Enabling Regulatory Compliance and Enforcement in Decentralized Anonymous Payment

Liang Xue, *Student Member, IEEE*, Dongxiao Liu, *Member, IEEE*, Jianbing Ni, *Member, IEEE*, Xiaodong Lin, *Fellow, IEEE*, and Xuemin (Sherman) Shen, *Fellow, IEEE*

Abstract—Decentralized anonymous payment (DAP) enables users to directly transfer cryptocurrencies privately without passing through a central authority. Anonymous cryptocurrencies have been proposed to improve the privacy degree of DAP systems, such as Zerocash and Monero. However, the strong degree of privacy may cause new regulatory concerns, i.e., the anonymity of transactions can be used for illegal activities, such as money laundering. In this paper, we propose a novel DAP scheme that supports regulatory compliance and enforcement. We first introduce regulators into the system, who define regulatory policies for anonymous payment, and the policies are enforced through commitments and non-interactive zero-knowledge proofs for compostable statements. By doing so, users can prove that transactions are valid and comply with regulations. A tracing mechanism is embedded in the scheme to allow regulators to recover the real identities of users when suspicious transactions are detected. The formal security model and proof are provided to demonstrate that the proposed scheme can achieve desired security properties, and the performance evaluation shows its high efficiency.

Index Terms—Security and privacy, decentralized anonymous payment, blockchain, cryptocurrencies, regulation enforcement

1 Introduction

Blockchain, which can record information in a transparent and unalterable manner through the use of cryptography and decentralization, has attracted much attention from academia and industry. Due to its immutability, verifiability, and programmability, blockchain has found its applications in various areas, such as financial services [1], data storage [2], and Internet of Things [3]. Among them, the most successful application is cryptocurrency, as the blockchain can facilitate safe and easy transactions without a trusted authority. Moreover, cryptocurrency enables rapid payments and can greatly reduce the transaction costs. It provides an innovative approach to transaction execution, and has the potential to revolutionize the financial industry and drive economic changes on a large scale. According to the CoinMarketCap [4], there have been 8795 cryptocurrencies on the market, with the market value reaching more than $1809 billion.

Although cryptocurrencies have many attractive characteristics, transactional privacy is one of the most challenging problems when applying them in practice. The nature of blockchain makes that all transactions can be accessed by each node in the blockchain network. Thus, the sensitive information such as payers, payees, and transferred amounts can be obtained by anyone in the network. In Bitcoin system, the pseudonym mechanism is employed to disguise identities of users. However, researchers have shown that one can de-anonymize users and trace transactions by using graph analysis and address clustering [5], [6]. To improve the privacy of cryptocurrencies, decentralized anonymous payment (DAP) schemes are proposed, such as Zerocash [7] and Monero [8]. A DAP scheme enables users to pay others privately, where the addresses of transaction participants and transferred amounts are hidden from others by using cryptographic techniques including zero-knowledge proofs and pseudorandom functions.

However, the strong degree of privacy, which enables anonymity and untraceability of transactions, creates new regulatory concerns [9], [10]. By using blockchain, traditional currency regulatory processes have been bypassed. Moreover, since cryptocurrencies are not legal tender in any jurisdiction and are not issued by a monetary authority, payers and recipients of cryptocurrencies are not under control of any third authority, causing cryptocurrency markets are largely unregulated. Thus, the convertible cryptocurrencies can be exploited to conduct illicit activities, such as money laundering and terrorist financing [11].

In traditional banking industry, there is an authority who controls the details of the users' identities and transaction amounts, making it can support financial regulation

- *Liang Xue, Dongxiao Liu, and Xuemin (Sherman) Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: {liang.xue, dongxiao.liu, sshen}@uwaterloo.ca.*
- *Jianbing Ni is with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON K7L 3N6, Canada. E-mail: jianbing.ni@queensu.ca.*
- *Xiaodong Lin is with the School of Computer Science, University of Guelph, Guelph, N1G 2W1, Canada. E-mail: xlin08@uoguelph.ca.*

Manuscript received 18 Apr. 2021; revised 1 Nov. 2021; accepted 3 Jan. 2022. Date of publication 25 Jan. 2022; date of current version 14 Mar. 2023. (Corresponding author: Jianbing Ni.)
Digital Object Identifier no. 10.1109/TDSC.2022.3144991

inherently. However, in distributed setting, no authority can be relied on, and due to the public visibility of the blockchain, the participants and amounts of transactions need to be protected, causing the regulation of cryptocurrencies an intractable problem. In the past years, many efforts have been made by countries to address the regulatory concerns of cryptocurrencies [12], [13]. In the US, Financial Crimes Enforcement Network (FinCEN) issued a guidance document that extends Money Services Business (MSB) of the US Bank Secrecy Act to cryptocurrencies, in which companies working with cryptocurrencies should comply with *know-your-customer* regulations, the same rules that apply to banks and other financial institutions. Moreover, for the third-party service providers, such as exchanges, payment services provides and wallet provides, licensing regimes [14] are introduced to protect customers and combat money laundering. Canada has also enacted Proceeds of Crime and Terrorist Financing Act, and transactions over CA $10,000 should be reported. To meet regulatory requirements, reputable exchanges and online wallets enforce real-name identity registration. Thus, they hold a large amount of information of their users, including the participants of transactions and the balance of their accounts, which results in little privacy for users.

Blockchain technology also provides an effective opportunity for governments to issue their own digital currencies, which use blockchain-based token to represent a country's official currency and are regulated by monetary authority of the country. One of the benefits of Central Bank Digital Currency (CBDC) is visibility to the movement of money, which will significantly improve the insight into the economy and is helpful for monetary policy making. However, since the central authority can monitor and regulate the transactions within the network, CBDC has serious privacy implications, which hinders its wide adoption. For many countries, a positive developmental environment for digital currencies, supported by regulation, is desired. Thus, a balance between the regulation and privacy of decentralized payment must be struck.

In this paper, for regulatory compliance of anti-money laundering, we introduce regulators into our system, and define three regulatory policies, which include: 1) the total amount of the cryptocurrency one can transfer in a time period is limited; 2) the frequency of transactions in a time period is restricted; 3) When suspicious transactions are detected, the identities of transaction participants can be recovered by regulators. Based on the defined policies, we propose a regulated and decentralized anonymous payment scheme, called RDAP, which can enforce the regulatory policies while preserving users' privacy. When conducting a transaction, a user needs to prove that the transaction complies with the policies, or else the user needs to include its identity information in the transaction. Note that, if the transaction is policy-compliant, identity of the user and transaction content are concealed from regulators. When illegal or suspicious transactions are discovered, regulators can recover the real identities of payers and conduct investigations on the transactions. The challenges in designing a regulated and decentralized anonymous payment scheme include: 1) How to design a mechanism that can audit the amount of transactions from one user as well as the number of transactions conducted by the user, while the unlinkability of transactions of the same user is guaranteed. An external observer should not be able to associate a counter with a user, and the connection between an old counter and an updated counter should be hidden. 2) How to achieve the regulatory enforcement, i.e., the validity and compliance of regulation for the transactions can be verified by validators. 3) How to achieve accountability, i.e., when a transaction violates the policies, the identity of the user can be recovered. Note that a simple encryption scheme will give the regulator too much power since it can decrypt the identities of users for each transaction.

To address the challenges, we designed a transaction counter, which is represented in an algebraic form and is used to obtain the cumulative transferred amounts and number of transfers in a time period. We use zero-knowledge proofs (ZKP) to break the linkability of transactions and prove the compliance of a transaction to the regulatory policies. Both Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge (zk-SNARKs) [15] and sigma protocol-based Zero-knowledge proofs [16] are utilized and bound together to prove the validity of transactions. A tracing mechanism is embedded in our scheme to enable regulators to reveal the identities of users. The contributions of the paper are summarized as follows:

- We define the regulatory policies for cryptocurrency and propose a decentralized anonymous payment scheme supporting regulatory enforcement, where regulators are introduced into the system and they can detect and prevent suspicious transactions. To ensure both privacy protection and regulatory compliance, the total transferred amount and number of transfers can be accumulated without leaking the links between transactions, i.e., our scheme achieves anonymity and unlinkability of transactions. Moreover, the identities of users can be recovered when they violate the regulatory policies.
- Considering that there are arithmetic statements and algebraic statements that need to be proved in the scheme, we use both zk-SNARKs and sigma protocols, which are suitable for proving arithmetic statements and algebraic statements respectively, to generate the zero-knowledge proofs for regulation compliance. Moreover, the two kinds of proofs are elaborately connected to each other. Compared with the one that only adopts zk-SNARKs, our scheme is more efficient for users.
- We define a security model for the regulated and decentralized anonymous payment schemes, with the consideration of both regulatory compliance and privacy protection, and prove that the proposed RDAP achieves the desired security properties. Moreover, simulation results demonstrate the low computation cost for both regulators and users.

The organization of this paper is as follows. In Section 2, we review the related work about anonymity and accountability in cryptocurrencies. We describe the system model and security model in Section 3. In Section 4, we briefly present preliminaries. We propose decentralized anonymous payment with regulatory compliance scheme in Section 5. In Section 6, we prove the security of our scheme. Performance evaluation is given in Section 7, followed by the conclusion in Section 8.

2 Related Works

2.1 Anonymity in Cryptocurrencies

Considering the transparency of blockchain [17], to achieve privacy-preserving cryptocurrency, Saberhagen proposed Cryptonote [18], which utilizes traceable ring signatures to hide the participants of transactions and one-time keys to prevent double-spending of a coin. CoinJoin [19] uses coin mixing services to protect the originators of transactions, however, a centralized CoinJoin server is required in the system. Monero protocol [8], which is designed based on CryptoNote, uses the technique of Confidential Transactions to hide the amounts of transactions. Considering that Bitcoin only preserves users' privacy through pseudonyms, Miers *et al.* proposed Zerocoin [20], which uses an accumulator scheme and non-interactive ZKP to break the link between the Mint transactions and Spend transactions. Due to Zerocoin still reveals the destinations and amounts of transactions, Ben-Sasson *et al.* formulated decentralized anonymous payment (DAP) schemes [7], and proposed Zerocash as a practical instantiation. By using zk-SNARKs, commitment schemes [21], and collision-resistant hash function-based Merkle tree, Zereocash preserves the confidentiality of the origin, destination, and transferred amount of a payment.

