Towards Understanding and Demystifying Bitcoin Mixing Services

A Preprint

Lei Wu[1], Yufeng Hu[1], Yajin Zhou*[1], Haoyu Wang[2], Xiapu Luo[3], Zhi Wang[4], Fan Zhang[1], and Kui Ren[1]

[1]*Zhejiang University*
[2]*Beijing University of Posts and Telecommunications*
[3]*The Hong Kong Polytechnic University*
[4]*Florida State University*

November 2, 2020

Abstract

The popularity of Bitcoin benefits a lot from its anonymity. However, the anonymity of Bitcoin is pseudonymity, or *relationship anonymity* between addresses. Researchers have proposed several heuristics to break it by clustering addresses. Meanwhile, new approaches are invented to provide enhanced anonymity. As one of the most promising approaches, many third-party services named *mixing services* have emerged and been widely used in recent years. Unfortunately, they are already abused to facilitate money laundering for criminal activities. Despite that there is an urgent need to understand Bitcoin mixing services, however, few studies have been proposed to systematically demystify them.

In this paper, we take the first step to study state-of-the-art Bitcoin mixing services. Specifically, we propose a generic abstraction model for mixing services. According to our investigation, most mixing services share a same three-phase procedure but differ in mixing mechanisms. There are two mechanisms used in the wild, i.e. *swapping* and *obfuscating*. According to this model, we conducted a graph-based analysis and successfully revealed the mixing mechanisms and workflows of four representative mixing services. Besides, we propose a method to further demystify mixing services that apply obfuscating mechanism by identifying mixing transactions. The proposed approach is capable of identifying most (over 92%) of the mixing transactions. Based on identified transactions, we then estimated the profit of mixing services and provided a case study of tracing the money flow of stolen Bitcoins.

1 Introduction

Bitcoin [39] has become the most representative cryptocurrency with the largest market share since its birth in 2009. As of the first quarter in 2020, the total market capitalization for Bitcoin is over 117 billion dollars [21], occupying more than 65% of the overall volume of cryptocurrency market. In contrast to traditional payment channels (e.g., paper and card), the decentralization essence of Bitcoin has three inborn characteristics: 1) money can be transferred online directly from payers to payees without the intervention of any third-party banking services; and 2) transactions are verifiable and cannot be reversed; and 3) the *pseudonymity* makes linking addresses to real-world entities very hard. The *anonymity* provided by Bitcoin is regarded as a key factor leading to its popularity [12].

However, the anonymity offered by Bitcoin is classified as *relationship anonymity* [45], and can be broken due to the following features of Bitcoin. First, the complete transaction history is publicly available, namely, the money flow between Bitcoin addresses can be fully revealed. Second, the mechanism relies on the pseudonymity of addresses used in transactions, which can be broken by aggregating addresses into clusters (or user identities) with simple

*Corresponding author (yajin_zhou@zju.edu.cn).

heuristics [26] or publicly available data sources [35]. Once address clusters are identified, the complete money flows between clusters (corresponding to different users) can be immediately revealed. As a result, the anonymity is no longer preserved.

To improve the anonymity of Bitcoin, several solutions have been proposed in recent years. Some of them aim to hide the transaction information by modifying the Bitcoin protocol (or building additional infrastructures), to provide additional anonymity. Such solutions include *Zerocash* [50] and *Monero* [41]). Others try to set up third-party services to provide enhanced anonymity without modifying the Bitcoin protocol, e.g., *Mixcoin* [13] and *Blindcoin* [57]. Corresponding to these approaches, many *altcoins* and *mixing services* (or *tumblers*) emerged to provide additional anonymity. Although altcoins can achieve stronger anonymity properties [41], the migration cost from Bitcoin to altcoins hinders the popularity of altcoins and makes mixing service a good alternative choice.

Unfortunately, anonymity is a double-edged sword. Apart from the benign applications, Bitcoin has been abused as a primary cryptocurrency for criminal activities [29], including ransomware like *WannaCry* [6], notorious underground markets like *Silk Road* [17] and *Ponzi* schemes [4]. Specifically, mixing services are extremely widely used in those activities to facilitate money laundering. For example, a previous study [17] shows that Silk Road used mixing services extensively. It has also been reported [56] that the attacker laundered $7,170$ Bitcoin through *Bitcoin Fog* (one of the earliest and most famous mixing services), after attacking *Bter.com* (a former Chinese cryptocurrency exchange). In addition, on May 8, 2019, cryptocurrency exchange giant *Binance* reported that it has suffered from a large scale security breach, resulting in loss of around $7,074$ BTC (about 40 million dollars at that time) [5]. Further investigation indicated that a large portion of stolen Bitcoins were sent to *Chipmixer* [18], a popular mixing service provider.

The extensive use of mixing services makes it extremely difficult for regulators and criminal fighters to trace suspicious money flow. What's worse, the presence of mixing services may lead to erroneous money flow trace results as they deliberately obfuscate the relationship between senders and recipients. Although there is an urgent need to demystify the mixing services, only a few previous works have been published. In one of the earliest research on mixing services [38], the authors performed a simple graph analysis based on data collected from experiments of selected mixing services, while others focused on security issues of mixing services themselves [22]. In short, there is a lack of a comprehensive understanding of Bitcoin mixing services.

Our approach. In this paper, we take the first step to systematically study Bitcoin mixing services. Our goal is to understand mixing services in a comprehensive way, and demystify them based on insights we harvested.

To facilitate our analysis, we first propose a three-phase model, to depict the workflow of mixing services. Our study suggests that most mixing services share a same procedure but differs when applying the *mixing mechanisms*. Based on this abstraction, we categorized state-of-the-art mixing mechanisms into two types, namely, *swapping* and *obfuscation*.

We then conduct an empirical study to analyze mixing services based on real world transactions. To this end, four representative mixing services are selected and analysed. Then we collect sample transactions for each service to analyse the mixing mechanisms used by these services, and their mixing workflows. Finally, we propose a heuristic-based algorithm to further analyse the mixing services with obfuscation mechanism.

Results. We apply the approach to analyze 4 representative Bitcoin mixing services that are currently popular and active, i.e., *Chipmixer* [16], *Wasabi Wallet* [62], *ShapeShift* [51], and *Bitmix.biz* [11].

For Chipmixer and Bitmix.biz, we interact with these service by sending Bitcoins to the services to collect sample transactions (inputs to the service and outputs from the service). We conduct 10 experiments with 4 inputs to Chipmixer and 6 inputs to Bitmix.biz. In total, we collected 8 and 14 outputs from them, respectively. For ShapeShift and Wasabi Wallet, we are able to reconstruct mixing records using provided public APIs. Accordingly, we harvested $4,850$ mixing transactions from Wasabi Wallet, and $27,411$ cryptocurrency convert records from ShapeShift.

Based on these sample transactions, we conduct a transaction-based analysis to first determine the mixing mechanism used, and then reveal their workflow. Then, we perform an advanced analysis for services using the obfuscation mechanism to further identify mixing transactions. The evaluation result demonstrates that the proposed algorithm is capable of identifying most (over **92%**) of the mixing transactions. We further estimate the profit of these services, and use a real attack to demonstrate the capability of our approach to trace the stolen Bitcoins that have been mixed.

Contributions. We make the following contributions.

- We proposed an abstraction model and approach to systematically demystify state-of-the-art Bitcoin mixing services, including the mixing mechanisms and workflows.
- We applied the proposed approach to four representative Bitcoin mixing services, and successfully revealed the mixing mechanisms and workflows of these services.

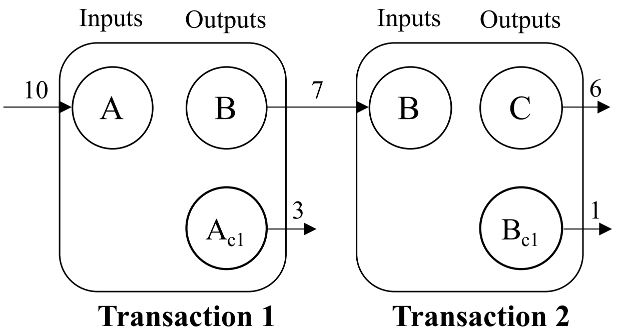


Figure 1: An example of Bitcoin transactions and UTXOs.

- We proposed an advanced analysis to effectively reveal mixing services that employ the obfuscation mechanism by identifying *most* (over 92%) mixing transactions. The evaluation results demonstrated the effectiveness of the proposed approach.

2 Background

2.1 Bitcoin

Bitcoin is a decentralized cryptocurrency proposed by an identity with pseudonym Satoshi Nakamoto [39]. The idea behind Bitcoin is a publicly available and verifiable distributed ledger recording the money flow between addresses. To prove the integrity of this public ledger, Bitcoin uses a Peer-to-Peer (P2P) network and employs the Proof-of-Work (PoW) consensus algorithm, which introduces the concept of "mining", a mechanism to issue Bitcoin.

Transactions. Transaction is a basic unit describing money flow from input to output addresses. Every input in a transaction is a reference to an *unspent transaction output* (UTXO) [1], which is an output in a previous transaction that has not been referenced in other transactions.

Figure 1 gives a concrete example to illustrate the use of UTXOs. Alice has 10 BTC in address A (as a UTXO) and wants to send 7 BTC to address B belonging to user Bob. To achieve this, Alice initiates a transaction (Transaction 1) by referring to this UTXO as the input, and specifies two outputs: address B with 7 BTC and the change address A_{c1} with 3 BTC. All outputs in Transaction 1 are UTXOs before they are referenced by other transactions. Likewise, to send 6 BTC to address C belonging to user Charlie, Bob initiates Transaction 2 by referring to the UTXO generated in Transaction 1 as the input, and specifies outputs accordingly.

