mpXim: A Decentralized Mixing Scheme based on a Multi-Party Discovering Protocol

Xuan Wang[1], Li Liao[2], and Dongyu Cao[3]
School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu, China
lliao@seu.edu.cn

2023 IEEE 23rd International Conference on Software Quality, Reliability, and Security Companion (QRS-C) | 979-8-3503-5939-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/QRS-C60940.2023.00097

Abstract—Bitcoin is not truly anonymous because of several de-anonymization technologies which attackers can use to reveal a relationship between a user's pseudonym and true identity. Coin-mixing is a well-known technology to address this problem by splitting the relationship between users' input and output addresses. However, some coin-mixing schemes have several limitations such as relying on a trusted third party, being vulnerable to Sybil attacks, or insufficient anonymity. Thus in this paper, we propose mpXim, a decentralized coin-mixing scheme based on the existing coin-mixing scheme Xim. We transform the two-party coin-mixing protocol in Xim into a multi-party coin-mixing protocol. Specifically, we change the role selection part and the corresponding contacting process. Moreover, we introduce a new economic incentive mechanism to ensure the sustainability of mpXim. From our theoretical analysis and experimental validation, it can be seen that our scheme has a better mixing success rate and resistance to Sybil attacks compared with Xim.

Keywords–Blockchain; privacy protection; coin-mixing

1. INTRODUCTION

Since Bitcoin is introduced by Satoshi Nakamoto [1] in 2008, blockchain technology, as its core underlying technology, has brought about a new wave of revolution in both the industry and academic [2]. Blockchain itself is a publicly distributed ledger maintained by multiple nodes. By employing methods such as data encryption, timestamps, consensus algorithms, and economic incentives, blockchain technology can maintain mutual trust between various institutions in a decentralized manner [3]. Consequently, it has attracted many researchers to explore various applications of blockchain, such as the Internet of Things [4], fog computing [5], vehicular ad-hoc networks [6], healthcare [7], and smart city [8] etc. Cryptocurrency is also one of the significant applications of blockchain. The interest in cryptocurrency, especially Bitcoin, can be proved in their continuously growing market value and significant trading volume [9]. Until July 1, 2023, the market value of Bitcoin has reached 589,867,517,980 USD, ranking first among all cryptocurrencies and accounting for 49.63 % of the total market value.

However, it is well known that Bitcoin faces privacy and security issues. In Bitcoin, users possess their accounts through a pair of public-private keys generated locally. Intuitively, the transaction address is the hash value of a random public key, making it difficult for attackers to directly associate the transaction address with the user's real identity. However, attackers can use transaction tracing techniques [10] to determine the originating nodes of transactions and further link them to the user's real IP address. They can also infer the real identity of nodes by conducting clustering analysis [11][12] to identify addresses with similar characteristics. Additionally, several de-anonymization techniques [13][14] exist that can determine the real identities of nodes by analyzing the transaction topology.

Researchers have proposed coin-mixing approaches for privacy protection to address these privacy and security issues. Coin-mixing is a method aimed at separating the relationship between users' input addresses and output addresses. Based on whether a trusted third-party server is introduced or not, coin-mixing services can be categorized into centralized mixing service and decentralized mixing service [15].

Centralized mixing service is typically provided by third-party mixing servers. The third-party servers receive all users' cryptocurrencies and then send a specified amount of cryptocurrencies to the users' output addresses. As a result, the transactions generated during these processes are difficult to distinguish from the perspective of external examination of blockchain transactions. The advantages of centralized mixing services lie in their high efficiency and convenience, but users often require to pay fees for such services. Moreover, the security of centralized mixing services relies entirely on the trustworthiness of the third-party server. If the third-party server is no longer trusted or is compromised by an attacker, users' privacy will be exposed.

The decentralized coin-mixing service proposed for solving the centralized mixing service problem involves a group of users with coin-mixing demands for coin-mixing. It is mainly divided into the following two types. The first type is a decentralized transaction pattern represented by CoinJoin [16], where multiple users generate a combined transaction containing the transaction addresses of all users. This approach has been applied to Bitcoin. However, this approach has the following issues: Firstly, the number of transaction addresses accommodated in the combined transaction is limited. Secondly, CoinJoin-like schemes usually lack effective mechanisms to resist Sybil attacks or DoS attacks. Moreover, such schemes often lack a complete discovering mechanism for pairing users. The second type of decentralized mixing scheme no longer requires users to generate a combined transaction. Instead, it enables users to transact with a randomly selected user. Representative schemes include Xim [17]. Since users are randomly selected, the resulting transaction will have unrelated transaction addresses. This type of scheme is not limited by the maximum number of transaction addresses. However, the introduction of new mechanisms may give rise to new security issues, such as the validity of security assumptions or an

2693-9371/23/$31.00 ©2023 IEEE
DOI 10.1109/QRS-C60940.2023.00097

excessive number of mixing rounds.

Based on the issues of the aforementioned mixing schemes, this paper aims to propose a decentralized mixing scheme mpXim based on a multi-party discovering protocol. The main contributions of this paper are as follows:

- This paper proposes a decentralized mixing approach that does not rely on a third-party server. The proposed scheme avoids the problem of third-party servers being vulnerable to single-point attacks and does not require additional fees.
- The scheme designs a multi-party discovering protocol that not only provides a way for pairing users randomly but also enhances the overall scheme's resistance to attacks. Unlike the existing Xim scheme based on two-party teaming protocols, this scheme no longer relies on random selection to determine the content of the protocol executed by mixing users. This scheme also introduces economic incentives to ensure the sustainability of the scheme.
- This paper quantifies the mixing success rate and the ability to resist Sybil attacks and DoS attacks through experiments. The experiments demonstrate a better mixing success rate and resistance against Sybil attacks under the same fraction of Sybils compared with Xim.

The remaining content of this paper is organized as follows: In Section 2, we will briefly introduce the related work. Section 3 will formulate assumptions on the attacker's behavior, analyze the Xim scheme, and provide the principles of mpXim. The details of mpXim will be presented in Section 4. Subsequently, in Section 5, we will present the experiments, results, and analysis. The conclusion and future work will be presented in Section 6.

