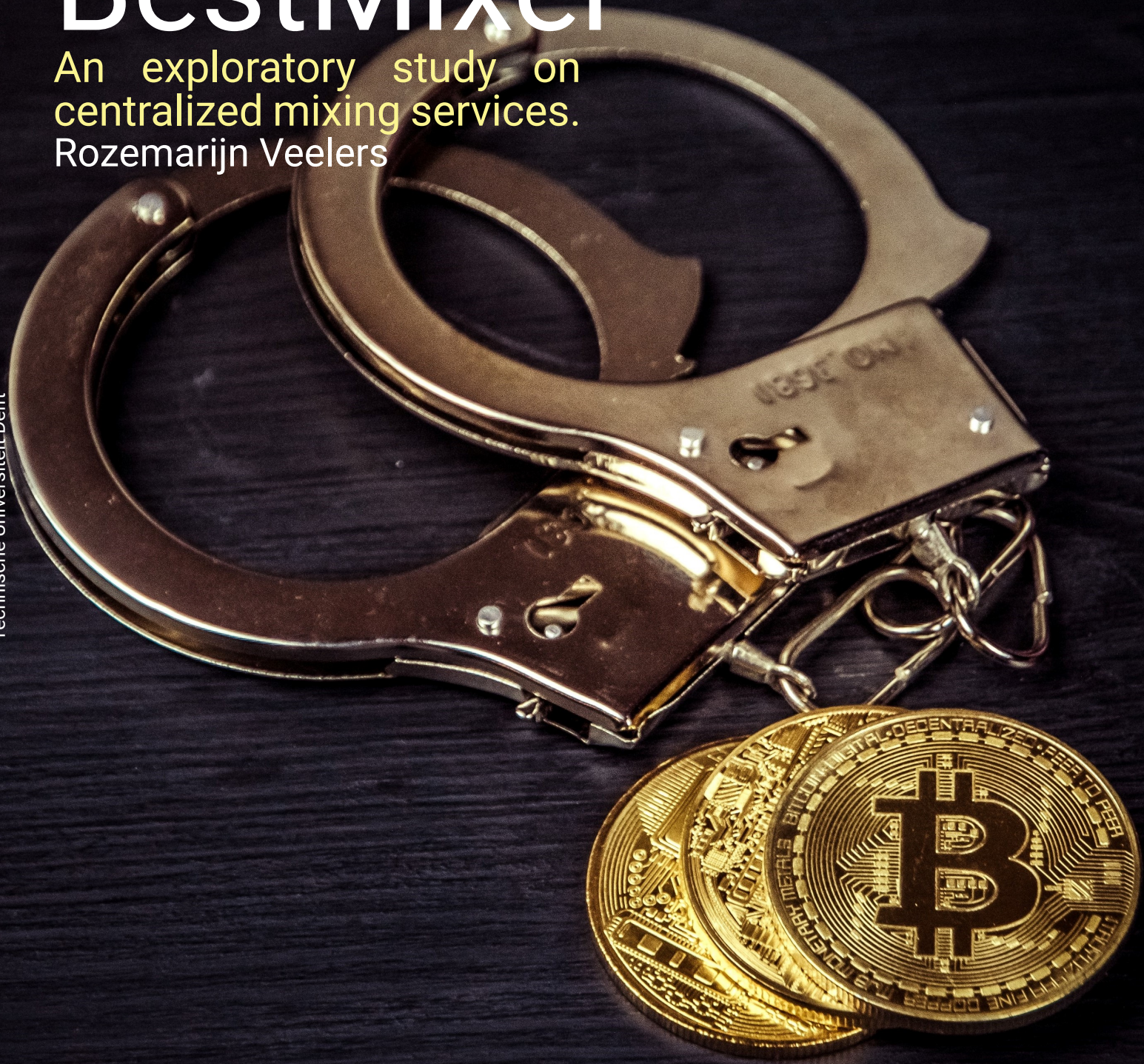
Demixing BestMixer

An exploratory study on centralized mixing services.

Rozemarijn Veelers



Technische Universiteit Delft

TU Delft

FACT Financial Advanced Cyber Team

Demixing BestMixer

An exploratory study on centralized mixing services.

by

Rozemarijn Veelers

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday March 23, 2021 at 13:00 AM.

Student number: 4350189
Project duration: May 1, 2021 – March 23, 2022
Thesis committee:

Prof. dr. P. H. Hartel,	TU Delft, chair
Dr. R. S. van Wegberg,	TU Delft, daily supervisor
Prof. dr. M. J. G. van Eeten,	TU Delft
K. J. M. Lubbertsen MSc.,	FIOD, external advisor

TU Delft

Abstract

Mixing services try to distort cash flow tracking of cryptocurrencies and obfuscate the origin of the customers' earnings by substituting customers' cryptocurrency funds with the funds of other customers or the mixers' private assets. This quality makes mixing services interesting for money laundering, and they are therefore often used by criminals. As such, there is an urgent need to systematically understand how to restore the relationship between deposits and payouts of centralized mixing services. Unfortunately, there is minimal knowledge of how mixing processes of centralized mixing services work, and few attempts exist to create these demixing methods. This research aimed to develop a demixing method for centralized mixers with knowledge gained from ground-truth data. The ground-truth data contains information on orders and the transaction history of mixing service BestMixer. Demixing consists of collecting all addresses that are part of the mixer (attribution) and finding the correct payout to a deposit (reconstruction).

Multiple statistical analysis techniques were applied to this data to verify existing attribution heuristics and find new characteristics of mixing services. Also, filtering techniques to reconstruct the relation between deposits and payouts were tested on the order data.

This research verifies that BestMixer likely did not reuse addresses in the mixing process. It also showed that the lifespan of most addresses was shorter than 24 hours. In addition, many BestMixer addresses received or sent a transaction to another BestMixer address, which created sequences of BestMixer transactions. The sequences show that the mixer used a peeling chain pattern in combination with multi-input transactions. These characteristics can be used to attribute other centralized mixing services. The results also show that the mixer increased in popularity throughout time.

Overall, the reconstruction attempt with filtering techniques did not perform well on BestMixer orders, as it returned an impracticable amount of possible payout combinations. The mixer showed less activity in the beginning days of the service, and there are signs that the reconstruction works better in this earlier stage of the mixer. This means that when a mixer becomes more popular, it could become more difficult to demix the orders correctly.

From this research can be concluded that the ground-truth data of BestMixer does help in developing attribution heuristics for centralized mixers, but not in developing a general reconstruction method that correctly restores the relation between deposits and payouts, thus not suffice in demixing centralized mixers.

Preface

This thesis is the final project I have to complete before I receive my MSc degree in Computer Science, and it also means that my time as a student at TU Delft is drawing to an end. The last seven and a half years have been amazing, and I am proud of the things I have accomplished. I started as a student at the faculty of Technology, Policy and Management, where I made many good friends and where my interest in all sorts of engineering was fueled. However, after a few years, I realized that even though my interests in most socio-technical topics were there, I missed having skills that could help solve some of the problems in these environments. Therefore, I decided to pursue a degree in Computer Science, which was very challenging initially, but nevertheless, I was able to finish in 2 years. This thesis shows how much I have learnt (and how much I still have to learn).

I would like to extend my gratitude to all the people that helped me through the process of finishing this thesis. First, I would like to thank Rolf for introducing me to the world of cybercrime, for all of his guidance during our weekly meetings, and for putting a brake on my overenthusiastic unrealistic planning. Second, I would like to thank Kelvin for his supervision and for sharing his extensive knowledge of everything. Also, if I would get 10 cents for every time you said 'Komt goed' or 'Rustig aan', I would be a millionaire by now. Furthermore, I would like to thank Michel and Pieter for being part of my committee and evaluating my work.

I would not have been able to write this thesis without the collaboration between TU Delft and the Financial Advanced Cybercrime Team. Not only did they share the unique data that forms the foundation of this thesis, they also let me be part of their team, which was a lot of fun and very helpful, for which I am grateful.

Lastly, I would like to thank Tim, my family and friends for their patience during some of my tantrums, always listening and trying to understand and improve my writing.

Rozemarijn Veelers
Delft, March 2022

Contents

List of Figures

List of Tables

1 Introduction

With the publication of Nakamoto's paper on Bitcoin, we entered the era of cryptocurrencies. Digital currencies are a cheap and seemingly fraud-resistant alternative for fiat money. The currencies make use of blockchains, which function as digital ledgers stored on a distributed network of computer systems. The decentralized ledgers contain all information on transactions between wallet addresses. New transaction requests are combined into a block that are created by *miners*. These miners are rewarded for their service with newly created coins. Since mining is an exhaustive computational task, most people buy coins from cryptocurrency exchange services instead of generating a new block.

Caused by the existence of cryptocurrencies, there is no need for a mediating third party, which prevents fraud from banks & central authorities, and the high costs of owning a bank account. In contrast to a bank account, addresses on a decentralized ledger do not reveal the owner's identity. Banks have to comply with laws and regulations, whilst miners and exchange services enjoy more freedom.

Making transactions without showing your identity is very beneficial for people that value their privacy, and that is exactly why criminals are eager to use cryptocurrencies. Approximately 0.34% of transactions made in 2020 can be linked to illicit activities, and this percentage is likely to grow (Chainalysis, 2021). Many dark markets accept digital coins as payment: transactions occur between wallet addresses from the dark market and the purchasers of illegal goods. This is also why ransomware attackers often want to receive the ransom in cryptocurrency; the address is not directly linked to their real identity.

However, some cryptocurrencies - like Bitcoin - do not provide complete anonymity. Once an address is connected to a person, all previous transactions to that address can be extracted from the ledger. Androulaki et al. (2013) showed that they could profile 40% of the users in their simulation. Because of this privacy implication, we say Bitcoin is pseudonymous instead of anonymous. Even though Bitcoin is pseudonymous, criminals still use it for their illicit activities because *mixing services* or *tumblers* help users launder cryptos anonymously. Customers sent their often criminally required coins to the mixer and received other coins back. For example, the Chinese exchange Bter.com reported that approximately 1.75 million dollars worth of Bitcoins were stolen and laundered via Bitcoin Fog (Ghoshal, 2015). In addition, exchange Binance informed that hackers stole 7,000 BTC, of which at least 4,836 BTC were sent to tumbler Chipmixer (Partz, 2019).

In 2019, the Dutch Fiscal Information and Investigation Service (FIOD) was able to take down one of the most popular mixing services named BestMixer (Europol, 2019). The mixing service transferred at least $200 million when it was active. Figure 1.1 gives a general overview of how using the mixing service works. In short, BestMixer's customers placed a mixing request or order via a Web browser. Then, they received an address on which the customer's had to deposit the coins they wanted to mix. What happens after this is as of yet largely unknown, it comes down to BestMixer distributing the received coins to the customer so that each customer received a payout in cryptos that have no link to their previously owned coins on the blockchain.

The Financial Intelligence Unit (FIU) identified mixing services as a money laundering typology (Financial Intelligence Unit, 2017). It is almost impossible to take down all tumblers, as the location of the servers they run on can be anywhere in the world. Therefore, restoring the relationship between the payout transaction(s) to a customer corresponding to the deposit placed by the same customer is crucial. In other words, mixing services have to be demixed.

Figure 1.1: General overview of BestMixer's mixing process

The generic process of demixing can be divided in two phases: the attribution phase and reconstruction phase. In the attribution phase, all addresses that belong to the mixing service are collected. Unfortunately, mixing services are not open about which addresses belong to their service, and they take full advantage of their anonymity on the blockchain. Therefore, methods have to be used to find as many addresses as possible.

In the reconstruction phase, an attempt to correctly combine deposits and payouts is made. The transaction history of all the collected addresses is vital for the reconstruction phase. The outgoing transactions from the attributed addresses can be possible payout transactions, and the incoming transactions can be possible deposits. The more correctly attributed addresses, the more accurate the reconstruction can become.

1.1. Research Goal

Currently, there is a knowledge gap because of the minimal knowledge on how mixing processes of centralized mixing services work, the limited availability of demixing methods. The studies performed on demixing lack data on the mixing services, which leads to researchers making assumptions about mixing strategies without being able to verify them. It is essential to tighten the research gap because mixing services obstruct Law Enforcement in their investigations. This thesis represents research on demixing the mixer BestMixer. With the confiscation of the tumbler, data on mixing requests and addresses came into the possession of the FIOD. Even though Law Enforcement parties have taken down multiple mixing services, no scientific research has been published that use the seized data. Therefore, it is a unique opportunity that this data was made available for this thesis. The main aim of this research is to develop a generic demixing method for centralized mixers based on ground-truth data. With this main research goal in mind, the following research question is formulated:

To what extent can ground-truth data help in developing a demixing method for centralized mixing services?

The following sub-questions will each help answer parts of the main question. The first question focuses on the attribution phase, while the second focuses on the reconstruction phase.

Sub-question 1: What are characteristics of BestMixer's addresses and transactions?
The seized data of BestMixer contains a list of all addresses that are part of the mixer, which means that the attribution of BestMixer is redundant. However, the addresses and complete transaction history can help discover characteristics of the mixing process, which can help detect other mixers, as many mixers use similar mixing techniques (Tironsakkul et al., 2020). The addresses of BestMixer and all transactions are essential building blocks in the modus operandi. In addition, the transaction history can show the economics of the mixer, for instance the balance and activity of the mixer. This can help substantiate the severity of potential laundering in mixing services. Chapter 5 discusses the methodology and results of this question.

Sub-question 2: To what extent does the demixing method filtering succeed in reconstructing payouts of BestMixer from the deposits?
This question aims to restore the relationship between deposit and payout of the mixer, using the knowledge we retrieved from the literature and the previous question. Literature on demixing only offers one method for reconstructing payouts to deposits: filtering all possible payout transactions on heuristics. No new reconstruction method is created in this research, but the current filtering methods are combined and optimized to

create a reconstruction tool that can easily be applied on all centralized mixers. The methodology and results of this question will be further discussed in Chapter 6.

1.2. Contributions

In this thesis, no new technique for reconstructing the deposits to payouts was created, but existing techniques were combined. Therefore, the results of this research form an important part to the contribution. This study contributes to the research on centralized mixing services as follows:

- New mixing characteristics were discovered, which can help in the attribution of other mixing services.
- This is the first research in which the economics of the entire transaction history of a centralized mixer are presented. For example, the balance of the mixing service throughout time and the amount of activity are described. The economics show that BestMixer was used very often, which means that it was a popular mixing service.
- An attempt to reconstruct deposits to payouts has been made. The attempt resulted in an advice on how to tackle demixing a centralized mixer: first map the activity of a mixer and then decide whether filtering can be a feasible technique for demixing.

1.3. Research Structure

The remaining part of this thesis will be presented in Chapters 2-8. Chapter 2 gives background information on mixing services and BestMixer in particular. Chapter 3 provides an overview of literature related to this research. In Chapter 4, the available data is described. The methodologies and results of the two proposed sub-questions are presented in Chapters 5 and 6, followed by the discussion in Chapter 7. Finally, Chapter 8 contains the conclusions of this research.

2

Background

In the introduction section, the mixing service BestMixer was made acquainted. This section elaborates on the concept of tumblers to guarantee that the reader has the fundamental knowledge for understanding the remainder of this thesis. An explanation of how mixing services are used to launder money and elaborating on what is generally known about BestMixer so far is given in separate sections.

2.1. Money laundering with mixing services

Over the years, we have used many kinds of money; gold, multiple fiat currencies, and now cryptocurrencies. Whilst the concept has evolved, the purpose of funds has not: we use it to value products and services. This makes money indispensable to everyone in our society, including criminals. They eventually would like to spend their illicitly acquired profits on products without getting caught for the crimes they committed. Whilst for some forms of money, it has almost no risk to hand in illegal profits, this is challenging due to the publicly available ledger for cryptocurrencies.

Mixing services are a commonly used money laundering technique for cryptos. An ideal mixer provides users with a correctly working protocol that maintains anonymity and deniability (Ziegeldorf et al., 2018). This means that there should not exist a relation between input and output coins and that users should not be able to be identified if the mixing service is seized.

There are two types of mixing services, decentralized and centralized. Decentralized mixers are protocols that facilitate peers to pool coins for a single transaction that randomly returns the money to all parties involved. An example is depicted in Figure 1: Alice has two inputs that are sent to Bob, Bob's input is transacted to Carol, and Carol's input is distributed to Alice and Bob. This has all occurred in one transaction with multiple inputs and multiple outputs. When there are more contributors to one transaction, there is more randomization, which makes tracing back the relation to input and output addresses even harder. One of the first decentralized mixing services is CoinJoin (Maxwell, 2013). This type of mixing is very secure. No party has all the information about the transaction and other peers, but this also makes it challenging to implement and comes with limitations (Crawford & Guan, 2020).

Centralized mixers act as a third party in the money laundering process. They allow users to substitute their money with other money in exchange for a service fee. Figure 1 shows how Alice sends money to the mixer and receives back a different set of coins. Bob and Carol use the mixer simultaneously, but they do not acquire Alice's cash, as centralized mixers often use liquidity and a transfer delay to make the mixing process more complex. Users need to trust the centralized mixer, as there is no guarantee that you get your money back (van Wegberg et al., 2018). Also, the service can store information on orders that includes the relation between the deposits and payouts. When a mixing service is seized or compromised in any way, the information can help in tracing back mixed orders.

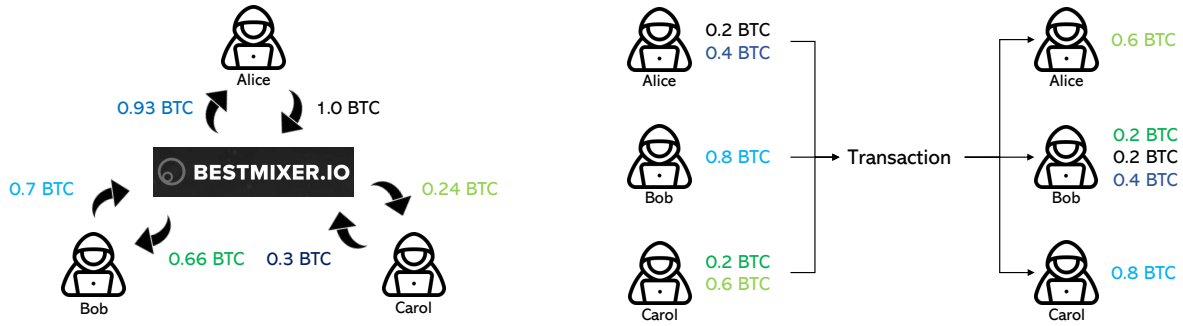
In most mixing services, clients can customise their orders. The customisation consists of choosing the values for certain parameters. The following parameters are used frequently by mixers (Hong & Lee, 2018):

- **Service Fee:** Using a mixer often comes with a price: the administrators keep a percentage of the coins as profit. Either this fee is determined within a specific range by the customer, or the administrators determine the service fee.

- **Transfer Delay:** This parameter determines the time it takes between the moment the customers deposit their coins and when they receive their mixed coins back. Customers often can pick from a range of delays: the delay of mixed coins with *Bitcoin Fog* could take up to 48 hours, and the delay of *CoinMixer* could be set to be maximally 120 hours.
- **Number of Output Addresses:** Many mixing services allow customers to split the deposits into multiple payouts to different output addresses. For example, Bitcoin Fog allowed for maximally 20 output addresses. However, the more common maximum output addresses are 5 and 10.

The more parameters a mixer uses, the more complex it becomes to demix. Some mixers used a fixed value for one or more parameters. For example, mixing services *Helix* and *Bitcoin Mixer* use a fixed service fee, which means that the payout to the output addresses could easily be calculated to a specific value instead of a range of possible payout values. This means that using a fixed value can be a weakness in the mixing process, as it is simpler to search for a specific value in all payouts than a range of values.

Mixing services are harmful to Law Enforcement because when a mixer is used by a suspect, they can falsely prosecute a different user of the mixer. Because of the anonymous nature of cryptocurrency addresses, it is primarily unknown what addresses are part of a mixing service, so the use of a tumbler is not always noticeable.



(a) Centralized mixing

(b) Decentralized mixing

Figure 2.1: Example of centralized and decentralized mixing procedures

2.2. BestMixer

BestMixer was a top-rated centralized mixing service. The mixing service was active from 27 March 2018 until 22 May 2019, a total of 421 days. Customers neither had to make an account nor had to provide identification information when using the mixer. The following information was requested when a user wanted to use the service:

- **Currency**: BestMixer offered its service for Bitcoin, Litecoin and Bitcoin Cash. The administrators were planning on supporting Ethereum, but the service was taken down before this was realized.
- **BestMixer code**: Customers could use the service multiple times. To guarantee that the customer did not get his coins back from an earlier transaction, he received a unique code that could be used during future mixing sessions. Also, the administrators rewarded loyalty; customers that used the BestMixer code were given a personal discount.
- **Output address(es)**: BestMixer transferred the customer's new coins minus the service fee to the requested output address. The customer had the option to receive back their deposited coins on multiple addresses, 11 in total. The option of more output addresses makes demixing more difficult since there are more possible combinations of transactions that can form the payout.
- **Service fee**: For Bitcoin, the quantity of the service fee was determined by the customer, within a range of 0.5% to 3%. The choice of the service fee determined in what pool the coins would circulate. According to the administrators, each pool has its characteristics:

  - *Alpha* when the service fee was lower than 0.85%. All coins in this pool belonged to users of the mixing service. This means that if criminals used the mixer for their money laundering, other users received these criminally linked coins back.
  - *Beta* when the service fee was between 0.85% to 1.25%. The creators of the service claimed that this pool consisted of large transactions of users, private system resources and investors' funds. The chance of receiving criminally linked coins was lower in this pool than in the Alpha pool. However, the chance was higher than 0 as this pool still contains coins of other users.
  - *Gamma* when the service fee was higher than 1.25%. This pool only contained coins from private system assets and investors' coins, guaranteeing users that the coins they received came from legal sources.

  Besides the fee percentage, a fixed was added for each output addresses. This means that customers had to pay more when they wanted to receive back money on 10 output addresses than when they only used 2 output addresses.

- **Percentage distribution**: The customer had to decide how the coins should be divided over the provided output addresses if they had chosen more than one output address.
- **Transfer delay**: The transfer delay was chosen individually for each output address. The maximum delay was 72 hours.
- **Amount**: This was not requested from the customer, but given as information since there is a minimum and maximum that a customer could mix. The minimum amount of transfer was 0.001 BTC, BCH or LTC. The maximum amount of transfer differed throughout time.

Once the customer had filled in all the information, an order page was generated. The order page showed the customers' BestMixer code, a summary of the placed order, the address where the coins should be deposited, and a Letter of Guarantee. This letter of guarantee would verify that the address on the order page was created by BestMixer.io and not a person with ill intentions.

An order page was available for 24 hours, in which the customer should transfer the cryptos. If the money was not transferred within this time frame, the order page would get deleted. The administrators made videos to explain how to use the mixer in 11 different languages on their YouTube channel. They also provided an API so web developers could incorporate the mixing service on their websites. The mixer was accessible through the Clear Web and TOR.

The mixing service operated as a regular company, with strategic marketing plans. For example, customers who showed loyalty by placing multiple orders with their BestMixer code received discounts. In addition, the administrators launched a marketing campaign targeting addresses that transferred large sums of money as potential clients. For these campaigns, one address was used that the administrators referred to as 'The official address': 1BestMixVhna91MkP7pKRtjej3bFq6Ze46. Another marketing strategy was the partnership program: the BestMixer team would pay 50% of the income of orders to partners if the order was placed via the reference link of the partner. Partners could enter this program by filling in a form. It is unclear whether every partner would be excepted or the administrators would check the potential partners first. Partners could post their referral links on any website to redirect potential clients to the mixing service.

2.3. Cryptocurrencies in BestMixer

As mentioned before, BestMixer is capable of mixing Bitcoin, Litecoin and Bitcoin Cash. All these cryptocurrencies have their own ledger. Most of the customers of BestMixer used it to launder Bitcoins, which means that most data will be related to the Bitcoin ledger. Therefore, this research focuses solely on Bitcoin transactions.

3

Related Work

The amount of literature on mixing service BestMixer is minimal: when BestMixer.io is queried in Google Scholar, there are 38 results. In these results, the mixer is often mentioned as an example, and only one result showed an attempt to demix BestMixer (Anton et al., 2019). Therefore, a broader approach for searching relevant literature was used, where the focus is on both phases of demixing: the attribution and reconstruction phase. The first section of this chapter describes attribution techniques, followed by a section on previous attempts of reconstructions. Besides gaining new insights from these academic papers, a knowledge gap is presented in the final section.

3.1. Attribution methods

In the attribution phase, addresses that belong to a mixer are collected. The results of demixing will not be reliable if the attribution is incorrect. When addresses are missing, the reconstruction will not always find the correct payout, whilst when too many addresses are attributed that do not belong to the mixer, false accusations can be made to innocent parties. The attribution phase is, therefore, a delicate process. This section describes three commonly used attribution methods in the literature. Each subsection will highlight how Best Mixer's data can contribute to improving these methods or how these methods can be used within this research, even though attribution is redundant for BestMixer.

3.1.1. Clustering bitcoin addresses

One of the techniques that is often used for the attribution of mixers is clustering. Clusters can be created by looking at transaction behaviour, which can be translated into heuristics. Clustering addresses can help identify users and their illegal activities; if one of the addresses can be linked to an entity, the transactions of all the other addresses in that clusters are likely to belong to that entity. Different heuristics have been developed for different types of behaviour of mixers.

Multiple input is such a behavioural heuristic (Nakamoto, 2008). One transaction can have multiple input wallet addresses. The *multiple-input* or *common spending* heuristic indicates that these input wallet addresses belong to the same entity. All input addresses must use their private key to sign the digital signature to validate the transaction. Unfortunately, this heuristic can create clusters that can contain multiple users. For example, CoinJoin transactions have multiple input addresses that belong to different users, so these users will be inaccurately grouped together (Tironsakkul et al., 2020).

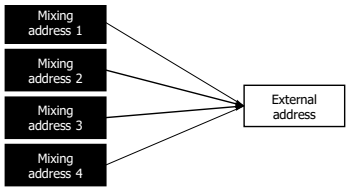


Figure 3.1: Pattern multi-input heuristic, the addresses in black boxes are clustered.

Another clustering method is based on spare change. The *one-time change* heuristic clusters the input address with the output that is most likely to be the change address (Meiklejohn et al., 2016). Conforming to the Bitcoin protocol, Bitcoins from an address must be spend all at once. The only way to divide a transaction is by using a change address. Part of the transaction will be sent to its new owner, and the excess money is sent back to the sender to the change address (Ermilov et al., 2017).

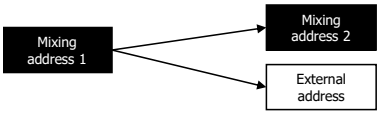


Figure 3.2: Pattern spare change heuristic, the addresses in black boxes are clustered.

In addition, a heuristic can be generated based on the behaviour or reuse of addresses. To enhance privacy, most mixing services use a deposit address only once (Möser et al., 2013). This information can be used for a heuristic that excludes all addresses that have been used multiple times.

