TU WIEN Informatics

Analysis of Decentralized Mixing Services in the Greater Bitcoin Ecosystem

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering und Internet Computing

eingereicht von

Johann Stockinger
Matrikelnummer 01255547

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Matteo Maffei
Mitwirkung: Dr. Bernhard Haslhofer

Wien, 5. August 2021

Johann Stockinger	Matteo Maffei

TU WIEN Informatics

Analysis of Decentralized Mixing Services in the Greater Bitcoin Ecosystem

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering and Internet Computing

by

Johann Stockinger
Registration Number 01255547

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Matteo Maffei
Assistance: Dr. Bernhard Haslhofer

Vienna, 5th August, 2021

Johann Stockinger	Matteo Maffei

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Johann Stockinger

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 5. August 2021

Johann Stockinger

Kurzfassung

Durch die steigende Popularität von Bitcoin steigen auch die Anforderungen von Benutzern nach effektiven Techniken zur Wahrung ihrer Privatsphäre. Zwei häufig empfohlene Dienste sind Wasabi Wallet und Samourai Wallet, welche durch CoinJoins das robuste und sichere Vermischen von Bitcoins versprechen. Diese Arbeit untersucht die Rolle dieser beiden Dienste im Bitcoin Ökosystem, wie sich diese Rolle im Laufe der Zeit entwickelt hat und ob es möglich ist, Benutzer dieser Dienste zu deanonymisieren.

Um die Rolle beider Dienste zu analysieren werden Heuristiken entwickelt, welche die CoinJoin Transaktionen von Wasabi Wallet und Samourai Wallet erkennen. Auf Basis der so entdeckten Transaktionen wird sichtbar, dass sowohl die Anzahl der Transaktionen als auch die Anzahl der vermischten Bitcoins stetig wächst, was auf eine wachsende Nutzerbasis hinweist. Darüber hinaus wurden Adressen von Entitäten, welche in Verbindung zu kriminellen Aktivitäten wie Ransomware und Service-Hacks stehen, in der Nähe von CoinJoin Transaktionen beider Dienste identifiziert.

Schlussendlich wird das zugrundeliegende System beider Dienste in Hinblick auf Diebstahl, Denial-of-Service sowie Deanonymisierung von Benutzern analysiert. Es wird gezeigt, dass ein bösartiger CoinJoin-Koordinator theoretisch Nutzer deanonymisieren kann, dies jedoch nur in einer eher offensichtlichen Art, welche zu nachträglichem Misstrauen führen kann. Des Weiteren sind sowohl Wasabi Wallet als auch Samourai Wallet robust gegenüber Diebstahl und verfügen über Maßnahmen gegen Denial-of-Service Angriffe.

Abstract

With the rising popularity of Bitcoin, the desire for effective privacy preserving techniques rises as well. Wasabi Wallet and Samourai Wallet are two often recommended wallet services based on decentralized CoinJoins which promise robust and secure mixing of bitcoins. This thesis investigates the role of both wallet services in the greater Bitcoin ecosystem, how it has evolved over time, and whether it is possible to de-anonymize participants.

In order to analyze the role of both wallet services, heuristics are developed which detect CoinJoin transactions by both services. The discovered transactions are subsequently analyzed, showing that the number of transactions and the amount of mixed coins is steadily increasing, indicating a growing user base. Furthermore, addresses of entities which are connected to various criminal activities, such as service hacks and ransomware, have been observed within two hops of CoinJoin transactions conducted by both Wasabi Wallet and Samourai Wallet.

Finally, the underlying framework used by both wallet services is analyzed in regards to the dangers of coin theft, denial-of-service, and de-anonymization. We show that while an adversarial coordinator could potentially de-anonymize users, such actions would likely lead to retroactive suspicions as they would need to be conducted in an overt fashion. Furthermore, both wallet services are robust against coin theft from any party, and feature measures against denial-of-service attacks.

Contents

CHAPTER 1

Introduction

Cryptocurrencies such as Bitcoin have greatly shaped the concept of digital currencies, promising a decentralized payment scheme independent from central institutions such as banks. Since its initial publication by Satoshi Nakamoto in 2008 [Nak08] and its first release in 2009, Bitcoin has become increasingly popular. In fact, on February 21st, 2021, the market capitalization (the current price multiplied by the currently circulating supply) of Bitcoin exceeded one trillion US dollars[1].