2.2 Auditability and Accountability in Cryptocurrencies

To solve the regulatory issues raised by anonymous transactions, many efforts have been made to balance the privacy protection and enforcement of regulatory policies. Garman *et al.* [9] added the policy-enforcement mechanisms to Zerocash, and the proposed approach allows selective user tracing and tained coins tracing. However, they are achieved by using zk-SNARKs techniques, resulting in the poor performance of the system. Wu *et al.* proposed a regulated digital currency [22], where transactions are supervised by an auditor, who can monitor the flow of money and obtain identities of users. In the scheme, coins in the system have fixed denominations, and transferred amounts are not concealed. Ma *et al.* proposed SkyEye [23], which allows a regulator to trace users' identities by adding identity proofs in each anonymous transaction. The identity proof is used to prove users' legitimacy and achieve tracing. However, the regulator can recover the identities of all transaction participants regardless of whether a user violates the policies. Wust *et al.* proposed PRCash [24], which is designed based on Mimblewimble and can achieve private and regulated transactions in permissioned blockchain setting. Lin *et al.* presented a decentralized condition anonymous payment system DCAP [25], where for each transaction, a new anonymous address will be created by a transaction participant from its long-term address. In the scheme, users' long-term addresses are encrypted by the manager's public key, and the manager needs to obtain the long-term address of a user to determine whether the user is legitimate, which causes the large overhead of managers and the disclosure of user privacy.

Chatzigiannis *et al.* investigated the distributed payment schemes [26] that provide auditability functionalities for regulators. In the paper, the relevant work falls into two categories: organization-based auditability, where some third parties such as exchanges need to offer cryptographic solvency proofs [27], and user/transaction level-based auditability, where regulators can learn about the previous transactions of a specific user or the involved users of a transaction. The authors also summarized the related schemes in terms of security guarantees, efficiency, and general properties such as the account model and consensus protocol used. Chatzigiannis *et al.* also proposed a distributed payment scheme [28], which allows a third party to audit the transactions generated from an authorized set of entities. The provided pruning function can save the storage overhead of a ledger without affecting the audit functionalities. Li *et al.* proposed the concept of Traceable Monero [29], a novel method to balance the user anonymity and accountability on top of Monero. In their framework, the tracing authority is optimistic, which is only involved when investigations in malicious transactions are required. They presented a construction by cleverly taking advantage of the trick of verifiable encryption to identify the long-term account and the one-time account, respectively. Androulaki *et al.* proposed an auditable token system [30] for enterprise networks. The system is built based on a permissioned blockchain and uses conservative computational assumptions such as discrete-logarithm assumption.

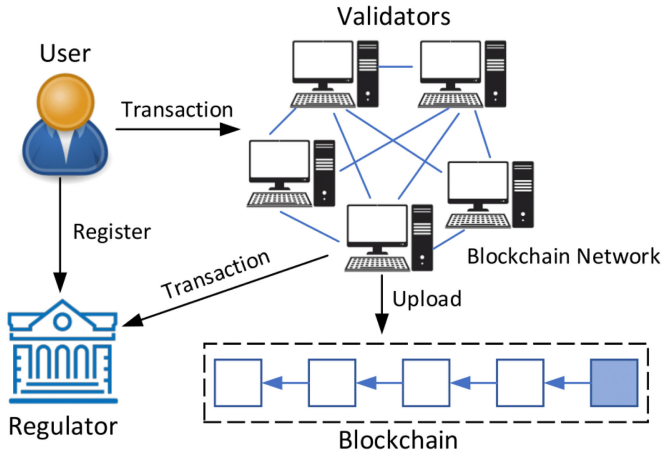


Fig. 1. System model.

3 System Model and Security Model

3.1 System Model

As shown in Fig. 1, we focus on a decentralized payment system supporting regulatory compliance and enforcement. The involved entities are described as follows:

- Regulators: To allow the regulation of the flow of cryptocurrencies, we assume in each jurisdiction, there is a regulator, which is responsible for generating pseudo-identity certificates for users and tracing users' identities for suspicious transactions. Only the regulators have the right to recover users' identities.
- Users: Users need to first register with a regulator to obtain initial counter certificates and pseudo-identity certificates, which are used for potential identity tracing. Users can be payers or recipients in the system. They can create address public keys and private keys for coin minting and transferring.

- Validators: Validators are responsible for verifying the correctness of transactions and maintaining the public ledgers. If a transaction is valid and complies with the regulatory policies, they will add it on the blockchain. Otherwise, validators will send the transaction to the corresponding regulator, who can recover the user's real identity and conduct the follow-up investigation.
- Trust Authority (TA): TA is responsible for initializing the system and generating public parameters for users and validators.

Our system works as follows: Users in the system should first register with the local regulator. Validators maintain a public and append-only ledger. To spend the cryptocurrency, users should first mint coins and propose a mint transaction. The transaction is put on the ledger only when the user has paid the corresponding amount of Bitcoin or other currency to a backing escrow pool. To hide the origin of the payer, a commitment of the coin is included in the mint transaction, and the commitment is added to the list of coin commitments. When the user spends the coin, he needs to prove that the coin commitment is in the list and the serial number of the coin is exposed to prevent the double-spending. Users can transfer coins through pour transactions and spend the received coins. To preserve users' transaction privacy, a zero-knowledge proof is included in a pour transaction to prove that the transaction is valid without leaking the participants and amount transferred. Moreover, a regulatory compliance proof should be embedded in the transaction to ensure the effective regulation. When violations are detected by validators, they will send the corresponding transactions to the local regulator, who can recover the identities of the users and investigate the transactions.

3.2 Security Model

For a regulated and decentralized anonymous payment system, the desired properties include anonymity, authentication, balance, and traceability. In the following, we formally define the properties by games [31] involving a challenger $\mathcal{C}$, an adversary $\mathcal{A}$, and an RDAP oracle $\mathcal{O}^{RDAP}$, where $\mathcal{A}$ is a polynomial time adversary. $\mathcal{A}$ can send $\mathcal{C}$ different types of queries, which include Register, CreateAddress, Mint, Pour, and Receive queries. After the sanity checks of the queries, $\mathcal{C}$ forwards the queries to $\mathcal{O}^{RDAP}$, which maintains a ledger and executes the queries according to the RDAP scheme, and outputs the resulting transactions. In this way, $\mathcal{A}$ can elicit the actions of honest users, and learn the public outputs. Note that $\mathcal{A}$ cannot obtain the private inputs in generating a transaction. $\mathcal{A}$ can also send mint or pour transactions to $\mathcal{C}$, which is called insert queries. After the sanity check, $\mathcal{C}$ forwards the transactions to $\mathcal{O}^{RDAP}$.

GAME Anonymity: For the anonymity, as in [31], we define the property by using ledger indistinguishability, i.e., the ledger reveals no information about users other than public information on a ledger, such as the total number of transactions and transaction fees paid by a user. $\mathcal{C}$, $\mathcal{A}$, and two oracles $\mathcal{O}_0^{RDAP}$, $\mathcal{O}_1^{RDAP}$ are involved in the game.

Initialization: $\mathcal{C}$ randomly selects a bit $b \in \{0,1\}$ and initializes $\mathcal{O}_0^{RDAP}$ and $\mathcal{O}_1^{RDAP}$. For $i \in 0,1$, $\mathcal{O}_i^{RDAP}$ maintains a ledger L_i.

Query: $\mathcal{A}$ can submit different queries, such as Register, Mint, and Pour queries, to the two oracles. Each time, $\mathcal{A}$ submits two queries Q, Q' with the same type and identical public information to $\mathcal{C}$. $\mathcal{C}$ provides the view of two ledgers L_0, L_1 to $\mathcal{A}$ with a random order, i.e., $L_{left} = L_b, L_{right} = L_{1-b}$, where $b \in \{0,1\}$. If the queries are Register, Mint, Pour, Receive queries, after the sanity check of the two queries, $\mathcal{C}$ sends Q to $\mathcal{O}_0^{RDAP}$, and Q' to $\mathcal{O}_1^{RDAP}$. If the queries are Insert queries, Q is sent to $\mathcal{O}_{left}^{RDAP}$, and Q' is sent to $\mathcal{O}_{right}^{RDAP}$.

Guess: After the queries, $\mathcal{A}$ needs to determine that whether the ledgers it sees are $L_{left} = L_0, L_{right} = L_1$, which means $b = 0$, or $L_{left} = L_1, L_{right} = L_0$, i.e., $b = 1$. $\mathcal{A}$ returns a bit b' to $\mathcal{C}$, which is the guess of $\mathcal{A}$.

$\mathcal{A}$ wins the game if the $b' = b$. The anonymity property requires that for a polynomial-time $\mathcal{A}$, the advantage of $\mathcal{A}$, i.e., $Pr[b' = b] - \frac{1}{2}$, is negligible.

GAME Authentication: Authentication property captures the requirement that users in the system should register with a local regulator to ensure the effective supervision of the cryptocurrency. The GAME Authentication involves $\mathcal{A}$, $\mathcal{C}$, and an oracle $\mathcal{O}^{RDAP}$.

Initialization: $\mathcal{C}$ initializes $\mathcal{O}^{RDAP}$, which maintains a ledger L.

