Invisible Warning Line: Efficient and Generic Regulation for Anonymous Cryptocurrencies

Rui Gao, Zhiguo Wan, Huaqun Wang, Shaoteng Luo

Abstract—Decentralized finance based on blockchain has experienced rapid development. To safeguard the privacy of participants, decentralized anonymous payment (DAP) systems such as ZCash and Zether have emerged. These systems employ cryptographic techniques to conceal the trader addresses and payment amounts. However, this anonymity presents challenges in terms of regulation. To address this issue, we propose the Walsh-DAP (WDAP) scheme, an efficient and generic regulation scheme for decentralized anonymous payments that strikes a balance between regulation and privacy preservation. Our scheme introduces two regulation policies: first, users who have exceeded their spending limits within a certain period will be identified during the regulation process; second, the supervisor possesses the capability to trace any anonymous transaction. To implement regulation effectively, we have designed an innovative commitment scheme, Walsh commitment, which leverages the orthogonal properties of Walsh codes to achieve the features of aggregatability and extractability. The supervisor in WDAP only needs to deal with the aggregation result of the Walsh commitments instead of the huge amount of raw transactions information, which greatly increases the efficiency. In a DAP system with 256 users, 10 transactions per second and 30 days as a regulation period, we reduced the communication cost for regulation from 14 GB to 94.20 KB, and the computing cost from 1.6×10^5 s to 2.17 s. Both improvement is of over five orders of magnitude. We formally discussed the security of the whole system, and verified its feasibility and practicability in the ZCash system.

Index Terms—Regulation, Decentralized anonymous payment, Privacy protection

1 Introduction

Blockchain is a distributed ledger technology used to record and store information about digital transactions. It allows users to conduct secure data exchanges and transactions without a central authority. One of its famous applications is the decentralized payment such as Bitcoin [1] and Etherum [2]. In 2022, a staggering $8.2 trillion was transferred via the Bitcoin blockchain.

Common cryptocurrency systems typically employ the pseudonymous mechanism to conceal the direct link between the address and the user's real identity. Although malicious attackers cannot recover the identities of traders from pseudonyms, the users' real identities can be revealed under attacks such as network analysis [3], address clustering [4], and transaction graph analysis [5] since all information of transactions is publicly available.

In order to protect the privacy of traders, researchers have developed decentralized anonymous payment (DAP) systems such as Monero [6] and ZCash [7], which employ cryptographic techniques to provide privacy and anonymity to participants in financial transactions. DAP allows users to make transactions while concealing their identities and the transferred amount.

The problem. The decentralized nature of cryptocurrencies has made them susceptible to being used as tools for financial crimes, such as money laundering and terrorist financing [8]. The money laundering amount in decentralized payment system has achieved $23.8 billion in 2022, a 68.0% increase over 2021[1].

The issue becomes even more severe in the context of DAP, since the trader identities and the transaction amount are hidden in DAP. Economic criminal activities will become more rampant and harder to be regulated. While DAP offers enhanced privacy, they have also raised concerns related to their potential use in illicit activities.

The existing countermeasures are quite trival that some governments and exchanges banned the cryptocurrencies of DAP systems. Taking Monero as an example, major world economies such as Japan and South Korea have already banned Monero from exchanges in an effort to curb money laundering and reduce organized crime[2]. Many cryptocurrency exchanges have also taken action to end Monero support for similar reasons. Bittrex, BitBay, and Huobi are three of these exchanges.

Challenge. To strike a balance between privacy protection and regulation measures, some researchers [9] introduce a strong supervisor in the DAP system. Only transactions that have been reviewed by the supervisor can be uploaded to the blockchain system. This approach can support a wide range of regulatory strategies but it lowers the decentralized

- *Corresponding authors: Zhiguo Wan, Huaqun Wang.*
- *R. Gao is with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: 501874794@qq.com)*
- *Z. Wan is with the Zhejiang Lab, Hangzhou 310000, China (e-mail: wanzhiguo@zhejianglab.com).*
- *H. Wang is with the Jiangsu Key Laboratory of Big Data Security & Intelligent Processing, College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China, and also with the Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China (e-mail: wanghuaqun@aliyun.com).*
- *S. Luo is with the School of Software Engineering, Hangzhou Institute for Advanced Study, UCAS, Hangzhou 310024, China (e-mail: luoshaoteneg21@mails.ucas.ac.cn)*

∗. This is an extended version of our manuscript.

1. https://blog.chainalysis.com/reports/crypto-money-laundering-2022/

2. https://blog.chainalysis.com/reports/all-about-monero

nature of the blockchain system. To reduce the interactivity of the system, some schemes [10] [11] designed conditional anonymity payment systems that allow the supervisor to recover all privacy hidden in the anonymous transactions.

Both of the two solutions require the supervisor to obtain all complete transactions and extract the privacy information from them. This imposes a significant demand on the supervisor's communication and computational capabilities. The supervisors must keep online and have access to all information in real-time. It is evident that such a regulatory system is unscalable and inefficient.

We consider that this inefficiency stems from the lack of a mechanism for aggregating regulated transactions and regulators have to deal with raw information.

Our approach. We present Walsh-DAP (WDAP), an efficient regulation scheme of DAP system. Concretely, we set two regulation policies:

- Users who have exceeded their spending limits within a certain period will be identified during the regulation process.
- The supervisor can retrieve the identity of the transaction sender. But he cannot retrieve the receiver address nor the transferred value.

The building block of our work is a novel commitment scheme called Walsh commitment, which exploits the orthogonal property of the Walsh codes to achieve aggregatability and extractability. When a user makes a new transaction of transferring value v, he needs to attach a Walsh commitment of transaction amount and prove its validity using a zero-knowledge proof. The miner aggregates all of the commitments when sealing a new block. After a fixed regulation period, each user needs to upload a range proof showing that his total payment is under limit. The supervisor can check if the user has violated the regulatory policies using the aggregation result of all commitments and the range proof sent by users.

Advantages. Our WDAP scheme has the following advantages:

- **Regulation efficiency**: In WDAP, the supervisor only need to deal with the aggregation result of commitments and the range proofs. So the overhead of regulation is only related to the number of users no matter how many transactions there are.

  Our scheme performs excellent when there is a high transaction volume. Compared to the former work DCAP [10], in a DAP system with 256 users, 10 transactions per second and 30 days as a regulation period, we reduced the communication cost for regulation from 14 GB to 94.20 KB, and the computing cost from 1.6×10^5 s to 2.17 s. Both improvement is of over five orders of magnitude.
- **Restricted power**: The supervisor can only retrieve the sender address of an anonymous transaction but he cannot retrieve the payment amount.
- **Generality**: The scheme we proposed is generic and works like a plugin. It can be integrated in almost all native DAP scheme to give it regulatory attribute. In Section 9, we give the instance of integrating the WDAP scheme in the ZCash system.

  As far as we know, our scheme is the the first regulation scheme satisfying generality.
- **Non-interaction**: Users make transactions without interaction with the supervisor, so that the supervisor can keep offline for most of the time.

Our contributions

- We proposed WDAP, an efficient regulation scheme for anonymous cryptocurrencies, which satisfies *regulation efficiency*, *restricted power*, *generality* and *non-interaction*.
- To implement regulation effectively, we have designed an innovative commitment scheme, Walsh commitment, which leverages the orthogonal properties of Walsh codes to achieve the features of aggregatability and extractability. With the extraction key, the supervisor can extract the Pedersen commitment of the total payment amount spent by any user from the aggregation result of the Walsh commitments. We regard it is of independent research interest.
- We provide the security definitions that WDAP needs to satisfy and prove its security. We define two types of adversaries and prove that WDAP satisfies the security against these adversaries.
- We deployed WDAP scheme on the ZCash system. Experimental results have demonstrated that our solution is highly efficient. It takes 12.00 s to generate a transaction (including the cost for generation of the native Zcash transaction, 7.27 s) and 0.43 s to verify a transaction. In a regulation period, the computational cost of the WDAP regulation is 2.17 s, and the transmission cost is 94.20 KB.

Paper outline

Section 2 introduces the relevant literature on DAP and the regulation of DAP. Section 3 presents the symbols and the related knowledge used in this paper. Section 4 briefly describes the system model and the intuitive idea. Section 5 proposes a new commitment scheme called Walsh commitment. Section 6 describes the framework and security definition of the WDAP system, and detailed construction is provided in Section 7. Section 8 supplements the consistency proof that was omitted earlier. Section 9 demonstrates the efficiency of WDAP through experiments. Section 10 further discusses our scheme. Finally, Section 11 concludes our work. In Appendix , we present the security proof.

2 Related Work

2.1 DAP

Decentralized anonymous payment systems can be broadly categorized into two types: the UTXO model and the account model.

Under the UTXO model, Zcash [7] employs zk-SNARKs to ensure privacy by concealing the sender addresses, recipient addresses, and transaction amounts. Monero [6] accomplishes the same objective utilizing a protocol that relies on ring signatures. Additionally, Zcoin [12] utilizes Pedersen commitments to construct zero-knowledge proofs that enable unlinkability and untraceability. Dash [13] and CoinShuffle [14] utilize CoinJoin, an approach inspired by mixing ideas, to obfuscate the connections between transactions and the identities of their senders and receivers.

Under the account model, DSC [15] offers an efficient NIZK scheme that employs homomorphic encryption to conceal the users' balance and the transaction amount. It also utilizes the NIZK scheme to prove transaction validity. Zether [16] leverages a smart contract to enable privacy-preserving transactions using a combination of homomorphic encryption and zero-knowledge proofs. Guan *et al.* [17] proposed BlockMaze, which employs a dual-balance model and achieves strong privacy guarantees.

2.2 Regulation of DAP

In order to address the regulatory concerns arising from anonymous transactions, there have been ongoing efforts to strike a balance between privacy protection and the enforcement of regulatory policies.

Some researchers have designed interactive systems to achieve regulatory and auditing functionalities. MiniLedger [9] enables extensive auditing capabilities, but they necessitate auditors to engage in interactions with users. ZkLedger [18] introduces a distributed ledgers to protect the privacy of users and provide fast, provably auditing. An auditor can send queries to the users and get a response and cryptographic assurance that the response is correct. These solutions need the participants to keep online and require constant interaction between users and regulators, which brings large overhead of computation and communication.

Conditional anonymity is another approach to design regulatory DAP. The supervisor can recover the identities of traders from the anonymous transactions using the trapdoor. Wu *et al.* [11] presented DAPS, a design for a regulatory DAP based on the blind signatures and key derivation mechanisms. The supervisor can recover the identities of traders using his secret key, and the transferred amount is of plaintext since the coins have fixed denominations. Garman et al. [19] designed a DAP scheme based on Zerocash that forces users to comply with specific policies and grants regulators the power of coin tracing and user tracing. Lin *et al.* [10] introduced a decentralized conditional anonymous payment system called DCAP, where the sender generates a new anonymous address from his long-term address when making transactions and the supervisor can invert the anonymous address to the original long-term one. Similar to [11], the transferred amount is not concealed. These approaches are not efficient enough since acquiring and analyzing each transaction is expensive.

Xue *et al.* [20] took efficiency into account and proposed a new scheme. They set regulation policies that limits the the total payment and the frequency of transactions in a time period. Users can prove that transactions are valid and comply with regulation policy using zk-SNARK and non-interactive zero-knowledge proofs. However, the zk-SNARK circuit involves proofs of two merkle trees, leading to excessive transaction generation time for users.

The properties comparison among different regulation of DAP schemes are given in Table 1.

TABLE 1: Properties comparison among different regulation schemes for DAP systems

Scheme	Regulation Efficiency	Restricted power	Generality	Non-interaction
Our scheme	✓	✓	✓	✓
[20]	✓	✓	✗	✓
[11]	✗	✗	✗	✓
[10]	✗	✗	✗	✓
[19]	✗	✗	✗	✓
[9]	✗	✗	✗	✗
[18]	✗	✗	✗	✗

3 PRELIMINARIES

3.1 Notations

Bold characters $\mathbf{v} = (\mathbf{v}[0], ..., \mathbf{v}[n-1])$ represent vectors, where $\mathbf{v}[k]$ represents the k-th component of the vector.

TABLE 2: Notations

$\mathcal{U}$	User
$\mathcal{M}$	Miner
$\mathcal{S}$	Supervisor
$\mathbf{wcom}$	Walsh commitment
$\mathbf{code}_i$	Walsh code of $\mathcal{U}_i$
$\mathbf{accu}$	Aggregation result of many Walsh commitments
n	User number and the code length, which is a power of 2
$\mathtt{atx}$	Anonymous transaction in the native DAP system
$\mathtt{wtx}$	Walsh transaction in WDAP
$\mathtt{block}$	Block in the blockchain

$\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the inner product of vectors, and $\mathbf{a} \circ \mathbf{b}$ represents the Hadamard product of vectors. The operation rules are as follows.

$$\mathbf{a} = (\mathbf{a}[0], ..., \mathbf{a}[n-1]), \mathbf{b} = (\mathbf{b}[0], ..., \mathbf{b}[n-1])$$
$$\mathbf{a} + \mathbf{b} = (\mathbf{a}[0] + \mathbf{b}[0], ..., \mathbf{a}[n-1] + \mathbf{b}[n-1])$$
$$\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{k=0}^{n-1} \mathbf{a}[k]\mathbf{b}[k]$$
$$k \cdot \mathbf{a} = (k \cdot \mathbf{a}[0], ..., k \cdot \mathbf{a}[n-1])$$
$$\mathbf{g}^k = (\mathbf{g}[0]^k, \mathbf{g}[1]^k, ..., \mathbf{g}[n-1]^k)$$
$$\mathbf{g}^{\mathbf{a}} = (\mathbf{g}[0]^{\mathbf{a}[0]}, \mathbf{g}[1]^{\mathbf{a}[1]}, ..., \mathbf{g}[n-1]^{\mathbf{a}[n-1]})$$
$$\mathbf{a} \circ \mathbf{b} = (\mathbf{a}[0]\mathbf{b}[0], ..., \mathbf{a}[n-1]\mathbf{b}[n-1])$$
$$\prod_{j=0}^{m} \mathbf{v}_j = \mathbf{v}_0 \circ \mathbf{v}_1 \circ ... \circ \mathbf{v}_{m-1}$$

Uppercase bold letters $\mathbf{A}$ represent matrices. The notation $\mathbf{A}_{i,:}$ is used to denote the i-th row of matrix $\mathbf{A}$, i.e., $\mathbf{A}_{i,:} = (\mathbf{A}_{i,0}, ..., \mathbf{A}_{i,n-1})$. Non-bold capital letter S represents a set. A negligible function is denoted as $\mathsf{negl}(\lambda)$ and 'probabilistic polynomial-time' is abbreviated as PPT. Other used symbols are listed in Table 2. $\mathsf{Hash} : \{0,1\}^* \to \mathbb{Z}_q^*$ is a secure hash function.

3.2 Bilinear pairing

The pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear pairing if it satisfies the following properties:

- **Bilinearity**: $\forall (g, q, a, b) \in (\mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{Z}_p \times \mathbb{Z}_p)$: $e(g^a, q^b) = e(g^a, q)^b = e(g, q^b)^a = e(g, q)^{ab}$
- **Non-degeneracy**: $e(g, q) \neq 1$
- **Computability**: the map e is efficiently computable.

$\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic multiplicative groups of the prime order p. We denote a pairing instance by $\mathsf{bp} = \mathsf{BilGen}(1^\lambda) = (p, e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, q)$. The bilinear pairing e used in our paper is a Type III bilinear pairing, where Discrete Logarithm (DL) Problem and Decisional Diffie-Hellman (DDH) Problem are hard and satisfies the assumptions as follow.