Moreover, clustering can occur on a chain of transactions instead of individual transactions. A clustering method that looks into chains of transactions is the *peeling chain* heuristic. This heuristic is similar to the spare change heuristic, but all transactions in the sequence have one input and two output addresses. According to this heuristic, the spare change address of a transaction is the input address of the next transaction. When it is evident that a mixing service uses a transaction pattern like a peeling chain, this pattern can be used to establish whether a transaction was potentially part of a mixing process. The transaction to the output address should conform to the peeling chain format. All other transactions should be excluded from the cluster. This is not a very reliable heuristic because a mixer can change the designs or randomize the pattern. This risks leaving out potential output addresses.

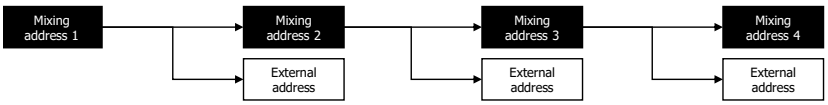


Figure 3.3: Pattern peeling chain heuristic, the addresses in black boxes are clustered.

Tironsakkul et al. (2020) used the fact that mixing services often use a fixed transaction fee for a certain amount of time. This means that if the transaction fee is known at a particular time, all other transactions from the mixer should also use this transaction fee so that all different transactions can be excluded. If mixers do change the transfer fee at random, this condition cannot be used.

Clustering does not have to rely solely on a heuristic. Ermilov et al. (2017) combined the multi-input and one-time change heuristics with off-chain information derived from the Internet. The authors collected information frames that contain both the wallet address and a 'tag', for instance, a username or company name. The information is collected by web crawling and manually analysing known crypto companies. These information frames can be used to check whether addresses in one cluster have different tags. If so, these addresses and the clusters they are possibly already part of should not be merged. The experiments performed by the authors show that the combination of the blockchain and off-chain information outperforms only using the heuristics.

One of the weaknesses of clustering is that it assumes that the behaviour of all addresses in a mixer can be translated to a heuristic. It is possible that the actual transaction behaviour is more complex than expected or that a mixer uses multiple different patterns for mixing, which could lead to missing a lot of the addresses in the cluster. Analysing BestMixer could help in verifying whether this mixer can be completely attributed with the current heuristics, or if more complex transaction behaviour was adopted to make detection of the mixer more difficult. In the latter case, the more complex behaviour can be used to create new heuristics.

3.1.2. Detecting mixing services with machine learning

In recent years, machine learning has become a widespread technique in analysing cryptocurrencies, which includes numerous studies that examine the use of machine learning for mixing detection. Studies on using Support Vector Machines for detecting services on the blockchain show conflicting results regarding the performance. When using an unsupervised approach to detect anomalies, the model detected 2 out of 30 known anomalous behaviours (Pham & Lee, 2016). A supervised approach for detecting exchange addresses

seems more promising. In an experiment on identifying exchange addresses, the technique was able to identify the addresses with high F1 scores (Liang et al., 2019), and since in this thesis research ground-truth data is available, a Support Vector Machines can be applied with the supervised technique.

Likewise, a Long Short-Term Memory Transaction Tree Classifier that uses deep learning to extract features and classification is used to identify mixing activity (Sun et al., 2021). Their experiment shows that the technique performs better than rule-based approaches, for example, clustering. The labels in the dataset created by the authors are based on heuristics and not ground-truth data, which makes the outcome of their research less reliable. In the conclusive section, the authors admit to this problem themselves.

Another paper uses an ensemble tree model to predict the class of an address, such as a Ponzi, dark market or mixing service (Nerurkar et al., 2021). The paper's authors compare their model with multiple popular machine learning models; SVM, Logistic Regression and Random Forest. The benchmark shows that their proposed model performs comparable to Random Forest and outperforms SVM and Logistic Regression on the accuracy, but is more resource-intensive. The difference in performance can be caused by fine-tuning the proposed model and not the benchmark models. The authors point out that the performance of their ensemble tree model is divided over the different classes due to limited data of some of these classes.

A recurrent limitation in research on detecting mixing services is the accessibility of ground-truth data. Companies like Chainalysis have categorised addresses, but this data is expensive and incomplete. Ermilov et al. (2017) describe that they collected and tagged millions of Bitcoin addresses, but unfortunately, they kept the data set private. Therefore, attribution with machine learning will likely only work with enough available data. Since all BestMixer addresses are known, it could be verified how well attribution with machine learning actually works. However, this is outside of the scope of this research.

3.1.3. Tracking the crypto trail with taint analysis

A commonly used technique in understanding mixing services better is taint analysis, where the movement of coins are tracked and marked as 'tainted' after they are deposited on an address belonging to the mixer. In literature, multiple tainting analysis methods are considered. For example, Möser et al. (2013) used the taint analysis tool from blockchain.info and found a direct connection between input and output coins in one of the three mixers used in the experiment. On the other hand, van Wegberg et al. (2018) used the same taint analysis tool and found no taint between the deposited and withdrawn coins in their cash-out experiment. Unfortunately, this tool is deprecated, so it cannot be used for further research.

Another paper by Möser et al. (2014) specifies two different taint analysis policies, 'poison' and 'haircut'. The poison method will mark all transaction outputs -starting from the transaction made from the user to the mixer - as 'tainted', whilst the haircut will proportionally taint the amount of incoming tainted cryptos to the normal cryptos. A limitation is that a relatively large amount of transactions still will be tainted when using the Poison method.

The poison tainting method has been used in combination with clustering in tracking mixers (Tironsakkul et al., 2020). This approach has the potential to connect the deposited cryptos to their mixed outputs. Although the clustering method from Tironsakkul et al. (2020) still leads to false-positive clusters, the authors are convinced that false positives can be lowered by using even more complex clustering methods and exploiting external information.

Using taint analysis can help gain more insight into the proportion of criminally linked coins in a cluster. It is possible to find a connection between input and output. However, taint analysis does not lead directly to recognizing patterns in the mixing process, so attribution with taint analysis alone will likely result in many addresses that are falsely labeled to be part of a mixer. Taint analysis can be used on BestMixer's addresses to see if it is possible to link all addresses of a mixer using taint analysis. This will be done in Chapter 5.

3.2. Reconstruction of mixers

The second phase of demixing is the reconstruction phase. Only two attempts to build a reconstruction tool can be found in the literature: one generic demixing tool and one demixing mechanism specific to BestMixer. Both tools make use of the same technique: brute-forcing all potential solutions. Specific filtering heuristics are used to determine whether a payout or set of payouts are possible solutions. The following filtering criteria have been used in these previous attempts:

- **Service Fee**: Mixing services usually work with a service fee. Therefore, the total amount of payout coins is not equal to the amount of deposited coins. The maximum payout amount can be calculated by extracting the minimum service fee from the deposit amount. All transactions in which more than

this amount was sent can be filtered out as possible payouts. The level of difficulty to demix a service increases when a services uses a range of service fees instead of a fixed service fee. For example, mixing services DarkLaunder, Bitlaunder and CoinMixer used a fixed service fee, this in combination with a very poorly designed laundering algorithm made it possible to link a small set of deposits and payouts de Balthasar and Hernandez-Castro (2017). As the service fee in BestMixer ranges from 0.5% to 3%, there is a range of maximum withdrawn coins. This filtering criterion can therefore still result in many possible payout transactions.

- **Transaction Delay**: When a mixer makes use of a transfer delay, this can be used to filter out transactions. The maximum payout date can be calculated by adding the maximum transfer delay to the deposit date. All transactions that occurred after this maximum payout date can be filtered from the list of possible payouts.

- **Amount of Output Addresses**: Most mixers offer their service such that multiple output addresses can be used by the customers, which means that a set of payout transactions cannot contain more than the maximum amount of output addresses. This can be used to stop the brute-forcing process once all possible combinations with the maximum amount of outputs is reached.

Hong and Lee (2018) designed a generic demixing algorithm with these three filtering criteria. They tested their tool on a mixer with a vulnerability in the mixing process, namely a fixed service fee. According to the authors' evaluation of the Helix mixer, their demixing algorithm is 99.14% accurate. Unfortunately, BestMixer does offer its service with multiple output addresses and a non-fixed service fee. Therefore the algorithm might not perform well on BestMixer and other comparable mixing services.

The researcher that attempted to demix BestMixer incorporated clustering heuristics and filtering to develop a demixing tool (Anton et al., 2019). The clustering heuristics were used to map all addresses that belonged to the mixing service. However, these heuristics were taken from literature without verifying if they also applied to BestMixer, which makes the tool unreliable as a lot of addresses that belong to BestMixer are not included in the collected addresses.

Overall, it seems that filtering all transactions is not a very efficient solution as it requires significant computing power. Also, with the bulk of Bitcoin transactions within a certain amount of hours, is it very likely that the filter returns multiple possible combinations, not one. However, the current filtering methods can be improved by optimizing the brute-forcing algorithms and adding more specific filtering heuristics.

3.3. Knowledge Gap

The current knowledge on demixing centralized mixers is minimal. A significant part of the research on mixing services focuses on the attribution phase and not the reconstruction phase. Scarce public data sources restrict the research on discovering patterns that can help verify and create new clustering heuristics. Also, it limits the attribution using machine learning; testing and training a model is difficult with a small dataset. Only one method for reconstructing deposits to payouts is known and its not efficient yet.

4

Data

One of the conclusions of the previous section was that many researchers have little data at their disposal. Fortunately, the FIOD has made data on BestMixer available that can help to gain new insights in mixing. This section will elaborate on the available data that was used during this research. The first data source is publicly available, whilst the other data is private.

4.1. Blockchain

The blockchain of Bitcoin is publicly available; information kept on the blockchain can be extracted by using Bitcoin API calls (Bitcoin.org, 2021). The most important command for this research is the getrawtransaction command that requires the transaction ID (txid) as input. The output is in JSON format and gives the information shown in Table 4.1.

Feature	Description
Txid	Transaction id.
Hash	Transaction hash.
Version	Transaction version, represents what technology the transaction supports.
Size	Amount of bytes of the serialized transaction size.
Vsize	Virtual transaction size.
Weight	Weight of transaction in weight units.
Locktime	Date and time transaction can be mined into a block.
Vin	List of sending addresses.
Vin: Txid	Transaction id of transaction when this address was received.
Vin: Vout	Index in the Vout list when this address was received.
Vin: scriptSig	Script contains asm and hex of transaction
Vin: sequence	Sequence number of the script.
Vout	List of receiving addresses.
Vout: value	Amount of received BTC.
Vout: n	Index in the Vout list.
Vout: scriptPubKey	List with assembly (asm), hex of the transaction (hex), required signatures (regSigs), type of signature and receiving address.
Hex	Hex-encoded transaction id.
Blockhash	Hash of the block where the transaction is part of.
Confirmations	Amount of confirmations.
Time	Date and time the block was generated.
Blocktime	Date and time the block was generated.

Table 4.1: Features in the transaction JSON returned after a getrawtransaction API.

4.2. BestMixer

The data on BestMixer used in this research is bundled in two different data files. The data was collected over a non-continuous period of 5 months. The first data file is a list with addresses belonging to the BestMixer wallet. This file will be referred to as the Wallet list during the remainder of this thesis. The second data file contains information on orders placed by customers. Data on orders is very important for this research, as it contains ground-truth information that can be used for building models and verifying these models. Table 4.2 shows all information that was sent to BestMixer when receiving an order. The data file contains 23,031 orders. It is worthwhile to note that the Order data contains 15 output address columns, whilst the interface of the mixer for customers only facilitated 11 output addresses. A limitation of this data file is that it only contains orders made through the Clear Web, so if there were any orders made via TOR we are missing information. This can impact the results of this research.

Feature	Description
Source	Data source.
Deposit Address	Adress that belongs to the BestMixer. The customer should deposit on this address.
Date time	Timestamp that the customer last viewed the order, in GMT+1 time.
IP Address	IP address of the customer. This address is hashed for privacy reasons.
Status	Last known status of the order during the last webpage view by the customer.
Deposit Amount	Amount of BTC transferred by the customer during the last webpage view.
Coin	Currency of the transaction.
Language	Chosen language for the website by the customer.
Application	Session ID used for the network communication.
Partner ID	Affiliate ID that belongs to 'partner' websites that link to the BestMixer.
UID	Customer ID.
User Agent	Type of web browser used by the customer.
Request URI	URL of the order that uses language setting and order id.
Order ID	ID of the order.
Fee pa	Fee per address.
Fee sr	Standard added fee.
Output Address 1 to 15	Wallet addresses where the money is paid out by the mixer.
Unnamed: 32	Extra output address.
IP range	IP range of the IP address.
IP hash	Hashed IP address.

Table 4.2: Features per order in the Order data file.

4.3. Chainalysis

Chainalysis is a company that provides government agencies, exchanges, financial institutions and insurance and cybersecurity companies with tools and data for investigating financial crime related to cryptocurrencies (Chainalysis, n.d.). Chainalysis has collected a significant amount of information on illegal activities on the blockchain and uses techniques to cluster and label addresses. This is a private data source, as it is only accessible when paid for. Chainalysis provides an API that can be used to query specific data, for instance, all transactions from a certain cluster or address. Another method for querying data from Chainalysis is using AJAX requests in the browser. In Chainalysis, data can be requested with an address as input, whilst Bitcoin API calls need the transaction hash. Table 4.3 shows what features can be extracted from an AJAX request for all transfers from and to a particular address. The output file is in JSON format.

Feature	Description
Datetime	Timestamp of transaction in UNIX time.
Hash	Hash of transaction ID.
Asset	Type of currency.
Value	Amount of money sent/received in BTC.
ValueUSD	Amount of money sent/received in US dollars.
Counterparty: RootAddress	Address of the counterparty in the transaction.
Counterparty: Category	If categorized, the service that the counterparty belongs to.
Counterparty: systemName	If categorized, the name of the service.
Receiving Address	Wallet address that will receive the money.

Table 4.3: Features per transaction in a Chainalysis transaction JSON.

5

BestMixer Characteristics

5.1. Methodology

There is minimal knowledge on the mixing process of BestMixer or even mixers in general. This knowledge gap makes it challenging for Law Enforcement to cope with mixing, as they can lose track of the rightful owner of tainted coins. A better understanding of the mixing process can help demix centralized mixers. When patterns are discovered in the mixing process, they can be tested and used on the attribution of other mixers. The mixing process can also help design additional filtering heuristics specific to BestMixer for the reconstruction. Therefore, it is essential to discover more about BestMixer's modus operandi.

The amount of transactions from and to BestMixer addresses makes it almost impossible for the mixer to have been manually operated. Therefore, the modus operandi must rely on automation to some extent. The mixing service consisted of addresses that sent and received transactions to addresses belonging to the mixer or addresses outside the mixer. When an address received and sent to another BestMixer address, a sequence of transactions can be observed. Sequences can reveal patterns in the mixing procedure. For instance, the peeling chain pattern has been acknowledged as a possible framework to use in the mixing processes (Tironsakkul et al., 2020).

As mentioned before, there should not exist a relation between the deposit and payout coins that enter and leave the mixer. Therefore, there should not exist a relation between the deposit address and the payout address that sends the coins to the customer, which indicates that all addresses of BestMixer cannot form one large interacting cluster but multiple smaller clusters instead. These clusters can share characteristics and might give new insights into how the mixing service operates.

The sub-question *"What are characteristics of BestMixer's addresses and transactions?"* was answered using data analysis techniques. The components of the mixer could reveal more about the automated mixing process. Prior to conducting this statistical analysis, data was collected and pre-processed. Section 5.1.1 will describe the data collection, followed by descriptions of obtaining sequences and clusters in Sections 5.1.2 and 5.1.3. Finally, the used analyses techniques are explained in Section 5.1.4.

5.1.1. Data Collection and Pre-processing

In order to find characteristics of BestMixer, it is important to gather as much information as possible on the addresses. This information has to come from different sources. The data was stored efficiently and accessible for further analysis by combining and filtering the information. The first step was collecting all transactions from and to BestMixer addresses. A query was sent to the Chainalysis server in a Web Console using the Chainalysis API for all addresses. The result of this step was a JSON file containing transactions. Since each call to the API has a limit of 500 transactions in the response, a loop around the API call in combination with an increasing offset assures that all transactions are extracted. For sending transactions, the sender's address is not included in the result of the query; therefore, the filename of each JSON is equal to the address used in the query.

Chainalysis clusters addresses, and if an address belongs to a cluster, all transactions from that cluster are returned. For example, Chainalysis clustered 45,304 addresses with the label 'bestMixer.io'. This cluster contains 106,076 transactions, so all addresses that belong to this cluster have 106,076 transactions in the JSON file. This means that the information on counterparties in the JSON files from clustered addresses was

distorted. They contained the information of the root address of the cluster and not the actual address, which made filtering without extra information impossible. We used the Bitcoin API getrawtransaction call and the transaction hash to extract additional information about the transactions. There are two criteria for filtering the file:
1. If the value of the transaction is greater than 0 and the receiving address is equal to the filename,
2. If the value of the transaction is less than 0 and the sending address is equal to the filename.
The existing JSON files were overwritten only to contain transactions that meet the criteria. The second filtering step divided the data into three different types of interactions: internal, deposit, and payout.

Internal All transactions between two BestMixer addresses are referred to as internal interactions. If a transaction meets one of the following two criteria, it was considered an internal interaction:
1. If the value of the transaction is greater than 0 and the sending address equals an address in the Wallet list,
2. If the value of the transaction is less than 0 and the receiving address equals an address in the Wallet list.

Deposit All transactions sent to BestMixer addresses by addresses that do not belong to BestMixer are deposit interactions. These deposits do not necessarily have to be deposits made by customers but could be funds from investors and administrators, dusting attacks et cetera. Therefore, the interactions are filtered on one criterium:
1. If the value of the transaction is greater than 0 and the sending address does not equal an address in the Wallet list.

Payout All transactions that were sent from BestMixer addresses to addresses that do not belong to BestMixer are payout interactions. These interactions can be payouts to customers, payouts of the profits to the administrators, deposits to external services that could be added to the mixing protocol, and so forth. The following criterium was used to extract all payout interactions:
1. If the value of the transaction is less than 0 and the sending address does not equal an address in the Wallet list.

During the analysis, many requests to the data had to be made which could slow down the analysis. Therefore storing the data efficiently was essential. This was accomplished by selecting all relevant information from transactions and storing all information in two types of csv files; one file contained all relevant information on addresses and the other file included all relevant information on transactions. All features that were included in the two files are shown in Tables 5.1 & 5.2.

Features	Description
Address	Address from BestMixer's Wallet list
Sent addresses	Quantity of addresses that transactions were sent to
Received addresses	Quantity of addresses that transactions were received from
Sent transactions	Quantity of sent transactions
Received transactions	Quantity of received transactions
Deposit addresses	Quantity of non-BestMixer addresses that transactions were received from
Payout addresses	Quantity of non-BestMixer addresses that transactions were sent to
Sent internal addresses	Quantity of BestMixer addresses that transactions were sent to
Received internal addresses	Quantity of BestMixer addresses that transactions were received from
First transaction	Date and time of first transaction
Last transaction	Date and time of last transaction

Table 5.1: Features included in addresses file, with a description for each feature.

Bitcoin transactions can have multiple input and output addresses, making it more complex to store the data. Since some transactions contain multiple input and output addresses, these transactions were split into multiple mutations of which an example is depicted in Figure 5.1. The example transaction contains three input addresses and two outputs, which resulted in five mutations in the dataset with the same transaction hash, but with different combinations of receiving and sending addresses. A mutation can be part of multiple different types of transactions, for instance, a payout and internal transaction.

In addition to the challenge of storing more complex transactions, the multi-input and multi-output transactions also led to a complex challenge for storing the information on addresses. For example, the ex-

Features	Description
Datetime	Date and time of transaction of blockchain
Hash	Transaction hash
Value	Amount of transferred Bitcoins
Receiving address	Bitcoin address(es) receiving from the transaction
Sending address	Bitcoin address(es) sending the transaction
Types	Types of transactions
Fee	Amount of fee paid in BTC
Fee size	Amount of fee paid in satoshi per Byte

Table 5.2: Features included in transaction file, with a description for each feature.

ample transaction in Figure 5.1 Address 1 is involved in one transaction, but it received Bitcoins from three different addresses. Therefore, a distinction had to be made between how many addresses an address received from and the number of transactions an address was involved in receiving Bitcoins. The actual number of transactions the address was involved in is expressed in the features *sent transactions* and *received transactions*, where the calculation is based on the number of unique hashes. The variables *sent addresses* and the *received addresses* reflect the number of addresses the address sent to and received from. If an address received multiple transactions from the same counterparty, the counterparty was counted per transaction instead of only once.

Transaction hash: 1

Input		**Output**	
BestMixer 1	0.5	Address 1	0.6
BestMixer 2	0.3	BestMixer 4	0.4
BestMixer 3	0.2		

Mutations

Hash	**Value**	**Receiving**	**Sending**	**Type**
1	-0.5	Address 1, BestMixer	BestMixer 1	Payout, Internal
1	-0.3	Address 1, BestMixer	BestMixer 2	Payout, Internal
1	-0.2	Address 1, BestMixer	BestMixer 3	Payout, Internal
1	0.6	Address 1	BestMixer 1,2,3	Payout
1	0.4	BestMixer 4	BestMixer 1,2,3	Internal

Figure 5.1: Example of how mutations are stored

5.1.2. Transaction Sequences

The first step in finding all sequences was to determine what addresses are a begin address of a sequence. A begin address has no incoming transactions from another BestMixer address, so all addresses of type deposit were selected as begin address. Then, chains of transactions that were all sent to a BestMixer address was search starting at the begin addresses. The end address of a sequence is always an address that sent no transactions to another BestMixer address. All sequences were built and stored in a JSON file in the next step. Listing 1 show the storing format that was used for each begin address.

```
{
    "length": amount of addresses in sequence,
    "addressestime": {
        "address": unix time of transaction
        ..
    },
    "begin value": value of first transaction,
    "end value": value of last transaction,
    "begin": first address in sequence,
    "end": last address in sequence,
}
```

Listing 1: JSON example of a sequence

5.1.3. Clustering Addresses

With the list of transactions, the addresses were clustered. An address was added to a cluster once a transaction occurred between the cluster and the address. If an address already belonged to a different cluster, the two clusters were merged. An example of this process is depicted in Figure 5.2: a transaction between address 1 and address 2 that are part of different clusters results in the merging of the two clusters. The clusters and additional information was stored in a csv file, of which the features are shown in Table 5.3.

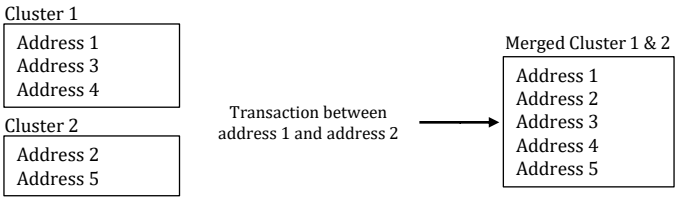


Figure 5.2: A graphic example of how two clusters are merged.

Features	Description
Addresses	List of all addresses in cluster
Amount received	Amount of received transactions of addresses in cluster
Amount sent	Amount of sent transactions of addresses in cluster
Begin date	First transaction made by address in cluster
End date	Last transaction made by address in cluster
Size	Amount of addresses in cluster

Table 5.3: Features included in clusters file, with a description for each feature.

5.1.4. Analyzing the Data

Data analysis techniques were used to find characteristics of BestMixer's addresses, transactions, sequences and clusters (Wu et al., 2021). Four categories of statistics were used:

Central tendency The mean and mode for all features. If the difference between mean and mode is small, a significant amount of the addresses or transactions should have similar behaviour for that feature.

Frequency distribution The number of occurrences of a feature within a certain interval, visualised in a line plot or histogram. A distribution clearly demonstrates how features are spread across a certain feature.

Correlation Heatmap The linear correlation between two variables, visualized in a correlation heatmap. The correlation is expressed in a number between -1 and 1, where 0 is no correlation between the two variables, 1 is a complete positive correlation and -1 complete negative correlation. The correlation is calculated to conform to Pearson's correlations coefficient formula, shown as Equation 5.1.

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{5.1}$$

The result of this equation is the correlation r between variable x and variable y. The mean of variable x and y are defined by $\bar{x}$ and $\bar{y}$. The variables x_i and y_i represent the values of variable x and y in datapoint i. The number of data points is defined by n.

Principal Component Analysis A Principal Component Analysis (PCA) aims to reduce dimensionality of the data whilst conserving most of the information captured in all data variables. The smaller set of variables can be used to observe trends, clusters and outliers. The first step in this analysis was creating the Principal Components, which are new variables that reflect the linear relation between the initial variables. The aim of creating the Principal Components is to capture as much of the covariance within the first component, which is accomplished by calculating the eigenvalues and eigenvectors from the covariance matrix that contains all covariances for each combination of variables. The amount of covariance a principal component accounts for is calculated descendingly: the first component explains most covariance, whilst the last component explains the least covariance. The covariance is calculated with Equation 5.2.

$$cov_{x,y} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{N-1} \tag{5.2}$$

The result of this equation is the covariance $cov_{x,y}$ between variable x and variable y. The mean of variable x and y are defined by $\bar{x}$ and $\bar{y}$. The variables x_i and y_i represent the values of variable x and y in datapoint i. The number of data points is defined by N.

The next step in this analysis technique was determining the number of variables that the data should be reduced to. This is done by calculating the percentage of how much covariance is explained by each component. The cumulative sum of these percentages helped decide how many of the components were needed. If one, two or three components explain a satisfactory amount of covariance, the results of the PCA can be visualised in a score plot.

5.2. Results

The following section describes the results of the exploratory study on BestMixer, starting with general observations of BestMixer's structure, followed by more in-depth analyses on addresses, transactions, sequences and clusters, each explained in separate sections. Then, a summary of the results is provided in which is highlighted how the characteristics can be used for attribution of other mixers and input for the next chapter's reconstruction.

5.2.1. General observations

The goal of these analyses is to find characteristics of the mixing procedure in the data. For some transactions it was known in advance that it was not part of the mixing process: the administrators launched what they called their advertising technology, where they transferred a very small amount of Bitcoins to wallet owners with a short message, to draw their attention and make these wallet owners aware of the existence of the mixer. They announced a campaign on the Bitcoin talk Forum [1], and by checking the transactions in the data, it was clear that they sent 0.00000888 BTC to 56,748 different addresses. As these transactions do not adhere to the mixing process, they were removed from the data set and were only used to calculate the mixer's balance. Therefore, all transactions after 21 May 2019 were also deleted from the data.

After pre-processing all transactions received and sent by BestMixer addresses, a total of 176,278 transactions were identified. In these transactions, 181,874 different BestMixer addresses were involved and on average each address was involved in 2.002 transactions. The first transaction was made on 17 February 2018. The last transaction was made after the mixing service was seized, which means that people kept sending money without a guarantee of receiving it back.

[1] https://bitcointalk.org/index.php?topic=3140140.msg47196426msg47196426

It is worth noting that 14,035 addresses from the Wallet list were unused, which is rather high given the total amount of wallet addresses. This might be due to the fact that orders could be placed, but never deposited. If the mixer generated a new deposit address for all orders, it appears that the administrators did not restate unused addresses.

