Conditional Privacy-Preserving Transaction for the Unspent Transaction Output-Based Multi-Chain Blockchain System

Jie Cui, *Senior Member, IEEE*, Wenting Zhuang, Hong Zhong, Qingyang Zhang, Fengqun Wang, and Debiao He, *Member, IEEE*

***Abstract*—The anonymity of blockchain may be exploited by criminals for illegal fund transfers, thus a conditional privacy-preserving scheme is important for blockchain regulation. Currently, sharding technology under a multi-chain architecture is used to improve blockchain scalability. However, current conditional privacy-preserving schemes cannot work on this architecture. To protect the privacy of the transaction, we present a conditional privacy-preserving transaction scheme (MC-CPPT) for multi-chain blockchain system. In this system, we proposed a zero-knowledge proof based anonymous transaction, in terms of the identities of transaction participants and amounts, which also enables the unlinkability of transactions and indistinguishability between cross-chain and intra-chain transactions in multi-chain blockchain system. In addition, a multi-node regulatory agency is introduced to control the transaction amount and frequency in the system without a single point of failure. Moreover, an ECC-based encryption scheme is proposed to achieve the traceability of suspicious transactions. A security model is defined and the security of MC-CPPT is demonstrated to meet the expected security goals. Evaluating the prototype revealed acceptable performance and additional security features.**

***Index Terms*—Blockchain, scalability, UTXO model, conditional privacy-preserving.**

I. Introduction

BLOCKCHAIN technology leverages security mechanisms, such as public key encryption, hash chains, and proof-of-work consensus, to achieve a decentralized, tamper-resistant, and forgery-proof shared ledger [1], [2], [3], [4]. Owing to its decentralization, immutability, transparency, and trustlessness, blockchain has become a groundbreaking technology, applied in various fields [5], [6], [7], [8], [9]. Blockchain technology currently faces two major issues: poor scalability and lack of regulatory oversight in privacy preservation during cryptocurrency transactions. To address the scalability and efficiency concerns, blockchain systems based on sharding technology have been proposed [10], [11], [12], such as Pyramid [13], Fleetchain [14], Rapidchain [15], HIBEChain [16], and OmniLedger [17]. Furthermore, HIBEChain [16] combines sharding mechanisms with hierarchical multi-chain parallel processing to handle transactions, thereby increasing the system throughput. This structure is suitable for the blockchain system of smart cities with IoT consumer electronics [18], [19].

The aforementioned studies addressed the scalability and efficiency issues to a certain extent. However, owing to the anonymity of the blockchain, the cryptocurrency transactions it supports still face a lack of regulatory oversight for privacy preservation. Currently, there are two main methods for recording transactions on a blockchain: the account model and the unspent transaction output (UTXO) model. Transactions in the account model inevitably expose identity and balance privacy, making the UTXO model more advantageous for privacy preservation. Consequently, researchers have focused on privacy preservation in UTXO models. Although strong privacy preservation provides users with a high level of data security, it also creates opportunities for criminals [20]. Some criminals exploit the anonymity of the UTXO model blockchain to hide their true identities during illegal fund transfers, evading traditional financial regulations. Therefore, regulating privacy preservation and implementing traceable privacy-preserving transactional schemes are imperative.

Recently, many conditional privacy-preserving schemes [21], [22], [23], [24], [25], [26], [27] have been proposed for UTXO blockchains; however, most of them are designed for single-chain systems, such as DCAP [21], PRCash [24], TRCT [25], and Traceable Monero [26]. However, solutions for existing single-chain systems cannot be directly applied to multi-chain systems. These blockchains may have some form of connection at a physical or logical level, whereas single-chain systems involve only a single blockchain. In multi-chain systems, intra-

Received 5 August 2024; revised 10 March 2025; accepted 25 April 2025. Date of publication 5 May 2025; date of current version 11 June 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62202005, Grant U24A20243, Grant 62372002, and Grant 62272002; in part by the Natural Science Foundation of Anhui Province, China under Grant 2208085QF198; and in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2022-049. Recommended for acceptance by C. Li. *(Corresponding author: Hong Zhong.)*

Jie Cui, Wenting Zhuang, Hong Zhong, Qingyang Zhang, and Fengqun Wang are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China, and also with Anhui Engineering Laboratory of IoT Security Technologies, Anhui University, Hefei 230039, China (e-mail: zhongh@ahu.edu.cn).

Debiao He is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with Shanghai Key Laboratory of Privacy Preserving Computation, MatrixElements Technologies, Shanghai 201204, China (e-mail: hedebiao@163.com).

Digital Object Identifier 10.1109/TC.2025.3566868

chain or cross-chain transactions may become easily distinguishable because of the information contained in the transactions. This allows attackers to infer sensitive information, such as user identities or transaction habits, by analyzing transaction types combined with big data, leading to privacy leaks. Although HIBEChain employs a decentralized hierarchical identity signature scheme to enhance system security, it disregards privacy preservation. Moreover, Deshpande [27] uses adapter signatures and Schnorr signatures to ensure that cross-chain transactions are indistinguishable from intra-chain transactions, albeit with an added overhead. HyperMaze [28] is one of the few solutions that consider both privacy preservation and scalability; however, it is designed for the account model, which can hardly hide the identity information of the payer. Currently, multi-chain architectures increase transaction complexity, and hiding the indistinguishability between intra-chain and cross-chain transactions remains a challenge that is crucial for privacy preservation.

To address these issues, we propose a conditional privacy-preserving transaction scheme (MC-CPPT) for the UTXO-based multi-chain blockchain system. The contributions of the proposed scheme are as follows:

- To address the scalability of blockchain, an extended hierarchical framework based on HIBEChain is proposed for UTXO-based blockchain transactions. This framework supports the parallel processing of transactions across multiple blockchains, allowing the conversion of incompatible cryptocurrencies such as Bitcoin and Ethereum into new compatible cryptocurrencies and reverse conversion.
- To improve the anonymity of blockchain transactions, a conditional privacy-preserving transaction scheme within our proposed framework is proposed based on zero-knowledge proof and ECC-based encryption. In the proposed scheme, the users could send anonymous transactions, in terms of the identities of the transaction participants or specific amounts and the cross/intra-chain transactions are indistinguishable. Moreover, the information encrypted by ECC-based algorithms is provided for the traceability of suspicious transactions.
- Security analyses of the scheme are conducted, and the results indicate that MC-CPPT can achieve the expected security features. We also evaluated the performance of MC-CPPT and compared it with existing schemes. Compared to other privacy-preserving transaction schemes, the evaluation results prove that the MC-CPPT conditional privacy-preserving scheme only generates a considerably low additional overhead.

The remainder of this paper is organized as follows. Section II reviews the related work on blockchain privacy preservation. Section III introduces the knowledge prerequisites used in this study. Section IV describes the system model and security model of MC-CPPT. Section V outlines the MC-CPPT and details its design. Section VI demonstrates the security of MC-CPPT. Section VII evaluates the performance of the MC-CPPT. Section VIII concludes the study.

II. Related Works

The current state of research on privacy preservation and conditional privacy preservation will be provided in the following text.

A. Privacy-Preserving Transaction Scheme

To protect the privacy of both transaction parties, Miers et al. [29] proposed Zerocoin, an encrypted extension of Bitcoin, which utilizes zero-knowledge proofs in the process of minting and redeeming Zerocoin to hide the information of transaction senders and receivers. However, this scheme does not support users directly paying Zerocoin to other users, and the denominations of Zerocoin are fixed. Based on Zerocoin, Eli et al. [30] proposed Zerocash, allowing users to directly pay Zerocoin to other users and supporting transactions of any denomination, increasing flexibility. However, anonymous transactions in this scheme are untraceable. Nosouhi et al. [31] proposed an unlinkable coin called Ucoin, which does not rely on trusted third-party entities. It destroys the link between input addresses and output addresses in transactions by mixing multiple users' transactions into a single aggregated transaction. Xiao et al. [32] introduced a decentralized mixed-coin scheme where users with transaction intentions form a group. By splitting and mixing coins, the transaction participants and transaction amounts are concealed. However, the information for establishing the group needs to be published on third-party social software, increasing the risk of privacy leakage. Ruffing et al. [33] combined CoinJoin, confidential transactions and additionally stealth addresses to propose ValueShuffle, a mixing confidential transactions scheme that provides comprehensive privacy for cryptocurrencies, including sender anonymity, receiver anonymity, and transaction amount privacy. However, this scheme is vulnerable to DoS attacks and sybil attacks. Guan et al. [34] proposed BlockMaze, an account-based privacy-preserving scheme that employs a dual-balance mechanism and a two-stage fund transfer process to ensure unlinkability between two transactions. However, this scheme is designed for single-chain systems, with low throughput, which is difficult to meet the needs of large-scale financial trading systems. Liu et al. [28] applied the dual-balance account model of BlockMaze to each leaf blockchain in the HIBEChain hierarchical multi-chain architecture, proposing a privacy-preserving and scalable permissioned blockchain system called HyperMaze. However, transactions in this scheme are untraceable.

B. Conditional Privacy-Preserving Transaction Scheme

Liang et al. [21] proposed a traceable distributed anonymous payment scheme based on Zerocash, utilizing homomorphic commitments to tally transaction amounts and frequencies. However, the sender of a pour transaction in this scheme can determine whether the receivers of his two pour transactions are the same person. Lin et al. [22] introduced a decentralized conditionally anonymous payment system called DCAP, where users' anonymous addresses are derived from long-term addresses. Managers can use anonymous addresses to trace suspicious transaction users' long-term addresses (i.e., real identities) and revoke their permissions. However, abnormal detection to determine whether a transaction is suspicious requires additional resources as it involves tracking transaction addresses and fund flows to monitor intra-chain and cross-chain financial activities on the blockchain. Lin et al. [23] designed a traceable anonymous key generation mechanism, publicly verifiable authorization mechanism, knowledge signatures and smart contracts using public-key encryption, partially homomorphic encryption and accumulators, proposing an anonymous, confidential, and auditable transaction system called ACA. However, this scheme only limits the maximum expenditure amount of a user in a single transaction, and criminals intending to launder money may utilize this rule to initiate multiple transactions while controlling each transaction within the limit. Wüst et al. [24] proposed PRCash, a blockchain currency that enables fast payments, privacy preservation, and regulatory control, limiting the total expenditure of users within a period. However, attackers can link two transactions based on pseudonymous identities in anonymous transactions. Duan et al. [25] combined the partially extractable zero-knowledge proof scheme EPoK and the classical anonymous transaction protocol RingCT to propose a traceable anonymous transaction protocol called TRCT, which is the first scheme that publicly verifies the traceability of transactions while maintaining anonymity, ensuring that users cannot forge relevant proofs to evade tracing. Li et al. [26] improved the traditional cryptocurrency Monero by introducing traceable Monero and proposed two tracing mechanisms. Regulatory agencies can trace one-time addresses and long-term addresses separately and issue traceability proofs that can be verified by any user. However, this scheme is tailored specifically for Monero.