Assumption 1. ***d-Knowledge of Exponent Assumption (KEA3):*** [21]. For any PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathcal{X}_{\mathcal{A}}$, such that.

$$Pr\left[\begin{array}{c|c} \mathsf{bp}=\mathsf{BilGen}(1^\lambda) & \widehat{A}=A^\alpha \bigwedge \\ \alpha \xleftarrow{\$} \mathbb{Z}_p, h \xleftarrow{\$} \mathbb{G}_1 & A \neq g^v h^r \\ (A,\widehat{A};v,r) \leftarrow (\mathcal{A}||\mathcal{X}_{\mathcal{A}})(\mathsf{bp}, g^\alpha, h, h^\alpha) & \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Assumption 2. ***l-Decisional Diffie-Hellman inversion Problem (l-DDHI):*** For a problem instance $(g, g^\alpha, g^{\alpha^2}, ..., g^{\alpha^{l-1}}, z)$, where $g \in \mathbb{G}_1$ and $\alpha \in \mathbb{Z}_p^*$, l-DDHI problem [22] is to determine whether $z = g^{\frac{1}{\alpha}}$. If e is a Type III bilinear pairing, the advantage of any PPT adversary $\mathcal{A}$ in solving the l-DDHI problem is negligible.

3.3 Decentralized anonymous payment systems (DAP)

A DAP scheme such as Zerocash , Monero or Zether can be highly generalized into four algorithms as follows.

- $\mathsf{DAP.Setup}(1^\lambda) \rightarrow \mathtt{pp}_{\mathsf{DAP}}$. This algorithm takes a security parameter λ as input and outputs a public parameter $\mathtt{pp}_{\mathsf{DAP}}$.
- $\mathsf{DAP.GenAddr}(\mathtt{pp}_{\mathsf{DAP}}) \rightarrow (addr_{\mathsf{DAP}}, sk_{\mathsf{DAP}})$. This algorithm takes the public parameter $\mathtt{pp}_{\mathsf{DAP}}$ as input and outputs an address $addr_{\mathsf{DAP}}$ and a private key sk_{DAP}.
- $\mathsf{DAP.GenTx}(\mathtt{pp}_{\mathsf{DAP}}, \mathtt{pri}_{\mathsf{DAP}}, \mathtt{pub}_{\mathsf{DAP}}) \rightarrow \mathtt{atx}$. This algorithm takes some private inputs $\mathtt{pri}_{\mathsf{DAP}}$ and some private inputs $\mathtt{pub}_{\mathsf{DAP}}$ as input and outputs an anonymous transaction $\mathtt{atx}$. Usually, $\mathtt{pri}_{\mathsf{DAP}}$ includes the addresses of traders, the payment amount v and a private key sk_{DAP} and some other privacy.
- $\mathsf{DAP.VerifyTx}(\mathtt{pp}_{\mathsf{DAP}}, \mathtt{atx}, \mathtt{pub}_{\mathsf{DAP}}) \rightarrow 0/1$. This algorithm takes a transaction $\mathtt{atx}$ as input and outputs 1 if $\mathtt{atx}$ is valid or 0 otherwise.

3.4 Pedersen Commitment and Range proof

Pedersen commitment [23] is a secure commitment scheme constructed of two algorithms:

- $\mathsf{Setup}(1^\lambda) : g, h \xleftarrow{\$} \mathbb{G}$.
- $\mathsf{PCom}(g, h, v) : pcom \leftarrow g^v h^r$, where $r \xleftarrow{\$} \mathbb{Z}_p$.

A range proof scheme can generate a zero-knowledge proof for a Pedersen commitment, proving that the committed value v is smaller than max. Assume that $pcom = g^v h^r$ and the range proof algorithms are as follows.

- $\mathsf{RangeProof.Prove}(pcom, v, r, max) \rightarrow \pi_R$
- $\mathsf{RangeProof.Verify}(pcom, max, \pi_R) \rightarrow 0/1$

In practice, BulletProof [24] is used as a range proof instance in our system.

3.5 NIZK argument

Definition 1. a binary relation is represented as $\mathcal{R}(\phi, \varpi)$, where ϕ and ϖ are considered as a statement and a witness. $\mathcal{R}(\phi, \varpi) = 1$ if ϕ and ϖ satisfies $\mathcal{R}$. A Non-Interactive Zero Knowledge (NIZK) argument system [25] for an NP relation $\mathcal{R}(\phi, \varpi)$ consisting of a triple of PPT algorithms $(\mathsf{Setup}, \mathsf{Prove}, \mathsf{Verify})$ is secure if it satisfies the following properties:

Completeness: for each $\mathtt{crs} \leftarrow \mathsf{Setup}(1^\lambda)$ and $\mathcal{R}(\phi, \varpi) = 1$,

$$Pr[\pi \leftarrow \mathsf{Prove}(\mathtt{crs}, \phi, \varpi) | \mathsf{Verify}(\mathtt{crs}, \varpi, \pi) = 1] \geq 1 - \mathsf{negl}(\lambda)$$

(Adaptive) Soundness: all PPT malicious prover $\mathcal{A}$ cannot convince the verifier of a false statement.

$$Pr\left[\begin{array}{c|c} \mathtt{crs} \leftarrow \mathsf{Setup}(1^\lambda) & \mathcal{R}(\phi, \varpi) \neq 1 \wedge \\ (\phi, \varpi, \pi) \leftarrow \mathcal{A}(\mathtt{crs}) & \mathsf{Verify}(\mathtt{crs}, \phi, \pi) = 1 \end{array}\right] \leq \mathsf{negl}(\lambda)$$

(Adaptive) Zero-knowledge: the proof does not reveal anything but the truth of the statement, in particular it does not reveal the prover's witness. Formally, there exists a PPT simulator $\mathcal{S}$, such that for PPT adversaries $\mathcal{A}$, we have: (τ is the trapdoor of the $\mathtt{crs}$)

$$Pr\left[\begin{array}{c|c} \mathtt{crs} \leftarrow \mathsf{Setup}(1^\lambda) & \mathcal{R}(\phi, \varpi) = 1 \wedge \\ (\varpi, \phi) \leftarrow \mathcal{A}(\mathtt{crs}) & \mathcal{A}(\mathtt{crs}, \varpi, \phi) = 1 \\ \pi \leftarrow \mathsf{Prove}(\mathtt{crs}, \phi, \varpi) & \end{array}\right] -$$
$$Pr\left[\begin{array}{c|c} (\mathtt{crs}, \tau) \leftarrow \mathcal{S}(1^\lambda) & \mathcal{R}(\phi, \varpi) = 1 \wedge \\ (\varpi, \phi) \leftarrow \mathcal{A}(\mathtt{crs}) & \mathcal{A}(\mathtt{crs}, \varpi, \phi) = 1 \\ \pi \leftarrow \mathcal{S}(\mathtt{crs}, \phi, \tau) & \end{array}\right]$$
$$\leq \mathsf{negl}(\lambda)$$

3.6 zk-SNARK

A relation $\mathcal{R}(\phi, \varpi)$ can be represented as an arithmetic circuit C. A Zero-knowledge succinct non-interactive argument of knowledge (zk-SNARK) for C, consists of three algorithms as follows [26].

- $\mathsf{Zk.Setup}(1^\lambda, \mathcal{R}) \rightarrow \mathsf{pp}$: given the security parameter λ and the relation $\mathcal{R}$, outputs parameters pp.
- $\mathsf{Zk.Prove}(\mathsf{pp}, \phi, \varpi) \rightarrow \pi$: given the parameters pp, a statement ϕ and a witness w, it returns a proof π for $\mathcal{R}(\phi, \varpi)$.
- $\mathsf{Zk.Verify}(\mathsf{pp}, \phi, \pi) \rightarrow 0/1$: given the parameters pp, the statement ϕ and the proof π, it accepts or rejects the proof.

A zk-SNARK scheme is secure if it satisfies *Completeness*, *Soundness* and *Zero-knowledge*.

3.7 Walsh code

The Hadamard matrix is an $n \times n$ matrix composed of '+1' and '-1' elements that satisfies $\mathbf{H}_n \mathbf{H}_n^T = n\mathbf{I}_n$, where $\mathbf{H}_n^T$ is the transpose of $\mathbf{H}_n$ and $\mathbf{I}_n$ is the identity matrix. Below is the Hadamard matrix $\mathbf{H}_4$ of order four:

$$\mathbf{H}_4 = \begin{bmatrix} +1 & +1 & +1 & +1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix} \begin{array}{ll} \mathbf{code}_0 & = (+1, +1, +1, +1) \\ \mathbf{code}_1 & = (+1, -1, +1, -1) \\ \mathbf{code}_2 & = (+1, +1, -1, -1) \\ \mathbf{code}_3 & = (+1, -1, -1, +1) \end{array}$$

Walsh codes [27] are the row vectors of the Hadamard matrix. We denote the i-th Walsh code as $\mathbf{code}_i$. The set of Walsh codes of order four $\{\mathbf{code}_i | 0 \leq i < 4\}$ is listed above.

The Walsh codes have the following mathematical properties.

1) The length of each code is a power of 2 and the code is composed of elements of '+1' or '-1'.
2) All Walsh codes are mutually orthogonal:
$$\langle \mathbf{code}_i, \mathbf{code}_j \rangle = \begin{cases} 0 & i \neq j \\ n & i = j \end{cases}$$
3) The first element of Walsh code is '+1'.

4 System Model and Intuitive Idea

4.1 System model

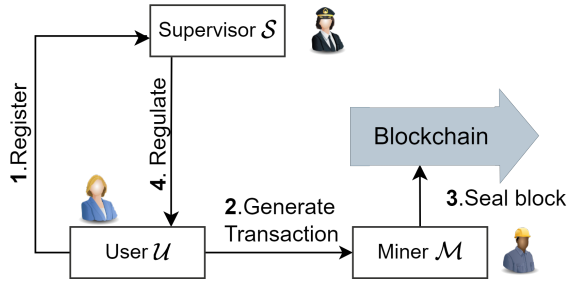


Fig. 1: The system model

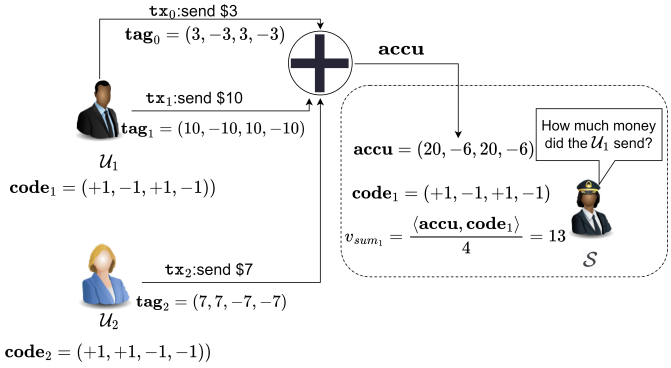


Fig. 2: The toy transaction system

Our WDAP is realized in a permissioned blockchain which provides the decentralized anonymous payment service. Before providing a high-level overview of our anonymous blockchain regulation system WDAP, we first introduce the various entities in the scheme, and define their roles and purposes:

As shown in Fig. 1, our system involves the following roles:

- **User** $\mathcal{U}$: utilizes blockchain for anonymous transactions and accept regulation from the supervisor.
- **Supervisor** $\mathcal{S}$: a trusted institution with the following tasks.
  - User registration: $\mathcal{S}$ assigns a unique Walsh code, sets a consumption limit and issues a certificate to the user.
  - Periodic regulation: $\mathcal{S}$ conducts periodic regulation to detect abnormal money flows.
- **Miner** $\mathcal{M}$: verifies the anonymous transactions and seals transactions to generate a new block.

4.2 Regulation policy

In the traditional financial systems, there exist fundamental regulation rules that limit the total amount of currency an individual can transfer within a fixed period. For instance, US laws impose restrictions on the total value of foreign currency an individual can exchange annually. In the context of cryptocurrency, some exchanges (such as Binance) have also set limits on the daily transaction amount for users. Similarly, in our DAP system, we have implemented the same regulation rules. Additionally, $\mathcal{S}$ has the authority to uncover the true identity of the sender in cases where a transaction is suspected to be illicit.

Our regulation policies are as follows.

- Users who have exceeded their spending limits within a certain period will be identified during the regulation process.
- $\mathcal{S}$ can retrieve the identity of the transaction sender. But he cannot retrieve the transferred value.

4.3 Intuitive idea

Inspired by the orthogonal properties of Walsh codes described in §3.7, we design a very simple transaction system as shown in Fig. 2, where users are honest and all data is public. Each user $\mathcal{U}_i$ is assigned a unique Walsh code $\mathbf{code}_i$ that is linked to their identity. When $\mathcal{U}_i$ makes a transaction of transferring value v, he attaches a $\mathbf{tag}$ to it, where $\mathbf{tag} = v \cdot \mathbf{code}_i$. Then m tags can be aggregated by accumulation

$$\mathbf{accu} = \sum_{j=0}^{m} \mathbf{tag}_j$$

The supervisor can efficiently extract the total payment of any user from the aggregation result $\mathbf{accu}$ by computing

$$v_{sum_i} = \frac{\langle \mathbf{accu}, \mathbf{code}_i \rangle}{4}$$

$\mathcal{S}$ can easily check if $\mathcal{U}_i$'s total payment v_{sum_i} is over limit.

This toy example demonstrates the aggregatability and extractability of transaction amounts in a plaintext transaction system. Following this idea, we designed a new commitment scheme, Walsh commitment to implement the same target of the toy example while the trader identities and the transferred money are concealed. Based on the Walsh commitment scheme, we proposed WDAP, a regulation scheme for anonymous cryptocurrencies in DAP, which involves five algorithms as follows.

0) **Setup**. Generate the public parameters.
1) **Register**. $\mathcal{S}$ issues the certificate for $\mathcal{U}_i$ and assigns a unique Walsh code $\mathbf{code}_i$ to $\mathcal{U}_i$.
2) **Generate transaction**. $\mathcal{U}_i$ transfers value v in the native DAP system to the receiver, attaches a Walsh commitment of v and $\mathbf{code}_i$ to it and proves its validity using zero-knowledge prof.
3) **Seal block**. $\mathcal{M}$ verifies the transactions and aggregates the commitments to generate a new block.
4) **Regulate**. After the regulation period ends, each user $\mathcal{U}_i$ generates a range proof, showing that his total payment is under limit. $\mathcal{S}$ extracts the Pedersen commitment of each user's total payment from the aggregation result of all Walsh commitments and checks the range proofs.

5 Walsh Commitment

5.1 Description of Walsh commitment

In this section, we propose an innovative commitment scheme, Walsh commitment, which leverages the orthogonal properties of Walsh codes to achieve the features of aggregatability and extractability.

Formally, the Walsh commitment scheme is composed of the following algorithms.

- $\mathsf{WCom.Setup}(1^\lambda) \rightarrow (\mathsf{pp}_{\mathsf{WCom}}, s)$. This algorithm takes as input a security parameter λ and outputs public parameters pp and an extraction key s.
- $\mathsf{WCom.Commit}(\mathsf{pp}, \mathbf{code}_i, v) \rightarrow \mathbf{wcom}$. This algorithm is run by user $\mathcal{U}_i$ to generate a Walsh commitment. It takes pp, his $\mathbf{code}_i$ and an input value v as input, and outputs the Walsh commitment $\mathbf{wcom}$.
- $\mathsf{WCom.Trace}(\mathsf{pp}, \mathbf{wcom}, s) \rightarrow \mathbf{code}$. This algorithm retrieves the Walsh code committed in a Walsh commitment. It takes as input pp, a Walsh commitment $\mathbf{wcom}$ and an extraction key s, and outputs the Walsh code $\mathbf{code}$ committed in $\mathbf{wcom}$.
- $\mathsf{WCom.Aggregate}(\mathsf{pp}, \{\mathbf{wcom}_j | 0 \leq j < m\}) \rightarrow \mathbf{accu}$. This algorithm aggregates a set of Walsh commitment. It takes as input pp and a set of Walsh commitment $\{\mathbf{wcom}_j | 0 \leq j < m\}$, and outputs an aggregation result $\mathbf{accu}$.
- $\mathsf{WCom.Extract}(\mathsf{pp}, \mathbf{accu}, \mathbf{code}_i, s) \rightarrow pc_{v_{sum_i}}$. This algorithm extracts the Pedersen commitment of the total payment of $\mathcal{U}_i$ from $\mathbf{accu}$. It takes as input pp, an aggregation result $\mathbf{accu}$, a target Walsh code $\mathbf{code}_i$ and the extraction key s, and outputs the extraction result $pc_{v_{sum_i}}$.