The semantic of a transaction is to fully consume UTXOs specified in inputs, and distribute to its output addresses with specified values. Note that in order to make this transaction verified and confirmed by the Bitcoin network, additional validation information to verify the ownership of each UTXO and the integrity of the whole transaction must be provided. Besides, to broadcast a transaction to others in the P2P network, users pay *network fees* to the miners who spent their computational work to verify transactions.

Addresses. There are three types of *addresses* in Bitcoin. Addresses calculated directly from private keys (using hash functions) in outputs are called Pay-to-Public-Key-Hash (P2PKH) addresses and start with `1`. In 2012, a new type of address called Pay-to-Script-Hash (P2PSH) was introduced to simplify the long redeem script in transaction output for the multiple signature (Multisig) protocol, and these addresses start with `3`. In 2017, another new mechanism was introduced in Bitcoin as the segregated witness (Segwit) to separate witness data (to verify the ownership of UTXOs) in transaction inputs, and the corresponding addresses start with `bc1q`.

2.2 Mixing Service

Originating from the Bitcoin community [8, 2, 7], the underlying idea for *mixing* is to obfuscate the relationship between inputs and outputs, thereby preserve the *relationship anonymity*.

Centralized Mixing Service. A mixing service is called a centralized mixing service if it relies on a central mixing server to perform the mixing. Many famous and leading services, such as Bitcoin Fog [7], are centralized mixing services. However, centralization inevitably introduces the inherent *trust* problem. First, such services provide no

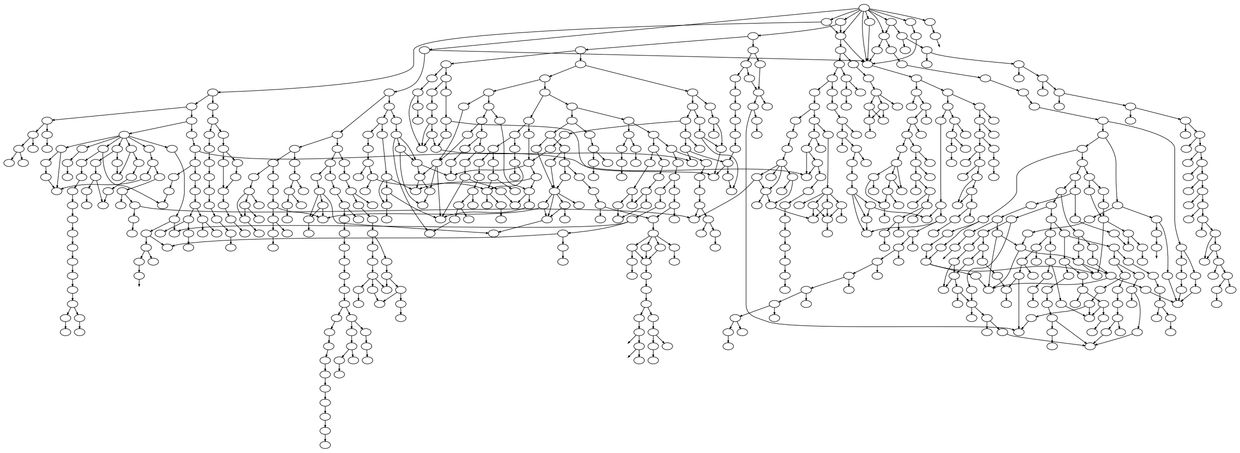


Figure 2: Simplified transaction graph for the Binance May Hack case. Nodes represent transactions. An edge means outputs of the source transaction are spent by target transaction.

guarantee that it will send the mixed coins to addresses specified by users, or delete service logs routinely as they announced. Second, because the service is centralized, it records the *original* relationship between inputs and outputs. Thus, if the service itself is compromised, the anonymity will be revealed. Lots of centralized mixing services disappeared in recent years, including BestMixer [23], Helix [31] and BitMixer [25].

Decentralized Mixing Service. As the name suggests, the decentralized mixing service does not rely on a centralized server to perform the mixing. *CoinJoin* [33] is a generic *decentralized* mixing protocol proposed by Bitcoin Core developer [2]. The basic idea for *CoinJoin* is to exploits the structure of transactions to combine different inputs and outputs in a single transaction, thereby make matching outputs with inputs much harder. A number of works have been proposed on the basis of CoinJoin, including *CoinShuffle* [49] and *SecureCoin* [28].

Cross-Blockchain Mixing Service. There is also a special type of mixing services provided by cryptocurrency exchanges/converters (e.g., ShapeShift [51], Changelly [15] and Flyp.me [24]). These services allow users to exchange Bitcoin with other cyptocurrencies, including altcoins of Bitcoin (e.g., Monero) and coins from other platforms (e.g., Ethereum). Obviously, tracking the money flow across different ledgers is not a trivial task.

In this work, we also perform a study on ShapeShift to understand its mixing mechanism. However, we only focus on mixing data and activities within the Bitcoin blockchain.

3 Abstraction Model for Mixing Mechanisms

As introduced in Section 2, the basic idea of mixing is to hide relationships between senders and recipients (inputs and outputs), to provide *relationship anonymity* [38]. We propose an abstraction model by separating the mixing workflow into three phases, and then illustrate the mixing mechanisms.

3.1 A Motivating Example

Here we use a real attack called the *Binance May Hack* event [5], as the motivating example to demonstrate the difficulty and complexity to trace the money flow associated with a mixing service. According to the official announcement of Binance [5], the attacker stole $7,074$ BTC and withdrew them in just one transaction [3].

Unfortunately, Binance did not provide any detailed information. As a matter of fact, the stolen digital assets were then distributed by using Chipmixer to perform the money laundering. Figure 2 gives a simplified transaction graph of this attack. Specifically, the root node (i.e., the topmost node) in Figure 2 represents the withdrawal transaction initiated by the attacker to transfer out the stolen digital assets, and the subsequent graph shows the tainted money flow through multiple transactions.

[2] CoinJoin can be implemented in centralized mixing services as well [30].

[3] The transaction hash is `e8b406091959700dbffcff30a60b190133721e5c39e89bb5fe23c5a554ab05ea`, and we will use `e8b406` to refer to it in the following context.

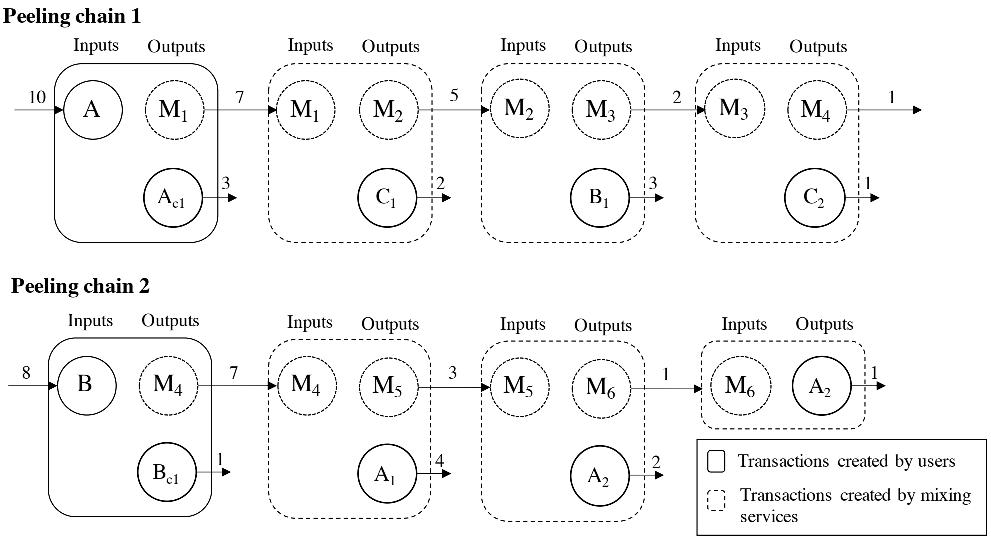


Figure 3: An example of the swapping mechanism. In this figure, we use M_1 to M_6 to denote addresses maintained by the mixing service. By swapping different user inputs and outputs, the relationship anonymity for all addresses is preserved. For instance, the relationship from A to A_1 and A_2 is anonymized.

Obviously, the graph of Figure 2 is quite complicated to distinguish transactions due to the existence of the mixing service, which inevitably brings in unrelated transactions that need to be excluded. To solve this problem, we first provide a general understanding and novel classification of current mixing services.

3.2 Mixing in Three Phases

The workflow of a mixing service can be modeled as a three-phase procedure, i.e., *taking inputs*, *performing mixing* and *sending outputs*.

Specifically, the mixing service will first take Bitcoin to be mixed as the inputs. This is achieved mostly by requiring users to send inputs to a service-provided deposit address. Some other services (e.g., those using CoinJoin protocol) may require users to send an UTXO to them. For these services, the whole UTXO will be consumed by the mixing service on behalf of the user. Besides, some necessary parameters, including output addresses, are typically specified in this phase.

After taking inputs, the mixing service is responsible for performing mixing with its mixing mechanism. The mixing mechanism consumes the collected user inputs, and prepares the desired outputs for each user. Note that there exists different mixing mechanisms, which will be discussed in Section 3.3.

Finally, the mixing service will send the desired outputs to the users. Typically, users specify some output addresses to the service to indicate where the mixing output should be sent. Alternatively, some services may provide users with private keys of those mixing outputs that can be imported by wallets. In this way, the users can spend these outputs at will and save transaction fees.

3.3 Mixing Mechanisms

The relationship anonymity is mainly achieved by the mixing mechanism of the second phase. According to different implementations, current mixing mechanisms can be further categorized into two types, i.e., *swapping* and *obfuscation*.

3.3.1 Type I – Swapping

The basic idea of this mechanism is to swap the inputs and outputs of different users to preserve the relationship anonymity. As shown in Figure 3, instead of directly sending 7 BTC from A to A_1 and A_2, the mixing service will swap the outputs of B to them (in *Peeling chain* 2). Similarly, B_1 is from outputs of M_2, which is from A. In this paper, we call the transactions crated by mixing services as *mixing transactions*.