The aforementioned studies have contributed to privacy preservation and conditional privacy preservation, but many challenges and limitations remain. For instance, the use of third-party social software may lead to an increased risk of privacy breaches, necessitate additional resources for detecting suspicious transactions, result in linkability, or may only be applicable to certain specific cryptocurrencies.

III. Preliminaries

A. Hierarchical Blockchain System

HIBEChain [16] is a hierarchical blockchain system that enhances scalability by parallel processing cryptocurrency transactions.

As shown in Fig. 1, HIBEChain consists of multiple blockchains organized in a tree structure. Each node in the tree represents an independent blockchain. Based on their position in the tree, blockchains can be categorized into leaf blockchains, intermediate blockchains, and root blockchains. Each leaf blockchain manages terminal devices within a specific area, where these devices submit transactions to their respective leaf blockchain without interaction with other types of blockchains. Here is an overview of the functions of each type of blockchain:

- Leaf Blockchains: They are responsible for handling payment transactions from terminal devices within their jurisdictions. The validators of the leaf blockchains, upon successful verification, collect the transactions into a block and submit a header transaction to their parent chain. The header transaction contains the hash value of the block, which is referred to as the block header.
- Intermediate Blockchains: They are responsible for handling header transactions from their child chains. After verifying the authenticity of header transactions, the validators of the intermediate blockchains collect them into a block and submit a header transaction, containing the hash value of this block to their parent chain.
- Root Blockchain: The root blockchain is unique and responsible for handling header transactions submitted by its child chains. The validators of the root blockchain verify the authenticity of header transactions, collect them into a block to form a final block header and broadcast it to its child chains. Similarly, child chains also broadcast it to their respective child chains, ultimately achieving global consensus.

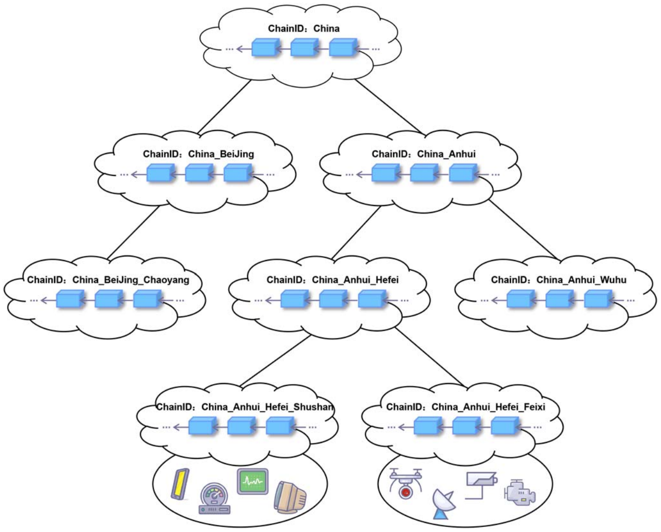

ChainID：China
ChainID：China_BeiJing
ChainID：China_Anhui
ChainID：China_BeiJing_Chaoyang
ChainID：China_Anhui_Hefei
ChainID：China_Anhui_Wuhu
ChainID：China_Anhui_Hefei_Shushan
ChainID：China_Anhui_Hefei_Feixi

Fig. 1. Example of a 4-layer hierarchical blockchain system.

B. Extended Merkle Tree

Unlike the Merkle tree (MT) used in single-chain systems, HyperMaze [28] employs an extended Merkle tree (EMT) to prove the validity of specific transactions. The EMT is essentially a combination of multiple Merkle trees. Structurally, the EMT is obtained by replacing the blockchains in the hierarchical blockchain system's tree structure with standard Merkle trees. The root node of a child Merkle tree corresponding to a child chain is also a leaf node of the parent Merkle tree corresponding to the parent chain. To prove the validity of a transaction, it is only necessary to demonstrate that the Extended Merkle proof from that transaction to the root are valid. Fig. 2 presents an example of the EMT proof. EMT Distributed across the hierarchical blockchain system, each blockchain maintains a portion of the Merkle tree and synchronizes the tree structure with each other. Users only need to acquire the complete proof when they want to demonstrate the existence of a specific transaction,

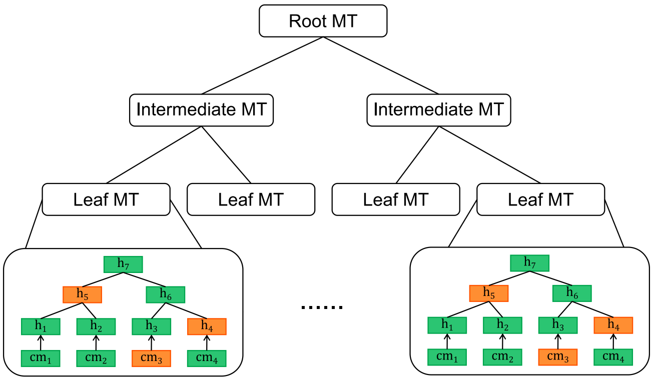


Fig. 2. Example of EMT proof.

which can be achieved by inquiring the blockchain validator in their respective jurisdiction.

C. Zk-SNARK

Zk-SNARK (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) [35] is suitable for proving statements that are denoted as arithmetic circuits. Assuming C represents an arithmetic circuit, which R_C expresses a NP relation $R_C = \{(x,\omega)|C(x,\omega)=0\}$. The language of is $L_C = \{x|\exists\omega, s.t. C(x,\omega)=0\}$. zk-SNARK is suitable for statements that can be expressed as arithmetic circuits, producing proofs of small size that can be verified in milliseconds. A zk-SNARK system for the language L_C consists of three polynomial algorithms $(KeyGen, Prove, Verify)$:

- $(PK_{\mathbb{Z}}, VK_{\mathbb{Z}}) \leftarrow KeyGen(1^\lambda, C)$. The algorithm takes a security parameter λ and a circuit C as input, and outputs proving key $PK_{\mathbb{Z}}$ and verification key $VK_{\mathbb{Z}}$. Both of them are public to the prover and validator.
- $\pi \leftarrow Prove(PK_{\mathbb{Z}}, x, \omega)$. The algorithm takes a proving key $PK_{\mathbb{Z}}$, a public input x and a private input ω for the circuit C, and outputs a proof π.
- $\{0,1\} \leftarrow Verify(VK_{\mathbb{Z}}, x, \pi)$. The algorithm takes a verification key $VK_{\mathbb{Z}}$, a public input x, and a proof π, and outputs 1 if the verification is successful, or 0 otherwise.

D. ECC-Based Threshold Encryption and Signature Scheme

ECC-based threshold encryption and signature scheme require participants to collaboratively generate secret shares and system public keys. In the ECC-based threshold encryption scheme without a trusted center [36], any t participants can jointly decrypt without revealing the system private key. In the ECC-based threshold signature scheme without a trusted center [37], any t participants can jointly sign without revealing the system private key. This encryption and signature scheme consists of six algorithms $(Setup, KeyGen.Enc, Dec, Sig, Ver)$:

- $pp \leftarrow Setup(1^\lambda)$. The $Setup$ algorithm inputs security parameters λ and outputs public parameters pp, including finite fields F_p and elliptic curves $\mathbb{G}_1$. The $Setup$ algorithm also sets participant identities $\{ID_i\}$ and thresholds (t, n).
- $(PK, \{SK_i\}) \leftarrow KeyGen(pp)$. The $KeyGen$ algorithm inputs public parameters pp. Each participants generates a partial private key SK_i and uses it to compute public information, which is then used to compute the system public key PK based on the public information. PK is made public to all users.
- $c \leftarrow Enc(PK, m)$. The Enc algorithm inputs public key PK and plaintext message m, and outputs ciphertext c.
- $m \leftarrow Dec(\{SK_i\}, c)$. The Dec algorithm inputs ciphertext c and private keys $\{SK_i\}$ of any t participants, and uses the private keys to generate decryption factors. The plaintext message m is then computed using these decryption factors.
- $\sigma \leftarrow Sig(\{SK_i\}, m)$. The Sig algorithm inputs private keys of any combination of t participants SK_i and the message m to be signed. First, the hash of m is computed, and then participants calculate partial signatures using SK_i. Any combination of t partial signatures can be combined to form a complete signature σ.
- $\{0,1\} \leftarrow Ver(PK, m, \sigma)$. The Ver algorithm inputs message m, signature σ, and public key PK. If verification is successful, it outputs 1; otherwise, it outputs 0.

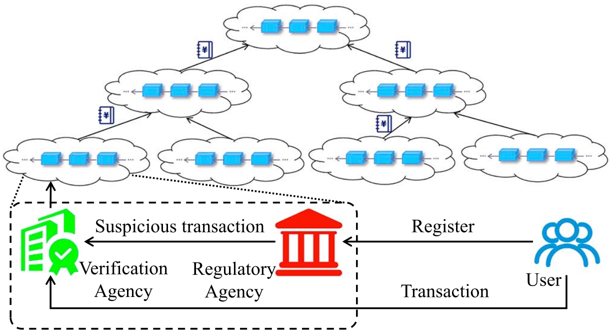


Fig. 3. System model.

IV. System Model and Security Model

A. System Model

MC-CPPT adopts a sharded hierarchical multi-chain architecture [16] to achieve system scalability. Non-leaf blockchain in the system is maintained solely by verification agency, while the leaf blockchain system model mainly comprises the following four entities, as shown in Fig. 3:

- Leaf blockchain: The leaf blockchain acts as a distributed ledger, recording the transactions submitted by validators. All leaf blockchains are based on the UTXO model.
- Verification agency: The verification agency consists of multiple validators. It is responsible for verifying transaction correctness, adding compliant transactions to the leaf blockchain and submitting suspicious transactions to the regulatory agency.
- Regulatory agency: The regulatory agency consists of multiple regulators. It is responsible for negotiating and publicly disclosing parameters in regulatory guidelines and revealing the true identities of senders involved in transactions that exceed limits.

```
MC-CPPT$_{\mathcal{A}}^{AUT}(\lambda)$ :
$pp \leftarrow Setup(1^{\lambda})$
$L \leftarrow \mathcal{A}^{\mathcal{O}}(pp)$
$tx_{Send} \leftarrow \mathcal{A}^{\mathcal{O}}(L)$
$if\ tx_{Send}.sender\ has\ not\ yet\ registered\ then\ return\ 1$
$return\ 0$
```

```
MC-CPPT$_{\mathcal{A}}^{IND}(\lambda)$ :
$pp \leftarrow Setup(1^{\lambda})$
$(L_0, L_1) \leftarrow \mathcal{A}^{\mathcal{O}_0,\mathcal{O}_1}(pp)$
$b \overset{R}{\leftarrow} \{0,1\}$
$Q \overset{R}{\leftarrow} \{Register, Mint, Send, Receive\}$
$a \leftarrow Query_{L_b}(Q)$
$b' \leftarrow \mathcal{A}^{\mathcal{O}_0,\mathcal{O}_1}(L_0, L_1, a)$
$return\ b = b'$
```

```
MC-CPPT$_{\mathcal{A}}^{BAL}(\lambda)$ :
$pp \leftarrow Setup(1^{\lambda})$
$L \leftarrow \mathcal{A}^{\mathcal{O}}(pp)$
$tx_{Send} \leftarrow \mathcal{A}^{\mathcal{O}}(L)$
$if\ tx_{Send}.input \neq tx_{Send}.output\ then\ return\ 1$
$return\ 0$
```

```
MC-CPPT$_{\mathcal{A}}^{UNL}(\lambda)$ :
$pp \leftarrow Setup(1^{\lambda})$
$L \leftarrow \mathcal{A}^{\mathcal{O}}(pp)$
$(tx_{Mint}, tx'_{Mint}) \leftarrow \mathcal{A}^{\mathcal{O}}(L)$
$if\ tx_{Mint}.minter = tx_{Mint}.minter\ then\ return\ 1$
$(tx_{Mint}, tx_{Send}) \leftarrow \mathcal{A}^{\mathcal{O}}(L)$
$if\ tx_{Mint}.minter = tx_{Send}.sender\ then\ return\ 1$
$(tx_{Send}, tx'_{Send}) \leftarrow \mathcal{A}^{\mathcal{O}}(L)$
$if\ tx_{Send}.sender = tx'_{Send}.sender$
    $||tx_{Send}.sender = tx'_{Send}.receiver$
    $||tx_{Send}.receiver = tx'_{Send}.receiver\ then\ return\ 1$
$return\ 0$
```

```
MC-CPPT$_{\mathcal{A}}^{UMA}(\lambda)$ :
$pp \leftarrow Setup(1^{\lambda})$
$L \leftarrow \mathcal{A}^{\mathcal{O}}(pp)$
$tx_{Mint} \leftarrow \mathcal{A}^{\mathcal{O}}(L)$
$if\ \exists tx'_{Mint} \in L\ s.t.\ tx_{Mint}.cmsn = tx'_{Mint}.cmsn$
    $then\ return\ 1$
$tx_{Send} \leftarrow \mathcal{A}^{\mathcal{O}}(L)$
$if\ \exists tx'_{Send} \in L\ s.t.\ tx_{Send}.cmsn = tx'_{Send}.cmsn$
    $then\ return\ 1$
$return\ 0$
```

```
MC-CPPT$_{\mathcal{A}}^{T\text{-}IND}(\lambda)$ :
$pp \leftarrow Setup(1^{\lambda})$
$L \leftarrow \mathcal{A}^{\mathcal{O}}(pp)$
$tx_{Send} \leftarrow \mathcal{A}^{\mathcal{O}}(L)$
$gtype \leftarrow \mathcal{A}^{\mathcal{O}}(tx_{Send})$
$if\ tx_{Send}.type = gtype\ then\ return\ 1$
$return\ 0$
```

```
MC-CPPT$_{\mathcal{A}}^{TRA}(\lambda)$ :
$pp \leftarrow Setup(1^{\lambda})$
$L \leftarrow \mathcal{A}^{\mathcal{O}}(pp)$
$tx_{Send} \leftarrow \mathcal{A}^{\mathcal{O}}(L)$
$if\ \exists tx'_{Mint} \in L\ \&\&\ violates\ regulatory\ guidelines$
    $\&\&\ Trace(tx_{Send}) \neq tx_{Send}.sender\ then\ return\ 1$
$return\ 0$
```

Fig. 4. Security experiments.

- Users: Users are entities requiring transactions.

We assume users are untrustworthy; malicious users may attempt to spend assets that do not belong to them, spend the same asset twice or even more, or exploit the anonymity of blockchain for illegal fund transfers. Furthermore, external attackers may infer sensitive information such as users' real identities, transaction amounts, account balances, transaction habits, etc., by observing and analyzing transactions on the blockchain. They may also exploit publicly available transaction data to link two transactions together. Due to the traceability and immutability of blockchain, it ensures that the entire blockchain is trustworthy to external observers. In order to simplify the design of the scheme, we assume that the verification agency and regulatory agency are honest and curious.

B. Security Model

As a blockchain transaction scheme for conditional privacy preservation, MC-CPPT needs to satisfy authenticity, indistinguishability, balance, unlinkability, non-malleability, and traceability. These properties are defined as follows, with experiments shown in Fig. 4, including challenger $\mathcal{C}$, adversary $\mathcal{A}$, and oracle $\mathcal{O}$, where $\mathcal{A}$ is a polynomial-time adversary.

- Authenticity of the user: A user is considered authentic if transactions submitted by users not registered with regulatory agency cannot be published on the ledger even if they are valid and compliant with regulatory guidelines. If for every probabilistic polynomial-time adversary $\mathcal{A}$ and sufficiently large λ, we have $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{AUT}(\lambda) = 1] \leq \frac{1}{2} + \mathrm{negl}(\lambda)$, then the user is considered authentic (Where $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{AUT}(\lambda) = 1]$ is the probability of $\mathcal{A}$ winning in the AUT experiment).
- Indistinguishability of the ledger: The ledger is considered indistinguishable if it does not reveal any information other than publicly disclosed information, such as user identities, transaction counts, and transaction amounts. The ledger is indistinguishable if for every probabilistic polynomial-time adversary $\mathcal{A}$ and

sufficiently large λ, we have $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{IND}(\lambda)=1]\leq\frac{1}{2}+\mathrm{negl}(\lambda)$ (Where $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{IND}(\lambda)=1]$ is the probability of $\mathcal{A}$ winning in the *IND* experiment).

- Balance of transactions: A transaction is considered balanced if the total input amount equals the total output amount. A transaction is balanced if for every probabilistic polynomial-time adversary $\mathcal{A}$ and sufficiently large λ, we have $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{BLA}(\lambda)=1]\leq\frac{1}{2}+\mathrm{negl}(\lambda)$ (where $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{BLA}(\lambda)=1]$ is the probability of $\mathcal{A}$ winning in the *BLA* experiment).
- Unlinkability of transactions: A transaction is considered unlinkable if the publicly disclosed information in the transaction cannot be used to link two transactions. A transaction is unlinkable if for every probabilistic polynomial-time adversary $\mathcal{A}$ and sufficiently large λ, we have $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{UNL}(\lambda)=1]\leq\frac{1}{2}+\mathrm{negl}(\lambda)$ (where $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{UNL}(\lambda)=1]$ is the probability of $\mathcal{A}$ winning in the *UNL* experiment).
- Non-malleability of transactions: A transaction is considered non-malleable if the adversary cannot generate a new transaction with the same disclosed data (i.e., coin serial numbers) as a previous transaction but different from it. A transaction is non-malleable if for every probabilistic polynomial-time adversary $\mathcal{A}$ and sufficiently large λ, we have $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{NMA}(\lambda)=1]\leq\frac{1}{2}+\mathrm{negl}(\lambda)$ (where $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{NMA}(\lambda)=1]$ is the probability of $\mathcal{A}$ winning in the *NMA* experiment).
- Indistinguishability of transaction types: The transaction types are considered indistinguishable if the adversary cannot distinguish whether a transaction belongs to on-chain transactions or cross-chain transactions. Transaction types are indistinguishable if for every probabilistic polynomial-time adversary $\mathcal{A}$ and sufficiently large λ, we have $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{T\_IND}(\lambda)=1]\leq\frac{1}{2}+\mathrm{negl}(\lambda)$ (where $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{T\_IND}(\lambda)=1]$ is the probability of $\mathcal{A}$ winning in the *T_IND* experiment).
- Traceability of transactions: A transaction is traceable if regulatory agency can correctly recover the real identity of the sender for transaction that violate regulatory guidelines. A transaction is traceable if for every probabilistic polynomial-time adversary $\mathcal{A}$ and sufficiently large λ, we have $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{TRA}(\lambda)=1]\leq\frac{1}{2}+\mathrm{negl}(\lambda)$ (where $\Pr[\mathrm{MC-CPPT}_{\mathcal{A}}^{TRA}(\lambda)=1]$ is the probability of $\mathcal{A}$ winning in the *TRA* experiment).

V. System Design

A. Overview of MC-CPPT

The system process of MC-CPPT is divided into 6 stages: Setup, Register, Mint, Send, Receive, and Trace. During the Register stage, users must apply to the regulatory agency for registration and authentication. To conduct transactions using the universal cryptocurrency, users must mint coins in the Mint stage, generate tx_{Mint}, and submit them to the verification agency. In the Send stage, users can transfer received or minted coins in the Mint stage to others, generating tx_{Send} that include proofs of transaction validity and compliance. The verification agency is required to forward suspicious transactions to the regulatory agency for traceability. The receivers of the transaction can obtain coins in the Receive stage. During the Trace stage, the regulatory agency can recover the true identity of users who registered in the Register stage.

Due to the UTXO model used by MC-CPPT, where the input coins in tx_{Send} are outputs from another tx_{Send}, attackers may use this feature to link these transactions. The MT Proof can hide the input coins in tx_{Send}. Considering MC-CPPT operates within a multi-chain framework, the root of the MT reveals the specific leaf blockchain, which allows attackers to distinguish between intra-chain transactions and cross-chain transactions. Therefore, we employ EMT Proof [28] instead of MT Proof to achieve indistinguishability. To prevent external observers from deducing the specific coin identity through the path in EMT Proof, we use zero-knowledge proof to conceal the path. To prevent double-spending, each coin is assigned a serial number, which users must reveal when sending to prove the coin hasn't been used before. Since the denomination of coins is unknown, malicious actors may exploit this loophole for money laundering. Thus, we introduce regulatory guidelines on transaction amounts and frequencies. Specifically, the total amount and total frequency that the user can send within a certain period of time are both limited. Each user has a counter to accumulate transaction amounts and frequencies. To prevent attackers from correlating tx_{Send} using the counter, we use zero-knowledge proof and MT Proof to hide the counter.

B. Details of MC-CPPT

Each leaf blockchain is equipped with n regulators, and user identity authentication or tracing requires the participation of at least t regulators. The processing of transactions on Leaf blockchain is parallel and independent. The proposed scheme mainly consists of six stages: Setup, Register, Mint, Send, Receive, and Trace. The following will provide a detailed explanation of the system from the perspective of one leaf blockchain.

Setup: Input security parameter λ, the root blockchain selects a q-order elliptic curve group G, where points on the elliptic curve are defined over a finite field F_p. g is a generator of the group G. Define the collision-resistant hash function $H:\{0,1\}^*\rightarrow\{0,1\}^\lambda$. Define the commitment scheme COM and pseudo-random function PRF. We use the hash function to instantiate the commitment scheme and pseudo-random function. Construct circuits $C_{Register}$, C_{Mint}, C_{Send_com}, C_{Send_exc} for zero-knowledge proof, where C_{Send_com} is for transactions that comply with regulatory guidelines, and C_{Send_exc} is for transactions that do not comply with regulatory guidelines. Employ $KeyGen(1_\lambda,C)$ to compute the proving keys and verification keys $(PK_{\mathbb{Z}_{Register}},VK_{\mathbb{Z}_{Register}})$, $(PK_{\mathbb{Z}_{Mint}},VK_{\mathbb{Z}_{Mint}})$, $(PK_{\mathbb{Z}_{Send\_com}},VK_{\mathbb{Z}_{Send\_com}})$, $(PK_{\mathbb{Z}_{Send\_exc}},VK_{\mathbb{Z}_{Send\_exc}})$.