5.2 Security properties of Walsh commitment

To evaluate the security of our scheme, two types of adversaries are introduced:

- *Type-1 Adversary ($\mathcal{A}_1$)*: can be considered as the user $\mathcal{U}$ who attempts to obtain the committed code $\mathbf{code}$ and value v from the Walsh commitment generated by the other users.
- *Type-2 Adversary ($\mathcal{A}_2$)*: can be considered as the supervisor $\mathcal{S}$ who holds the extraction key s and attempts to obtain the committed value v from the Walsh commitment.

Definition 2. We say that a Walsh commitment scheme is secure if it satisfies the following properties: *Hidng*, *Binding*, *Traceability*, *Extractability*.

1) *Hiding*. for all PPT adversaries $\mathcal{A}_1, \mathcal{A}_2$, $|Pr[\mathsf{exp}^{\mathsf{Hiding}}_{\mathcal{A}_1}(\lambda) = 1] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$ and $|Pr[\mathsf{exp}^{\mathsf{Perfect-Hiding}}_{\mathcal{A}_2}(\lambda) = 1] - \frac{1}{2}| = 0$.
2) *Binding*. for all PPT adversaries $\mathcal{A} \in \{\mathcal{A}_1, \mathcal{A}_2\}$, $Pr[\mathsf{exp}^{\mathsf{Binding}}_{\mathcal{A}}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$.
3) *Traceability*. With the extraction key s, one can efficiently retrieve the input $\mathbf{code}$ from a Walsh commitment, i.e. $Pr[\mathsf{exp}^{\mathsf{Traceability}}(\lambda) = 1] = 1$.
4) *Extractability*. With the extraction key s, one can efficiently retrieve the Pedersen commitment of the sum of the value committed with $\mathbf{code}_i$ from the aggregation result $\mathbf{accu}$. $\mathbf{wcom}$, i.e. $Pr[\mathsf{exp}^{\mathsf{Extractability}}(\lambda) = 1] = 1$.

- **Hiding experiment 1** $\mathsf{exp}^{\mathsf{Hiding}}_{\mathcal{A}_1}(\lambda)$
  1: $(\mathsf{pp}, s) \leftarrow \mathsf{WCom.Setup}(1^\lambda)$
  2: $(\mathbf{code}_{i_0}, v_0, \mathbf{code}_{i_1}, v_1) \leftarrow \mathcal{A}_1(\mathsf{pp})$
  3: $b \xleftarrow{\$} \{0,1\}$
  4: $\mathbf{wcom}_b \leftarrow \mathsf{WCom.Commit}(\mathbf{code}_{i_b}, v_b)$
  5: $b' \leftarrow \mathcal{A}_1(\mathbf{wcom}_b, \mathbf{code}_{i_0}, v_0, \mathbf{code}_{i_1}, v_1)$
  6: return 1 if $b = b'$ or 0 otherwise
- **Hiding experiment 2** $\mathsf{exp}^{\mathsf{Perfect-Hiding}}_{\mathcal{A}_2}(\lambda)$
  1: $(\mathsf{pp}, s) \leftarrow \mathsf{WCom.Setup}(1^\lambda)$
  2: $(\mathbf{code}_i, v_0, v_1) \leftarrow \mathcal{A}_2(\mathsf{pp}, s)$
  3: $b \xleftarrow{\$} \{0,1\}$
  4: $\mathbf{wcom} \leftarrow \mathsf{WCom.Commit}(\mathbf{code}_i, v_b)$
  5: $b' \leftarrow \mathcal{A}_2(\mathbf{wcom}, \mathbf{code}_i, v_0, v_1, s)$
  6: return 1 if $b = b'$ or 0 otherwise
- **Binding experiment** $\mathsf{exp}^{\mathsf{Binding}}_{\mathcal{A}}(\lambda)$
  1: $(\mathsf{pp}, s) \leftarrow \mathsf{WCom.Setup}(1^\lambda)$
  2: $(\mathbf{code}_{i_0}, v_0, \mathbf{code}_{i_1}, v_1) \leftarrow \mathcal{A}(\mathsf{pp}, s)$
  3: $\mathbf{wcom_0} \leftarrow \mathsf{WCom.Commit}(\mathbf{code}_{i_0}, v_0)$
  4: $\mathbf{wcom_1} \leftarrow \mathsf{WCom.Commit}(\mathbf{code}_{i_1}, v_1)$
  5: return 1 if $\mathbf{wcom_0} = \mathbf{wcom_1} \wedge$ $(\mathbf{code}_{i_0} \neq \mathbf{code}_{i_1} \vee v_0 \neq v_1)$ or 0 otherwise
- **Traceability experiment** $\mathsf{exp}^{\mathsf{Traceability}}(\lambda)$
  1: $(\mathsf{pp}, s) \leftarrow \mathsf{WCom.Setup}(1^\lambda)$
  2: $\mathbf{wcom} \leftarrow \mathsf{WCom.Commit}(\mathsf{pp}, \mathbf{code}, v)$
  3: $\mathbf{code}' \leftarrow \mathsf{WCom.Trace}(\mathsf{pp}, \mathbf{wcom}, s)$
  4: return 1 if $\mathbf{code}' = \mathbf{code}$ or 0 otherwise
- **Extractability experiment** $\mathsf{exp}^{\mathsf{Extractability}}(\lambda)$
  1: $(\mathsf{pp}, s) \leftarrow \mathsf{WCom.Setup}(1^\lambda)$
  2: $i \xleftarrow{\$} [0, n-1]$
  3: Init J_i as an empty set
  4: **for** $j \in [0, m-1]$ **do**
  5: $\quad sender_j \xleftarrow{\$} [0, n-1]$
  6: $\quad v_j, r_j \xleftarrow{\$} \mathbb{Z}_p$
  7: $\quad$ **If** $sender_j == i$
  8: $\quad\quad \mathbf{wcom}_j \leftarrow \mathsf{WCom.Commit}(\mathsf{pp}, \mathbf{code}_i, v_j)$
  9: $\quad\quad J_i \leftarrow J_i \cup \{j\}$
  10: $\quad$ **Elif** $sender_j \neq i$
  11: $\quad\quad \mathbf{wcom}_j \leftarrow \mathsf{WCom.Commit}(\mathsf{pp}, \mathbf{code}_{sender_j}, v_j)$
  12: **end for**
  13: $\mathbf{accu} \leftarrow \mathsf{WCom.Aggregate}(\mathsf{pp}, \{\mathbf{wcom}_j | 0 \leq j < m\})$
  14: $pc_{v_{sum_i}} \leftarrow \mathsf{WCom.Extract}(\mathsf{pp}, \mathbf{accu}, \mathbf{code}_i, s)$
  15: return 1 if $pc_{v_{sum_i}} = g^{\sum_{j \in J_i} v_j} h^{\sum_{j \in J_i} r_j}$ or 0 otherwise

5.3 Construction of Walsh commitment

WCom.Setup. Generate an n-order Hadamard matrix $\mathbf{H}$, where $n = 2^d$ and d is an integer. Set the Walsh code $\mathbf{code}_i = \mathbf{H}_{i,:}$ and allocate $\mathbf{code}_i$ to the User $\mathcal{U}_i$. Generate $\mathsf{bp} = (p, e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, q) = \mathsf{BilGen}(1^\lambda)$, and randomly select another generator $h \xleftarrow{\$} \mathbb{G}_1$. Randomly choose extraction key $s \xleftarrow{\$} \mathbb{Z}_p$. Set $\mathbf{g} = (g_0, ..., g_{n-1}) = (g^{s^0}, ..., g^{s^{n-1}}), \mathbf{h} = (h_0, ..., h_{n-1}) = (h^{s^0}, ..., h^{s^{n-1}})$. Output the public parameters $\mathsf{pp} = (\mathsf{bp}, \mathbf{g}, \mathbf{h}, \{\mathbf{code}_i | 0 \leq i < n\})$, and send s to $\mathcal{S}$ as the extraction key.

WCom.Commit. $\mathcal{U}_i$ generates a Walsh commitment to his Walsh code $\mathbf{code}_i$ and value v when he transfers v to another user. This algorithm selects a random number $r \xleftarrow{\$} \mathbb{Z}_p$, and computes

$$\begin{aligned}\mathbf{wcom} &\leftarrow (\mathbf{g}^v \mathbf{h}^r)^{\mathbf{code}_i} \\ &= ((g_0^v h_0^r)^{\mathbf{code}_i[0]}, ..., (g_{n-1}^v h_{n-1}^r)^{\mathbf{code}_i[n-1]})\end{aligned}$$

WCom.Trace. This algorithm retrieves the committed Walsh code from a Walsh commitment using the extraction key s. Init an empty vector $\mathbf{code} = (0, 0, ..., 0) = \{0\}^n$. For $k \in [0, n-1]$, if $\mathbf{wcom}[0]^{s^k} == \mathbf{wcom}[k]$, set $\mathbf{code}[k] = 1$, otherwise set $\mathbf{code}[k] = -1$. Output $\mathbf{code}$.

Theorem 1. The Walsh commitment scheme satisfies *traceability.*

PROOF. Since the first element of any Walsh code is '+1', i.e. $\mathbf{code}[0] = 1$, so $\mathbf{wcom}[0] = g_0^v h_0^r$. If $\mathbf{code}[k] = 1$, then $\mathbf{wcom}[k] = g_k^v h_k^r = \mathbf{wcom}[0]^{s^k}$, else if $\mathbf{code}[k] = -1$, then $\mathbf{wcom}[k] = (g_k^v h_k^r)^{-1} = \mathbf{wcom}[0]^{-s^k}$. So we can retrieve $\mathbf{code}[k]$ by checking if $\mathbf{wcom}[k] == \mathbf{wcom}[0]^{s^k}$.

Therefore, the committed $\mathbf{code}$ can be efficiently retrieved and the Walsh commitment scheme satisfies *traceability*. □

WCom.Aggregate. A set of Walsh commitment $\{\mathbf{wcom}_j | 0 \leq j < m\}$ can be aggregated by simple accumulation.

$$\mathbf{accu} \leftarrow \prod_{j=0}^{m-1} \mathbf{wcom}_j$$

WCom.Extract. $\mathcal{S}$ uses the extraction key s to construct a vector $\mathbf{s} = (s^0, s^{-1}, ..., s^{-n+1}) \in \mathbb{Z}_p^n$, and computes

$$pc_{v_{sum_i}} \leftarrow \prod_{k=0}^{n-1} \mathbf{accu}[k]^{\frac{\mathbf{s}[k] \cdot \mathbf{code}_i[k]}{n}}$$

$pc_{v_{sum_i}}$ is considered as the Pedersen commitment of the total payment amount of $\mathcal{U}_i$. The correctness proof is as follows.

Theorem 2. The Walsh commitment scheme satisfies *Extractability.*

PROOF. $\{\mathbf{wcom}_j | 0 \leq j < m\}$ is a set of Walsh commitments and its aggregation result is $\mathbf{accu} = \mathsf{WCom.Aggregate}(\mathsf{pp}, \{\mathbf{wcom}_j | 0 \leq j < m\})$.

The sender of $\mathbf{wcom}_j$ is denoted as $\mathcal{U}_{sender_j}$ and $\mathbf{code}_{sender_j}$ is his Walsh code, and $\mathbf{wcom}_j = (\mathbf{g}^{v_j} \mathbf{h}^{r_j})^{\mathbf{code}_{sender_j}}$. The set of indices of Walsh commitments sent by $\mathcal{U}_i$ is denoted as $J_i = \{j | \mathcal{U}_{sender_j} = \mathcal{U}_i\}$. Then,

$$\begin{aligned}
pc_{v_{sum_i}} &= \prod_{k=0}^{n-1} \mathbf{accu}[k]^{\frac{\mathbf{s}[k] \cdot \mathbf{code}_i[k]}{n}} \\
&= \prod_{k=0}^{n-1} \prod_{j=0}^{m-1} \mathbf{wcom}_j[k]^{\frac{\mathbf{s}[k] \cdot \mathbf{code}_i[k]}{n}} \\
&= \prod_{k=0}^{n-1} \prod_{j=0}^{m-1} (g_k^{v_j} h_k^{r_j})^{\frac{s^{-k}}{n} \cdot \mathbf{code}_{sender_j}[k] \cdot \mathbf{code}_i[k]} \\
&= \prod_{k=0}^{n-1} \prod_{j=0}^{m-1} (g^{v_j} h^{r_j})^{\frac{1}{n} \cdot \mathbf{code}_{sender_j}[k] \cdot \mathbf{code}_i[k]} \\
&= \prod_{j=0}^{m-1} (g^{v_j} h^{r_j})^{\frac{1}{n} \cdot \sum_{k=0}^{n-1} (\mathbf{code}_{sender_j}[k] \cdot \mathbf{code}_i[k])} \\
&= \prod_{j \in J_i} (g^{v_j} h^{r_j})^{\frac{1}{n} \cdot \langle \mathbf{code}_{sender_j}, \mathbf{code}_i \rangle} \cdot \\
&\prod_{j \notin J_i} (g^{v_j} h^{r_j})^{\frac{1}{n} \cdot \langle \mathbf{code}_{sender_j}, \mathbf{code}_i \rangle} \\
&= \prod_{j \in J_i} g^{v_j} h^{r_j} = g^{\sum_{j \in J_i} v_j} h^{\sum_{j \in J_i} r_j}
\end{aligned}$$

So $pc_{v_{sum_i}}$ can be considered as the Pedersen commitment of the total payment amount sent by $\mathcal{U}_i$. Therefore, the Walsh commitment scheme satisfies *Extractability*. □

The WCom commitment

WCom.Setup

- $\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{pp}, s)$
  - Procedures:
    1: generate an n-order Hadamard matrix $\mathbf{H}_n$, $n = 2^d$, d is an integer, and set $\mathbf{code}_i = \mathbf{H}_{i,:}$.
    2: $\mathsf{bp} = (p, e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, q) \leftarrow \mathsf{BilGen}(1^\lambda)$. $h \xleftarrow{\$} \mathbb{G}_1$.
    3: $s \xleftarrow{\$} \mathbb{Z}_p$.
    4: set $\mathbf{g} = (g_0, ..., g_{n-1}) = (g^{s^0}, ..., g^{s^{n-1}})$, $\mathbf{h} = (h_0, ..., h_{n-1}) = (h^{s^0}, ..., h^{s^{n-1}})$.
    5: publish the public parameters $\mathsf{pp} = (\mathsf{bp}, \mathbf{H}, \mathbf{g}, \mathbf{h}, \{\mathbf{code}_i | 0 \leq i < n\})$ and keep s as the extraction key.
    6: send s to $\mathcal{S}$ as the extraction key

WCom.Commit

- $\mathsf{WCom.Commit}(\mathsf{pp}, \mathbf{code}_i, v) \rightarrow \mathbf{wcom}$
  - Procedures:
    1: $r \xleftarrow{\$} \mathbb{Z}_p$
    2: $\mathbf{wcom} \leftarrow (\mathbf{g}^v \mathbf{h}^r)^{\mathbf{code}_i}$

WCom.Trace

- $\mathsf{WCom.Trace}(\mathsf{pp}, \mathbf{wcom}, s) \rightarrow \mathbf{code}$
  - Procedures:
    1: set $\mathbf{code} = (0, 0, ..., 0) \in \{0\}^n$
    2: **for** $k \in [0, n-1]$ **do**
    3: **If** $\mathbf{wcom}[0]^{s^k} == \mathbf{wcom}[k]$
    4: set $\mathbf{code}[k] = 1$, otherwise set $\mathbf{code}[k] = -1$
    5: **end for**

WCom.Aggregate

- $\mathsf{WCom.Aggregate}(\mathsf{pp}, \{\mathbf{wcom}_j | 0 \leq j < m\}) \rightarrow \mathbf{accu}$
  - Procedure:
    1: $\mathbf{accu} \leftarrow \prod_{j=0}^{m-1} \mathbf{wcom}_j$.