Collecting all sequences with internal transactions revealed 864,229 internal sequences. Clustering BestMixer's addresses in groups with transactions between the addresses resulted in 12,523 clusters. With 181,874 used addresses and approximately 10 times fewer clusters, we can assume that many addresses are involved in internal transactions.

5.2.2. Addresses

BestMixer's addresses were analysed by performing several analyses. Multiple aspects of addresses were considered: the interaction with other addresses (received and sent), the period in which the address received and sent transactions, the date of the first transactions and the date of the last transactions. The three types of identified transactions were used; internal transactions, deposit transactions, and payout transactions. The addresses with no activity are not taken into account in this analysis.

Table 5.4 shows the means and modes of the explored aspects of addresses. The mean and mode for the *received addresses* and *deposit addresses* features differ the most compared to the other features. Therefore it is likely that addresses will show more differing behaviour for these features than for the other features. Furthermore, the means of both features are higher than the modes. This means that the distributions are positively skewed. The low modes for all features related to the amount of incoming and outgoing transactions show that BestMixer addresses do not interact much with other addresses and are likely to be used only once. This is a logical choice for a mixing service, as the reuse of addresses makes it easier to detect addresses that belong to the mixer. Another interesting finding from the table is that the first and last transactions' mode is not within the same month. This means that it is possible that the mixer did not have a peak in popularity within a particular month but that the popularity was more continuous after 12 March 2019. In order to confirm this speculation, the amount of transactions made per month is further explored in Section 5.2.3.

Feature	Mean	Mode
Received addresses	2.142	1
Received transactions	1.002	1
Sent addresses	2.064	2
Sent transactions	1.000	1
Deposit addresses	1.235	0
Payout addresses	1.158	1
Sent internal addresses	0.905	1
Received internal addresses	0.891	1
First transaction	-	12 March 2019
Last transaction	-	18 May 2019

Table 5.4: Means and mode of address features.

Received An address can receive transactions from counterparties, which could be other BestMixer addresses (*internal transactions*) and non-BestMixer addresses (*deposit transactions*). Table 5.5 shows the occurrences of the features *received addresses* and *received transactions*, where the first feature expresses the number of counterparties from which an address received, and the second feature expresses the number of transactions an address received. For simplicity, all addresses that received multiple transactions are grouped to have received >1 transaction, and all addresses that received from multiple counterparties are grouped to have received from >1 addresses. The table shows that most addresses received one transaction from one address: a total of 156,956 addresses. This is consistent with the earlier notion that the interaction with other addresses is low. This table indicates that there are no occurrences of addresses that received more than one transaction from one address. However, a counterparty was counted per transaction in the data collection, therefore it still is possible that addresses received multiple transactions from one address, but in this table it would be expresses as an address that received multiple transactions from multiple addresses.

There are 24,700 addresses that receive from multiple addresses, which is distributed between 2 and 705 addresses. Remarkable is that most addresses only receive from one transaction; only 218 addresses receive from multiple transactions. This means that 24,482 addresses received multi-input transactions. One address

received from significantly more addresses than all others: 705 addresses, which is almost ten times more than the second-highest amount of received addresses. Due to the frequent reuse of this address, is likely that it has a unique function in the mixer.

		Received transactions	
		1	**>1**
Received addresses	**1**	156,956	0
	>1	24,700	218

Table 5.5: Occurrences of the address features received addresses and received transactions.

To gain further insights into the receiving characteristics of BestMixer addresses, Table 5.6 was generated to show the distribution of interaction types of the received transactions per address. As a transactions could have multiple input addresses, all transaction types were counted per counterparty in a transaction. There are two types of counterparties that addresses could have received transactions from: received internal or deposit addresses. The deposit addresses are the result of transactions received from non-BestMixer addresses, and the received internal addresses are the result of transactions received from other BestMixer addresses.

Two notable characteristics can be derived from the table. Firstly, there are addresses that receive from multiple deposit addresses and there are addresses that receive from multiple internal addresses. It is not unexpected that BestMixer addresses received multi-input or multiple transactions as deposits. Customers could have had multiple resources for their acquired Bitcoins, for instance, multiple ransomware attacks. The time span given by the administrators between placing the order and making a deposit was 24 hours. Suppose a customer ran a business on a dark market, for example. In that case, they could use the same BestMixer order for all profits they made within 24 hours, which did not necessarily have to be transferred within one transaction. It is interesting that BestMixer also received multi-input transactions from other BestMixer addresses. Financially it makes not much sense to transfer money from multiple addresses to one address instead of sending it directly to a customer or external exchange step because it costs extra money in service fees. One of the reasons this was incorporated in BestMixer could be to mislead snoopers and Law Enforcement. As these are unexpected transactions, likely, they would not be recognized as part of a mixing process.

The other notable characteristic is that the number of addresses that receive from 1 internal and 0 deposits is very high. Table 5.6 shows that this is the most occurring combination of the features in all addresses. Since more transactions are received from an internal address than a deposit, there may be sequences of internally sent transactions. This is further explored in Section 5.2.4. The table shows that $49,218+22,278+17+11+1 = 71,525$ addresses received from a deposit address. With the assumption that deposit addresses are used only once, the maximum amount of processed orders is 71,496. It is possible that some of these addresses received their Bitcoins from investors or that the mixer makes use of an external mixing step and that the deposit is the payout from the other service.

Addresses that receive from both deposits and internal transactions are surprising. All of these addresses received from multiple transactions, which means that these addresses are reused. It is possible that these addresses have a function in the mixer that differs from an automated mixing process.

		Received internal addresses		
		0	**1**	**>1**
Deposit addresses	**0**	0	107,956	1,889
	1	49,218	17	0
	>1	22,782	11	1

Table 5.6: Occurrences of the address features deposit addresses and received internal addresses.

Many addresses of BestMixer show similar behaviour in regards to receiving transactions. The reuse heuristic from Möser et al. (2013) that states that a mixing service uses a deposit address only once would capture 99.9% of BestMixer's addresses. This means that the heuristic would perform well in attributing BestMixer. However, many addresses on the blockchain only receive one transaction, which means that the

heuristic would also attribute a lot of non-BestMixer addresses. Therefore, more heuristics are needed to single out the correct BestMixer addresses. Since 109,874 addresses received transactions from other BestMixer addresses, it would be interesting to see if the spare change heuristic could help attribute addresses. More information on how sending behaviour of addresses is needed, as the spare change heuristic assumes that a transaction has two outgoing addresses, of which one belongs to the mixer and one to an external address.

Sent When an address sends Bitcoins to another BestMixer address, the transaction is internal; if an address sends Bitcoins to a non-BestMixer address, the transaction is a payout transaction. Table 5.7 shows the occurrences of the features *sent addresses* and *sent transactions*, where the first feature expresses the number of counterparties to which an address sent money, and the second feature expresses the number of transactions an address sent. The values in the table represent the number of addresses that share the same characteristics. According to this table, most addresses sent one transaction: 181,656 addresses, which accounts for approximately 99.9% of all addresses. This number is equal to the number of addresses that received one transaction. However, where most addresses received transactions from only one counterparty, most addresses sent transactions to more than one counterparty.

There are 115,968 addresses that sent to more than one other address. The number of sent addresses that are higher than one seem to be distributed between 2 and 56,709 addresses, where two sent addresses occurs the most. This means that most sent transactions were multi-output transactions. For instance, an address that sent to 500 different addresses sent it within one transaction. One address sent transactions to 56,709 other addresses within 79 transactions. Converting this to an average leads to 717 receiving addresses per transaction, which is very high compared to all other transactions made by BestMixer addresses.

		Sent transactions	
		1	**>1**
Sent addresses	**1**	65,688	0
	>1	115,968	218

Table 5.7: Occurrences of the address features sent addresses and sent transactions.

Similarly to the frequency analysis on the receiving characteristics, a deeper understanding of the sending characteristics was needed. Table 5.8 was put together to show the distribution of all sending transactions, which means that both internal and payout transactions were included. The second column of this table represents the amount of all payout addresses. The second row display the number of sent internal addresses. An interesting result that can be extracted from the table is that most addresses sent to both types internal and payout. It is very likely that the payout and internal interaction was within one transaction, as there are only 218 addresses that sent more than one transaction. This is a persuasive indicator that BestMixer used a peeling chain in its payout procedure.

Furthermore, zero sent transactions are the second-largest group of addresses for both types. The table shows that more addresses sent to multiple payout addresses than to multiple BestMixer addresses. This means that multiple-output transactions are not necessarily a characteristic for internal transactions nor for payout transactions. In addition, the combination of zero payout addresses and one internally sent address occurs more often than the combination of one payout transaction and zero internally sent transactions, which could suggest that the mixer makes multiple internal transactions in one sequence.

		Sent internal addresses		
		0	**1**	**>1**
Payout addresses	**0**	0	58,676	396
	1	11,163	103,319	58
	>1	8,141	97	24

Table 5.8: Occurrences of the address features payout addresses and sent internal addresses.

Table 5.7 shows that the number of addresses that sent more than one transaction is very low compared to the addresses that sent one transaction. This means that the reuse heuristics would also capture 99.9% of

the addresses based on their sending behaviour. As many addresses sent a transaction to a BestMixer and a non-BestMixer address, the spare change heuristic would capture 57% of the addresses.

Using the information retrieved on receiving and sending behaviour per address, the addresses can be divided into five different types of addresses. All addresses that received from more than one or sent to more than two counterparties are labelled as an outlier. These types are shown in Table 5.10. A distinction has been made between *Addresses* and *Transactions*: in Addresses, the types are based on features sent addresses and received addresses, whilst Transactions is based on the features sent transactions and received transactions. The mode of the Addresses column is one received, two sent. BestMixer contains 86,513 addresses that share this characteristic. With this many addresses that share this combination of features, BestMixer likely uses a peeling chain to distribute coins to customers.

The combination one received, one sent also occurs often, especially when the transaction column is considered. It is clear that most BestMixer addresses were used only once in a mixing procedure; the other 227 addresses were used more often. A total of 2,429 addresses with combination one received, one sent receive from a deposit and sent to a payout. This is an exciting observation because transferring the amount directly to another party means that the total amount of the transaction can be sent away externally. Multiple things could explain the minimal interaction with the mixer:

- The address received from an extra external mixing step or an investor. These newly received coins are likely not linked to any of the deposits made by customers, which makes extra transactions to mix the coins further unnecessary. If the received value was high enough such that the money could directly be transferred as payout, there was zero interaction with other BestMixer addresses.
- The deposit that a client made was directly sent to another customer, an external mixing step, or transferred and kept as profit for the administrators.

From this table can be concluded that most addresses received and sent one transaction. Only 0.01% of the addresses show different behaviour. Therefore, it can be verified that the reuse heuristic would work well on the BestMixer data.

Type	Received	Sent	Addresses	Transactions
1 received, 1 sent	1	1	65.679	181.647
1 received, 2 sent	1	2	86.513	9
Sent outlier	1	>1	4982	0
Received outlier	>1	1	4.160	9
Sent & received outlier	>1	>1	20.540	209

Table 5.9: Occurrences of different types of received and sent behaviour of addresses.

Active Period Another interesting feature that can be extracted from the address data is the amount of days an address was active within the mixer. This period is calculated by extracting the date of the first transaction from the date of the last transaction. The distribution of the amount of active days per address is displayed in Figure 5.3 in the form of a line plot. The x-axis represents the number of days, the y-axis represents the number of addresses and has a logarithmic scale. The addresses were grouped first in earlier mentioned different types, which is represented by the different colours in the plot. Then, the height of each coloured line was determined with the following formula:

$$y = \text{number of addresses that have been active at most } x \text{ days} \tag{5.3}$$

This means that the line has to decline toward the end of the x-axis, as more and more addresses do no comply with the condition. When no addresses comply with the condition, the value for the y-axis is zero and therefore disappears from the plot. The height where x is zero represents the total amount of addresses.

The addresses that received their funds within the active period of the mixer differ a lot in the length of the active period: between 0 and 433 days. Most of the addresses are active for less than a day, which means that most transactions are conducted quickly. On the other hand, 1,974 addresses have an active period that is longer than one week. Some of the addresses with a long active period received their deposited funds before the mixer was active as a service. This shows that deposited funds are not always used as payouts for other customers that made deposits around the same time, which indicates that the mixer has to rely on reserves for the mixing process.

Another notable observation is that the address with the longest active period behaves differently from all other addresses. It receives and sends multiple transactions throughout time. In addition, it is the only address that was active during the beginning of the mixing service until after the mixing service was seized.

The differing active periods make it more difficult to find patterns in the mixing process. A heuristic based on the active period of less than 24 hours would exclude a lot of the addresses, and a heuristic that uses the longest active time would likely capture a lot of addresses on the blockchain as 433 days is really long to keep money on one address. Therefore, the active period was not used to generate a heuristic.

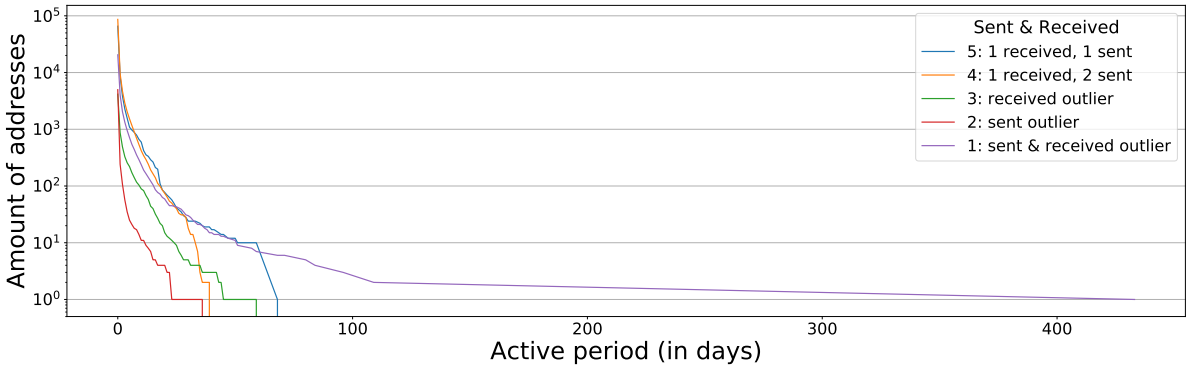


Figure 5.3: Distribution of the amount of active days for all addresses.

Correlation The correlation between address variables shows the strength of the linear relation between two variables, which indicates whether there are patterns between certain variables. These patterns can be used in other mixing services with similar modus operandi to predict and discover features from these new mixers. Figure 5.5 shows that there seems to be linear relations between received and sent features. For example, all data points in the scatter plot with features *sent addresses* and *payout addresses* appear to be distributed on a straight line. There also seems to be a linear relation between the features *first transaction* and *last transaction* (Figure 5.4). This means that it can be expected that there are linear relationships between address features, which makes the Pearson correlation function a good fit for calculating the correlation score.

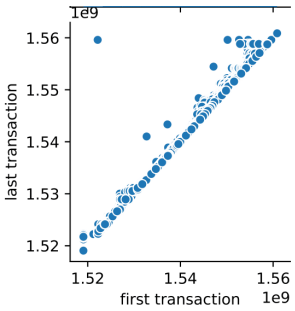


Figure 5.4: Scatter plot with address features first transaction and *last transaction*.

The correlation heatmap in Figure 5.6 represents the correlations between the features in a coloured matrix. Since all variables are on both axes, the heatmap is triangular in shape. The matrix demonstrates that many variable combinations have correlation scores between -0.1 and 0.1, which means they share almost no correlation. For instance, the variables *first transaction* and *last transaction* have low correlations with all the variables, but the correlation between the first and last transactions is perfect. It was expected that this linear relation would be strong in a positive direction, as the last transaction date cannot be sooner than the first transaction. Therefore, if the first transaction value is increased, the last transaction value will increase in the same direction as the first transaction. The low correlation with all other features could mean that all types of addresses were used throughout the active time of the mixer. The other time-related feature is the *active period*. It has a moderate correlation with some of the sending and receiving features, which means that it will be hard to predict the active period of an address if the first transaction is received.

There is a strong correlation between *amount sent* and *unique amount sent, unique amount received* and *amount payout*, but not between *amount received* and *unique amount received*. This means that in identify-

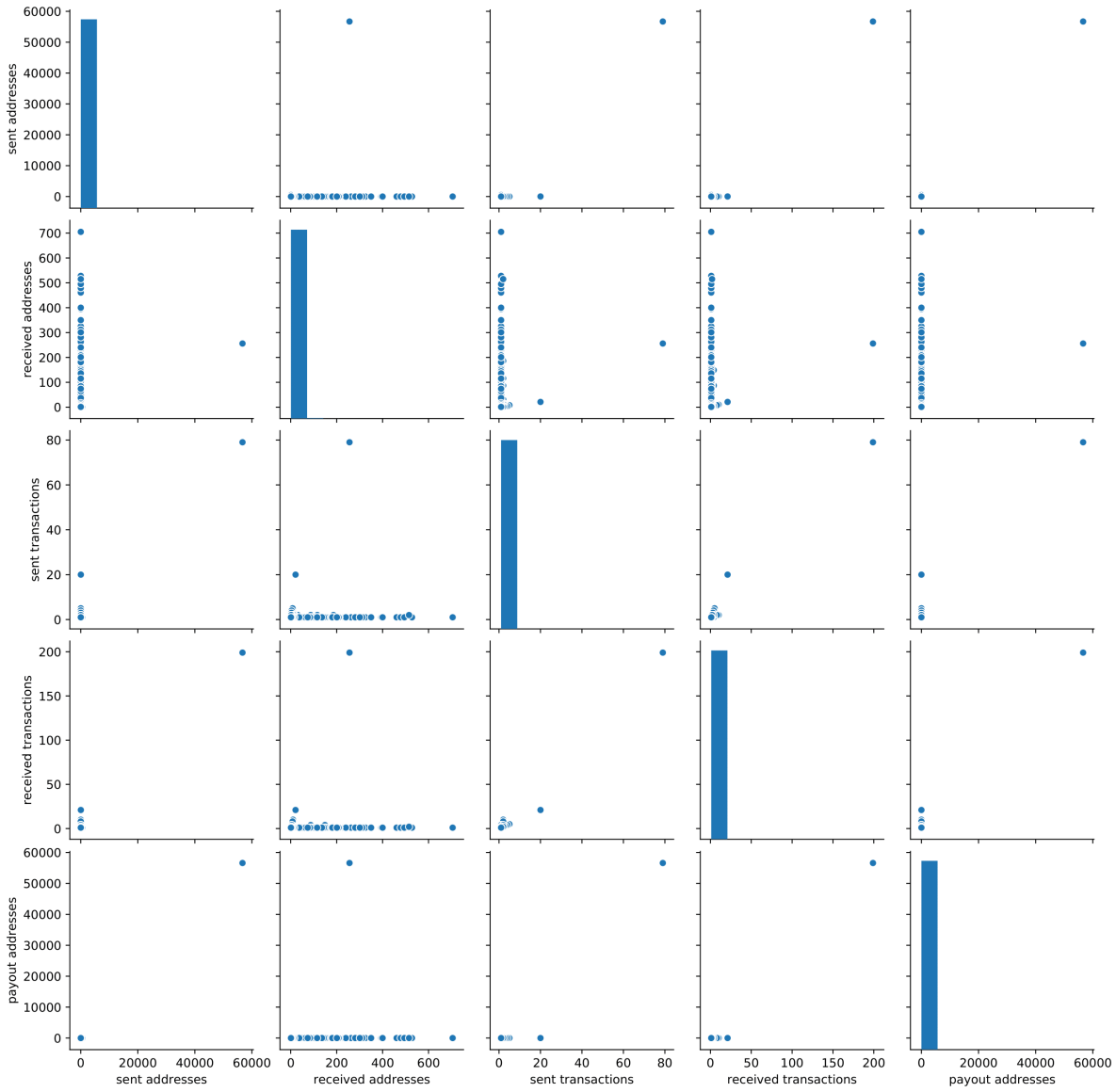


Figure 5.5: Scatter plots with received and sent address features on the axes.

ing patterns in the use of addresses, the amount sent feature can be used to predict other features with high accuracy. If the receiving features are known, it is harder to predict other features correctly.

Overall there is much difference in the correlation between all variables, which means that with knowledge of certain features, some other features are more challenging to predict than others. The features related to time are not strongly correlated to the sending and receiving features, which is unfortunate.

Principal Component Analysis In order to discover trends between the different types of addresses, a PCA was performed. The first step of this analysis was determining the number of components that the new dataset should be transformed to. The scree plot in Figure 5.7 displays the cumulative explained variance (y-axis) per index of the components (x-axis). When the cumulative explained variance is one, all covariance is accounted for in the components. The figure demonstrates how component 0 and component 1 already accounts for 100% of the covariance compared to using all components. Therefore, two components were used for the remaining part of this PCA.

Figure 5.8 shows the PCA scores for the first two components, with indexes 0 and 1. The first component, Component 1, is depicted on the x-axis and Component 2 is depicted on the y-axis. The colours in the graph represent the five types of addresses, where the numbers correspond to the following types:

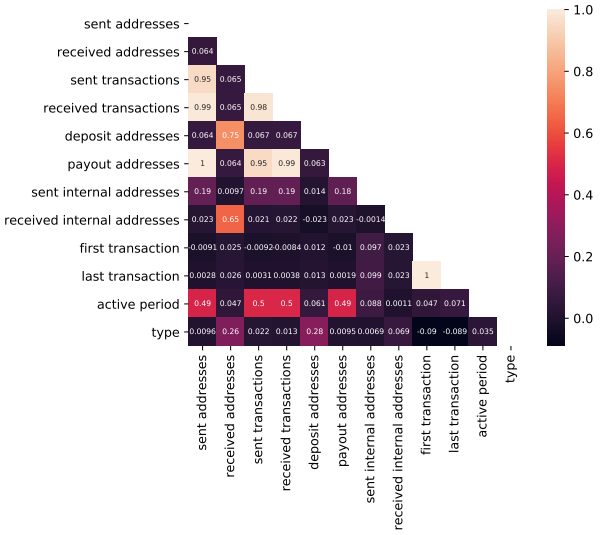


Figure 5.6: Correlation heatmap with address features.

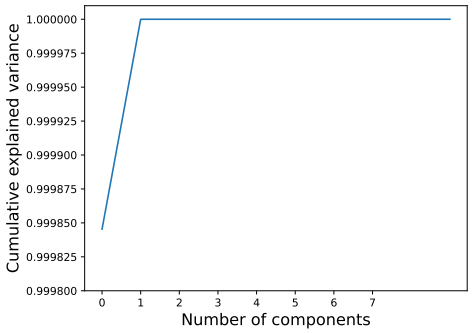


Figure 5.7: Scree plot of address variables.

1. 1 received, 1 sent
2. 1 received, 2 sent
3. received outlier
4. sent outlier
5. sent received outlier

The score plot clearly shows that most data points are aligned on the axis of Component 2. All types of addresses have data points that go above this line and are spread across the whole length of the Component 1 axis, meaning they cannot be segregated as three different types of addresses based on these two components. Type 5 relatively contains most data points with values that exceed the y-axis. For all other types, the values of Component 2 are within a range of 0.0 to 0.5. Type 5 has data points with values much higher than 0.5. These points occur in areas with a low density of data points and can be marked as outliers compared to the other data.

The PCA analysis shows that the data points cannot be seperated into clusters of the five types of addresses based on the dimension reduction, which means that the types share characteristics. This means that it is likely that most addresses can be captured within certain heuristics. However, one of the address types that was marked as outlier does clearly show that it has outlying values for component 2 compared to the other data types. This could mean that this data type has a different purpose in the mixing process than the other data types and that these addresses are not part of the automated mixing process. It will be more difficult to attribute these addresses.

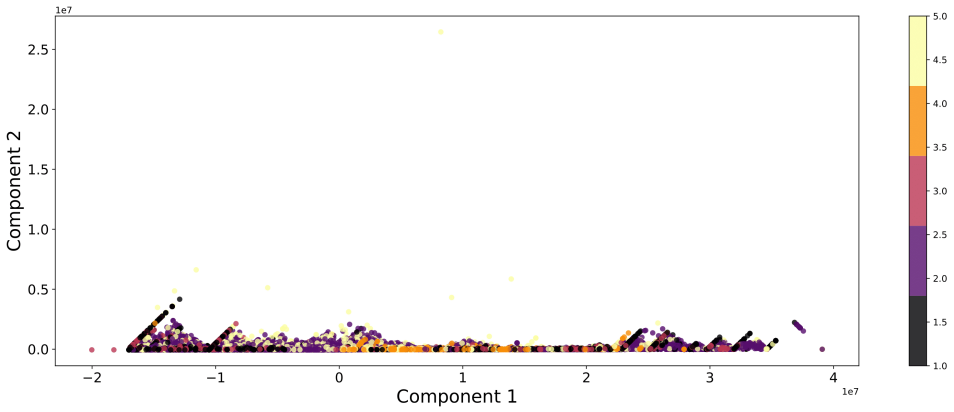


Figure 5.8: PCA score plot of address variables.

5.2.3. Transactions

Transactions to, between, and from BestMixer addresses do not only show which addresses interact with each other; they can also give more insight into the economics of the mixing service. For example, the balance of the mixer at any point in time can be extracted. Multiple features concerning transactions were explored: transactions over time, the value of transactions, fee, and balance.

The three types of interactions between addresses were used for the analyses: internal, deposit, and payout. As some transactions have multiple input or output addresses, combinations of the three types of interactions occur. This resulted in a new type: *Payout, Internal*. The amount of interactions for each type is displayed in Table 5.10. Most interactions are of the combined type Payout, Internal, which is another strong indicator for the peeling chain pattern. The amount of transactions from type internal and payout are the lowest. The internal transactions are multi-input or multi-output transactions 99,5% of the time. These transactions could have been carried out to split larger funds into smaller portions, or the opposite in the case of multi-output transactions: merge smaller values in a transaction. When looking at the amount of transactions of type *Deposit* and those of types *Payout* and *Payout, Internal*, the ratio between the deposits and payouts is around 1:1.50. This could indicate that not many customers chose to receive back their money on multiple output addresses.

	Deposit	**Internal**	**Payout**	**Payout, Internal**
Amount	69,479	2,444	1,195	103,159

Table 5.10: Occurrences of mutation types

The mean and mode of transaction features are shown in Table 5.11. All features except for *received addresses* show a significant difference between the mean and mode. The mean and more for the features related to the value and fee of the transaction even differ in magnitude, which can indicate that these features have a wide distribution. The features *sent addresses* and *received addresses* have means and modes within the same magnitude and are negatively skewed as the modes are lower than the means.

Feature	**Mean**	**Mode**
Date	-	14 March 2019
Value (in Satoshi)	8,554,914	888
Fee (in Satoshi)	4,083	88,320
Fee size (in Satoshi/Byte)	3,037	19,999
Sent addresses	2,167	1
Received addresses	2,068	2
Type	-	Payout, Internal

Table 5.11: Means and mode of transaction and mutation features.