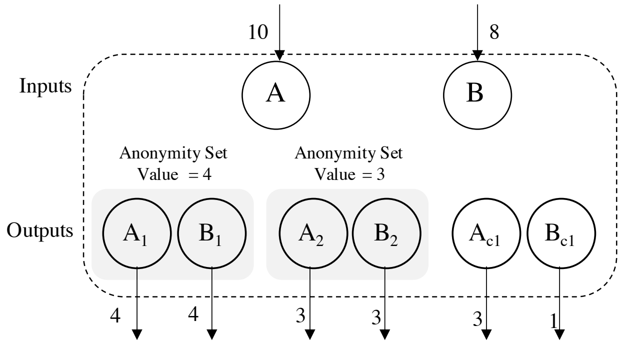


Figure 4: An example of obfuscation within a single mixing transaction. The mixing service generates two *anonymity sets* with the size 4 and 3, respectively.

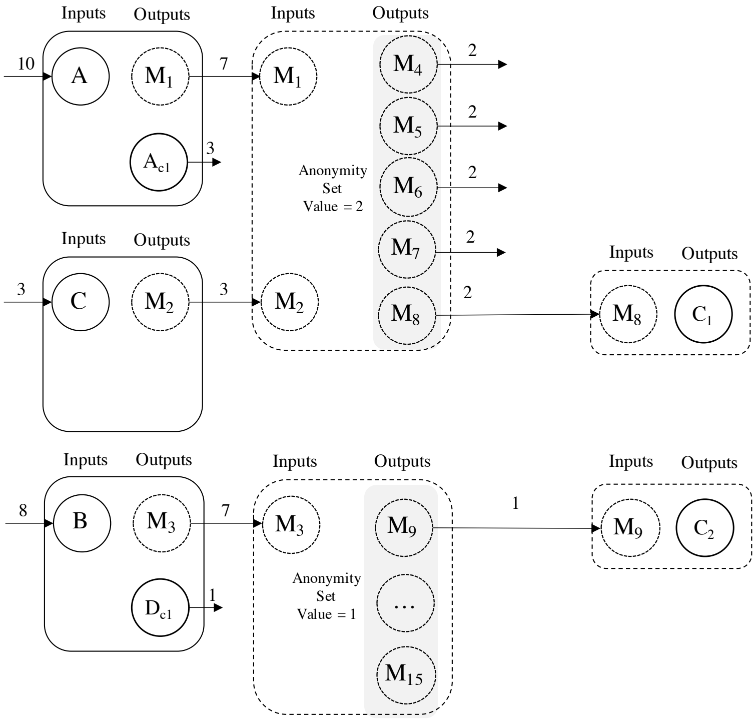


Figure 5: An example of obfuscation with multiple mixing transactions. The mixing service generates multiple anonymity sets with different values (2 and 1 in this case). From the figure, C_1 and C_2 are from M_8 and M_9 with input value 2 and 1 BTC. Other outputs, e.g., M_4 and M_{15} will be used to mix other inputs.

Despite the simple and effective idea of swapping, there is an important assumption that mixing transactions are hidden by the service. Otherwise if we can identify all mixing transactions, matching can be used to recover the original relationship between inputs and outputs. For instance, if we discovered all mixing transactions in Figure 3, then we can find out that output value M_1 is equal to the input value M_4 of a mixing transaction. Then we can infer that M_1 and M_4 are swapped and the original output of A is A_1 and A_2.

To prevent mixing transactions from being identified, the concept of *peeling chain* was observed in the wild [38]. A peeling chain is a set of transactions generated by mixing service that form a chain to distribute user outputs. The unique property of the peeling chain is that transactions in the chain are similar to normal user transactions, with one input and two outputs [26]. Thus, mixing transactions cannot be easily distinguished from normal user transactions.

For a mixing transaction in the peeling chain, one of the outputs is used to generate the output for a target address and another is used for the change, which in turn becomes the input of the next chain node. In Figure 3, the input to M_1 is separated into two outputs, one is 2 BTC to C_1 and another is 5 BTC to M_2. The latter output then becomes the input to another mixing transaction. We will discuss the peeling chain in detail in Section 5.3.

3.3.2 Type II – Obfuscating

This mechanism aims to preserve the relationship anonymity by breaking the matching procedure between user inputs and outputs. Specifically, obfuscating is achieved by hiding user outputs in *groups*.Figure 4 shows an example of mixing transaction of obfuscating mechanism. There are two groups of outputs with the same value within this transaction and outputs in each group are indistinguishable. For example, it is impossible to determine which output A_1 or B_1 in the first group originates from input A. These groups of outputs with the same value is called *anonymity sets*. It is hard to identify the real outputs for each input without additional information.

Moreover, the obfuscation could be achieved within single or multiple mixing transactions. Figure 4 and Figure 5 are two examples, respectively. Compared with single mixing transaction, multiple mixing transactions can generate fine-grained outputs with user-specified value. For instance, when using the mixing service, users C can specify that the outputs for C_1 and C_2 are 2 and 1 BTC. Conversely, the service determines value for each anonymity set for each transaction in the case of single mixing transaction. Note that in both implementations, there are some transactions that involve multiple inputs and outputs. Compare with *consecutive* outputs in the previous mixing mechanism, they are inevitable as the service must generate anonymity sets with the same value.

4 Methodology

In this section, we will introduce our methodology used to analyze mixing services. In particular, we will first select representative mixing services and then collect sample transactions. After that, based on these transactions, we will perform transaction-based analysis to identify mixing mechanisms used by these services.

4.1 Select Representative Mixing Services

To select representative mixing services, we use *BitcoinTalk* [9] and other public media as the information sources. As the biggest Bitcoin-dedicated forum, BitcoinTalk is served as the “official” forum for Bitcoin. Besides, we also pay attention to reports from other public media. For example, ShapeShift was investigated and reported by the Wall Street Journal [54] for being used by criminal activity as the money laundering tool.

Besides, we take the diversity of mixing mechanisms into account. We aim to select services with different mixing schemes to provide a more comprehensive analysis.

4.2 Collect Transactions

To analyze a mixing service, we first obtain some ground truth transactions to understand the behavior of the service. We refer to them as *sample transactions* in this work. For a typical mixing service, sample transactions include *input transactions* used by users to send inputs to the service, and *output transactions* used by service to send mixed outputs to user specified target addresses. The input transactions are initiated by users, while the mixing and output transactions are initiated by the service.

In this work, we have used two complementary methods to acquire sample transactions.

Method I: Interacting with Mixing Services. We use the mixing service to actually mix Bitcoins by sending real Bitcoins to the service and then collecting the input and output transactions. This method would be restricted by the budget constraint as some mixing services may place a high input threshold or may charge a relatively high mixing fee. Therefore, we can only conduct several experiments with these services.

Method II: Using Public APIs of Services. Some mixing services provide public APIs to facilitate their usage. For instance, they provide APIs that could be used to query detailed information and update status of a mix or inspect current statistics of a mixing service. Some of them may reveal too much information or require less authentication so that mixing records of mixing service users can be reconstructed.

4.3 Basic Transaction Analysis

Based on obtained sample transactions, our next step is to determine the mixing mechanism used by a service provider and understand its mixing workflow. This is achieved by performing a transaction-based analysis on the transaction graph. There exist two challenges in performing such analysis. In the following, we will first discuss these challenges and our way to solve them, and then illustrate our approach to study mixing mechanisms and workflows.

4.3.1 Solve Challenges

We have to deal with the following two challenges to conduct our analysis.

Challenge I: Identify Address Types. When constructing the transaction graph from sample input transactions, we first need to distinguish the users' addresses and addresses used by mixing services. Otherwise, our graph is too big (has too many nodes and paths) to be analyzed and introduces false positives.

To address this challenge, we pay special attention to the user behavior in transaction analysis, and observe that users' addresses tend to belong to the same address type (address types is introduced in Section 2). This also applies to addresses of mixing services.

Based on this observation, we can distinguish addresses when there exist two different types of addresses in a sample transaction. For example, if there are two types of addresses in a transaction and one of them is determined to be used by mixing services, then the other type is considered to be used by users. We can then prune the branch of the transaction graph at the user output side because it does not belong to the mixing service. Besides, it also helps to distinguish user and service outputs in a sample transaction.

Challenge II: Identify Peeling Chains. Although peeling chains are commonly observed [38, 22], they have not been carefully analyzed. A peeling chain can be modeled as a structure consists of three components, including *starting point*, *chain nodes* and *ending point*, which will analyzed and distinguished accordingly.

Specifically, a *starting point* is the transaction that user sends an input to an address given by a mixing service, e.g., M_1 in Figure 3. There are two possible methods to distinguish the starting point in sample transactions. First of all, based on the *multi-input and change address heuristic* [46], this transaction initiated by user should have only two outputs, one of which is service provided deposit address and another is the change address. Secondly, the address type can also be used to distinguish change output from service output, if the two outputs have different address types.

Chain nodes are used to distribute user outputs and continue the peeling chain. The structure of chain nodes is simple with one input (which is a reference to output from the previous node), and two outputs (one for user output and another for the successive node). Cases are that chain nodes are indistinguishable from the starting point. In this case we trace backwards until a transaction with multiple inputs are found, and inspect manually to find the starting point.

An *ending point* is the end of a peeling chain. The remaining changes at the tail will be handled by the service. For instance, these changes could be used as inputs for another mixes. Our observation suggests that if the changes from a chain node is used in a transaction with many inputs, the corresponding chain node will be regarded as the ending point. Mixing services will collect these remaining changes for future use.