WCom.Extract

- $\mathsf{WCom.Extract}(\mathsf{pp}, \mathbf{accu}, \mathbf{code}_i, s) \rightarrow pc_{v_{sum_i}}$
  - Inputs:
    1) aggregation result $\mathbf{accu}$
    2) target Walsh code $\mathbf{code}_i$
    3) extraction key s
  - Outputs:
    1) extraction result $pc_{v_{sum_i}}$
  - Procedures:
    1: set $\mathbf{s} = (s^0, s^{-1}, ..., s^{-n+1}) \in \mathbb{Z}_p^n$
    2: $pc_{v_{sum_i}} \leftarrow \prod_{k=0}^{n-1} \mathbf{accu}[k]^{\frac{\mathbf{s}[k] \cdot \mathbf{code}_i[k]}{n}}$

Theorem 3. The Walsh commitment scheme is a secure commitment scheme (as defined in Definition 2), if the DL problem and the l-DDHI problem hold.

A proof of Theorem 3 is provided in Appendix A. The proof of hiding property is based on the l-DDHI problem and the proof of binding property is based on the the DL problem.

6 SYSTEM FRAMEWORK

6.1 Data structure

Transaction format: In WDAP, the transaction sent by user $\mathcal{U}_i$ is called Walsh transaction and is denoted as `wtx`, which

consists of the following three components.

$$\texttt{wtx} := (\texttt{atx}, \mathbf{wcom}, \Pi)$$

- atx: the native transaction of DAP transferring v from the sender to the receiver.
- **wcom**: the Walsh commitment to the payment v and the Walsh code $\mathbf{code}_i$ corresponding to the sender $\mathcal{U}_i$.
- Π: the consistency proof, proving that **wcom** is valid, the input value of commitment **wcom** equals to the transferred value v in atx and some other statements.

Block format: The block format in WDAP is defined as follows.

$$\texttt{block} := (\texttt{header}, \{\texttt{atx}_j | 0 \leq j < m\}, \mathbf{accu})$$

- header: the header of the block.
- $\{\texttt{atx}_j | 0 \leq j < m\}$: a set of atx transactions sealed in the block.
- **accu**: the aggregation result of all Walsh commitments of transactions that occurred so far during this regulation period.

The only difference between the block format of WDAP and the native DAP is that we add an **accu** field for storing the aggregation result of commitments.

6.2 Formalized definition

Formally, a WDAP scheme is composed of the following algorithms.

- $\mathsf{Setup}(1^\lambda) \rightarrow (\texttt{pp}, s, sk_{\mathcal{S}})$. This algorithm takes as input a security parameter λ and outputs public parameters pp, an extraction key s and the secret key of $\mathcal{S}$: $(pk_{\mathcal{S}}, sk_{\mathcal{S}})$.
- Register. $\mathcal{U}_i$ registers in the WDAP system. This algorithm is constructed of two sub-algorithms:
  - $\mathcal{U}_i : (\texttt{pp}) \rightarrow (sk_i, pk_i)$. $\mathcal{U}_i$ generates a key pair (pk_i, sk_i) and sends his public key pk_i to $\mathcal{S}$.
  - $\mathcal{S} : (\texttt{pp}, sk_{\mathcal{S}}, pk_i) \rightarrow (\mathbf{code}_i, v_{max_i}, \zeta_i)$. $\mathcal{S}$ assigns a unique Walsh code $\mathbf{code}_i$ and a payment limit v_{max_i} to $\mathcal{U}_i$. Then, $\mathcal{S}$ generates a signature ζ_i as a certificate for $\mathcal{U}_i$.
- $\mathsf{GenTx}(\texttt{pp}, v, \mathbf{code}_i, sk_i, pk_i, \zeta_i, \texttt{pri}_{\mathsf{DAP}}, \texttt{pub}_{\mathsf{DAP}}) \rightarrow \texttt{wtx}$. $\mathcal{U}_i$ generates new Walsh transaction. It takes as input pp, $\mathbf{code}_i$, the payment amount v, sk_i, pk_i, ζ_i, some private and public inputs for DAP, $\texttt{pri}_{\mathsf{DAP}}$ and $\texttt{pub}_{\mathsf{DAP}}$ as input, and outputs a Walsh transaction wtx.
- $\mathsf{SealBlock}(\texttt{pp}, \texttt{block}_{l-1}, \texttt{pub}_{\mathsf{DAP}}, \{\texttt{wtx}_j | 0 \leq j < m\}) \rightarrow \texttt{block}_l$. $\mathcal{M}$ verifies transactions and seals them into a new block. It takes as input the transaction set $\{\texttt{wtx}_j | 0 \leq j < m\}$ and the former block $\texttt{block}_{l-1}$, and outputs a new block $\texttt{block}_l$.
- Regulate. This algorithm is constructed of two sub-algorithms:
  - $\mathcal{U}_i : (\texttt{pp}, v_{max_i}, Tx_i) \rightarrow \pi_{R_i}$. takes his consumption limit v_{max_i} and his transaction set Tx_i as input and generates a range proof π_{R_i}, showing that his total payment is under limit v_{max_i}.
  - $\mathcal{S} : (\texttt{pp}, \texttt{block}_{\mathsf{last}}, \{v_{max_i}, \pi_{R_i} | 0 \leq i < n\}) \rightarrow \mathbf{b} = \{0/1\}^n$. Takes the last block in the regulation period $\texttt{block}_{\mathsf{last}}$ and $\{v_{max_i}, \pi_{R_i} | 0 \leq i < n\}$ as input. It outputs the regulation result vector $\mathbf{b}$, where $\mathbf{b}[i] = 1$ if the regulation passed or $\mathbf{b}[i] = 0$ otherwise. $\mathcal{U}_i$ is suspected of overspending if $\mathbf{b}[i] = 0$.

In this formal definition, we ignore the Trace *algorithm that how $\mathcal{S}$ retrieves the sender identity from a transaction* wtx*, because it is quite trival. $\mathcal{S}$ can parse* $\texttt{wtx} = (\texttt{atx}, \mathbf{wcom}, \Pi)$*, and compute* $\mathbf{code}_{sender} = \mathsf{WCom.Trace}(\texttt{pp}, s, \mathbf{wcom})$.

6.3 Security properties

Definition 3. We say that a WDAP scheme is secure if it satisfies *Transaction indistinguishability*, *Restricted power*, *Consistency*, *Traceability* and *Extractability*.

1) *Transaction indistinguishability*. The Walsh transaction wtx reveals no information about the transaction privacy for $\mathcal{A}_1$. We say that a WDAP scheme satisfies *Transaction indistinguishability* if for any PPT adversaries $\mathcal{A}_1$, $Pr[\mathsf{exp}^{\mathsf{Ind}}_{\mathcal{A}_1}(\lambda) = 1] \leq \mathsf{negl}(\lambda)$.

- **Indistinguishability experiment** $\mathsf{exp}^{\mathsf{Ind}}_{\mathcal{A}_1}(\lambda)$
  1: $(\texttt{pp}, s) \leftarrow \mathsf{WCom.Setup}(1^\lambda)$
  2: $(\texttt{input}_0, \texttt{input}_1) \leftarrow \mathcal{A}_1(\texttt{pp})$
  3: $b \xleftarrow{\$} \{0, 1\}$
  4: $\texttt{wtx}_b \leftarrow \mathsf{GenTx}(\texttt{input}_b)$
  5: $b' \leftarrow \mathcal{A}_1(\texttt{wtx}_b, \texttt{input}_0, \texttt{input}_1)$
  6: return 1 if $b = b'$ or 0 otherwise

2) *Restricted power*. For any PPT adversaries $\mathcal{A}_2$, he cannot retrieve any information about the payment amount nor the receiver address from the Walsh transaction wtx. We say a WDAP scheme satisfies *Restricted power* if the WCom scheme satisfies *hiding*, the DAP scheme is secure and the proof Π satisfies *zero-knowledge*.
3) *Consistency and unforgeability*. This property means that for a $\texttt{wtx} := (\texttt{atx}, \mathbf{wcom}, \Pi)$, The commitment **wcom** is generated correctly and the committed value v equals to the payment in atx and the committed code $\mathbf{code}_i$ is consistent with the User identity $\mathcal{U}_i$. We say that a WDAP scheme satisfies *Consistency* if the proof Π satisfies *soundness*.
4) *Traceability*. This property means that $\mathcal{S}$ can efficiently retrieve sender identity with the extraction key s. We say that a WDAP scheme satisfies *Traceability* if the WCom scheme satisfies *Traceability*.
5) *Extractability*. This property means that $\mathcal{S}$ can efficiently retrieve the Pedersen commitment of the sum of payment of $\mathcal{U}_i$ from an aggregation result **accu** using the extraction key s. We say that a WDAP scheme satisfies *Extractability* if the WCom scheme satisfies *Extractability*.

In definition 3, we ignore some common properties which also need to be followed in an anonymous transaction system such as *Double-Spending Resilience* and *Replay Attacks Resilience*, since these properties directly derive from the native DAP system.

7 Construction of WDAP

In this section, we provide the specific construction of the WDAP system. However, the detailed implementation of how to generate and verify the consistency proof Π, i.e. the algorithms named CP.Setup, CP.Prove and CP.Verify, will be provided in §8.

Setup. Run

$$(\mathsf{pp}_{\mathsf{WCom}}, s) \leftarrow \mathsf{WCom.Setup}(1^\lambda)$$

$$\mathsf{pp}_{\mathsf{DAP}} \leftarrow \mathsf{DAP.Setup}(1^\lambda)$$

$$\mathtt{crs} \leftarrow \mathsf{CP.Setup}(\mathsf{pp}_{\mathsf{WCom}}, \mathsf{pp}_{\mathsf{DAP}})$$

Send the extraction key s to $\mathcal{S}$ using a secure channel. $\mathcal{S}$ randomly chooses the private key $sk_{\mathcal{S}} \xleftarrow{\$} \mathbb{Z}_p^*$ and computes the public key $pk_{\mathcal{S}} \leftarrow g^{sk_{\mathcal{S}}}$. Publish the public parameters $\mathsf{pp} = (\mathsf{pp}_{\mathsf{WCom}}, \mathsf{pp}_{\mathsf{DAP}}, \mathtt{crs}, pk_{\mathcal{S}}, \mathcal{T})$.

Register.

(1) $\mathcal{U}_i$ randomly chooses a private key $sk_i \xleftarrow{\$} \mathbb{Z}_p^*$ and generates a public key $pk_i = \mathsf{Hash}(sk_i)$. Notably, the private key here has nothing to do with the private key in the native DAP. Then $\mathcal{U}_i$ sends his public key pk_i to $\mathcal{S}$ for registration.

(2) $\mathcal{S}$ assigns a unique Walsh code $\mathbf{code}_i$ to $\mathcal{U}_i$ and sets the consumption limit v_{max_i} to $\mathcal{U}_i$ based on the capability of $\mathcal{U}_i$. $\mathcal{S}$ computes $g_{mul_i} = \prod_{k=0}^{n-1} g_k^{\mathbf{code}_i[k]}$ and $h_{mul_i} = \prod_{k=0}^{n-1} h_k^{\mathbf{code}_i[k]}$. Finally, $\mathcal{S}$ issues a digital certificate ζ_i by signing the message $\mathfrak{m} = \mathsf{Hash}(pk_i, g_{mul_i}, h_{mul_i})$

$$\zeta_i \leftarrow \mathsf{GenSig}(sk_{\mathcal{S}}, \mathfrak{m})$$

GenTx. $\mathcal{U}_i$ runs

$$\mathtt{atx} \leftarrow \mathsf{DAP.GenTx}(v, \mathtt{pri}_{\mathsf{DAP}}, \mathtt{pub}_{\mathsf{DAP}})$$

to generate the anonymous transaction $\mathtt{atx}$ in the native DAP system. Then computes

$$\mathbf{wcom} \leftarrow \mathsf{WCom.Commit}(\mathbf{code}_i, v)$$

Afterwards, runs

$$\Pi \leftarrow \mathsf{CP.Prove}(\mathsf{pp}, v, r, \mathbf{code}_i, sk_i, pk_i, \mathbf{wcom}, \zeta_i, \mathtt{atx})$$

to generate the consistency proof Π. Finally, set transaction $\mathtt{wtx} = (\mathtt{atx}, \mathbf{wcom}, \Pi)$.

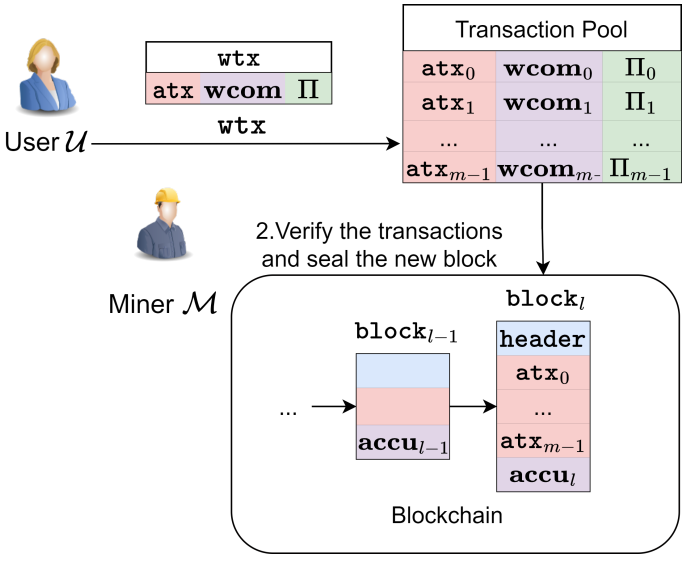


Fig. 3: Seal block

SealBlock. As shown in Fig. 3, $\mathcal{M}$ fetches the transaction set $\{\mathtt{wtx}_j | 0 \le j < m\}$ from the transaction pool, verifies transactions and seals them into a new block.

To verify a transaction $\mathtt{wtx}$, parse it as $(\mathtt{atx}, \mathbf{wcom}, \Pi)$. A Walsh transaction $\mathtt{wtx}$ is valid if $\mathtt{atx}$ is valid and $\mathbf{wcom}$ is consistent with $\mathtt{atx}$ (the definition of the consistency is given in §8). $\mathcal{M}$ computes

$$b_0 \leftarrow \mathsf{DAP.VerifyTx}(\mathtt{atx}, \mathtt{pub}_{\mathsf{DAP}})$$

$$b_1 \leftarrow \mathsf{CP.Verify}(\mathbf{wcom}, \mathtt{atx}, \Pi)$$

set $b = b_0 \cdot b_1$. If $b = 1$, $\mathtt{wtx}$ is valid, otherwise $\mathtt{wtx}$ is invalid.