Value Analyzing the values of mutations can help determine how difficult demixing can become. The mixing services accepted a wide range of deposit values. The values of incoming mutations are plotted in a

histogram. Suppose the figure has the shape of a Poisson distribution. In that case, it is more challenging to restore the relationship between deposit and payout. This is due to the fact that when more addresses receive comparable amounts of deposits, the payouts are also likely to be more similar in value, which makes finding the correct payout more difficult. At the tails of a Poisson distribution, there is a small number of deposit addresses with a small number of possible payouts, so these relations are easier to restore. However, the peak accounts for a much more significant amount of mutations, so a high percentage of the deposit-payout pairs is hard to restore. If the plot is uniformly shaped, all deposit transactions will likely have a comparable number of possible payouts.

Figure 5.9 demonstrates the distributions of values for all mutations. The x-axis represents the magnitude of the value, the y-axis is the amount of transactions on a logarithmic scale. The difference between the smallest and largest value is of magnitude 10^8. The distribution has the shape of a Poisson distribution, which means it will be more difficult to restore deposits to payouts.

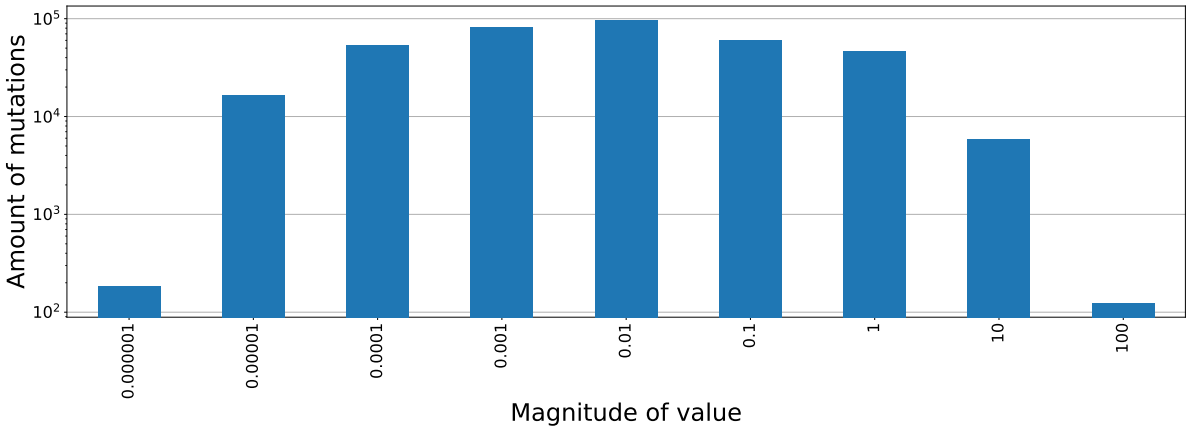


Figure 5.9: Distribution of magnitude of values for all mutations.

It is interesting to see how the values of the different types of interactions are distributed. Especially the mutations of type deposit, as these are potential deposits from clients and therefore can indicate the magnitudes of the orders that were placed. Therefore, Figure 5.10 was created. The plot shows the distribution of the magnitude of value and types of interactions for all mutations. The axes are equal to those of Figure 5.9, but the difference is that each transaction type has its colour and is depicted in separate bars. From the figure can be derived that most deposit transactions are of magnitude 0.01. However, the distribution seems to be positively skewed, so more transactions have a magnitude higher than 0.01 than lower. This shape can also be seen in the type' payout, internal', but the two other types are negatively skewed and have their highest peak for magnitude 0.001.

Notable is the large amount of deposit mutations that are lower than 0.0001 BTC. The minimal deposit amount is 0.001 BTC. All mutations with values lower than 0.001 BTC could not have come from deposit transactions from customers unless the deposit address received multiple transactions that sum up to 0.001. These low mutation values could be the result of dusting attacks, which is a technique that aims to deanonymize the owner of an address.

As multi-input and multi-output transactions were split into multiple transactions conform Figure 5.1, a mutation of type 'payout, internal' always has an 'internal' mutation with the same transaction hash. This internal counter mutation can have a value with a lower magnitude, as it is highly likely that the address was used to store spare change. This explains why the bars for 'payout, transaction' are higher than the 'internal' bars for some magnitudes of value. No transactions of type' payout, internal' were made for the smallest magnitude.

Transactions over Time The amount of transactions over time can show more insight into the activity of the mixer. In order to display the distribution of transactions over time, multiple figures were created. Figure 5.11 reflects the distribution of the amount of transactions over time. The x-axis represents the year and month, and the y-axis is the amount of transactions on a logarithmic scale. The figure clearly displays an upward trend in the amount of transactions throughout time. This could indicate that the mixer gained popularity as time passed, with a stabilisation towards the beginning of 2019. February 2019 and May 2019 had fewer

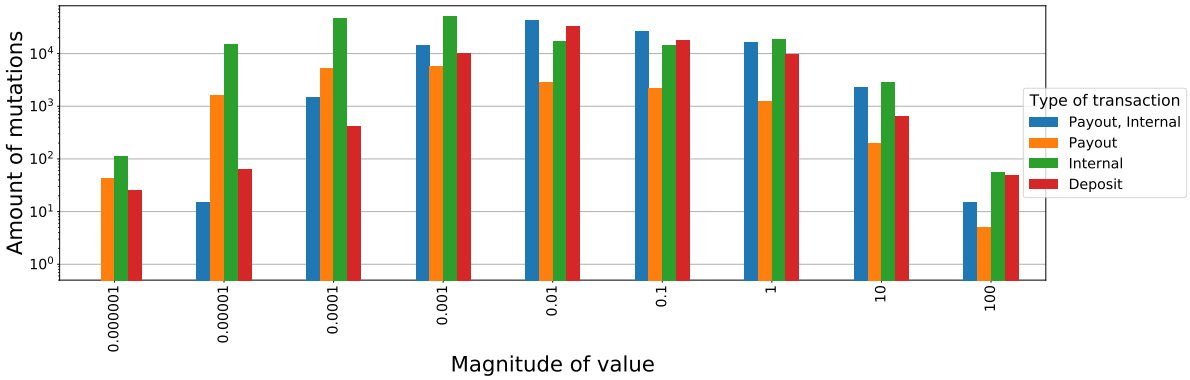


Figure 5.10: Distribution of magnitude of values per type of mutations.

transactions than the preceding month. This is not unexpected, as February has fewer days, and the mixing service ceased in May 2019. Most transactions were made in March.

Notable is that transactions occurred before the mixer offered its service on the market: four transactions were made in February 2018. Perhaps these transactions were made to ensure that the first orders could receive back a payout that was not linked to the deposit. The administrators could have also tested the mixer before it became available for others. This can be confirmed by looking at the types of transactions made in February 2018, which is discussed later in this subsection.

Another interesting result in the figure is the significant difference in height between June 2018 and July 2018. Almost ten times more transactions were made in July, whilst the growth in the preceding months was much lower. This large jump could be explained by, for instance, an increase in popularity, another campaign from the administrators, or a change in the mixing protocol.

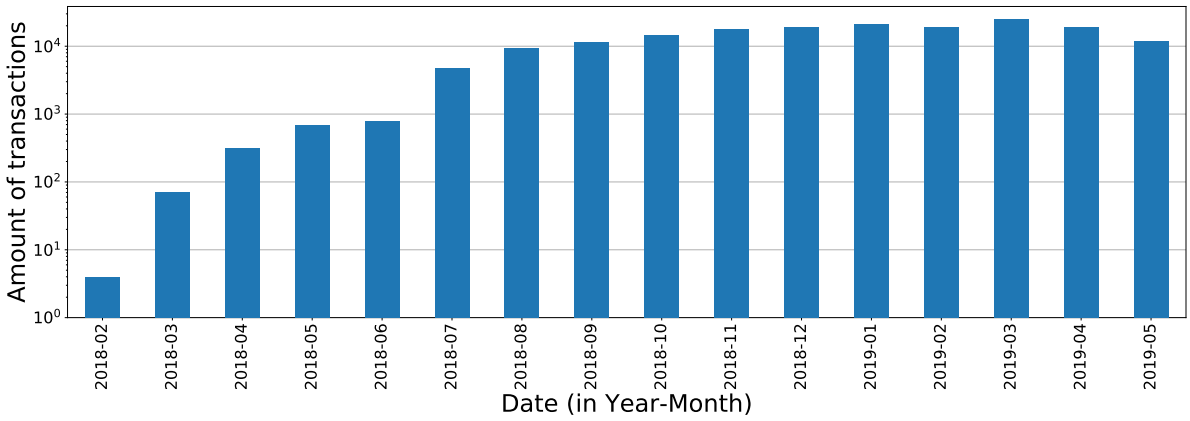


Figure 5.11: Distribution of amount of transactions per month.

In addition to the amount of transactions per month, it is interesting to look at the magnitude of the transactions. If the transaction magnitudes grow over time, customers likely make larger deposits, which could indicate that they think BestMixer is a reliable service or that it attracted a new type of customers with larger profits to launder. Figure 5.12 shows the distribution of the magnitude of values over time. The colours in the bars stand for the different magnitudes. The x-axis represents the year and month, the y-axis the percentage of the amount of transactions.

From the figure can be concluded that transactions with values of most magnitudes were made in all months that the mixing service was active, except for the first month. The most used magnitudes in the range between 0.001 and 0.1 are relatively consistent in the distribution of percentages throughout time. For instance, transactions with magnitude 0.01 form approximately 25-30% of all transactions within each month. This makes reconstructing the input to output easier, as the transactions are distributed almost uniformly when considering the magnitude of the value. The similarity in percentage distribution throughout time also indicates that the magnitude of the value of transactions does not grow or shrink over time, so there is likely

not much difference in the mixing behaviour of customers.

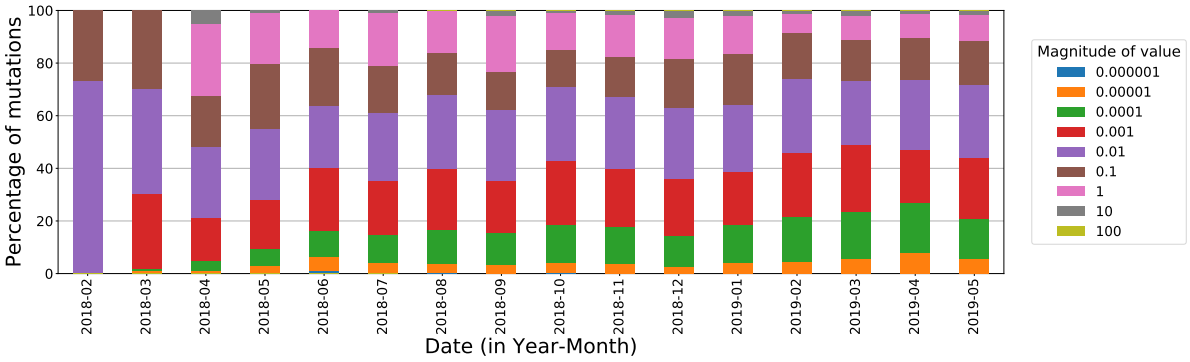


Figure 5.12: Distribution of mutations over time combined with the distribution of magnitude of values.

Another interesting feature combined with the transactions over time is the different types of transactions. The difference in the types of transactions can reveal changes in the mixing procedure or order behaviour. For instance, if the deposit percentage is lower than the sum of the 'payout, internal' and payout percentages, it can indicate that customers wanted to receive back their coins on more addresses or that something changed in the mixing procedure. Figure 5.13 displays the distribution of the different types of addresses over time. The x-axis represents the year and month, and the y-axis is the percentage of the amount of transactions. The colours in the bars differentiate the types of transactions.

It is worth discussing the distribution of transactions of type payout over time. From April 2018 until October 2018, payout transactions play a significant role in the mixer. However, its share started to decrease considerably from November 2018 to the point where it is almost not visible in the figure. The BestMixer administrators likely changed the mixing process in November, where the payouts to customers are probably only carried out by transactions of type 'payout, internal'.

As mentioned earlier, there was a significant difference in the number of transactions between June 2018 and July 2018. The results confirm that the mixer likely gained popularity, as the deposit transactions percentage is comparable for both months. At the end of June 2018, the administrators posted their videos on YouTube that explained how and why to use BestMixer; this could have helped make more potential clients aware of the existence of the mixer.

Another earlier observation that needed confirmation was the purpose of the transactions made before customers could use the mixer. This figure shows that all transactions made in February were either deposits or internal transactions. From this observation can be derived that it is likely that these transactions were made to have reserves in the mixer, as the money did not leave the mixer.

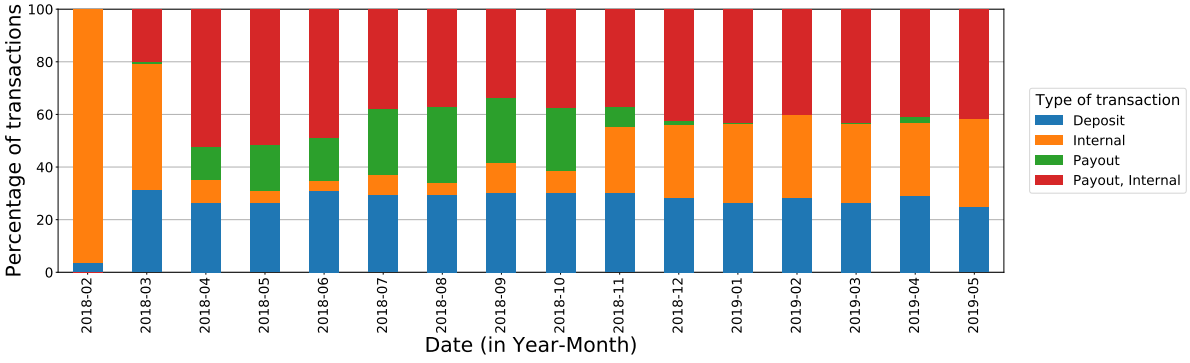


Figure 5.13: Distribution of transactions over time and distribution of types of transactions.

Online services are not restricted by time and location like physical stores: a mixer can be used 24/7 worldwide. It is interesting to see if there is a difference in the activity of the mixer throughout the day. Figure 5.14 demonstrates how transactions are distributed over the hours of a day. The x-axis shows the hour of the day; the y-axis shows the amount of transactions. The hour of the day was derived from the UNIX timestamp

of the transactions and is, therefore, according to Coordinated Universal Time (UTC). The figure reveals that transactions occur each hour of the day, and the total amount of transactions is always of magnitude 10^3. The transactions of types *Internal* and *Payout* only represent a small part of all transactions and are therefore almost not visible in this figure. The distribution has a sinusoidal shape: the amount of transactions decreases between 0 to 5, then increases until 17 and decreases again after until 23. The distribution with only the deposits follows this same pattern, so there is likely more activity around 17:00:00 UTC. During this timestamp, it is the evening for the right side of the prime meridian and office hours for most of the left part of the prime meridian.

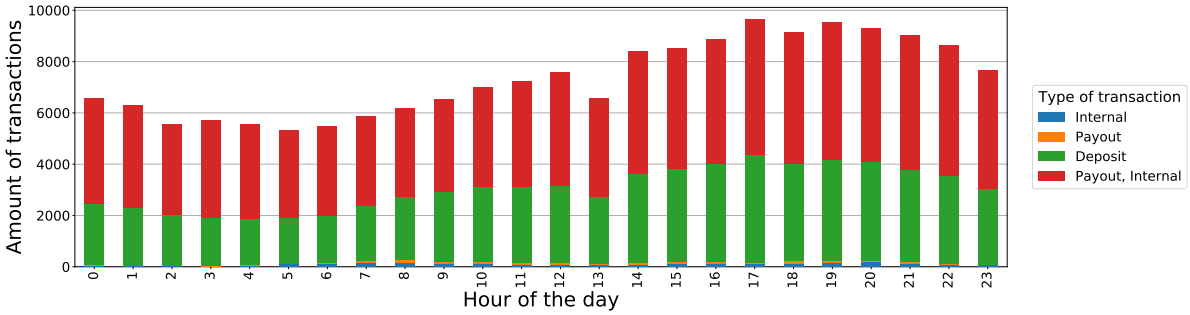


Figure 5.14: Distribution of amount of transaction made per hour and the distribution of type of transaction.

Fee Insights in the fee of transactions carried out by BestMixer addresses can help discover what kind of strategy was used for calculating the fee. For instance, BestMixer could have used a standard percentage based on the amount or size of the transaction, an automated calculated fee from a particular wallet service, or a different strategy. A pattern in the fee choice could help to create a new heuristic for attributing similar mixers based on the fee.

Multiple figures were made to help discover patterns in the choices of the fee, where only the transaction sent by BestMixer addresses were taken into consideration. Figure 5.15 displays the distribution of the total amount of fee paid per transaction. The x-axis represents the different magnitudes of the fees, and the y-axis shows the amount of transactions.

The figure shows that the fees are distributed over five different magnitudes, whereas there were nine different magnitudes for the values. This makes it unlikely that the mixer used a standard percentage based on the value of the transaction. It makes sense that the administrators did not choose to use a standard percentage, as miners decide to include a transaction in their block based on the ratio between the size of the transaction and the fee they can earn. Since a transaction with a higher value does not automatically make it larger in size, paying more for a large transaction is unnecessary to ensure its priority on the blockchain. Most transactions occurred with a fee in magnitude 10^{-5}, transactions with a fee in magnitude 0.01 occurred only twice.

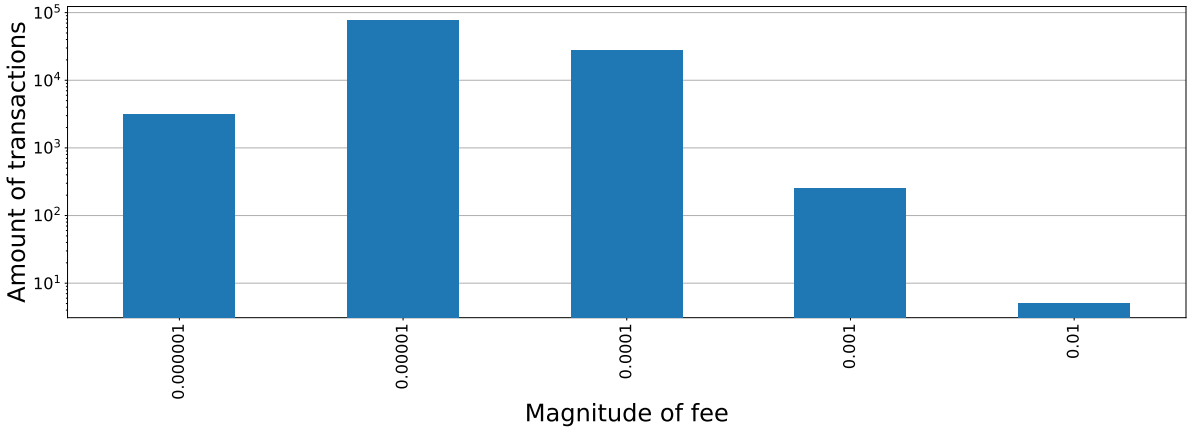


Figure 5.15: Distribution of magnitude of fee for all transactions.

The amount of fee compared to the total amount of transacted money can be expressed as the fee per-

centage of a transaction. Figure 5.16 shows the distribution of the fee percentage per transaction. The x-axis represents the different percentages, and the y-axis shows the amount of transactions on a logarithmic scale. The colours represent the different magnitudes of values. The figure demonstrates that a wide range of percentages has been used for paying the fee. This makes it unlikely that a constant fee percentage was used for the transactions. It is also unlikely that a constant percentage was used for all different magnitudes, as all the coloured lines are spread over the plot. The plot does show that for most transactions, less than 1% of the transaction value was paid.

Notable are the high percentages in the figure. For example, the highest fee paid for a transaction was 95% of the transaction value. This means that 95% of the input is paid to the miner and only 5% to the receiver. It seems unnecessary, and a waste of money to pay such a high percentage of the transaction value to fee, as such a large part of the money is lost to a random miner. An explanation for these high fees could be that the administrators wanted to transfer a specific amount of money to an address without receiving any spare change.

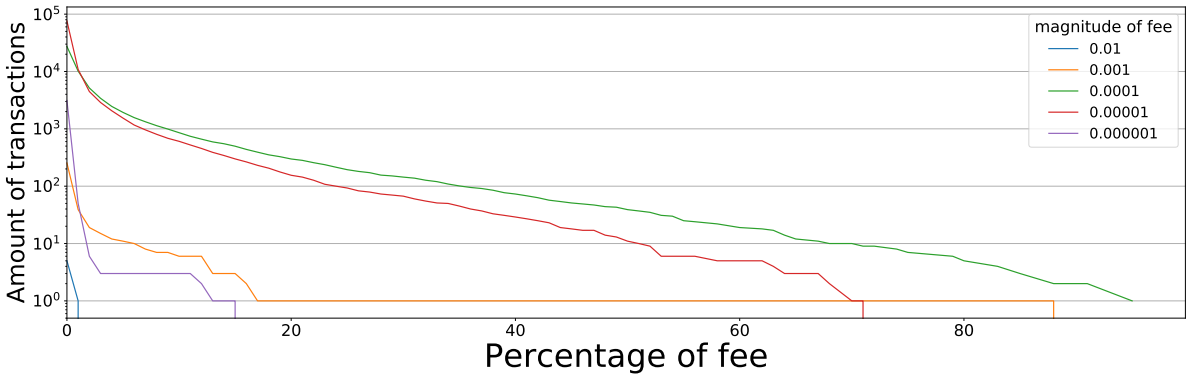


Figure 5.16: Distribution of fee percentage and fee magnitude for all transactions.

The fee is often determined based on the size of a transaction by standard fee calculators. It can help discover patterns to look at the amount of fee paid per size unit. This is displayed in Figure 5.17. The x-axis represents the different ratios between transaction size and fee, and the y-axis shows the amount of transactions. Again, the colours represent the different magnitudes of values.

From the plot it can be concluded that a standard fee per size was not used to calculate the fee for all transactions. The fees per size range from 0 to 250 satoshi/Byte, which is a large difference. It is also unlikely that a constant fee per size ratio was used for all different magnitudes, as the lines are not vertical. The line for magnitude 0.000001 is vertical, which means that transactions of magnitude 0.000001 use the same value/size ratio. It is interesting to see that the magnitudes of the transaction values are not divided over the entire plot: the lowest two magnitudes only occurs on the left side of the plot. The other two magnitudes are spread over the entire figure.

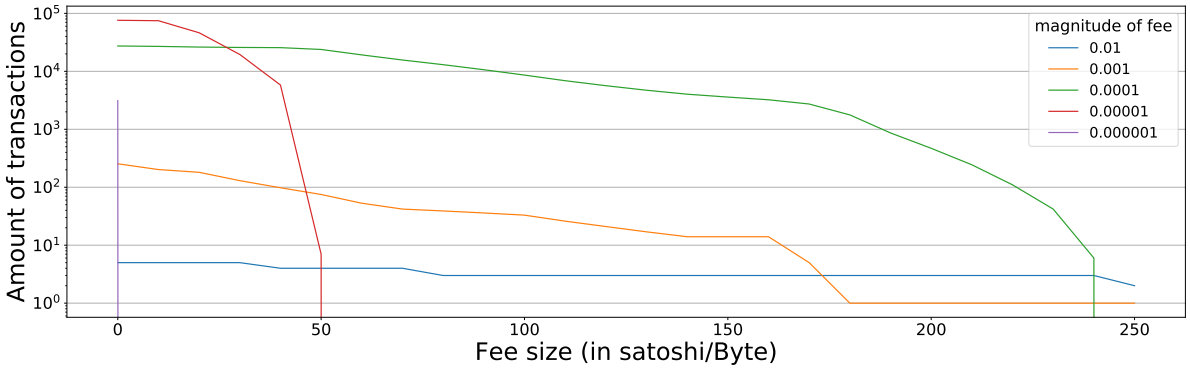


Figure 5.17: Distribution of magnitude of values and distribution of types of transactions.

The administrators of the mixer could have also chosen to use a different fee strategy for internal and payout transactions. Figure 5.18 shows the percentage of transactions per type of transaction for each mag-

nitude of the fee. The x-axis shows the different magnitudes of fees, and the y-axis shows the percentages. The colours represent the three different types, where for each magnitude, the sum of the percentages of the three types has to add up to 100%. The plot shows that the lowest two magnitudes contain no transactions of type payout, and the highest two magnitudes contain no transactions of type payout, internal. This could indicate that the administrators used a strategy where the fee differs for the different types of transactions they sent. Transactions of the type internal occur for all magnitudes.

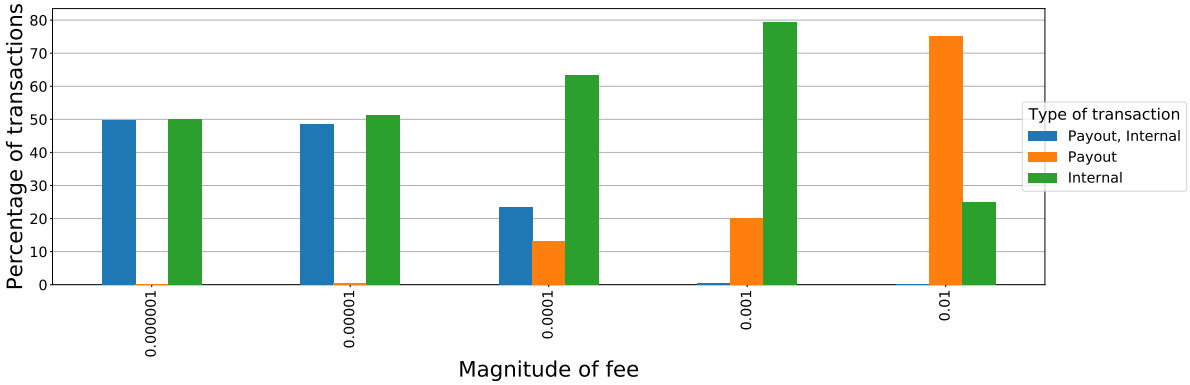


Figure 5.18: Distribution of magnitude of values and distribution of types of transactions.

From these figures, it is hard to confirm a pattern in how BestMixer determines the mixing fee: there seems not to be a consistent connection between the amount of fee and the fee percentage or size. It is possible that the administrators made changes in their fee strategy or that they used a fee strategy that was dependent on other external factors. For instance, they could base their fee on the average fee paid in the previous block, which makes finding a pattern more difficult and time-dependent.

Balance Figure 5.19 displays the amount of Bitcoins that are present on BestMixer addresses throughout time. This figure shows the date on the x-axis and the balance in Bitcoins on the y-axis. Three different lines show three different aspects of BestMixer's balance. The blue line shows the cumulative sum of all transactions per day, which implies how much reserves the mixer has at any point in time.The figure clearly shows that the reserves grow a little during the first six months and are kept around 25 BTC, with one peak at the end of July. After September 2018, the balance starts to become more unpredictable, with many high peaks and troughs. These sharp peaks can result from orders with such a delay that the payout is carried out the next day. The reserves grew on the deposit day but decreased again when the payout was made. Another reason for the peaks can be deposits from investors. The troughs have to be the result of withdrawals from the administrators, because the mixing process itself should never result in a lower balance.