Bitcoin features several advantages over traditional fiat[2] currencies. For example, Bitcoin is essentially immune to inflation caused by an excessive production of the currency as it has a fixed maximum supply which can be minted. Once this limit has been reached, no party is able to introduce additional units of the currency and its value can only be influenced by supply and demand. Furthermore, it is resistant to transactions with counterfeit money due to cryptographic proofs, and transactions are generally considered to be immutable, without the possibility to *undo* a transaction. Bitcoin transactions are also typically faster than traditional transactions and, in theory, feature lower fees. Moreover, the currency cannot be seized by governments or institutions, is robust against censorship and offers transparent transactions due to the usage of a public ledger [RKB15].

Another important aspect of Bitcoin is that anyone can create a Bitcoin address without any kind of formal verification as is typically required when, for instance, creating a bank account. While this pseudo-anonymity grants users a form of privacy when conducting Bitcoin transactions, research has shown that Bitcoin suffers from various weaknesses in regards to the privacy and anonymity. In order to combat these issues, a technique called *coin mixing* was introduced, which essentially shuffles the coins of participants in order to improve privacy.

[1] According to `https://coinmarketcap.com/currencies/bitcoin`.

[2] Fiat currencies are typically government issue currencies which are not backed by physical commodities, e.g., the Euro or the US dollar.

Coin mixing can be broadly categorized into two classes: centralized mixing services, and decentralized mixing protocols. While the role of centralized mixing services, where a trusted third party performs this shuffling of coins, has been researched and understood by the community, the role of decentralized mixing services has not yet been analyzed in detail. This thesis aims to explore two popular Bitcoin wallets, Wasabi Wallet and Samourai Wallet, which support decentralized mixing in order to gain a better understanding of the role of decentralized mixing services in the greater Bitcoin ecosystem. This is of particular importance, as coins mixed by these services become increasingly difficult to link to an individual entity and can therefore be used for nefarious purposes.

Furthermore, the question arises whether these services are truly able to obscure the link between individual coins and entities or whether this mixing of coins can be reversed by an adversary.

1.1 Research Questions & Methodological Approach

The goal of this thesis is therefore to provide an insight into the role and place of decentralized mixing schemes in the larger Bitcoin ecosystem by providing answers to the following research questions:

1. How do the decentralized mixing services provided by Wasabi Wallet and Samourai Wallet work and how can mixing transactions issued by these wallets be detected?

2. What is the role of Wasabi Wallet and Samourai Wallet in the greater Bitcoin ecosystem and how has it evolved over time?

3. Do these services suffer from security and/or privacy related vulnerabilities and is it possible to establish a link between input- and output addresses?

The methodological approach for research question (1) is to analyze both Wasabi Wallet and Samourai Wallet as well as publicly available mechanisms which aim at detecting mixing transactions issued by both wallets. Building on this existing work, new and improved heuristics are defined and used to find coin mixing transactions.

In order to answer research question (2), the transactions found by the developed heuristics are analyzed in detail. An exploratory data analysis is used to visualize the activity and mixed volume for both services over time. Incoming and outgoing addresses are clustered into entities which are subsequently mapped together with their neighboring entities in order to create a graph of the participants of both wallets.

Finally, research question (3) is answered by conceptually analyzing the underlying ZeroLink framework, which is the basis for both Wasabi Wallet and Samourai Wallet. The framework, as well as the relevant respective implementational differences are analyzed in regards to the attack vectors *de-anonymization*, *coin theft*, and *denial-of-service*.

1.2 Overview

This thesis is structured to fit the aforementioned research questions and is organized into four main chapters.

Chapter 2 provides an overview of the working mechanisms of Bitcoin and its underlying ledger. It also offers an overview on the concepts of anonymity and privacy in general, and how Bitcoin transactions can be de-anonymized before discussing the basics of privacy-preserving techniques, namely CoinJoins. It then explores the ZeroLink framework, which serves as a basis for both Wasabi Wallet and Samourai Wallet before discussing the fundamentals of both of these wallet services.

