Towards Understanding and Demystifying Bitcoin Mixing Services

Lei Wu
Zhejiang University

Yufeng Hu
Zhejiang University

Yajin Zhou*
Zhejiang University

Haoyu Wang
Beijing University of Posts and Telecommunications

Xiapu Luo
The Hong Kong Polytechnic University

Zhi Wang
Florida State University

Fan Zhang
Zhejiang University

Kui Ren
Zhejiang University

ABSTRACT

One reason for the popularity of Bitcoin is due to its anonymity. Although several heuristics have been used to break the anonymity, new approaches are proposed to enhance its anonymity at the same time. One of them is the mixing service. Unfortunately, mixing services have been abused to facilitate criminal activities, e.g., money laundering. As such, there is an urgent need to systematically understand Bitcoin mixing services.

In this paper, we take the first step to understand state-of-the-art Bitcoin mixing services. Specifically, we propose a generic abstraction model for mixing services and observe that there are two mixing mechanisms in the wild, i.e. swapping and obfuscating. Based on this model, we conduct a transaction-based analysis and successfully reveal the mixing mechanisms of four representative services. Besides, we propose a method to identify mixing transactions that leverage the obfuscating mechanism. The proposed approach is able to identify over 92% of the mixing transactions. Based on identified transactions, we then estimate the profit of mixing services and provide a case study of tracing the money flow of stolen Bitcoins.

CCS CONCEPTS

• **Security and privacy → Pseudonymity, anonymity and untraceability**; *Domain-specific security and privacy architectures.*

KEYWORDS

Bitcoin, Mixing Service, Anonymity, Pseudonymity

ACM Reference Format:
Lei Wu, Yufeng Hu, Yajin Zhou, Haoyu Wang, Xiapu Luo, Zhi Wang, Fan Zhang, and Kui Ren. 2021. Towards Understanding and Demystifying Bitcoin Mixing Services. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3442381.3449880

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

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449880

1 INTRODUCTION

Bitcoin [38] has become one of the representative cryptocurrencies. As of the first quarter in 2020, the total market capitalization of Bitcoin is over 117 billion US dollars [20]. In contrast to traditional payment channels, the decentralization essence of Bitcoin has three characteristics: 1) money can be transferred online directly without the intervention of any third-party banking services; and 2) transactions are verifiable and cannot be reversed; and 3) the *pseudonymity* makes the linkage between Bitcoin addresses and real-world entities hard. *Anonymity* is regarded as a key factor leading to Bitcoin's popularity [11].

However, the anonymity is *relationship anonymity* [44], 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 heuristics [25] or publicly available data sources [34]. Once address clusters are identified, the complete money flows between clusters (corresponding to different users) can be revealed. As a result, the anonymity is no longer preserved.

To improve the anonymity of Bitcoin, several approaches have been proposed. Some of them aim to hide the transaction information by modifying the Bitcoin protocol or building additional infrastructures. Such solutions include *Zerocash* [49] and *Monero* [40]. Others try to set up third-party services to provide enhanced anonymity without modifying the Bitcoin protocol, e.g., *Mixcoin* [12] and *Blindcoin* [56]. Corresponding to these approaches, many *altcoins* and *mixing services* emerged. Although altcoins can achieve stronger anonymity properties [40], the migration cost from Bitcoin to altcoins hinders the popularity of altcoins and makes the 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 [28], including ransomware like *WannaCry* [5], notorious underground markets like *Silk Road* [16] and *Ponzi* schemes [3]. Specifically, mixing services are extremely widely used in those activities to facilitate money laundering. For example, a previous study [16] showed that Silk Road extensively uses mixing services. It has also been reported [55] 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 the loss of around 7,074 BTC (about 40 million dollars at that time) [4]. Further investigation indicated that a large portion of stolen Bitcoins were sent to *Chipmixer* [17], a popular mixing service provider.

The extensive use of mixing services makes it difficult to trace suspicious money flow, 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. For example, the authors performed a simple graph analysis based on data collected from experiments of selected mixing services [37], while others focused on security issues of mixing services themselves [21]. In short, there lacks 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.

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 the same procedure but differ in the *mixing mechanisms*. Based on this abstraction model, we categorize state-of-the-art mixing mechanisms into two types, namely, *swapping* and *obfuscating*.

We then conduct an empirical study to analyze mixing services based on real Bitcoin transactions. To this end, four representative mixing services are selected and analyzed. Then we collect sample transactions for each service to analyze the mixing mechanisms. Finally, we propose a heuristic-based algorithm to identify the mixing transactions of the mixing services with the obfuscating mechanism.

Results. We apply the approach to analyze four representative Bitcoin mixing services, i.e., *Chipmixer* [15], *Wasabi Wallet* [62], *ShapeShift* [50], and *Bitmix.biz* [10].

For Chipmixer and Bitmix.biz, we interact with these services by sending Bitcoins to them 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 collected 4,850 mixing transactions from Wasabi Wallet, and 27,411 cryptocurrency convert records from ShapeShift.

Based on the collected sample transactions, we conduct a transaction-based analysis to first determine the mixing mechanism they used, and then reveal their workflow. Then, we perform an advanced analysis for services using the obfuscating mechanism to identify mixing transactions. The evaluation result demonstrates that the proposed algorithm is able to identify 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. In summary, this paper makes the following main contributions.

- We proposed an abstraction model and approach to systematically demystify state-of-the-art Bitcoin mixing services.
- We applied the proposed approach to four representative Bitcoin mixing services, and successfully revealed the mixing mechanisms and workflows of these services.
- We proposed an advanced analysis to effectively reveal mixing services that employ the obfuscating mechanism by identifying *most* (over 92%) mixing transactions. The evaluation results demonstrated the effectiveness of our approach.

To engage the community, the dataset of this study is released at the following link [1].

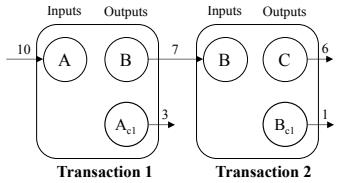


Figure 1: Example Bitcoin transactions.

2 BACKGROUND

2.1 Bitcoin

Bitcoin is a decentralized cryptocurrency proposed by an identity with pseudonym Satoshi Nakamoto [38]. The idea behind Bitcoin is a publicly available and verifiable distributed ledger. To protect the integrity of this public ledger, Bitcoin employs the Proof-of-Work (PoW) consensus algorithm.