Leaf Blockchain uses DKG technology to build and distribute keys within its jurisdiction, as follows. Suppose the threshold of regulators for a leaf blockchain is (t,n), where each regulator Reg_i(where $i\in\{1,n\}$) has a unique identity ID_i. Reg_i chooses a

$t-1$ degree polynomial $f_i(x) = a_{i0} + a_{i1}x + \cdots + a_{i(t-1)}x^{t-1}$ mod$q$, computes $f_i(ID_i)$ and $f_i(ID_j)$, and sends $f_i(ID_j)$ to Reg_j. Reg_i broadcasts $\{b_{iw} = g^{a_{iw}}\}(w = 0, 1, ..., t-1)$ among the regulators of this leaf blockchain. When regulator Reg_j(where $j \in \{1, n\}$ and $j \neq i$) receives $\{b_{iw}\}$ sent by the other $n-1$ regulators, it can determine the validity of $f_i(ID_j)$ by verifying whether $g^{f_i(ID_j)} = \prod_{w=0}^{t-1} b_{iw}^{ID_j^w}$ holds. If it does not hold, reject $f_i(ID_j)$ and notify Reg_i to recalculate and resend it. When regulator Reg_i receives $f_j(ID_i)$ sent by the other $n-1$ regulators, it can calculate its personal regulatory private key $sk_i = \sum_{j=1}^{n} f_j(ID_i)modq$. Then, Reg_i calculates $pk_i = g^{sk_i}$ based on sk_i and broadcasts pk_i among the regulators of this leaf blockchain. According to Lagrange interpolation, the regulators of this leaf blockchain can calculate the regulatory public key $pk = g^{sk} = g^{\sum_{i=1}^{n} a_{i0}} = \prod_{i=1}^{t} pk_i^{L(i)}modp$ (where $L(i) = \prod_{j=1, j\neq i}^{t} \frac{ID_j}{ID_j - ID_i}$), and publicly announce the regulatory public key pk to the users of this leaf blockchain. Additionally, according to regulatory guidelines, the regulators of this leaf blockchain also need to jointly determine the maximum total transaction amount N_{limit} and total frequency V_{limit} for anonymous transactions that users can conduct within a period T, and publicly announce T, N_{limit}, and V_{limit} to the users of this leaf blockchain.

Register: The user U with identity of id selects $rt \xleftarrow{R} Z_q^*$, computes the initial counter serial number $ctsn = PRF(id|||rt)$ and the commitment of the initial counter ctm_0. Since the counter commitment format is as follows: $ctm = COM(id\|ctsn\|V_0\|N_0)$, where V_0 and N_0 represent the accumulated transaction amount and transaction frequency of the counter, respectively, and the initial counter has not been used to accumulate transaction amounts or frequencies, thus $ctm_0 = COM(id\|ctsn\|0\|0)$. At the same time, the user also needs to generate a proof $\pi_{Register}$ using algorithm $Prove(PK_{\mathbb{Z}_{Register}}, x, \omega)$, where $x = (id, ctm_0)$ and $\omega = (rt, ctsn)$. $\pi_{Register}$ proves that

- $ctsn = PRF(id\|rt)$
- $ctm_0 = COM(id\|ctsn\|0\|0)$

U will send $x, \pi_{Register}$ to any of the regulators of the respective leaf blockchain. Upon receiving the information from the user, the regulator Reg_j will first use algorithm $Verify(VK_{\mathbb{Z}_{Register}}, x, \pi_{Register})$ to verify the correctness of $\pi_{Register}$. If it is correct, the process will continue to the next step; if it is incorrect, the regulator will refuse to authenticate the user.

Reg_j publicly disclose ctm_0 to be signed at the regulatory agency. Other regulators such as Reg_i selects $d_i \xleftarrow{R} Z_q^*$, calculates $D_i = g^{d_i}$, and send D_i to Reg_j. After Reg_j receives $t-1$ D_i, the signature group $\mathbb{RT}$ forms. Reg_j need to calculate $e_0 = H(ctm_0)$, $(X_0, Y_0) = D = \prod_{i=1}^{t} D_i$ and $l = X_0 - e_0$mod$q$. Reg_i (where $i \in \mathbb{RT}$) calculates $o_i = L(i)sk_i l + d_i$mod$q$ and publicly announces o_i within the group. Reg_j aggregates o_i to form the signature $\sigma_{ctm} = (l, o) = (l, \sum_{i=1}^{t} o_i)$. Similarly, the signature group can obtain σ_{id} by signing id. (ctm_0, σ_{ctm}) will be sent to the validators, who will add it to the MT T_{ctm} with the counter commitment as the leaf node. At the same time, the signature of the user identity σ_{id} and the initial counter σ_{ctm} will be sent to U.

After receiving σ_{id} and σ_{ctm}, the user can verify its correctness. Taking verification of $\sigma_{ctm} = (l, o)$ as an example, the user calculates $e_0' = H(ctm_0)$, $X_0' = l + e_0'$ mod q, and using elliptic curve calculations to obtain the corresponding Y_0', then checking if $(X_0', Y_0') = g^o / pk^l$ holds true. If it does, the signature is valid; otherwise, the user notifies regulator Reg_j to recalculate.

Mint: If U needs to mint a coin with a value of v, they must first pay an equivalent amount of Bitcoin or other cryptocurrency to the backing escrow pool. Then, U selects $ra, rs \xleftarrow{R} Z_q^*$, computes a one-time address $addr = PRF(id\|ra)$, the coin's serial number $cmsn = PRF(addr\|rs)$ and the coin commitment $cm = COM(addr\|cmsn\|rs\|v)$, and store the coin in the wallet in the following format $c = (cm, addr, cmsn, rs, v)$. Additionally, the user also needs to generate a proof π_{Mint} using algorithm $Prove(PK_{\mathbb{Z}_{Mint}}, x, \omega)$, where $x = (v, cm)$ and $\omega = (id, addr, ra, rs, cmsn)$. π_{Mint} proves that

- $addr = PRF(id\|ra)$
- $cmsn = PRF(addr\|rs)$
- $cm = COM(addr\|cmsn\|rs\|v)$

Subsequently, U submits the $Mint$ transaction $tx_{Mint} = (x, \pi_{Mint})$ to the validators.

Upon receiving tx_{Mint}, the validators first use algorithm $Verify(VK_{\mathbb{Z}_{Mint}}, x, \pi_{Mint})$ to verify the correctness of π_{Mint}. After successful verification, the validators add the coin commitment cm to the T_{coin}^{leaf} of this leaf blockchain (where T_{coin}^{leaf} is the Merkle tree with coin commitment as leaf nodes) and sends the root of T_{coin}^{leaf} to the validators of the parent blockchain. Since the leaf blockchain processes transactions in parallel, the parent blockchain collects a large number of similar roots in a short period. These roots are used as leaf nodes to construct a new Merkle tree, obtaining a higher-level root, which is then sent to a higher-level parent blockchain. The coin commitments generated by the leaf blockchains are passed layer by layer in this manner until they reach the root blockchain. Ultimately, the root of the Merkle tree constructed by the root blockchain is obtained by hashing all the cm generated during the period.

Send: If a user wishes to transfer cryptocurrencies to another user, the sender S needs to create a $Send$ transaction. Without loss of generality, let's assume that both the number of input coins and output coins in the $Send$ transaction are 2. For $k \in \{1, 2\}$, the receiver R_k who is identified as id_k selects $ra_k^{new} \overset{R}{\leftarrow} Z_q^*$, compute a one-time address $addr_k^{new} = PRF(id_k\|ra_k^{new})$, and selects an encryption secret key $esk_k \overset{R}{\leftarrow} Z_q^*$ to generate an encryption public key $epk_k = g^{esk_k}$, sending $addr_k^{new}$ and epk_k to S. To mint the new coin c_k, the sender first selects $rs_k^{new} \overset{R}{\leftarrow} Z_q^*$, utilizes R_k's $addr_k^{new}$ to calculate the coin's serial number $cmsn_k^{new} = PRF(addr_k^{new}\|rs_k^{new})$ and the coin commitment $cm_k^{new} = COM(addr_k^{new}\|cmsn_k^{new}|rs_k^{new}\|v_k^{new})$. The new coin format is as follows: $c_k^{new} = (cm_k^{new}, addr_k^{new}, cmsn_k^{new}, rs_k^{new}, v_k^{new})$. S needs to encrypt $(cmsn_k^{new}\|rs_k^{new}\|v_k^{new})$ using AES algorithm to obtain Mh_k. Meanwhile, the sender encrypts the AES key mh_k using R_k's encryption public key epk_k in the following manner: choosing $rh_k \overset{R}{\leftarrow} Z_q^*$, computing $ch_k = (ch_{k,1}, ch_{k,2}) = (g^{rh_k}, mh_k * epk_k^{rh_k})$ and sending (Mh_k, ch_k) to R_k.

To calculate the user's transaction amount V and transaction frequency N over a period, the user needs to add the total amount involved in this transaction v^{old} to V^{old} and increase the transaction frequency N^{old} by 1, then record the new amount V^{new} and N^{new} into the new counter: select $rt^{new} \overset{R}{\leftarrow} Z_q^*$, compute $ctsn^{new} = PRF(id \| rt^{new})$, $ctm^{new} = COM(id \| ctsn^{new} \| V^{new} \| N^{new})$.