After verifying all transactions, $\mathcal{M}$ aggregates all the Walsh commitments in this transaction set

$$\mathbf{accu} \leftarrow \mathsf{WCom.Aggregate}(\{\mathtt{wtx}_j.\mathbf{wcom} | 0 \le j < m\})$$

If the block to be sealed is the first block within this period, set

$$\mathbf{accu}_l \leftarrow \mathbf{accu}$$

Otherwise, set $\mathbf{accu}_l$ to the sum of $\mathbf{accu}$ and the previous aggregation result that was stored in the previous block.

$$\mathbf{accu}_l \leftarrow \mathbf{accu} + \mathtt{block}_{l-1}.\mathbf{accu}$$

$\mathbf{accu}_l$ is considered as the aggregation result of all Walsh commitments of transactions that occurred so far during the regulation period. Finally, create a new block

$$\mathtt{block}_l = (\mathtt{header}, \{\mathtt{wtx}_j.\mathtt{atx} | 0 \le j < m\}, \mathbf{accu}_l)$$

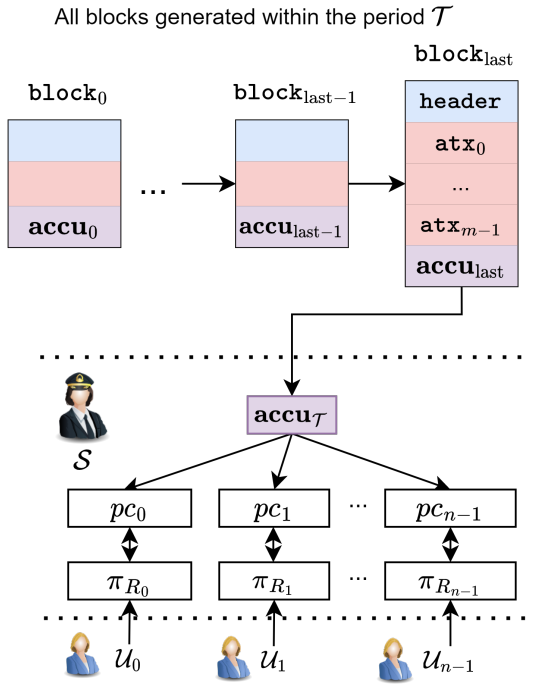


Fig. 4: Regulate

Regulate. As shown in Fig. 4, after the end of the regulation period $\mathcal{T}$, $\mathcal{S}$ conducts regulation, which consists of two steps:

1) $\mathcal{U}_i$ generates a range proof, showing that his total payment is under limit v_{max_i}.
2) $\mathcal{S}$ extracts the Pedersen commitments of users' total payments from the aggregation result and use them to verify the range proofs.

(1) $\mathcal{U}_i$ generates a range proof, showing that his total payment is under limit v_{max_i}. First he computes the Pedersen commitment of his total payment. Let Tx_i represent the set of transactions sent by $\mathcal{U}_i$, and $\{v_j, r_j | \mathtt{wtx}_j \in Tx_i\}$ denote the values and random numbers used to generate Walsh commitment in each transaction. Then he computes

$$v_{sum_i} \leftarrow \sum_{\mathtt{wtx}_j \in Tx_i} v_j, \quad r_{sum_i} \leftarrow \sum_{\mathtt{wtx}_j \in Tx_i} r_j$$

and the corresponding Pedersen commitment

$$pcom_{v_{sum_i}} \leftarrow g^{v_{sum_i}} h^{r_{sum_i}}$$

Finally, he generates a range proof, showing that $v_{sum_i} < v_{max_i}$:

$$\pi_{R_i} \leftarrow \mathsf{RangeProof.Prove}(pcom_{v_{sum_i}}, v_{sum_i}, r_{sum_i}, v_{max_i})$$

and sends π_{R_i} to $\mathcal{S}$.

(2). $\mathcal{S}$ sets $\mathbf{accu}_{\mathcal{T}} = \mathtt{block}_{\mathsf{last}}.\mathbf{accu}$, where $\mathtt{block}_{\mathsf{last}}.\mathbf{accu}$ is the **accu** field of the last block in the regulation period $\mathcal{T}$, which is considered as the aggregation result of all Walsh commitments of transactions that occurred during the whole regulation period.

Init an empty regulation result vector $\mathbf{b} = (0, ..., 0) = \{0\}^n$. To check if $\mathcal{U}_i$ has overspent, $\mathcal{S}$ extracts the Pedersen commitment of $\mathcal{U}_i$'s total payment by computing

$$pc_i \leftarrow \mathsf{WCom.Extract}(\mathbf{accu}_{\mathcal{T}}, \mathbf{code}_i, s)$$

Then verifies $\mathcal{U}_i$'s range proof

$$\mathbf{b}[i] \leftarrow \mathsf{RangeProof.Verify}(pc_i, v_{max}, \pi_{R_i})$$

$\mathcal{U}_i$ is suspected of overspending if $\mathbf{b}[i] == 0$, and $\mathcal{S}$ will investigate the specific consumption of users offline and impose corresponding punishment measures. Repeat the verification of range proofs for each user and output the regulation result $\mathbf{b}$.

Theorem 4. The WDAP scheme is secure (as defined in Definition 3), if the Walsh commitment, the consistency proof scheme and the native DAP system are secure.

A proof of Theorem 4 is provided in the Appendix C.

The WDAP protocol

I Initialization Phase

- Setup :
  - Inputs:
    1) λ: security parameter
  - Outputs:
    1) pp: public parameters
    2) s: extraction key
    3) $sk_{\mathcal{S}}$: secret key of $\mathcal{S}$
  - Procedures:
    1: $(\mathtt{pp}_{\mathsf{WCom}}, s) \leftarrow \mathsf{WCom.Setup}(1^\lambda)$
    2: $\mathtt{pp}_{\mathsf{DAP}} \leftarrow \mathsf{DAP.Setup}(1^\lambda)$
    3: $\mathtt{crs} \leftarrow \mathsf{CP.Setup}(\mathtt{pp}_{\mathsf{WCom}}, \mathtt{pp}_{\mathsf{DAP}})$
    4: set the regulation period $\mathcal{T}$
    5: send the extraction key s to $\mathcal{S}$ with a secure channel
    6: $\mathcal{S}$ generates key pair: $sk_{\mathcal{S}} \xleftarrow{\$} \mathbb{Z}_p^*$, $pk_{\mathcal{S}} \leftarrow g^{sk_{\mathcal{S}}}$
    7: publish the public parameters $\mathtt{pp} = (\mathtt{pp}_{\mathsf{WCom}}, \mathtt{pp}_{\mathsf{DAP}}, \mathtt{crs}, \mathcal{T}, pk_{\mathcal{S}})$

User $\mathcal{U}_i$, Supervisor $\mathcal{S}$:

- Register :
  - Procedures:

    $\mathcal{U}_i$: $(\mathtt{pp}) \rightarrow (sk_i, pk_i)$:

    1: $sk_i \xleftarrow{\$} \mathbb{Z}_p^*$, $pk_i \leftarrow \mathsf{Hsah}(sk_i)$. Publish pk_i
    2: send pk_i to $\mathcal{S}$ for registration

    $\mathcal{S}$: $(\mathtt{pp}, sk_{\mathcal{S}}, pk_i) \rightarrow (\mathbf{code}_i, v_{max_i}, \zeta_i)$:

    1: assign a Walsh code $\mathbf{code}_i$ to $\mathcal{U}_i$ and set a consumption limit v_{max_i} to $\mathcal{U}_i$.
    2: $g_{mul_i} \leftarrow \prod_{k=0}^{n-1} g_k^{\mathbf{code}_i[k]}$, $h_{mul_i} \leftarrow \prod_{k=0}^{n-1} h_k^{\mathbf{code}_i[k]}$
    3: $\zeta_i \leftarrow \mathsf{GenSig}(sk_{\mathcal{S}}, \mathtt{m})$, where $\mathtt{m} = \mathsf{Hash}(pk_i, g_{mul_i}, h_{mul_i})$

II Transacting Phase

User $\mathcal{U}_i$:

- $\mathsf{GenTx}(\mathtt{pp}, v, \mathbf{code}_i, sk_i, pk_i, \zeta_i, \mathtt{pri}_{\mathsf{DAP}}, \mathtt{pub}_{\mathsf{DAP}}) \rightarrow \mathtt{wtx}$
  - Procedures:
    1: $\mathtt{atx} \leftarrow \mathsf{DAP.GenTx}(v, \mathtt{pri}_{\mathsf{DAP}}, \mathtt{pub}_{\mathsf{DAP}})$
    2: $\mathbf{wcom} \leftarrow \mathsf{WCom.Commit}(\mathbf{code}_i, v)$
    3: $\Pi \leftarrow \mathsf{CP.Prove}(\mathtt{pp}, v, r, \mathbf{code}_i, sk_i, pk_i, \mathbf{wcom}, \zeta_i, \mathtt{atx})$
    4: set $\mathtt{wtx} = (\mathtt{atx}, \mathbf{wcom}, \Pi)$

Miner $\mathcal{M}$:

- SealBlock :
  - Inputs:
    1) $\{\mathtt{wtx}_j | 0 \le j < m\}$: a set of Walsh transaction
    2) $\mathtt{block}_{l-1}$: previous block
  - Outputs:
    1) $\mathtt{block}_l$: new block
  - Procedure:
    1: init the result vector $\mathbf{b} = (0, ..., 0) = \{0\}^n$
    2: **for** $i \in [0, j-1]$ **do**
    3: parse $\mathtt{wtx}_j$ as $(\mathtt{atx}_j, \mathbf{wcom}_j, \Pi_j)$
    4: $b_{j,0} \leftarrow \mathsf{DAP.VerifyTx}(\mathtt{atx}_j, \mathtt{pub}_{\mathsf{DAP}})$
    5: $b_{j,1} \leftarrow \mathsf{CP.Verify}(\mathbf{wcom}_j, \mathtt{atx}_j, \Pi_j)$
    6: $b_j \leftarrow b_{j,0} \cdot b_{j,1}$
    7: If $b_j == 1$, $\mathtt{wtx}_j$ is valid, otherwise invalid
    8: **end for**
    9: $\mathbf{accu} \leftarrow \mathsf{WCom.Aggregate}(\{\mathbf{wcom}_j | 0 \le j < m\})$
    10: **If** the block to be sealed is the first block of the regulation period
    11: $\mathbf{accu}_l \leftarrow \mathbf{accu}$
    12: **Else**
    13: $\mathbf{accu}_l \leftarrow \mathbf{accu} + \mathtt{block}_{l-1}.\mathbf{accu}$
    14: set new block $\mathtt{block} = (\mathtt{hedear}, \{\mathtt{atx}_j | 0 \le j < m\}, \mathbf{accu}_l)$

III Regulation Phase

User $\mathcal{U}$, Supervisor $\mathcal{S}$:

- Regulate**:**
  - Procedures:
  - $\mathcal{U}_i$: $(\mathtt{pp}, v_{max_i}, Tx_i) \rightarrow \pi_{R_i}$
    1: $v_{sum_i} =\leftarrow \sum_{\mathtt{wtx}_j \in Tx_i} v_j, r_{sum_i} = \sum_{\mathtt{wtx}_j \in Tx_i} r_j$,
    2: $pcom_{v_{sum_i}} \leftarrow g^{v_{sum_i}} h^{r_{sum_i}}$
    3: $\pi_{R_i} \leftarrow \mathsf{RangeProof.Prove}(pcom_{v_{sum_i}}, v_{sum_i}, r_{sum_i}, v_{max_i})$
    4: send π_{R_i} to $\mathcal{S}$
  - $\mathcal{S}$: $(\mathtt{block}_{\mathsf{last}}, \{v_{max_i}, \pi_{R_i} | 0 \le i < n\}) \rightarrow \mathbf{b} = \{0/1\}^n$
    1: set $\mathbf{accu}_{\mathcal{T}} = \mathtt{block}_{\mathsf{last}}.\mathbf{accu}$
    2: init the result vector $\mathbf{b} = (0, ..., 0) = \{0\}^n$
    3: **for** $i \in [0, n-1]$ **do**
    4: $pc_i \leftarrow \mathsf{WCom.Extract}(\mathbf{accu}_{\mathcal{T}}, \mathbf{code}_i, s)$
    5: $\mathbf{b}[i] \leftarrow \mathsf{RangeProof.Verify}(pc_i, v_{max}, \pi_{R_i})$
    6: **end for**

7: output $\mathbf{b}$, where $\mathcal{U}_i$ is suspected of overspending if $\mathbf{b}[i] == 0$

8 Consistency Proof

In this section, we present the concrete construction of generating and verifying the consistency proof Π uisng zk-SNARK and NIZK protocol.

Formally, CP.Setup, CP.Prove and CP.Verify are defined as follow.

- $\mathsf{CP.Setup}(\mathsf{pp_{WCom}}, \mathsf{pp_{DAP}}) \rightarrow \mathtt{crs}$. Generate the public parameters $\mathtt{crs}$.
- $\mathsf{CP.Prove}(\mathtt{pp}, v, r, \mathbf{code}_i, sk_i, pk_i, \mathbf{wcom}, \zeta_i, \mathtt{atx}) \rightarrow \Pi$. $\mathcal{U}_i$ generates the consistency proof Π.
- $\mathsf{CP.Verify}(\mathtt{pp}, \mathbf{wcom}, \mathtt{atx}, \Pi) \rightarrow 0/1$. $\mathcal{M}$ verifies the proof Π.

Definition 4. We consider that **wcom** is consistent with atx, if **wcom** and atx satisfy the following five conditions. (The variables with red background color are public inputs, while the variables with gray background color are private inputs.)

1. $\mathbf{wcom} = \mathsf{WCom.Commit}(v, r, \mathbf{code}_i)$. **wcom** is well-formed, i.e. $\mathbf{wcom} = (\mathbf{g}^v \mathbf{h}^r)^{\mathbf{code}_i}$.
2. $1 = \mathcal{R}_{\mathsf{DAP}}(v, \mathtt{atx})$, where $\mathcal{R}_{\mathsf{DAP}}(v, \mathtt{atx}) = 1$ if v is the payment amount in atx. (For example, in ZCash, we prove that v is committed in the transferred coin.)
3. $1 = \mathsf{VerSig}(pk_{\mathcal{S}}, \mathtt{m})$. $\mathcal{U}_i$ has a correct certificate.
4. $\mathtt{m} = \mathsf{Hash}(pk_i, g_{mul_i}, h_{mul_i})$.
5. $pk_i = \mathsf{Hash}(sk_i)$. $\mathcal{U}_i$ has the private key sk_i corresponding to the public key pk_i.

Though these five conditions can be proved directly by zk-SNARK, this imposes significant computational burden on the user, as it involves n group exponentiation operations. So the consistency proof protocol we present combines a batch power knowledge of exponentiation [28] check protocol and the zk-SNARK scheme to reduce the computational cost.

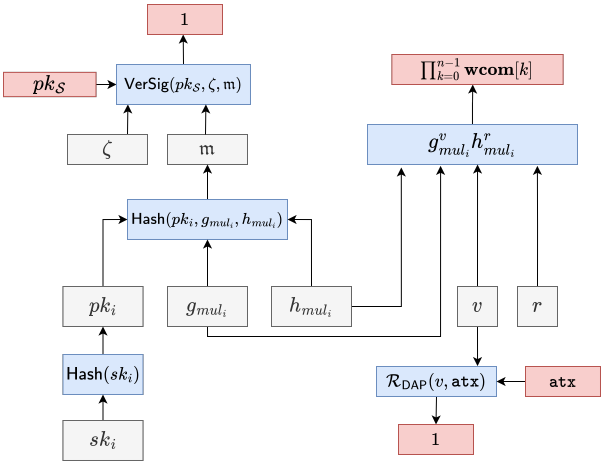


Fig. 5: zk-SNARK circuit

In the zk-SNARK part, a circuit C is built as shown in Fig. 5. $\mathcal{R}_{SNARK}$ is a binary relation constructed by C. The statement of $\mathcal{R}_{SNARK}$ is $\phi = \{pk_i, \mathbf{wcom}, \mathtt{atx}\}$. The witness of C is $\varpi = \{sk_i, pk_i, g_{mul_i}, h_{mul_i}, \zeta_i, v, r\}$. C imposes the following constraints on ϕ and ϖ:

1. $\prod_{k=0}^{n-1} \mathbf{wcom}[k] = g_{mul_i}^v h_{mul_i}^r$
2. $1 = \mathcal{R}_{\mathsf{DAP}}(v, \mathtt{atx})$
3. $\mathtt{m} = \mathsf{Hash}(pk_i, g_{mul_i}, h_{mul_i})$.
4. $1 = \mathsf{VerSig}(pk_{\mathcal{S}}, \mathtt{m})$.
5. $pk_i = \mathsf{Hash}(sk_i)$

The concrete construction is listed as follows.