Transaction. A transaction is a basic unit describing money flow from input addresses to output addresses. Every input is a reference to an unspent transaction output (UTXO) [59], which is an output in a previous transaction that has not been referenced in other transactions.

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

A transaction is to fully spend UTXOs specified in inputs, and distribute the remaining one to its output addresses with specified values. Note that in order to make this transaction verified and confirmed by the Bitcoin network, additional information that verifies the ownership of each UTXO and the integrity of the whole transaction is included in the transaction. Besides, to broadcast a transaction in the P2P network, users pay network fees to the miners who spend the computational resources to verify transactions.

Addresses. There are three types of *addresses* in Bitcoin. Addresses calculated directly from private keys (using hash functions) are called Pay-to-Public-Key-Hash (P2PKH) addresses. They begin with the number prefix 1. In 2012, a new type of address called

[1] https://github.com/blocksecteam/bitcoinmixing

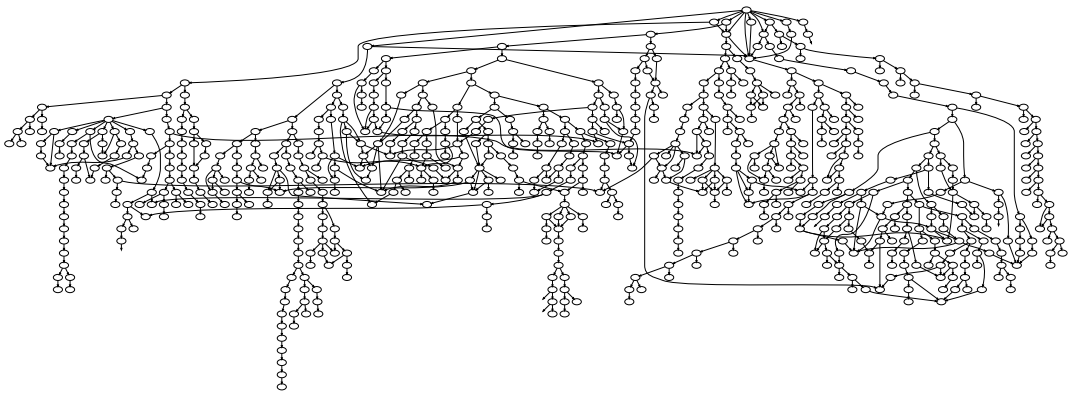


Figure 2: The simplified transaction graph for the Binance May Hack case. Nodes represent transactions. An edge means the money flow between transactions.

Pay-to-Script-Hash (P2PSH) was introduced to simplify the redeem script in transaction output for the multiple signature (MultiSig) protocol, and these addresses begin with the number prefix 3. In 2017, another new type was introduced in Bitcoin as the segregated witness (SegWit) to separate witness data (to verify the ownership of UTXOs) in transaction inputs. These addresses begin with the prefix bc1q.

2.2 Mixing Service

Originating from the Bitcoin community [1, 6, 7], the underlying idea for *mixing* is to obfuscate the relationship between inputs and outputs, thereby preserving 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 mixing services, such as Bitcoin Fog [6], are centralized mixing services. However, the centralized mixing service has the *trust* issue. First, there is no guarantee that the services providers will send the mixed coins to addresses specified by users. Second, they can record the *original* relationship between user inputs and outputs. Thus, if the services themselves are compromised, the anonymity will be broken. Mostly due to these reasons, many centralized mixing services disappeared in recent years, including BestMixer [22], Helix [30] and BitMixer [24].

Decentralized Mixing Service. The decentralized mixing service does not rely on a centralized server to perform the mixing. CoinJoin [32] is a generic decentralized mixing protocol proposed by Bitcoin Core developers [2]. The basic idea is to exploit the structure of transactions to combine different inputs and outputs in a single transaction, thus the recovery of the relationship between outputs and inputs is becoming harder. A number of works have been proposed on the basis of CoinJoin, including CoinShuffle [48] and SecureCoin [27].

Cross-Blockchain Mixing Service. There is also a special type of mixing services provided by cryptocurrency exchanges or converters (e.g., ShapeShift [50] [3], Changelly [14] and Flyp.me [23]). These services allow users to exchange Bitcoin with other cryptocurrencies, e.g., Zcash and Ether. Obviously, tracking the money flow across different ledgers is not trivial.

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* [37]. In this section, we propose an abstraction model by separating the mixing process into three steps, and illustrate the mixing mechanisms.

3.1 A Motivating Example

We use a real attack called the *Binance May Hack* [4], as the motivating example to demonstrate the difficulty to trace the money flow associated with mixing services. According to the official announcement of Binance [4], the attacker stole 7,074 BTC and withdrew them in one transaction [4]. The stolen Bitcoins were then distributed 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) represents the withdrawal transaction initiated by the attacker to transfer the stolen Bitcoins, and the subsequent graph shows the tainted money flow through multiple transactions.

Obviously, the graph in Figure 2 is too complicated to distinguish mixing transactions from others. To solve this issue, we propose a general abstraction model in this section and perform analysis on mixing services in Section 4.

3.2 Mixing in Three Phases

The process of a mixing service can be modeled as a three-phase procedure, i.e., *taking inputs*, *performing mixing* and *sending outputs*. Formally, a *Mixing Service* (denoted as $\mathcal{S}$) can be defined as a triplet: $(\mathcal{I}, \mathcal{O}, \mathcal{M})$, where $\mathcal{I}$ and $\mathcal{O}$ represent *inputs* and *outputs*, respectively, while $\mathcal{M}$ means the *mixing mechanism*.

[2] CoinJoin can be implemented in centralized mixing services as well [29].

[3] In this paper, we also study ShapeShift to understand its mixing mechanism. However, we only focus on mixing activities within the Bitcoin network.

[4] The transaction hash is e8b406091959700dbffcff30a60b190133721e5c39e89bb5fe23c5a554ab05ea, and we will use e8b406 to denote this transaction in the following.