To ensure that the content of the payment transaction is not arbitrarily tampered with by malicious attackers, the sender also needs to sign the transaction data. Suppose the transaction data to be signed is mu. Let $mu = (root_{ctm} \| root_1 \| root_2 \| cmsn_1^{old} \| cmsn_2^{old} \| cm_1^{new} \| cm_2^{new} \| v_{pub} \| ctsn^{old} \| ctm^{new} \| info)$, where $root_{ctm}$ is the root of T_{ctm}, $root_1$, $root_2$ are the roots of EMT T_{coin} where cm_1^{old}, cm_2^{old} reside, $cmsn_1^{old}$,$cmsn_2^{old}$ is the serial number of c_1^{old},c_2^{old}, cm_1^{new},cm_2^{new} is the commitment of c_1^{new},c_2^{new}, v_{pub} is the amount to restore the coin to the original currency like Bitcoin, which is public, and the user can also set $v_{pub} = 0$, $info$ is the address to restore the original currency, such as a Bitcoin wallet public key, $ctsn_{old}$ is the serial number of the counter, ctm_{new} is the commitment of the counter. The sender first selects $sk_{sig}, ru \overset{R}{\leftarrow} Z_q^*$, computes the signature public key is $pk_{sig} = g^{sk_{sig}}$, next computes $e = H(mu)$, $(X, Y) = g^{ru}$, let $\rho = X \bmod p$, if $\rho = 0$ then reselect ru, computes $\tau = rk^{-1}(e + sk_{sig} * \rho) \bmod p$, and the signature $\sigma_{Send} = (\rho, \tau)$ is obtained. The sender computes $h_{sig} = H(pk_{sig})$, and $h_1 = PRF(addr_1^{old} \| h_{sig})$, $h_2 = PRF(addr_2^{old} \| h_{sig})$, these two values bind h_{sig} with the one-time addresses $addr_1^{old}$, $addr_2^{old}$. If the transaction amount and frequency do not exceed the limit, the user need to generate a proof π_{Send} using algorithm $Prove(PK_{\mathbb{Z}_{Send\_com}}, x, \omega)$, where $x = (root_1, root_2, cmsn_1^{old}, cmsn_2^{old}, cm_1^{new}, cm_2^{new}, v_{pub}, h_{sig}, h_1, h_2, root_{ctm}, ctsn^{old}, ctm^{new})$ and $\omega = (path_1, path_2, id, ra_1, ra_2, addr_1^{old}, addr_2^{old}, rs_1^{old}, rs_2^{old}, v_1^{old}, v_2^{old}, cm_1^{old}, cm_2^{old}, addr_1^{new}, addr_2^{new}, rs_1^{new}, rs_2^{new}, cmsn_1^{new}, cmsn_2^{new}, v_1^{new}, v_2^{new}, v^{old}, v^{new}, path_{ctm}, V^{old}, N^{old}, V^{new}, N^{new}, rt^{old}, ctm^{old}, rt^{new}, ctsn^{new})$. π_{Send} proves that

- $root_1 = EMT(path_1)$
- $root_2 = EMT(path_2)$
- $addr_1^{old} = PRF(id \| ra_1)$
- $addr_2^{old} = PRF(id \| ra_2)$
- $cmsn_1^{old} = PRF(addr_1^{old} \| rs_1^{old})$
- $cmsn_2^{old} = PRF(addr_2^{old} \| rs_2^{old})$
- $cm_1^{old} = COM(addr_1^{old} \| cmsn_1^{old} \| rs_1^{old} \| v_1^{old})$
- $cm_2^{old} = COM(addr_2^{old} \| cmsn_2^{old} \| rs_2^{old} \| v_2^{old})$
- $cmsn_1^{new} = PRF(addr_1^{new} \| rs_1^{new})$
- $cmsn_2^{new} = PRF(addr_2^{new} \| rs_2^{new})$
- $cm_1^{new} = COM(addr_1^{new} \| cmsn_1^{new} \| rs_1^{new} \| v_1^{new})$
- $cm_2^{new} = COM(addr_2^{new} \| cmsn_2^{new} \| rs_2^{new} \| v_2^{new})$
- $v^{old} = v_1^{old} + v_2^{old}$
- $v^{new} = v_1^{new} + v_2^{new}$
- $v^{old} = v^{new} + v_{pub}$
- $h_1 = PRF(addr_1^{old} \| h_{sig})$
- $h_2 = PRF(addr_2^{old} \| h_{sig})$
- $root_{ctm} = MT(path_{ctm})$
- $V^{new} = V^{old} + v^{old}$
- $N^{new} = N^{old} + 1$
- $ctsn^{old} = PRF(id \| rt^{old})$
- $ctm^{old} = COM(id \| ctsn^{old} \| V^{old} \| N^{old})$
- $ctsn^{new} = PRF(id \| rt^{new})$
- $ctm^{new} = COM(id \| ctsn^{new} \| V^{new} \| N^{new})$
- $V^{new} \in [0, V^{limit}]$
- $N^{new} \in [0, N^{limit}]$

S submits the $Send$ transaction $tx_{Send} = (x, \pi_{Send}, pk_{sig}, \sigma_{Send}, info)$ to the validators if tx_{Send} is compliant. If the transaction amount or frequency exceeds the limit, S does not need to prove the last two points of the proof above. Instead, the user need to include the identity information for investigation. To protect identity privacy, the information needs to be encrypted using the regulatory public key pk. The encryption method is as follows: encrypt $(id \| \sigma_{id})$ using AES algorithm to obtain Md, select $rd \overset{R}{\leftarrow} Z_q^*$, calculate $cd = (cd_1, cd_2) = (g^{rd}, md * pk^{rd})$ to encrypt the AES key md, add Md, cd to tx_{Send}. To prevent identity theft, S is required to additionally prove in the zero-knowledge proof that the encrypted id matches the id that generated the $addr^{old}$, i.e. $tid = H(id \| \sigma_{id})$. In this case, S generate π_{Send} using algorithm $Prove(PK_{\mathbb{Z}_{Send\_exc}}, x, \omega)$. At the same time, the user needs to add tid to x and submits the $Send$ transaction $tx_{Send} = (Md, cd, x, \pi_{Send}, pk_{sig}, \sigma_{Send}, info)$ to the validators.

When the validators receive tx_{Send}, they first check whether it contains (Md, cd). If it does, validators use algorithm $Verify(VK_{\mathbb{Z}_{Send\_exc}}, x, \pi_{Send})$ to verify the correctness of π_{Send}; If not, they use algorithm $Verify(VK_{\mathbb{Z}_{Send\_com}}, x, \pi_{Send})$ to verify the correctness of π_{Send}. If the verification fails, the transaction request will be rejected. Otherwise, the validators verify the correctness of $\sigma_{Send} = (\rho, \tau)$. The validators calculate $e' = H(root_{ctm} \| root_1 \| root_2 \| cmsn_1^{old} \| cmsn_2^{old} \| cm_1^{new} \| cm_2^{new} \| v_{pub} \| ctsn^{old} \| ctm^{new} \| info)$, $\varsigma = \tau^{-1} \bmod p$, $(X', Y') = \chi = g^{e' \cdot \varsigma} \cdot pk_{sig}^{\rho \cdot \varsigma}$, check if X' equals ρ. If they are equal, it indicates that the signature is correct. If the transaction complies with regulatory guidelines, meaning that tx_{Send} does not contain (Md, cd), the validators add ctm_{new} to T_{ctm}, add $ctsn_{old}$ to the set of the counter serial numbers (indicating that the counter has been used), add cm_k^{new} to T_{coin}^{leaf} and add $cmsn_k^{old}$ to the set of coin serial numbers (indicating that the coin has been used). Similar to cm from tx_{Mint}, cm from tx_{Send} will be eventually added to T_{coin}^{root} of the root blockchain through layer-by-layer propagation. If the transaction does not comply with regulatory guidelines, meaning that tx_{Send} contains (Md, cd), the validators send tx_{Send} to the regulatory agency to trace its real identity.

Receive: After receiving (Mh, ch) sent by S, R first decrypts ch using the encrypted secret key esk, computes $mh' = ch_2 * ch_1^{-esk}$, decrypts Mh with the AES key mh' to obtain $(cmsn \| rs \| v)$. Then R receives the coin in the following format: $c = (cm, addr, cmsn, rs, v)$.

Trace: In the case of tracing, any arbitrary t regulators from the leaf blockchain can conduct identity tracing on transactions that violate regulatory guidelines. These t regulators form a regulatory group. Each member Reg_i of the regulatory group computes $ss_i = cd_1^{-sk_i * L(i)}$ and broadcasts ss_i within the regulatory group, enabling members to compute $md' = cd_2 * cd_1^{-sk}$ and decrypt Md with the AES key md' to obtain $(id\|\sigma_{id})$. At the same time, the regulatory is able to judge whether the user is a registered and legitimate user by verifying the validity of σ_{id}. In order to ensure the unlinkability of transactions, regulatory agency also need to generate a new signature σ_{id} for the sender if the user is ultimately proven not to have participated in illegal fund transfers.

VI. Security Proof

In this section, we will prove the security of MC-CPPT, demonstrating that it satisfies the following properties: user authenticity, ledger indistinguishability, ledger balance, transaction unlinkability, transaction non-malleability, transaction type indistinguishability and transaction traceability.

Theorem 1: If the signature scheme is existentially unforgeable under chosen message attacks (EUF-CMA secure), and the hash function is collision-resistant, then MC-CPPT can achieve user authenticity.

Proof: Since the validator can verify the *ctm* during the validation of the *Send* transaction, it ensures that the user is a legitimate user registered with the regulatory agency in its jurisdiction. Regarding the types of *ctm*, we categorize them for discussion. For the initial counter ctm_0, it has a signature σ_{ctm} authenticated by the regulatory agency. If a forged signature that can successfully deceive the validator, it contradicts the existence of the signature scheme's unforgeability. As for non-initial counters, they are constructed based on the previous counter. Similarly, the previous counter is also constructed based on the counter before it, and this backward induction continues until the initial counter. Hence, it can be categorized as the first case.

Theorem 2: If zk-SNARK is zero-knowledge, the encryption scheme is indistinguishable under chosen plaintext attacks (IND-CPA secure), the commitment scheme is statistically hiding, and the pseudorandom function is indistinguishable from a random function, then MC-CPPT can achieve ledger indistinguishability.

Proof: We need to ensure that the ledger does not reveal any information other than the publicly disclosed information, such as user identities, transaction frequencies, and transaction amounts. We construct an experiment sequence $\{E_{real}, E_1, E_2, E_3, E_{sim}\}$, where $\mathcal{C}$ modifies the original experiment E_{real} in the security model. We will demonstrate that the difference in advantage between $\mathcal{A}$ in E_{real} and E_{sim} can be negligible.

Firstly, $\mathcal{C}$ sends the public parameters pp to $\mathcal{A}$ and initializes two oracles $\mathcal{O}_0$ and $\mathcal{O}_1$. In experiment E_1, after sampling $b \in \{0, 1\}$, $\mathcal{C}$ uses a simulator $\mathcal{S}$ to modify the key generation and proof generation of zk-SNARK in E_{real}. $\mathcal{C}$ does not call the key generation algorithm $KeyGen(1^\lambda, \mathcal{C})$ of zk-SNARK, but uses $\mathcal{S}$ to generate proof keys $PK_{\mathbb{Z}}$, verification keys $VK_{\mathbb{Z}}$, and *trapdoor*. For the proof algorithm in zk-SNARK, $\mathcal{S}$ uses the public input x and *trapdoor* to generate proof π without using private input ω. Since zk-SNARK is zero-knowledge, the distributions of proofs generated by $\mathcal{S}$ and the proof algorithm $Prove(PK_{\mathbb{Z}}, x, \omega)$ are the same. Therefore, the difference between E_1 and E_{real} is 0.

In experiment E_2, $\mathcal{C}$ replaces the ciphertext generated by the receiver's public key with ciphertext of random strings. Specifically, for each oracle, it first replaces the address *addr* with randomly strings, and then replaces the plaintext to be encrypted with randomly chosen strings from the plaintext space. Let q_S be the total number of *Send* queries sent by $\mathcal{A}$. If $\mathcal{A}$ has an advantage ϵ_1 in the IND-CPA experiment of the encryption scheme, then the advantage difference between experiments E_1 and E_2 is at most $4 \cdot q_S \cdot \epsilon_1$.

