as the signer and B as the verifier, the digital signature process for SM9 is as follows:

Key Generation: The KGC selects a random number $ks \in [1, N-1]$ as the master private key for signing and computes $P_{pub-s} = [ke]P_2$ as the master public key. Therefore, the master key pair is established as $\left(ke, P_{pub-s}\right)$. The identification of user A is ID_A. To create A's private signing key ds_A, the KGC computes $t_1 = H(ID_A, N) + ks$ and $t_2 = ks \cdot t_1^{-1}$ within the field F_N, subsequently obtaining $ds_A = [t_2]P_1$.

Signing: To sign a message M, A's signing process is as follows:

1. $g = e\left(P_1, P_{pub-s}\right)$;
2. Select a random number $r \in [1, N-1]$;
3. $w = g^r, h = H_2(M \| w, N), l = (r-h)modN$;
4. $S = [l]ds_A$. Then, M's signature is (h, S).

Verification: For verifying a signature $(\mathrm{h}', \mathrm{S}')$ on the message M', B follows the following steps:

1. $g = e\left(P_1, P_{pub-s}\right)$;
2. $t = g^{h'}, h_1 = H(ID_A, N)$;
3. $P = [h_1]P_2 + P_{pub-s}, u = e(S', P), w' = u \cdot t$;
4. $\mathrm{h}_2 = \mathrm{H}_2(\mathrm{M}' \| \mathrm{w}', \mathrm{N})$, if $\mathrm{h}_2 = \mathrm{h}'$, the signature verification is successful; if not, it fails.

2.5. Password Commitment Program

Cryptographic commitment is a two-stage interactive protocol involving a sender and a receiver. In Monero, the Pedersen commitment is a widely utilized homomorphic commitment scheme; it satisfies perfect hiding and computational binding properties and is used to protect the confidentiality of transaction values. The formula is:

$$P = r \cdot G + v \cdot H \tag{1}$$

P represents the concealed transaction amount, G and H are base points in elliptic curve cryptography, r is a random number, and v is the transaction amount. Additionally, the Bulletproofs zero-knowledge proof technique [21] is utilized to efficiently prove the range of transaction amounts.

2.6. Graph Neural Networks

GNNs (graph neural networks) [22] are a deep learning model for processing graph-structured data. They are capable of capturing complex relationships and dependencies between nodes in a graph. The representation of each node is updated based on information from neighboring nodes and propagated through the neural network layers. This enables GNNs to handle a wide variety of graphical architecture tasks. The important strength of GNNs is they can operate directly on the graph structure, effectively utilizing the topological information of the graph. They have been widely used for tasks such as social network analysis, recommender systems and knowledge graphs.

Common GNN variants include GCNs (graph convolutional networks) [23], GATs (graph attention networks) [24], and GAEs (graph autoencoders) [25], which use different mechanisms to aggregate and update node information. GCNs process graph data based on the concept of convolution, and its core principle is to update the representation of each node by aggregating information from neighboring nodes, thus capture the topology of the graph. In GCNs, the new feature representation of each node is realized by weighted average aggregation of its own features and the features of neighboring nodes. This process can be considered as a special convolution operation. GATs are neural network models for graph-structured data; they incorporate an attention mechanism to aggregate information from neighboring nodes. The core feature of GATs are that different neighbors contribute differently to the central node, which is reflected by attention coefficients. Each node updates its feature representation by first calculating the attention coefficients of all its neighbors (including itself) and then aggregating the neighbors' features weighted by these coefficients, considering the relative positional relationships of the nodes and individual characteristics. GAEs are a type of autoencoder model specifically for graph data, combining the characteristics of autoencoders with the capabilities of graph neural networks to capture reduced-dimensional embeddings of graph nodes. The core idea of GAEs are to encode graph nodes into a compressed space using a graph neural network, and then use a decoder to restore the structural details of the graph, such as the interconnections between nodes.

Blockchain transaction data can form complex graph structures, encompassing a variety of transaction entities and their interactions. This presents an ideal application scenario for GNNs. For instance, IBM's AI Lab has proposed using GNNs to identify money laundering rings in Bitcoin transactions [26]. Researchers created a temporal graph dataset comprising over 200,000 Bitcoin transactions (known as the Elliptic dataset) for identifying and classifying legal and illegal transactions. The graph in the dataset consists of 203,769 nodes and 234,355 edges, where nodes denote transaction entities and edges denote Bitcoin transaction flows between two entities. Each node is associated with 166 transaction-related features, including the first 94 local features representing the node's time step, in-degree and out-degree, payment expenditure fees, and derived features like the average amount of Bitcoin transactions; the remaining 72 are aggregated features obtained by aggregating maximum, minimum, standard deviation, and correlation coefficient information of neighbor transactions from the central node. About 2% of the nodes in the data are fraudulent, 21% are non-fraudulent, and the rest are unlabeled. Using GNN-based semi-supervised learning, each unlabeled Bitcoin transaction can be classified as illegal or legal.

The purpose of anomaly identity authentication detection is to infer account identities by capturing characteristics of transaction patterns. For instance, analyzing identity addresses can help determine the causes of cryptocurrency price fluctuations and their association with specific types of accounts, which is essential in safeguarding the blockchain's ecological integrity and establishing standardized transaction protocols. Traditional methods of identity recognition mainly include manual annotation and source code analysis. The former requires considerable manpower and is virtually impossible to accomplish for hundreds of millions of identity information. Although source code analysis for address identification (which involves analyzing the source code of a smart contract to identify potential backdoors or vulnerabilities) is more accurate, this approach is difficult to implement and many smart contracts do not disclose their source code. In contrast, graph neural network technology offers a novel solution. Liu et al. [27] designed a blockchain address identification method based on graph deep learning, and its framework is shown in Figure 2.