2. RELATED WORK

Although there have been several studies proposing anonymous cryptocurrencies, such as ZeroCoin [18], ZeroCash [19], and Monero [20], their designs are incompatible with the existing Bitcoin structure. Moreover, any mechanism adjustments in Bitcoin require consensus from the entire Bitcoin community; otherwise, it may lead to a hard fork. This paper will focus on coin-mixing approaches that can be applied to Bitcoin or Bitcoin-similar cryptocurrencies. The following section presents concise explanations of several existing centralized and decentralized mixing methods.

In 2014, Bonneau [21] proposed a centralized coin-mixing scheme called MixCoin. MixCoin introduces an accountable centralized third-party server called Mixer, which separates users' transactions into two parts: transactions between users and Mixer, and transactions between Mixer and another random user to achieve anonymity. However, accountability cannot guarantee that Mixer will not steal Bitcoin. Additionally, Mixer can directly obtain the relationship between users' addresses, weakening the scheme's anonymity. Valenta [22] adjusted MixCoin's second problem in the BlindCoin scheme by introducing blind signature, making Mixer unable to directly obtain the relationship between users' input and output addresses, but the first problem still exists. In 2020, Lu [15] introduced CoinLayering, which uses multiple Mixers to separate holding and trading Bitcoin operations to establish transparent connections between user addresses. The scheme also introduces a higher-level centralized authority to supervise Mixers and provides economic compensation to users in case of theft.

Centralized coin-mixing schemes have good scalability and anonymity and can be applied to large-scale Bitcoin transaction scenarios. However, their security and anonymity are often limited by third-party servers.

The earliest decentralized coin-mixing scheme, CoinJoin, was proposed by Maxwell [16] in a Bitcoin forum. CoinJoin allows multiple users to provide their signatures to create a combined transaction, making it difficult for external users to trace the relationship between input and output addresses in the combined transaction [23]. Users will not suffer economic losses even if the coin-mixing process is incomplete, but for internal nodes the relationship between addresses is transparent, and the maximum number of addresses that can be accommodated in the combined transaction is limited.

In 2014, Ruffing et al. proposed CoinShuffle [24], which uses layered encryption to ensure that the relationship between users' input and output addresses is not visible to internal nodes and introduces a blaming mechanism to remove malicious nodes from the coin-mixing process [25]. CoinShuffle has a certain level of resistance to DoS attacks, but the low cost of node participation makes it vulnerable to Sybil attacks.

Bissias [17] proposed the Xim to resist the Sybil attacks, which randomly pairs nodes with coin-mixing demands through fee-based advertisements on the blockchain. Xim requires nodes to randomly select, with equal probability, to become either an advertiser or a respondent. Nodes that become advertisers need to publish advertisements on the blockchain and wait for respondents to contact them, while respondents periodically check the advertisements on the blockchain and contact the advertisers for coin-mixing. If one advertiser successfully pairs with a respondent, the two will use the fair exchange protocol [26] to complete a coin-mixing transaction. This is the coin-mixing process of Xim in a single round, and users can engage in multiple rounds of the same coin-mixing to achieve a higher mixing success rate. The blockchain-based advertising pairing mechanism provides Xim with a complete approach to pair nodes with coin-mixing demands, while simultaneously enhancing the overall scheme's resistance against attacks. However, the Xim requires multiple rounds to ensure an effective mixing success rate. Moreover, if the fraction of Sybil nodes reaches a certain level, the sustainability of the scheme is difficult to guarantee.

Xiao [27] designed a decentralized coin-mixing scheme based on multi-party signature, which, like CoinJoin, is also limited by the maximum number of addresses that can be accommodated in a transaction.

Decentralized coin-mixing schemes undoubtedly conform to the nature of Bitcoin. They no longer rely on third-party servers to provide services, avoiding the impact of single-point attacks and eliminating the need for additional fees.

In this paper, we will design a new decentralized coin-

mixing scheme based on the Xim. Our scheme's discovering protocol will no longer require users to randomly select roles, but instead allows them to directly choose to be advertisers or respondents based on their demands. In this case, our scheme will have a better mixing success rate and resistance to Sybil attacks compared to the Xim. Our scheme also introduces a new economic incentive mechanism to ensure the sustainability of the protocol.

3. PROBLEM FORMULATION

3.1 Adversary Model

In the Bitcoin environment, users can freely create a large number of nodes for transactions at little or no cost. So we consider the following two types of malicious behavior:

The first type is dishonest behavior, which aims to undermine the anonymity of the mixing scheme, that is, seeking to reveal the relationship between the input and output addresses of honest nodes. Dishonest behavior includes stealing the transaction privacy of honest nodes during the coin-mixing process. Dishonest attackers may engage in the mixing process normally or create numerous Sybil accounts to masquerade as honest nodes for their dishonest behavior.

The second type is disruptive behavior, which aims to disrupt the normal operation of the protocol. Disruptive nodes may be motivated by illegal profit to increase the cost for honest nodes to participate in coin-mixing, or simply obstruct the protocol's regular operation without any apparent reason, even at some economic cost. Disruptive behavior includes providing false information during the discovering process, pretending to execute some of the protocol while intentionally refusing to complete the remaining part, or refusing to participate in the coin-mixing process. Disruptive nodes can also create numerous Sybil fake accounts to masquerade as honest nodes.

3.2 Analysis on Xim

The drawback of the Xim lies in its relatively low single-round mixing success rate, which results in users being required to perform multiple rounds of coin-mixing. This is because of the significant difference between users' mixing success rate as advertisers and respondents. When users act as advertisers, attackers can increase the possibility of contacting users by creating multiple Sybil accounts, ultimately leading to the failure of the mixing round. On the other hand, the single-round mixing success rate for users acting as respondents depends on the proportion of malicious advertisements. Malicious nodes cannot prevent honest nodes from publishing advertisements, allowing respondents to achieve a reasonable single-round mixing success rate. Since Xim employs a random selection mechanism to determine the user's identity, there is a 50% chance of the user failing to mix in a round as an advertiser or successfully mixing as a respondent. This contributes to the overall low average single-round mixing success rate. Additionally, from the perspective of mixing success rate, it is more reasonable for users to participate as respondents. However, if all users act as respondents, the sustainability of the Xim scheme becomes challenging to ensure.