In experiment E_3, $\mathcal{C}$ modifies E_2 by replacing the generated pseudo-random function values with random strings. Specifically, the old coin serial number $cmsn_k^{old}$ in the *Send* query will be replaced with random strings of the same length. Let q_P be the total number of times $\mathcal{A}$ uses *PRF*. If $\mathcal{A}$ has an advantage ϵ_2 in distinguishing pseudo-random functions from random functions, then the advantage difference between experiments E_2 and E_3 is at most $q_P \cdot \epsilon_2$.

In experiment E_{sim}, $\mathcal{C}$ modifies E_3 by replacing the generated coin commitment with commitments of random strings. Specifically, $addr||cmsn||cm||v$ in the *Mint* query will be replaced with commitments of random strings of the same length, and cm_k^{new} in the *Send* query will also be replaced with commitments of random strings of the same length. Let q_M be the total number of Mint queries sent by $\mathcal{A}$. If $\mathcal{A}$ has an advantage ϵ_3 in the hiding experiment of the commitment scheme, then the advantage difference between experiments E_3 and E_sim is at most $(q_M + 4 \cdot q_P) \cdot \epsilon_3$.

Since in experiment E_{sim}, the responses and ledgers shown to $\mathcal{A}$ are independent to b, the advantage of $\mathcal{A}$ in experiment E_{sim} is 0. Therefore, the advantage of $\mathcal{A}$ in E_{real} is $Adv_{\mathcal{A}} \leq 4 \cdot q_P \cdot \epsilon_1 + q_P \cdot \epsilon_2 + (q_M + 4 \cdot q_P) \cdot \epsilon_3$.

Theorem 3: If the commitment scheme is computationally binding, the hash function is collision-resistant, zk-SNARK is sound, and the pseudo-random function is collision-resistant, then MC-CPPT can achieve ledger balance.

Proof: The balance property is proven by modifying the defined balancing experiment in the security model, allowing $\mathcal{C}$ to obtain an augmented ledger where, for each payment transaction, in addition to the original parameters, there is also a private input ω. However, for $\mathcal{A}$, the view of the ledger remains unchanged. We represent the augmented ledger as $(L, \vec{a})$, where a_i is the private input x_i of the i-th *Send* transaction. For transactions generated by $\mathcal{O}$, $\mathcal{C}$ can obtain the private input by querying $\mathcal{O}$. For *Send* transactions created and inserted into the ledger by $\mathcal{A}$, $\mathcal{C}$ can use the knowledge extractor of zk-SNARK to obtain the corresponding private input.

The ledger $(L, \vec{a})$ satisfies balancing if the following conditions are met:

- For any tx_{Send}, the coin commitment cm_k^{old} must correspond to the output parameter of a previous tx_{Mint} or tx_{Send}.
- There cannot be one coin commitment with two different opening values.
- The total input amount in any tx_{Send} must be equal to the total output amount.
- If a cm_k^{old} is an output parameter of tx_{Mind}, then the value v_k^{old} of the coin must be equal to the value of the cryptocurrency deposited by the user into the backing escrow pool. If a cm_k^{old} is an output parameter of tx_{Send}', then the

value v_k^{old} of the coin must be equal to the value $v_k^{new\prime}$ corresponding to cm_k^{new} in $tx_{Send}{}'$.

- The coins spent by $\mathcal{A}$ must either be minted by $\mathcal{A}$ itself or received by $\mathcal{A}$.

If the ledger L is unbalanced, it implies that $\mathcal{A}$ violates at least one of the above conditions with a non-negligible probability. We analyze each condition as follows and point out the contradictions with the assumptions:

If condition 1 is not met, then cm_k^{old} does not exist on T_{coin}, and consequently, there is no verification path $path_k$. If the authentication path passes verification, it implies a collision in the hash function, contradicting its collision resistance property.

If condition 2 is not met, it means $cm_k^{old} = cm_k^{old'}$ but $cmsn_k^{old} \neq cmsn_k^{old'}$, contradicting the binding property of the commitment scheme.

If condition 3 is not met, it contradicts the soundness of zk-SNARK.

If condition 4 is not met, it means $cm_k^{old} = cm_k^{old'}$ but $v_k^{old} \neq v_k^{old'}$, contradicting the binding property of the commitment scheme.

If condition 5 is not met, it means there exist $addr_k^{old'}$ and ra' such that $addr_k^{old} = PRF(id\|ra')$, contradicting the collision resistance property of the pseudo-random function.

Theorem 4: If zk-SNARK is zero-knowledge, then MC-CPPT can achieve transaction unlinkability.

Proof: Transaction unlinkability relies on the same parameter appearing in two different transactions. For example, the $cmsn$ in a tx_{Mint} is equal to $cmsn_k^{old}$ in a tx_{Send}, the cm_k^{old} in a tx_{Send} is equal to the cm in a tx_{Mint}, the cm_k^{old} in a tx_{Send} is equal to the cm_k^{new} in a $tx_{Send}{}'$, the $cmsn_k^{new}$ in a tx_{Send} is equal to the $cmsn_k^{old}$ in a $tx_{Send}{}'$, and the ctm^{old} in a tx_{Send} is equal to the ctm^{new} in a $tx_{Send}{}'$. However, $cmsn$ in tx_{Mint} is a private input of π_{Mint}, and cm_k^{old}, $cmsn_k^{new}$, ctm^{old} in tx_{Send} are private inputs of π_{Send}, which contradicts the zero-knowledge property of zk-SNARK.

Theorem 5: If the signature scheme is unpredictable, the hash function is collision-resistant, the pseudo-random function is collision-resistant, then MC-CPPT can achieve transaction non-malleability.

Proof: $\mathcal{A}$ may attempt to alter the *info* field of tx_{Send}, changing the original address to their own wallet public key in order to gain benefits. Below, we categorize and discuss attack scenarios. If $\mathcal{A}$ does not change the original signature, they would need to find $info'$ such that the signature $\sigma_{Send}{}'$ of $(root_{ctm}\|root_1\|root_2 \|cmsn_1^{old}\|cmsn_2^{old}\|cm_1^{new} \ \|cm_2^{new}\|v_{pub}\|ctsn^{old} \ \|ctm^{new}\|info)'$ is equal to the original σ_{Send}. This contradicts the unpredictability of the signature scheme and the collision resistance of the hash function. If $\mathcal{A}$ uses their own signature key to sign, they would need to find $(pk_{sig}{}', sk_{sig}{}')$ such that satisfies $h_k' = h_k$. This contradicts the collision resistance of the pseudo-random function.

Theorem 6: If zk-SNARK is zero-knowledge and the commitment scheme is statistically hiding, then MC-CPPT can achieve indistinguishability of transaction types.

Proof: Because the Merkle tree storing coin commitments is global, regardless of which leaf blockchain's Merkle tree T_{coin}^{leaf} ultimately integrates into the final block header hash $hash_{root}$ of the root blockchain, the difference in $root_k$ can only indicate that the coin was minted or sent at different times. Therefore, $\mathcal{A}$ cannot distinguish whether a given tx_{Send} is an intra-chain transaction or a cross-chain transaction based on $root_k$. If zk-SNARK is zero-knowledge, the path $path_k$ of cm_k^{old} in T_{coin}^{Send} will not be revealed in π_{Send} of tx_{Send}. If the commitment scheme is statistically hiding, the difference in R_k s' addresses $addr_k$ will not affect the probability distribution of cm_k^{new}, thereby preventing $\mathcal{A}$ from using cm_k^{new} to determine the transaction type.

Theorem 7: If the signature scheme is existentially unforgeable under chosen message attacks (EUF-CMA secure) and zero-knowledge proofs are reliable, then MC-CPPT can achieve transaction traceability.

Proof: If a transaction violates regulatory guidelines, $\mathcal{A}$ cannot generate a valid range proof, as doing so would contradict the reliability of the range proof scheme. Therefore, when a transaction violates regulatory guidelines, the user needs to include their identity certificate encrypted within the transaction. Regulatory agency can decrypt it using their regulatory private key to recover the user's identity certificate. Additionally, since the identity certificate is signed by the jurisdiction's regulatory agency, by verifying the signature's correctness, regulatory agency can confirm that the user is legitimately registered. If $\mathcal{A}$ can forge signatures and pass verification, it contradicts the unforgeability of the signature scheme.

VII. Performance Evaluation

A. Functional Comparison

We conducted a functionality comparison with several relevant schemes and summarize the comparison results in Table I. We comprehensively analyzed the schemes from different functional aspects, including ledger indistinguishability, non-malleability, unlinkability, transaction type indistinguishability, scalability and traceability. Due to some studies only involving single chain transactions, the indistinguishability between intra-chain and cross-chain transactions, as well as the scalability of blockchain, are not within the scope of discussion.

B. Performance Analyses

To evaluate the performance of MC-CPPT, we conducted an experiment on a computer with an Intel(R) Core(TM) i7-11700 @ 2.50GHz and 16GB RAM. We used the circomlib library as the zero-knowledge proof library, the BLS12381 curve as the elliptic curve, and the SHA-512 hash function as the secure hash function. To better analyze the computational overhead of the proposed scheme, we listed the involved notations in Table II. We conducted a theoretical analysis of computational overhead, as shown in the Table III.