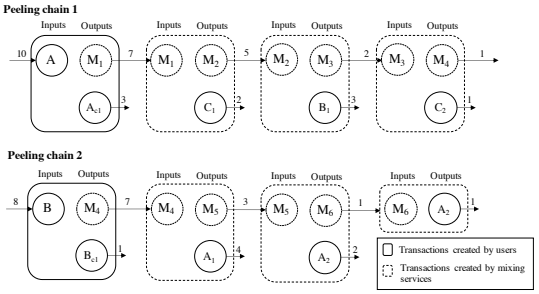


Figure 3: An example of the swapping mechanism. In this figure, we use M_1 to M_6 to denote addresses maintained by the mixing service. By swapping different user inputs and outputs, the relationship anonymity for all addresses is preserved. For instance, the relationship from A to A_1 and A_2 is anonymized.

Specifically, the mixing service $\mathcal{S}$ first takes Bitcoins to be mixed as the inputs ($\mathcal{I}$). This is achieved mostly by requiring users to send $\mathcal{I}$ to a service-provided deposit address. After taking $\mathcal{I}$, $\mathcal{S}$ is responsible for performing mixing with its mixing mechanism ($\mathcal{M}$), which consumes the collected user inputs, and prepares the desired outputs ($\mathcal{O}$) for each user. Finally, $\mathcal{S}$ will send $\mathcal{O}$ to the users. Typically, users specify some output addresses to $\mathcal{S}$ to indicate where the mixing output should be sent.

The procedure to handle $\mathcal{I}$ and $\mathcal{O}$ is similar in different mixing services. In the following, we will focus on different types of $\mathcal{M}$.

3.3 Mixing Mechanisms

The relationship anonymity is mainly achieved by mixing mechanisms. According to different implementations, they can be further categorized into two types, i.e., *swapping* and *obfuscating*.

To make it more clear, we first give the following definitions:

- $\mathcal{M}_S$, the *swapping mechanism*;
- $\mathcal{M}_O$, the *obfuscating mechanism*;
- $T_{\mathcal{M}}$, a *mixing transaction* [5];
- A_N, an *anonymity set* [6] with capacity N ($N \geq 2$), i.e., it has N outputs in a transaction with the same value.

3.3.1 Type I – $\mathcal{M}_S$. The basic idea of $\mathcal{M}_S$ is to swap the inputs and outputs from different users to preserve relationship anonymity. Note that in any $T_{\mathcal{M}}$ of $\mathcal{M}_S$, there is only one user output.

Figure 3 gives an example. 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. Similarly, B_1 is swapped from outputs of M_2, which originates from A.

Despite the simple and effective idea of swapping, there is an important assumption that $T_{\mathcal{M}}$s are hidden by the service. Otherwise if we can identify all $T_{\mathcal{M}}$s, the original relationships between inputs and outputs can be recovered. For instance, if we discover all $T_{\mathcal{M}}$s in Figure 3, then we can find out that the output value M_1 is equal to the input value M_4 of a mixing transaction. Consequently, we can infer that M_1 and M_4 are swapped and the original output of A is A_1 and A_2.

[5] In this paper, we call the transactions created by mixing services as *mixing transactions*.
[6] Groups of outputs with the same value are called *anonymity sets*.

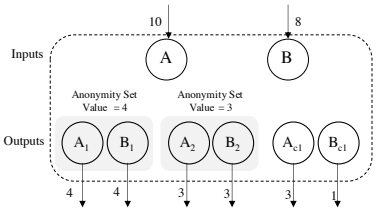


Figure 4: An example of obfuscating with a single mixing transaction. The mixing service generates two anonymity sets with the size 4 and 3, respectively.

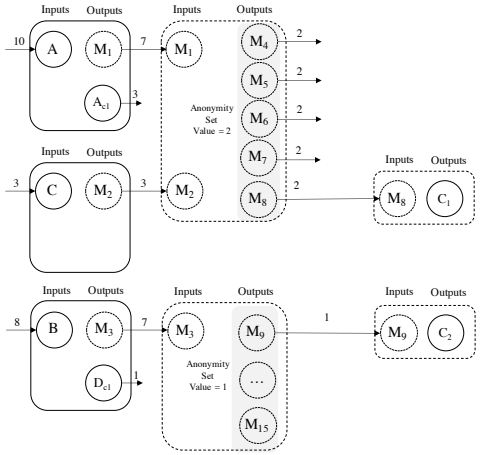


Figure 5: An example of obfuscating with multiple mixing transactions. The mixing service generates multiple anonymity sets with different values (2 and 1 in this case). From the figure, C_1 and C_2 are from M_8 and M_9 with input value 2 and 1 BTC. Other outputs, e.g., M_4 and M_{15} will be used to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

To prevent $T_{\mathcal{M}}$s from being identified, the concept of *peeling chain* was observed in the wild [37]. A peeling chain is a set of transactions generated by mixing services that form a chain to distribute outputs. The unique property of the peeling chain is that transactions in the chain are similar to normal user transactions with two outputs [25]. Thus, $T_{\mathcal{M}}$s cannot be easily distinguished from normal user transactions.

For a $T_{\mathcal{M}}$ in the peeling chain, one of the outputs is used to generate the output for the specified output 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 $T'_{\mathcal{M}}$. The peeling chain will be detailed in Section 5.3.

3.3.2 Type II – $\mathcal{M}_O$. $\mathcal{M}_O$ aims to preserve the relationship anonymity by breaking the matching procedure between user inputs and outputs. It is achieved by using anonymity sets to hide user outputs. Note that in any $T_{\mathcal{M}}$ of $\mathcal{M}_O$, there is at least one anonymity set A_N.

Figure 4 shows an example of a mixing transaction of $\mathcal{M}_O$. There are two A_Ns, and the outputs in each of them are indistinguishable.

For example, it is impossible to determine which output A_1 or B_1 in the first group originates from input A. It is hard to identify the real outputs for each input without additional information.

Moreover, the obfuscating procedure could be achieved within single or multiple $T_{\mathcal{M}}$s. Figure 4 and Figure 5 are two examples, respectively. Compared with a single $T_{\mathcal{M}}$, multiple $T_{\mathcal{M}}$s can generate fine-grained outputs with user-specified values. 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 the value for each A_N for each transaction in the case of the single $T_{\mathcal{M}}$. Note that in both cases, there are some transactions that involve multiple inputs and outputs with the same value.

4 METHODOLOGY

In this section, we will introduce our methodology to analyze mixing services. 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* [8] and other public media as the information sources. As the biggest Bitcoin-dedicated forum, BitcoinTalk has served as the official forum for Bitcoin. Besides, we also pay attention to reports from other public media reports. For example, ShapeShift was investigated and reported by the Wall Street Journal [53] for being used for money laundering.