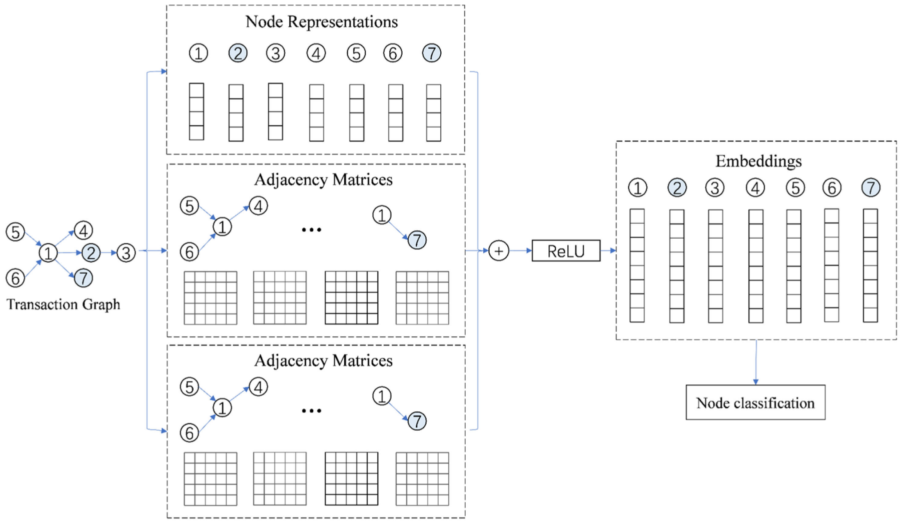

Figure 2. GNN Identity Recognition Framework (Numbers represent trading entities).

This model primarily comprises three major modules. The graph construction is based on node representation matrices, adjacency matrices and temporal density matrices to construct a directed weighted graph, generating distinctive feature representations for each node. In this framework, the node representation matrix includes information about the nodes' out-degree and in-degree, as well as the node type. The adjacency matrix is constructed with four types of edges based on transactions, contract calls, rewards, and other methods. The time density matrix is built according to the frequency and timing of interactions between account addresses. The model employs graph convolutional neural networks for learning and ultimately uses the softmax function to predict the node types for anomaly identity detection.

Shen et al. developed a model named I2BGNN [28], an end-to-end network specifically designed for processing the graph structure of blockchain transaction data. The I2BGNN model learns and captures patterns within transaction subgraphs and associates these patterns with user identities to enable de-anonymization. This approach infers user identities from transaction behaviors by analyzing transaction subgraphs, transforming the task of identity inference into a problem of graph classification. The model architecture of I2BGNN is shown in Figure 3:

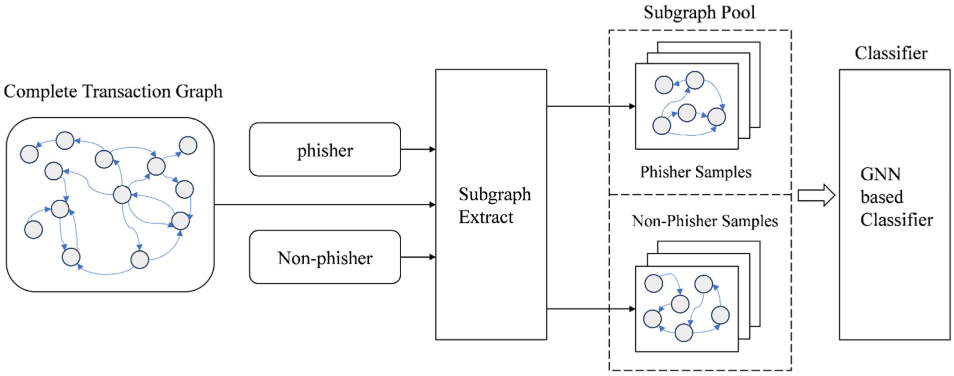

Figure 3. I2BGNN Model Framework.

The model initially constructs a graph network through blockchain transaction information, then samples labeled accounts, extracting subgraphs centered around the target accounts as input for the model. Finally, it trains a GNN model and evaluates the results. Experiments conducted on the EOSG and ETHG datasets demonstrate that this method achieves superior results in the domain of identity inference.

3. Deep Learning-Based Blockchain Transaction Privacy Protection Model

This paper integrates technologies such as the UTXO transaction model, the BG probabilistic public-key encryption algorithm, the IBC cryptographic system, Pedersen commitments, and graph neural networks to propose a supervised blockchain transaction privacy protection scheme. The design process is introduced in detail below.

3.1. Model

As shown in Figure 4, participants in the program primarily include the following: (1) The transaction's primary entities, the sender and the receiver, who aim to safeguard their anonymity and the confidentiality of the transaction sum through a secure transaction. In basic transaction activities, the sender transfers a certain amount of money to the receiver. (2) The miner, who verifies the legitimacy of the transaction and ensures that there is no double payment or fraud; after the transaction is verified, the miner packages it into a new block and stores it on the blockchain through a consensus mechanism to ensure the immutability of the transaction record. (3) Regulators, who are responsible for supervising and regulating the blockchain network. When necessary, they track the relevant participants and financial transactions to combat illegal financial activities. (4) Third parties, who may act as malicious actors using technical means to steal transaction-related information for undue gain.

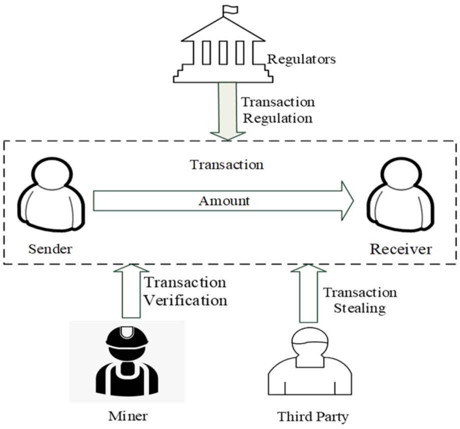

Figure 4. Blockchain Trading Entities.

The scheme comprises the following seven algorithms:

$BG.KeyGen(p,q)$: This is the key generation process of the BG algorithm. It generates the BG algorithm's public key pk and private key sk using large primes p and q. This algorithm provides probabilistic encryption, which generates different ciphertexts even if the same message is encrypted multiple times.

$BG.Enc(pk,m)$: This is the encryption process of the BG algorithm. It encrypts message m utilizing the public key pk of the probabilistic public-key BG algorithm to produce the ciphertext.

$BG.Dec(sk,ct)$: This is the decryption process of the BG algorithm. It decrypts ciphertext ct utilizing the private key sk of the probabilistic public-key BG algorithm to retrieve the plaintext. A user with the correct private key can successfully decrypt the ciphertext.

$IBC.KeyGen(sk,id)$: This is the key generation function of the SM9 algorithm based on IBC. Generate the user's private key by employing the SM9 algorithm's master key (*sk*) and the user's identifier (*id*).