Queries: $\mathcal{A}$ adaptively interacts with $\mathcal{O}^{RDAP}$ by sending different queries to $\mathcal{C}$, which forwards the queries to $\mathcal{O}^{RDAP}$ if the queries pass the sanity checks. $\mathcal{O}^{RDAP}$ executes the queries and provides $\mathcal{A}$ with the view of the ledger L.

Output: $\mathcal{A}$ outputs a pour transaction tx' that satisfies that: 1) the transaction is valid; 2) the payer of the transaction is not registered with the regulator.

$\mathcal{A}$ wins GAME Authentication if $\mathcal{A}$ outputs a transaction that satisfies the above two conditions. The authentication property requires that no adversary can have a non-negligible probability in winning the above game.

GAME Balance: Balance property captures the requirement that the total input value and the output value of a transaction should be equal. The GAME Balance involves $\mathcal{A}$, $\mathcal{C}$, and an oracle $\mathcal{O}^{RDAP}$.

Initialization: $\mathcal{C}$ initializes $\mathcal{O}^{RDAP}$, which maintains a ledger L.

Queries: $\mathcal{A}$ can submit queries to $\mathcal{O}^{RDAP}$, which simulates the behavior of honest users. $\mathcal{O}^{RDAP}$ executes the queries and provides $\mathcal{A}$ with the view of the ledger L.

Output: After the queries, $\mathcal{A}$ outputs a set of coins $\mathcal{S}_{coin}^{\mathcal{A}}$.

Let $v_{unspent}$ be the value of unspent coins of $\mathcal{A}$ in $\mathcal{S}_{coin}^{\mathcal{A}}$ and v_{public} be the sum of public values in the Pour transactions inserted by $\mathcal{A}$. v_{Mint} denotes the total value of coins $\mathcal{A}$ has minted. $v_{ADDR \to \mathcal{A}}$ is the total value that $\mathcal{A}$ received from honest users, and $v_{\mathcal{A} \to ADDR}$ is the total value that transferred by $\mathcal{A}$ to honest users. $\mathcal{A}$ wins GAME Balance if the total value it has spent and the remaining coins it can spend is greater than it has minted or received from others, i.e.,

$$v_{unspent} + v_{public} + v_{\mathcal{A} \to ADDR} \geq v_{mint} + v_{ADDR \to \mathcal{A}}.$$

The balance property requires that $\mathcal{A}$ can win the above game with only a negligible probability.

GAME Traceability: Traceability property captures the requirement that when a transaction violates the regulatory policies, the regulator can recover the user's real identity

from the transaction. The GAME Traceability involves $\mathcal{A}$, $\mathcal{C}$, and an oracle $\mathcal{O}^{RDAP}$.

Initialization: $\mathcal{C}$ initializes $\mathcal{O}^{RDAP}$, which maintains a ledger L.

Queries: $\mathcal{A}$ can interact with the oracle $\mathcal{O}^{RDAP}$ by sending different types of queries to $\mathcal{C}$, which proxies queries to $\mathcal{O}^{RDAP}$. $\mathcal{O}^{RDAP}$ executes the queries and provides $\mathcal{A}$ with the view of the ledger L, which simulates the behavior of honest parties. When transactions violate the regulatory policies, $\mathcal{O}^{RDAP}$ can recover the identities of parties by using the Trace algorithm in RDAP. $\mathcal{A}$ can also obtain the recovered identities of parties.

Output: In this phase, $\mathcal{A}$ outputs a pour transaction tx^*.

$\mathcal{A}$ wins the game if tx^* violates the regulatory policies, whereas it pass the verification of validators, or the regulator fails to recover the identity of the user, or the recovered user does not register with the regulator. RDAP achieves traceability if $\mathcal{A}$ can win the above game with no more than a negligible probability.

4 Preliminaries

4.1 Zero-Knowledge Proofs

Let L be the language $L = \{x| \exists w\, s.t.\, R(x,w) = 1\}$, where w is a witness of x. Zero-knowledge proofs enable a prover who owns w to convince a verifier that $x \in L$ without leaking w.

Sigma protocols [32] are interactive zero-knowledge proof protocols between two parties P and V. The two parties have a common input x, and P has a private witness w. In general, sigma protocols are suitable for algebra-related statements, for example, proving that one owns an x such that $g^x = y$, where g, y are public values in a multiplicative cyclic group. Sigma protocols only need a constant number of algebraic operations, and proof sizes are small. In general, a sigma protocol is a three-move protocol. In the protocol, P first transmits an initial message T to V. Then, V returns a challenge c. After that, P calculates and forwards a response z to V. According to the (T, c, z), V outputs 1 if the proof is verified. Let G be a multiplicative cyclic group and g is a generator. A sigma protocol that proves the knowledge of $x \in Z_p$ such that $y = g^x$ is shown as follows:

1) P selects a random $a \in Z_p$, and calculates $T = g^a$. Then, P sends T to V.
2) V randomly selects $c \in Z_p$, and returns c to P.
3) P calculates $z = a - cx \bmod p$, and forwards z to V.
4) V calculates $y^c g^z$, and if it is equal to T, V outputs 1. Otherwise, it outputs 0.

4.2 Zk-SNARKs

Let C denote an arithmetic circuit. R_C represents an NP relation $R_C = \{(x,w)|C(x,w) = 0\}$. The language for R_C is $L_C = \{x| \exists\, \omega, s.t.\, C(x,\omega) = 0\}$. Zk-SNARKs are suitable for proving statements that are denoted as arithmetic circuits. The proofs have short sizes and can be verified within a few milliseconds. A zk-SNARK scheme [33], [34] for language L_C contains three algorithms (KeyGen, Prove, Verify):

KeyGen(λ, C): The inputs of the algorithm include a security parameter λ and a circuit C. The outputs are a proving key pk and a verification key vk.

Prove(pk, x, ω): With the inputs pk and an witness ω of x, the algorithm returns a proof π for the statement x.

Verify(vk, x, π): Given vk, x, and π, the algorithm outputs 1 if π is a valid proof for $x \in L_C$.

4.3 Decentralized Anonymous Payment (DAP) Schemes

A DAP scheme [7] is a decentralized electronic currency scheme that enables anonymous payments among individuals. By which, a user can mint coins and transfer coins to others without leaking the participants and the transferred amount. A DAP scheme contains six polynomial algorithms: *Setup*, *CreateAddress*, *Mint*, *Pour*, *VerifyTransaction*, and *Receive*.

Setup: This algorithm is used to generate system parameters. Given a security parameter λ, the algorithm returns public parameters pp.

CreateAddress: Users can use this algorithm to create address key pairs. Given pp, the algorithm outputs an address public and private key pair $(addr_{pk}, addr_{sk})$, which can be used to send and receive coins.

Mint: Before users transfer coins to others, they need to mint coins first. Given pp, $addr_{pk}$, and a coin value v, the algorithm returns a mint transaction tx_{Mint} and a coin $\mathbf{c}$.

Pour: This algorithm is run by users. It can be used to transfer coins from payers to payees. The inputs of the algorithm include pp, the old coins c_1^{old}, c_2^{old}, the secret address keys $addr_{sk_1}^{old}, addr_{sk_2}^{old}$, and the public address keys of payees. The outputs are two new coins c_1^{new} and c_2^{new} for the recipients and a pour transaction tr_{Pour}.

VerifyTransaction: The algorithm is run by validators. By using this algorithm, they can verify whether a pour transaction or a mint transaction is valid. The algorithm takes as input pp, a mint or a pour transaction, and a public ledger L. If the validity of the transaction is verified, the algorithm outputs a bit 1. Otherwise, it outputs 0.

Receive: Users can use this algorithm to receive coins from others. The algorithm takes as input an address key pair $(addr_{pk}, addr_{sk})$ and a public ledger L, and outputs the received coins.

5 Decentralized Anonymous Payment Scheme With Regulatory Compliance

5.1 Our Policies

Considering that cryptocurrency can be exploited for illegal transactions, regulation policies should be enforced to ensure compliance with anti-money laws and counter-terrorist financing. In many countries, there is a restriction on the amount of the cryptocurrency one can transfer at a time. For example, in the US, transactions over \$1000 should be reported to Financial Crimes Enforcement Network (FinCEN). Exchanges such as Huobi and Binance also set a threshold for the total amount of cryptocurrency one can purchase in a day. According to existing requirements, We have similar restrictions on transactions. Moreover, in order to realize the supervision of cryptocurrency, users in the system should register with the regulators, who will issue them pseudo-random identity certificates that are used for identity tracing if needed. Taking into account users' privacy concerns, users' transaction privacy should be

preserved, which means a user's identity, balance, and transferred amount should be hidden from others. Meanwhile, there should be a mechanism that can detect the policy violations. When suspicious transactions are discovered, regulators can recover the identities of users and take corresponding measures.

Thus, the regulatory policies in our decentralized anonymous payment scheme are listed as follows:

- The total amount of cryptocurrency one can transfer in a time period is limited to v_{limit}.
- The number of the transactions one can conduct in a time period is limited to n_{limit}.
- Users should register with the local regulator. When violations are detected, the identities of the corresponding users can be recovered.

Note that by adjusting the length of a period and the amount one can transfer, regulators can detect and prevent the suspicious transactions as needed.

5.2 Overview of RDAP

To effectively enforce the regulatory policies defined above, a counter is built for each user at each time period to accumulate the amount of cryptocurrency transferred and the number of transfers in this time period. The initial counter is signed by the local regulator. After each transaction, the counter is updated according to the amount transferred. To guarantee the unlinkability of different transactions conducted by the same user, we build a Merkle Hash Tree (MHT) $\mathcal{T}_{cID}$ for counters. When a user transfers cryptocurrency to others, the user needs to prove that the counter used in the transaction is a valid counter, which means it is a leaf node of the tree $\mathcal{T}_{cID}$, and the counter is the latest counter of the user.