4.2 Collect Sample Transactions

To analyze a mixing service, we first obtain the transactions that are used for the mixing purpose. We denote them as *sample transactions* in our study. For a typical mixing service, sample transactions include *input transactions* by users to send inputs to the service, and *output transactions* by the service to send mixed outputs to user specified output addresses. The input transactions are initiated by users, while the mixing and output transactions are initiated by the service.

Specifically, the following two complementary methods are used.

Method I: Interacting with Mixing Services. We use the mixing service by sending 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 high mixing fee. Therefore, we only conduct a small number of experiments using this method.

Method II: Using Public APIs. Some mixing services provide public APIs to facilitate their usage. For instance, they provide APIs for users to query detailed information and update status of a mix or inspect statistics of a mixing service. Fortunately, the returned data usually contains redundant information that help to reveal or reconstruct users' mixing records.

4.3 Basic Transaction Analysis

Based on sample transactions, our next step is to determine the mixing mechanism used by the service and understand its mixing process. This is achieved by performing a transaction-based analysis on sample transactions. There are two challenges in performing such analysis. In the following, we will discuss these challenges and our methods to solve them.

4.3.1 Challenges. We have to deal with the following two challenges in our analysis.

Challenge I: Identify Address Types. When constructing the transaction graph from sample transactions, we first need to distinguish the users' addresses and those addresses used by mixing services. Otherwise, our graph would be too big (with too many nodes and edges) 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 addresses used by the mixing services tend to belong to the same type (address types are introduced in Section 2.1).

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 addresses of the other type are considered to be used by users. We can then prune the transaction graph to remove users' addresses and transactions.

Challenge II: Identify Peeling Chains. Although peeling chains are commonly observed [21, 37], they have not been carefully analyzed. A peeling chain can be modeled as a structure consisting of three components, including *starting point*, *chain nodes* and *ending point*, which will be analyzed and distinguished accordingly.

Specifically, a *starting point* is the transaction that a user sends the input to an address given by the 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* [45], this transaction initiated by users should have only two outputs, one of which is the service provided deposit address and another is the change address. Secondly, the address type can also be used to distinguish the change output from the 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 (a reference to output from the previous node), and two outputs (one for user output and another for the successive node). However, there exist cases that chain nodes are indistinguishable from the starting point. In this case we trace backwards until a transaction with multiple inputs are found, and manually inspect them 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 other mixes. Our observation suggests that if the changes from a chain node is used in a transaction with many inputs, the corresponding chain node can be regarded as the ending point. Mixing services will collect these remaining changes for future use.

4.3.2 Determine Mixing Mechanisms. During the analysis, we define the *context* as the destinations of inputs and sources of outputs in sample transactions. 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 obfuscating mechanism comes from the