The orange dashed line displays the weekly moving average of the daily cumulative sum. If all transactions in the mixing process were deposits and payouts, the balance would grow, which should show an upwards trend. This is clearly not the case in Figure 5.19. Therefore, it can be assumed that not all transactions are part of an automated mixing process. It could be possible that the administrators paid their salary manually, as the troughs differ in depth and frequency. The peaks and troughs are a lot less steep than those of the daily cumulative sum, which confirms that some of the peaks are the result of deposits and payouts from customers can be on different days.

The troughs could also be the result of other matters. For instance, the administrators reported on 2 April 2019 that the service had been attacked and that all payments were stopped until the service was fully restored. This day shows a significant decrease in the balance for both the daily cumulative sum and the weekly moving average. It is not clear whether these funds were stolen or if the administrators moved these coins themselves out of precaution.

It is worth discussing the height of the balance. After November 2018, the weekly moving average starts to grow above 200 BTC and rarely drops below this value. This means that the reserves of the mixer became higher, which could have facilitated mixing with higher deposits. However, the analysis on transactions over time showed that no increase in transaction magnitude occurred, so even though it was possible to mix deposits with higher values, customers did not use this.

The dotted green line illustrates the sum of deposits made per day, which clearly fluctuates throughout the time frame. Notable is that not all peaks from the sum of deposits result in a peak in the daily balance: during

those days, the amount of deposits also leaves the mixer the same day. This could indicate that customers want to receive back their deposits within the same day. One peak stands out as its sum reaches over 800 BTC in one day, whilst the daily cumulative sum only increases a few BTC. After November 2019, the sum of deposits never dropped below 20 BTC, so the turnover of BestMixer became substantial.

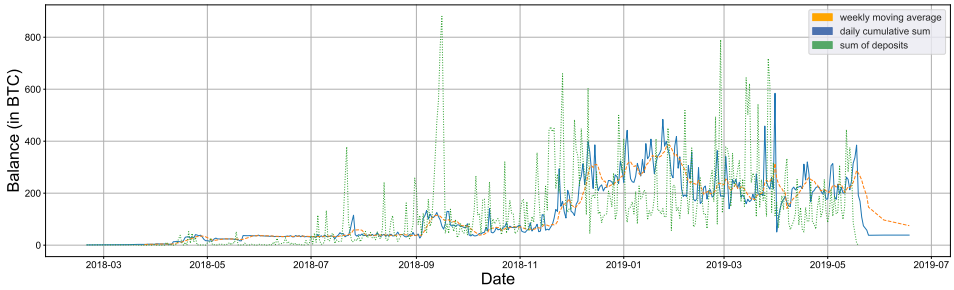


Figure 5.19: Balance of the mixer throughout time.

Correlations Correlations between transaction features could show if there are linear relations between certain aspects of transactions. Only transactions made by BestMixer addresses were considered because non-BestMixer addresses can distort these relations as they do not have to follow the same patterns. The scatter plots in Figures 5.21, 5.22 and 5.23 show the pairwise relation between mutation features. Noticeable is that all three figures contain scatter plots with lines parallel to either the horizontal or vertical axis. This means that the feature on that axis is in fact a constant, and not a variable. It is logical that the type feature is a constant because the data was separated on type. It is interesting to see that the feature *Sent addresses* is also a constant for the types *Payout* and *Payout, Internal*. This means that all addresses of these two types only sent transactions with one input address.

The three figures show little signs of linear relations. The scatter plots of types *Internal* and *Payout* do not reveal linear lines, the type *Payout, Internal* does show some linear lines, but these plots also contain outliers. Since only the type *Payout, Internal* potentially has high scores for the Pearson correlation, only this type was used for creating a correlation heatmap (Figure 5.20. Seven features are on both axes and create a triangular shape. The heatmap does contain one strong correlation: the relation between the fee size and fee have a correlation value of 0.8. This means that it is likely that the fee of transactions of type *Payout, Internal* was determined based on the size of a transaction. All other correlations are very weak.

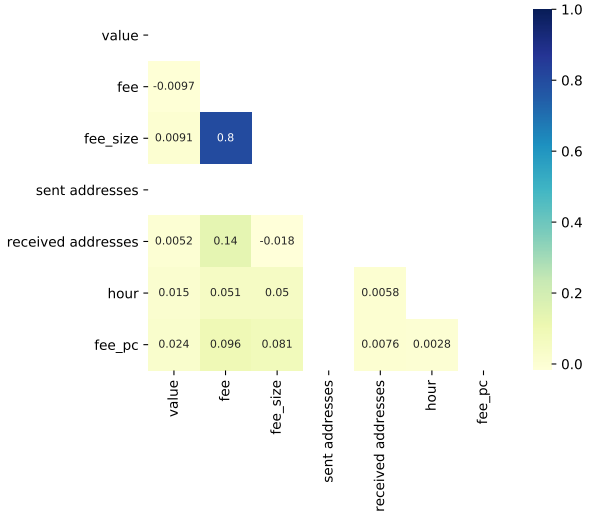


Figure 5.20: Heatmap of transaction features of transaction type *Payout, Internal*

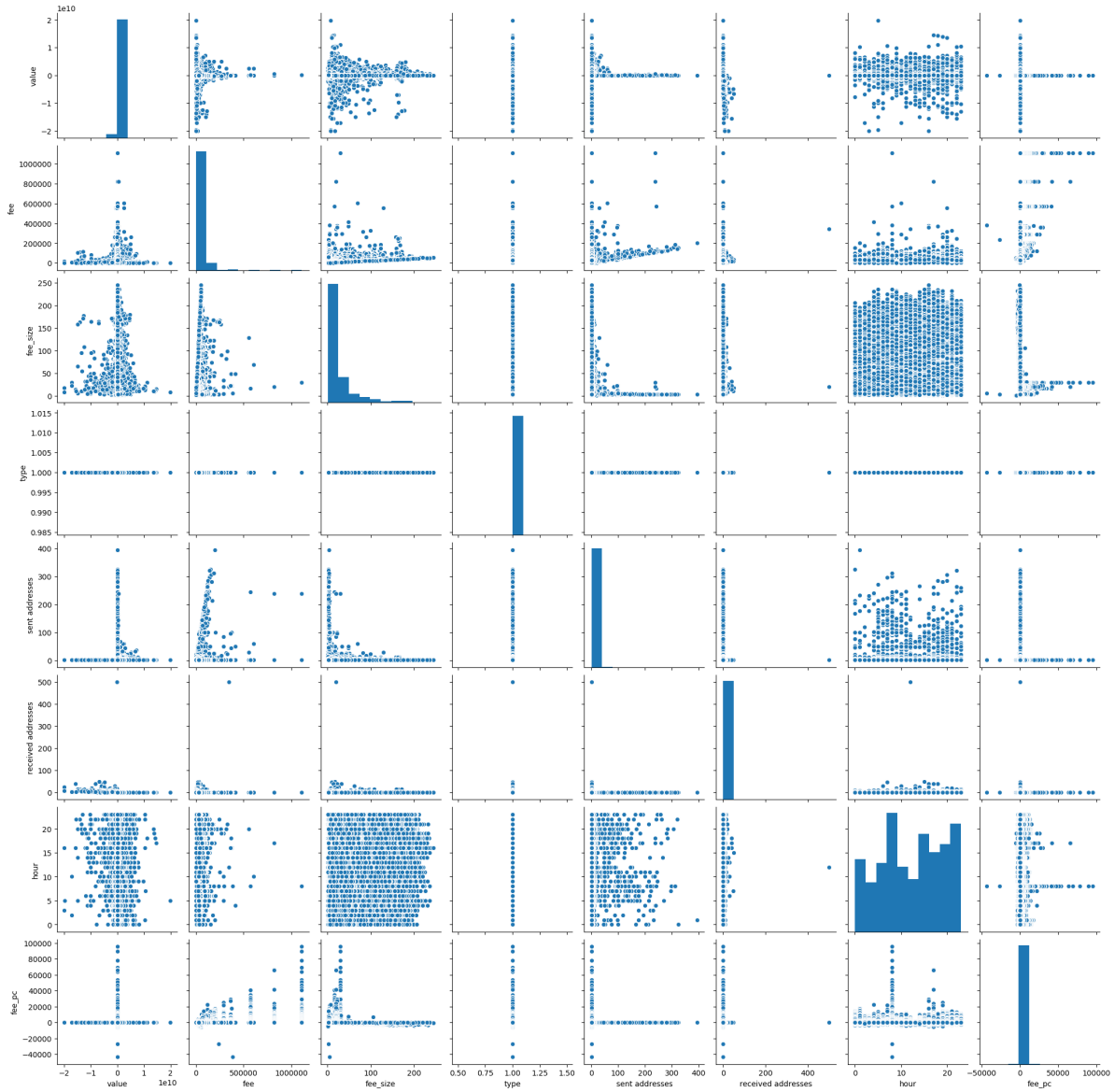


Figure 5.21: Scatter plots with mutation features of type *Internal*.

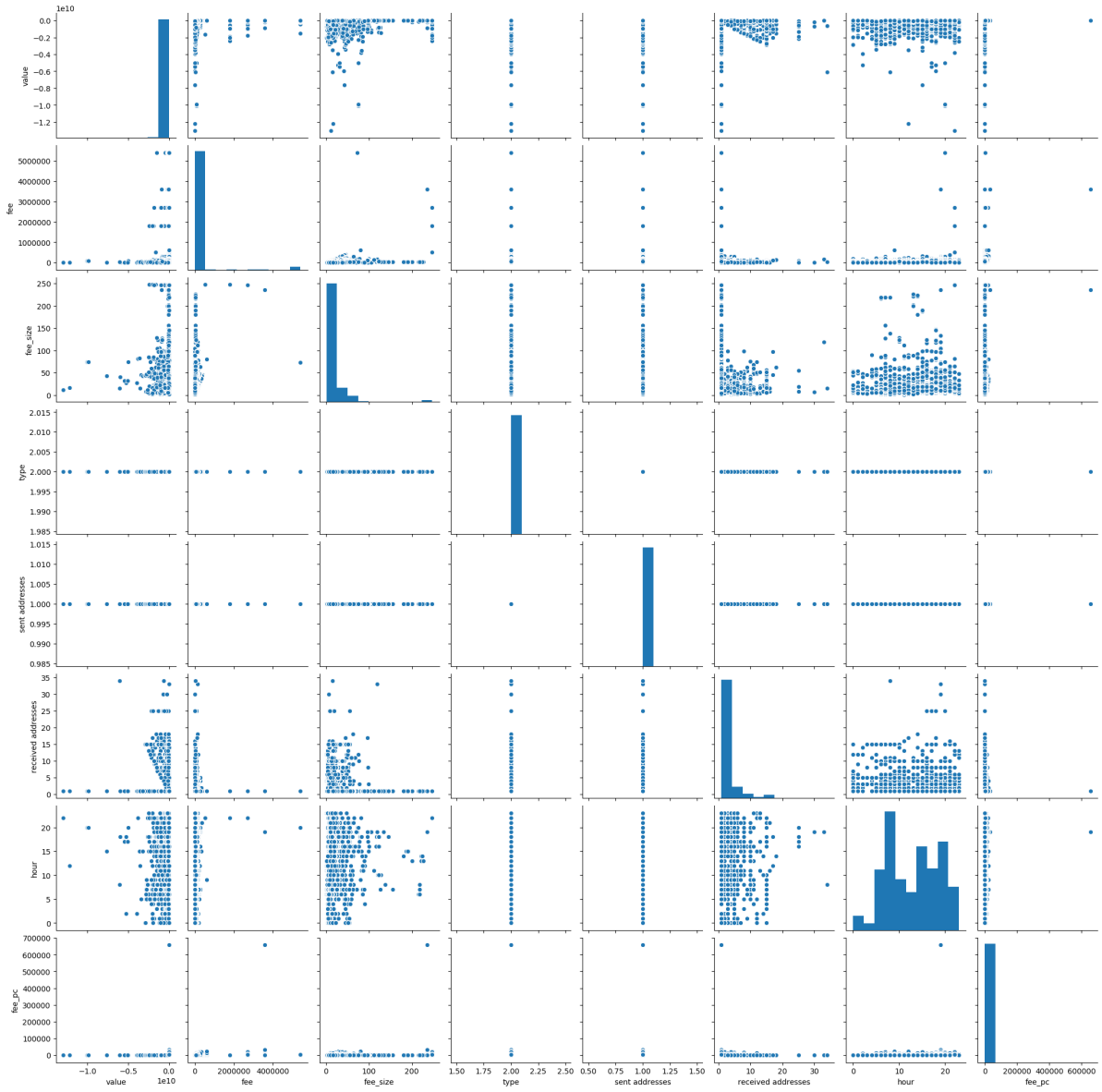


Figure 5.22: Scatter plots with mutation features of type *Payout*.

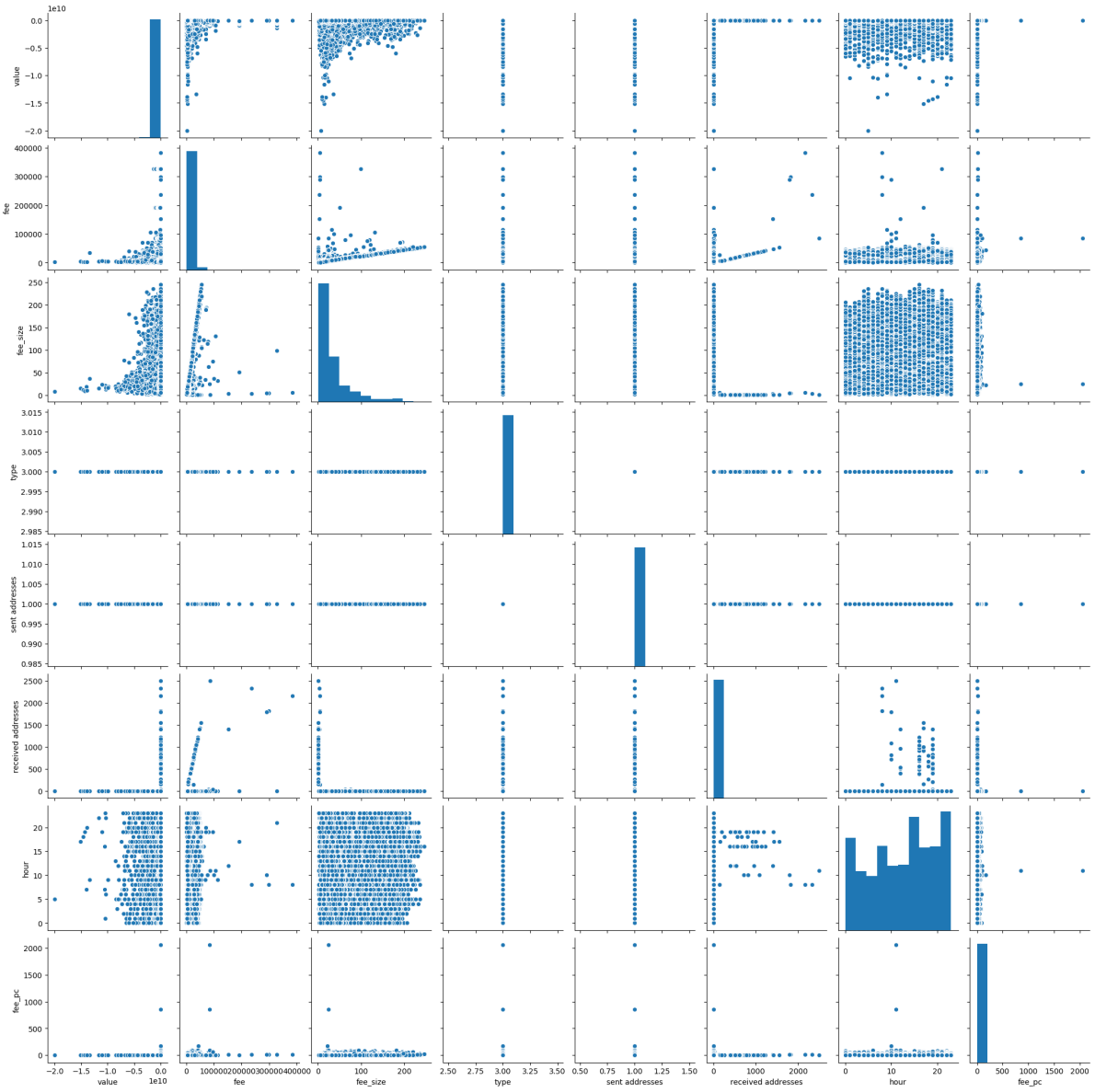


Figure 5.23: Scatter plots with mutation features of type *Payout, Internal*.

Principal Component Analysis It would be interesting to see if the different types of transactions can be separated based on their features. Therefore a PCA was performed on the transaction data. The first step of this analysis, where the number of components is determined, is depicted in the scree plot in Figure 5.24. The explained variance is displayed on the y-axis and the number of components on the x-axis. The plot shows that components 0 and 1 explain almost all variance, so these two components were used for the second part of the analysis.

The second step in the analysis is performing the dimension reduction to two components on the data. The results for each transaction are displayed in the PCA score plot in Figure 5.25. The component that explains most variance is displayed on the x-axis, the second component on the y-axis. The four different colours in the graph represent the four types of transactions, where the numbers correspond to the following types:

1. Deposit
2. Internal
3. Payout
4. Payout, Internal

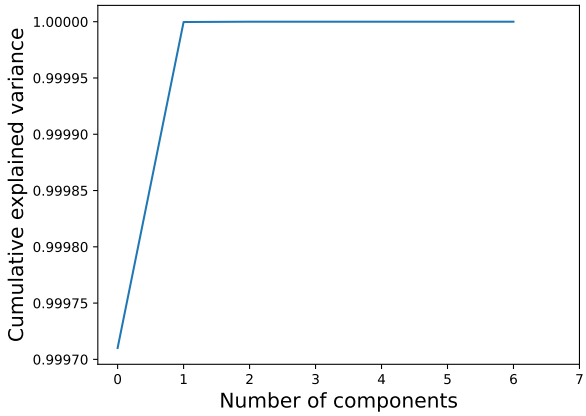


Figure 5.24: Scree plot of transaction variables.

In the plot, a clear distinction between type 1 and types 3 and 4 can be made at the centre of the x-axis. The points of type 2 seem to be distributed on both the negative and positive side of the x-axis, and they seem to be distributed almost vertically symmetric at the zero point. This makes sense because all internal transactions have almost identical points in the data, one from the sending address and one from the receiving address. The only difference is the value sign: positive for the receiving address and negative for the sending address. The covariance between the transaction feature 'value' and Component 1 is 1.00, which means that there is a perfect linear relation between Component 1 and the value. Most data points are centred towards the zero point: only a few points stretch to the minimum and maximum of the x-axis. Component 2 has a negative linear relation with the feature 'datetime', which makes sense as the amount of points on the y-axis increases from the top of the y-axis to the zero point.

The PCA analysis does not provide new insights into trends for transactions. The distinction of Component 1 primarily based on values is logical, as transactions of type payout and payout, internal are always negative, and transactions of type deposit are always positive. The distinction of Component 2, which is mostly based on the feature 'datetime', also shows what was already discovered in the analysis on transactions over time.

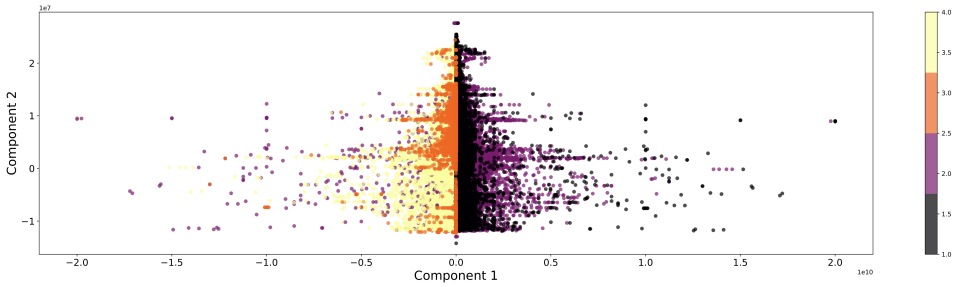


Figure 5.25: PCA score plot of transaction variables.

5.2.4. Sequences

Identifying sequences in the BestMixer data could reveal more information about the mixing process. For instance, it can show how long money stays in the mixer, from deposit to payout. Numerous aspects of sequences were explored: the length of sequences, the amount of branches from a begin address, the active period of a sequence and the difference types of transactions in sequences. All addresses without interaction with other BestMixer addresses were not considered in making the sequences.

Length of sequences The length of sequences is defined by the amount of addresses that are part of a chain of consecutive transactions. The first address is always an address that did not receive from an internal transaction; the last address sent no transactions to other BestMixer addresses. Figure 5.26 shows the distribution of sequence lengths. The x-axis shows the different lengths of sequences, the logarithmically scaled

y-axis the amount of sequences. The height of the line was determined with the following formula:

$$y = \text{number of sequences with a length of at most } x \tag{5.4}$$

The figure demonstrates that the length of sequences differs a lot: The shortest sequence contains two addresses, the longest 1,154. From this can be concluded that the sequences do not have a fixed length. The most occurring sequence length is two, and the steep decline at the beginning of the line shows that many sequences contain less than ten addresses. Another abrupt drop can be observed at sequence length 568. This means that there are relatively more sequences of length 568 than the surrounding lengths.

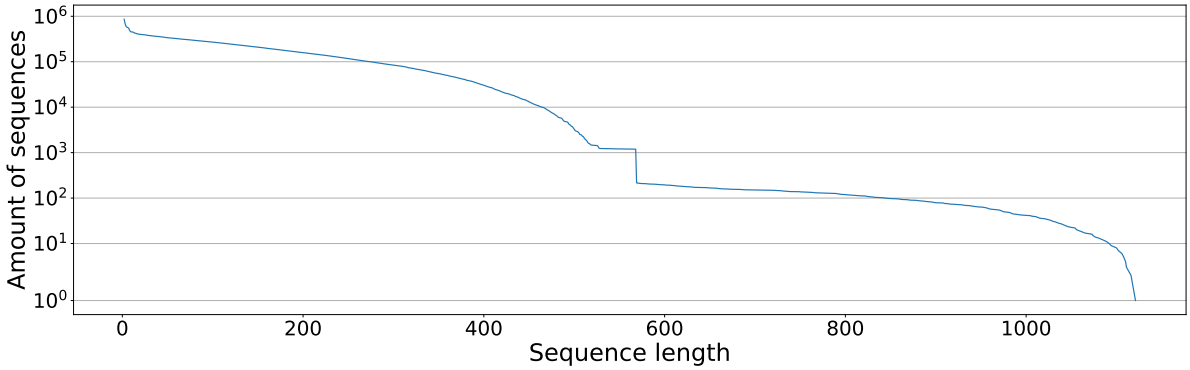


Figure 5.26: Distribution of sequence lengths.

Branches A transaction can have multiple outputs, which creates branches in the sequence. Whilst the length of the sequence reveals how long a sequence is, the amount of branches per begin address gives more information on the width of a sequence. When a begin address has multiple branches, it means that sequences have been merged. Figure 5.27 shows how the amount of branches in sequences is distributed. The x-axis represents the amount of branches, and the y-axis the amount of sequences. The height of the line was determined with the following formula:

$$y = \text{number of begin addresses with at most } x \text{ branches} \tag{5.5}$$

Noticeable in this figure is the shape of the line. First, there is a steep decline between 0 and 47, which means most begin addresses have 0 to 47 branches. Then, the line seems to be horizontal between 47 and 917, which means that there are not many begin addresses with 47 to 917 branches. Another steep decline is observed around 1000 branches. This means that the amount of merged sequences are mostly between 0 to 47 and 917 to 1000. Since the amount of branches often is larger than one, there have to be internal transactions in the chain of transactions.

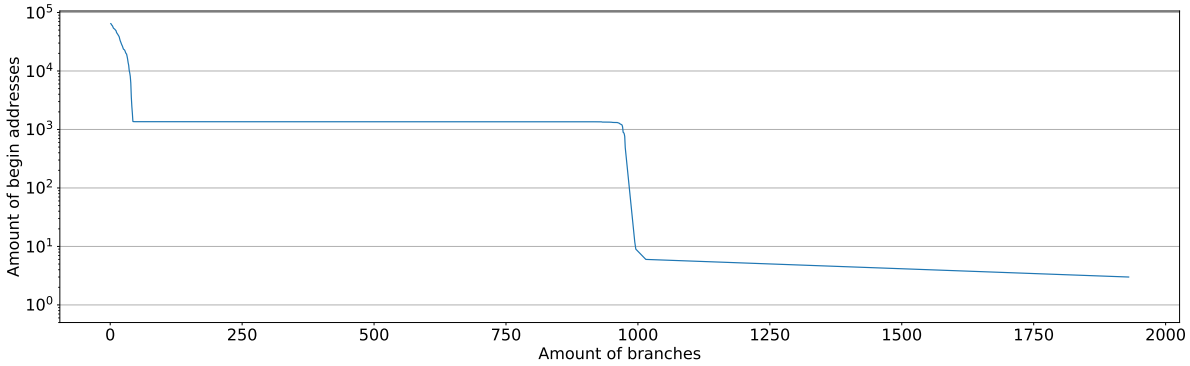


Figure 5.27: Distribution of amount of branches for all sequences.

Active period The active period of a sequence is the difference in time between the first and the last transaction in the sequence. Figure 5.28 exhibits the distribution of the active period of all sequences. The x-axis

represents the active period expressed in days, the y-axis the amount of sequences. The height of the line was determined with the following formula:

$$y = \text{number of sequences that are active at most } x \text{ days} \tag{5.6}$$

As can be seen from the figure, it is apparent that there is not a fixed active period for the sequences. The shortest sequences are active less than a day, whilst the longest is active for 114 days. What stands out in the figure is the horizontal line between 86 to 112 days, which means that none of the sequences had an active period of 86 to 112. After the horizontal line, a large decay is visible, which means that a significant group of sequences have a long active period. The shape of this figure and Figure 5.26 show some resemblance, which could mean that there exists a relation between the length of a sequence and the active period.

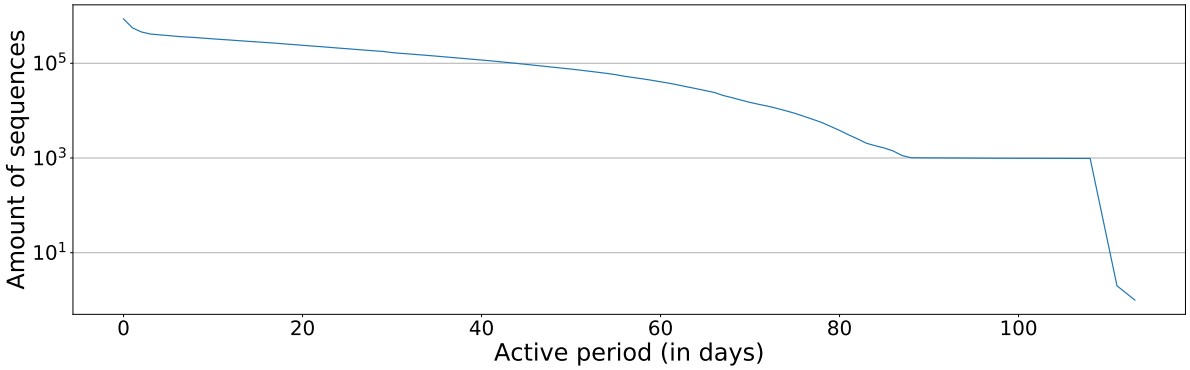


Figure 5.28: Distribution of amount of branches for all sequences.