Chapter 3 explores how CoinJoin transactions conducted by Wasabi Wallet and Samourai Wallet can be detected. It discusses existing heuristics and provides novel improvements which aim to increase the heuristics' accuracy. The results of these improved heuristics and how they compare to the results of the unmodified heuristics are presented in detail.

Following the results of Chapter 3, Chapter 4 explores the role of both Wasabi Wallet and Samourai Wallet in the greater Bitcoin ecosystem by analyzing among other aspects the overall volume mixed by both wallets, the amount of mixing transactions conducted, the amount of mixed bitcoins leaving and the amount of fresh bitcoins entering each wallets' ecosystem. Moreover, this chapter analyzes which entities send coins into and receive coins from the decentralized mixing service offered by Wasabi Wallet and Samourai Wallet.

The security and privacy aspects of both wallets are discussed in Chapter 5, which analyzes whether the mixing process of both wallets can be reversed by an adversary, whether participants are at risk of losing their coins, how an attacker can prevent participants from mixing their coins (denial-of-service attacks), and what measures can be taken against such attacks.

Finally, Chapters 6 and 7 summarize the findings of the thesis, its limitations, and possible avenues for future work.

1.2.1 Reproducibility

The heuristics defined in Chapter 3 are implemented in *Python 3.7* and have been published as a Git repository. This repository includes technical documentation and requirements and can be found at the following URL: `http://thesis.stockinger.io`.

CHAPTER 2

Background & Related Work

This section explores and discusses the technical foundations of Bitcoin, its underlying ledger (the Blockchain), aspects of anonymity in Bitcoin, coin mixing services and techniques, as well as exploring the inner workings of Wasabi Wallet and Samourai Wallet.

Section 2.1 provides a basic overview of Bitcoin and its history with more technical aspects such as the consensus protocol being described in Section 2.2. Section 2.3 explores the concepts of anonymity and privacy in Bitcoin, while Section 2.4 and Section 2.5 provide an overview of de-anonymization and privacy-preserving techniques respectively.

2.1 Bitcoin

As described in Chapter 1, Bitcoin is a digital cryptocurrency which has many advantages over traditional fiat currencies. However, the concept of cryptocrrencies is older than Bitcoin, with David Chaum conceiving **eCash**, a scheme for untraceable payments based on blind signatures, in 1983 [Cha83]. Other publications followed, including some that explored creating distributed ecosystem, such as **b-money** by Wei Dai [Pec12] and using proof-of-work such as **bit gold** by Nick Szabo [Pec12].

Bitcoin finally solved issues which were present in previous proposals by building a robust, decentralized consensus and preventing double spending. As it was released open source, many developers quickly used Bitcoins innovations and created other cryptocurrencies. While the exact number of existing cryptocurrencies is hard to quantify, at least several thousand different cryptocurrencies have been conceived with the total market capitalization of cryptocurrencies exceeding 1.5 trillion US dollars as of March 3rd, 2021[1]. Today, Bitcoin remains the most dominant implementation of a cryptocurrency

[1] According to `https://coinmarketcap.com`.

Cryptocurrency	Market cap. (USD)	Share of total
Bitcoin	~925bn	~60.6%
Ethereum	~180bn	~11.8%
Cardano	~39bn	~2.5%
Binance Coin	~38bn	~2.5%
Tether	~36bn	~2.4%

Table 2.1: Top 5 cryptocurrencies by market capitalization as of March 3rd, 2021 based on data from `https://coinmarketcap.com`.

by far, with Ethereum, the second largest cryptocurrency by market capitalization trailing Bitcoin by a large margin. Table 2.1 shows the top 5 cryptocurrencies by market capitalization as of March 3rd, 2021.

Bitcoin is based on a CPU based proof-of-work consensus protocol. The smallest denomination of the currency is called a Satoshi, with 100 000 000 (one hundred million) Satoshis comprising one bitcoin[2] (BTC). Bitcoins are not available as actual coins, but rather as the unspent output of digitally signed transactions (unspent transaction outputs or UTXOs), which are then used as inputs for new transactions. A transaction may contain multiple input and output addresses (public keys) and has to be signed by the corresponding private keys of all input addresses. Due to these digital signatures, anyone can verify whether the input addresses legitimately belong to whoever signed them [Nak08].