$IBC.Enc(pk, m)$: This is the encryption process of the SM9 algorithm. It encrypts message m using the public key pk of the SM9 algorithm to produce ciphertext. SM9 is an identity-based encryption algorithm, which means that the encryption can be performed directly using the user's public identity information.

$IBC.Dec(sk, ct)$: This is the decryption process of the SM9 algorithm. It decrypts ciphertext ct using the private key sk of the SM9 algorithm to retrieve the plaintext. A user with the correct private key can decrypt successfully.

$IBC.Sign(sk, m)$: This is the signature process of the SM9 algorithm. It signs message m using the private key sk of the SM9 algorithm to obtain the signature value. This ensures the message remains unaltered throughout transmission, ensuring data integrity and non-repudiation.

This scheme provides public-key encryption and decryption using the BG probabilistic public-key cryptography algorithm (with a key size of 2048 b), which provides strong security guarantees for transactions, especially in terms of its ability to counter selective plaintext attacks. We also use the SM9 algorithm based on the IBC cryptosystem (with a key size of 256 b), whose encryption strength is equivalent to the RSA encryption algorithm of 3072 b. The SM9 algorithm allows the direct use of a user's identification data as the public key, which simplifies the process of distributing and managing the key. In addition, it provides digital signature and authentication functions; this approach can secure transactions and verify user identities in certain situations. The use of these two algorithms enhances the system compatibility and flexibility, enabling the scheme to meet different transaction scenarios. By combining different encryption algorithms, a more complex security framework is constructed for the scheme, which enhances the security of the whole system.

3.2. Anonymous Identity Realization

During the initialization phase of the scheme, the regulatory authority needs to generate three public-private key pairs: firstly, using the BG algorithm to produce private key Sk_{BG} and public key Pk_{BG}; secondly, as the KGC within the IBC framework, the regulatory authority creates a master public key MPK and corresponding a master private key MSK; thirdly, defining the identity marker in IBC as ID_a, considering ID_a as the public key, the signature private key MSK is created using the master private key $Sk_a = IBC.KeyGen(MSK, ID_a)$ based on the IBC algorithm. Then, users apply for key distribution from the regulatory authority using unique identifiable information ID_u (which needs to be self-proving, such as an email address, ID card number, or phone number).

After verifying user identity information, the regulatory authority encrypts it using the public key PK_{BG} of the BG probabilistic cryptography algorithm to generate $AID_1 = BG.Enc(Pk_{BG}, ID_u)$. To ensure that ID_u is certified by the regulatory authority, it needs AID_1 to be signed by the authority, generating $AID_2 = IBC.Sign(Sk_a, AID_1)$. Define $AID_u = AID_1 \| AID_2$. Since AID_1 is obtained using the BG probabilistic public-key encryption algorithm and has good randomness, and AID_2 is obtained through the IBC signature, AID_u also possesses good randomness, effectively hiding the user's real identity information ID_u.

Next, AID_u is used as the public-key identity. Utilizing the IBC algorithm, the regulatory authority generates the corresponding private key $Sk_u = IBC.KeyGen(MSK, AID_u)$ for the user. The user's verifiable true identity is denoted as ID_u, while AID_u represents their calculated anonymous identity, Sk_u being the corresponding private key. Employing the BG algorithm enables the generation of various AID_u from the same ID_u, establishing a one-to-many connection between ID_u and AID_u. This relationship permits the theoretical creation of limitless AID_u from the same ID_u, allowing users to continuously renew their anonymous identities.

For ease of subsequent description, define the transaction sender and receiver's identity markers as ID_s and ID_r, respectively. Through the above process, their corresponding anonymous identities AID_s and AID_r, and private keys Sk_s and Sk_r can be calculated.

When the sender transacts with the receiver, they can utilize Sk_s to decrypt the UTXO input script and set AID_r as the receiver's address, thereby maintaining identity anonymity.

3.3. Transaction Data Privacy Protection

When AID_s transacts with AID_r, without loss of generality, suppose the structure of the transaction is as shown in Figure 5.

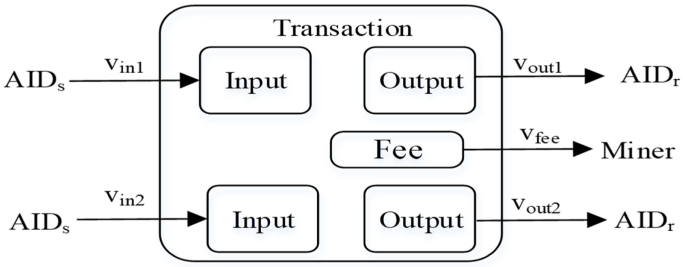

Figure 5. UTXO Transaction Structure.

In this transaction, there are two inputs with amounts v_{in1} and v_{in2} and two outputs: one for the transaction with AID_r, amounting to v_{out1}, and the other returning change to oneself, amounting to v_{out2}. Additionally, the v_{fee} portion is the transaction fee, serving as the miner's fee for packaging the transaction.

Our approach primarily employs Pedersen commitments to ensure the privacy protection of transaction amounts $(v_{in1}, v_{in2}, v_{out1}, v_{out2})$ while the transaction fees v_{fee} are publicly disclosed. For the transaction inputs, it is necessary to introduce previous outputs, represented as

$$P_{in1} = a_1G + v_{in1}H \tag{2}$$

$$P_{in2} = a_2G + v_{in2}H \tag{3}$$

where (a_1, v_{in1}) and (a_2, v_{in2}) can be decrypted by AID_s using the private key Sk_s.