The one-to-one pairing relationship between advertisers and respondents in Xim exposes another issue. Suppose a node group contains 1/3 of malicious nodes participating as respondents in mixing, with the remaining honest nodes divided equally between advertisers and respondents, each accounting for 1/3 of the node group. In such a case, the malicious respondents can easily pair with one-third of honest advertisers, which effectively launches DoS attacks against the remaining honest respondents. Consequently, the Xim scheme would become unsustainable when facing this case.

In summary, the unreasonable pairing mechanism employed in the Xim scheme and significant differences in mixing success rates result in the problem of a low single-round mixing success rate and difficulty in ensuring sustainability.

3.3 Design Goals of mpXim

Based on the above analysis, mpXim needs to improve the pairing mechanism and address the issue of mixing success rate. Table 1 summarizes the differences between Xim and mpXim. As the mixing success rate is significantly higher when users act as respondents, and random selection may be difficult to enforce in practice, as even honest nodes might prefer to be respondents to achieve a higher mixing success rate, the mpXim aims to have all mixing users participate as respondents in the protocol rather than relying on random selection. This means that users have the freedom to choose to act as respondents to complete coin-mixing or choose to act as advertisers to obtain bitcoin, which is preferable to a random selection mechanism. Additionally, mpXim changes the pairing relationship between advertisers and respondents to be many-to-many. This way, even if multiple malicious nodes contact an advertiser, the advertiser can pair with honest nodes, greatly enhancing the sustainability of the scheme.

However, this improvement raises a new issue of determining who will act as advertisers. To address this, mpXim introduces a new economic incentive mechanism similar to Bitcoin's mining mechanism. Advertisers who complete the protocol will be rewarded with Bitcoin as compensation for providing coin-mixing services. In this context, both advertisers and respondents participate in the mpXim protocol based on their respective demands, ensuring the sustainability of mpXim.

Thus, the design goals of mpXim are as follows: 1) Change the pairing relationship to be many-to-many; 2) Provide an economic incentive mechanism to ensure that users have the motivation to act as advertisers; 3) Users with coin-mixing demands only act as respondents in the actual mixing process.

4. MPXIM

In this section, we will provide a detailed description of the mpXim.

Table 1. Differences between Xim and mpXim

Protocol	Xim	mpXim
Pairing Mechanism	Blockchain-Based Advertising	Blockchain-Based Advertising
Exchange Protocol	FairExchange [26]	FairExchange [26]
Multi-Rounds Mixing	Support	Support
Pairing Relationship	One-to-One	Multi-to-Multi
Role Selection	Randomly	Freely
Actual-Mixing Roles	Advertiser and Respondent	Respondents
Incentive Mechanism	None	Support

4.1 Overview

The objective of mpXim is to enable users with coin-mixing demands to perform transactions with other completely random users through the FairExchange [26] protocol. Based on Xim, mpXim allows users to perform multiple rounds of coin-mixing for their mixing units to achieve a better mixing success rate, and each round of the mixing process is consistent. The single-round method of mpXim is illustrated in Figure 1.

4.2 Single Round of Mixing

The single-round coin-mixing process of mpXim consists of two phases: the discovery phase and the actual mixing phase. The discovery phase aims to provide each node with a set of contact addresses to form a coin-mixing P2P network. The second phase is the actual mixing process, responsible for enabling pairwise transactions between nodes within this set. The **Protocol 1** is the algorithm diagram of this scheme. The δ represents the amount of the mixed Bitcoin.

Protocol 1: mpXim(δ)

1:Alice: $[\alpha_{R_{i=1..n,j=1..\sum_{i=1}^{n} k_i}}] \longleftarrow$ **FormMixingGroup**()
2:Bob: $[\alpha_{R_{i=1..n,j=1..\sum_{i=1}^{n} k_i}}] \longleftarrow$ **FormMixingGroup**()
3:Carol: $[\alpha_{R_{i=1..n,j=1..\sum_{i=1}^{n} k_i}}] \longleftarrow$ **FormMixingGroup**()
4:Alice,Bob,Carol: **GroupExchange**($[\alpha_{R_{i=1..n,j=1..\sum_{i=1}^{n} k_i}}], \delta$)

Here, the protocol **FormMixingGroup** represents the protocol for the first phase, and **GroupExchange** represents the protocol for the second phase.

4.2.1 Discovering Phase

The discovering phase is the primary improvement of mpXim compared to Xim. This phase enables nodes with coin-mixing demands to discover other nodes, while also enhancing the overall scheme's resistance against attacks. The discovering phase utilizes a multi-party teaming protocol,