To guarantee the correctness and regulation enforcement of transactions, two kinds of proofs are embedded in a pour transaction. The first proof π_1 proves that the transaction is valid, i.e., the coins to be spent belong to the user and are not spent before, the new coins generated for the receiver are well-formed, and the input value and the output value are balanced. Thus, validators can verify the validity of transactions by checking π_1. Since the pseudorandom function PRF and the commitment scheme used to construct a coin can be instantiated via SHA256 hash function, π_1 can be achieved by using the zk-SNARKs technique. On the other hand, to guarantee the enforcement of regulatory policies, i.e., the whole number of cryptocurrency and the number of transactions one can conduct in a period are limited, or else, a regulator can recover the identities of the payer, we accumulate the values one transferred and the number of transfers. This is more convenient to represent it algebraically. Thus, we construct a transaction counter, where the cumulative values are in the exponential positions. In each transaction, the counter is updated according to the transaction amount. To guarantee that the counter is updated correctly, the transaction is in accordance with the regulatory policies, and a user cannot use others' counter to avoid supervision, we designed the second zero-knowledge proof π_2. If the transaction complies with the defined policies, the proof proves that the amount and number of times transferred are within limits, otherwise, the user needs to include the pseudo-identity certificate in the transaction and proves that it belongs to him/her. With the second proof, if a transaction violates the policies, the corresponding regulator can recover the identity of the user and investigate the transaction. If the transaction is judged legal, the regulator will sign the transaction and send it to validators, who will add the transaction to the blockchain.

5.3 Details of RDAP

The proposed RDAP consists of seven algorithms, namely, Setup, Register, Mint, Pour, Verify, Receive, and Trace.

Setup: With the input of a security parameter λ, TA chooses an elliptic curve group G of order p, and the points of the elliptic curve are defined over a field $\mathbb{F}_t$. $g, g_1, g_2, g_3, g_4, g_5, \bar{g}, h$ are generators of group G and for each two of them, the discrete logarithm of an element with respect to the other one is unknown. Let $q > 2t^3$ be a prime number, and $\mathbb{G}$ be an elliptic curve group of order q. $\hat{g}$ and $\hat{h}$ are generators of $\mathbb{G}$, and $log_{\hat{g}}\hat{h}$ is unknown. TA also generates three pseudorandom functions PRF_x^{addr}, PRF_x^{pk}, PRF_x^{sn}, and two secure hash function $H : G \times Z_p \rightarrow \{0,1\}^k$ and $H_1 : \{0,1\}^* \rightarrow \{0,1\}^k$, where $k > \lambda$. Let COM and Com be two commitment schemes and CRH be a collision-resistant hash function that is used for constructing Merkle trees. The main parameters in our scheme are listed in Table 1.

Register: Users in a jurisdiction need to register with the corresponding regulator. A regulator $\mathcal{R}$ selects a random number σ as its master secret key, and calculates the public key $P_{pub} = g^{\sigma}$. $\mathcal{R}$ also generates an ECDSA signing key pair (ssk_R, spk_R). For a time period, $\mathcal{R}$ sets the amount of cryptocurrency one can transfer as v_{limit}, and the limit of total number of transactions one can conduct to be n_{limit}.

When a user $\mathcal{U}$ registers with $\mathcal{R}$, they interact as follows:

1) $\mathcal{U}$ chooses a random number $r_p \in Z_p$ and calculates $PID_1 = g^{r_p}$ and $R_p = g_1^{r_p}$. Then, $\mathcal{U}$ generates a proof π_r for the following statement:

$$PK\{(r_p) : PID_1 = g^{r_p} \wedge R_p = g_1^{r_p}\}.$$

   $\mathcal{U}$ also creates an encryption public-private key pair (epk, esk). After that, $\mathcal{U}$ sends PID_1, R_p, π_r, epk, and its real identity RID to $\mathcal{R}$. Here, R_p is used for initial counter generation.
2) $\mathcal{R}$ chooses a $V \in Z_p$, which represents a period and is public to all validators and users. If RID and π_r are valid, $\mathcal{R}$ generates a pseudo-identity $PID = \{PID_1, PID_2\}$ for $\mathcal{U}$, where

$$PID_2 = RID \oplus H(PID_1^{\sigma}, V).$$

3) $\mathcal{R}$ randomly selects $\xi_0, \phi_0 \in Z_p$ and calculates the initial counter of $\mathcal{U}$ as $cID_0 = g_1^{r_p} g_2^{V} g_5^{\xi_0} h^{\phi_0}$. Note that a counter of $\mathcal{U}$ in epoch V has the form

$$cID = g_1^{r_p} g_2^{V} g^{\theta} g_4^{\mu} g_5^{\xi} h^{\phi},$$

   where $\theta, \mu, \xi, \phi \in Z_p$. θ denotes the accumulated transferred amount in epoch V, μ denotes the accumulated number of transfers in epoch V. ξ and ϕ are used for randomization. Since this is the initial counter, the transferred amount and the number of transfers for the user are 0. Thus, the initial counter obtained by

TABLE 1
Parameters in the System

Acronym	Definition
$\mathbf{c}$	Coin
v	Coin value
v_{pub}	Transaction fee
G	Elliptic curve group of order p
$g, g_1, g_2, g_3, g_4, g_5, \bar{g}, h$	Generators of group G
$\mathbb{G}$	Elliptic curve group of order q
$\hat{g}, \hat{h}$	Generators of group $\mathbb{G}$
σ	Master secret key of the regulator R
P_{pub}	g^{σ}, public key of R
spk_R, ssk_R	Signing key pair of R
r_p	A random number associated with a user
$PID = \{PID_1, PID_2\}$	Pseudo-identity of a user
RID	Real identity of a user
V	A time period
cID	Transaction counter of a user
cID_{sn}	Serial number of cID
θ	Accumulated transferred amount
μ	Accumulated number of transfers
$\mathcal{T}_{cID}$	MHT built based on $cIDs$
a_{pk}, a_{sk}	Address public key and private key
sn	Serial number of a coin
cm	Coin commitment
$\mathcal{T}_{coin}$	MHT built based on coin commitments
s, r	Random numbers used to construct a coin
ρ	Random number used to construct a serial number
pk_{sig}, sk_{sig}	One time signing key pair
v_1^{old}, v_2^{old}	Input values of a pour transaction
v_1^{new}, v_2^{new}	Output values of a pour transaction

the user is $cID_0 = g_1^{r_p} g_2^{V} g_5^{\xi_0} h^{\phi_0}$. $\mathcal{R}$ also encrypts ξ_0, ϕ_0 with epk, and sends the ciphertext to $\mathcal{U}$.

4) $\mathcal{R}$ signs PID and cID_0 with ssk, and obtains the signature $Sig(PID)$ and $Sig(cID_0)$. $\mathcal{R}$ sends the $(PID, Sig(PID), cID_0, Sig(cID_0))$ to $\mathcal{U}$, where $(PID, Sig(PID))$ is a pseudo-identity certificate PIC for $\mathcal{U}$, and $(cID_0, Sig(cID_0))$ is the initial counter certificate for $\mathcal{U}$.
5) $\mathcal{R}$ builds a MHT tree $\mathcal{T}_{cID}$ on the $cIDs$, and sends the authentication path of cID_0 to $\mathcal{U}$. $\mathcal{R}$ also sends $(cID_0, Sig(cID_0))$ to validators, who also maintain the MHT tree $\mathcal{T}_{cID}$ on $cIDs$ for transaction verification.

Note that $\mathcal{U}$ can only register once in a period. $\mathcal{U}$ generates its address secret key and public key (a_{sk}, a_{pk}) by choosing a random seed $a_{sk} \in \{0,1\}^*$, and computes $a_{pk} = \mathrm{PRF}^{addr}_{a_{sk}}(r_p)$. $\mathcal{U}$ can generate many address keys by itself.

Mint: When $\mathcal{U}$ mints a coin $\mathbf{c}$ with a value v, it first chooses a random ρ, and calculates the serial number of the coin as $sn = \mathrm{PRF}^{sn}_{a_{sk}}(\rho)$. Then, to generate the coin, $\mathcal{U}$ computes $k = \mathrm{COM}_r(a_{pk}||\rho)$ for a random r, and $cm = \mathrm{COM}_s(v||k)$ for a random s, where cm is the coin commitment. The mint transaction tx_{Mint} is $tx_{Mint} = (cm, s, v, k)$. Moreover, $\mathcal{U}$ needs to deposit v Bitcoin to make the mint transaction to be accepted.

Validators check whether $cm = \mathrm{COM}_s(v||k)$, and if the equation holds, tx_{Mint} is considered valid and can be published on blockchain. Validators build a Merkle tree $\mathcal{T}_{coin}$ over the list of coin commitments using CRH, where leaf nodes are coin commitments of minted coins. The minted coin $\mathbf{c}$ is denoted as $\mathbf{c} = (a_{pk}, \rho, r, s, v, cm)$, and is kept by $\mathcal{U}$.

Pour: When $\mathcal{U}$ transfers coins to others, it needs to create a pour transaction. Without loss of generality, we assume the number of input coins and output coins is 2. The values of input coins and output coins are denoted as v_1^{old}, v_2^{old} and v_1^{new}, v_2^{new}. Similar to Zerocash, for $i \in \{1,2\}$, $\mathcal{U}$ chooses a random number ρ_i^{new}, and generates the coin as $k_i^{new} = \mathrm{COM}_{r_i^{new}}(a_{pk_i}^{new}||\rho_i^{new})$, and $cm_i^{new} = \mathrm{COM}_{s_i^{new}}(v_i^{new}||k_i^{new})$ with a random r_i^{new}, the address public key of recipient pk_i, and a random s_i^{new}. The new generated two coins are

$$\mathbf{c}_i^{new} = (a_{pk,i}^{new}, \rho_i^{new}, r_i^{new}, s_i^{new}, v_i^{new}, cm_i^{new}).$$

$\mathcal{U}$ generates the encryption of $(\rho_i^{new}, r_i^{new}, s_i^{new}, v_i^{new})$ using the public encryption key of receiver $\mathcal{R}_i$, which is included in the transaction.