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 Q with all element in S;
Initialize E to be an empty set;
while Queue Q is not empty do
    Take a transaction T from 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 Q;
            end
        end
    end
end
```

output pattern. For the swapping mechanism, mixing outputs are consecutive, while the outputs are centralized for the obfuscating mechanism. We examine the context of outputs in sample transactions, and use the difference as a criteria to distinguish the mixing mechanism. We use the following heuristics in this study.

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

4.3.3 Understand Mixing Process. After the mixing mechanism is determined, we then figure out the mixing process, i.e., how the service performs the mixing.

Swapping Mechanism. For mixing services using the swapping mechanism, the peeling chain is the central structure used in a mixing process. In our study, we first draw the transaction graph and then identify the components of a peeling chain leveraging the previously discussed definitions in Section 4.3.1.

Obfuscating Mechanism. For mixing services using the obfuscating mechanism, we focus on transactions that generate anonymity sets. Specifically, for each input and output in the sample transaction, we find corresponding transactions that generate anonymity sets to spend the input or send the output.

4.4 Advanced Transaction Analysis

Besides the previous analysis, we also conduct further analysis to identify mixing transactions for services using the obfuscating mechanism. This is important as it helps to inspect money flow to mixing services and investigate money laundering activities. We take a two-step analysis with seed inputs.

Step I: Identify Anonymity Sets. Our first step is to identify anonymity sets using seed inputs, which are fed into the mixing service with service-provided addresses (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 do not color C_2 in the previous step since we only use the input A as the seed input. Input to B is not the seed input.

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

During this step, we may not find 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 is shown 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 methodology in Section 4 and summarize the results.

5.1 Selected Services

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

Chipmixer [15] is one of the most popular mixing services. Its popularity originates from its "Pay What You Want" (PWYW) pricing strategy. In addition, it was reported that Chipmixer was used by the attacker to launder over 4,000 BTC [17].

Wasabi Wallet [62] is one of the officially recommended desktop Bitcoin wallets [42], and the only (currently available and popular) wallet with the built-in CoinJoin functionality [62].

ShapeShift [50] is one of the most famous cryptocurrency converters. According to the report of the Wall Street Journal, it was used as a money laundering tool for over 9 million dollars of tainted funds over a time period of two years. Due to the pressure from the public media and regulators, it has applied Know-Your-Customer (KYC) policy and requires personal identification to set up an account.

Bitmix.biz [10] was announced in August, 2017 [9]. It claimed to have some improvements over its predecessors like dust-attack prevention, letters of guarantee (to redeem funds on exceptions), and randomized transaction fees and delays. The wider range of supported cryptocurrencies (Bitcoin, Litecoin and DASH) and lower mixing fee (from 0.4%) also contributes to its popularity.

Table 1: Sample transactions obtained for selected services.

Service	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)

5.2 Sample Transactions Collection

As introduced in Section 4.2, there are two complementary methods to obtain sample transactions. We first conduct a complete analysis on these services to determine which method to use. For Wasabi Wallet and ShapeShift, we find public APIs that can be used to obtain sample transactions. In contrast, for Chipmixer and Bitmix.biz we resort to 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. We performed the collection from October, 2019 to February, 2020.

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

This service first provides a generated address for users to send inputs. When an input is confirmed, the next step is to decide how to distribute the input into *chips* [7]. After the distribution of chips, users can withdraw these chips by either importing the provided 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. Parameters include output addresses, the delay from the mixing request to output received, value distribution (distributions for each address) and overall transaction fees. After receiving a request, the service will provide a temporary address to receive inputs. Once the inputs are confirmed, it will send corresponding outputs according to the requested delay.

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 public APIs used to obtain sample transactions.

Wasabi Wallet. It provides two APIs to fetch mixing-related data: 1) the API `states` [57] is used for the clients to query and update current phase and status of current CoinJoin transaction; and 2) the API `unconfirmed` [58] broadcasts transaction hashes of all successful CoinJoin transactions before they are confirmed.

These two API are for status querying and updating purposes. However, the Wasabi Wallet server does not require any authentication to access them. Therefore, we used a crawler to periodically retrieve information. 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. We will use these transactions as the seed set for our experiment.

[7] Chips are defined as user outputs with predefined values [15].

Figure 6: The transaction graph of the Chipmixer experiment. Gray and black nodes are our input and output transactions, respectively. Transactions in light gray nodes generate anonymity sets.

ShapeShift. There are two key APIs that can be used to obtain sample transactions. The first API is called `recenttx` [51]. It provides information about all recent convert records in ShapeShift. Each convert record is represented by a tuple of `<curIn, curOut, timestamp, value>`, which represents the cryptocurrency type of input and output, timestamp of the convert, and input currency value in decimal. The second API is called `txstat` [52]. For a given address, it provides detailed information if the address is used by ShapeShift. While ShapeShift requires a registered account and personal identification information, using these APIs requires no authentication.

In total, we crawled 27, 411 convert records from December 11, 2019 to March 18, 2020. We focused on converting 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 propose 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 and closest timestamps to ts. Finally, these transactions will be further validated by applying the `txstat` API. We have applied this algorithm on crawled 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 output samples to analyze the complete convert workflow. Besides, we want to analyze where output Bitcoin comes from in the case of converting other cryptocurrencies to Bitcoin. To this end, we chose 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, it uses the obfuscating mechanism. Apart from the Wasabi Wallet, the mixing mechanisms used by other three services are determined by analyzed the transaction graph of the obtained transactions.

Table 2: Mixing mechanisms used by services.

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