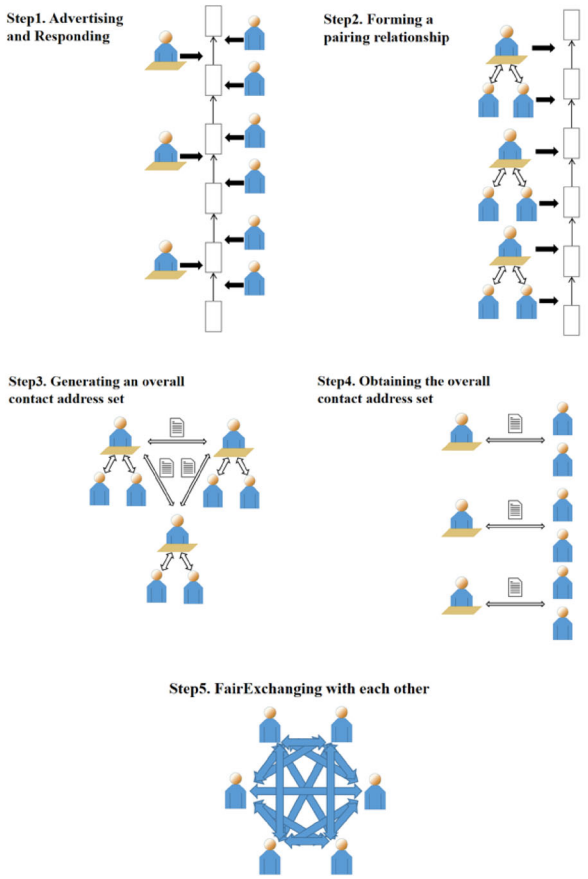

Figure 1. mpXim procedures

denoted as **FormMixingGroup**, as shown in **Protocol 2**. The algorithm inside the red box represents the distinct parts of mpXim compared with the Xim. In Xim, users randomly determine whether they act as advertisers or respondents during the mixing. However, in mpXim, users have the freedom to choose their roles based on their specific demands. Users who wish to obtain Bitcoin become advertisers, while those seeking to engage in coin-mixing become respondents. Only the respondents will participate in the actual mixing process, while the advertisers proceed to the next discovering phase after completing one.

Unlike Xim, the pairing relationship between advertisers and respondents in mpXim is many-to-many. In Step 3, advertisers need to specify the maximum number of respondents they can contact to provide coin-mixing services. In Steps 4 and 6, advertisers need to contact other advertisers and k_i respondents to enlarge the anonymous set for the second-phase mixing. To ensure that advertisers have indeed contacted k_i respondents, they must provide a list of hashed contact addresses of the respondents in Step 7. Each respondent can verify the correctness of the contact address list they received in Step 6. The actual confirmation of respondents' participation in the second-phase mixing occurs in Step 12 when they make the transfer of τ Bitcoin to the miners on the blockchain. Only

PROTOCOL 2: FormMixingGroup

1: Assume n roles of Advertiser $\mathcal{A}_{i=1..n}$ with address $A_{i=1..n}$ and location $\alpha_{A_{i=1..n}}$
2: Assume $\sum_{i=1}^{n} k_i$ roles of Respondent $\mathcal{R}_{i=1..n,j=1..k_i}$ with address $R_{i=1..n,j=1..k_i}$ and location $\alpha_{R_{i=1..n,j=1..k_i}}$
3: Advertiser $\mathcal{A}_i$: PUBLISHES $\mathbf{T}\{A_i \xrightarrow{0} A_i$,tip=$\tau$,TEXT (loc=$\alpha_{A_i}$,nonce=$N_{\alpha_i}$,pool=$\mathcal{P}$, num=$k_i$)$\}$
4: Advertiser $\mathcal{A}_i$: Selects other advertiseres $\mathcal{A}_{i'}$, STORES "$\mathbf{sig}_{A_i.sk}$ (N_{a_i},$N_{a_{i'}}$,α_{A_i})" to location $\alpha_{A_{i'}}$
5: Respondent $\mathcal{R}_{ij}$:Randomly selects advertiser $\mathcal{A}_i$, STORES "$\mathbf{enc}_{A_i.pk}(\mathbf{sig}_{R_{ij}.sk}(N_{\alpha_i}),R_{ij},\alpha_{R_{ij}})$" to location α_{A_i}
6: Advertiser $\mathcal{A}_i$: Selects k_i respondents. STORES $\mathbf{sig}_{A_i.sk}(N_{\alpha_i}$ paired to $[\alpha_{R_{i1}},...\alpha_{R_{ik_i}}])$ to location $\alpha_{R_{i1}},...\alpha_{R_{ik_i}}$
7: Advertiser $\mathcal{A}_i$: PUBLISHES $\mathbf{T}\{A_i \xrightarrow{0} A_i$, TEXT: (locklist=$[h(\alpha_{R_{i1}}), ...h(\alpha_{R_{ik_i}})]$ $\}$
8: **if** Advertiser $\mathcal{A}_i$'s transaction is not committed to blockchain by time t1 then
9: Respondent $\mathcal{R}_{ij}$: STORES "N_{α_i} aborted $\alpha_{R_{ij}}$; $\mathbf{sig}_{A_i.sk}(N_{\alpha_i}$ paired to $[\alpha_{R_{i1}},...\alpha_{R_{ik_i}}])$" to location α_{A_i}
10: **goto** line 5 (contact a new advertiser)
11: **end if**
12: Respondent $\mathcal{R}_{ij}$: PUBLISHES $\mathbf{T}\{R_{ij} \xrightarrow{0} R_{ij}$,tip=$\tau\}$
13: **if** Respondent $\mathcal{R}_{ij}$'s transaction is not committed to blockchain by time t2 then
14: Advertiser $\mathcal{A}_i$: STORES "$\mathbf{sig}_{A_i.sk}(N_{\alpha_i}$ unpaired from $\alpha_{R_{ij}})$" to location α_{A_i}
15: **goto** line 5 (wait for new respondents)
16: **end if**
17: Advertiser $\mathcal{A}_{i'}$: STORES $\mathbf{sig}_{A_{i'}.sk}([\alpha_{R_{i'1}},...\alpha_{R_{i'k_{i'}}}])$ to location α_{A_i}
18: Advertiser $\mathcal{A}_i$: STORES $\mathbf{sig}_{A_i.sk}[\alpha_{R_{i'1}},...\alpha_{R_{i'k_{i'}}}]$ to location $\alpha_{R_1},...\alpha_{R_{k_i}}$
19: Respondent $\mathcal{R}_{ij}$: PUBLISHES $\mathbf{T}\{R_{ij} \xrightarrow{\tau/k_i+f+\sum_{j=1}^{j\neq i} k_j/k_i*f} A_i\}$
20: **if** Respondent $\mathcal{R}_{ij}$'s transaction is not committed to blockchain by time t3 then
21: Advertiser $\mathcal{A}_i$: STORES "R_{ij} aborted N_{α_i}; $\mathbf{sig}_{A_i.sk}(N_{\alpha_i}$ unpaired from $\alpha_{R_{i1}})$" to location α_{A_i}
22: **goto** line 5 (wait for new respondents)
23: **end if**
24: **return** Addresses of the mixing group, $[\alpha_{R_{i=1..n,j=1..\sum_{i=1}^{n} k_i}}]$

then will they be considered participants in the current mixing round.