4.3.2 Determine Mixing Mechanisms

For the analysis, we define the *context* as the destinations of inputs and sources of outputs in sample transactions. Then we determine the mixing mechanism by examining contexts of these transactions.

As introduced in Section 3, the major transaction-level difference of the swapping and the obfuscation mechanism comes from their*pattern of outputs*. For swapping, mixing outputs are consecutive, while they are centralized for obfuscation. We examine the context of outputs in sample transactions and the difference serves as a centering criteria to distinguish the used mixing mechanisms.

1. If there is a transaction generating some outputs with identical values (i.e., anonymity sets) in the context of each output in sample transactions, it uses the obfuscation mechanism.
2. If most transactions have two outputs, and they form a chain using change addresses in the context of each outputs, it uses the swapping mechanism.

4.3.3 Understand Mixing Workflow

After the mixing mechanism is determined for each service, we then try to figure out their mixing workflows accordingly, or how does the service perform mixing.

Swapping Mechanism. For mixing services using the swapping mechanism, the peeling chain is the central structure used in a mixing workflow. As introduced in Section 4.2, sample transactions consist of user inputs to service provided addresses and the service outputs to distribute user outputs.

Thus, we find peeling chains for sample transactions to analyze the mixing workflow. To understand peeling chains and distinguish their three components, we visualize representative peeling chains of the service and perform manual analysis. We also identify the way the service handles remaining changes in each peeling chain.

Algorithm 1: The Seed-Expansion Algorithm

```
Data: Seed transaction set $S$ from the mixing service.
Result: Expanded transaction set $E$, in which each element is highly likely to be related to the mixing service.
Initialize a queue $\mathbf{Q}$ with all element in $S$;
Initialize $E$ to be an empty set;
while Queue $\mathbf{Q}$ is not empty do
    Take a transaction $T$ from $\mathbf{Q}$;
    Put $T$ into the set $E$;
    for every output $O$ in $T$ do
        Find transaction $T_O$ that uses the output $O$;
        for every input $I$ in $T_O$ do
            Find transaction $T_I$ referred by input $I$;
            if $T_I$ generates anonymity sets and $T_I$ not in $E$ then
                En-queue $T_I$ into $\mathbf{Q}$;
            end
        end
    end
end
```

Obfuscating Mechanism. For mixing services using the obfuscation mechanism, we focus on transactions that generate anonymity sets. For every inputs and outputs in sample transactions, we will find corresponding transactions that generates anonymity sets to consume the inputs or sending the outputs.

4.4 Advanced Transaction Analysis

Besides the previous analysis, we also take an further analysis to identify mixing transactions for services that adopt the obfuscation mechanism. This is important as it can generate money flow, calculate the service's profit and help investigate money laundering. We take a two-step analysis with a seed input.

Step I: Identify Anonymity Sets. Our first step is to identify anonymity sets using a seed input, which is fed into the mixing service with a service-provided address (e.g., M_1 in Figure 5). Then, we can locate addresses in the anonymity set by finding outputs with the same value. We color each address (M_4 to M_8) in the identified set. We also color the outputs for transactions that take the colored address as inputs, e.g., the address C_1 is colored.

Step II: Identify More Anonymity Sets. We then perform further analysis to identify more transactions. In particular, if we find a transaction with multiple inputs that takes a colored address as input, then we color other input addresses. For example, if we find there exists a transaction with C_1 and C_2 as inputs, then we will color C_2 too. We did not color C_2 in previous step since we only use the input A as the seed input. Input to B is not our seed input.

After that we perform a backward analysis from C_2. In particular, we move backward from address C_2 and try to locate transactions that have same output values, e.g., from M_9 to M_{15}. These outputs with same value means that new anonymity sets are detected. We color them and perform the similar analysis from each address.

Notice that during this step, we may not locate any anonymity set. In this case, we will remove the color accordingly. For instance, if E_1 and C_1 are inputs for a transaction, then E_1 will be colored. However, E_1 may come from outputs of normal user transactions. In this case, we will remove the color for E_1.

In summary, the whole analysis algorithm can be formalized in Algorithm 1. By applying it with seed inputs, we can identify mixing transactions and corresponding addresses used for mixing.

5 Evaluation Results

In this section, we apply the proposed approach in Section 4 and summarize the results.

5.1 Selected Services

In this work, we select the following four services for evaluation.

Chipmixer [16] is currently one of the most popular mixing services. Its popularity originates from its "Pay What You Want" (PWYW) pricing strategy and some innovations over traditional mixing services. In addition, it was reported

Table 1: Sample transactions obtained for selected services.

Service	Acquisition Method	# of Samples Obtained
Chipmixer	Interacting with the Service	20 (5 inputs + 15 outputs)
Wasabi Wallet	Using Public APIs	$4,850$
ShapeShift	Using Public APIs	$6,381$ (Bitcoin) + $1,089$ (Litecoin)
Bitmix.biz	Interacting with the Service	20 (6 inputs + 14 outputs)

that Chipmixer was involved in a security breach of Binance in 2019, where it was used as a money laundering tool by the attackers to launder over 4,000 BTC [18].

Wasabi Wallet [62] is one of the official recommended desktop Bitcoin wallets [43], and the only (currently accepted and popular) wallet with built-in CoinJoin functionality [62]. It claims to focus on user privacy protection.

ShapeShift [51] is one of the most famous cryptocurrency converters. It was reported by the Wall Street Journal to be used as a money laundering tool for over 9 million tainted funds over a time period of two years. Due to the pressure from the public media and regulator, ShapeShift has applied Know-Your-Customer (KYC) policy and requires strict identification information to set up an account before using the service.

Bitmix.biz [11] is yet another mixing service announced on August, 2017 [10]. It claimed to have some improvements over its predecessors like dust-attack prevention, letter of guarantees (to redeem funds on exceptions), and randomized transaction fees and delays. The wider range of supported cryptocurrencies (Litecoin and DASH) and lower mixing fee (from 0.4%) also contributes to its popularity.

5.2 Sample Transaction Acquisition

As introduced in Section 4.2, there are two possible methods to obtain sample transactions. We first conduct an complete analysis on these services to determine which method to use. For Wasabi Wallet and ShapeShift, we find public APIs provided can be used to obtain sample transactions. In contrary, for Chipmixer and Bitmix.biz we resort to direct interaction with the service. Table 1 summarizes the collected sample transactions.

5.2.1 Interacting with Services

In the following, we will describe the details of obtaining sample transactions by interacting with Chipmixer and Bitmix.biz, respectively.

Chipmixer. According to its pricing strategy, Chipmixer can be a free service. However, it only recognizes inputs up to 3 digits after the decimal point and any trailing value after that will be considered as service fees or donations.

This service provides a generated address for users to send inputs. When an input is confirmed, the next step is to decide how to distribute the output into *chips* [4]. After the distribution of outputs, users can withdraw these outputs by either importing the private keys or specifying output addresses separately.

In total, we conducted 5 experiments and received 15 outputs.

Bitmix.biz. Users of Bitmix.biz can directly set mixing parameters and send mixing requests, like target addresses, the delay from the mixing request to output received, value distribution (distributions for each address) and overall transaction fees. This service supports up to 20 output addresses while delay and distribution for each address can be independently specified. After a mixing request is sent, the service will provide a temporary address to receive user inputs. Once the inputs are confirmed, it will send corresponding user outputs according to delay requested by the user.

In total, we conducted 6 experiments and received 14 outputs.

5.2.2 Using Public APIs from Services

As stated in Section 4.2, services may provide different *levels* of public APIs. Specifically, Wasabi Wallet provides some APIs which are directly usable to obtain transactions. In contrary, ShapeShift only provides APIs that must be

[4] Chips are defined as user outputs with predefined values [16].

used with additional blockchain data and specific matching algorithms. We set up crawlers to collect data through these APIs.

Wasabi Wallet. It provides two APIs to fetch mixing-related data: 1) the API `states` [58] is used for the clients to query and update current phase and status of current CoinJoin transaction; and 2) the API `unconfirmed` [59] broadcasts transaction hashes of all successful CoinJoin transactions before they are confirmed by the blockchain.

These two API are for status querying and updating purposes. However, Wasabi Wallet server does not require any authentication to access them. Therefore, we set up a crawler that periodically retrieves information from these two APIs. The crawler accessed these APIs every 1 minute and continued for 82 days (from December 26, 2019 to March 15, 2020).

In total, we gathered $4,850$ transactions that were confirmed by the Wasabi Wallet. We will use these transactions as the seed set for our experiment described in next section.

ShapeShift. It provides some APIs to facilitate its use. While ShapeShift requires a registered account and personal identification information, using these APIs requires no authentication.

There are two key APIs that can be used to get sample transactions. The first API is called `recenttx` [52]. It provides information about all recent convert records in ShapeShift. For each convert record, it will return a tuple of `<curIn, curOut, timestamp, value>`, which records the type of input and output cryptocurrencies, timestamp of the convert, and input currency value in decimal. The second API is called `txstat` [53]. For a given address, it will provide detailed information if it belongs to ShapeShift. Note that these APIs require no authentication and can be used to obtain sample transactions.

In total, we crawled $27,411$ convert records from December 11, 2019 to March 18, 2020. We focus on convert records from Bitcoin to other cryptocurrencies. In the crawled records, we found $7,067$ records with Bitcoin as the input cryptocurrency.

To further identify corresponding transactions for a given convert record, we proposed a refined algorithm based on [61]. This algorithm consists of three steps. First, we obtain a list of recent cryptocurrency convert records using the `txstat` API. After that, for each record (with value v and timestamp ts), we locate candidate transactions with the closest values to v with closest timestamps to ts. Finally, these transactions will be further validated by applying the `txstat` API. We have applied this algorithm on these $7,067$ records, and successfully matched $6,381$ convert records (90.29% of all records) with detailed information.