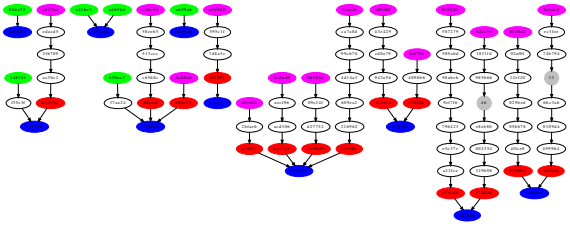


Figure 7: The transaction graph of Bitmix.biz experiments. Green nodes represent our input and output transactions. Blue nodes are ending points, and magenta nodes are potential starting points of the peeling chains. Gray circles with numbers denote omitted long chains. User output in each chain node is also omitted.

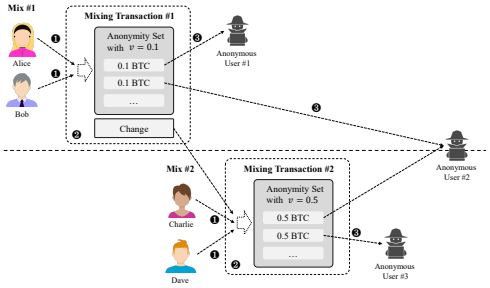
To determine the mixing mechanism used by Chipmixer, we first plot a transaction graph of sample transactions and their contexts. Figure 6 is the transaction graph for the experiments conducted with Chipmixer. The figure shows that all of our inputs (gray nodes) are immediately spent by mixing transactions (light gray nodes) by the service, and our outputs (black nodes) also come directly from them. Mixing transactions in light gray nodes generate anonymity sets, indicating 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 they use the swapping mechanisms. For example, in Figure 7, all of our outputs come from mixing transactions with only two outputs (i.e., no anonymity sets get involved). Tracing our outputs backward shows several chains, in which most transactions have single input and two outputs. They are connected with change addresses. Again, according to Section 3, this is a feature of the peeling chain. Results of all these services are summarized in Table 2.

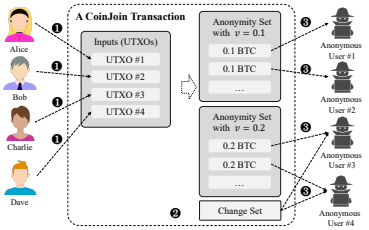
5.3.2 Understanding Mixing Process. To better understand the mixing services, we need to figure out their mixing workflows.

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

Figure 8(a) 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. For example, the inputs of mix #2 come from



(a) Chipmixer.



(b) Wasabi Wallet.

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

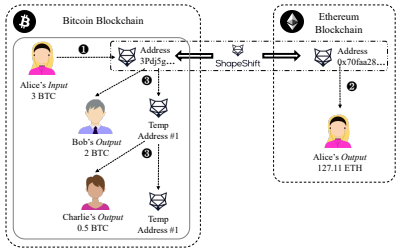
another two users (Charlie and Dave) along with the change of mix #1. The anonymity set generated by mix #2 have a value of 0.5, which fits the inputs without any change.

Wasabi Wallet. Unlike other services that create addresses for users to deposit Bitcoin, this service requires users to send UTXOs and output addresses in the wallet. 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.

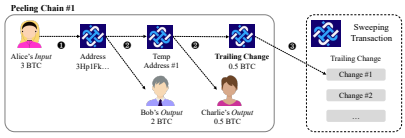
Figure 8(b) gives an example. In step 1, users of this CoinJoin round (i.e. this mix), Alice, Bob, Charlie and Dave, submit UTXOs they want to mix and output addresses to the service. Then in step 2, two anonymity sets with value 0.1 and 0.2 are generated. Finally in step 3, outputs in anonymity sets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

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

Figure 9(a) gives a concrete example. In the Bitcoin network, Alice sends 3 BTC to ShapeShift and receives 127.11 Ether in Ethereum later. Obviously, this service has to make efforts (e.g. in collaboration with cryptocurrency exchanges) to break even among different blockchain platforms. Due to the swapping mechanism, 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.



(a) ShapeShift.



(b) Bitmix.biz.

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

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 the peeling chain.

An example of a peeling chain for Bitmix.biz is shown in Figure 9(b). 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 the ending point of this chain. This transaction consumes remaining changes from multiple peeling chains and merges them into a large balance for further use.

5.4 Advanced Transaction Analysis

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

In the following, we first evaluate the effectiveness of the proposed Algorithm 1. Due to the 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 Bitcoin based on identified mixing transactions using our proposed algorithm.

5.4.1 Measuring the Effectiveness of the Algorithm. Due to the lack of ground truth, we manually investigated our own ground truth to support the measurement. Specifically, for each service, we first collected transactions according to the common features we observed from the sample transactions and then filtered false positives manually. Then we were able to evaluate the robustness of the proposed algorithm by comparing the result with the ground truth.

Table 3: 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%**			

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 mixing transactions identified in experiments as the seed set, we found 8, 279 transactions potentially generated by Chipmixer.
- *Experiment 2.* We performed the second experiment at block height 619, 700, and found additional 1, 056 transactions (9, 335 in total) mixing transactions from Chipmixer.
- *Experiment 3.* We conducted the third experiment at the same block height with experiment 2. The seed set was randomly chosen from the 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 the original seed set. Again, we achieved the same expansion set as in experiment 2.

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

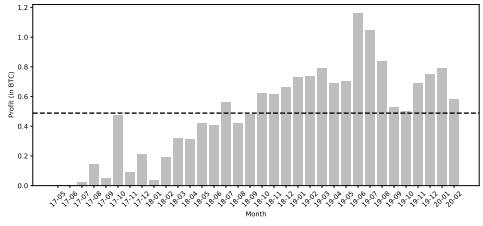
5.4.2 Calculating Profit of Mixing Services. For each service, we will calculate the profit based on identified mixing transactions.

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 [8].

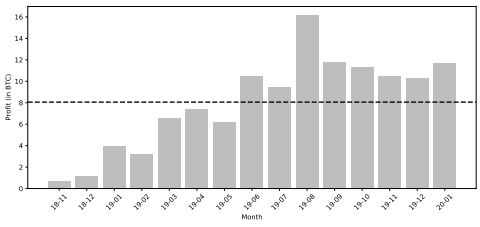
To calculate the profit earned by Chipmixer, we sum over all trailing changes of user inputs from May, 2017 to February, 2020. In total, Chipmixer received 16.6086 BTC as service fees (with monthly average value 0.4883 BTC), which was considerably less than the total user inputs 53, 044.8077 BTC during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

Wasabi Wallet. We present the analysis based on the 9, 788 transactions obtained using the 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 transaction identified, we found two common output addresses

[8] E.g., 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.



(a) Chipmixer.



(b) Wasabi Wallet.

Figure 10: Monthly profit for Chipmixer and Wasabi Wallet. The dash line represents the average value.

potentially for fee collection. Address 1[9] has been used in 5,319 CoinJoin transactions, but is no longer active since September 20, 2019. Address 2[10] 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 10(b) illustrates monthly profit earned by Wasabi Wallet. In total, these addresses collected 120.9932 BTC (with 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 help to reveal money laundering by tracing the money flow of stolen Bitcoins.

Specifically, we provide a simple case study for the Binance May Hack case [4]. 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 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 sends Bitcoin to Chipmixer. To solve the problem of dimension explosion, we set the maximum depth of tracing to 50 and ignore outputs less than 0.9 BTC.

In total, we found 157 transactions in identified transactions of Chipmixer, for a total value of 4,797.82 BTC [11]. Figure 11 is a *simplified* transaction graph to illustrate the case, where nodes are transactions. The blue nodes indicate transactions sending the tainted funds to Chipmixer, while the gray ones mean that their addresses are `bc1q` addresses, which are coherent with the original outputs in transaction `e8b406`. Without the proposed approach, obviously, it may require a lot of human efforts to investigate the provenance of the stolen Bitcoins.

[9] Address: `bc1qs604c7jv6amk4cxqlnvuxv26hv3e48cds4m0ew`

[10] Address: `bc1qa24tsgchvuxsaccp8vrnkfd85hrcpafg20kmjw`