The sender AID_s sets b_1 and b_2, then calculates

$$P_{out1} = b_1G + v_{out1}H \tag{4}$$

$$P_{out2} = b_2G + v_{out2}H \tag{5}$$

$$P_{fee} = v_{fee}G \tag{6}$$

P_{out1} and P_{out2} are primarily used to facilitate the miner's verification of the transaction's legitimacy. To enable the receiver to obtain (b_1, v_{out1}) and (b_2, v_{out2}), it needs to be encrypted using the public keys of both receiver and sender, resulting in

$$C_{out1} = IBC.Enc(AID_r, (b_1, v_{out1})) \tag{7}$$

$$C_{out2} = IBC.Enc(AID_s, (b_1, v_{out2})) \tag{8}$$

To ensure the transaction's legality, it must be verified that

$$v_{in1} + v_{in2} = v_{out1} + v_{out2} + v_{fee} \tag{9}$$

Consequently, the following can be calculated:

$$(P_{in1} + P_{in2}) - \left(P_{out1} + P_{out2} + P_{fee}\right) = (a_1 + a_2 - b_1 - b_2)H \tag{10}$$

Define the transaction's public key as

$$Pk_{Tx} = (a_1 + a_2 - b_1 - b_2)H \tag{11}$$

The transaction's private key is

$$Sk_{Tx} = (a_1 + a_2 - b_1 - b_2) \tag{12}$$

The entire transaction is defined as

$$M_{Tx} = \{P_{\text{in1}}, \text{P}_{\text{in2}}, (P_{\text{out1}}, \text{C}_{\text{out1}}), (P_{\text{out2}}, \text{C}_{\text{out2}}), \text{v}_{\text{fee}}\} \tag{13}$$

The transaction is signed using the SM9 signature algorithm, resulting in

$$Sig_{Tx} = IBC.Sign(SK_{Tx}, M_{Tx}) \tag{14}$$

Additionally, it is necessary to verify the transaction amount range to prevent negative values; Bulletproofs can achieve this. Due to the complexity involved, this paper will not elaborate further, but Reference [15] can be consulted for more information. The final transaction is

$$Tx = \left\{M_{Tx}, Sig_{Tx}, P_{range}\right\} \tag{15}$$

where P_{range} encompasses details verifying the range of the transaction value. Tx is broadcast across the network and after miners verify its legitimacy, it is incorporated into blocks and documented in the blockchain ledger via consensus protocols. The receiver can acknowledge the transaction using AID_r and then decrypt C_{out1} using their private key Sk_r to obtain transaction information, thereby completing the entire process of the transaction while concealing the amount of the transaction.

3.4. Transaction Legitimacy Verification

In a blockchain, transactions are recorded via a consensus mechanism. During the consensus process, miners verify transactions' legitimacy, which primarily includes the verification of participant identity and transaction amount legitimacy.

Identity legitimacy verification involves verifying the legitimacy of both the sender and receiver's identities. Within the UTXO model, the sender uses Sk_s to unlock UTXO inputs. Therefore, miners only need to use the sender's anonymous identity public key (called AID_s) to verify the legitimacy of the unlocking script signature.

Verifying the receiver's address is crucial to prevent fraudulent transactions and potential asset loss. Our scheme uses the receiver's anonymous identity public key (denoted as AID_r) as the receiver's address.

$$AID_r = AID_1 \| AID_2 \tag{16}$$

$$AID_1 = BG.Enc(Pk_{BG}, ID_u) \tag{17}$$

$$AID_2 = IBC.Sign(Sk_a, AID_1) \tag{18}$$

AID_2 represents the signature performed by the regulatory authority using its private key Sk_a on AID_1. Thus, to validate the unlocking script's signature, miners just have to utilize the sender's anonymized public identity key (designated as AID_2).

Transaction amount legitimacy also requires two aspects of verification: the equality of input and output amounts and the validity of the range of output amounts.

For the transaction M_{Tx} and Tx, miners need to calculate the transaction public key:

$$Pk_{Tx} = (P_{in1} + P_{in2}) - \left(P_{out1} + P_{out2} + v_{fee}G\right) \tag{19}$$

Using Pk_{Tx} to verify the legitimacy of the transaction signature Sig_{Tx}, fulfilling the requirement of input and output amount equality. To ensure the output amount is within a valid range, the existing Bulletproofs zero-knowledge proof technology is used to verify P_{range}, as referenced in [15].

3.5. Micro-Level Supervision Algorithm for Transaction Data

Blockchain transaction privacy protection is relative, primarily aimed at protecting user data from unauthorized access by malicious third parties. Nevertheless, regulatory authorities need transaction monitoring to combat illegal activities. Thus, it is crucial to ensure participant identities and transaction amounts can be regulated.

The anonymous identities of the participants in a transaction are AID_u, but their real identities ID_u are hidden.

$$AID_u = AID_1 \| AID_2 \tag{20}$$

The regulatory authority first verifies the legitimacy of the identity authentication using AID_2. Then, using its BG probabilistic public-key encryption algorithm's private key Sk_{BG}, it decrypts AID_1, obtaining the true identity of the transaction participant.

$$ID_u = BG.Dec(Sk_{BG}, AID_1) \tag{21}$$

To access the transaction amount information, the IBC algorithm is first utilized. As the KGC in the IBC cryptographic system, the regulatory authority can compute the receiver AID_r's private key Sk_r, denoted as

$$Sk_r = IBC.KeyGen(MSK, AID_r) \tag{22}$$

For transactions M_{Tx} and Tx, the regulatory authority then uses Sk_r to decrypt C_{out1}, obtaining

$$(b_1, v_{out1}) = IBC.Dec(Sk_r, C_{out1}) \tag{23}$$

The regulatory authority thus obtains the transfer amount v_{out1} to AID_r. Similarly, processing C_{out2} allows for the querying and monitoring of blockchain transactions.

3.6. Anomaly Transaction Data Detection Based on Graph Neural Networks

Anomaly detection is a method used to identify behaviors that deviate from the expected norm. The task of graph-based anomaly detection aims to uncover nodes, edges, or subgraphs within a network that exhibit significantly outlier characteristics. Anomaly detection of transaction data using GNNs is particularly useful in identifying fraudulent activities, money laundering and other anomalous patterns in financial transactions. This method is especially adept at handling complex financial networks, where transaction relationships can be modeled as graph structures, with nodes representing participants (such as individuals and companies) and edges representing transactions.

In this application, the GNNs' role is to leverage the structural information of the graph to learn underlying patterns within transaction data. Traditional fraud detection methods typically rely on rules or simple machine learning models that may not be able to capture complex non-linear fraud patterns. In contrast, GNNs can more effectively identify anomalous patterns by considering the relationships between nodes and transaction patterns in the transaction network.