So far, the transactions we obtained are input samples, we also need to determine their destinations to analyze the complete convert workflow. Additionally, we want to analyze where the Bitcoin comes from in the case of converting *altcoin*s to Bitcoin. To this end, we choose Litecoin by its popularity in Shapeshift, and found $1,097$ records converting Litecoin to Bitcoin. Then we apply the proposed algorithm to these records and $1,089$ (99.27%) records are matched with detailed information.

5.3 Basic Transaction Analysis

We have applied basic transaction analysis discussed in Section 4.3 to the four selected services. In the following, we briefly describe the results and findings in our analysis.

5.3.1 Determining Mixing Mechanisms

Obviously, as Wasabi Wallet implements the CoinJoin protocol [62] that generates anonymity sets, this service uses the obfuscating mechanism. Apart from the Wasabi Wallet, the mixing mechanisms used by other three services are determined by visualizing obtained samples and context transactions.

To determine the mixing mechanism used by Chipmixer, we first plot sample transactions and their contexts. Figure 6 is the transaction graph for the experiments conducted with Chipmixer.

The figure shows that, all of our inputs (grey nodes) are immediately consumed by mixing transactions (light grey nodes) by the service, and our outputs (black nodes) also come directly from them. Transactions in light grey nodes are of special type that generates anonymity sets (outputs with the same value). Therefore, the existence of mixing transactions generating anonymity sets indicates that Chipmixer uses the obfuscating mechanism. Because all outputs from these mixing transactions are of specified value (as mentioned in Section 4.2), Chipmixer generates a fixed number of large anonymity sets.

Similarly, we applied the same approach to the other two services. Based on the corresponding transaction graph, we conclude that both ShapeShift and Bitmix.biz use the swapping mechanisms. For example, in Figure 7, all of our

Table 2: Mixing mechanisms used by services.

Service	Swapping mechanism	Obfuscating mechanism
Chipmixer		√
Wasabi Wallet		√
ShapeShift	√	
Bitmix.biz	√	

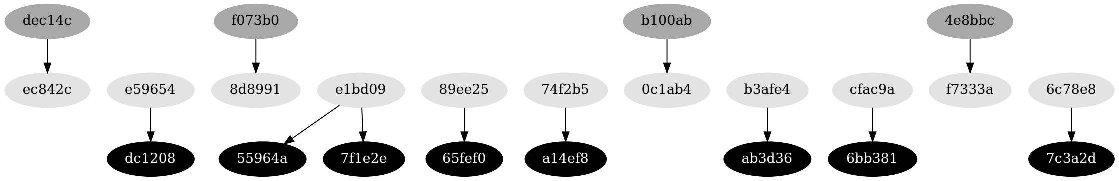


Figure 6: Transaction graph of Chipmixer experiments. Grey and black nodes are our input and output transactions respectively. Transactions in light grey nodes generates anonymity sets.

outputs come from mixing transactions with only two outputs (which means no anonymity sets get involved). Tracing our outputs backward shows several chains, in which most transactions have single input and two outputs, and they are connected with change addresses. Again, according to section 3, this is a feature of peeling chain. In the case of Shapeshift, some chain nodes have multiple inputs. Further analysis reveals that Shapeshift will return Bitcoin back to users with large inputs by "merging" several peeling chains in some chain nodes. Our model suggests that these nodes are the ending points of these chains. Then the service reserve the remaining change for future use, like starting a new chain.

The mixing mechanisms used by all these services are summarized in Table 2.

5.3.2 Understanding Mixing Workflows

To better understand the mixing services, we need to figure out the mixing workflow for these services.

Chipmixer. Users first send their inputs to the service. Then, the service generates chips (in anonymity sets) in mixing transactions. Lastly, the service returns those chips back to the users.

Figure 8a gives an example with two mixes. In mix #1, two users (Alice and Bob), send their inputs to the service. These inputs are aggregated by the service into mixing transaction #1, which generates an anonymity set with value 0.1. The outputs in this anonymity set will be distributed to users. If the inputs do not fit the anonymity set properly, then there will be change left as an input for another mix. As a result, the inputs of mix #2 come from another two users (Charlie and Dave) along with the change of mix #1. The anonymity set generated by mix #2 have a value of 0.5, which is compatible with the inputs without any change.

Wasabi Wallet. Unlike other services that create addresses for users to deposit Bitcoin, this service requires users to send UTXOs (in a wallet) and output addresses. Then, the service creates a number of anonymity sets with a change set in one CoinJoin transaction. Finally, the service transfers outputs to corresponding addresses.

Figure 8b gives a concrete example. In step 1, users of this CoinJoin round (i.e. this mix), Alice, Bob, Charlie and Dave, submit UTXOs they want to mix and target addresses to the service. Then in step 2, two anonymity sets with value 0.1 and 0.2 are generated respectively. Finally in step 3, outputs in anonymity sets and changes are sent correspondingly to the target addresses. As a result, outputs in the anonymity set are hidden, but the change are not anonymized (not in an anonymity set) and may require further CoinJoin rounds.

ShapeShift. Users first send their Bitcoin to addresses provided by the service and specify target addresses in the other blockchain. Then the service takes responsibility for the swapping by performing cross-blockchain transactions. Finally, users can receive the designated coins from the corresponding blockchain.

Figure 9a gives a concrete example. In the Bitcoin blockchain, Alice sends 3 BTC to ShapeShift and receives 127.11 Ether in the Ethereum network later. Obviously, this service has to make efforts (e.g. in collaboration with cryptocur-

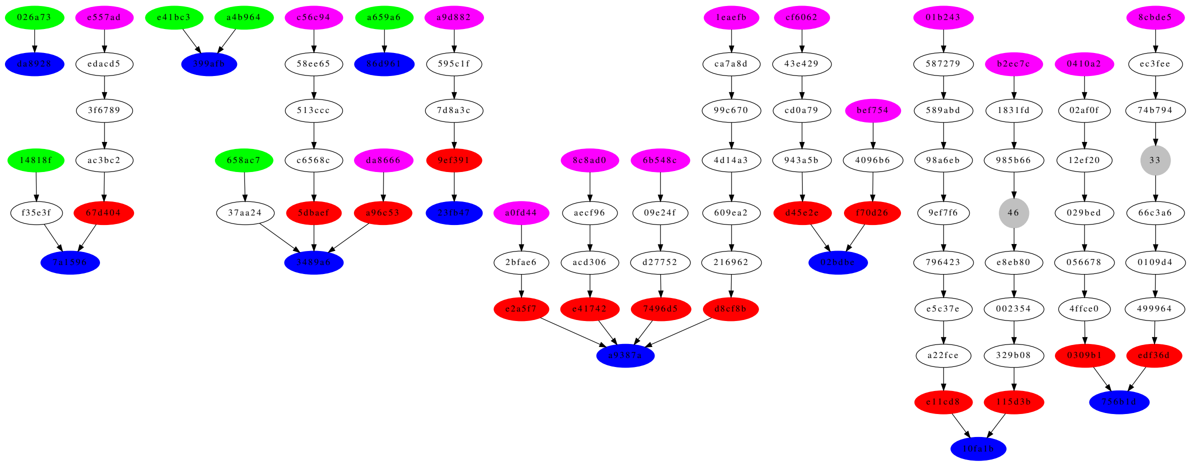


Figure 7: The transaction graph of Bitmix.biz. Green nodes represent our input and output transactions. Blue nodes are end points, while magenta nodes are potential starting points of the peeling chains. Grey circles with numbers denote chains and the number shows the length of the omitted chain. User output in each chain node is omitted.

rency exchanges) to break even among different blockchain platforms. Due to the swapping mechanism adopted by the service, the Bitcoin sent by Alice will be organized as a peeling chain to distribute Bitcoins to other users (e.g., Bob and Charlie in this figure) who swap other cryptocurrencies for Bitcoin.

Bitmix.biz. Users first send their Bitcoin to addresses provided by the service. Then, the service creates peeling chains to distribute the outputs. Finally, users receive their outputs from the chain nodes in peeling chain.

An example of peeling chain for Bitmix.biz is shown in Figure 9b. Similar to ShapeShift, Alice sends 3 BTC to deposit address 3Hp1Fk generated by the service. This input will be distributed to Bob with 2 BTC as output and an temporary change address #1 with 1 BTC. Then the balance of the address #1 will be distributed to Charlie with 0.5 BTC as output and another temporary change address #2 with 0.5 BTC. The address #2 is a special address that holds the remaining change after distributing user outputs and its balance is too small to enter the next round. As a result, this trailing change will be consumed by a special transaction. This transaction, like chain nodes with multiple inputs in Shapeshift, consumes remaining changes from multiple peeling chains and merge them into a large balance for further use.

5.4 Advanced Transaction Analysis

As discussed in Section 4.4, mixing services using obfuscating mechanism allow us to identify more mixing transactions using only a group of seeds. Therefore, Chipmixer and Wasabi Wallet can be further analyzed accordingly.