We tested the computational overhead and communication overhead of stages in MC-CPPT. We set the threshold (t, n) to $(7, 10)$, the system level to 4, the depth of T_{coin}^{leaf} to 6, and the depth of T_{ctm} to 10, the number of input and output coins in the *Send* transaction to 1. In the *Setup* stage, the computational overhead for the regulators is 58.497 seconds. In the *Register* stage, the computational overhead for the regulators is 0.549 seconds, for the validators is 0.018 milliseconds, and for the users is 0.641 seconds. In the *Mint* stage, the computational overhead

TABLE I
COMPARISON OF TRANSACTION SCHEME

Scheme	Wan et al. [16]	Liu et al. [28]	Guan et al. [34]	Xue et al. [21]	Ours
The numbers of blockchains	Multi-chain	Multi-chain	Single-chain	Single-chain	Multi-chain
Ledger indistinguishability	×	✓	✓	✓	✓
Non-malleability	✓	✓	✓	✓	✓
Transaction type indistinguishability	✓	✓	-	-	✓
Scalability	✓	✓	-	-	✓
Unlinkability	×	✓	✓	✓	✓
Traceability	✓	×	×	✓	✓

TABLE II
NOTATIONS AND DESCRIPTIONS

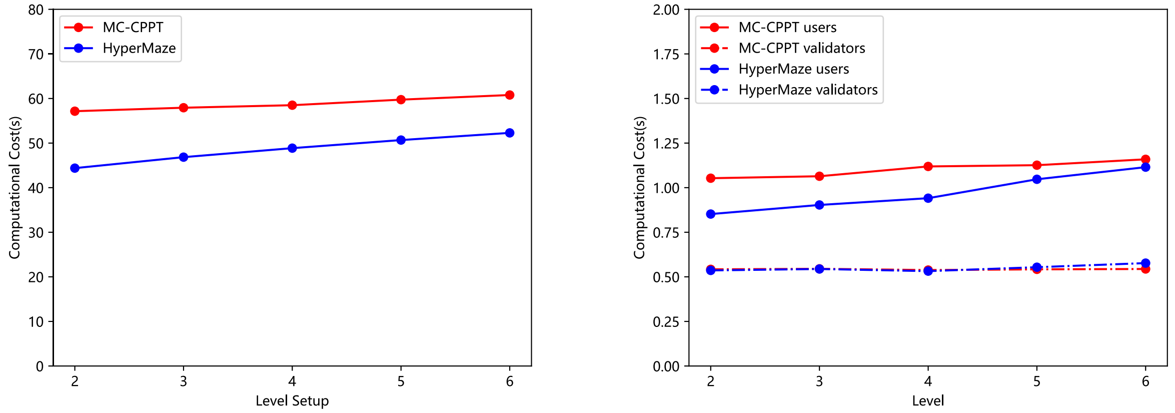
Notation	Description	Notation	Description
T_{genReg}	Time of generating an $\pi_{Register}$ proof operation	T_h	Time of a hash operation
T_{verReg}	Time of verifying an $\pi_{Register}$ proof operation	T_{ma}	Time of a modular addition operation
$T_{genMint}$	Time of generating an π_{Mint} proof operation	T_{mm}	Time of a modular multiplication operation
$T_{verMint}$	Time of verifying an π_{Mint} proof operation	T_{pa}	Time of a point addition operation
$T_{genSend}$	Time of generating an π_{Send} proof operation	T_{pm}	Time of a point multiplication operation
$T_{verSend}$	Time of verifying an π_{Send} proof operation		

TABLE III
COMPUTATIONAL COMPUTATIONAL OVERHEAD OF MC-CPPT

Stage	Regulators	Validators	Users
Setup	$(nt+t^2-1)T_{ma}$ $+(2nt+2t^2-3n-t)T_{mm}$ $+(nt-n)T_{pa}+(nt+n+t)T_{pm}$	-	-
Register	$2T_h+(2t+2)T_{ma}$ $4T_{mm}+(2t-2)T_{pa}+2T_{pm}$	$depth_{ctm}T_h+T_{verReg}$	$4T_h+2T_{ma}+2T_{pa}+4T_{pm}+T_{genReg}$
Mint	-	$depth_{coin}T_h+T_{verMint}$	$3T_h+T_{genMint}$
Send	-	$(coin_{in}*depth_{coin}+depth_{ctm}+1)T_h$ $+2T_{mm}+T_{pa}+2T_{pm}+T_{verSend}$	$(2coin_{out}+coin_{in}+4)T_h+T_{ma}+2T_{mm}$ $+coin_{out}T_{pa}+(2coin_{out}+2)T_{pm}+T_{genSend}$ $(+T_h+T_{pa}+2T_{pm})$
Receive	-	-	$T_{pa}+T_{pm}$
Trace	$2T_h+(t+1)T_{ma}+2tT_{mm}$ $+(t+1)T_{pa}+3T_{pm}$	-	-

Note: *In the table, n represents the number of leaf blockchain supervisors, t represents the threshold of supervisors, $depth_{coin}$ represents the depth of T_{coin}^{leaf}, $depth_{ctm}$ represents the depth of T_{ctm}, $coin_{in}$ represents the number of coins input in the *Send* transaction, and $coin_{out}$ represents the number of coins output in the *Send* transaction.

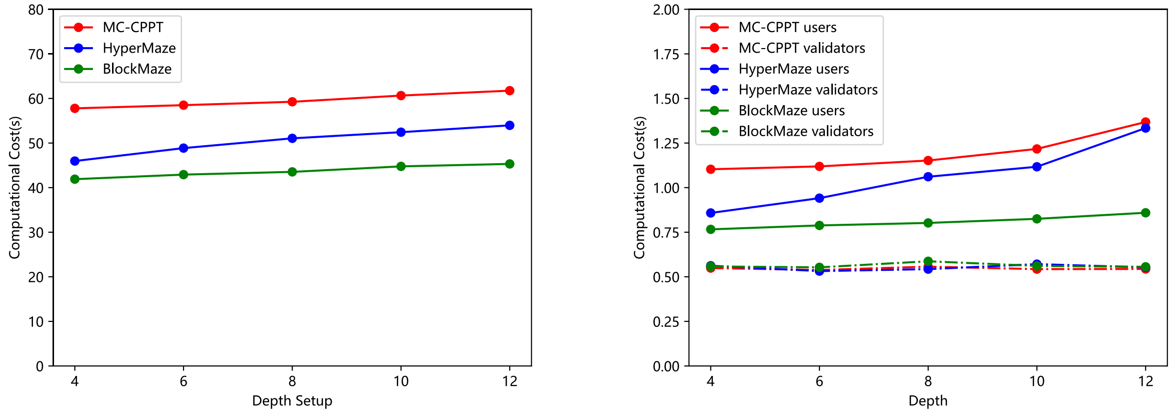
for the validators is 0.534 seconds, and for the users is 0.654 seconds, the communication overhead was 902B. In the *Send* stage, the computational overhead for the validators is 0.538 seconds, for users complying with regulatory guidelines is 1.119 seconds, and for users violating regulatory guidelines is 1.143 seconds, the communication overhead was 1447B. In the *Receive* stage, the computational overhead for the users is 0.197 milliseconds. In the *trace* stage, the computational overhead for the regulators is 0.652 milliseconds. Considering that the *Setup* stage only needs to be executed once, we deem its relatively long computational overhead acceptable. We also tested the computational overhead and communication overhead of HyperMaze [28] with 4-system level and 6-depth of MT. In the *Setup* algorithm, the computational overhead for the validator was 48.861 seconds. In the *Convert* algorithm, the computational overhead for the validator was 0.534 seconds and for the user was 0.686 seconds, the communication overhead was 1124B. In the *Redeem* algorithm, the computational overhead for the validator was 0.539 seconds and for the user was 0.696 seconds, the communication overhead was 1126B. In the *Send* algorithm, the computational overhead for the validator was 0.546 seconds and for the user was 0.712 seconds, the communication overhead was 1283B. In the *Receive* algorithm, the computational overhead for the validator was 0.532 seconds and for the user was 0.941 seconds, the communication overhead was 1219B. We also tested the computational overhead and communication overhead of BlockMaze [34] with 6-depth of MT. In the *Setup* algorithm, the computational overhead for the validator was 42.916 seconds. In the *Mint* algorithm, the computational overhead for the validator was 0.540 seconds and for the user was 0.682 seconds, the communication overhead was 1124B. In the *Redeem* algorithm, the computational overhead for the validator was 0.539 seconds and for the user was 0.696 seconds, the communication overhead was 1126B. In the *Send* algorithm, the computational overhead for the validator was 0.548 seconds and for the user was 0.762 seconds, the communication overhead was 1348B. In the *Deposit* algorithm, the computational overhead for the validator was 0.553 seconds and for the user was 0.788 seconds, the communication overhead was 1286B. The *Mint* stage in MC-CPPT serves a similar function to



(a) The computational overhead of the *Setup* stage

(b) The computational computational overhead of the *Send* stage

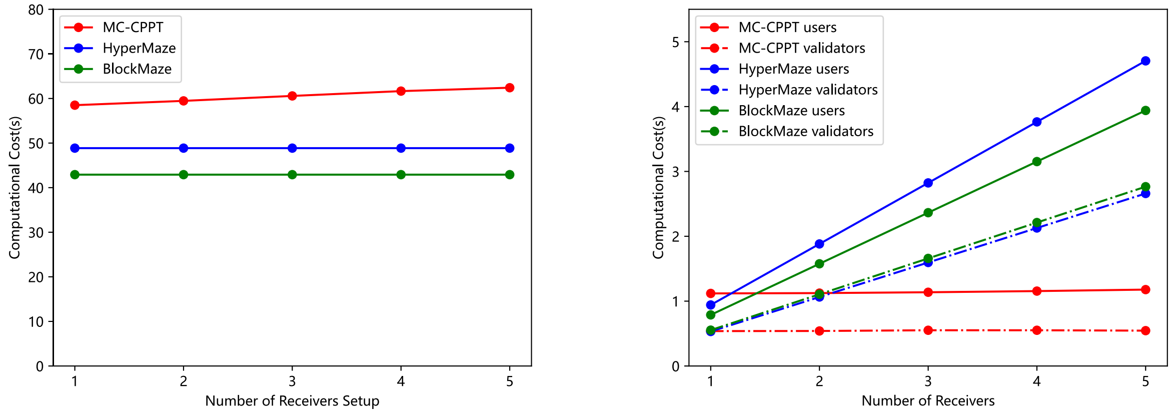
Fig. 5. Computational overhead across system level.



(a) The computational overhead of the *Setup* stage

(b) The computational computational overhead of the *Send* stage

Fig. 6. Computational overhead across depth of EMT.



(a) The computational overhead of the *Setup* stage

(b) The computational computational overhead of the *Send* stage

Fig. 7. Computational overhead across number of receivers.

the *Send* algorithms in [28], [34]. However, our computational overhead is significantly lower.

We kept other conditions constant and varied the system level to 2, 3, 4, 5, 6, comparing the results with [28], as shown in the Fig. 5. Similarly, we kept other conditions constant and varied the MT depth to 4, 6, 8, 10, and 12, comparing the results with HyperMaze [28] and BlockMaze [34], as shown in the Fig. 6. The higher overhead in the *Setup* stage for MC-CPPT compared to [28] is due to our need to generate two types of circuits for the *Send* transaction: one for users who comply with regulatory guidelines, which requires proving that the transaction amount and frequency are within the limits, and one for users who violate regulatory guidelines, which requires proving that the user has truthfully bound their identity. The *Send* stage in MC-CPPT is similar in function to the *Receive* algorithm in [28] and the *Deposit* algorithm in [34]. Our computational overhead is slightly higher because ours includes checks for the range of transaction amounts and frequencies, enhancing the system's security. Therefore, we consider the slightly increased computational overhead to be acceptable.

Additionally, MC-CPPT's *Send* stage allows the payer to transfer coins to multiple receivers in a single transaction, whereas users in [28], [34] need to initiate multiple transactions to transfer cryptocurrency to multiple receivers. Consequently, as the number of receivers increases, our computational overhead is significantly better, as shown in Fig. 7.

VIII. Conclusion

In this study, we proposed MC-CPPT, a conditional privacy-preserving transaction scheme for the UTXO-based blockchain with multi-chain architecture. A multi-node regulatory agency is introduced to establish regulatory guidelines, limit users' transaction amounts and frequencies, and restore the real identities of suspicious users. The validity and anonymity of the transactions are achieved through zero-knowledge proofs. The security of the proposed scheme is formally proven under a defined security model. In addition, we evaluated the performance, and experimental results showed that the performance of MC-CPPT is acceptable with more security features.