The general design process for GNNs is divided into four parts, as illustrated in the following Figure 6:

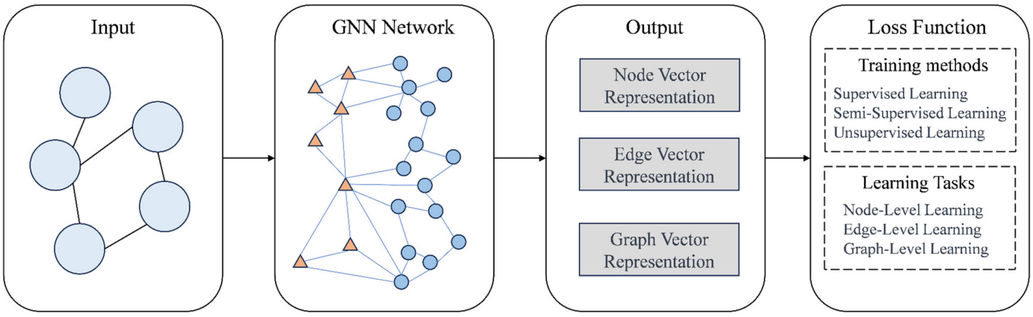

Figure 6. General GNN Design Process.

First, identify the graph structure relevant to the specific context and represent the data in graphical format. Next, determine the type of graph, such as directed/undirected or homogenous/heterogeneous. Subsequently, develop a loss function. Depending on the graph learning task, prediction types can be categorized at various levels: node, edge, community, or graph-wide. Finally, establish computational modules and train the model. The propagation module facilitates information exchange between nodes, enabling the aggregation of information to capture the graph's characteristics and topological details. The sampling module is responsible for graph sampling. For higher-dimensional subgraph representations, the pooling module can extract node information.

In a graph structure, each node is defined by its own features as well as the features of its connected neighbors. GNNs learn a state-embedding vector $h_v \in R^s$ for each node. This vector incorporates information from neighboring nodes. The node's state vector (denoted as h_v) can be utilized to generate an output O_v. Suppose $f(\cdot)$ is a function with parameters shared by all nodes, called the local transition function. This function updates node states based on neighboring node information. The local output function $g(\cdot)$ defines how the output is generated.

$$h_v = f\left(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]}\right) \tag{24}$$

$$o_v = g(h_v, x_v) \tag{25}$$

x_v denotes the feature vector of v, $x_{co[v]}$ denotes the feature vector of the edges linked to v, $h_{ne[v]}$ symbolizes the state vector for v's adjacent nodes, and $x_{ne[v]}$ indicates the feature vector for v's adjacent nodes. Suppose we assemble vectors of various types into their respective composite vectors. These composite vectors can be denoted as H for the state vectors, O for the output vectors, X for the feature vectors, and X_N for the node features. This aggregation makes the representation more compact:

$$H = F(H, X) \tag{26}$$

$$O = G(H, X_N) \tag{27}$$

F and G represent the global transition and output functions, respectively. They are obtained by stacking the node-wise functions f and g for all nodes in the graph. GNN iteratively computes state parameters using a traditional method based on the Banach fixed-point theorem.

$$H^{t+1} = F\left(H^t, X\right) \tag{28}$$

where H_t represents the tensor of H at the t-th iteration cycle.

Supervised learning is conducted using target information; the loss function is defined as follows:

$$loss = \sum_{i=1}^{p}(t_i - o_i) \tag{29}$$

With p supervised nodes, the loss function for GNN training incorporates true values t_i and predicted values o_i, and leverages a gradient descent strategy with the following steps: The state h_v^t is iteratively updated according to Equation (24) for T cycles until it approaches the fixed-point solution near Equation (26), at which point the obtained H will be close to the fixed-point solution $H^T \approx H$. During backpropagation, the gradient of the weight W is calculated from the loss, and then W is continuously updated based on the gradient computed in the previous step. After T cycles, the gradient with respect to h_v^0 is obtained, which is then used to update the model parameters.

The framework for anomaly detection in transaction data based on GNNs is a technique for identifying and locating abnormal information in transaction data, which plays a significant role in fields like finance, e-commerce, and insurance. The process flow of the GNN-based transaction data anomaly detection model can be broadly divided into the following:

1. Data Preprocessing: Initially, transaction data are converted into graph data where nodes represent transaction entities (e.g., users, merchants, banks, etc.) and edges represent transaction relationships (e.g., payments, transfers, refunds, etc.). Attributes of nodes and edges represent transaction characteristics (e.g., amount, time, frequency, type, etc.).
2. Graph Neural Networks: Subsequently, GNNs are employed for feature extraction and representation learning of the graph data. Utilizing the attribute information and structural information of nodes and edges, low-dimensional vector representations for each node and edge are obtained.
3. Anomaly Scoring: The vector representations of each node and edge are then assessed using an anomaly scoring function to compute their level of anomaly. Candidates for abnormal transactions are selected based on certain thresholds or ranking methods.
4. Anomaly Interpretation: Finally, the anomaly interpretation module explains the candidate nodes and edges involved in abnormal transactions. This analysis includes the causes and effects of anomalies, providing visual and interpretable results to help users understand and address abnormal transactions. The detailed design of the detection model includes the following:
    - Input: Transaction data forms an attribute graph G = (V, E, X). Nodes V denote transaction entities, while edges E depict the relationships between them. X is the node attribute matrix representing transaction features like amount, time, frequency, type, etc.
    - Output: An anomaly score $S \in R^{|V|}$ for each node, indicating the degree of anomaly. A higher score suggests a higher likelihood of anomaly.
    - Model Structure: The model has three parts: the graph neural network, the anomaly scoring function, and the loss function.
        - ◆ Graph Neural Network: The graph neural network extracts feature representations from the graph data. Various types of GNNs can be used. Assuming GCN as an example, the GNN formula is as follows:

          $$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \tag{30}$$

          Here, $H^{(l)} \in R^{|V| \times d_l}$ is the node feature matrix at layer l, $H^{(0)} = X$ is the feature dimension of layer l, $W^{(l)} \in R^{d_l \times d_{l+1}}$ is the trainable weight matrix at layer l, $\tilde{A} = A + l$ is the adjacency matrix with self-loops, $\tilde{D}$ is the degree matrix of $\tilde{A}$, and σ is an activation function like ReLU. After L layers of the GNN, the final node feature representation $H^{(L)} \in R^{|V| \times d_L}$ is obtained.
        - ◆ Anomaly Scoring Function: The anomaly scoring function computes the anomaly score based on the node's feature representation. Various types of scoring functions can be used, such as those based on reconstruction

error, distance, or density. Assuming a scoring function based on the reconstruction error as an example, the equation is

$$S = \left\|X - \hat{X}\right\|_F \tag{31}$$

Here, $\hat{X} \in R^{|V| \times d_0}$ is the reconstructed node attribute matrix, which can be decoded from H^L. $\|\cdot\|_F$ is the Frobenius norm, representing the root of the total sum of each matrix element squared. The larger the reconstruction error, the more inconsistent the node's attributes are with the normal pattern, hence the higher the anomaly score.