Another interesting feature related to time and sequences is the average amount of time between transactions within a sequence. The distribution of the average time between transactions is illustrated in Figure 5.29. The x-axis represents the average time between two transactions expressed in hours, the y-axis the amount of sequences. The height of the line was determined with the following formula:

$$y = \text{number of sequences that have an average time between transactions of at most } x \text{ hours} \tag{5.7}$$

From this figure can be concluded that the automated process in mixing did not use a standard time between transactions because the averages differ a lot. This means that the sequences of BestMixer cannot be recognized on the blockchain based on the time between transactions.

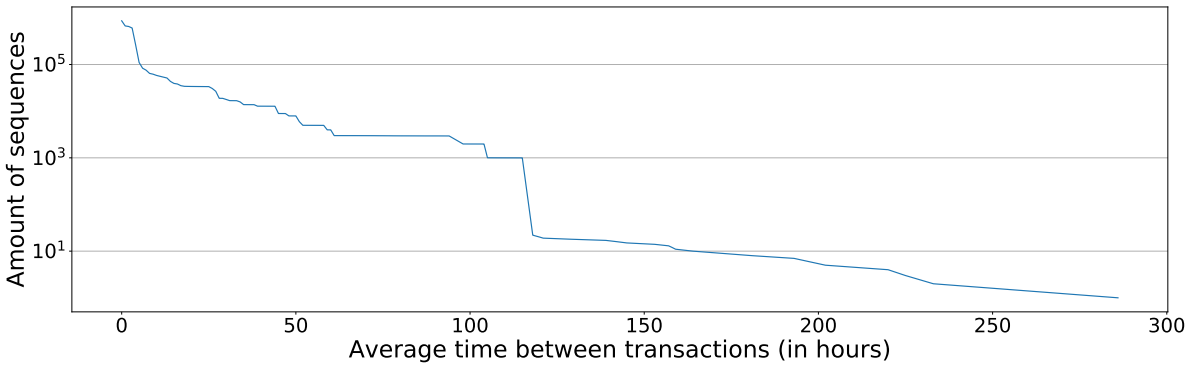


Figure 5.29: Distribution of average time between transactions for all sequences.

Types of Transactions All the transactions that are part of the sequence have had to receive from and sent to another BestMixer address, except for the transactions at the root or end of a sequence. There are two types of transactions identified in the previous section that consecutively transacted can form a sequence: *Payout, Internal* and *Internal* transactions. Transactions of type *Payout* can be the last transaction in a sequence. The pie chart in Figure 5.30 shows the distribution of the different types of transactions. The internal transactions are marked as either *Sent internal* or *Received internal*. The addresses of the latter type are also part of the

type *Payout, Internal*, which means that 59% of the transactions in all sequences are of this type. The ratio between *Sent Internal* and *Received Internal* is in line with the fact that there are many branches because more sent internal than received internal implies that these transactions were multi-input transactions.

A small number of transactions that did not adhere to the described types are marked as outliers. Some of these outliers resembled a *Payout, Internal* transactions, but the receiving internal address was the same as the sending address. This means that these addresses were reused, and therefore they were marked as outliers.

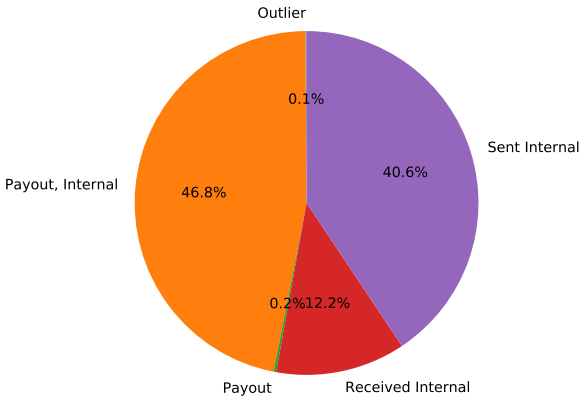


Figure 5.30: Distribution of types of transactions in sequences

Correlations Figure 5.32 was created to show the pairwise relation between sequence features. The figure contains scatter plots with all combinations of seven different features. There do not seem to be any features that result in perfect linear lines, but there are some linear relations between features as there are many scatter plots that contain a diagonal line with outliers. Therefore, it makes sense to use the Pearson correlation to calculate the correlation score.

The correlation heatmap in Figure 5.31 shows the strengths of the linear relation between sequence features. The high correlation between begin time and end time is evident, as the end time of the sequence should always be after the begin time. The almost perfect correlation between length and active period shows that longer sequences are active longer. This is not unexpected, as longer sequences mean more transactions are made, which takes time.

The matrix also shows two moderate correlations. The first moderate correlation is between begin value and end value. As there are multi-input transactions in sequences, the last address in a sequence may transact more than the first address. The second moderate correlation is between the begin time and active period. It is possible that the sequences became longer in a later stage of the mixer.

5.2.5. Clusters

A cluster contains all addresses that have interactions with each other. As mentioned before, in a well-performing mixing service, a deposit from and payout to the same customer should not be part of the same cluster. By exploring the clusters, it can be verified whether this also is the case for BestMixer. The size of clusters, the active periods, deposit and payout distribution, and correlations are explored in this exploration. The clusters containing only one address were removed in the clustering process, so the smallest cluster size is two addresses.

Size of clusters Where the sequence length provides insights into how long a chain of interacting addresses is, the size of clusters is the combination of the width en length of sequences that share addresses in their chains. The reconstruction of deposit and payout can benefit from large clusters if deposit and payouts never occur in the same cluster. Because if so, the large amount of addresses that belong to the same cluster as the deposit address can be written off as possible payout addresses. Figure 5.33 shows the distribution of cluster sizes, where the sizes are grouped in ranges to make the wide variety of sizes more manageable. The x-axis shows the size ranges, and the y-axis shows the amount of clusters. It is clear that most clusters are within the size range 1-10. This means that most clusters are relatively small, which is logical because of the short sequence lengths and the fact that addresses are not often reused. Figure 5.34 zooms in on the distribution of the smallest size range and shows that most clusters are of size 2, which means that in most cases, a deposit

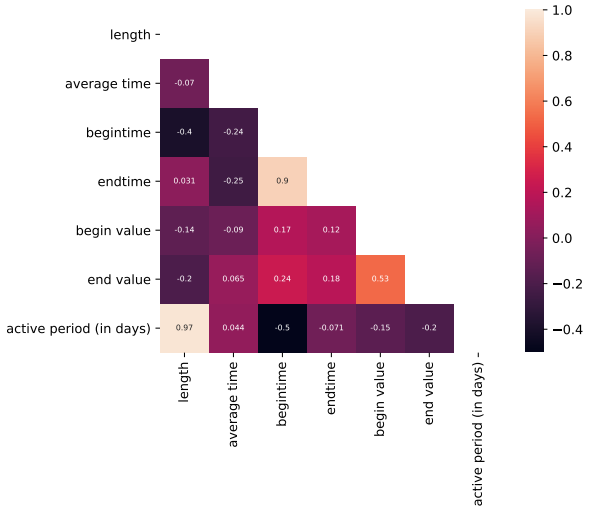


Figure 5.31: Correlation heatmap with sequence features.

was made to one BestMixer address, then sent internally to another BestMixer address and lastly, a payout to a non-BestMixer address was carried out.

It is important to highlight the cluster in size range 100,000+: this cluster contains 103,677 addresses, which is approximately 57% of all addresses. This large cluster makes it more unlikely that the deposits and payouts from all orders are divided over different clusters, as the majority of all addresses belong to the same cluster.

Another notable finding is the difference between the largest cluster and the largest sequence. This means that this cluster must contain multiple sequences. The large contrast can be the result of multi-input or multi-output transactions. All addresses in these transactions are merged in one cluster whilst they are split into multiple sequences.

Active periods Another interesting feature related to clusters is the amount of days that a cluster was used. This period is calculated by extracting the date of the first transaction received by the cluster from the last transaction sent by the cluster. Figure 5.35 presents the activity of all clusters, where the x-axis shows the date and the y-axis is used to stack all clusters, which were sorted on the first day of activity. Each cluster is depicted by a coloured line that is enclosed by two black dots, which indicate the begin and end time of the cluster.

Overall there is much difference in the active period of clusters. The lines in the figure differ a lot in length. As most lines are very short, most clusters had a short active period, but the few clusters with longer active periods can be much longer than the average active period. Most clusters only show activity within a period of 24 hours, which is a logical consequence of the combination of two factors: the establishment that most clusters consist of one sequence of size 2 and the fact that most sequences have an active period of less than 24. The cluster with the longest active period - 469 days - is longer than the amount of days the mixer offered its service. This longest active cluster is also the largest cluster.

The vertical steepness of the black dots shows how many clusters were simultaneously active, where a steeper line corresponds to more clusters active at the same time. This period between 2018-07 and 2018-11 shows a relatively steep linear line, which means that during these months, approximately the same amount of clusters was active. Most of the clusters during this time frame were active for relatively short periods, as the black dots show almost no coloured lines in between them. The period before 2018-07 shows a much flatter vertical line, but the horizontal coloured lines are much longer, which means that the amount of active clusters could still be equal to the amount of active clusters after 2018-07, but there is less variety in clusters. When comparing these results with the results of Figure 5.11, which showed the amount of transactions per month, it is clear that July 2018 did not only have a drastic change in the amount of transactions, it also had a significant change in amount of newly generated clusters. After 2018-11, a pattern more similar to the period before 2018-07 occurs with fewer clusters with longer activities.

Most interesting is the period between 2019-01 and 2019-04: there are almost no new clusters within this time period. The largest cluster is the only active cluster between 2 January 2019 and 13 March 2019,

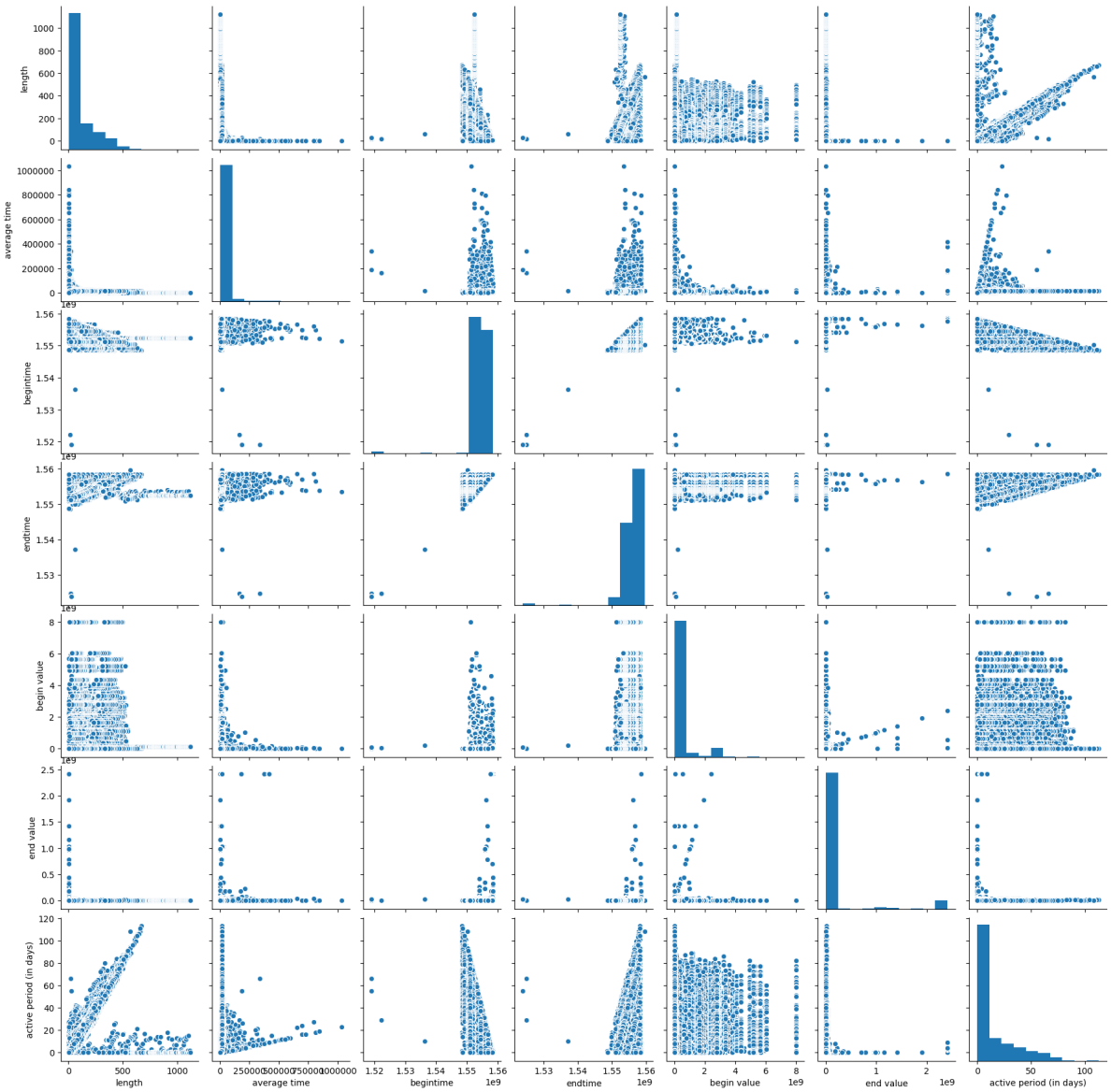


Figure 5.32: Scatter plots with sequence features.

except for three transactions that are clustered in a second cluster. During this period, 19,866 transactions were carried out where a BestMixer address was involved, so either the transactions could be linked to the largest cluster, or these transactions were made to addresses that had no interaction with other BestMixer addresses. Using the insights gained from the transaction over time, almost no payout transactions were made starting from January 2019, so it is not possible that funds were mixed without having interaction with other BestMixer addresses. Another possible explanation for the occurrence of only one cluster could be that a lot of large orders were placed during this time frame, which led to high demand on the liquidity of the mixer that resulted in the merging of clusters. However, the balance plot in Figure 5.19 shows that there are no abnormally high peaks in the sum of orders during this period, so this hypothesis can be rejected. Another explanation could be that the mixing procedure changed around 2 January 2019. Unfortunately, no change in the mixing procedure was announced by the administrator of the mixer in their blog post, so this is only speculation.

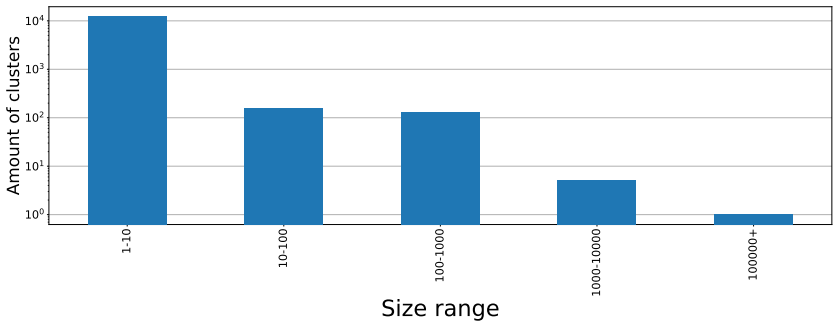


Figure 5.33: Distribution of size ranges of clusters.

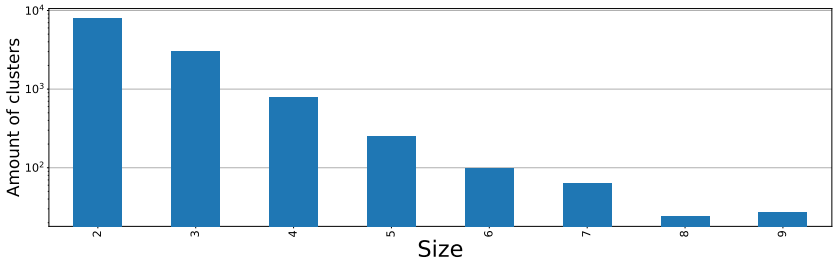
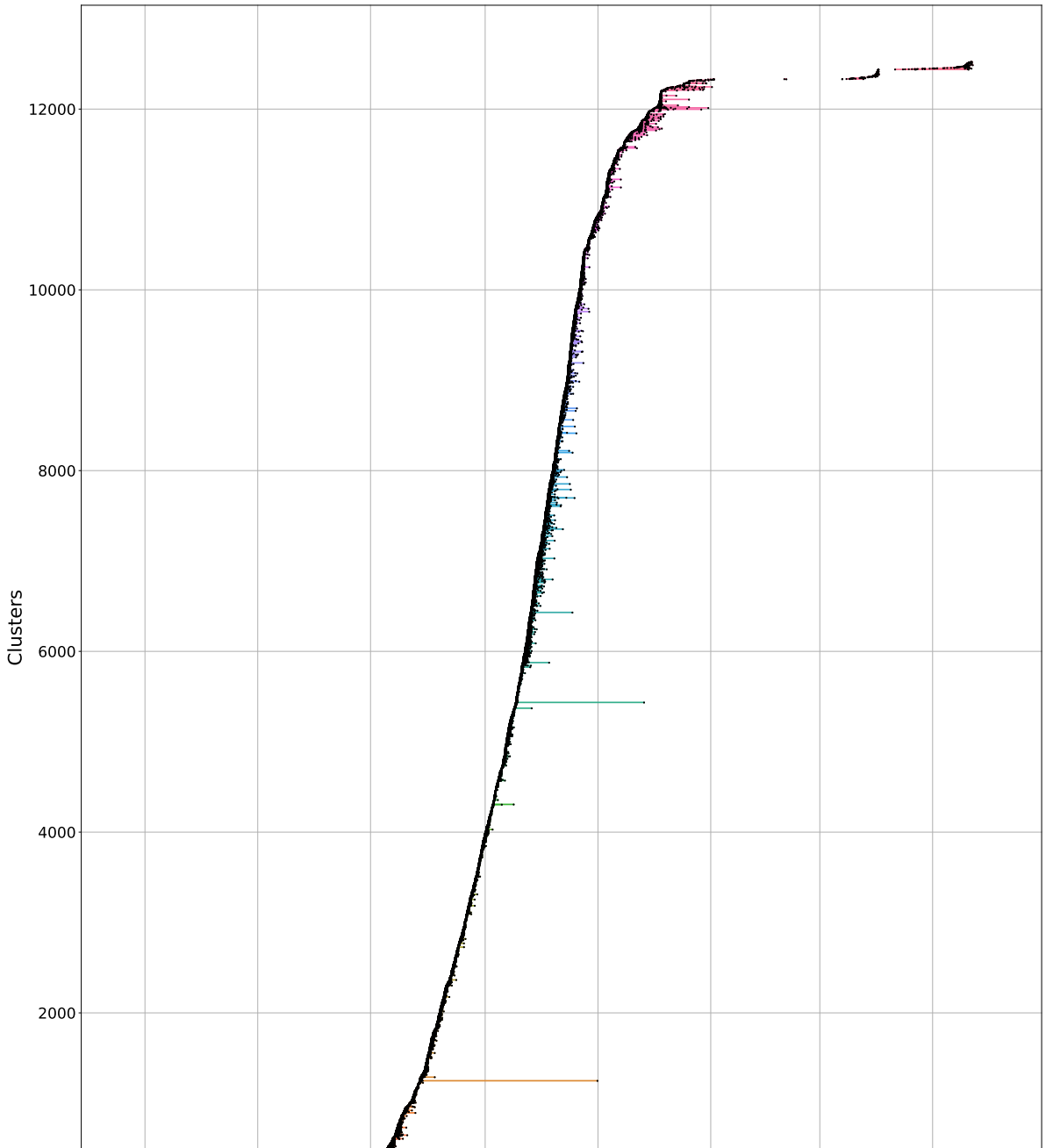


Figure 5.34: Distribution of size of clusters within size range 2 to 10.



Correlations Figure 5.36 displays multiple scatter plots which show the relation between cluster features. For example, the data points on the scatter plot with features *begin date* and *end date* on the axes show a diagonal line with only a few outliers. There are also a few scatter plots that only seem to have two data points. As there are a lot more than two data entries, it means that all data can be summarized in these two values. A straight line is easily drawn between two data points, so the linear relation score with Pearson's correlation coefficients between these values will likely be high.

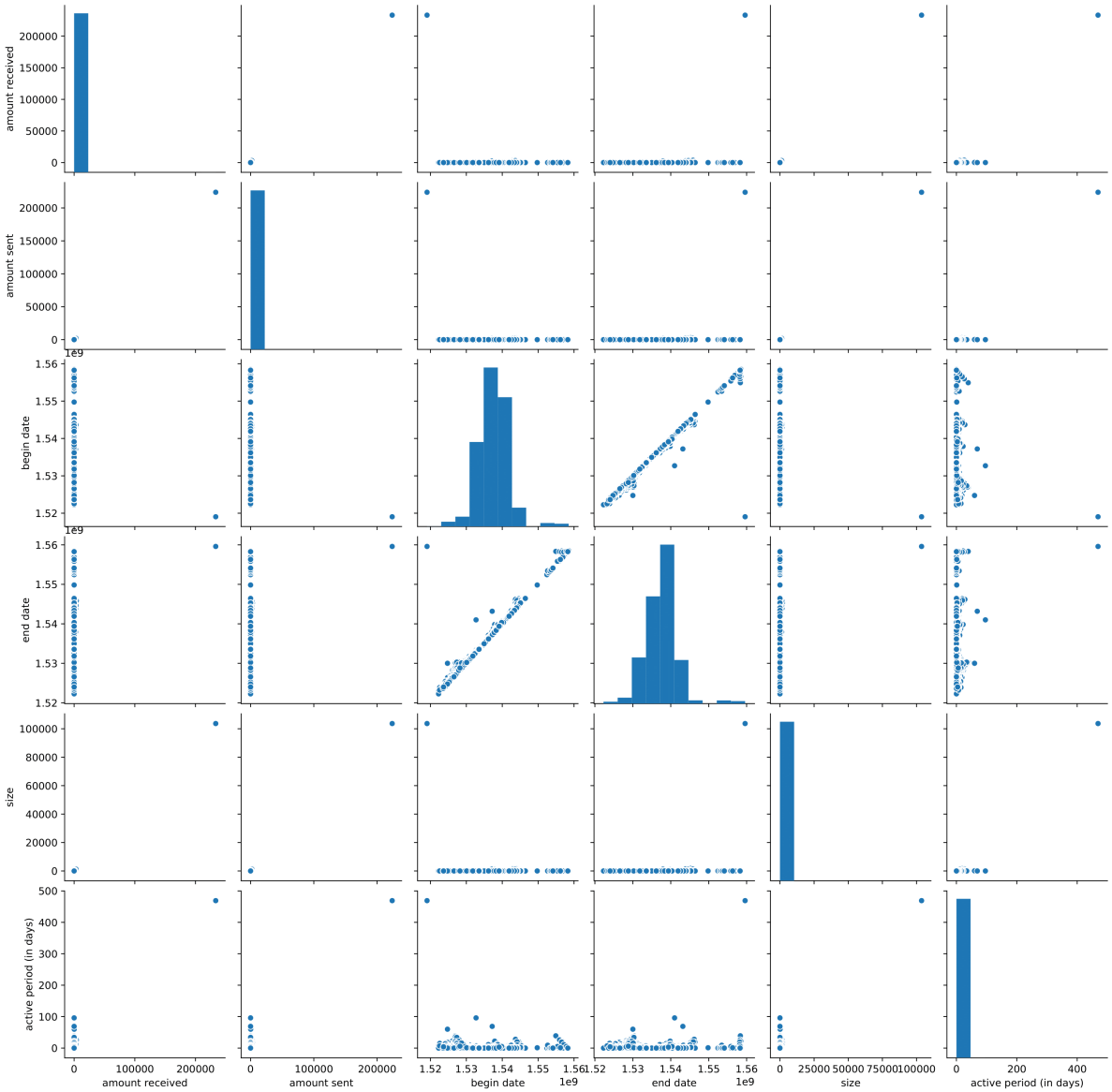


Figure 5.36: Scatter plots with cluster features.

The strengths of the linear relation between cluster features are depicted in the correlation heatmap in Figure 5.37. The heatmap shows that there are both very strong and very weak relations. The features *begin date* and *end date* are not correlated to any other features, but they have a high correlation with each other (0.99). This means that there is a linear connection between begin and end date of a cluster and that the end date of a cluster can be predicted with high probability from the begin date. The features *size*, *amount sent*, and *amount received* have perfect correlations with each other, which is not surprising as most addresses received and sent one transaction. These three features are strongly correlated with the feature active period on the positive side of the coefficient spectrum, which indicates that larger clusters have longer active periods.

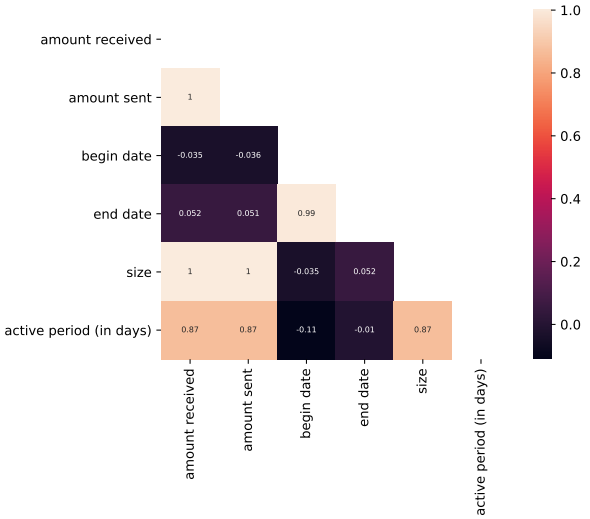


Figure 5.37: Correlation heatmap with cluster features.

5.3. Summary of results

In this chapter, an extensive exploratory analysis was conducted on four different aspects of BestMixer: the addresses, transactions, sequences and clusters. The insights gained from the results can help improve the attribution of centralized mixers. For instance, the analyses on addresses showed that there is little interaction with other addresses, in most cases, only one incoming and one outgoing transaction. Most addresses had interaction with at least one other BestMixer address. This means that attribution on reuse of addresses can be applied to BestMixer and similar mixers.

Suppose there is a strong presumption that a certain address belongs to a mixing service and this service uses a similar modus operandi as BestMixer. In that case, certain features can be derived from the blockchain (amount sent, amount received, unique amount sent, unique amount received) which can be used to predict the amount of payout addresses (perfect correlation), the amount of internally received addresses (moderate correlation), and amount deposits (high correlation). This can help in determining which addresses belong to a mixer, and which do not.