Let rt be the current root of Merkle tree $\mathcal{T}_{coin}$, and $path$ be the authentication path from a coin commitment to the root of $\mathcal{T}_{coin}$. A pour transaction includes $tx_{Pour} = (rt, sn_1^{old}, sn_2^{old}, cm_1^{new}, cm_2^{new}, v_{pub}, *)$, where sn_1^{old} is used to prevent double-spending of coins, and * denotes other auxiliary information, such as a zero-knowledge proof and the corresponding regulator of $\mathcal{U}$. v_{pub} is the value that denotes transaction fees. To guarantee the non-malleability of a pour transaction, $\mathcal{U}$ generates a signing key pair (pk_{sig}, sk_{sig}) for a one-time signature scheme. Then, $\mathcal{U}$ calculates $h_{Sig} = CRH(pk_{sig})$, and computes the values $h_1 = PRF^{pk}_{a^{old}_{sk,1}}(h_{Sig})$ and $h_2 = PRF^{pk}_{a^{old}_{sk,2}}(h_{Sig})$. $\mathcal{U}$ uses sk_{sig} to sign the pour transaction, and obtain a signature σ_{Pour}. Values $h_{Sig}, h_1, h_2, \sigma_{Pour}$, and pk_{sig} are included in tx_{Pour}.

To prove that a pour transaction tx_{Pour} is valid and complies with regulatory policies, two proofs are included in tx_{Pour}, which are π_1 and π_2. Given the witness

$$\omega = (c_1^{old}, c_2^{old}, c_1^{new}, c_2^{new}, addr_{sk,1}^{old}, addr_{sk,2}^{old}, path_1, path_2)$$

and the instance tr_{Pour}, π_1 proves the following statement:

- c_1^{old} and c_2^{old} are minted coins, i.e., $path_1$ and $path_2$ are valid authentication paths for cm_1 and cm_2.
- $\mathcal{U}$ owns c_1^{old} and c_2^{old}, i.e., $a_{pk_i}^{old} = \mathrm{PRF}^{addr}_{a^{old}_{sk,i}}(r_p)$.
- sn_1^{old} and sn_2^{old} are serial number of c_1^{old} and c_1^{old}, i.e., for $i \in \{1,2\}$, $sn_i^{old} = \mathrm{PRF}^{sn}_{a^{old}_{sk,i}}(\rho_i^{old})$.
- all the coin commitments are well-formed, i.e., for $i \in \{1,2\}$,

  $$cm_i^{new} = \mathrm{COM}_{s_i^{new}}(\mathrm{COM}_{r_i{}^{new}}(a_{pk,i}^{new}||\rho_i^{new})v_i^{new}||).$$

  Same as cm_i^{old}.
- The total number of input coins is equal to output values, i.e., $v_1^{old} + v_2^{old} = v_1^{new} + v_2^{new} + v_{pub}$.
- The address secret key $a_{sk,i}^{old}$ ties h_{Sig} to h_i, i.e., $h_i = PRF^{pk}_{a^{old}_{sk}}(h_{Sig})$.

The second proof π_2 is generated for regulatory policy enforcement. For a pour transaction, the user needs to prove the following statements in π_2:

1) The counter in the transaction belongs to the user.
2) The counter is a leaf node of the tree $\mathcal{T}_{cID}$.

3) The counter is updated correctly based on the transaction.
4) The common inputs values in π_2 and π_1 are the same.
5) If the transaction complies with the polices, the user proves that the transferred amounts and the number of transfers are within the limits in the time period. Otherwise, the user includes the pseudo-identity certificate PIC in the transaction and proves that the certificate belongs to him/her.

To be specific, $\mathcal{U}$ creates the proofs for the corresponding statements as follows:

1) $\mathcal{U}$ needs to prove that the counter in the transaction belongs to him/her. Assume the current counter is of the form $cID = g_1^{r_p} g_2^V g_3^\theta g_4^\mu g_5^\xi h^\phi$, where θ, μ, ξ and ϕ are known to $\mathcal{U}$. Let the coordinate values of cID be (cID_x, cID_y). $\mathcal{U}$ computes the commitment of cID as $C_1 = Com_q(cID_x)$, and $C_2 = Com_q(cID_y)$ [35]. Given $cID_{sn} = g_1^{r_p} g_2^V g_3^\theta g_4^\mu g_5^\xi$, which is the serial number of cID, and the commitment of cID, $\mathcal{U}$ proves that

$$\begin{aligned} PK\{(r_p, \theta, \mu, \phi, cID) : \\ C_1 = Com_q(cID_x) \wedge \\ C_2 = Com_q(cID_y) \wedge \\ cID = cID_{sn} h^\phi \wedge \\ cID_{sn} = g_1^{r_p} g_2^V g_3^\theta g_4^\mu g_5^\xi \wedge \\ C_{r_p} = Com_p(r_p)\}. \end{aligned} \tag{1}$$

The construction of the proof for this statement is shown in Appendix.

2) $\mathcal{U}$ proves that the counter is a valid counter, i.e., the counter is a leaf node of the tree $\mathcal{T}_{cID}$. Given $cID = (cID_x, cID_y)$ and the commitment $(C_1, C_2) = (Com_q(cID_x), Com_q(cID_y))$ of cID, $\mathcal{U}$ proves that cID is in the tree $\mathcal{T}_{cID}$, and the input cID is equal to cID in the commitment. Let $cID_1 = cID_x, cID_2 = cID_y$. The user utilizes a zk-SNARK scheme to prove that cID is in the tree $\mathcal{T}_{cID}$. Then, the user proves

$$\begin{aligned} PK\{(cID_1, cID_2, \eta) : y' = H^{(1/2)\eta} \prod_{i=1}^{2} G_i^{cID_i} \wedge \\ y_1 = Com_q(cID_1) \wedge \\ y_2 = Com_q(cID_2)\ \}. \end{aligned} \tag{2}$$

Here, H and $G_i, i \in \{1, 2\}$ are included in the CRS of the zk-SNARK (Section 4.2 of [35]) and G_i corresponds to cID_i. $\eta \in Z_p$ is a random number used to achieve zero-knowledge SNARK. Note that this proof can be achieved by using non-interactive zero-knowledge proofs for composite statements [35].

3) $\mathcal{U}$ proves that the counter is updated correctly based on the transferred amount. $\mathcal{U}$ first computes $cID' = cID_{sn} \times g_3^{v^{old}} g_4 g_5^{\xi'} h^{t'}$, where $v^{old} = v_1^{old} + v_2^{old}$, and ξ', t' are random numbers in Z_p. Note that this updates the counter by adding the transferred amounts by v^{old} and the number of transfers by 1. Then, $\mathcal{U}$ proves that

$$\begin{aligned} PK\{(r_p, v^{old}, \xi', t', v_1^{old}, v_2^{old}, t_1, t_2, t_3) : \\ Com_p(v_1^{old}) = \bar{g}^{v_1^{old}} h^{t_1} \wedge \\ Com_p(v_2^{old}) = \bar{g}^{v_2^{old}} h^{t_2} \wedge \\ Com_p(v^{old}) = \bar{g}^{v^{old}} h^{t_3} \wedge \\ v^{old} = v_1^{old} + v_2^{old} \wedge \\ cID' = cID_{sn} \times g_3^{v^{old}} g_4 g_5^{\xi'} h^{t'}\}. \end{aligned} \tag{3}$$

4) $\mathcal{U}$ proves that the common inputs values in two zero-knowledge proofs π_1 and π_2 are the same. Given the commitments of v_1^{old}, v_2^{old}, and r_p, $\mathcal{U}$ proves that v_1^{old}, v_2^{old} and r_p used in π_2 are same as the ones used in π_1. Let $a_1 = v_1^{old}, a_2 = v_2^{old}$ and $a_3 = r_p$. It proves

$$\begin{aligned} PK\{(a_1, a_2, a_3, \eta_1, t_1', t_2', t_r) : \\ y' = H^{(1/2)\eta_1} \prod_{j=1}^{3} G_j^{a_j} \wedge \\ Com_p(a_1) = \bar{g}^{a_1} h^{t_1} \wedge \\ Com_p(a_2) = \bar{g}^{a_2} h^{t_2} \wedge \\ Com_p(a_3) = \bar{g}^{r_p} h^{t_{r_p}}\}. \end{aligned} \tag{4}$$

Here, H and $G_j, j \in \{1, 2, 3\}$ are included in the CRS of the zk-SNARK (Section 4.2 of [35]) and G_j corresponds to a_j. η_1 is used to achieve zero-knowledge SNARK, and $Com_p(a_3)$ is the same as $Com_p(r_p)$ in step 1.