- ♦ Loss Function: The loss function optimizes the model's parameters to better distinguish between normal and abnormal nodes. Various types of loss functions can be used, such as contrastive, self-supervised, or adversarial. Assuming a contrastive-based loss function as an example, the formula can be expressed as

$$L = -\frac{1}{\|V\|} \sum_{v \in V} log \frac{exp(S_v/\tau)}{\sum_{u \in V} exp(S_u/\tau)} \tag{32}$$

Here, S_v is the anomaly score of node v, and τ is a temperature parameter for controlling the scaling of scores. The purpose of this loss function is to maximize the scores of anomalous nodes while minimizing the scores of normal nodes, thereby increasing the score differences between nodes. This loss function requires some prior anomaly labels, which can be obtained by simple rules or statistical methods, or by semi-supervised or unsupervised methods.

4. Scheme Analysis

4.1. Privacy Protection Capabilities

For user identity information, privacy is maintained by concealing the users' real information using AID_u. However, if AID_u is frequently used, such as in a transaction where AID_u is both an input and an output address, it becomes relatively easy to deduce that this represents change information for the transaction participant. To enhance privacy, we adopt the BG probabilistic random encryption, which can randomly choose different seeds s_0 for each encryption. One ID_u can generate many anonymous addresses AID_u, which cannot be distinguished from each other. Therefore, users can have the regulatory authority generate a batch of AID_u for them without changing ID_u. AID_u can be changed in each transaction, and third parties cannot even distinguish between the changed outputs in the transaction, let alone trace the entire process of the transaction. Third parties are unable to infer any effective information, thereby enabling this approach to achieve a robust privacy protection capability.

4.2. Performance Analysis

As previously described in Section 3.1, our scheme integrates various cryptographic techniques. To assess the overhead of our approach, we use processing time as a performance metric. t_{bg_kgen} represents the key generation time for the BG algorithm, t_{bg_enc} represents the encryption time for the BG algorithm, and t_{bg_dec} represents the decryption time for the BG algorithm. Similarly, t_{sm9_kgen} represents the key generation time for the SM9 algorithm, t_{sm9_enc} denotes the encryption time for the SM9 algorithm, t_{sm9_dec} denotes the decryption time for the SM9 algorithm, and t_{sm9_sign} represents the signing time for the SM9 algorithm. Additionally, t_{ped} represents the time for Pedersen commitments.

Our scheme includes the implementation of anonymous identities, transaction amount privacy protection, transaction legitimacy verification, and regulatory function implementation. For its corresponding computational overhead, refer to Table 1.

Table 1. Computational overhead of this scheme.

Scheme	Design Computation	Computational Cost
Anonymous Identity Implementation	One BG algorithm key generation, three SM9 algorithm key generations, one BG algorithm encryption, one SM9 algorithm signature	$t_{bg\_kgen} + 3t_{sm9\_kgen} + t_{bg\_enc} + t_{sm9\_sign}$
Transaction Amount Privacy Protection	Five Pedersen commitments, two SM9 algorithm encryptions, one SM9 algorithm signature	$5t_{ped} + 2t_{sm9\_enc} + t_{sm9\_sign}$
Transaction Legality Verification	One BG algorithm encryption, one SM9 algorithm signature, five Pedersen commitments	$t_{bg\_enc} + t_{sm9\_sign} + 5t_{ped}$
Regulatory Function Implementation	One BG algorithm encryption, one BG algorithm decryption, one SM9 key generation, one SM9 algorithm decryption	$t_{bg\_enc} + t_{bg\_dec} + t_{sm9\_kgen} + t_{sm9\_dec}$

Our experiments were performed on a multi-cluster configuration with CPU 3.8 GHz, GPU RTX 4090, and 32 GB of RAM per machine, completing the joint experiments on graph neural network detection and blockchain-based storage. Detailed information is provided in Table 2.

Table 2. Hardware configuration.

Hardware Environment	Configuration
CPU	3.80 GHz i9-13900 k
GPU	RTX 4090
RAM	32 GB

The BG and SM9 algorithms used in our scheme are mature and have been widely applied. As can be seen in Table 1, the transaction amount privacy protection and transaction legitimacy verification involve numerous Pedersen commitments, which incur some overheads. However, Pedersen commitments are currently key technologies and common methods in blockchain privacy protection. Therefore, the performance overhead of our scheme falls within a normal and acceptable range. We have analyzed the performance of this scheme; refer to Table 3 for details.

Table 3. Performance indicators.

Blockchain Performance		Anomaly Detection	
Average latency	6.13 s	Precision	0.802
Average throughput	14.53 TPS	Recall	0.756
Memory consumption	703 MB		
CPU usage	14%		

4.3. Comparative Analysis

Through experimental comparative analysis, a comprehensive multi-dimensional comparison is conducted with existing blockchain privacy protection algorithms in terms of privacy protection capability, technical implementation principles and features. The results are as shown in Table 4.

Table 4. Mainstream blockchain privacy protection technologies.

Name	Technical Implementation	Characteristics	Privacy Protection	Regulatory Function
Bitcoin	ECDSA, SHA256	Uses public keys for anonymous identities; transaction amounts are public	No	No
Ethereum	ECDSA, Keccak	Uses public keys for anonymous identities; transaction amounts are public	No	No
Dash	CoinJoin Technique	Simple approach; primarily relies on master nodes	Yes	No
Monero	Stealth Addresses, Ring Signatures, Pedersen Commitments	Ring signatures depend on other public keys, complex verification	Yes	No
Zcash	zkSNARKs, Pedersen Commitments	Strong anonymity, but complex parameter initialization and time-consuming proof generation	Yes	No
Beam/Grin	Pedersen Commitments, Aggregate Signatures	Utilizes MimbleWimble protocol, simple implementation but requires interactive process	Yes	No
BlockMaze [29]	zkSNARKs, Account Model	Adopting a dual-balance model combined with a two-step fund transfer process using zk-SNARK	Yes	No
Literature [13]	Multi-Chain Model	Uses a multi-chain architecture, complex node communication, loses decentralized features	No	Yes
Literature [14]	Consortium and Public Chain Technology	Implements dual-chain structure to exemplify regulatory model, ensuring transaction privacy, complex chain structure	Yes	Yes
Traceable Monero [12]	Elgamal encryption	Add user accountability to Monroe, track transaction information, and enable regulatory functions	Yes	Yes
Literature [30]	zkSNARKs, Attribute-Based Encryption, Account Model	Employs attribute-based encryption, establishing multi-level regulatory frameworks while ensuring privacy protection; this impacts transaction efficiency	Yes	Yes
This Paper	BG Probabilistic Public-Key Encryption, IBC Cryptosystem, Pedersen Commitments	Simple implementation but requires initial user authentication	Yes	Yes