[11] As a reference, an industry report [18] 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 the algorithm in [61]. It has a threshold parameter that limits the number of blocks to be examined. For the original algorithm, it is represented by two parameters (δ_a and δ_b), which are 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 it leads to poor performance (80.29% 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). Obviously, our refinement leads to a much better performance.

Traceability Beyond Mixing Services. Our approach only traces Bitcoins that are sent to mixing services. Tracing Bitcoins beyond mixing services is much more complicated because the money laundering may involve some off-chain activities (e.g., Over-The-Counter transactions) which cannot be traced through the on-chain information. However, our approach is still meaningful to serve many research and practical purposes (e.g., assisting criminal investigation involved with Bitcoins in Section 5.4.2).

The Scope and Completeness of Our Study. The scope of mixing services is limited due to the complexity of the ecosystem. In this paper, we only consider traditional mixing services that fully rely on on-chain mechanisms to operate, without additional protocols. Other real-world mixing protocols like Fair Exchange and CoinSwap (investigated in [35]), have much less popularity than traditional ones. Besides, there exist complex research mixing protocols like Blindcoin [56] and Mixcoin [12]. However, to the best of our knowledge, they have no real-world deployments.

Other Limitations of Our Work. Our work has several limitations. First, the advanced transaction analysis does not cover mixing services using the swapping mechanism. The design of peeling chains (see 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 transactions. Indeed, it is a technical challenge.

Another limitation arises from the two-step approach to identify anonymity sets in the advanced transaction analysis (Section 4.4). If any output generated by a mixing transaction is incidentally not used as part of any transaction's inputs [12], then our approach could not find this mixing transaction. Besides, it also relies on the size 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 reducing the possibility of being identified.

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 may have some bias that affects the effectiveness of the measurement.

[12] 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.

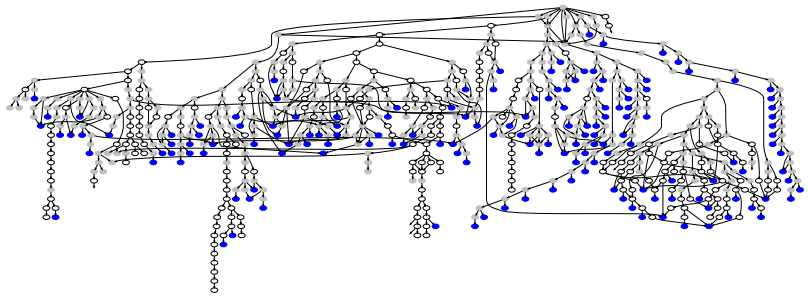


Figure 11: The simplified transaction graph for the Binance May Hack case. Transactions related to Chipmixer are annotated by blue nodes. In total, the attacker sent 4,792 BTC to Chipmixer.

7 RELATED WORK

Bitcoin Mixing Service. The basic idea of mixing is to preserve relationship anonymity by obfuscating the relations from senders to recipients. Several mixing services have been publicly announced since 2010, including *BitLaundry* [7], *Bitcoin Laundry* [1] and *Bitcoin Fog* [6]. In 2013, Maxwell made the idea of *CoinJoin* public to the community [32]. In 2014, *Mixcoin* [12] was proposed as the first academic work of mixing. Since then, a number of mixing approaches have been proposed, including *Fair Exchange Protocol* [26] and *Zero Knowledge Proof* [33], 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* [6], *Mixcoin* [12] and *Blindcoin* [56]) and decentralized (e.g., *CoinJoin* [32], *CoinShuffle* [48] and *CloakCoin* [19]). The centralized mixing services rely on central mixing servers to perform mixing, while decentralized mixing services allow users to perform mixing without any centralized mixing server. There are also centralized mixing services using decentralized protocols (like Wasabi Wallet using CoinJoin). Besides, mixing services like ShapeShift [50] allow mixing across different blockchains.

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. [37] 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. [21] applied the tool provided by Chainalysis [13] 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. [36] analysed the online *CoinJoin* market named *JoinMarket* and estimated its market volume. Jaswant Pakki [43] provides a more recent survey on mixing services in Bitcoin, in which the author provides a table of mixing services with 9 trusted services. Unlike these previous studies, we propose a generic model 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 [29] by either identifying the relations between Bitcoin addresses and user information, or clustering and labeling Bitcoin addresses. Our work is closed to those that mainly focused on Bitcoin addresses by analyzing blockchain data. Reid et al. [45] 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 [29]. Since then, several assumptions and methods were proposed and some of them have been used together to cluster Bitcoin addresses, including the multi-input heuristic [2, 31, 34, 41, 45, 46, 54], change addresses [2, 34, 39, 54] and behavior-based clustering [2, 47]. Although mixing services are rarely considered by these works, their methods and findings (e.g., the multi-input heuristic) form the basis of our work.

8 CONCLUSION

In this work, we aim to understand Bitcoin mixing services. Accordingly, we first categorize mixing services into two types based on mixing mechanisms, i.e., swapping and obfuscating. Then we propose a transaction analysis method to identify mixing mechanisms and workflows of these services. Lastly, we propose a heuristic-based algorithm to identify mixing transactions.

We then apply the proposed approach to four representative mixing services. The evaluation results demonstrate the effectiveness of our approach. Specifically, we successfully determine the mixing mechanisms of each service. We also show that it is able to identify most (over 92%) of the mixing transactions by applying the proposed algorithm. We finally provide two case studies, including calculating the profit and investigating the money laundering activity, to show the usage scenarios of our study.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments. that helped improve the presentation of this paper. This work was partially supported by the Fundamental Research Funds for the Central Universities (No. 2020QNA5019, 2019QNA5016), Leading Innovative and Entrepreneur Team Introduction Program of Zhejiang (No. 2018R01005), the National Natural Science Foundation of China (grant No.62072046), and Hong Kong RGC Projects (No. 152193/19E, 152223/20E). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of funding agencies.

REFERENCES

[1] Akemashite Omedetou. 2011. Bitcoin Laundry. https://en.bitcoin.it/wiki/Bitcoin_Laundry.

[2] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. 2013. Evaluating user privacy in bitcoin. In *Proceedings of the International Conference on Financial Cryptography and Data Security*.

[3] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. 2018. Data mining for detecting Bitcoin Ponzi schemes. In *Proceedings of the Crypto Valley Conference on Blockchain Technology*.

[4] Binance. 2019. Binance Security Breach Update. https://www.binance.com/en/support/articles/360028031711.

[5] Stefano Bistarelli, Matteo Parroccini, and Francesco Santini. 2018. Visualizing Bitcoin Flows of Ransomware: WannaCry One Week Later. In *Proceedings of the Second Italian Conference on Cyber Security*.

[6] Bitcoin Wiki. 2011. Bitcoin Laundry. https://bitcointalk.org/index.php?topic=50037.

[7] Bitcoin Wiki. 2011. BitLaundry. https://en.bitcoin.it/wiki/BitLaundry.

[8] BitcoinTalk. 2009. Official Website of BitcoinTalk. https://www.bitcointalk.org.

[9] Bitmix. 2017. Announcement Thread of Bitmix.biz on BitcoinTalk. https://bitcointalk.org/index.php?topic=2099519.

[10] Bitmix. 2017. Official Website of Bitmix.biz. https://bitmix.biz.