5) According to whether the transaction violates the regulation policies, $\mathcal{U}$ acts as follows:
   - When the transferred amounts and the number of transfers are within the limit in the time period, $\mathcal{U}$ generates the following proof: Given the current cID', the commitment of the total transferred coins $Com_p(\bar{v})$ in the period, and the commitment of the total number of transfers $Com_p(\bar{n})$ in the period, $\mathcal{U}$ proves that

$$\begin{aligned} PK\{(\bar{v}, \bar{n}, r_p, \bar{\xi}, \bar{t}, t_{\bar{v}}, t_{\bar{n}}) : \\ cID' = g_1^{r_p} g_2^V g_3^{\bar{v}} g_4^{\bar{n}} g_5^{\bar{\xi}} h^{\bar{t}} \wedge \\ Com_p(\bar{v}) = \bar{g}^{\bar{v}} h^{t_{\bar{v}}} \wedge \\ Com_p(\bar{n}) = \bar{g}^{\bar{n}} h^{t_{\bar{n}}} \wedge \\ \bar{v} \in [0, v_{limit}] \wedge \\ \bar{n} \in [0, n_{limit}]\}. \end{aligned} \tag{5}$$

     The range proof of $\bar{v}$ and $\bar{n}$ can be generated using bulletproof [36].
   - When the accumulated transferred value or number of transfers exceeds the predefined limit, $\mathcal{U}$ adds the pseudo-identity certificate PIC to tx_{Pour}, and prove that

$$PK\{r_p : PID_1 = g^{r_p}\}.$$

     The proof binds $\mathcal{U}$'s real identity to the tx_{Pour} and can be generated by using sigma protocol.

Verify: Given the public parameters, transaction tx_{Pour}, and the proof in the transaction, validators verify the correctness and regulation compliance of the transaction before it can be added to the ledger. If the total amount of

transferred coins and total number of transfers are within the corresponding limits, validators add cID' in tree $\mathcal{T}_{cID}$. The cID_{sn} prevents a user from using a previous counter. For the subsequent transactions, $\mathcal{U}$ needs to prove that cID' is in the $\mathcal{T}_{cID}$ given the commitment of cID'. If the transaction violates the regulation policies, validators send the transaction to the corresponding regulator, who will investigate and deal with it accordingly.

Receive: Given the current ledger, the ciphertext sent from $\mathcal{U}$, the decryption key of $\mathcal{R}_i$, the receiver first checks whether ρ_i^{new} is different from the ρ received before. If not, $\mathcal{R}_i$ notifies $\mathcal{U}$ and requires a replacement of ρ_i^{new}, since the same ρ will result in a same serial number. Otherwise, $\mathcal{R}_i, i \in \{1,2\}$, outputs the new coin $\mathbf{c}_i^{new} = (a_{pk,i}^{new}, \rho_i^{new}, r_i^{new}, s_i^{new}, v_i^{new}, cm_i^{new})$. The new coin can be spent by using pour transaction.

Trace: When receiving a pour transaction tx_{Pour}, a regulator parses the pseudo-identity certificate PIC as $\{PID, Sig(PID)\}$ and PID as $\{PID_1, PID_2\}$. The regulator first verifies the validity of the signature and the zero-knowledge proof of $PK\{r_p : PID_1 = g^{r_p}\}$. If it is valid, the regulator calculates the real ID of the user as $RID = PID_2 \oplus H(PID_1^q, V)$.

6 Security Proof

In this section, we prove the security of the proposed RDAP, i.e., showing that the proposed RDAP satisfies the anonymity, authentication, balance, and traceability.

Theorem 1. *RDAP achieves* Anonymity *if the zk-SNARK scheme is zero knowledge, the encryption scheme Enc is indistinguishable under the chosen plaintext attacks (IND-CPA), the COMM scheme is statistically-hiding, and the pseudorandom function PRF is indistinguishable from a random function.*

Proof. The anonymity of our scheme can be proved by ledger indistinguishability, i.e., the ledger reveals no information about users other than the publicly-revealed information, such as the total number of transactions, public strings in transactions, and values of minted coins.

We construct a sequence of games (G_{real}, G_1, G_2, G_3, G_{sim}), in which $\mathcal{C}$ makes a modification of the original game G_{real} defined in the security model. We will show that the difference between the advantages of $\mathcal{A}$ in G_{real} and in G_{sim} is negligible. □

In game G_1, after sampling $b \in \{0,1\}$, $\mathcal{C}$ modifies the G_{real} by using a simulator $\mathcal{S}$ for simulating Key generation and Proof generation of zk-SNARKs. Instead of invoking KeyGen(λ, C) of zk-SNARKs, $\mathcal{C}$ uses $\mathcal{S}$ to generate the proving key pk, verification key vk, and a trapdoor $trap$. C sends the public parameters to $\mathcal{A}$, and initializes two RDAP oracles $\mathcal{O}_0^{RDAP}$ and $\mathcal{O}_1^{RDAP}$. For the Prove algorithm in zk-SNARKs, $\mathcal{S}$ generates the proof π using the statement and $trap$ without utilizing the witness w. Since the zk-SNARK scheme is zero knowledge, the distributions of proofs generated by $\mathcal{S}$ and Prove algorithm are identical. Thus, the difference between the G_1 and G_{real} is zero.

In game G_2, $\mathcal{C}$ modifies G_1 by replacing the ciphertexts that are generated by using the public keys of the receivers with the encryptions of random strings. Specifically, when $\mathcal{A}$ submits a Pour query, in which the output addresses $(addr_{pk,1}^{new}, addr_{pk,2}^{new})$ are previous generated by CreatAddress queries. For each oracle, it creates the ciphertexts as follows: First, the address public keys are replaced by random generated public keys; Second, the plaintext to be encrypted is replaced by a random string chosen form the plain space. let q_P be the total number of Pour queries sent by $\mathcal{A}$. If $\mathcal{A}$ has the advantage ϵ_1 in Enc's IND-CPA experiments, the difference between game G_2 and G_1 is at most $4 \cdot q_P \cdot \epsilon_1$.

In game G_3, $\mathcal{C}$ modifies G_2 by replacing the generated PRF values with random strings. Specifically, for a CreatAddress query, the public key address returned is a random string of the same length. For the Pour query, the serial numbers sn_1^{old} and sn_2^{old} generated are also replaced by random strings of the same length. Let q_{CA} be the number of the CreatAddress queries sent by $\mathcal{A}$. Assume the advantage of $\mathcal{A}$ in distinguishing a PRF output from a random one is ϵ_2. The difference between the game G_3 and G_2 is at most $q_{CA} \cdot \epsilon_2$.

In game G_4, for the Register queries, $\mathcal{C}$ modifies G_3 by replacing the generated PID_2 with a random string of the same length. Let q_R be the number of the Register queries sent by $\mathcal{A}$. Assume the advantage of $\mathcal{A}$ in the discrete logarithm experiments is ϵ_3. The difference between the game G_4 and G_3 is at most $q_R \cdot \epsilon_3$.

In game G_{sim}, $\mathcal{C}$ modifies G_4 by replacing the generated coin commitment with the commitment of a random string. Specifically, for the Mint queries, the $v||k$ is substituted with a random input of the same length. For the Pour queries, if the output address $addr_{pk,j}^{new}$, $j \in \{1,2\}$ belongs to the address set generated by $\mathcal{C}$, cm_j^{new} is replaced by a commitment to a random string of the same length. Let q_M be the number of Mint queries and ϵ_4 be the advantage of $\mathcal{A}$ against the hiding property of COMM. The difference between the game G_{sim} and G_4 is at most $(q_M + 4 \cdot q_P) \cdot \epsilon_4$ [7].

Since in game G_{sim}, the responses and ledgers shown to $\mathcal{A}$ are independent to b, the advantage of $\mathcal{A}$ in G_{sim} is 0. As a result, the advantage of $\mathcal{A}$ in G_{real} is

$$Adv_A \leq 4 \cdot q_P \cdot \epsilon_1 + q_{CA} \cdot \epsilon_2 + q_R \cdot \epsilon_3 + (q_M + 4 \cdot q_P) \cdot \epsilon_4$$

Theorem 2. *The proposed RDAP achieves* Authentication *if the signature scheme used is existential unforgeable under the chosen message attacks (EUF-CMA) and the hash function CRH is collision-resistant.*

Proof. Since the anonymity and unlinkability need to be guaranteed in the scheme, where unlinkability means the transactions that belong to the same user are unlinkable. The pseudo-identity and the address public key of a user should not be included in a transaction. Nevertheless, the counter ID cID is verified by validators in Verify process, which can ensure that the user is a legitimate user who have registered with the local regulator, and the regulator can determine its jurisdiction for a transaction. □

For a cID, we consider two cases: For the initial counter cID_0, there is a signature $Sig(cID_0)$ that is signed by the regulator at the Register phase. If $\mathcal{A}$ forges a signature that can pass the verification, $\mathcal{A}$ can be invoked to attack the unforgeability of the signature scheme. For a subsequent cID, given the pubic key of the regulator, validators can check whether cID is in the tree $\mathcal{T}_{cID}$ that corresponds to the

regulator. If $\mathcal{A}$ can forge the proof and pass the verification, $\mathcal{A}$ can be used to break the collision resistance of the hash function. Thus, authentication of RDAP is proved.

Theorem 3. *RDAP achieves* Balance *if the commitment scheme COMM is computationally binding, the hash function CRH is collision resistant, the zk-SNARK scheme is sound, and the PRF is indistinguishable from a random function.*

Proof. To prove the balance property, we modify GAME balance defined in the security model by letting challenger $\mathcal{O}$ obtain an augmented ledger where for each pour transaction, besides the instance $x = (rt, sn_1^{old}, sn_2^{old}, cm_1^{new}, cm_2^{new}, v_{pub}, *)$, a witness $\omega = (c_1^{old}, c_2^{old}, c_1^{new}, c_2^{new}, addr_{sk,1}^{old}, addr_{sk,2}^{old}, path_1, path_2)$ is also attached. But for $\mathcal{A}$, the view of the ledger is not changed. We denote the augmented ledger as $(L, \vec{a})$, where a_i is the witness for the ith pour transaction instance x_i. Note that for the transactions that are generated by O^{RDAP}, $\mathcal{C}$ can obtain the witness just by asking the O^{RDAP}. For the Pour transactions that are created by $\mathcal{A}$ and inserted in the ledger, the corresponding witness can be obtained by using the knowledge extractor of the zk-SNARK. □

We say that the balance property holds for a ledger $(L, \vec{a})$ if the following conditions are met.