In existing cryptocurrencies, Bitcoin and Ethereum have weak identity anonymity and fully disclose transaction amounts, lacking privacy protection features. Dash uses coin mixing technology to mix inputs and outputs of multiple transactions through master nodes, but this poses a risk of centralization; if the master nodes are controlled, it could lead to user privacy leakage. Monero utilizes stealth addresses and ring signature technology that do not rely on centralized nodes, but it requires mixing with other users' public keys, which leads to complex verification. Zcash uses the zkSNARKs scheme, achieving extreme anonymity and privacy security. However, the implementation of zkSNARKs is complex, requiring the setting of initial trusted parameters, and proof generation is time-consuming, which affects the practical efficiency. Beam and Grin both use the MimbleWimble protocol, employing Pedersen commitments and aggregated signature technology. Their implementation is simple, but they require an interactive process between transaction parties, making them inconvenient to use. BlockMaze [29] proposes a blockchain privacy-protective account model. It features a dual-balance model and designs a two-step fund transfer process incorporating zk-SNARKs. This method conceals account balances, transaction values and the connection between the sender and the receiver for privacy.

None of the above schemes provide regulatory function. Some researchers have started to explore the regulation of blockchain transactions. Sun et al. proposed a regulatable multichain model, but the communication between nodes is more complicated and its superchain structure is not conducive to privacy protection. Zhang et al. proposed a digital currency regulatory model using a dual-chain structure that combines consortium blockchain and public blockchain, which ensures transaction privacy through secret sharing and provides regulatory feature, but the dual-chain structure is complex to implement. Traceable Monero adds user accountability to the original system, tracking the movement of funds and also inferring a user's long-term address. Jia et al. [30] proposed a multilevel regulatory model by employing zkSNARKs and Attribute Based Encryption (ABE). It allows selective disclosure of transaction details while enforcing privacy protection measures. However, this approach affects the efficiency of transactions.

Through comparison with existing solutions, our scheme does not rely on centralized master nodes, does not require the introduction of ring signatures or other public keys, does not need to implement complex zkSNARKs proof processes, and avoids cumbersome interaction processes and complex multi-chain structures. By leveraging probabilistic public-key encryption, IBC (identity-based cryptography) systems, and Pedersen commitments, the blockchain transaction achieves both privacy protection and regulatory functionality. Additionally, regulatory authorities are not required to store users' real identities and key data, significantly reducing storage and computational burdens.

4.4. Analysis Discussion

We compared several privacy protection methods and conducted a comprehensive analysis based on experimental results. Our method can balance the needs of both privacy protection and regulatory functions, and the transaction efficiency has not been significantly affected; the impact on the system performance requirements and the transaction efficiency is in an acceptable range, and the scheme does not depend on a specific consensus mechanism and can be integrated into existing blockchain technology. In addition, the GNN-based anomalous transaction data detection method can help to identify anomalous data in transaction activities, ensuring the security of transactions.

5. Conclusions and Discussions

To address the challenge of balancing privacy protection and regulatory requirements in blockchain transactions, we integrate multiple cryptographic technologies, utilizing probabilistic public-key cryptography, identity-based cryptography (IBC), Pedersen commitments, and Bulletproofs techniques, combined with deep learning graph neural networks, to propose a blockchain transaction and regulatory scheme that offers both privacy protection and regulatory functions. Our scheme can be applied as an independent module in existing blockchain technologies. Our analysis of its security performance reveals that the blockchain transaction scheme is simple and practical. It holds extensive applicative value in areas such as digital asset risk analysis and financial transaction regulation.

Although our scheme balances privacy protection and regulatory functions, it still has some limitations. For example, there is still room for optimizing the improvement in transaction efficiency. It is an important research direction to improve the transaction speed, reduce the verification time, and minimize the computational overhead as much as possible in regulatable blockchain transactions. In addition, it is also important to choose appropriate privacy protection schemes for different transaction scenarios. Therefore, future research will focus on realizing the balance of privacy protection, regulatory function, transaction efficiency and other elements in blockchain transactions.

Legal regulations are also an important issue to be considered for blockchain technology. Blockchain's decentralization, tamper-resistance, transparency, and security make it widely applicable in the financial sector, which has resulted in some illegal activities choosing to use blockchain technology for transactions. Therefore, countries are developing specific laws and regulations related to cryptocurrencies, such as registration requirements for trading platforms and anti-money laundering regulations. At the same time, they are actively supervising the digital currency market to prevent possible financial risks. In addition, several other countries have established specific regulatory frameworks for blockchain, aiming to maintain financial security, protect consumer interests, and enhance the ability to combat money laundering. Our scheme provides regulatory capabilities for blockchain trading activities under the premise of privacy protection, which can help relevant organizations avoid illegal transactions. However, the current legal system of each country for blockchain technology is still not perfect, and the relevant rules are still being optimized. Therefore, future research should enhance the compatibility and flexibility of the scheme to adapt to the legal requirements of different countries.

Author Contributions: Investigation, software, writing—original draft preparation, validation, and methodology, C.Z.; conceptualization, methodology, supervision, and writing—review and editing, J.L.; computing resources and automated data collection, J.Z.; data curation and writing, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Basic Research Project of the National Defense Science and Industry Bureau (Project No. JCKY2022405C010) and the Translational Application Project of the "Wise Eyes Action" (Project No. F2B6A194). We would like to express our deepest gratitude to these organizations for their generous funding and support.

Data Availability Statement: The data can be shared up on request and the data are not publicly available due to security reasons.

Acknowledgments: We would like to express our gratitude to Yanhua Shao (The Sixth Research Institute of China Electronics and Information Industry Corporation) for providing computational resources and support.