The observations on the addresses and transactions features combined result in strong indications for BestMixer to have used a peeling chain pattern. Therefore, the peeling chain heuristic and reuse heuristic could be applied to centralized mixers similar to BestMixer. However, a peeling chain on itself will not capture all BestMixer addresses. From the analyses on sequences, it became clear that there exist sequences of BestMixer transactions. In these sequences, transactions of type *Payout, Internal* are alternated by transactions of type Internal. A sequence of Payout, Internal transactions indicates a peeling chain, the internal transactions in the sequence merge sequences. This procedure is depicted in Figure 5.38. The first two transactions in the figure are of type *Payout, Internal*; the third transaction merges the values of three BestMixer addresses to another BestMixer address. After this, the peeling chain continues. Therefore, the peeling chain heuristic should be complemented with an additional heuristic that also captures the fact that the peeling chain can be interrupted by a multi-input transaction.

Analyzing the generated clusters show that there is a lot of difference in cluster size, but most clusters consist of less than ten addresses, and are active for less than 24 hours. There are strong correlations between cluster features, which makes it possible to predict the end date of a cluster if the begin date is known.

Another important aspect from the analysis is the economics of BestMixer, which can be derived from the analysis on address mutations and transactions. When considering the economics of the mixer, the mixer seemed to gain in popularity, especially after July 2018. The amount of reserves in the mixer started to fluctuate a lot after August 2018: before this date, there was almost a constant reserve of 20 BTC; after this date, the reserves fluctuated between 200 and 400 BTC. It seems that the distribution of transactions made per hour of the day has a sinusoidal shape, with a peak at 17:00:00 UTC and the lowest amount of transactions at 5:00:00 UTC. It is unlikely that BestMixer used a standard percentage or a standard fee per Byte ratio for all transactions. However, the administrators could have used different strategies for determining the fee over time, or for the different types of transactions.

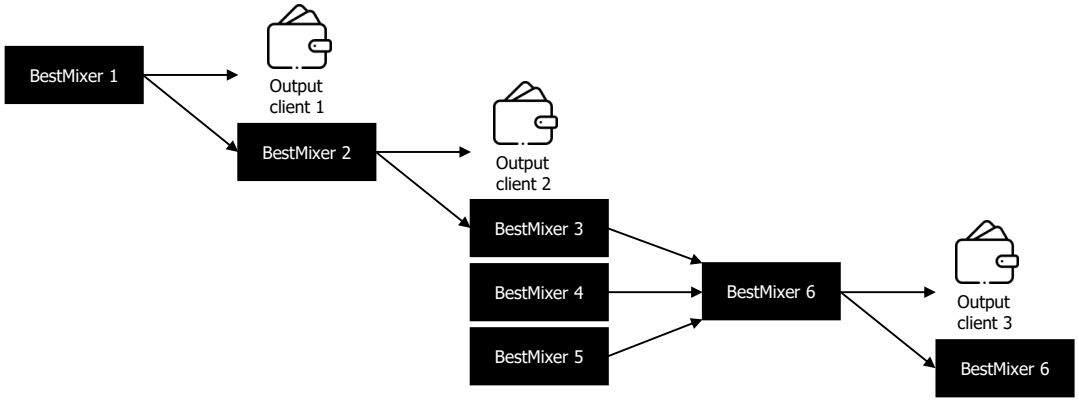


Figure 5.38: Example of pattern found in BestMixer's data

Some of these results are helpful input for the methodology of the second sub-question. Since most addresses are not reused, a Machine Learning approach for reconstructing the deposit and payout will likely not be able to predict the exact payout addresses belonging to a deposit. Another useful observation is that when the balance showed peaks in the sum of deposits, it did not necessarily show peaks in the daily cumulative sum. This means that approximately the same amount of input also left the mixer the same day, making it likely that many customers want to receive back their mixed coins within 24 hours. This can be used in deciding what payouts are more likely to belong to a certain deposit. It is likely that dusting attacks took place, and these transactions can be deleted from the list of possible deposit addresses. Another observation that can help in the reconstruction is the ratio between transactions of type *Deposit* and those of types *Payout* and *Payout, Internal*, which was 1:1.50. As this could indicate that not many customers chose to receive back their money on multiple output addresses, it can help determine the most likely payout.

Because there was only one active cluster between January 2019 and March 2019, in combination with the lack of payout transactions during this period, there seems to have been a change in the mixing strategy. This means that deposits and payouts during this period must be included in this large cluster. Therefore, it is not possible to filter out all transactions of addresses that belong to the same cluster as the deposit address since the payout address is also part of this cluster.

6

Reconstructing BestMixer's Deposits to Payouts

6.1. Methodology

This section describes how the sub-question *"To what extent does the demixing method filtering succeed in reconstructing payouts of BestMixer from the deposits?"* was answered. Even though all BestMixer addresses are known, it is not clear which deposit transaction belongs to which payout transaction(s). This is mainly caused by the number of orders, which result in an abundance of possible payout transactions. Reconstructing the relation between deposit and payout is very beneficial for Law Enforcement because it will help pursue traces of criminals on the blockchain even after the obfuscation attempt.

Two techniques for restoring the relation between deposits and payouts were considered. The only reconstruction technique described in literature is filtering, the other technique that was considered is Machine Learning. Unfortunately, the latter would not be able to exactly predict which payout transaction(s) belonged to which deposits. The results of the previous sub-research question show that most addresses are used only once within a very short period of time, which means that all addresses that were used besides the addresses that appear in the training data, would never be selected as payout addresses.

Even though filtering is not a time-efficient method, it is more likely to succeed in finding the correct payout transaction(s) than the machine learning approach. Therefore, a filtering approach was chosen for the reconstruction. Section 6.1.1 describes the data collection for each filtering request, followed by the filtering step in Section 6.1.2. Next, the combining of all remaining possible payout transactions is elaborated on in Section 6.1.3. Finally, Section 6.1.4 shows how the filtering method was tested.

6.1.1. Data collection

The filtering starts by collecting a list of all outgoing transactions from BestMixer and all incoming transactions. The list of outgoing transactions form the set of all possible payout transactions, and the list of incoming transactions establish the set of all possible deposit transactions. A list with all transactions per BestMixer address were collected for the previous sub-question already. These were used to extract the two different lists of transactions. The selection criteria for the deposit transactions list:
1. The value of the transaction is greater than 0,
2. The sending address does not equal an address in Wallet list,
3. The receiving address equals an address in Wallet list.
The selection criteria for the payout transactions list:
1. The value of the transaction is smaller than 0,
2. The receiving address does not equal an address in Wallet list,
3. The sending address equals an address in Wallet list.
The last selection criterion for both lists will always hold if the second criterion holds since at least one of the addresses in a transaction must belong to BestMixer. All transactions that remain after filtering are stored in two CSV files, one for the deposits and one for the payouts.

6.1.2. Filtering

In the next step, multiple filtering heuristics were created and used to decrease the amount of candidate payout transactions. These filtering criteria were derived from literature. The criteria need specific information from a deposit, for instance, the value of the deposit transaction and the time of the transaction.

Transaction Delay: The maximum delay between the deposit and the payout was set to 72 hours. This information can be captured in the following heuristic:
1. All payout transactions that are carried out more than 72 hours after the deposit can be removed.

Value of Withdrawn Bitcoins (a): The service fee of a BestMixer order was chosen by the customer within a range from 0.5 to 3%. This means that the sum of payout transaction values is maximally 99.5% of the deposited amount. This leads to the following filter heuristic:
2. All payout transaction values that are greater than 99.5% of the deposited value can be removed.

Value of Withdrawn Bitcoins (b): When customers wanted to receive their deposit back on multiple output addresses, they had to choose how to distribute the payout over these output addresses. The administrators set a minimum percentage of the total payout that each address had to receive, which can be translated in a heuristic:
3. All payout transaction values that are smaller than 0.1% of the deposited value can be removed.

Value of Deposited Bitcoins: When customers wanted to use the mixer, they had to transact at least the minimum deposit value. The administrators ignored all deposits lower than the minimum, as they are likely to be dusting attacks. This information can be translated to a heuristic that applies to the deposit transactions and not the payout transactions:
4. All deposit transaction values that are smaller than 0.001 BTC can be removed.

6.1.3. Combining

After the filtering step, combinations of the remaining payouts were made that can be the possible payouts to the deposit. These combinations have to adhere to the following two criteria and were derived from the available information in the literature. For space efficiency reasons, all possible combinations are first checked on these criteria and then added to a list, instead of creating all combinations in a list first and then removing them based on the criteria.

Value of Withdrawn Bitcoins (c): The service fee of a BestMixer order was chosen by the customer within a range from 0.5 to 3%. This means that the sum of payout transaction values is maximally 99.5% of the deposited amount and minimally 97%. This leads to the following combination heuristic:
5. The sum of the payout transaction values is between 97% and 99.5% of the deposited value.

Max Amount of Output Addresses: The service provided mixing up until 11 output addresses. The output addresses are the addresses that receive the mixed coins from the payout transactions. With this information, another combining heuristic was created:
6. There are at most 11 output addresses in the combination of payout transactions.

Creating all possible combinations is a time consuming process. The maximum size k of combinations is 11, and n is the number of remaining payouts. Creating all possible combinations with a fixed amount of addresses takes $\frac{n!}{(n-k)!*k!} = \frac{(n-k+1)(n-k+2)...(n-k+k)}{k!} = \frac{n^k+n^{k-1}}{k!}$ time steps. Since all combinations with less than 11 addresses also had to be created, the time steps are $\frac{n^k+n^{k-1}}{k!}! = n^k - n^{k-1}$, which makes the time complexity of creating all combinations $\mathcal{O}(n^k)$. Since k is 11, it can take days to generate all possible payout combinations if there are a lot of remaining payouts n. Therefore, altering the heuristics with tightener bounds was considered. In order to know what changes could be made, the data in the Order file was analyzed.

As the results of the exploratory analysis in the previous Chapter led to the observation that the high deposit peaks did not necessarily lead to peaks in the balance, it is possible that most customers want to receive their laundered money within the same day. To verify this, the time between deposits and payouts from the Order file has to be explored. Based on the results, the number of hours in the filtering criterion *Transaction Delay* could be decreased significantly. The smaller time frame could result in fewer possible

payouts after the filtering step.

Another aspect of the Orders data that can be explored is the amount of chosen output addresses. Analyzing the transactions in the previous Chapter revealed the ratio between transactions of deposit and payout types. As the ratio was relatively low, using a lower amount for the maximum size k of combinations could decrease the amount of possible combinations. To determine a new value for k, the amount of output addresses in each order has to be checked.

The last feature that was looked into is the percentage of the paid service fee. If a large part of the orders contained a service fee percentage that is higher than 97%, it could be considered to increase the fee percentage in criterion *Value of Withdrawn Bitcoins (c)*. This can decrease the amount of possible payouts.

Another approach that was applied for reducing the time it takes to create all combinations is depicted in Algorithm 1. After each iteration of creating all possible combinations of a certain size, all transactions that are too large to be added to the smallest sum of values from the combinations of the previous iteration are removed. For instance, if a deposit of value 100 resulted in transactions with the following possible payouts [20, 30, 98], the last payout would be removed in the iteration of size 3, as $max\_value = (dep\_value * 0.995) - min\_combi = (100 * 0,995) - 50 = 49.5$ and $98 > 49.5$.

Algorithm 1 Greedy approach for creating all combinations

Input: Payout transactions N; maximum size of combinations k; Deposit value d
Output: Possible payouts P

```
1:  function getCombinations(n, k, d)
2:      P ← []                                          ▷ initialize possible payouts
3:      max_val ← d * 0.995                             ▷ initialize maximum value of transactions
4:      for i ⊆ k do                                    ▷ loop through all possible sizes for combinations
5:          N ← n ⊆ N if n_value <= max_val             ▷ keep all transactions with lower values than max_value
6:          min_combi ← max_deposit
7:          for combination ⊆ combinations(i) do        ▷ loop through all possible combinations of N of size i
8:              sum ← 0
9:              for transaction ⊆ combination do        ▷ loop through all transactions in the combination
10:                 sum ← sum + transaction_value
11:             end for
12:             if sum < min_combi then
13:                 min_combi ← sum
14:             end if
15:             if sum <= (0.995 * dep_val) and sum >= (0.97 * dep_val) then
16:                 P ← P + combination
17:             end if
18:             max_val ← (dep_val * 0.995) − min_combi
19:         end for
20:     end for
21:
22:     return P                                        ▷ Return the list of possible payouts
23: end function
```

6.1.4. Testing

Testing was performed on the Order data, in which the deposit and output addresses of orders were both available. The output addresses were used to retrieve the corresponding payout transactions from the BestMixer transaction history. These payout transactions were combined with the correct deposit transactions to form test cases. The Order data also contains orders with requests to mix LiteCoin and Bitcoin Cash. Therefore, only orders placed with currency *BTC* were included as test cases. The status of an order was updated each time a customer visited their order page. The only status that ensures that an order was executed entirely is the *Complete* status: therefore, only the orders with status *Complete* were taken into consideration. More orders had likely been paid out than just those with status Complete, as customers did not necessarily have to check their order page to confirm that the order was complete: they could also check the balance of their payout address. However, the Complete status helped create a reliable test set that assured that both the

deposit and payout transactions are included in the transaction history of BestMixer, without checking the entire transaction history.

In the previous subsection was determined that some parameters will likely need some fine-tuning in order for the reconstruction to find the possible payouts in a feasible timeframe. For each test case, the filtering step was performed with multiple values for three parameters: the delay and the minimum and maximum service fee percentage. The amount of remaining possible payout transactions after the filtering step used to determine which parameters would likely perform well in the reconstruction.

After determining which parameters to use, the filtering step was performed with the new parameter combination, followed by the combining step. For each test case, the amount of payout possibilities were checked: the more possibilities, the less useful the filtering technique is. There are two important factors in deciding whether the reconstruction works well: the accuracy of the result and the amount of time it takes to create the results. The runtime of the algorithm is measured for all test cases. To measure the performance of the accuracy of the reconstruction, a contingency matrix was used (Table 6.1). In this matrix, each payout combination in the outcome of the reconstruction and the true payout are compared. Three commonly used evaluation metrics were used to measure how well the reconstruction works: the Probability of Detection (POD), the False Alarm Rate (FAR), and the Heidke Skill Score (HSS) (Gold et al., 2020). The POD score demonstrates how well the reconstruction works in finding the correct payout. A score of 0% means that the reconstruction never finds the correct payout and a score of 100% means that the reconstruction found all correct payouts. The FAR score demonstrates how often the reconstruction returns a possible payout combination that is ,in fact, incorrect, where the score of 0% means no incorrect possible payout combinations are returned, and a score of 100% means that all returned payouts are incorrect. The last metric, HSS, measures how well the reconstruction works compared to randomly guessing the correct payout. The range of the score is between 0 and 1, where the score 1 means that the reconstruction works perfect and 0 means that randomly guessing works just as well as the reconstruction. The calculations of these metrics are shown in Table 6.2.

		Reconstruction	
		Payout	No payout
Ground-truth	Payout	A	C
	No payout	B	D

Table 6.1: Contingency matrix

Metric	Formula
POD	$\frac{A}{A+C}$
FAR	$\frac{B}{A+B}$
HSS	$\frac{2(AD-BC)}{(A+C)(C+D)+(A+B)(B+D)}$

Table 6.2: Evaluation metrics

6.2. Results

The following section describes the results of the reconstruction of payouts to deposits. First, the results of the analysis on characteristics of orders are described. Then, the performance of the reconstruction method on the test data is given.

6.2.1. Exploration of Orders

The Order data contains 3,632 orders with the status 'Complete', and the coin is 'BTC'. During the process of retrieving all transactions to output addresses, it was discovered that there were a few output addresses that never received a transaction. This is unexpected, as it would be excepted that the status 'Complete' suggests that the mixing procedure is finished. It was also discovered that some payout addresses were used multiple times within a 72 hours time frame, which makes the possible payout combination ambiguous: it could not be said with certainty which transaction belonged to which order. Therefore it was decided to remove these transactions from the data, which resulted in 2870 test cases. For all remaining orders, the delay, amount of

output addresses and service fee were explored.

Transfer Delay To determine how many payouts were paid within a fixed delay, the cumulative sum of all payouts that were carried out within a delay was calculated. The cumulative sum was then transformed to a percentage to clarify the effect of increasing the time steps. The result of this calculation is shown in Figure 6.1. This figure highlights that 90% of the customers that placed orders in the data chose a delay within 24 hours. The slope of the cumulative growth starts steep, with 66% of the orders paid back within two hours, and flattens more after each hour, starting from 2 to 72.

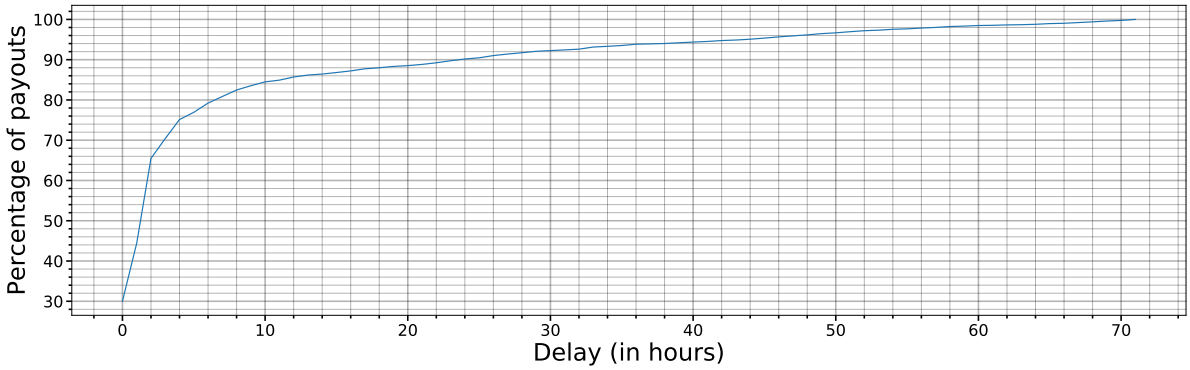


Figure 6.1: Cumulative sum of percentage of payouts paid within the delay.

To determine how to fine-tune this parameter, the values 2, 10, 15, 20, 24 and 72 were used to test the efficiency of decreasing the delay. Figure 6.2 shows the distribution of the different values for the delay. The x-axis represents the delays, and the y-axis shows the amount of candidate payout transactions. From the figure it can be seen that there is a clear decrease in the median of candidate payouts when the delay time decreases. Also, the distance between the whiskers that display the minimum and maximum is a lot bigger for the delay of 72 hours, with a maximum of almost 1400 candidate payouts, than for all the other delays. Notable is that the number of outliers increases as the delay becomes smaller, which is logical as the box sizes are also smaller. This means that a smaller delay leads to smaller differences in the number of payout candidates, which is desirable as this makes the reconstruction more consistent in terms of the time it takes to run the code.

Figure 6.1 shows an increase in accuracy when the delay is increased, whilst the possible efficiency of the reconstruction decreases when the delay is increased, according to Figure 6.2. When both accuracy and efficiency are taken into consideration, there seems to be an acceptable balance between the two when the delay is set to 24 hours. With this new delay, 91% of the correct payout candidates could be found whilst the maximum of the amount of candidate payouts is equal to the median of the original delay of 72 hours.

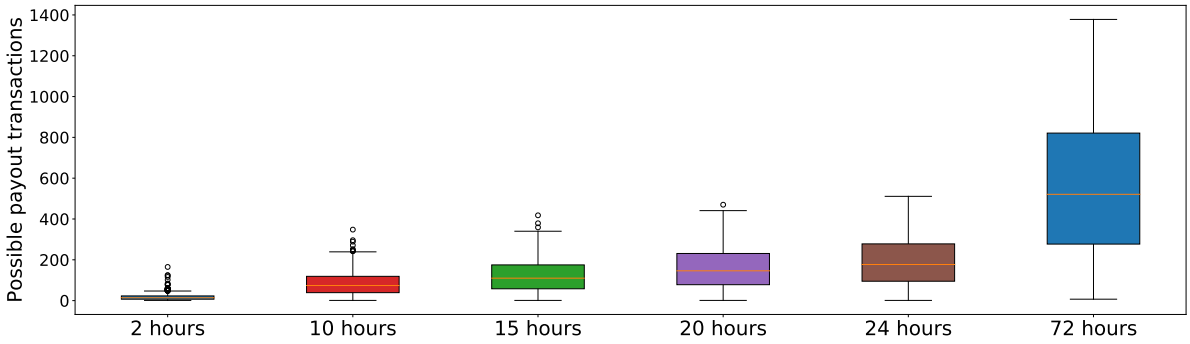


Figure 6.2: Distribution of amount of candidate payout transactions after filtering with different delays.

Amount of Output Addresses The value for the parameter maximum amount of output addresses k has the most effect on the code's runtime, as the exponent of the time complexity depends entirely on this parameter. Therefore, using a smaller value for k would make the reconstruction much faster. Figure 6.4 shows what percentage of orders have at most a certain amount of output addresses. For instance, 83% of the orders have

at most one output address, whilst approximately 95% of the orders have at most two output addresses. This means that if the parameter k is set to 2 instead of 11, 95% of the orders will have the correct payout included in the set of possible payouts. When both accuracy and efficiency are considered, it seems that when the parameter is set to four, the possible payout set can be generated within a feasible time frame and will contain the correct payout with 99% certainty. Therefore, the parameter maximum amount of output addresses k was set to 4 instead of 11.

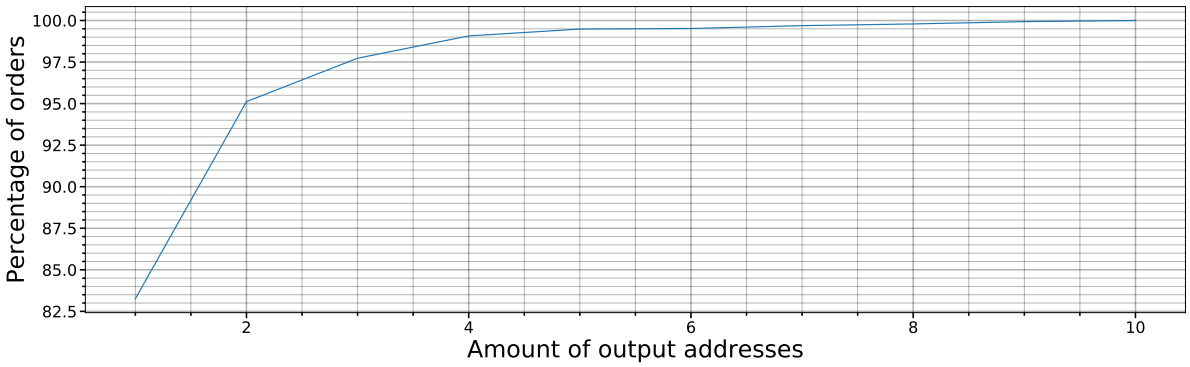


Figure 6.3: Cumulative sum of percentage of orders with less or equal amount of output addresses.

Service Fee The service fee was used as a parameter in both the filtering and the combining step, which makes it an essential feature in the reconstruction. The maximum value of the service fee is used in both the filtering and the combining, whereas the minimum value of the service fee is only used in the combination step. If the difference between the minimum and the maximum value is smaller than the original 2.5%, the number of possible payout combinations can be reduced significantly. Figure 6.4 shows the percentage of orders that paid at least a certain amount of fee. The amount of fee is depicted on the x-axis, the percentage of orders on the y-axis.

What stands out in the figure is the range of the found service fees. The range of the service fee that could be chosen by the customers was between 0.5 and 3%, but the largest fee paid by a customer was 93.3%, and nine customers paid only 0.3% service fee. This means that the current reconstruction tool will not accurately find all the payouts: only 71% of these orders fit in the service fee window. Therefore, to increase the probability of finding the correct payout combination, the service fee range is expanded from 0.5% to 5%, as this will include more than 90% of the correct payouts.

Figure 6.4: Cumulative sum of percentage of orders with less or equal amount of output addresses.

6.2.2. Evaluation of Reconstruction

The reconstruction was used to run all test cases, and all test cases returned the list of possible payouts within 49 minutes, whilst the mean runtime was 6 minutes and 46 seconds. If the amount of maximum output addresses is increased, finding the possible payouts can likely take more than a few hours. The outcomes were used to fill in the contingency matrix in Table 6.3. This table is quite revealing in several ways. First,

Since there were 2,870 test cases it is clear that for most test cases the correct payout combination is included. Secondly, the numbers in the second row of the matrix are significantly larger than in the first row. On average, each test case would return 3,062,719 possible payout combinations, of which only one is the correct one, which explains the high number in the leftmost column. All possible reconstructions that were not returned by the reconstruction are considered to be no payout. Since the mixer made 104,354 transactions of type *Payout*, for each test case, there are $104,354^4$ minus the amount of returned possible payouts that are filtered out and are therefor labeled as no payouts.

		Reconstruction	
		payout	no payout
Real payout	payout	2,621	249
	no payout	87,900,042,415	3^{23}

Table 6.3: Contingency matrix with all test cases

The contingency matrix was used to calculate the POD, FAR and HSS, of which the resulting scores are displayed in Table 6.4. The score of the POD metric is very high, which means that the reconstruction finds most of the correct payout. However, the value of the FAR metric is even higher. This indicates that even though the reconstruction correctly finds most payouts, it returns even more payout combinations that are incorrect. This is not unexpected, as the amount of payouts per test case is very high. The last metric that is shown in the table is very near to 0, which indicates that this tool is only slightly better in finding the correct payout combination than if the payout was found by using chance. This is a very disappointing outcome, as it suggests that the reconstruction performs very poor on the available data and is therefore insufficient in demixing BestMixer orders placed via Clear Web.

Metric	Formula
POD	0.913
FAR	0.999999
HSS	0.00000000596

Table 6.4: Evaluation metrics for reconstruction BestMixer

The analysis on all BestMixer transactions in Section 5.2.3 showed that the distribution of the magnitude of transaction values was Poisson shaped. This means that deposits with a magnitude of 0.01 will likely have more candidate payout transactions than deposits with a magnitude of 0.001. The reconstruction is mostly based on parameters that customers could choose when using the mixer and are not made public. However, the value of the deposit is known. If the reconstruction tool does prove to be useful for deposits with smaller values, the effectiveness of the reconstruction can differ based on the deposit value. To test this theory, the test cases were split based on the magnitude of their deposit value and the smallest possible deposit magnitude was used: deposits with a magnitude of 0.001. The outcomes of the 454 remaining test cases are placed in the contingency matrix in Table 6.5. The average amount of returned possible payouts is 36,878, which is still too many for a sufficient demixer. Unfortunately, this means that the reconstruction is also not useful for smaller deposits.