Acknowledgment

The authors are very grateful to the anonymous referees for their detailed comments and suggestions regarding this paper.

References

[1] H. Mao, T. Nie, H. Sun, D. Shen, and G. Yu, "A survey on cross-chain technology: Challenges, development, and prospect," *IEEE Access*, vol. 11, pp. 45527–45546, 2023.

[2] J. Cui, Y. Li, Q. Zhang, H. Zhong, C. Gu, and D. He, "DSchain: A blockchain system for complete lifecycle security of data in internet of things," *IEEE Trans. Dependable Secure Comput.*, vol. 21, no. 4, pp. 3977–3993, Jul./Aug. 2024.

[3] A. G. Gad, D. T. Mosa, L. Abualigah, and A. A. Abohany, "Emerging trends in blockchain technology and applications: A review and outlook," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 9, pp. 6719–6742, 2022.

[4] Q. Zhang, D. Sui, J. Cui, C. Gu, and H. Zhong, "Efficient integrity auditing mechanism with secure deduplication for blockchain storage," *IEEE Trans. Comput.*, vol. 72, no. 8, pp. 2365–2376, Aug. 2023.

[5] J. Abou Jaoude and R. G. Saade, "Blockchain applications–Usage in different domains," *IEEE Access*, vol. 7, pp. 45360–45381, 2019.

[6] D. Tse, B. Zhang, Y. Yang, C. Cheng, and H. Mu, "Blockchain application in food supply information security," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manage. (IEEM)*. Piscataway, NJ, USA: IEEE Press, 2017, pp. 1357–1361.

[7] Y. Guo and C. Liang, "Blockchain application and outlook in the banking industry," *Financial Innov.*, vol. 2, pp. 1–12, 2016.

[8] M. Hashemi Joo, Y. Nishikawa, and K. Dandapani, "Cryptocurrency, a successful application of blockchain technology," *Managerial Finance*, vol. 46, no. 6, pp. 715–733, Jun. 2020.

[9] Q. Zhang, X. Zhou, H. Zhong, J. Cui, J. Li, and D. He, "Device-side lightweight mutual authentication and key agreement scheme based on chameleon hashing for industrial internet of things," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 7895–7907, 2024.

[10] A. Matani, A. Sahafi, and A. Broumandnia, "A comprehensive review on blockchain scalability," *J. Elect. Comput. Eng. Innov. (JECEI)*, vol. 12, no. 1, pp. 187–216, Jan. 2024.

[11] M. Zhai et al., "Secret multiple leaders & committee election with application to sharding blockchain," *IEEE Trans. Inf. Forensics Secur.*, vol. 19, pp. 5060–5074, 2024.

[12] F. Hashim, K. Shuaib, and N. Zaki, "Sharding for scalable blockchain networks," *SN Comput. Sci.*, vol. 4, no. 1, 2022, Art. no. 2.

[13] Z. Hong, S. Guo, P. Li, and W. Chen, "Pyramid: A layered sharding blockchain system," in *Proc. IEEE INFOCOM 2021-INFOCOM or IEEE Conf. Comput. Commun.* Piscataway, NJ, USA: IEEE Press, 2021 pp. 1–10.

[14] Y. Liu, J. Liu, D. Li, H. Yu, and Q. Wu, "Fleetchain: A secure scalable and responsive blockchain achieving optimal sharding," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.* New York: Springer, 2020, pp. 409–425.

[15] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 931–948.

[16] Z. Wan, W. Liu, and H. Cui, "Hibechain: A hierarchical identity-based blockchain system for large-scale IoT," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1286–1301, Feb. 2022.

[17] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy (SP)*. Piscataway, NJ, USA: IEEE Press, 2018, pp. 583–598.

[18] Y. Li, X. Luo, W. Zhao, and H. Gao, "Reputation-based stable blockchain sharding scheme for smart cities with IoT consumer electronics: A deep reinforcement learning approach," *IEEE Trans. Consum. Electron.*, vol. 70, no. 3, pp. 5737–5746, Aug. 2024.

[19] Q. Zhang, J. Wu, H. Zhong, D. He, and J. Cui, "Efficient anonymous authentication based on physically unclonable function in industrial internet of things," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 233–247, 2023.

[20] L. Xue, D. Liu, J. Ni, X. Lin, and X. S. Shen, "Balancing privacy and accountability for industrial mortgage management," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4260–4269, Jun. 2019.

[21] L. Xue, D. Liu, J. Ni, X. Lin, and X. S. Shen, "Enabling regulatory compliance and enforcement in decentralized anonymous payment," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 931–943, Feb. 2022.

[22] C. Lin, D. He, X. Huang, M. K. Khan, and K.-K. R. Choo, "Dcap: A secure and efficient decentralized conditional anonymous payment system based on blockchain," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 2440–2452, 2020.

[23] C. Lin, X. Huang, J. Ning, and D. He, "ACA: Anonymous, confidential and auditable transaction systems for blockchain," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 6, pp. 4536–4550, Nov./Dec. 2023.

[24] K. Wüst, K. Kostiainen, V. Čapkun, and S. Čapkun, "Prcash: Fast, private and regulated transactions for digital currencies," *Financial Cryptography and Data Secur.: 23rd Int. Conf., FC, Frigate Bay, St. Kitts and Nevis, Feb. 18–22, Revised Sel. Papers 23*. New York: Springer, 2019, pp. 158–178.

[25] J. Duan, L. Wang, W. Wang, and L. Gu, "TRCT: A traceable anonymous transaction protocol for blockchain," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 4391–4405, 2023.

[26] Y. Li, G. Yang, W. Susilo, Y. Yu, M. H. Au, and D. Liu, "Traceable monero: Anonymous cryptocurrency with enhanced accountability," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 679–691, 2019.

[27] A. Deshpande and M. Herlihy, "Privacy-preserving cross-chain atomic swaps," in *Proc. Int. Conf. Financial Cryptography Data Secur.*. New York: Springer, 2020, pp. 540–549.

[28] W. Liu, Z. Wan, J. Shao, and Y. Yu, "Hypermaze: Towards privacy-preserving and scalable permissioned blockchain," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 360–376, 2021.

[29] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *Proc. IEEE Symp. Secur. Privacy*. Piscataway, NJ, USA: IEEE Press, 2013, pp. 397–411.

[30] E. B. Sasson et al., "Zerocash: Decentralized anonymous payments from bitcoin," in *Proc. IEEE Symp. Secur. Privacy*. Piscataway, NJ, USA: IEEE Press, 2014, pp. 459–474.

[31] M. R. Nosouhi et al., "Ucoin: An efficient privacy preserving scheme for cryptocurrencies," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 242–255, Jan. 2021.

[32] R. Xiao, W. Ren, T. Zhu, and K.-K. R. Choo, "A mixing scheme using a decentralized signature protocol for privacy protection in bitcoin blockchain," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 4, pp. 1793–1803, 2019.

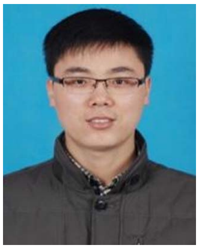
[33] T. Ruffing and P. Moreno-Sanchez, "Valueshuffle: Mixing confidential transactions for comprehensive transaction privacy in bitcoin," *Financial Cryptography and Data Secur.: FC Int. Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, Apr. 7, Revised Sel. Papers 21*. New York: Springer, pp. 133–154, 2017.

[34] Z. Guan, Z. Wan, Y. Yang, Y. Zhou, and B. Huang, "Blockmaze: An efficient privacy-preserving account-model blockchain based on zk-snarks," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 3, pp. 1446–1463, 2020.

[35] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct {Non-Interactive} zero knowledge for a von Neumann architecture," in *Proc. 23rd USENIX Secur. Symp. (USENIX Secur. 14)*, 2014, pp. 781–796.

[36] Y. Han, X. Yang, J. Sun, and D. Li, "Verifiable threshold cryptosystems based on elliptic curve," in *Proc. Int. Conf. Comput. Netw. Mobile Comput. (ICCNMC)*. Piscataway, NJ, USA: IEEE Press, pp. 334–337.

[37] Q. Pei and J. Ma, "ECC-based threshold digital signature scheme without a trusted party," in *Proc. Int. Conf. Comput. Intell. Secur.*, vol. 2. Piscataway, NJ, USA: IEEE Press, pp. 288–292.



Jie Cui (Senior Member, IEEE) was born in Henan Province, China, in 1980. He received the Ph.D. degree from the University of Science and Technology of China, in 2012. Currently, he is a Professor and Ph.D. Supervisor with the School of Computer Science and Technology, Anhui University. His research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security, and software-defined networking (SDN). He has over 150 scientific publications in reputable journals (e.g. IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS), academic books, and international conferences.

Wenting Zhuang is a Research Student in the School of Computer Science and Technology, Anhui University. Her research focuses on the security of the blockchain system.

Hong Zhong was born in Anhui Province, China, in 1965. She received the Ph.D. degree in computer science from the University of Science and Technology of China, in 2005. Currently, she is a Professor and Ph.D. Supervisor with the School of Computer Science and Technology, Anhui University. Her research interests include applied cryptography, IoT security, vehicular ad hoc network, cloud computing security, and software-defined networking (SDN). She has over 200 scientific publications in reputable journals (e.g. IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY), academic books, and international conferences.

Qingyang Zhang was born in Anhui Province, China, in 1992. He received the B.Eng. and Ph.D. degrees in computer science from Anhui University, in 2021. Currently, he is an Associate Professor with the School of Computer Science and Technology, Anhui University. His research interests include edge computing, computer systems, and security. He has over 30 scientific publications in reputable journals (e.g. PROCEEDINGS OF THE IEEE, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON COMPUTERS) and international conferences.

Fengqun Wang is currently working toward the Ph.D. degree with the School of Computer Science and Technology, Anhui University, Hefei, China. His research interests include IoT security, blockchain, and applied cryptography.

Debiao He (Member, IEEE) received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2009. Currently, he is a Professor with the School of Cyber Science and Engineering, Wuhan University, Wuhan, China, and Shanghai Key Laboratory of Privacy Preserving Computation, MatrixElements Technologies, Shanghai, China. His research interests include cryptography and information security, in particular, cryptographic protocols. He has published over 100 research papers in refereed international journals and conferences, such as IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION SECURITY AND FORENSIC, AND USENIX SECURITY SYMPOSIUM. He is the recipient of the 2018 IEEE Systems Journal Best Paper Award and the 2019 IET Information Security Best Paper Award. His work has been cited more than 10000 computationals at Google Scholar. He is on the Editorial Board of several international journals, such as *Journal of Information Security and Applications*, *Frontiers of Computer Science*, and *Human-Centric Computing & Information Sciences*.