Consistency proof

- **CP.Setup** $(\mathsf{pp_{WCom}}, \mathsf{pp_{DAP}}) \rightarrow \mathtt{crs}$
  1: $\{\alpha_k \xleftarrow{\$} \mathbb{Z}_p, \widehat{g_k} \leftarrow g_k^{\alpha_k}, \widehat{h_k} \leftarrow h_k^{\alpha_k}, \widehat{z_k} \leftarrow q^{\alpha_k} | 0 \le k < n\}$
  2: set $\widehat{\mathbf{g}} = (\widehat{g_0}, ..., \widehat{g_{n-1}})$, $\widehat{\mathbf{h}} = (\widehat{h_0}, ..., \widehat{h_{n-1}}) \in \mathbb{G}_1^n$
  3: set $\widehat{\mathbf{z}} = (\widehat{z_0}, ..., \widehat{z_{n-1}}) \in \mathbb{G}_2^n$
  4: generate the arithmetic circuit C
  5: $(pk_{zk}, vk_{zk}) \leftarrow \mathsf{ZK.GenKey}(\mathtt{pp}, \mathtt{C})$
  6: output $\mathtt{crs} = (\widehat{\mathbf{z}}, \widehat{\mathbf{g}}, \widehat{\mathbf{h}}, pk_{zk}, vk_{zk})$
  7: destroy $\{\alpha_k | 0 \le k < n\}$

User $\mathcal{U}_i$

- **CP.Prove**$(\mathtt{pp}, v, r, \mathbf{code}_i, sk_i, pk_i, \mathbf{wcom}, \zeta_i, \mathtt{atx}) \rightarrow \Pi$
  1: set statement $\phi = \{\mathbf{wcom}, \mathtt{atx}, pk_{\mathcal{S}}\}$
  2: set witness $\varpi = \{sk_i, pk_i, g_{mul_i}, h_{mul_i}, v, r, \zeta_i\}$
  3: $\pi \leftarrow \mathsf{ZK.Prove}(\phi, \varpi, pk_{zk})$
  4: $\sigma \leftarrow \prod_{k=0}^{n-1} (\widehat{g_k}^v \widehat{h_k}^r)^{\mathbf{code}_i[k]}$
  5: set $\Pi = (\sigma, \pi)$

Miner $\mathcal{M}$:

- **CP.Verify**$(\mathtt{pp}, \mathbf{wcom}, \mathtt{atx}, \Pi) \rightarrow 0/1$
  1: parse Π as (σ, π)
  2: set statement $\phi = \{\mathbf{wcom}, \mathtt{atx}, pk_{\mathcal{S}}\}$
  3: $b_0 \leftarrow \mathsf{ZK.Verify}(\phi, \pi, vk_{zk})$
  4: if $e(\sigma, q) == \prod_{k=0}^{n-1} e(\mathbf{wcom}[k], \widehat{z_k})$, set $b_1 = 1$, otherwise, $b_1 = 0$
  5: output $b = b_0 \cdot b_1$

Theorem 5. The consistency proof scheme is secure (as defined in Definition 2), if KEA3 holds and the zk-SNARK scheme is secure.

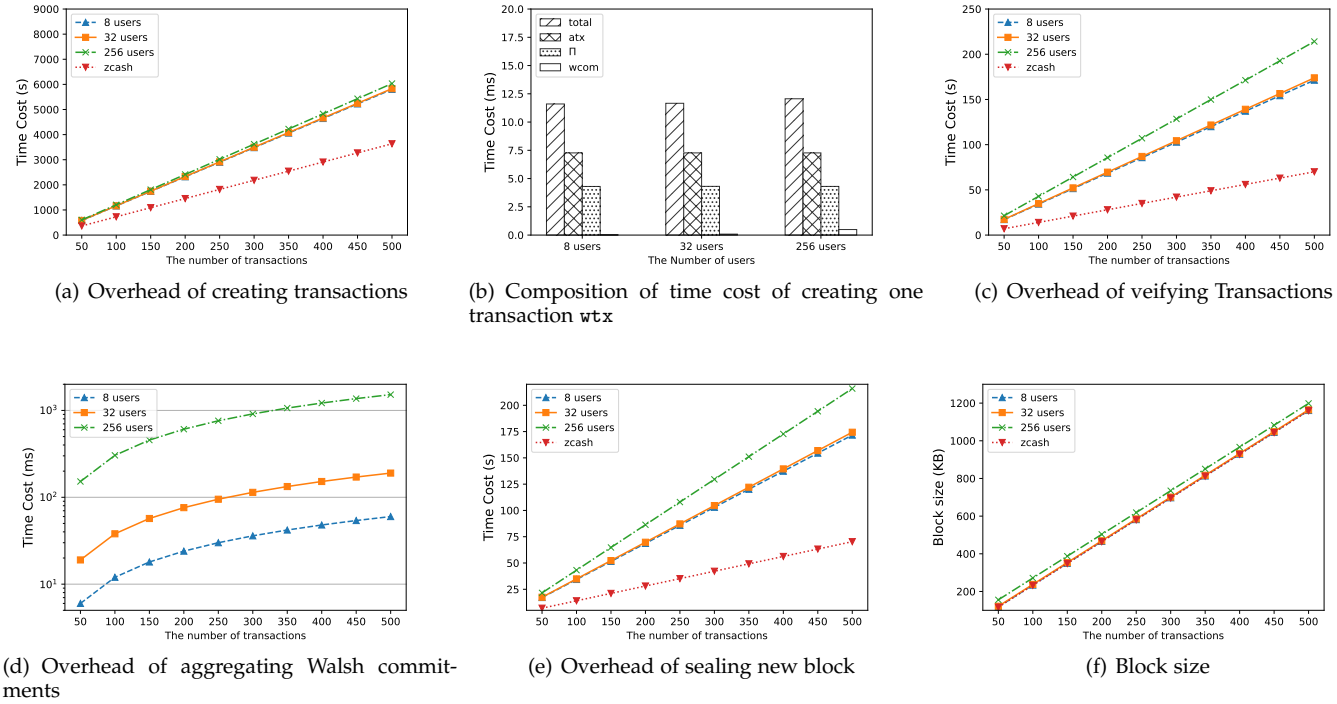
A proof of Theorem 5 is provided in the Appendix B.

9 Experiment

In this section, we evaluated the performance of the WDAP system. We implemented the WDAP scheme based on the ZCash system (ZCash-5.10 sapling transaction version). We ran and tested the performance of WDAP on the ZCash-5.10 regtestnet.

We conducted five experiments to evaluate the performance of WDAP. Experiments were conducted on a local PC with an Apple M1 Pro chip, 8GB RAM, and Ubuntu 20.04 OS. The programming languages used in the experiments were cpp and rust, and we used zero-knowledge proof tools such as bellman and snarkjs to generate and verify zk proofs in the system.

Setup. In Experiment I, we tested the time taken in the Setup phase of the WDAP system in cases with 8, 32, and 256 users. The results are shown in Table 3. The setup phase of the WDAP system consists of generating three main parts: crs, $\mathsf{pp_{DAP}}$ and the other public parameters. The majority



(a) Overhead of creating transactions

(b) Composition of time cost of creating one transaction `wtx`

(c) Overhead of veifying Transactions

(d) Overhead of aggregating Walsh commitments

(e) Overhead of sealing new block

(f) Block size

Fig. 6: The performance of WDAP components deployed in ZCash.

TABLE 3: Setup cost

User number	`crs`	Other parameters	Total
8	2m52.03s	0.10s	2m52.13s
32	2m52.51s	0.23s	2m52.74s
256	2m54.23s	1.11s	2m54.34s

of the time is spent on the CP.Setup for generating `crs`. The public parameters of the ZCash system was directly download from the official website[3] and we did not measure its generation time. The total time for the entire Setup phase was approximately 3 minutes, highlighting the efficiency of the WDAP system in the Setup phase.

Generate transactions. In Experiment II, we tested the transaction generation time of the WDAP system with 8 users, 32 users, and 256 users as shown in Fig. 6(a). In the case of 256 users, it takes approximately 12.00 s to generate one transaction, and generating 500 transactions takes 6033 s. Fig. 6(b) illustrates the composition of time required to generate a transaction `wtx`. In the case of 256 users, time cost of generating the native transaction of Zcash `atx` is 7.27 s, accounting for 60% of total time. Time cost of computing the consistency proof Π costs 4.30s, accounting for 35% of total time. The generation time of **wcom** is significantly shorter than the others.

Verify transactions. In Experiment III, we tested the transaction verification time. The verification time for 8 users and 32 users is almost the same as shown in Fig. 6(c). In the case of 256 users, it takes approximately 0.43 s to verify one transaction, and verifying 500 transactions takes less than 250 s. The results of Experiment III demonstrate the efficiency of the WDAP system in the transaction verification phase.

Seal block. In Experiment IV, we tested the time required for miners to seal blocks and the block sizes. Fig. 6(d) shows the time consumption of aggregating commitments, and Fig. 6(e) displays the total time required to seal blocks. In the case of 256 users, it takes less than 220 s for a miner to seal a block containing 500 transactions. Fig. 6(f) shows the block sizes in the cases of 8 users, 32 users, and 256 users, which are almost the same as the block sizes in the original Zcash. The block size for a block containing 500 transactions is approximately 1170 KB, reflecting the high efficiency of WDAP in terms of storage cost.

TABLE 4: Regulation cost

user number	8	32	256
$\mathcal{U}$			
generate range proof (ms)	65.0	65.0	65.0
communication overhead (KB)	0.34	0.34	0.34
$\mathcal{S}$			
verify all range proofs (ms)	48.0	224.0	1792.0
Extract (ms)	0.32	4.9	384.9
Total (ms)	48.3	228.9	2176.9
communication overhead (KB)	2.94	11.78	94.20

Regulate. In Experiment V, we tested the consumption of the regulation phase in the WDAP system in the case of 8 users, 32 users and 256 users as shown in Table 4. The regulation time consumption is only related to the number of users. The time cost of a user generating proof is 65 ms. It takes 2.17 s for the supervisor to run the regulation algorithm, of which the verification range proof accounts for the main time consumption, reflecting the efficiency of

3. https://download.z.cash/downloads/

WDAP in the regulation phase.

10 FURTHER DISCUSSION

In WDAP, the length of commitments is linearly related to the number of users, which is not conducive to scalability. As the number of users continues to grow, we choose to use a grouping approach. For example, for a system with 2560 users, 256 users are divided into a group. Users need to clearly indicate the group they belong to when making transactions. Accordingly, the **accu** field of a block will store 10 aggregation result of 10 groups. This approach is somewhat similar to ring signatures, where attackers can only know the group to which the transaction sender belongs, but they cannot ascertain the identity of the individual transactor.

11 CONCLUSION

In this paper, we have proposed WDAP, an efficient regulation scheme of decentralized anonymous payment that supports regulation while protecting privacy. We set two regulation policies that users who have exceeded their spending limits within a certain period will be identified during the regulation process and the supervisor can trace any anonymous transactions. To conduct regulation, we designed an efficient aggregatable and extractable commitment scheme, Walsh commitment. User generates a commitment of the payment value when transferring money in the native anonymous transaction and generate a consistency proof to prove its validity. The supervisor can retrieve a Pedersen commitment of the total payment of each user and can check if the user violates the regulation policies. A future direction is to design schemes to achieve higher efficiency and support more regulation policies.

APPENDIX A
SECURITY PROOFS FOR WALSH COMMITMENT

A.1 Hiding property

A.1.1 Hidng property for $\mathcal{A}_2$

We prove that given a $\mathtt{crs} \leftarrow \mathsf{CP.Setup}$ and a commitment **accu**, adversary $\mathcal{A}_2$, $\mathcal{A}_2$ has no information about the committed values. Given $\mathbf{code}_i$, v, r and any v', there must exists $r' = (v - v')\beta^{-1} + r$,

$$\begin{aligned}
&\mathsf{WCom.Commit}(\mathbf{code}_i, v', r') \\
&= (\mathbf{g}^{v'}\mathbf{h}^{r'})^{\mathbf{code}_i} \\
&= (\mathbf{g}^{v'}\mathbf{h}^{(v-v')\beta^{-1}}\mathbf{h}^{r})^{\mathbf{code}_i} \\
&= (\mathbf{g}^{v}\mathbf{h}^{r})^{\mathbf{code}_i} \\
&= \mathsf{WCom.Commit}(\mathbf{code}_i, v, r)
\end{aligned}$$

So the probability of **wcom** committing to any v equals to $\frac{1}{q}$, where q is the order of the group. The Walsh commitment is perfect-hiding. □

A.1.2 Hidng property for $\mathcal{A}_1$

Theorem 6. *If the l-DDHI problem is hard, the ElGamal encryption scheme satisfies hiding property.*

PROOF. We use mathematical induction and security reduction to prove this theorem. First of all, we define a new experiment as follow. Obviously, $\mathsf{exp}^{\mathsf{k\text{-}Hiding}}_{\mathcal{A}_1}(\lambda) = \mathsf{exp}^{\mathsf{Hiding}}_{\mathcal{A}_1}(\lambda)$ when $k = n$.

- **Hiding experiment 1** $\mathsf{exp}^{\mathsf{k\text{-}Hiding}}_{\mathcal{A}_1}(\lambda)$
  1: $(\mathsf{pp}, s) \leftarrow \mathsf{WCom.Setup}(1^\lambda)$
  2: $(\mathbf{code}_{i_0}, v_0, \mathbf{code}_{i_1}, v_1) \leftarrow \mathcal{A}_1(\mathsf{pp})$
  3: $b \xleftarrow{\$} \{0, 1\}$
  4: $\mathbf{wcom}_b \leftarrow \mathsf{WCom.Commit}(\mathbf{code}_{i_b}, v_b)$
  5: $b' \leftarrow \mathcal{A}_1(\overline{k\text{-}\mathbf{wcom}_b}, \mathbf{code}_{i_0}, v_0, \mathbf{code}_{i_1}, v_1)$, where $\overline{k\text{-}\mathbf{wcom}_b}$ is a sub-vector of $\mathbf{wcom}_b$ and $\overline{k\text{-}\mathbf{wcom}_b} = (\mathbf{wcom}_b[0], ..., \mathbf{wcom}_b[k-1])$.
  6: return 1 if $b = b'$ or 0 otherwise

For the same reason in A.1.1, the value v is perfect hiding for any adversary $\mathcal{A}_1$, and thus any adversaries $\mathcal{A}_1$ cannot distinguish $g^{v_b}h^{r_b}$ from

Base Step: If $k = 1$:

For all PPT adversaries $\mathcal{A}_1$, $|Pr[\mathsf{exp}^{\mathsf{1\text{-}Hiding}}_{\mathcal{A}_1}(\lambda) = 1] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$.

PROOF. This holds naturely since $\overline{1\text{-}\mathbf{wcom}_b} = (\mathbf{wcom}_b[0])$ is just one Pedersen commitments to $v \cdot \mathbf{code}_i[0]$. The hiding property derives from the Pedersen commitment scheme.

Recurrence Step: if $k \geq 0$:

If the l-DDHI problem is hard and for all PPT adversaries $\mathcal{A}_1$, $|Pr[\mathsf{exp}^{\mathsf{k\text{-}Hiding}}_{\mathcal{A}_1}(\lambda) = 1] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$. Then $|Pr[\mathsf{exp}^{\mathsf{(k+1)\text{-}Hiding}}_{\mathcal{A}_1}(\lambda) = 1] - \frac{1}{2}| \leq \mathsf{negl}(\lambda)$ also holds.

PROOF. Suppose there exists an adversary $\mathcal{A}_1$ who can break the hiding property. We construct a simulator $\mathcal{B}$ to solve the l-DDHI problem. Given as input a problem instance $(\widehat{g}, \widehat{g}^a, \widehat{g}^{a^2}, ..., \widehat{g}^{a^{k-1}}, z)$, $\mathcal{B}$ runs $\mathcal{A}_1$ and works as follows.