- For a pour transaction tx_{Pour}, the distinct old coin commitments cm_1^{old} and cm_2^{old} open to two different values. Moreover, the old commitments are the outputs of mint or pour transactions that precedes tx_{Pour} on L.
- No two pour transactions contain different openings of a same coin commitment.
- For the v_1^{old}, v_2^{old} of input values and $v_1^{new}, v_2^{new}, v_{pub}$ of output values, the condition $v_1^{old} + v_2^{old} = v_1^{new} + v_2^{new} + v_{pub}$ holds.
- For each tx_{Pour} and its witness a, if cm_i^{old}, $i \in \{1, 2\}$ is the output of a mint transaction tx_{Mint}, the public value v in tx_{Mint} is equal to v_i^{old} in a. If cm_i^{old} is the output of another pour transaction tx'_{Pour}, the opening v' to cm_i^{old} is equal to v_i^{old}.
- For the pour transactions generated by $\mathcal{A}$, if cm_i^{old}, $i \in \{1, 2\}$ is the output of a mint or pour transaction tx, the public address of the receiver for tx belongs to $\mathcal{A}$, which means $\mathcal{A}$ can only spend the coins minted or received by itself.

If Ledger L is not balanced, 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 show the contradictions with the assumption.

If condition 1 does not hold, it means that $cm_1^{old} = cm_2^{old}$, or cm_i^{old} is not on the ledger. Since for a pour transaction, validators will verify that the two serial numbers are different. If $cm_1^{old} = cm_2^{old}$, the fact that $sn_1^{old} \neq sn_2^{old}$ means that there two openings of cm_1^{old}, one is derived from ρ_1^{old}, and the other is derived from ρ_2^{old}, which violates the binding property of COMM scheme. If cm_i^{old} is not on the ledger precedes tx_{Pour}, there does not exist a valid authentication path which can prove that cm_i^{old} is unspent. If the authentication path passes the verification, there exists a collision for the hash function CRH, which violates the collision resistance of CRH.

If condition 2 does not hold, it implies that there are two pour transactions tx_{Pour} and tx'_{Pour} that spend the same coin cm twice, i.e., cm can be opened to two different values, which corresponds to two serial numbers. This contradicts the binding property of COMM scheme.

If condition 3 does not hold, it means that $v_1^{old} + v_2^{old} \neq v_1^{new} + v_2^{new} + v_{pub}$, which will be checked by validators during Verify process. If the inequality holds, the soundness of the zk-SNARK is violated.

If condition 4 does not hold, L contains a pour transaction that opens cm_{old} to a value v_{old}. But for the mint or pour transaction that corresponding to cm_{old} that precedes tx_{Pour} on L, the transaction opens cm_{old} to a value v'_{old}. This breaks the binding property of COMM scheme.

If condition 5 does not hold, it implies that $\mathcal{A}$ can spend a coin that belongs to an honest user whose $a_{pk} = PRF_{ask}^{addr}(r_p)$. Since the witness a of the pour transaction tx_{Pour} inserted by $\mathcal{A}$ contains an ask and r_p that can generate a_{pk}, the security of PRF is violated. Thus, RDAP achieved the balance property of the ledger.

Theorem 4. *The proposed RDAP achieves* Traceability *if the signature scheme is existential unforgeable under the chosen message attacks (EUF-CMA) and the zero-knowledge proofs are sound.*

Proof. If a transaction violates the defined regulatory policies, the adversary cannot generate a valid range proof, otherwise, it can be utilized to break the soundness of the range proof scheme. Therefore, a pseudo-identity certificate is included in a transaction when the transaction does not comply with policies. Since the pseudo-identity certificate is signed by the local regulator, by verifying the signature and the sigma proof which proves that the user knows r_p, the regulator ensures that the user is a legitimate user that previously registered. Based on the pseudo-identity certificate and its secret key, the regulator can recover the user's real identity. If $\mathcal{A}$ can forge the signature and pass the verification, $\mathcal{A}$ can be utilized to break the unforgeability of the underlying signature scheme. Moreover, since the initial counter cID of a user is also signed by the regulator and included in the tree $\mathcal{T}_{cID}$, and the updated cID must be calculated based on the initial counter, the user cannot get around the regulation policies by generating multiple address public-private keys. □

7 Performance Evaluation

To evaluate the performance of RDAP, we conduct simulations and analysis on a computer with Intel(R) Core(TM) i7-7500U of 2.9 GHz and 8GB RAM. We use Barreto-Naehrig (BN) curve as the elliptic curve whose order is 256-bit length. The secure hash functions used in the scheme are SHA-256 hash functions. As in [7], we instantiate the COMM and PRF functions by using SHA-256 hash functions.

Compared with Zerocash, we add the algorithm of Register and Trace, and a zero-knowledge proof π_2 in a pour transaction, which is used for regulation compliance. A user needs to prove that the accumulated transferred value is within the predefined limit, or else, its pseudo-identity certificate needs to be involved in the transaction to help

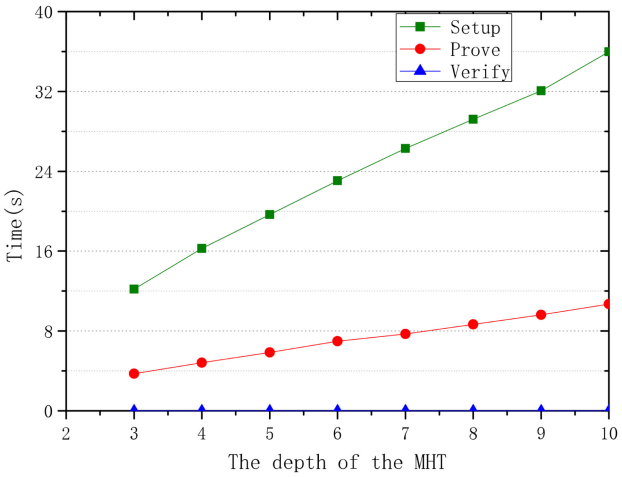


Fig. 2. Computation cost for the MHT proof.

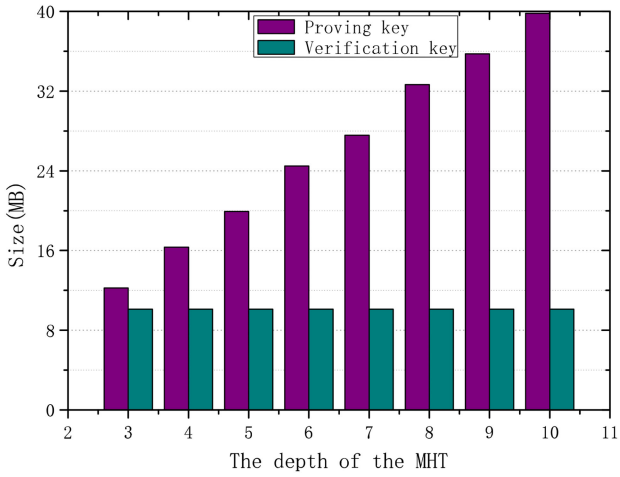


Fig. 3. Storage cost for the MHT proof.

recover the identity of the user. We simulate and analyze the overhead of the added algorithms in RDAP. The details are shown as follows.

For the Register algorithm, a user needs to compute PID_1 and R_p, and a zero-knowledge proof which proves that $PK\{(r_p) : PID_1 = g^{r_p} \wedge R_p = g_1^{r_p}\}$. Moreover, the user also generates its address public-private key pair and encryption public-private key pair. The overall computation cost for the user is 5.95 ms. For the regulator, it first verifies the zero-knowledge proof, and then generates PID_2, and an initial counter cID for the user. After that, the regulator signs $PID = \{PID_1, PID_2\}$, cID, and sends the pseudo-identity certificate and initial counter certificate to the user. In Register phase, the time cost for the regulator is 9.51 ms.

For the Pour algorithm, we add a zero-knowledge proof π_2, which is related to regulatory policy enforcement. For π_2, we simulate the time overhead for both users and validators. Note that in step 2, the user needs to prove that cID is in the tree $\mathcal{T}_{cID}$. We simulate the corresponding MHT proof generation and verification by invoking the libsnark library [37]. We test the time cost of key generation, proof generation, and verification for the MHT proof when the number of the leaf nodes changes from 8 to 1024, i.e., the depth of the tree grows from 3 to 10. The experiment results are shown in Fig. 2, from which we can observe that the computation time grows linearly with the depth of the MHT, due to the number of nodes on the authentication path is the same as the depth of the tree. The time cost for creating the proving key and verification key for the MHT proof in π_2 is about 36 s when there are 1024 leaf nodes. As the key generation is performed only once, the time cost is acceptable.

In π_2, given all the leaf nodes and a specific cID, the user needs to generate a zk-SNARK proof. Based on the root of $\mathcal{T}_{cID}$ and the proof, a verifier can check whether a leaf node is in the tree. From Fig. 2, we can see that when there are 1024 leaf nodes, it costs 10.7 s to generate a proof. The verification for a proof is fast, and the time spent stays at a constant 0.007 s when the depth of $\mathcal{T}_{cID}$ varies. We also evaluate the storage cost of the proving key, the verification key, and the generated proof when the depth of $\mathcal{T}_{cID}$ changes from 3 to 10. Results show that the sizes of the verification key and the generated proof are fixed, which are 10.12 KB and 128 bytes, respectively. From Fig. 3, we can see that the size of the proving key increases as the depth of $\mathcal{T}_{cID}$ grows. When the depth of the tree is 10, the size of the generated proving key is 39.8 MB. For the sigma proof in π_2, we analyze the time overhead of different sigma proofs based on the cryptographic operations involved. For the third statement in π_2, the time cost for a user to generate the sigma proof is about 7.88 ms, and a validator needs about 14.03 ms to verify the proof. For the fourth statement in π_2, it takes 8.76 ms for a user to create the corresponding sigma proof, and the time cost for a validator to verify the sigma proof is about 12.28 ms.