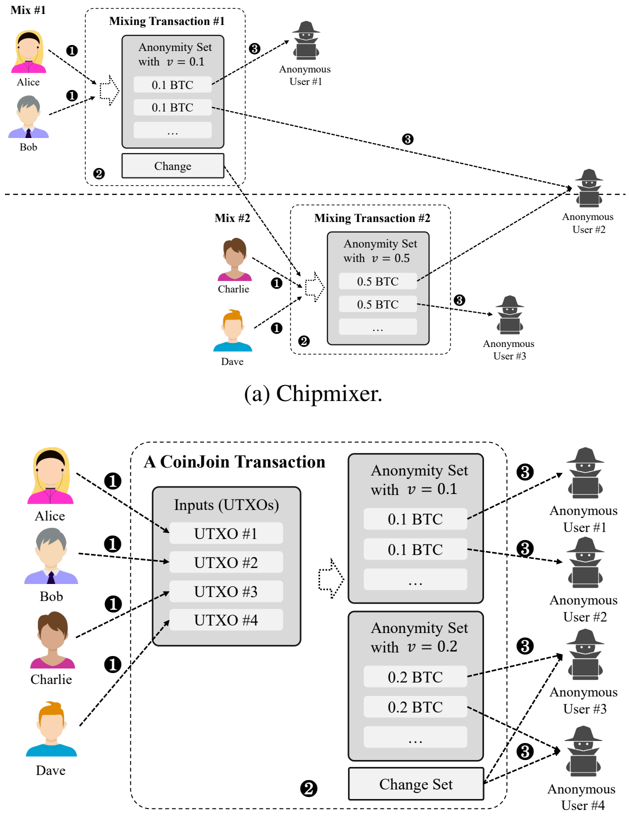
In the following section, we first evaluate the effectiveness of the proposed Algorithm 1. Due to space limit, we only report the result for Chipmixer. Then based on insights observed from identified transactions, we are able to measure the profit made by each service. Finally, we provide a case study to demonstrate the capability of tracking the provenance of Bitcoin based on identified mixing transactions from our proposed algorithm.

5.4.1 Measuring the Effectiveness of the Algorithm

Due to the lack of ground truth, we manually investigate our own ground truth to support the measurement. Specifically, for each service, we first collect transactions according to the common features we observed from the sample transactions and then filter false positives with manual investigation. After that, we are able to evaluate the robustness the proposed algorithm by comparing the result with that ground truth.

Chipmixer. We conducted the following four experiments with different seeds (note that the 20 sample transactions in Section 5.2 are used as the original seed set).

- *Experiment 1.* We performed the first experiment at block height $609,750$. Using all centralized transactions spotted in experiments as the seed set, we found $8,279$ transactions, which are potentially generated by Chipmixer.
- *Experiment 2.* We performed the second experiment at block height $619,700$ with the same seed set, and found additional $1,056$ transactions ($9,335$ in total) mixing transactions from Chipmixer.



(a) Chipmixer.

(b) Wasabi Wallet.

Figure 8: Examples for mixing workflows of Chipmixer and Wasabi Wallet.

Table 3: Three Experiments to Evaluate the Seed-Expansion Algorithm for Chipmixer.

Experiment	#1	#2	#3	#4
Date	Dec 25, 2019	Mar 1, 2020	Mar 1, 2020	Mar 1, 2020
Block Height	609,750	619,700	619,700	619,700
Seed Set	20	20	10	**1**
Expansion Set	8,279	9,335	9,335	9,335
Ground Truth	9,027	10,119	10,119	10,119
Coverage	91.71%	92.25%	92.25%	92.25%
Average Coverage	**92.07%**			

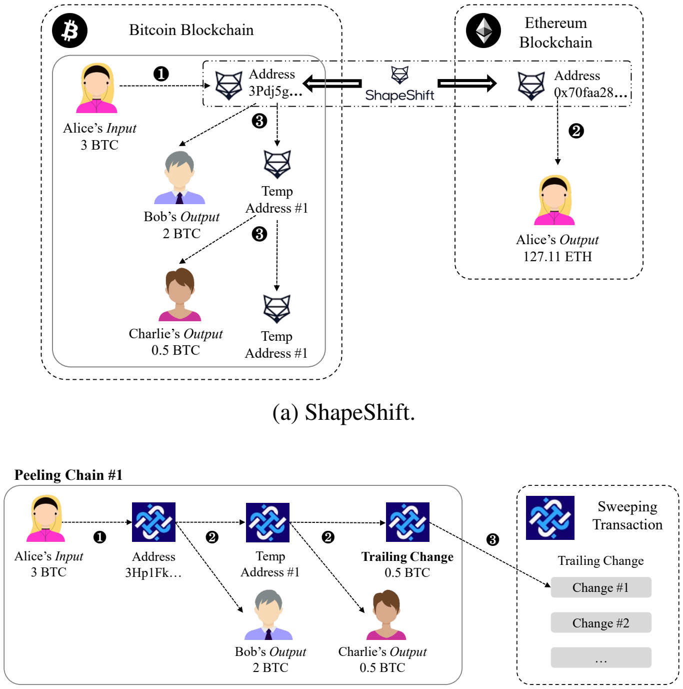
- *Experiment 3.* We conducted the third experiment at the same block height with experiment 2. The seed set was randomly chosen from original seed set with only half the size (10 transactions in total). We achieved the same expansion set as in experiment 2.

- *Experiment 4.* We conducted the final experiment at the same block height. The seed set was only *one* transaction randomly picked from original seed set. Again, we achieved the same expansion set as in experiment 2.

These four experiments demonstrate that our method to locate mixing transaction is robust against *different sizes* of the seed sets, and the same seed set E can be used in *different time* to identify mixing transactions from the same service. The summary of the experiments is shown in Table 3.

5.4.2 Calculating Profit of Mixing Services

Here we calculate profit made by the two services on the basis of the mixing transactions we identified.



(a) ShapeShift.

(b) Bitmix.biz.

Figure 9: Examples for mixing workflows of Shapeshift and Bitmix.biz.

Chipmixer. This service uses the Pay-What-You-Want (PWYW) pricing strategy (as described in Section 5.1), and will treat any change less than 0.001 BTC as fees or donations [5].

To calculate the profit earned by Chipmixer from such fees, we sum over all trailing changes of user inputs from May, 2017 to February, 2020. In total, Chipmixer received 16.6086 BTC as service fees (with the monthly average value 0.4883 BTC), which was considerably less than the total user inputs $53,044.8077$ BTC during this period. Figure 10a illustrates monthly profit earned by Chipmixer. Our calculation for Chipmixer only serves as a lower bound for overall profit for Chipmixer.

Wasabi Wallet. We present further analysis based on the $9,788$ transactions obtained using proposed Algorithm 1. Similarly, our goal is to estimate the profit harvested by Wasabi Wallet for the CoinJoin fees. As introduced in Section 5.2.1, Wasabi Wallet's profit comes from the CoinJoin coordinate fees. By analyzing every CoinJoin transactions identified, we found two common output address potentially for fee collection. Address 1[6] has been used in $5,319$ CoinJoin transactions, but is no longer active since September 20, 2019. Address 2[7] has been used in $3,204$ transactions and is currently active. In every CoinJoin transaction, output value to these two addresses is close to the estimated coordinator fees.

Therefore, it is likely that these two addresses are used to collect coordinate fees. Figure 10b illustrates monthly profit earned by Wasabi Wallet. In total, these addresses collected 120.9932 BTC (with the monthly average 8.058 BTC), and it serves as a good estimation for the fees collected from Wasabi Wallet CoinJoin service.

5.4.3 Tracing Money Flow of A Real Attack

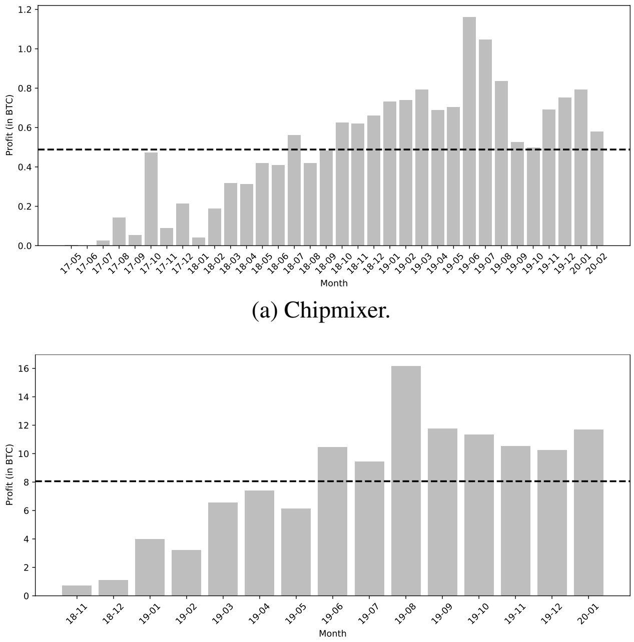
Finally, we demonstrate that our approach and results can be used to help to reveal money laundering by tracing the money flow of digital assets being hacked.

Specifically, we provide a simple case study for the Binance May Hack case [5]. In this case, the attacker stole $7,074$ BTC and used Chipmixer for money laundering. Starting from the attacker's output transaction `e8b406`, we track

[5] For example, user input of 0.0015 BTC will result in one chip with 0.001 BTC (with 0.0005 as the service fee), and an 0.0005 BTC user input will be considered as fees or donations.

[6] Address: `bc1qs604c7jv6amk4cxqlnvuxv26hv3e48cds4m0ew`

[7] Address: `bc1qa24tsgchvuxsaccp8vrnkfd85hrcpafg20kmjw`



(a) Chipmixer.

(b) Wasabi Wallet.

Figure 10: Monthly profits for Chipmixer and Wasabi Wallet. The profit is calculated by summing over all extra changes of inputs, and the dash line represents the average value.