Finally, advertisers contact each other to combine their lists of contacted respondents and send them to their respective respondents. The steps inside the blue box are used for failure recovery. If an advertiser fails to complete Step 7 on the blockchain within a certain time frame, the respondent can provide the pairing record from Step 6 to terminate the relationship with the advertiser. In this case, neither party incurs any losses. If a respondent refuses to tip Bitcoin to the miners in Step 12, they will be excluded from the current mixing round. In case the respondent refuses to provide the profit to the advertiser in Step 19, the advertiser has sufficient evidence to prove the respondent's violation of the protocol, as the respondent must have previously published the transaction on the blockchain in Step 12. In this scenario, the advertiser can still reuse their previous advertisement without losing any Bitcoin.

4.2.2 Mixing Phase

This paper does not focus on the specific implementation details of the second stage. This section will only provide an overview of the general ideas for the second stage. After the completion of the first phase, all respondents will form a P2P communication network. Coin-mixing nodes can randomly pair up and perform transactions using Barber's fair exchange protocol [26]. In the case where an odd number of nodes enter the second phase, the remaining odd-numbered nodes can halve the amount of the mixing unit and perform pairwise mixing among the remaining nodes.

4.3 Cost and Incentives

This section will discuss the setting of the fees τ and f involved in mpXim.

The fee τ is to assign a fixed cost to the participating nodes in the protocol, thereby increasing the cost of malicious behavior. The value of τ can be determined solely by the mixing pool P for a single mixing round. Advertisers are required to prepay τ Bitcoin to successfully complete the discovering phase, while respondents need to pay (1+1/k_i)*τ Bitcoin. If an advertiser sets an unreasonably high prepayment cost, potential respondents may opt to pair with other advertisers offering lower τ values. As a result, the advertiser reduces the probability of being maliciously attacked, but their rewards

may also decrease, and vice versa. Thus, both advertisers and respondents have the same motivation to set a reasonable τ value to achieve a balance between the security of the protocol and the cost borne by both parties.

In mpXim, the fee f serves as an incentive for advertisers and is set in relation to the transaction fee, such as 1% of the transaction fee, which is determined by the Bitcoin network. Setting f to a value related to the transaction fee is justified as our scheme's economic incentive mechanism is inspired by Bitcoin's mining incentive mechanism. Additionally, using a value set by the Bitcoin network has a certain degree of consensus, thereby avoiding obstacles in the protocol's operation caused by unreasonable f values set by advertisers and respondents.

The cost borne by respondents in this scheme is inversely related to the value of k_i. If advertisers can contact more respondents, resulting in higher rewards for themselves, the average cost borne by respondents will also be lower. Furthermore, if advertisers can contact other advertisers expanding the coin-mixing set, their final rewards will also increase. The design of the economic incentive aims to encourage advertisers to contact as many respondents as possible to increase the size of the mixing pool in the second phase, thereby enhancing the scheme's sustainability and resistance to attacks.

5. EXPERIMENT AND ANALYSIS

In this section, we will experimentally verify the mixing effectiveness and the resistance to attacks of mpXim. Since this scheme is an improvement over the Xim, experimental data from the Xim will also be provided for comparison. The experimental data for the Xim is reproduced based on the proofs and formulas presented in [17].

Although the second phase does not provide specific implementation details, it still provides a transaction framework. The experimental section primarily focuses on obtaining the results of the transactions, without fully simulating the transaction process, such as considering the FairExchange process between nodes or the specific attack methods employed, which means that pairing with malicious nodes is equivalent to transaction failure.

To ensure a fair comparison of experimental results, we will use consistent experimental parameters and metrics as in the Xim. This section will simulate the following three experiments using Python 3.9, with the random functions from the random library. Each experiment will assume 1000 participating nodes and will be simulated 1000 times to obtain average outcomes.

5.1 Mixing Effectiveness

Suppose a participant, Alice, needs to mix $m\delta$ Bitcoin, and the quantity of each mixing unit is δ. Since malicious nodes can control a certain proportion of nodes in a mixing round by creating multiple Sybil accounts, Alice's coin-mixing may require multiple rounds to succeed. Similar to Xim, assuming Alice requires multiple rounds to achieve successful mixing, this experiment aims to examine the probability of successfully

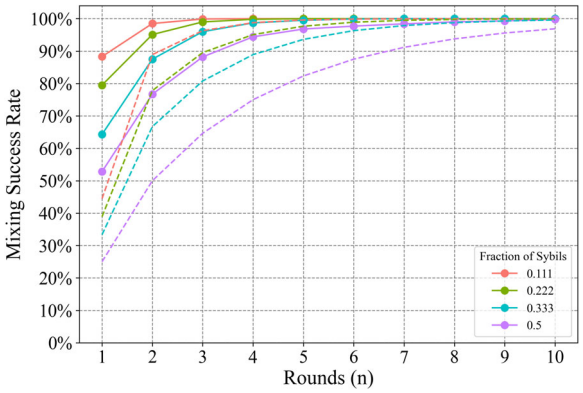

Figure 2. The mixing success rate over n rounds

mixing one unit of Alice's Bitcoin by a certain round, while varying the proportion of Sybil accounts created by malicious nodes [17]. In this case, Alice's one unit of Bitcoin is successfully mixed with another completely random honest user's one unit of Bitcoin, while the relationship between the input and output addresses remains unknown to the malicious nodes.

The experimental steps are as follows:

1) Set the proportion of Sybil accounts as x, and mark $1000*x$ as malicious accounts.

2) For each experiment, randomly select an honest node (not a malicious account) and randomly pair two nodes for a transaction until the honest node has been paired. If the honest node is paired with a malicious node by the end of a round, it is considered a coin-mixing failure; otherwise, it is considered a successful mixing round.

3) Record the number of successful mixing rounds for each honest node, which can deduce the mixing success rate.