[11] Benjamin M. Blau. 2017. Price dynamics and speculative trading in bitcoin. *Research in International Business and Finance* (2017).

[12] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A Kroll, and Edward W Felten. 2014. Mixcoin: Anonymity for bitcoin with accountable mixes. In *Proceedings of the International Conference on Financial Cryptography and Data Security*.

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

[14] Changelly. 2015. Official Website of Changelly. http://changelly.com/.

[15] Chipmixer. 2017. Official Website of Chipmixer. https://chipmixer.com/.

[16] Nicolas Christin. 2013. Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace. In *Proceedings of the 22nd International Conference on World Wide Web*.

[17] Clain. 2019. Binance Hack 2019. https://blog.clain.io/binance-hack-2019-\deep-dive-into-the-money-laundering/.

[18] Clain Team. 2020. Binance Hack 2019 – A Deep Dive Into Money Laundering And Mixing. https://blog.clain.io/binance-hack-2019-deep-dive-into-the-money-laundering/.

[19] CloakCoin Official Portal. 2014. CloakCoin. https://www.cloakcoin.com/.

[20] CoinMarketCap. 2020. Global Charts of CoinMarketCap. https://coinmarketcap.com/charts/.

[21] Thibault de Balthasar and Julio Hernandez-Castro. 2017. An analysis of bitcoin laundry services. In *Proceedings of the Nordic Conference on Secure IT Systems*.

[22] Europol. 2019. Multi-Millon Euro Cryptocurrency Laundering Service BestMixer.io Taken Down. https://www.europol.europa.eu/newsroom/news/multi-million-euro-cryptocurrency-laundering-service-bestmixerio-taken-down/.

[23] Flyp.me. 2012. Official Website of Flyp.me. https://flyp.me/en/.

[24] Rupert Hackett. 2017. BitMixer Shuts Down to "Make Bitcoin Ecosystem More Clean". https://venturebeat.com/2017/07/25/bitmixer-shuts-down-to-make-bitcoin-ecosystem-more-clean/.

[25] Martin Harrigan and Christoph Fretter. 2016. The unreasonable effectiveness of address clustering. In *Proceedings of the International 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)*.

[26] Ethan Heilman, Leen Alshenibr, Foteini Baldimtsi, Alessandra Scafuro, and Sharon Goldberg. 2017. TumbleBit: An Untrusted Bitcoin-Compatible Anonymous Payment Hub. In *Proceedings of the 24th Annual Network and Distributed System Security Symposium*.

[27] Maged Hamada Ibrahim. 2017. SecureCoin: A Robust Secure and Efficient Protocol for Anonymous Bitcoin Ecosystem. *International Journal of Network Security* (2017).

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

[29] Merve Can Kus Khalilov and Albert Levi. 2018. A Survey on Anonymity and Privacy in Bitcoinlike Digital Cash Systems. *Proceedings of the IEEE Communications Surveys and Tutorials*.

[30] Larry Dean Harmon. 2017. Helix Shutdown Announcement Thread on Reddit.com. http://archive.fo/paKIO.

[31] Matthias Lischke and Benjamin Fabian. 2016. Analyzing the Bitcoin network: The First Four years. *Future Internet* (2016).

[32] Gregory Maxwell. 2013. CoinJoin: Bitcoin privacy for the real world. https://bitcointalk.org/index.php?topic=279249.0.

[33] Maxwell, Gregory. 2016. The first successful Zero-Knowledge Contingent Payment. https://bitcoincore.org/en/2016/02/26/zero-knowledge-contingent-payments-announcement/.

[34] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. 2013. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the Conference on Internet Measurement Conference*.

[35] Malte Möser and Rainer Böhme. 2017. Anonymous alone? measuring Bitcoin's second-generation anonymization techniques. In *Proceedings of the 2017 IEEE European Symposium on Security and Privacy Workshops*.

[36] Malte Möser and Rainer Böhme. 2017. The price of anonymity: empirical evidence from a market for Bitcoin anonymization. *Journal of Cybersecurity* (2017).

[37] Malte Möser, Rainer Böhme, and Dominic Breuker. 2013. An inquiry into money laundering tools in the Bitcoin ecosystem. In *Proceedings of the 2013 APWG eCrime Researchers Summit*.

[38] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.

[39] Till Neudecker and Hannes Hartenstein. 2017. Could Network Information Facilitate Address Clustering in Bitcoin. In *Proceedings of the International Conference on Financial Cryptography and Data Security*.

[40] Shen Noether. 2015. Ring Signature Confidential Transactions for Monero. https://eprint.iacr.org/2015/1098.

[41] Micha Ober, Stefan Katzenbeisser, and Kay Hamacher. 2013. Structure and anonymity of the bitcoin transaction graph. *Future Internet* (2013).

[42] Bitcoin Official. 2017. Choose Your Wallet. https://bitcoin.org/en/choose-your-wallet?step=5&platform=windows.

[43] Jaswant Pakki. 2020. *Everything You Ever Wanted to Know About Bitcoin Mixers (But Were Afraid to Ask)*. Ph.D. Dissertation. Arizona State University.

[44] Andreas Pfitzmann and Marit Köhntopp. 2001. Anonymity, unobservability, and pseudonymity—a proposal for terminology. In *Proceedings of the International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*.

[45] Fergal Reid and Martin Harrigan. 2011. An analysis of anonymity in the bitcoin system. In *Proceedings of the 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*.

[46] Dorit Ron and Adi Shamir. 2013. Quantitative analysis of the full bitcoin transaction graph. In *Proceedings of the International Conference on Financial Cryptography and Data Security*.

[47] Dorit Ron and Adi Shamir. 2014. How Did Dread Pirate Roberts Acquire and Protect his Bitcoin Wealth?. In *Proceedings of the International Conference on Financial Cryptography and Data Security*.

[48] Tim RuffingPedro and Moreno-SanchezAniket Kate. 2014. CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin. In *Proceedings of the 19th European Symposium on Research in Computer Security*.

[49] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. 2014. Zerocash: Decentralized anonymous payments from bitcoin. In *Proceedings of the 2014 IEEE Symposium on Security and Privacy*.

[50] ShapeShift. 2014. Official Website of ShapeShift. https://www.shapeshift.io.

[51] ShapeShift. 2020. `recenttx` API of Shapeshift. http://shapeshift.io/recenttx/500.

[52] ShapeShift. 2020. `txstat` API of Shapeshift. https://shapeshift.io/txstat/[addr].

[53] Shifflett, Shane and Scheck, Justin. 2019. 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.

[54] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. 2014. BitIodine: Extracting intelligence from the Bitcoin network. In *Proceedings of the International Conference on Financial Cryptography and Data Security*.

[55] The Next Web. 2015. 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/.

[56] Luke Valenta and Brendan Rowan. 2015. Blindcoin: Blinded, accountable mixes for bitcoin. In *Proceedings of the International Conference on Financial Cryptography and Data Security*.

[57] Wasabi Wallet. 2020. `states` API of Wasabi Wallet CoinJoin Service. https://wasabiwallet.io/api/v3/btc/chaumiancoinjoin/states.

[58] Wasabi Wallet. 2020. `unconfirmed-coinjoins` API of Wasabi Wallet CoinJoin Service. https://wasabiwallet.io/api/v3/btc/chaumiancoinjoin/unconfirmed-coinjoins.

[59] Wikipedia. 2020. Unspent transaction output. https://en.wikipedia.org/wiki/Unspent_transaction_output.

[60] Yuriy Yanovich, Pavel Mischenko, and Aleksei Ostrovskiy. 2016. Shared Send Untangling in Bitcoin. https://bitfury.com/content/downloads/bitfury_whitepaper_shared_send_untangling_in_bitcoin_8_24_2016.pdf.

[61] Haaroon Yousaf, George Kappos, and Sarah Meiklejohn. 2019. Tracing transactions across cryptocurrency ledgers. In *Proceedings of the 28th USENIX Security Symposium*.

[62] zkSNACKs. 2017. Official Website of Wasabi Wallet. https://wasabiwallet.io/.