Setup. Let $\mathtt{bp} = \mathsf{BilGen}(1^\lambda) = (p, e, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g, q)$, where specially set $g = \widehat{g}^{a^{k-1}}$ $\mathcal{B}$ randomly chooses $\beta \in \mathbb{Z}_p$ and computes $h = g^\beta$. Set $(g_0, ..., g_{k-1}, g_k) = (\widehat{g}^{a^{k-1}}, ..., \widehat{g}, z)$ and $(h_0, ..., h_{k-1}, h_k) = (g_0^\beta, ..., g_{k-1}^\beta, g_k^\beta)$.

Challenge. $\mathcal{A}_1$ outputs two distinct tuples $(\mathbf{code}_{i_0}, v_0, \mathbf{code}_{i_1}, v_1)$ to be challenged. $\mathcal{B}$ chooses $r_0 \leftarrow \mathbb{Z}_p$ and computes $r_1 = (v_0 - v_1')\beta^{-1} + r_0$ such that $g_k^{v_0}h_k^{r_0} = g_k^{v_1}h_k^{r_1}$. Denote $\gamma = v_0 + \beta \cdot r_0$. $\mathcal{B}$ randomly chooses $b \in \{0, 1\}$ and sets the challenge commitment

$$\overline{(k+1)\text{-}\mathbf{wcom}_b} = ((\widehat{g}^{a^{k-1}})^{\gamma \cdot \mathbf{code}_{i_b}[0]}, ..., \widehat{g}^{\gamma \cdot \mathbf{code}_{i_b}[k-1]}, z^{\gamma \cdot \mathbf{code}_{i_b}[k]})$$

where $(\widehat{g}, \widehat{g}^a, \widehat{g}^{a^2}, ..., \widehat{g}^{a^{k-1}}, z)$ are from the problem instance. Let $s = \frac{1}{a}$. If $z = \widehat{g}^{\frac{1}{a}}$ we have

$$\begin{aligned}
&\overline{(k+1)\text{-}\mathbf{wcom}_b} = ((\widehat{g}^{a^{k-1}})^{\gamma \cdot \mathbf{code}_{i_b}[0]}, ..., \widehat{g}^{\gamma \cdot \mathbf{code}_{i_b}[k-1]}, z^{\gamma \cdot \mathbf{code}_{i_b}[k]}) \\
&= (g^{\gamma \cdot \mathbf{code}_{i_b}[0]}, ..., (g^{s^{k-1}})^{\gamma \cdot \mathbf{code}_{i_b}[k-1]}, (g^{s^k})^{\gamma \cdot \mathbf{code}_{i_b}[k]}) \\
&= ((g_0^{v_b}h_0^{r_b})^{\mathbf{code}_{i_b}[0]}, ..., (g_{k-1}^{v}h_{k-1}^{r_b})^{\mathbf{code}_{i_b}[k-1]}, (g_k^{v}h_k^{r_b})^{\mathbf{code}_{i_b}[k]})
\end{aligned}$$

Therefore, $\overline{(k+1)\text{-}\mathbf{wcom}_b}$ is a correct challenge commitment, which committs to $((\mathbf{code}_{i_b}[0], ..., \mathbf{code}_{i_b}[k]), v_b)$.

Guess. $\mathcal{A}$ outputs a guess b' of of b. The simulator outputs true if $b' = b$. Otherwise, false.

Probability of breaking the challenge commitment. Denote the success probability of $\mathcal{A}$ as ε.

If z is true, the simulation is indistinguishable from the real attack, and thus the adversary has probability $\frac{1}{2}+\frac{\varepsilon}{2}$ of guessing the committed message correctly. If z is false, it is easy to see that the challenge commitment

$$\overline{(k+1)\text{-}\mathbf{wcom}_b} = (\overline{(k)\text{-}\mathbf{wcom}_b}, z^{\gamma\cdot\mathbf{code}_{i_b}[k]})$$

where $\overline{k\text{-}\mathbf{wcom}_b}$ is indistinguishable and $z^{\gamma\cdot\mathbf{code}_{i_b}[k]}$ is a one-time pad because, which is random and cannot be calculated from the other parameters given to the adversary. Therefore, the adversary only has probability $\frac{1}{2}$ of guessing the committed code bit $\mathbf{code}_{i_b}[k]$ correctly.

In conclusion, if the l-DDHI problem is hard and for all PPT adversaries $\mathcal{A}_1$, $|Pr[\mathsf{exp}^{\mathsf{k\text{-}Hiding}}_{\mathcal{A}_1}(\lambda)=1]-\frac{1}{2}|\le \mathsf{negl}(\lambda)$. then $|Pr[\mathsf{exp}^{(\mathsf{k+1})\text{-}\mathsf{Hiding}}_{\mathcal{A}_1}(\lambda)=1]-\frac{1}{2}|\le \mathsf{negl}(\lambda)$ also holds.

Combine the **Recurrence Step** and the **Base Step**, this completes the proof of the theorem. □

A.2 Binding property

Suppose there exists an adversary $\mathcal{A}_1$ or $\mathcal{A}_2$ who can break the binding property. We construct a simulator $\mathcal{B}$ to solve the DL problem. Given as input a problem instance $(\widehat{g},\widehat{h})$, $\mathcal{B}$ runs $\mathcal{A}_1$ and works as follows.

Setup Let $\mathsf{bp}=\mathsf{BilGen}(1^\lambda)=(p,e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e,g,q)$, where specially set $g=\widehat{g}, h=\widehat{h}$. $\mathcal{B}$ randomly chooses $s\xleftarrow{\$}\mathbb{Z}_p$, Set $\mathbf{g}=(g_0,...,g_{n-1}),\mathbf{h}=(h_0,...,h_{n-1})\in\mathbb{G}_1^n$, where $g_i=\widehat{g}^{s^i}, h_i=\widehat{h}^{s^i}, 0\le i<n$. Output the public parameters $\mathsf{pp}=(\mathsf{bp},\mathbf{g},\mathbf{h},\{\mathbf{code}_i|0\le i<n\})$,

Challenge Suppose that there exists $\mathcal{A}_1$ or $\mathcal{A}_2$ that breaks the binding property by outputting $(\mathbf{code}_i,v,r,\mathbf{code}_{i'},v',r')$ such that

$$(\mathbf{g}^{v'}\mathbf{h}^{r'})^{\mathbf{code}_i}=(\mathbf{g}^{v}\mathbf{h}^{r})^{\mathbf{code}_{i'}}\wedge(\mathbf{code}_i\neq\mathbf{code}_{i'}\vee v\neq v')$$

$\mathcal{B}$ computes $\beta=\mathsf{log}_g^h$ by computing

$$\beta=\frac{\sum_{k=0}^{n-1}(s^k\cdot v\cdot\mathbf{code}_i[k]-s^k\cdot v'\cdot\mathbf{code}_{i'}[k])}{\sum_{k=0}^{n-1}(s^k\cdot r'\cdot\mathbf{code}_{i'}[k]-s^k\cdot r\cdot\mathbf{code}_i[k])}$$

Significantly, it is impossible that constructed $\sum_{k=0}^{n-1}(s^k\cdot r'\cdot\mathbf{code}_{i'}[k]-s^k\cdot r\cdot\mathbf{code}_i[k])=0\wedge(\mathbf{code}_i\neq\mathbf{code}_{i'}\vee v\neq v')$ by setting $r'=-r\wedge\mathbf{code}_{i'}[k]=-\mathbf{code}_i[k]$. Since $\langle\mathbf{code}_{i'}[k],\mathbf{code}_i[k]\rangle=-n$, $\mathbf{code}_i$ and $\mathbf{code}_{i'}$ cannot be legal codes at the same time.

Therefore, $\mathcal{B}$ can solve the instance of the DL problem with the same success probability of $\mathcal{A}$. The Walsh commitment satisfies binding. □

APPENDIX B
SECURITY PROOFS FOR CONSISTENCY PROOF

B.1 Equivalence

Theorem 7. *The Relation 1 $\mathcal{R}_1$:*

$$\mathcal{R}_1(\phi=\mathbf{wcom};\varpi=(v,r,\mathbf{code}_i))=1\ \text{ if }\ \mathbf{wcom}=(\mathbf{g}^v\mathbf{h}^r)^{\mathbf{code}_i}$$

is computationally equivalent to the combinations of the following two relations:

$$\mathcal{R}_{1.1}(\phi=\mathbf{wcom};\varpi=(\mathbf{x},\mathbf{y}))=1\ \text{ if }\ \mathbf{wcom}[k]=g_k^{\mathbf{x}[k]}h_k^{\mathbf{y}[k]},0\le k<n$$

$$\mathcal{R}_{1.2}(\phi=\mathbf{wcom};\varpi=(v,r))=1\ \text{ if }\ \prod_{k=0}^{n-1}\mathbf{wcom}[k]=g_{mul_i}^v h_{mul_i}^r$$

i.e. for all PPT *adversaries* $\mathcal{A}\in\{\mathcal{A}_1,\mathcal{A}_2\}$, $Pr[\mathsf{exp}^{\mathsf{Equ}}_{\mathcal{A}}(\lambda)=1]\le\mathsf{negl}(\lambda)$

- **Equivalence experiment** $\mathsf{exp}^{\mathsf{Equ}}_{\mathcal{A}_1}(\lambda)$
  1: $(\mathsf{pp},s)\leftarrow\mathsf{WCom.Setup}(1^\lambda)$
  2: $(\mathbf{x},\mathbf{y},v,r,\mathbf{wcom},\mathbf{code}_i)\leftarrow\mathcal{A}_1(\mathsf{pp})$
  3: $g_{mul_i}\leftarrow\prod_{k=0}^{n-1}g_k^{\mathbf{code}_i[k]}\ h_{mul_i}\leftarrow\prod_{k=0}^{n-1}h_k^{\mathbf{code}_i[k]}$
  4: return 1 if $(\prod_{k=0}^{n-1}\mathbf{wcom}[k]=g_{mul_i}^v h_{mul_i}^r\wedge$
  $\mathbf{wcom}[k]=g_k^{\mathbf{x}[k]}h_k^{\mathbf{y}[k]})\wedge$
  $(\mathbf{wcom}\neq(\mathbf{g}^v\mathbf{h}^r)^{\mathbf{code}_i}),0\le k<n$ or 0 otherwise

EQUIVALENCE PROOF. Suppose that there exists $\mathcal{A}_1$ or $\mathcal{A}_2$ that breaks the equivalence by outputting $(\mathbf{x},\mathbf{y},v,r,\mathbf{wcom},\mathbf{code}_i)\leftarrow\mathcal{A}_1(\mathsf{pp})$ such that $(\prod_{k=0}^{n-1}\mathbf{wcom}[k]=g_{mul_i}^v h_{mul_i}^r\wedge\ \mathbf{wcom}[k]=g_k^{\mathbf{x}[k]}h_k^{\mathbf{y}[k]})\wedge(\mathbf{wcom}\neq(\mathbf{g}^v\mathbf{h}^r)^{\mathbf{code}_i}),0\le k<n$. We can construct an algorithm $\mathcal{B}$ that uses $\mathcal{A}$ to efficiently compute $\beta=\mathsf{log}_g^h$ by computing

$$\beta=\frac{\sum_{k=0}^{n-1}-s^k\cdot\mathbf{x}[k]+s^k\cdot\mathbf{code}_i[k]\cdot v}{\sum_{k=0}^{n-1}s^k\cdot\mathbf{y}[k]-s^k\cdot\mathbf{code}_i[k]\cdot r}$$

□

B.2 Correctness of NIZK scheme

$$\mathsf{PoK}_{1.1}\{(\mathbf{x},\mathbf{y}):\mathbf{wcom}[k]=g_k^{\mathbf{x}[k]}h_k^{\mathbf{y}[k]},0\le k<n\}$$

CORRECTNESS PROOF

$$\begin{aligned}e(\sigma,q)&=e(\prod_{k=0}^{n-1}(\widehat{g_k}^v\widehat{h_k}^r)^{\mathbf{code}_i[k]},q)\\&=\prod_{k=0}^{n-1}e((\widehat{g_k}^v\widehat{h_k}^r)^{\mathbf{code}_i[k]},q)\\&=\prod_{k=0}^{n-1}e((g^{\alpha_k\cdot v}h^{\alpha_k\cdot r})^{\mathbf{code}_i[k]},q)\\&=\prod_{k=0}^{n-1}e((g^vh^r)^{\mathbf{code}_i[k]},q^{\alpha_k})\\&=\prod_{k=0}^{n-1}e(\mathbf{wcom}[k],\widehat{z_k})\end{aligned}$$

B.3 Soundness of batch power knowledge of Exponent proof

Knowledge of Exponent Assumption: *(KEA3). For any* PPT *adversary $\mathcal{A}$, there exists a* PPT *extractor $\mathcal{X}_\mathcal{A}$, such that.*

$$Pr\left[\begin{array}{c|c}\mathsf{bp}=\mathsf{BilGen}(1^\lambda) & \widehat{A}=A^\alpha\bigwedge\\ \alpha\xleftarrow{\$}\mathbb{Z}_p,h\xleftarrow{\$}\mathbb{G}_1 & A\neq g^vh^r\\ (A,\widehat{A};v,r)\leftarrow(\mathcal{A}||\mathcal{X}_\mathcal{A})(\mathsf{bp},g^\alpha,h,h^\alpha) & \end{array}\right]\le\mathsf{negl}(\lambda).$$

SOUNDNESS PROOF.

Here we give the specific definition of Soundness of batch knowledge of exponentiation argument protocol: For all PPT adversaries $\mathcal{A}$, $|Pr[\mathsf{exp}^{\mathsf{Soundness}}_{\mathcal{A}_1}(\lambda)=1]|\le\mathsf{negl}(\lambda)$.

- **Soundness experiment** $\exp^{\mathsf{Soundness}}_{\mathcal{A}_1}(\lambda)$
  1: $(\mathsf{pp}, s) \leftarrow \mathsf{Setup}(1^\lambda)$
  2: $(\sigma, \mathbf{wcom}; \mathbf{x}, \mathbf{y}) \leftarrow (\mathcal{A}||\mathcal{X}_\mathcal{A})(\mathsf{bp}, \mathbf{g}, \mathbf{h}, \widehat{\mathbf{g}}, \widehat{\mathbf{h}}, \widehat{\mathbf{z}})$
  3: return 1 if $\sigma = \prod_{k=0}^{n-1} \mathbf{wcom}[k]^{\alpha_k} \wedge$ $(\exists k, \mathbf{wcom}[k] \neq \mathbf{g}[k]^{\mathbf{x}[k]}\mathbf{h}[k]^{\mathbf{y}[k]})$ or 0 otherwise

Without loss of generality, assume that $\mathcal{A}$ break the soundness by outputting $\sigma, \mathbf{wcom}$ and the extractor $\mathcal{X}_\mathcal{A}$ can output $\mathbf{x}, \mathbf{y}$, such that for a fixed $k^*, \sigma = \prod_{k=0}^{n-1} \mathbf{wcom}[k]^{\alpha_k} \wedge \mathbf{wcom}[k^*] \neq \mathbf{g}[k]^{\mathbf{x}[k^*]}\mathbf{h}[k]^{\mathbf{y}[k^*]}$

We can construct a simulator $\mathcal{B}$ to solve the KEA3 problem. Given as input a problem instance $(\mathsf{bp}, \widetilde{g}, \widetilde{h}, \widetilde{g^\gamma}, \widetilde{h^\gamma})$, $\mathcal{B}$ runs $\mathcal{A}$ and works as follows.