The experimental results are shown in Figure 2. The solid lines represent the results of the mpXim, while the dashed lines represent the results of the Xim.

The figure illustrates the variation of the mixing success rate with the number of rounds for different fractions of Sybil accounts (0.111, 0.222, 0.333, and 0.5). When the fraction of Sybil accounts is the same, the mixing success rate of mpXim is significantly higher than that of the Xim. This is because, in our scheme, nodes only participate as respondents in the protocol, avoiding the lower mixing success rate associated with being an advertiser in the Xim. Specifically, when the fraction of Sybil accounts is 0.111, 0.222, and 0.333, our scheme achieves a 100% mixing success rate in the third, fourth, and fifth rounds, respectively. In contrast, the Xim requires 7, 8, and 10 rounds to achieve the same level of mixing success rate. When the fraction of Sybil accounts is 0.5, although neither scheme achieves a 100% mixing success rate within 10 rounds, our scheme outperforms the Xim in the fifth round compared to the tenth round of the Xim. Therefore, mpXim can reduce the required mixing rounds by half compared to the Xim for the same desired mixing success rate. Overall, our proposed scheme demonstrates a superior

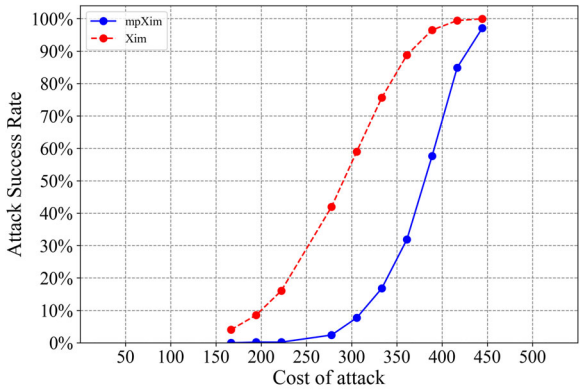

Figure 3. The success rate and cost of the Sybil attacks

mixing success rate compared to the Xim.

5.2 Resistance of Attacks

5.2.1 Sybil-Attack Resistance

In the Bitcoin environment, if an attacker invests unlimited costs to create as many Sybil accounts as possible, they can always guarantee successful attacks on coin-mixing schemes. Therefore, the Sybil attack resistance experiment in the Xim evaluates the scheme's resistance against Sybil attacks by measuring the cost incurred by the attacker to establish a successful relationship between the user's input and output addresses [17]. The same metric is adopted in mpXim for comparison. To maintain consistency with the Xim experiment parameters, the experiment assumes that a participant, Alice, needs to mix 10 units of her Bitcoin and perform 10 rounds of mixing for each unit, costing $1 each round. If the attacker successfully attacks at least one of Alice's mixing units, it is considered a successful attack.

The experimental steps are as follows:

- Each mixing unit requires 10 rounds of coin-mixing following the steps outlined in Experiment 1. The success or failure of each mixing unit for each participant is recorded.
- Count the number of participants whose at least one mixing unit failed among the 10 mixing units, which can be used to calculate the rate of successful attacks by malicious nodes.
- The cost incurred by the attacker is calculated as the total number of participants multiplied by the cost of each attack ($1) and the fraction of Sybil accounts.

The experimental results are presented in Figure 3. The blue solid lines represent the mpXim, while the red dashed lines represent the Xim.

The proportion of Sybil accounts in the experiment ranges from 1/3 to 8/9, with the horizontal axis representing the cost incurred by the attacker and the vertical axis representing the attacker's attack success rate. It is evident that the attacker's success rate increases with the proportion of Sybil accounts, and correspondingly, the cost incurred by the attacker also increases.

It can be observed that when the attacker aims for an 80% attack success rate, Xim requires a cost of approximately $340, consistent with the original data from the Xim [17]. In contrast, mpXim requires the attacker to spend around $420, which is an increase of approximately 27% in cost. From the perspective of the attack success rate, if the attacker only has a budget of $340 for the attack, their success rate would be around 26%, significantly lower than the 80% success rate achieved in the Xim. Therefore, mpXim demonstrates superior resistance against Sybil attacks compared to the Xim.

5.2.2 Dos-Attack Resistance

The goal of this experiment is to investigate the impact of a denial-of-service (DoS) attack launched by an attacker on the honest nodes in mpXim. The impact is reflected in the additional proportion of the cost incurred by honest nodes due to DoS attacks in [17]. In the Xim, if the attacker controls a proportion of x Sybil accounts, the cost for honest nodes increases to 1 + x. The same experimental metrics will be used to evaluate the DoS resistance capability of mpXim.

By analyzing mpXim, the following cases may be vulnerable to DoS attacks: The attacker, acting as an advertiser, launches a DoS attack, it may occur in Step 4, Step 6, Step 7, or Step 18, where the attacker maliciously or refuses to execute the protocol.

- If the attacker chooses not to contact other advertisers in Step 4, the resulting total set of contacted nodes would be equivalent to its own set of contacted nodes. In this case, the scheme can still be completed normally.
- If the attacker chooses to launch a DoS attack in Step 6 or Step 7, the attacker must provide additional legitimate accounts that can be used for transactions in Step 12. Otherwise, the honest nodes will not proceed with Step 12 and will not incur additional losses. Therefore, if the attacker chooses to launch a DoS attack at this step, they can create some Sybil accounts to impersonate honest nodes and provide the TEXT information in Step 6 and Step 7. Due to the latency of Bitcoin transactions (requiring 6 confirmations for successful transactions), it is possible that the honest nodes have already performed the tipping operation in Step 12, while the fake nodes did not actually transfer funds to the miners. Even if the honest nodes discover that the contact addresses provided by the advertisers are fake, the transaction has been confirmed, and the honest nodes will suffer the loss of τ. In this case, the DoS attack is successfully launched.
- If the attacker chooses to launch a DoS attack in Step 18, the impact differs depending on whether the mixing set obtained by the honest nodes in Step 6 is valid or fake. If the mixing set is valid, the respondents may choose to skip the incentive step (Step 19) because of the disruptive behavior and proceed directly to the second phase of mixing. In this case, the DoS attack does not increase the cost for the respondents. However, if the mixing set is fake, the case is equivalent to launching a DoS attack in Step 6 and Step 7.