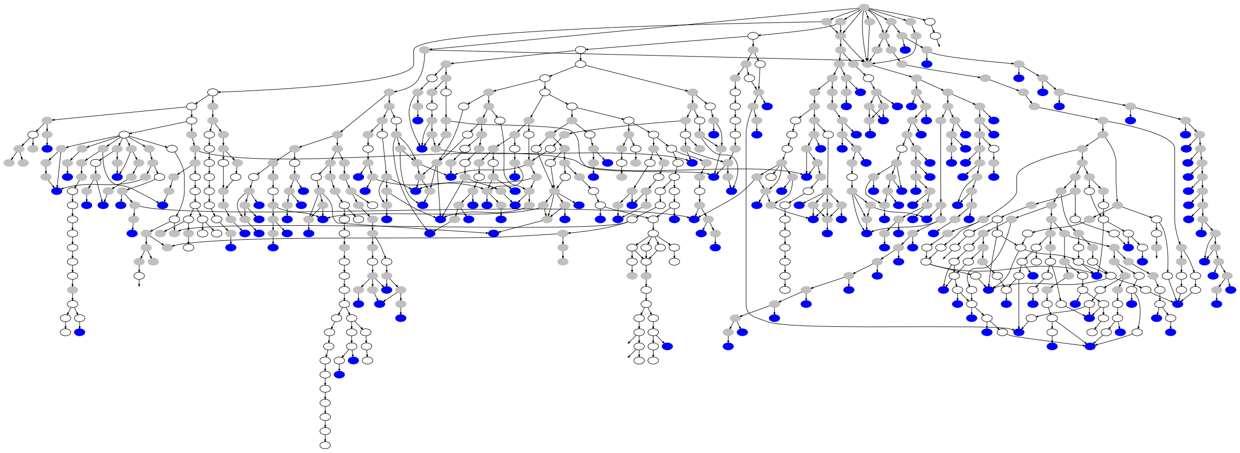


Figure 11: Simplified transaction graph for the Binance May Hack case. with transactions related to Chipmixer annotated by blue nodes. In total, attacker sent 4,792 BTC to Chipmixer.

down the transaction graph to see whether any tainted funds are sent to Chipmixer. We use the identified transactions in previous experiments to test if a transaction send Bitcoin to Chipmixer. To solve the problem of graph dimension explosion, we set the maximum depth of tracing to 50 and ignores outputs less than 0.9 BTC. It means that we only track those transactions that are within 50 steps from the original transaction and with large enough value.

In total, we found 157 transactions in identified transactions for Chipmixer, for a total value of $4,797.82$ BTC [8].Figure 11 is a *simplified* transaction graph to illustrate the case. In this figure, vertices are transactions where blue vertices indicates transactions sending the tainted funds to Chipmixer. The gray vertices indicates that addresses used in this transaction are `bc1q` addresses, which is coherent with the original outputs in transaction `e8b406`. Without the proposed approach, obviously, such a simplified graph implies that it may require lots of human efforts to manually investigate the provenance of the hacked Bitcoins.

[8] As a reference, an industry report [19] gives an estimate of $4,836$ BTC were laundered through Chipmixer.

6 Discussion

Threshold Parameter in Refined Algorithm. In Section 5.2, we propose a refined version of algorithm in [61]. In this algorithm, there is a threshold parameter limits the range of blocks examined. For the original algorithm, it is denoted as δ_a and δ_b and determined by an optimization algorithm to examine 2 blocks in total ($\delta_a = 1$, $\delta_b = 0$, plus the block with the closest timestamp). However, in our evaluation the original algorithm leads to poor performance (80.29% of all records matched, compared with 90.29% of our refined algorithm). After trying with different values, we manually set this parameter to examine 7 blocks in total, which is a trade-off between block coverage (larger threshold means more blocks examined) and performance (larger threshold means more false positives and less efficiency). As shown above, our refinement leads to a much better performance.

The Scope and Completeness of Our Model. To keep the completeness and conciseness of our model, we limit the scope of mixing services we inspected. In this paper, we only consider traditional mixing services that fully rely on on-chain mechanism to operate, without additional protocols. There are many complex research mixing protocols like Blindcoin [57] and Mixcoin [13]. To the best of our knowledge, they have no real-world implementations. Other real-world mixing protocols like Fair Exchange and CoinSwap, as investigated by [36], has much less popularity than traditional mixing services. Due to the protocol complexity and less popularity, we exclude them in our model and investigation.

Under this scope, our abstract model and classification for mixing services are **complete** because these two model are exclusive and complementary. The major features for the swapping mechanism are much less outputs in each mixing transaction (less than 3) and consecutive output pattern. In contrary, mixing transactions of obfuscating mechanism have more outputs and one or more anonymity sets, as well as the centralized output pattern.

Based on this model, the investigation and classification method described in Section 4 for four selected mixing services can be applied to **any** mixing services in our scope. In addition, our proposed Algorithm 1 can be applied to any mixing service that uses obfuscating mechanism.

Limitations of Our Work. Our work does have several limitations. First of all, the advanced transaction analysis does not cover mixing services using swapping mechanism. The design of peeling chains (as introduced in Section 3.3.1) deliberately hide mixing transactions by mimicking the features of normal user transactions. We may have to seek other approaches to identify peeling chains and recover the relations of the services. Indeed, it definitely becomes a challenge for further research.

Another major concern arises from our two-step approach to identify anonymity sets in advanced transaction analysis (Section 4.4). If any output generated by mixing transaction is incidentally not used as part of any transaction's inputs [9], then this approach would not be able to find that mixing transaction. Besides, this approach also relies on the sizes of the anonymity sets generated by mixing transactions. The smaller size will decrease the opportunity for outputs within the anonymity set to be used by other transactions as inputs, and thereby reduce the possibility to identify those transactions.

In addition, as there does not exist any available data, we have to build the ground truth by ourselves. Although we have made our best efforts to eliminate the false positives, it inevitably has some bias that affects the effectiveness of the measurement.

7 Related Work

Bitcoin Mixing Service. The basic idea of mixing is to preserve anonymity by obfuscating the relations from senders to recipients of a transaction. Several mixing services have been publicly announced since 2010, including *BitLaundry* [8], *Bitcoin Laundry* [2] and *Bitcoin Fog* [7]. In 2013, Maxwell made the idea of *CoinJoin* public to the community [33]. One year later, *Mixcoin* [13] was proposed as the first academic work. Since then, a number of mixing approaches have been proposed by applying different techniques, including *Fair Exchange Protocol* [27] and *Zero Knowledge Proof* [34], and some of them have been implemented as services. Generally speaking, there are mainly two types of mixing services, i.e., centralized (e.g., *Bitcoin Fog* [7], *Mixcoin* [13] and *Blindcoin* [57]) and decentralized (e.g., *CoinJoin* [33], *CoinShuffle* [49] and *CloakCoin* [20]). The centralized mixing service relies on a central mixing server to perform mixing, while decentralized mixing service allows users to perform mixing without any mixing server. There are also centralized mixing services using a decentralized protocol (like Wasabi Wallet using CoinJoin). Besides, mixing services like ShapeShift [51] allow mixing across different blockchains.

[9] Or in some rare cases, these outputs are used as part of a transaction's inputs, but all the other parts do not belong to any other anonymity sets.

Analyzing Bitcoin Mixing Service. Though mixing services have been widely used in the Bitcoin ecosystem, few studies have been published to understand them. Möser et al. [38] conducted the first empirical study to analyze three Bitcoin mixing services focused on money laundering. Yanovich et al. [60] provided a heuristic-based algorithm to detect mixing transactions, and revealed that mixing transactions constituted about 2.5% of all transactions. Balthasar et al. [22] applied the tool provided by Chainalysis [14] to analyze three selected services and discovered severe security flaws in these services. However, their methods are specific to selected services and cannot be generalized to other mixing services. Möser et al. [37] analysed the online *CoinJoin* market named *JoinMarket* and estimated its market volume. Jaswant Pakki [44] provides a more recent survey on mixing services in Bitcoin, in which author provides a complete table of mixing services with 9 trusted services. Unlike these previous studies, we propose a generic model and framework to systematically analyze state-of-the-art mixing services.

Analyzing Raw Anonymity of Bitcoin. A number of research papers have been published to analyze raw anonymity properties of Bitcoin [30] by either identifying the relations between Bitcoin addresses and user information, or clustering Bitcoin addresses according to user identification. Our work is closed to those that mainly focused on Bitcoin addresses by analyzing blockchain data. Reid et al. [46] proposed the first analytical results on the basis of two network structures, i.e., *transaction network* and *address network*, which can be used to depict money flow between transactions and users respectively. These two structures are widely used in subsequent researches [30]. Since then, several assumptions/methods were proposed and some of them have been used together to cluster Bitcoin addresses by users, including multi-input transactions [46, 47, 3, 35, 42, 55, 32], change addresses [3, 35, 55, 40] and behavior-based clustering [3, 48]. Although mixing services are rarely considered by these works, whose methods/findings (e.g., the multi-input heuristic) form the basis of our work.

8 Conclusion

In this work, we present a generic abstraction model and framework to demystify Bitcoin mixing services. Accordingly, we first categorize mixing services into two types based on different mixing mechanisms they used, i.e., the swapping mechanism and the obfuscating mechanism. Then we propose a transaction analysis to identify mixing mechanisms and workflows of these services. Lastly, we further propose a heuristic-based algorithm to analyse mixing services employing the obfuscating mechanism.

We then apply the proposed approach to four representative mixing services, and the evaluation results demonstrated the effectiveness of our approach. Specifically, we first successfully determine the mixing mechanisms and workflows for each service by conducting a basic transaction analysis. Then, we show that it is able to identify most (over 92%) of the transactions related to some type of mixing services by applying the proposed algorithm. Finally, we provide two case studies, including calculating the profit of services and investigating the money laundering activity, to demonstrate that our approach can be used to analyze behaviors relying on mixing services to hide their traces, which require identifying transactions related to the involved mixing services.

References

[1] Unspent transaction output. https://en.wikipedia.org/wiki/Unspent_transaction_output, 2020. (visited on 2020-05-02).

[2] Akemashite Omedetou. [ANNOUNCE] Bitcoin Fog: Secure Bitcoin Anonymization. https://en.bitcoin.it/wiki/Bitcoin_Laundry, 2011. (visited on 2020-05-22).

[3] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating user privacy in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 34–51. Springer, 2013.

[4] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. Data mining for detecting bitcoin ponzi schemes. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 75–84. IEEE, 2018.

[5] Binance. Binance Security Breach Update. https://www.binance.com/en/support/articles/360028031711, 2019. (visited on 2020-02-18).