Setup. $\mathcal{B}$ randomly chooses $s \xleftarrow{\$} \mathbb{Z}_p$ and set $g = \widetilde{g}, h = \widetilde{h}$. Set $\mathbf{g} = (g_0, ..., g_{n-1}), \mathbf{h} = (h_0, ..., h_{n-1}) \in \mathbb{G}_1^n$, where $\{g_i \leftarrow g^{s^i}, h_i \leftarrow h^{s^i} | 0 \leq i < n\}$. Computes $\{\alpha_k \xleftarrow{\$} \mathbb{Z}_p, \widehat{g_k} \leftarrow g_k^{\alpha_k}, \widehat{h_k} \leftarrow h_k^{\alpha_k} | 0 \leq k < n-1 \wedge k \neq k^*\}$. Set $\widehat{g_{k^*}} = (\widetilde{g^\gamma})^{s^{k^*}}, \widehat{h_{k^*}} = (\widetilde{h^\gamma})^{s^{k^*}}$. Set $\widehat{\mathbf{g}} = (\widehat{g_0}, ..., \widehat{g_{n-1}})$, $\widehat{\mathbf{h}} = (\widehat{h_0}, ..., \widehat{h_{n-1}})$. Then, publishes $\{\mathbf{g}, \mathbf{h}, \widehat{\mathbf{g}}, \widehat{\mathbf{h}}\}$.

Forgery. With the inputs $\{\mathbf{g}, \mathbf{h}, \widehat{\mathbf{g}}, \widehat{\mathbf{h}}\}$, $\mathcal{A}_1$ outputs $(\sigma, \mathbf{wcom})$ and the extractor $\mathcal{X}_{\mathcal{A}_1}$ outputs $(\mathbf{x}, \mathbf{y})$ s.t. $\sigma = \prod_{k=0}^{n-1} \mathbf{wcom}[k]^{\alpha_k} \bigwedge \mathbf{wcom}[k] \neq \mathbf{g}[k]^{\mathbf{x}[k]}\mathbf{h}[k]^{\mathbf{y}[k]}$.

Challenge. $\mathcal{B}$ computes

$$A = \mathbf{wcom}[k], \quad \widehat{A} = \frac{\sigma}{\prod_{k \in [0,n-1] n \{k^*\}} \mathbf{wcom}[k]}$$

$$v = \mathbf{x}[k], \quad r = \mathbf{y}[k]$$

Set the output as $(A, \widehat{A}; v, r) \leftarrow (\mathcal{B}||\mathcal{X}_\mathcal{B})$, such that $\widehat{A} = A^\alpha \bigwedge A \neq g^v h^r$. Therefore, $\mathcal{B}$ can solve the instance of the KEA3 problem with the same success probability of $\mathcal{A}$. The Walsh commitment satisfies binding. □

B.4 Zero Knowledge of NIZK protocol

For any PPT adversary $\mathcal{A}$ and the witness $\mathbf{x}, \mathbf{y}$, there must exist a PPT simulator $\mathcal{S}$, such that the outputs of the following two experiments are indistinguishable.

- **Real experiment**
  1: $(\mathsf{pp}) \leftarrow \mathsf{Setup}(1^\lambda)$
  2: $\mathbf{wcom} \leftarrow \mathsf{WCom.Commit}(\mathsf{pp}, \mathbf{code}_i, v, r)$
  3: $\sigma \leftarrow \mathsf{CP.Prove}(\mathsf{pp}, \mathbf{x}, \mathbf{y})$, where $\mathbf{x} = v \cdot \mathbf{code}_i, \mathbf{y} = r \cdot \mathbf{code}_i$
  4: The output of the experiment is $(\mathsf{pp}, \mathbf{wcom}, \sigma)$
- **Ideal experiment**
  1: $(\mathsf{pp}, \{\alpha_k | 0 \leq k < n\}) \leftarrow \mathcal{S}im(1^\lambda)$
  2: $\mathbf{wcom}' \leftarrow \mathcal{S}im(\mathsf{pp})$
  3: $\sigma' \leftarrow \mathcal{S}im(\mathsf{pp}, \mathbf{wcom}, \{\alpha_k | 0 \leq k < n\})$
  4: The output of the experiment is $(\mathsf{pp}, \mathbf{wcom}', \sigma')$

To prove the ZK property, we describe the simulator $\mathcal{S}im$ for the ideal experiment above. In Step 1 of the ideal experiment, $\mathcal{S}im$ runs $(\mathsf{pp}, s) \leftarrow \mathsf{Setup}(1^\lambda)$ honestly, but knows the trapdoor $\tau = \{\alpha_k | 0 \leq k < n\}$. $\mathcal{S}im$ then outputs $\mathbf{wcom}' \xleftarrow{\$} \mathbb{G}_1^n$. At this point we note that since S knows the trapdoor $\tau = \{\alpha_k | 0 \leq k < n\}$, he can create witnesses for arbitrary values with respect to $\mathbf{wcom}'$ by outputting $\sigma' = \prod_{k=0}^{n-1} \mathbf{wcom}'[k]^{\alpha_k}$. It is clearly that σ' can pass the verification since $e(\sigma, q) = \prod_{k=0}^{n-1} e(\mathbf{wcom}[k], \widehat{z_k})$

The simulated witnesses have the same distribution as honest witnesses since $\mathbf{wcom}'$ is chosen at random.

$\mathcal{S}im$ can compute $\sigma' = \prod_{k=0}^{n-1} \mathbf{wcom}[k]^{\alpha_k}$ and apparently $e(\sigma', q) = \prod_{k=0}^{n-1} e(\mathbf{wcom}[k], \widehat{z_k})$. The simulated output σ can pass the verification. It is clear that this simulation is indistinguishable from a real excution.

Since the outputs from the experiments are indistinguishable, the ZK property is satisfied. □

B.5 Security properties of zk-SNARK scheme

The correctness, soundness and zero-knowledge of zk-SNARK scheme derives from the zk-SNARK algorithm.

APPENDIX C
INDISTINGUISHABILITY OF WDAP TRANSACTIONS

Indistinguishability of Walsh transaction We use $\mathcal{G}_{real}$ to denote the experiment $\exp^{\mathsf{Ind}}_{\mathcal{A}_1}(\lambda)$ executed in the real world. To analyze the security of WDAP, we design a simulation $\mathcal{G}_{sim}$. In $\mathcal{G}_{sim}$, the challenger $\mathcal{C}$ interacts with $\mathcal{A}_1$ as in $\mathcal{G}_{real}$. The only modification is that $\mathcal{C}$ outputs a challenge transaction $\mathtt{wtx}_{\mathcal{G}_{sim}} = \{\mathtt{atx}^*, \mathbf{wcom}^*, \Pi^*\}$ in which $\{\mathtt{atx}^*, \mathbf{wcom}^*, \Pi^*\}$ are independent of b.

Game $\mathcal{G}_1$. This game $\mathcal{G}_1$ is the same as $\mathcal{G}_{real}$, but the only modification is that $\mathcal{C}$ uses trapdoor τ to simulate proofs in the challenge transaction. To generate trapdoor, $\mathcal{C}$ executes $(pk_{zk}, vk_{zk}, \tau) \leftarrow \mathsf{CP.Setup}(1^\lambda, C)$ in the setup phase. Then $\mathcal{C}$ computes $\Pi^* \leftarrow \mathsf{CP.Setup}(\phi, pk_{zk}, \tau)$. The challenge transaction in $\mathcal{G}_1$ is $\mathtt{wtx}_{\mathcal{G}_1} = \{\mathtt{atx}_b, \mathbf{wcom}_b, \Pi^*\}$. Considering that our consistency proof scheme satisfies zero-knowledge, the distribution of Π^* is identical to that of Π_b in $\mathcal{G}_{real}$. Therefore, $\mathcal{E}_{\mathcal{G}_1} < \mathbf{neg}(\lambda)$.

Game $\mathcal{G}_2$. This game is the same as $\mathcal{G}_1$, but the only modification is that $\mathcal{C}$ uses $\mathbf{wcom}^*$ to replace $\mathbf{wcom}_b$. $\mathcal{C}$ computes $\mathbf{wcom}^* \leftarrow \mathsf{WCom.Com}(v^*, \mathbf{code}^*)$, where $v^* \neq v_b \wedge \mathbf{code}^* \neq \mathbf{code}_{i_b}$. The challenge transaction in $\mathcal{G}_2$ is $\mathtt{wtx}_{\mathcal{G}_2} = \{\mathtt{atx}_b, \mathbf{wcom}^*, \Pi^*\}$. Since the Walsh commitment scheme satisfies Hiding, the advantage of $\mathcal{A}_1$ in distinguishing $\mathbf{wcom}^*$ from $\mathbf{wcom}_{i_b}$ is negligible. Thus, $|\mathcal{E}_{\mathcal{G}_2} - \mathcal{E}_{\mathcal{G}_1}| < \mathsf{negl}(\lambda)$.

Game $\mathcal{G}_{sim}$. This game is the same as $\mathcal{G}_2$, but the only modification is that $\mathcal{C}$ uses $\mathbf{atx}^*$ to replace $\mathbf{atx}_b$. $\mathcal{C}$ computes $\mathbf{atx}^* \leftarrow \mathsf{DAP.GenTx}(v^{**}, \mathtt{pri}^*_{\mathsf{DAP}}, \mathtt{pub})_{\mathsf{DAP}}$, where $v^{**} \notin \{v_0, v_1, v^*\} \wedge \mathtt{pri}^* \neq \mathtt{pri}_{\mathsf{DAP}_b}$. The challenge transaction in $\mathcal{G}_{\mathsf{sim}}$ is $\mathtt{wtx}_{\mathcal{G}_{sim}} = \{\mathtt{atx}^*, \mathbf{wcom}^*, \Pi^*\}$. Since the native transaction in DAP system satisfies Indistinguishability, the advantage of $\mathcal{A}_1$ in distinguishing $\mathtt{atx}^*$ from $\mathtt{atx}_b$ is negligible. Thus, $|\mathcal{E}_{\mathcal{G}_{sim}} - \mathcal{E}_{\mathcal{G}_2}| < \mathsf{negl}(\lambda)$.

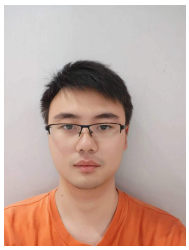
Since the advantage of $\mathcal{A}_i$ in winning $\mathcal{G}_{real}$ is $\mathcal{E}_{\mathcal{G}_{real}}$ is $|\mathcal{E}_{\mathcal{G}_{real}} - \mathcal{E}_{\mathcal{G}_{sim}}| < \mathsf{negl}(\lambda)$, and since each part of $\mathtt{wtx}_{\mathcal{G}_{sim}}$ is generated randomly, $\mathcal{E}_{\mathcal{G}_{sim}} < \mathsf{negl}(\lambda)$ and $\mathcal{E}_{\mathcal{G}_{real}} < \mathsf{negl}(\lambda)$. Therefore, the WDAP scheme satisfies Indistinguishability if the DAP scheme, the Consistency proof scheme and the Walsh commitment scheme is secure. □

ACKNOWLEDGMENTS

REFERENCES

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review*, p. 21260, 2008.

[2] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.
[3] P. Koshy, D. Koshy, and P. D. McDaniel, "An analysis of anonymity in bitcoin using P2P network traffic," in *Financial Cryptography and Data Security*, 2014, pp. 469–485.
[4] D. Vandervort, "Challenges and opportunities associated with a bitcoin-based transaction rating system," in *Financial Cryptography and Data Security*, 2014, pp. 33–42.
[5] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Financial Cryptography and Data Security*, 2013, pp. 6–24.
[6] S. Noether, A. Mackenzie *et al.*, "Ring confidential transactions," *Ledger*, vol. 1, pp. 1–18, 2016.
[7] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *2014 IEEE symposium on security and privacy*. IEEE, 2014, pp. 459–474.
[8] Y. Song, B. Chen, and X.-Y. Wang, "Cryptocurrency technology revolution: are bitcoin prices and terrorist attacks related?" *Financial Innovation*, vol. 9, no. 1, pp. 1–20, 2023.
[9] P. Chatzigiannis and F. Baldimtsi, "Miniledger: Compact-sized anonymous and auditable distributed payments," in *Computer Security - ESORICS*, 2021, pp. 407–429.
[10] C. Lin, D. He, X. Huang, M. K. Khan, and 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.
[11] Z. Wang, Q. Pei, X. Liui, L. Ma, H. Li, and S. Yu, "Daps: A decentralized anonymous payment scheme with supervision," in *Algorithms and Architectures for Parallel Processing: 19th International Conference, ICA3PP 2019, Melbourne, VIC, Australia, December 9–11, 2019, Proceedings, Part II 19*. Springer, 2020, pp. 537–550.
[12] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *2013 IEEE Symposium on Security and Privacy*. IEEE, 2013, pp. 397–411.
[13] E. Duffield and D. Diaz, "Dash: A privacycentric cryptocurrency," 2015.
[14] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "Coinshuffle: Practical decentralized coin mixing for bitcoin," in *Computer Security-ESORICS*. Springer, 2014, pp. 345–364.
[15] S. Ma, Y. Deng, D. He, J. Zhang, and X. Xie, "An efficient nizk scheme for privacy-preserving transactions over account-model blockchain," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 641–651, 2020.
[16] B. Bünz, S. Agrawal, M. Zamani, and D. Boneh, "Zether: Towards privacy in a smart contract world," in *Financial Cryptography and Data Security*. Springer, 2020, pp. 423–443.
[17] Z. Guan, Z. Wan, Y. Yang, Y. Zhou, and B. Huang, "Blockmaze: An efficient privacy-preserving account-model blockchain based on zk-snarks," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1446–1463, 2020.
[18] N. Narula, W. Vasquez, and M. Virza, "zkledger: Privacy-preserving auditing for distributed ledgers," in *USENIX*, 2018, pp. 65–80.
[19] C. Garman, M. Green, and I. Miers, "Accountable privacy for decentralized anonymous payments," in *Financial Cryptography and Data Security*, 2016, pp. 81–98.
[20] L. Xue, D. Liu, J. Ni, X. Lin, and X. S. Shen, "Enabling regulatory compliance and enforcement in decentralized anonymous payment," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 931–943, 2022.
[21] M. Bellare and A. Palacio, "The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols," in *Annual International Cryptology Conference*. Springer, 2004, pp. 273–289.
[22] M. Belenkiy, M. Chase, M. Kohlweiss, and A. Lysyanskaya, "Compact e-cash and simulatable vrfs revisited," in *Pairing-Based Cryptography*. Springer, 2009, pp. 114–131.
[23] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual International Cryptology Conference*. Springer, 1991, pp. 129–140.
[24] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *IEEE symposium on security and privacy*. IEEE, 2018, pp. 315–334.
[25] M. Abe and S. Fehr, "Perfect nizk with adaptive soundness," in *Theory of Cryptography Conference*. Springer, 2007, pp. 118–136.
[26] J. Groth, "On the size of pairing-based non-interactive arguments," in *Advances in Cryptology–EUROCRYPT*. Springer, 2016, pp. 305–326.
[27] K. Singhal, "Walsh codes, pn sequences and their role in cdma technology," *Term paper, EEL*, vol. 201, pp. 1–4, 2012.
[28] A. Gabizon, "On the efficiency of pairing-based proofs under the d-pke," *IACR Cryptol. ePrint Arch.*, p. 148, 2019.



Rui Gao Rui Gao is a PhD student currently studying at Cyberspace Security, Nanjing University of Posts and Telecommunications, Jiangsu, Nanjing 210003, China. E-mail: 501874794@qq.com. His research interests are zero-knowledge proofs and blockchain security.

Zhiguo Wan Zhiguo Wan is a principal investigator in the Zhejiang Lab, Hangzhou, Zhejiang, China. His main research interests include security and privacy for cloud computing, Internet-of-Things and blockchain. He received his B.S. degree in computer science from Tsinghua University in 2002, and Ph.D. degree in information security from National University of Singapore in 2007. He was a postdoc in Katholieke University of Leuven, Belgium and an assistant professor in the School of Software, Tsinghua University, Beijing, China.

Huaqun Wang received the BS degree in mathematics education from Shandong Normal University, in China, in 1997, the MS degree in applied mathematics from East China Normal University, in China, in 2000, and the PhD degree in cryptography from the Nanjing University of Posts and Telecommunications, in 2006. Now, he is a professor in Nanjing University of Posts and Telecommunications. His research interests include applied cryptography, blockchain, network security, and cloud computing security.