To summarize, if the malicious node chooses to attack as an advertiser, they can only create Sybil accounts to deceive the respondents in Step 6 and Step 7, causing honest respondents to incur costs in Step 12. In this case, the additional cost incurred by honest respondents is τ.

Next, we will discuss the cases where the attacker chooses to launch a DoS attack as a respondent. There are three cases in which attackers can launch Dos attacks as respondents.

- Similar to the previous discussion, if no colluding malicious nodes act as advertisers, the malicious respondents' behavior of refusing to select an advertiser in Step 5 or refusing to tip Bitcoin to the miners in Step 12 will result in their removal from contact set by honest advertisers, thus not affecting the cost of honest respondents. In the case of collusion, launching a DoS attack in Step 12 is equivalent to the case in Step 6 and Step 7 as a malicious advertiser.
- Furthermore, malicious respondents may refuse to provide a reasonable amount of Bitcoin as an incentive in Step 19. In this case, since these malicious respondents fail to publish the expected transactions on the blockchain, honest respondents will refuse to proceed to the second phase with these malicious respondents. Therefore, the malicious respondents cannot affect the cost incurred by honest respondents.
- The final scenario is when malicious respondents pay the incentive in Step 19, proceed to the second phase, and then refuse to perform pairwise mixing with honest nodes. As a result, some honest nodes will need to repeat the protocol for another round because they cannot mix with the other honest respondents. In this case, the cost incurred by honest nodes is $(1 + 1/k_i)\tau + f + \sum_{j=1}^{j\neq i} k_j/k_i * f$ Bitcoin.

In summary, there are two cases in which malicious nodes launch DoS attacks. One is to add a large number of Sybil accounts in Step 6 and Step 7, pretending to be honest nodes about to incur costs in Step 12, deceiving honest respondents into tipping fees to miners. The other case is to refuse to proceed with the protocol in the second phase, causing some honest nodes to be unable to mix.

The experimental steps of DoS attacks in Step 6 and Step 7 are as follows:

1) Set the proportion of Sybil accounts as x, and mark 1000*x accounts as malicious accounts. All malicious accounts participate as advertisers in the experiment.

2) Simulate the first phase of the scheme, and if an honest respondent is paired with a malicious advertiser, it is considered subject to an attack in Step 6 and Step 7. The honest node needs to repeat the scheme for another round to pair with an honest advertiser.

3) Calculate the total cost incurred by honest nodes and calculate the percentage of the additional cost incurred by honest nodes.

The steps of simulation of DoS attacks in the second phase of the protocol are as follows:

1) Set the proportion of Sybil accounts as x, and mark 1000*x accounts as malicious accounts. All malicious accounts participate as respondents in the experiment.

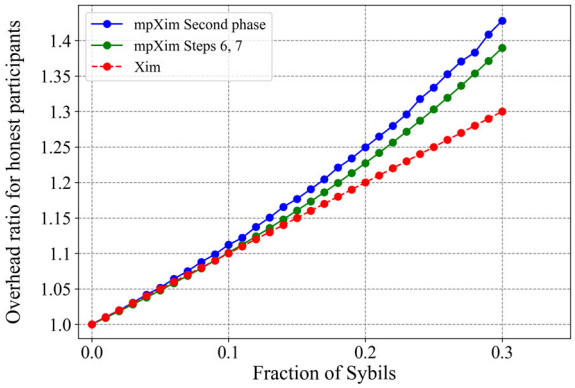

Figure 4. The additional overhead rate given the fraction of Sybils

2) Simulate the mixing phase of the scheme, and if an honest node engages in mixing with a malicious node, it is considered subject to a DoS attack in the second phase of the protocol. The honest node needs to repeat the scheme for another round to complete the mixing.

3) Calculate the total cost incurred by honest nodes and calculate the percentage of the additional cost.

The obtained conclusion is shown in Figure 4:

The x-axis in the figure represents the fraction of Sybil accounts controlled by malicious nodes, ranging from 1% to 30%. The y-axis represents the additional cost ratio incurred by honest nodes. The blue, green, and red lines correspond to the outcomes when Dos attacks are launched in the second phase, in Steps 6-7, and in the Xim, respectively.

It is evident that as the fraction of Sybil accounts increases, the additional cost incurred by honest nodes also increases. As x increases from 1% to 30%, the blue line remains consistently above the green line, and the green line is above the red line. This indicates that in mpXim if attackers choose to launch a Dos attack in the second phase, they can impose more cost on honest nodes compared to the attack in Steps 6-7. When x reaches 30%, the additional costs imposed by attackers in the second phase, Steps 6-7, and the Xim are approximately 43%, 39%, and 30%, respectively. This experimental outcome demonstrates that 1) In mpXim, attackers can prioritize launching DoS attacks in the second phase to achieve better attack effectiveness, and 2) mpXim has weaker resistance to Dos attacks than the Xim when the fraction of Sybil accounts is less than 1/3. However, there are two noteworthy points to consider:

- Taking into account the results from Experiment 1, mpXim exhibits a higher mixing success rate compared with the Xim. Users can achieve the same mixing success rate as the Xim with only half the number of rounds. When the fraction of Sybil accounts is 30%, Xim users require nearly 10 rounds to achieve a 100% mixing success rate, they actually need to pay the cost of 13 rounds, considering the possibility of dos attacks. On the other hand, users of mpXim only need less than 5 rounds, and even with dos attacks, they only

require approximately 8 rounds (5 * (1+43%)) to achieve a 100% mixing success rate. Considering the number of rounds needed to ensure the mixing success rate in practice, mpXim outperforms the Xim.

- The Xim is limited by the requirement that the fraction of Sybil accounts must be less than 1/3. This is because the Xim utilizes one-to-one pairing, and if more than 1/3 of attackers choose to act as respondents to launch dos attacks, the Xim would no longer be able to provide mixing services. However, in this scheme, even if more than 1/3 of attackers choose to act as respondents for Dos attack, the protocol's sustainability is still guaranteed due to the multi-party pairing mechanism employed, where an advertiser can contact multiple respondents.