[6] Stefano Bistarelli, Matteo Parroccini, and Francesco Santini. Visualizing bitcoin flows of ransomware: Wannacry one week later. In *ITASEC*, 2018.

[7] Bitcoin Wiki. Bitcoin Laundry. https://bitcointalk.org/index.php?topic=50037, 2011. (visited on 2020-05-22).

[8] Bitcoin Wiki. BitLaundry. https://en.bitcoin.it/wiki/BitLaundry, 2011. (visited on 2020-05-22).

[9] BitcoinTalk. Official Website of BitcoinTalk. https://www.bitcointalk.org, 2009. (visited on 2020-05-03).

[10] Bitmix. Announcement Thread of Bitmix.biz on BitcoinTalk. https://bitcointalk.org/index.php?topic=2099519, 2017. (visited on 2020-05-02).

[11] Bitmix. Official Website of Bitmix.biz. https://bitmix.biz, 2017. (visited on 2020-05-02).

[12] Benjamin M Blau. Price dynamics and speculative trading in bitcoin. *Research in International Business and Finance*, 41:493–499, 2017.

[13] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A Kroll, and Edward W Felten. Mixcoin: Anonymity for bitcoin with accountable mixes. In *International Conference on Financial Cryptography and Data Security*, pages 486–504. Springer, 2014.

[14] Chainalysis. Official Portal of Chainalysis. https://www.chainalysis.com/, 2020. (visited on 2020-05-21).

[15] Changelly. Official Website of Changelly. http://changelly.com/, 2015. (visited on 2020-03-04).

[16] Chipmixer. Official Website of Chipmixer. https://chipmixer.com/, 2017. (visited on 2020-05-02).

[17] Nicolas Christin. Traveling the silk road: A measurement analysis of a large anonymous online marketplace. In *Proceedings of the 22nd international conference on World Wide Web*, pages 213–224, 2013.

[18] Clain. Binance Hack 2019. https://blog.clain.io/binance-hack-2019-\deep-dive-into-the-money-laundering/, 2019. (visited on 2020-02-18).

[19] Clain Team. Binance Hack 2019 – A Deep Dive Into Money Laundering And Mixing. https://blog.clain.io/binance-hack-2019-deep-dive-into-the-money-laundering/, 2020. (visited on 2020-08-07).

[20] CloakCoin Official Portal. CloakCoin. https://www.cloakcoin.com/, 2014. (visited on 2020-05-22).

[21] CoinMarketCap. Global Charts of CoinMarketCap. https://coinmarketcap.com/charts/, 2020. (visited on 2020-05-02).

[22] Thibault de Balthasar and Julio Hernandez-Castro. An analysis of bitcoin laundry services. In *Nordic Conference on Secure IT Systems*, pages 297–312. Springer, 2017.

[23] Europol. Multi-Millon Euro Cryptocurrency Laundering Service Best-Mixer.io Taken Down. https://www.europol.europa.eu/newsroom/news/multi-million-euro-cryptocurrency-laundering-service-bestmixerio-taken-down/, 2019. (visited on 2020-05-23).

[24] Flyp.me. Official Website of Flyp.me. https://flyp.me/en/, 2012. (visited on 2020-03-04).

[25] Rupert Hackett. BitMixer Shuts Down to “Make Bitcoin Ecosystem More Clean”. https://venturebeat.com/2017/07/25/bitmixer-shuts-down-to-make-bitcoin-ecosystem-more-clean/, 2017. (visited on 2020-05-23).

[26] Martin Harrigan and Christoph Fretter. The unreasonable effectiveness of address clustering. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, pages 368–373. IEEE, 2016.

[27] Ethan Heilman, Leen Alshenibr, Foteini Baldimtsi, Alessandra Scafuro, and Sharon Goldberg. Tumblebit: An untrusted bitcoin-compatible anonymous payment hub. In *NDSS*, 2017.

[28] Maged Hamada Ibrahim. Securecoin: A robust secure and efficient protocol for anonymous bitcoin ecosystem. *I. J. Network Security*, 19:295–312, 2017.

[29] Sesha Kethineni, Ying Cao, and Cassandra Dodge. Use of bitcoin in darknet markets: Examining facilitative factors on bitcoin-related crimes. *American Journal of Criminal Justice*, 43(2):141–157, 2018.

[30] Merve Can Kus Khalilov and Albert Levi. A survey on anonymity and privacy in bitcoinlike digital cash systems. *IEEE Communications Surveys and Tutorials*, 20(3):2543–2585, 2018.

[31] Larry Dean Harmon. Helix Shutdown Announcement Thread on Reddit.com. http://archive.fo/paKIO, 2017. (visited on 2020-05-23).

[32] Matthias Lischke and Benjamin Fabian. Analyzing the bitcoin network: The first four years. *Future Internet*, 8(1), 2016.

[33] Gregory Maxwell. CoinJoin: Bitcoin privacy for the real world. https://bitcointalk.org/index.php?topic=279249.0, 2013. (visited on 2020-02-18).

[34] Maxwell, Gregory. The first successful Zero-Knowledge Contingent Payment. `https://bitcoincore.org/en/2016/02/26/zero-knowledge-contingent-payments-announcement/`, 2016. (visited on 2020-05-22).

[35] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140, 2013.

[36] Malte Möser and Rainer Böhme. Anonymous alone? measuring bitcoin's second-generation anonymization techniques. In *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 32–41. IEEE, 2017.

[37] Malte Möser and Rainer Böhme. The price of anonymity: empirical evidence from a market for bitcoin anonymization. *Journal of Cybersecurity*, 3(2):127–135, 2017.

[38] Malte Möser, Rainer Böhme, and Dominic Breuker. An inquiry into money laundering tools in the bitcoin ecosystem. In *2013 APWG eCrime Researchers Summit*, pages 1–14. IEEE, 2013.

[39] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[40] Till Neudecker and Hannes Hartenstein. Could network information facilitate address clustering in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 155–169. Springer, 2017.

[41] Shen Noether. Ring signature confidential transactions for monero. Cryptology ePrint Archive, Report 2015/1098, 2015. `https://eprint.iacr.org/2015/1098`.

[42] Micha Ober, Stefan Katzenbeisser, and Kay Hamacher. Structure and anonymity of the bitcoin transaction graph. *Future Internet*, 5(2):237–250, 2013.

[43] Bitcoin Official. Choose Your Wallet. `https://bitcoin.org/en/choose-your-wallet?step=5&platform=windows`, 2017. (visited on 2020-05-02).

[44] Jaswant Pakki. *Everything You Ever Wanted to Know About Bitcoin Mixers (But Were Afraid to Ask)*. PhD thesis, Arizona State University, 2020.

[45] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity—a proposal for terminology. In *Designing privacy enhancing technologies*, pages 1–9. Springer, 2001.

[46] Fergal Reid and Martin Harrigan. An analysis of anonymity in the bitcoin system. In *Security and privacy in social networks*, pages 197–223. Springer, 2012.

[47] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*, pages 6–24. Springer, 2013.

[48] Dorit Ron and Adi Shamir. How did dread pirate roberts acquire and protect his bitcoin wealth? In *International Conference on Financial Cryptography and Data Security*, pages 3–15. Springer, 2014.

[49] Tim RuffingPedro and Moreno-SanchezAniket Kate. Coinshuffle: Practical decentralized coin mixing for bitcoin. In *European Symposium on Research in Computer Security (ESORICS)*, pages 345–364. Springer, 2014.

[50] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.

[51] ShapeShift. Official Website of ShapeShift. `https://www.shapeshift.io`, 2014. (visited on 2020-05-02).

[52] Shapeshift. `recenttx` API of Shapeshift. `http://shapeshift.io/recenttx/500`, 2020. (visited on 2020-05-02).

[53] Shapeshift. `txstat` API of Shapeshift. `https://shapeshift.io/txstat/[addr]`, 2020. (visited on 2020-05-02).

[54] Shifflett, Shane and Scheck, Justin. The Wall Street Journal: How Dirty Money Disappears Into the Black Hole of Cryptocurrency.
https://www.wsj.com/articles/how-dirty-money-disappears-into-the-black-hole-of-cryptocurrency-1538149743, 2019. (visited on 2020-02-18).

[55] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. Bitiodine: Extracting intelligence from the bitcoin network. In *International Conference on Financial Cryptography and Data Security*, pages 457–468. Springer, 2014.

[56] The Next Web. Chinese Bitcoin exchange Bter will pay back users after losing $1.75 million in cyberattack. https://thenextweb.com/insider/2015/03/12/chinese-bitcoin-exchange-bter-will-pay-back-users-after-losing-1-75-million-in-cyberattack/, 2015. (visited on 2020-02-18).

[57] Luke Valenta and Brendan Rowan. Blindcoin: Blinded, accountable mixes for bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 112–126. Springer, 2015.

[58] Wasabi Wallet. states API of Wasabi Wallet CoinJoin Service. https://wasabiwallet.io/api/v3/btc/chaumiancoinjoin/states, 2020. (visited on 2020-05-02).

[59] Wasabi Wallet. unconfirmed-coinjoins API of Wasabi Wallet CoinJoin Service. https://wasabiwallet.io/api/v3/btc/chaumiancoinjoin/unconfirmed-coinjoins, 2020. (visited on 2020-05-02).

[60] Yuriy Yanovich, Pavel Mischenko, and Aleksei Ostrovskiy. Shared send untangling in bitcoin. https://bitfury.com/content/downloads/bitfury_whitepaper_shared_send_untangling_in_bitcoin_8_24_2016.pdf, 2016. (visited on 2020-05-22).

[61] Haaroon Yousaf, George Kappos, and Sarah Meiklejohn. Tracing transactions across cryptocurrency ledgers. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 837–850, 2019.

[62] zkSNACKs. Official Website of Wasabi Wallet. https://wasabiwallet.io/, 2017. (visited on 2020-05-02).