For the Trace algorithm, a regulator first verifies the validity of the signature and the proof of $PK\{(r_p) : PID_1 = g^{r_p}\}$, which costs 2.7 ms. If they are valid, it takes 3.18 ms for the regulator to obtain the real identity of the user. Hence, the whole cost for the regulator is 5.88 ms in the Trace phase.

From the above experiment results, we can see that the designed RDAP can achieve efficient proof generation and verification for regulatory enforcement. Moreover, when suspicious transactions are discovered, only small overheads are required for regulators to recover users' identities.

8 Conclusion

In this paper, we have proposed a regulated and decentralized anonymous payment scheme (RDAP) where regulatory policies can be enforced and users' privacy can be protected. Payers and payees of transactions and the transferred amounts are hidden from others, while a user's total amount of transferable cryptocurrency and the number of transactions in a time period are limited. The validity and regulatory compliance of transactions are guaranteed by zero-knowledge proofs. RDAP achieves anonymity, authentication, balance, traceability, and may shed light on the further research on decentralized anonymous payment with regulatory compliance. For the future work, we aim to design a regulated cryptocurrency scheme that can support additional regulatory policies such as tax enforcement.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, p. 21 260, 2008.

[2] H. Wang, Q. Wang, and D. He, "Blockchain-based private provable data possession," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2379–2389, Sep./Oct. 2019.

[3] D. E. Kouicem, Y. Imine, A. Bouabdallah, and H. Lakhlef, "A decentralized blockchain-based trust management protocol for the Internet of Things," *IEEE Trans. Dependable Secure Comput.*, 2020.

[4] Website, "CoinMarketCap," [Online]. Available: https://coinmarketcap.com/

[5] M. Conti, E. S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of bitcoin," *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 3416–3452, Fourth Quarter 2018.

[6] X. Shen *et al.*, "Data management for future wireless networks: Architecture, privacy preservation, and regulation," *IEEE Netw.*, vol. 35, no. 1, pp. 8–15, Jan./Feb. 2021.

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

[8] S.-F. Sun, M. H. Au, J. K. Liu, and T. H. Yuen, "Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2017, pp. 456–474.

[9] C. Garman, M. Green, and I. Miers, "Accountable privacy for decentralized anonymous payments," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2016, pp. 81–98.

[10] L. Xue, D. Liu, J. Ni, X. Lin, and X. Shen, "Balancing privacy and accountability for industrial mortgage management," *IEEE Trans. Ind. Inform.*, vol. 16, no. 6, pp. 4260–4269, Jun. 2020.

[11] S. Mabunda, "Cryptocurrency: The new face of cyber money laundering," in *Proc. Int. Conf. Adv. Big Data Comput. Data Commun. Syst.*, 2018, pp. 1–6.

[12] E. Fletcher, C. J. Larkin, and S. Corbet, "Cryptocurrency regulation: Countering money laundering and terrorist financing," 2020. [Online]. Available: https://ssrn.com/abstract=3704279

[13] H. Nabilou, "How to regulate bitcoin? decentralized regulation for a decentralized cryptocurrency," *Int. J. Law Informat. Technol.*, vol. 27, no. 3, pp. 266–291, 2019.

[14] T. S. Sobh, "An intelligent and secure framework for anti-money laundering," *J. Appl. Secur. Res.*, vol. 15, no. 4, pp. 517–546, 2020.

[15] M. Petkus, "Why and how zk-snark works," 2019, *arXiv: 1906.07221*.

[16] Y. Lindell, "An efficient transform from sigma protocols to nizk with a CRS and non-programmable random oracle," in *Proc. Theory Cryptogr. Conf.*, 2015, pp. 93–109.

[17] D. Liu, J. Ni, C. Huang, X. Lin, and X. Shen, "Secure and efficient distributed network provenance for IoT: A blockchain-based approach," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7564–7574, Aug. 2020.

[18] N. Van Saberhagen , "Cryptonote v 2.0," 2013.

[19] G. Maxwell, "CoinJoin: Bitcoin privacy for the real world," *Post Bitcoin Forum*, 2013.

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

[21] I. Damgård and E. Fujisaki, "A statistically-hiding integer commitment scheme based on groups with hidden order," in *Proc. Int. Conf. Theory Appl. Cryptol. Informat. Secur.*, 2002, pp. 125–142.

[22] Y. Wu, H. Fan, X. Wang, and G. Zou, "A regulated digital currency," *Sci. China Informat. Sci.*, vol. 62, no. 3, 2019, Art. no. 32109.

[23] T. Ma, H. Xu, and P. Li, "SkyEye: A traceable scheme for blockchain," *IACR Cryptol. ePrint Arch.*, vol. 2020, 2020, Art. no. 34.

[24] K. Wüst, K. Kostiainen, V. Čapkun, and S. Čapkun, "Prcash: Fast, private and regulated transactions for digital currencies," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2019, pp. 158–178.

[25] 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 Security*, vol. 15, pp. 2440–2452, 2020.

[26] P. Chatzigiannis, F. Baldimtsi, and K. Chalkias, "SoK: Auditability and accountability in distributed payment systems," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, 2021, pp. 311–337.

[27] K. Chalkias, K. Lewi, P. Mohassel, and V. Nikolaenko, "Distributed auditing proofs of liabilities," *Cryptol. ePrint Arch.*, 2020.

[28] P. Chatzigiannis and F. Baldimtsi, "MiniLedger: Compact-sized anonymous and auditable distributed payments," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2021, pp. 407–429.

[29] 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, Mar./Apr. 2021.

[30] E. Androulaki, J. Camenisch, A. D. Caro, M. Dubovitskaya, K. Elkhiyaoui, and B. Tackmann, "Privacy-preserving auditable token payments in a permissioned blockchain system," in *Proc. 2nd ACM Conf. Adv. Financial Technol.*, 2020, pp. 255–267.

[31] E. B. Sasson *et al.*, "Zerocash: Decentralized anonymous payments from bitcoin," *IEEE Symp. Secur. Privacy*, 2014.

[32] J. Camenisch, "Group signature schemes and payment systems based on the discrete logarithm problem," Ph.D. dissertation, ETH Zurich, Zürich, Switzerland, 1998.

[33] B. Parno, J. Howell, C. Gentry, and M. Raykova, "Pinocchio: Nearly practical verifiable computation," in *Proc. IEEE Symp. Secur. Privacy*, 2013, pp. 238–252.

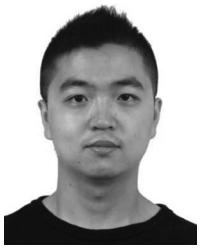
[34] J. Groth, "On the size of pairing-based non-interactive arguments," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Techn.*, 2016, pp. 305–326.

[35] S. Agrawal, C. Ganesh, and P. Mohassel, "Non-interactive zero-knowledge proofs for composite statements," in *Proc. Annu. Int. Cryptol. Conf.*, 2018, pp. 643–673.

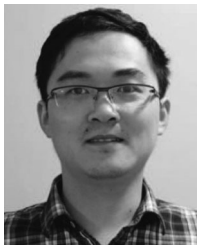
[36] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *Proc. IEEE Symp. Secur. Privacy*, 2018, pp. 315–334.

[37] Website, "StarLI-Trapdoor," [Online]. Available: https://github.com/StarLI-Trapdoor/libsnark_sample

Liang Xue (Student Member, IEEE) received the BS and MS degrees from the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), China, in 2015 and 2018, respectively. She is currently working toward the PhD degree with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. Her research interests include applied cryptography, cloud computing, and blockchain.



Dongxiao Liu (Member, IEEE) received the PhD degree from the Department of Electrical and Computer Engineering, University of Waterloo, Canada, in 2020. He is currently a postdoctoral research fellow with the Department of Electrical and Computer Engineering, University of Waterloo. His research interests include security and privacy in mobile networks and blockchain.



Jianbing Ni (Member, IEEE) received the BE and MS degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2011 and 2014, respectively, and the PhD degree in electrical and computer engineering from the University of Waterloo, Waterloo, Canada, in 2018. He is currently an assistant professor with the Department of Electrical and Computer Engineering, Queen's University. His research interests include applied cryptography and network security, with current focus on cloud computing, smart grid, mobile crowdsensing and Internet of Things.

Xiaodong Lin (Fellow, IEEE) received the PhD degree in information engineering from the Beijing University of Posts and Telecommunications, China, and the PhD degree (with Outstanding Achievement in Graduate Studies Award) in electrical and computer engineering from the University of Waterloo, Canada. He is currently a professor with the School of Computer Science, University of Guelph, Canada. His research interests include computer and network security, privacy protection, applied cryptography, computer forensics, and software security.

Xuemin (Sherman) Shen (Fellow, IEEE) received the PhD degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is currently a professor with the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks. He is a registered professional engineer of Ontario, Canada, an Engineering Institute of Canada fellow, a Canadian Academy of Engineering fellow, a Royal Society of Canada fellow, a Chinese Academy of Engineering foreign member, and a distinguished lecturer of the IEEE Vehicular Technology Society and Communications Society. He received the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory (CSIT) in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society, and Technical Recognition Award from Wireless Communications Technical Committee (2019) and AHSN Technical Committee (2013). He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom'16, IEEE Infocom'14, IEEE VTC'10 Fall, IEEE Globecom'07, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is the President of the IEEE Communications Society. He was the vice president for technical and educational activities, vice president for publications, Member-at-Large on the Board of Governors, chair of the Distinguished Lecturer Selection Committee, member of IEEE fellow Selection Committee of the ComSoc. He served as the editor-in-chief of the *IEEE IoT Journal*, *IEEE Network*, and *IET Communications*.