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

Abbreviations

UTXO	Unspent Transaction Output
IBC	Identity-Based Cryptography
PKI	Public-Key Infrastructure
ZKP	Zero-Knowledge Proof
KGC	Key Generation Center
GNN	Graph Neural Network
GCN	Graph Convolutional Network
GAT	Graph Attention Network
GAE	Graph Autoencoder
ECDSA	Elliptic Curve Digital Signature Algorithm
SHA	Secure Hash Algorithm

References

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System [EB/OL]. 2009. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 22 February 2024).
2. Liu, Y.Z.; Liu, J.W.; Zhang, Z.Y.; Xu, T.G.; Yu, H. Overview on Blockchain Consensus Mechanisms. *J. Cryptologic Res.* **2019**, *6*, 395–432.
3. Buterin, V. A Next-Generation Smart Contract and Decentralized Application Platform. 2014, Volume 3. Available online: https://finpedia.vn/wp-content/uploads/2022/02/Ethereum_white_paper-a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf (accessed on 22 February 2024).
4. Zhu, L.; Gao, F.; Shen, M.; Li, Y.; Zheng, B.; Mao, H.; Wu, Z. Survey on Privacy Preserving Techniques for Blockchain Technology. *J. Comput. Res. Dev.* **2017**, *54*, 2170–2186.
5. Koshy, P.; Koshy, D.; McDaniel, P. *An Analysis of Anonymity in Bitcoin Using P2P Network Traffic*; Christin, N., Safavi-Naini, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8437, pp. 469–485. [CrossRef]
6. Ron, D.; Shamir, A. *Quantitative Analysis of the Full Bitcoin Transaction Graph*; Sadeghi, A.-R., Ed.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 7859, pp. 6–24. [CrossRef]
7. Fu, S.; Xu, H.; Li, P.; Ma, T.A. Study on the Anonymity of Digital Currencies. *Chin. J. Comput.* **2019**, *42*, 1045–1062.
8. Liu, Z.; Wang, D.; Wang, B. Privacy preserving technology in blockchain. *Comput. Eng. Des.* **2019**, *40*, 1567–1573.
9. Duffield, E.; Diaz, D. Dash: A Payments-Focused Cryptocurrency. Whitepaper. 2018. Available online: https://github.com/dashpay/dash/wiki/Whitepaper (accessed on 22 February 2024).
10. Miers, I.; Garman, C.; Green, M.; Rubin, A.D. Zerocoin: Anonymous distributed e-cash from bitcoin. In Proceedings of the 2013 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 19–22 May 2013; pp. 397–411.
11. Sasson, E.B.; Chiesa, A.; Garman, C.; Green, M.; Miers, I.; Tromer, E.; Virza, M. Zerocash: Decentralized anonymous payments from bitcoin. In Proceedings of the 2014 IEEE Symposium on Security and Privacy, Berkeley, CA, USA, 18–21 May 2014; pp. 459–474.
12. Li, Y.; Yang, G.; Susilo, W.; Yu, Y.; Au, M.H.; Liu, D. Traceable monero: Anonymous cryptocurrency with enhanced accountability. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 679–691. [CrossRef]
13. Sun, H.; Mao, H.; Bai, X.; Chen, Z.; Hu, K.; Yu, W. Multi-blockchain model for central bank digital currency. In Proceedings of the 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT), Taipei, Taiwan, 18–20 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 360–367.
14. Zhang, J.; Wang, Z.; Xu, Z.L.; Ouyang, Y.; Yang, T. A Regulatable Digital Currency Model Based on Blockchain. *J. Comput. Res. Dev.* **2018**, *55*, 2219–2232.
15. Chang, X.; Zhao, Y. Scaling bitcoin: The state of development and future trend. *Comput. Appl. Softw.* **2019**, *36*, 49–56.
16. Goldwasser, S.; Micali, S. Probabilistic encryption. *J. Comput. Syst. Sci.* **1984**, *28*, 229–270. [CrossRef]
17. Blum, M.; Goldwasser, S. An efficient probabilistic public key encryption scheme which hides all partial information. In *Advances in Cryptology Proceedings of CRYPTO' 84 (LNCS196), Santa Barbara, CA, USA, 19–22 August 1985*; Springer: Berlin/Heidelberg, Germany, 1985; pp. 289–299.
18. Vybornova, Y.D. Password-based key derivation function as one of Blum-Blum-Shub pseudo-random generator applications. *Procedia Eng.* **2017**, *201*, 428–435. [CrossRef]
19. Shamir, A. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology: Proceedings of CRYPTO 84 4, Santa Barbara, CA, USA, 19–22 August 1984*; Springer: Berlin/Heidelberg, Germany, 1985; pp. 47–53.
20. *GM/T 0044-2016 SM9*; People's Republic of China Cryptography Industry Standard. Identity-Based Cryptographic Algorithm. State Cryptography Administration: Beijing, China, 2016.
21. Bünz, B.; Bootle, J.; Boneh, D.; Poelstra, A.; Wuille, P.; Maxwell, G. Bulletproofs: Short proofs for confidential transactions and more. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 315–334.
22. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef] [PubMed]
23. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
24. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
25. Kipf, T.N.; Welling, M. Variational graph auto-encoders. *arXiv* **2016**, arXiv:1611.07308.
26. Weber, M.; Domeniconi, G.; Chen, J.; Weidele DK, I.; Bellei, C.; Robinson, T.; Leiserson, C.E. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv* **2019**, arXiv:1908.02591.
27. Liu, X.; Tang, Z.; Li, P.; Guo, S.; Fan, X.; Zhang, J. A graph learning based approach for identity inference in dapp platform blockchain. *IEEE Trans. Emerg. Top. Comput.* **2020**, *10*, 438–449. [CrossRef]
28. Shen, J.; Zhou, J.; Xie, Y.; Yu, S.; Xuan, Q. Identity inference on blockchain using graph neural network. In *Blockchain and Trustworthy Systems: Third International Conference, BlockSys 2021, Guangzhou, China, 5–6 August 2021*; Revised Selected Papers 3; Springer: Singapore, 2021; pp. 3–17.

29. Guan, Z.; Wan, Z.; Yang, Y.; Zhou, Y.; Huang, B. BlockMaze: An efficient privacy-preserving account-model blockchain based on zk-SNARKs. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 1446–1463. [CrossRef]
30. Jia, W.; Xie, T.; Wang, B. A privacy-preserving scheme with multi-level regulation compliance for blockchain. *Sci. Rep.* **2024**, *14*, 438. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.