		Reconstruction	
		payout	no payout
Real payout	payout	388	66
	no payout	16,742,567	7^{21}

Table 6.5: Contingency matrix test cases with deposit value of magnitude 0.001

The reconstruction was tested on the available Order data of the mixer. All these orders were placed during a period when the mixer was very popular. It is possible that the tool performs better on deposits placed in the earliest months of the mixer, as there were a lot fewer transactions. Therefore, deposit transactions that were placed between 28 March 2018 and 1 June 2018 were used to test how many possible payout combinations would be returned. In total, 3,174 deposits were identified that were placed within the time frame. Figure

6.5 shows the percentage of deposits that resulted in fewer or equal returned payout combinations. The amount of returned payouts is depicted on the x-axis, the percentage of deposits on the y-axis. The plot shows that approximately 90% of the deposits have 50 or less potential payouts. On average, the reconstruction presented 37 possible payout combinations, which is a thousand times smaller than the returned possible payouts for the orders with the smallest magnitude. The largest set of possible payouts was 7,234, and 1,847 deposits returned 0 possible payout combinations. From this figure can be concluded that it can be helpful to use the tool for reconstructing deposit-payout combinations that were placed before 1 June 2018. As there is no ground-truth data available of this time period, it is unclear whether the correct payout combination is included in these sets.

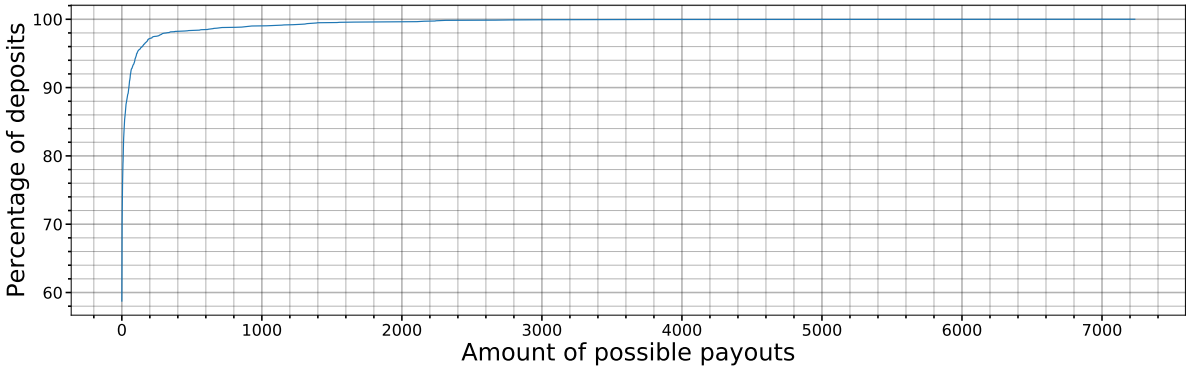


Figure 6.5: Cumulative sum of percentage of orders with less or equal amount of returned payout combinations.

7

Discussion

In this chapter, the results of this research will be discussed. First, the results are interpreted in Section 7.1. The contributions of this research are summarized in Section 7.2, followed by a description of the limitations regarding the research in Section 7.3. Lastly, recommendations for future research are provided in Section 7.5.

7.1. Discussion of results

The results of this study are twofold: the exploratory analysis shows new insights for the attribution of centralized mixers, and the reconstruction attempt sheds light on order characteristics and how well BestMixer obfuscates the money of customers. The first part of this research focused on answering the first sub-question, the second part on answering sub-question 2. The findings are discussed in separate subsections.

7.1.1. Sub-question 1

The results show that most addresses of BestMixer are used for only one incoming and one outgoing transaction, which means that there is almost no reuse of addresses. This is a logical and strategic choice from the administrators, as reuse of addresses can make linking addresses controlled by the same user easier (Zhang et al., 2020), which permits tracking a money-laundering service. For example, it was discovered that mixing services DarkLaunder, Bitlaunder, and CoinMixer used a central address that was used in the mixing procedure of orders, which made it possible to find patterns and addresses of other users of the mixers (de Balthasar & Hernandez-Castro, 2017).

Other results of this study indicate that BestMixer uses a peeling chain pattern to distribute payouts to customers. The first indicator for the peeling chain pattern is that most BestMixer addresses have interactions with other BestMixer addresses. The second indicator is that most transactions that were sent or received by BestMixer are of type *Payout, Internal*. These transactions contain one input address that is part of BestMixer, and two output addresses: one non-BestMixer address and one BestMixer address. The interaction between BestMixer addresses represents the shackles in the chain. Money was 'peeled away' from the mixer through the interaction with non-BestMixer addresses. The peeling chain pattern has been discovered in mixing services Helix and Alphabay (de Balthasar & Hernandez-Castro, 2017), but BestMixer differs from the standard peeling chain by combining the peeling chain with multi-input transactions of type *Internal*. This new insight into how a peeling chain is combined with internal transactions in BestMixer can help in clustering other mixing services by searching for comparable patterns.

Additionally, the results show a gain in popularity of the mixer, as the number of transactions starts to increase over time. More transactions make demixing a lot more complicated. This can be compared to the mixing process of a decentralized mixer: more inputs and outputs in a mixing procedure means more possible combinations of inputs and outputs, which makes finding the true combination much more difficult (Ziegeldorf et al., 2018). The same holds for BestMixer, as more transactions mean more possible payout candidates, which leads to more possible combinations of deposits and payouts.

The clustering method demonstrates that more than 50% of all addresses are contained in one large cluster, whilst the mode of cluster sizes is 2. The difference in cluster sizes could perhaps be the result of the three different pools that BestMixer uses. The large clusters contain a lot of long sequences, and these sequences

are merged by multi-input transactions between BestMixer addresses. Those large clusters possibly belong to the alpha pool, as it is highly likely that they contain deposits from customers. Because there was only one active cluster between January 2019 and March 2019 and the lack of payout transactions during this period, there seems to have been a change in mixing strategy around January 2019. This means that all deposits and payouts also belong to the same cluster, which means that the reconstruction cannot filter out all addresses that belong to the same cluster.

7.1.2. Sub-question 2

The results of sub-question 2 are described in Chapter 6. The study on the chosen transfer delays showed that 90% of the customers wanted to receive back their order within 24 hours, and 66% even within 2 hours. This means that most customers do not make use of the longer delay, whilst the longer delay makes the set of candidate payouts larger, which makes demixing more difficult. It is possible that clients want to receive back their money rather sooner than later to be able to spend it because they believe that the mixer will obfuscate the money no matter how long the delay is, which means that customers put a lot of trust in the service. However, it is also possible that due to the lack of trust, customers want to receive back their money as soon as possible, which means that both the lack and the presence of trust can be a motivation for short transfer delays.

The analysis on the amount of output addresses showed that 99% of the customers used at most four output addresses. The fact that most customers only chose one payout address can be the result of the added fee per output address. It seems that customers do not want to pay more for extra safety by using more output addresses.

Furthermore, a notable result that arose after analysing the orders is that the service fee paid by customers was not always between 0.5 and 3%. All orders that paid less van 0.5% fee could have gotten a discount from the administrators. Customers who paid more than 3% could have had to pay a fixed price if they used multiple payouts addresses, which was not incorporated in the calculation of the service fee. If the mixed amount was very low, this fixed price would increase the percentage more than when the mixed amount was higher. More analysis on this standard added fee per output address is necessary, as announcements from the administrators imply that this fee differs per pool.

Another explanation for the differing fees could be that the algorithm of the administrators was not flawless. Balthasar et al. (2017) noticed irregularities when investigating mixing service DarkLaunder. The researchers sent 69 orders to the mixer and discovered that for one of these order they paid more fee than agreed upon, and once they received the payout twice. They dedicated this to a bug in the algorithm: this could also be the case with BestMixer, as some of the payout addresses never received the payouts whilst the administrators declared that the order had status 'Complete'.

The results of the reconstruction show that there is a significant difference in the amount of candidate payouts per test case. The popularity of the mixer can have had an effect on this, as more transactions can lead to more candidate payouts. Another possible explanation can be the value of the deposit; all transactions that are equal to or less than 99.5% of the deposit are candidate payouts, which means that deposits of 1 BTC will likely get more candidate payouts than a deposit of 0.01 BTC. Customers that mix larger values during a period where the mixer was very popular will therefore be more challenging to demix without choosing settings like the gamma pool that cost extra money.

The overall result of the reconstruction shows that the reconstruction was able to find the correct payout for approximately 91% of the orders, but the amount of possible payouts are often very large. This means that when the reconstruction would, for example, return more than a million possible payout combinations, there is a 91% chance that the correct payout is included in this large set. Based on the current filtering and combining setup, it is not feasible to track all those possible payouts. With the current settings, most results are found within an adequate time frame. The filtering parameters can be tuned to find fewer possible payouts, but this will also decrease the accuracy of the reconstruction.It seems that the popularity and the wide range of settings that customers could choose lead to obstructions in recovering the relation between deposits and payouts since there are too many possible combinations. Whilst at first it was expected that the time efficiency of filtering and combining would make the technique inefficient, the returned amount of possible payouts combinations make the tool even more inefficient.

After consulting experts from the Financial Advanced Cybercrime Team it became evident that giving in on accuracy is not desirable because it is most important that the correct payout is found. Therefore, the reconstruction tool will likely not be helpful for demixing BestMixer in practice. However, the tool could be further improved, which will be further described in section 7.5, and the tool could be useful for less popular

centralized mixers.

7.2. Scientific contribution

The literature study showed that there is limited knowledge on how centralized mixing services operate. This research contributes to this knowledge gap by providing a better understanding of how a centralized mixer distributes payouts by analyzing the entire network of addresses and transactions that are part of the mixer. Three main contributions can be extracted from this thesis:

- **Mixing Process Characteristics:** New characteristics of mixing were discovered. For instance, there are strong indications that addresses are not reused, most addresses are active less than 24 hours, and most addresses have interactions with at least one other BestMixer address. The exploratory analysis also resulted in the discovery of a new sequence pattern, which is a combination of a peeling chain and internal transactions. These new insights can help detect and map copycats or other mixing services that use a similar mixing strategy as BestMixer.

- **Economics of a Mixer:** This research is the first to focus on the complete scope of a mixing service. Thereby, unique information on the economics of a mixer was presented. The balance of the mixer showed how much reserves the mixing service had throughout time. The analysis on the amount of activity of the mixer showed an increase in popularity of the mixer and that the mixer was used extensively.

- **Reconstruction Tool:** An attempt to reconstruct BestMixer deposits to payouts based on ground-truth data has been made. Even though the reconstruction performs poorly, it shows that filtering is not a feasible solution in popular mixing services that offer mixing with flexible variables. It is therefore advised to first map the activity of a mixing service and then decide whether filtering can be a sufficient solution.

7.3. Limitations

This section reflects upon the limitations of this study. Two of these limitations follow from the data source; the other limitations arise from choices made to scope this research. The first limitation relates to the available data on orders. All these orders were placed using the Clear Web version of the website, whilst there was also the possibility to place an order by accessing the service via TOR. Customers who placed their orders via TOR were possibly more careful than customers that accessed the service on the Web, as it is harder to trace traffic on TOR than Clear Web. This caution could indicate that TOR customers also show different behaviour in placing orders, as the administrators claim that certain settings will make it even more difficult to demix an order. Therefore the results of the reconstruction test cannot be generalised to all orders, as it is unknown how many of the orders outside the Order data were placed via TOR and Clear Web.

Another limitation is that the data consists of orders that were placed during four non-consecutive periods in time, which means that there are periods in which there is no information on orders available. Thus, the results on the performance metrics only hold for those periods. As the amount of sent and received transactions differ a lot over time, it could be possible that the reconstruction tool performs a lot better on orders that were placed during the first months of that the service was provided than during the last months that the mixer was active.

The focus of this research was on the mixing process with Bitcoin, whilst the mixer also offered its service for Litecoin and Bitcoin Cash. The order data mostly contains requests to mix Bitcoin, which could indicate that the tumbler received fewer orders with other currencies. This is supported by a post on the bitcoin talk forum, which is displayed in Figure 7.1. It shows that the BestMixer team announced that there was a low demand for Litecoin and Bitcoin Cash, which could indicate less activity from these addresses. If there are fewer transactions made by BestMixer with the other currencies, it could be easier to match deposits and payouts as there are less candidate payouts. However, as Bitcoin was more popular than the other currencies, more data was available, which was helpful in finding patterns in the mixing process.

Lastly, during the exploration of orders and testing the reconstruction on the data, the data was filtered to only exist of orders with status Complete. Whilst this status assured that BestMixer was finished with the mixing process, it does cut down the amount of information on orders that were placed. This extra data could have been used for the exploration, and the orders that were completed but had a different status could have been used for testing.


Re: [ANN] BESTMIXER.IO THE FUTURE OF BITCOIN MIXING! TECHNOLOGY IS HERE #173
January 14, 2019, 06:21:32 PM

Dear Friends,

we have to increase a Service fee of LTC and BCH mixing in Alpha, Beta and Gamma pools due to the low demand for these cryptocurrencies among our customers.
As soon as the interest and volume of mixing of these cryptocurrencies increase, we will return to the previous rates.

Best regards,
BestMixer Team

BESTMIXER.IO - BITCOIN MIXING & BITCOIN TUMBLING SERVICE

TOR: http://bestmixer7o57mba.onion/ | Clearnet: https://bestmixer.io/ | Bitcointalk thread: https://bitcointalk.org/index.php?topic=3140140.0

Figure 7.1: Post on Bitcointalk forum thread by administrators.

7.4. Recommendations for Law Enforcement

Successful attempts to demix mixing services have a social impact because they can scare criminals into not developing and using mixers. The new insights on centralized mixers that are presented in this thesis can be helpful for Law Enforcement in their demixing attempts. Therefore, this section describes how the results of this thesis can be applied to tackle crime.

The newly discovered pattern can be used to recognize and cluster other centralized mixing services that use a similar pattern. Most heuristics assume that there is a consistent pattern, for instance, the peeling chain, but this research showed that a sequence of transactions could consist of different types of transactions. This should be taken into consideration when analyzing new centralized mixers.

When trying to demix a centralized mixer, it is advised first to map the activity of the mixer. This starts with identifying addresses that are part of the mixing service. Then, the transactions to and from these addresses can be used to check for known patterns, the number of transactions made over time, et cetera. The mapping of the mixer can be used to decide what demixing strategy is suitable for this specific mixer.

Another piece of advice for Law Enforcement is to explore demixing BestMixer for the other currencies further. The data on orders contained significantly fewer orders with the currencies Litecoin and Bitcoin Cash than Bitcoin. This means that the reconstruction with filtering could work better on orders made with these currencies and is therefore worthwhile investigating.

The analysis of the balance of the mixer shows that there are peaks in the value of deposits placed on certain days. These peaks can correspond to known hacks, ransomware attacks or other cybercrimes. This could be verified by checking if the criminals behind those attacks used BestMixer to launder the money.

7.5. Future research

Future research is proposed based on the discussion of results and the limitations. Most of these suggestions focus on BestMixer specifically, but the outcomes can help in improving demixing in general. Suggestions for both the attribution phase and reconstruction phase are made.

The first suggestion is to look into the three different pools that customers could choose from. For a certain amount of deposit and payout addresses, it is known what pools were chosen by the customers. It could be checked if cash flows that belong to orders within the gamma pool differ from cash flows that can be linked to the other pools. The administrators claim that customers who chose the gamma pool would receive coins from private system assets and investors' coins. This claim can be confirmed by checking whether the cash flows of gamma orders contain deposits from other customers. Note that the administrators announced two increases of the service fees for the Beta and Gamma pool on the Bitcointalk thread, where the first increase was on 5 September 2018, and the second on 3 December 2018. If there indeed is a difference in cash flow for these other pools, it could help separate the transaction history for the different pools. If it is known what pool a customer chose, the separated transaction histories will likely return less possible payouts than the complete transaction history. This will improve the reconstruction.

In the reconstruction of this research, a deposit was used as input and a list of possible payout combinations was returned. The reconstruction tool can be reversed-engineered, which means that a payout can be entered, and the possible deposits are returned. When all the possible deposits that are associated with a payout have an illicit origin, it can be used as evidence that the person that received the payout deposited illicit coins. Also, all information on orders that is available can be used to filter out payouts that most definitely cannot belong to other orders. This data does not solely have to be from the Order data, but open source intelligence (OSINT) could be used as well. OSINT is data collected from publicly available sources,

for instance the Internet and academic papers. By using the Order data and OSINT as an extra filtering step, they cannot be used for testing anymore, as it is evident that the reconstruction tool will return the correct payout.

Furthermore, the data sources contain much more information than was used in this research. For example, Chainalysis provides information on the category and system name of some counter addresses that were sent to or received from BestMixer addresses. This information can be used to expand the knowledge on what type of services the mixer interacts with, which could help understand what type of clients use the mixer. It could also help in discovering whether the mixer used an external mixing step in the procedure. Other information that can be used to get a better image of the customers can be found in the Order data, where the chosen language and IP address can indicate which countries are most likely that the customers come from. This information can be combined with the timestamp of their deposits to see during what period of the day most customers are active. This will not improve the attribution or reconstruction of mixers, but it will help in better understanding what type of customers make use of mixing services.

An improvement to the attribution of centralized mixers would be to look further into the transaction fees used by BestMixer addresses. The exploratory analyses on the fees showed that for transactions of types *Payout* and *Payout, Internal*, there is a moderate to strong correlation between the size of the transaction and the fee value. However, this correlation is not perfect. Perhaps the strategy could have changed throughout time. Therefore, it could be interesting to see how the fee changes in time. If there is a clear pattern in transaction fee, it could result in a new heuristic that improves the attribution of a mixer.

Machine Learning can be applied in the application and reconstruction phase. In the attribution phase, it can be used to replace heuristics or in combination with heuristics to attribute addresses. As mentioned before, the results showed a high correlation between the amount of sent transactions and the amount of payouts. The ground-truth data can be used to build a model that predicts which output addresses of a transaction are non-BestMixer addresses and which are BestMixer addresses. For instance, a Machine Learning approach can be used that uses the BestMixer data for training and testing. If the prediction model works well on BestMixer, it can be used to cluster other centralized mixing services that use a similar pattern. This approach can also be adapted to other features of BestMixer's characteristics with high correlations. Features that can be derived from the blockchain are *amount sent, amount received, unique amount sent* and the *unique amount received*, which can be used to predict the amount of BestMixer addresses is sent to and received from (moderate correlation), and amount non-BestMixer addresses it received from (high correlation).

Another application of Machine Learning can be researched for the reconstruction phase. Even though Machine Learning cannot precisely predict the payout address, it can perhaps help predict characteristics of the most likely payouts, based on characteristics of the deposit. For instance, if a deposit is placed with a particular value at a specific time of the day, it could be more likely that the customer used one payout address than more payout addresses. The likelihood of the payouts having specific characteristics can help prioritise the possible payout combinations after filtering. Possible payout combinations with a higher priority can then be investigated first, but it will not improve the accuracy of finding the correct payout.

8

Conclusion

The existence of mixing services obstructs Law Enforcement in prosecuting criminals that received income from illegal activities. There was hope that cryptocurrencies would help prevent fraud, but the lack of control made it more appealing for criminals to acquire and launder digital coins. Academic approaches can support unravelling the black boxes of tumbling services. This chapter elaborates on the conclusions of this research. First, the two sub-questions are answered. Then, the conclusion to this research is presented.

Sub-question 1: What are characteristics of BestMixer's addresses and transactions? The analyses of the available data resulted in the discovery of multiple characteristics. It became clear that most addresses received and sent one transaction within 24 hours and had interaction with at least one other BestMixer address. BestMixer uses four types of transactions: *Deposit*, *Payout, Internal* and *Payout, Internal*. The growing amount of transactions shows that there was an increase in popularity of the mixer after July 2018 and that the mixer has a fluctuating amount of reserves in the mixer. The research has also shown that BestMixer makes use of a peeling chain, which is a chain of transactions of BestMixer addresses where each transaction has two receiving addresses, of which one is also a BestMixer address. The peeling chain is irregularly interrupted by a multi-input transaction between multiple BestMixer addresses that were sent to one other BestMixer address. By clustering addresses that were involved in the same transaction, this study also identified that a lot of BestMixer addresses have interactions with each other. Between January 2019 and March 2019, all addresses could be clustered into one cluster, which means that it is not possible to filter out addresses that belong to the same cluster as the deposit address since the payout address can also be part of this cluster. The results also show that it can be possible to predict the end date of a cluster if the begin date is known.

Sub-question 2: To what extent does the demixing method filtering succeed in reconstructing payouts of BestMixer from the deposits?
This research shows an attempt to reconstruct payouts from the deposits by using filtering techniques. From the test cases can be concluded that the reconstruction generally does not work well because it returns a lot of possible combinations of payout transactions, and it does not always contain the correct one. The reconstruction also performed poorly on the test cases with low deposits values, limiting the amount of possible payouts. The tests on transactions of type *Deposit* that were transacted in the first few months during which the mixer was functional show that the reconstruction could potentially work for deposits that were placed in less popular periods of the mixing service. Therefore it can be concluded that correctly reconstructing the payouts of BestMixer to deposits depends on the amount of activity of the mixer during the 72 hours following the deposit.

The main research question in this work was as follows:

To what extent can ground-truth data help in developing a demixing method for centralized mixing services?

As mentioned before, a demixing tool is defined to work sufficiently when the correct payouts to deposits can be singled out. Ground-truth data does help in understanding the mixing process better and finding

new heuristics for attribution, but it has not resulted in demixing a centralized mixer yet as the reconstruction did not work well on BestMixer. This research showed that the correct combinations of deposits and payouts could be reconstructed, but in most cases, multiple possible payouts are found instead of a single solution. Thus, the reconstruction presented in this research does not suffice in completely demixing a centralized mixer. However, the reconstruction and insights in the mixing process did help in establishing that the popularity of the mixer in combination with the adjustable mixing settings makes it impossible to demix all placed orders with 100% certainty. Therefore, it can be concluded that ground-truth data helped in discovering heuristics for the attribution of centralized mixers, but that it is not possible to reconstruct all orders of very popular mixers without addition information and improvements to the model.

Bibliography

Androulaki, E., Karame, G. O., Roeschlin, M., Scherer, T., & Capkun, S. (2013). Evaluating user privacy in Bitcoin. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 7859 LNCS*, 34–51. https://doi.org/10.1007/978-3-642-39884-1_4

Anton, B. M. Š., Veselý, I. V. I. M. Í. R., Ph, D., & Práce, V. Í. (2019). HEURISTIKY PRODEANONYMIZACI V SÍTÍCH.

Bitcoin.org. (2021). Bitcoin API getrawtransaction. Retrieved October 15, 2021, from https://developer.bitcoin.org/reference/rpc/getrawtransaction.html

Chainalysis. (n.d.). Chainalysis Professional Services. Retrieved September 2, 2021, from https://www.chainalysis.com/professional-services/

Chainalysis. (2021). The 2021 Crypto Crime Report. (February).

Crawford, J., & Guan, Y. (2020). Knowing your bitcoin customer: Money laundering in the bitcoin economy. *Proceedings - 2020 13th Systematic Approaches to Digital Forensic Engineering, SADFE 2020*, 38–45. https://doi.org/10.1109/SADFE51007.2020.00013

de Balthasar, T., & Hernandez-Castro, J. (2017). An analysis of Bitcoin laundry services. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10674 LNCS*(November), 297–312. https://doi.org/10.1007/978-3-319-70290-2_18

Ermilov, D., Panov, M., & Yanovich, Y. (2017). Automatic bitcoin address clustering. *Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017*, *2017-Decem*, 461–466. https://doi.org/10.1109/ICMLA.2017.0-118

Europol. (2019). MULTI-MILLION EURO CRYPTOCURRENCY LAUNDERING SERVICE BESTMIXER.IO TAKEN DOWN. https://www.europol.europa.eu/newsroom/news/multi-million-euro-cryptocurrency-laundering-service-bestmixerio-taken-down

Financial Intelligence Unit. (2017). Money laundering typologies. https://www.fiu-nederland.nl/en/general-legislation/money-laundering-typologies

Ghoshal, A. (2015). Bitcoin exchange bter will pay back users after losing $1.75m. https://thenextweb.com/news/chinese-bitcoin-exchange-bter-will-pay-back-users-after-losing-1-75-million-in-cyberattack

Gold, S., White, E., Roeder, W., McAleenan, M., Kabban, C. S., & Ahner, D. (2020). Probabilistic contingency tables: An improvement to verify probability forecasts. *Weather and Forecasting*, *35*(2), 609–621. https://doi.org/10.1175/WAF-D-19-0116.1

Hong, Y., & Lee, J. (2018). A Practical De-mixing Algorithm for Bitcoin Mixing Services, 15–20.

Liang, J., Li, L., Luan, S., Gan, L., & Zeng, D. (2019). Bitcoin exchange addresses identification and its application in online drug trading regulation. *Proceedings of the 23rd Pacific Asia Conference on Information Systems: Secure ICT Platform for the 4th Industrial Revolution, PACIS 2019*, (May).

Maxwell, G. (2013). CoinJoin: Bitcoin privacy for the real world. https://bitcointalk.org/index.php?topic=279249.0

Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G. M., & Savage, S. (2016). A fistful of Bitcoins: Characterizing payments among men with no names. *Communications of the ACM*, *59*(4), 86–93. https://doi.org/10.1145/2896384

Möser, M., Böhme, R., & Breuker, D. (2013). An inquiry into money laundering tools in the Bitcoin ecosystem. *eCrime Researchers Summit, eCrime*. https://doi.org/10.1109/eCRS.2013.6805780

Möser, M., Böhme, R., & Breuker, D. (2014). Towards risk scoring of bitcoin transactions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8438*, 16–32.

Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System.

Nerurkar, P., Bhirud, S., Patel, D., Ludinard, R., Busnel, Y., & Kumari, S. (2021). Supervised learning model for identifying illegal activities in Bitcoin. *Applied Intelligence*, *51*(6), 3824–3843. https://doi.org/10.1007/s10489-020-02048-w

Partz, H. (2019). Binance hackers bombard chipmixer to launder at least 4,836 btc. https://cointelegraph.com/news/binance-hackers-bombard-chipmixer-to-launder-at-least-4-836-btc

Pham, T., & Lee, S. (2016). Anomaly Detection in Bitcoin Network Using Unsupervised Learning Methods. http://arxiv.org/abs/1611.03941

Sun, X., Yang, T., & Hu, B. (2021). LSTM-TC: Bitcoin coin mixing detection method with a high recall. *Applied Intelligence*. https://doi.org/10.1007/s10489-021-02453-9

Tironsakkul, T., Maarek, M., Eross, A., & Just, M. (2020). Tracking Mixed Bitcoins. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12484 LNCS*, 447–457. https://doi.org/10.1007/978-3-030-66172-4_29

van Wegberg, R., Oerlemans, J. J., & van Deventer, O. (2018). Bitcoin money laundering: mixed results?: An explorative study on money laundering of cybercrime proceeds using bitcoin. *Journal of Financial Crime*, *25*(2), 419–435. https://doi.org/10.1108/JFC-11-2016-0067

Zhang, Y., Wang, J., & Luo, J. (2020). Heuristic-Based Address Clustering in Bitcoin. *IEEE Access*, *8*, 210582–210591. https://doi.org/10.1109/ACCESS.2020.3039570

Ziegeldorf, J. H., Matzutt, R., Henze, M., Grossmann, F., & Wehrle, K. (2018). Secure and anonymous decentralized Bitcoin mixing. *Future Generation Computer Systems*, *80*, 448–466. https://doi.org/10.1016/j.future.2016.05.018