Table 2 displays a comparison of experimental results between mpXim and Xim. To conclude, although mpXim has slightly weaker resistance to dos attacks compared with the Xim, overall, it performs better than the Xim.

Table 2. Experimental comparison between Xim and mpXim

Protocol	Xim	mpXim
Coin-Mixing Effectiveness	Low (44.5%) First Round Sybils = 11%	High (88%) First Round Sybils = 11%
Sybil-Attack Resistance	Low (340$) AttackRate:80%	High (420$) AttackRate:80%
Dos-Attack Resistance	High (30%) Sybils = 30%	Medium (43%) Sybils = 30%
Sustainability	Requiring Fraction of Sybils $<$ 1/3	No Limit

6. Conclusion

This paper proposes a new decentralized mixing scheme mpXim based on the existing Xim. In mpXim, the two-party pairing structure of Xim is modified to a multi-party pairing structure, avoiding randomly determining the roles in protocol execution, and a new economic incentive mechanism is introduced. Compared with the original scheme, this scheme improves the mixing success rate and resistance to Sybil attacks. The experiment results demonstrate that, under the same mixing success rate, users can reduce the number of mixing rounds by half. Under the same Sybil attack success rate, mpXim increases the attacker's cost.

However, this scheme does not provide specific implementation details for the second phase, which is a limitation in analyzing the actual mixing process. Future work may consider: 1) Introducing new mechanisms to further enhance the overall scheme's advantages, and 2) Complementing the implementation details and conducting more attack analysis and experiments for this scheme.

Acknowledgment

This work was supported by the Key Research and Development Program of Jiangsu Province (under no.BE2021002-3), and the National Key Research and Development Program of China (under no.2019YFE0105500). This paper has benefited greatly from the valuable guidance of Professor Bixin Li. I would like to express my sincere gratitude for his invaluable support and mentorship throughout this research.

References

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

[2] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–34, 2019.

[3] L. Peng, W. Feng, Z. Yan, Y. Li, X. Zhou, and S. Shimizu, "Privacy preservation in permissionless blockchain: A survey," *Digital Communications and Networks*, vol. 7, no. 3, pp. 295–307, 2021.

[4] B. Deebak, F. H. Memon, K. Dev, S. A. Khowaja, W. Wang, and N. M. F. Qureshi, "Tab-sapp: A trust-aware blockchain-based seamless authentication for massive iot-enabled industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 243–250, 2022.

[5] Y. I. Alzoubi, A. Al-Ahmad, and H. Kahtan, "Blockchain technology as a fog computing security and privacy solution: An overview," *Computer Communications*, vol. 182, pp. 129–152, 2022.

[6] S. K. Dwivedi, R. Amin, A. K. Das, M. T. Leung, K.-K. R. Choo, and S. Vollala, "Blockchain-based vehicular ad-hoc networks: A comprehensive survey," *Ad Hoc Networks*, p. 102980, 2022.

[7] T. Hovorushchenko, A. Moskalenko, and V. Osyadlyi, "Methods of medical data management based on blockchain technologies," *Journal of Reliable Intelligent Environments*, vol. 9, no. 1, pp. 5–16, 2023.

[8] C. Huang, L. Xue, D. Liu, X. Shen, W. Zhuang, R. Sun, and B. Ying, "Blockchain-assisted transparent cross-domain authorization and authentication for smart city," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 17 194–17 209, 2022.

[9] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers 17*. Springer, 2013, pp. 6–24.

[10] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using p2p network traffic," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18*. Springer, 2014, pp. 469–485.

[11] F. Reid and M. Harrigan, *An analysis of anonymity in the bitcoin system*. Springer, 2013.

[12] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun, "Evaluating user privacy in bitcoin," in *Financial Cryptography and Data Security: 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers 17*. Springer, 2013, pp. 34–51.

[13] M. OSMANOĞLU and A. A. Selcuk, "Privacy in blockchain systems," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 2, pp. 344–360, 2022.

[14] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis," *arXiv preprint arXiv:1502.01657*, 2015.

[15] N. Lu, Y. Chang, W. Shi, and K.-K. R. Choo, "Coinlayering: an efficient coin mixing scheme for large scale bitcoin transactions," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 3, pp. 1974–1987, 2020.

[16] G. Maxwell, "Coinjoin: Bitcoin privacy for the real world," in *Post on Bitcoin forum*, vol. 3, 2013, p. 110.

[17] G. Bissias, A. P. Ozisik, B. N. Levine, and M. Liberatore, "Sybil-resistant mixing for bitcoin," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, 2014, pp. 149–158.

[18] 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.

[19] 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.

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

[21] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten, "Mixcoin: Anonymity for bitcoin with accountable mixes," in *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18*. Springer, 2014, pp. 486–504.

[22] L. Valenta and B. Rowan, "Blindcoin: Blinded, accountable mixes for bitcoin," in *Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, January 30, 2015, Revised Selected Papers*. Springer, 2015, pp. 112–126.

[23] S. Marciante and Á. Herrero, "The evolution of privacy in the blockchain: A historical survey," in *Computational Intelligence in Security for Information Systems Conference*. Springer, 2019, pp. 23–34.

[24] T. Ruffing, P. Moreno-Sanchez, and A. Kate, "Coinshuffle: Practical decentralized coin mixing for bitcoin," in *Computer Security-ESORICS 2014: 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7-11, 2014. Proceedings, Part II 19*. Springer, 2014, pp. 345–364.

[25] Y. Cui, B. Pan, and Y. Sun, "A survey of privacy-preserving techniques for blockchain," in *Artificial Intelligence and Security: 5th International Conference, ICAIS, New York, NY, USA, July 26–28, Proceedings, Part IV 5*. Springer, 2019, pp. 225–234.

[26] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to better—how to make bitcoin a better currency," in *Financial Cryptography and Data Security: 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers 16*. Springer, 2012, pp. 399–414.

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