Another interesting property of Bitcoin is that *coins* can only be divided through spending. This stems from the aforementioned fact that users do not have access to *coins* as they would in fiat currencies, but can only use UTXOs as input for new transactions. A direct consequence of this is the fact that if a user wants to transmit only part of the available BTC of an UTXO, they will have to specify at least two output addresses: one that receives the payment the user wants to conduct, and another that receives the *change* the user wants to keep. Figure 2.1 shows a simplified example of a Bitcoin transaction.

2.2 The Global State

In order to keep a robust record of all spendings conducted in Bitcoin, which is a prerequisite to prevent subsequent manipulations of records or to prevent users from using the same UTXO in multiple transactions (so-called *double-spending*), transactions are grouped together into a single block. The transactions within the blocks are stored as a Merkle tree[3], with each block also containing a number of additional fields and meta

[2] Bitcoin as a technology is typically capitalized while the actual currency is typically written in lower case [MPJ+13]

[3] A Merkle tree, also known as a hash tree, is a tree whose leaf nodes contain the cryptographic hash sums of data blocks and whose non-leaf nodes contain the concatenation of the cryptographic hash sums of their child nodes.

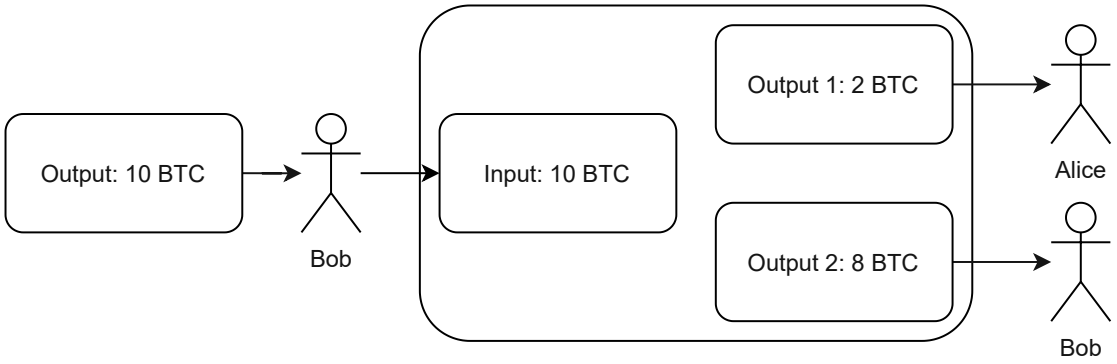

Figure 2.1: A simplified Bitcoin transaction in which Bob has access to 10 BTC and wants to transfer 2 BTC to Alice. As Bob can only spend the 10 BTC available as a single UTXO at once, he will have to specify a second output address that receives the remaining 8 BTC.

4 Magic	4 Size	4 Version	32 Hash of previous block	32 Hash of Merkle root
4 Time	4 Target	4 Nonce	1 - 9 TX counter	n Merkle tree of transactions

Figure 2.2: The structure of a Bitcoin block.

information such as the number of transactions held within the block (the transaction counter), the size of the block in bytes, the block version, and its timestamp. Furthermore, every block contains a cryptographic hash of the previous block, a cryptographic hash of the Merkle root, a nonce and the current block difficulty. The structure of a Bitcoin block is visualized in Figure 2.2 [NBF+16].

As every block contains the hash of its predecessor, modifying the contents of a block will cause cryptographic hash functions to return a different hash when using the modified block as an input, thereby creating a conflict with the hash stored in its successor. Therefore, modifying the contents of a single block will require the modification of all blocks following the modified block. This chain of blocks forms a tree structure, with every node having a single predecessor or parent (except for the root of the tree, i.e., the very first block, referred to as the Genesis block) but potentially with multiple successors or child nodes. However, only the longest chain (in regards to accumulated block difficulty) is considered valid [NBF+16].

In order for blocks to be accepted as part of the chain, a special hash value has to be computed over the hash of the previous block, the Merkle tree of all transactions contained in the block and the block nonce. The challenging part in this is the fact that the calculated hash must feature a certain number of leading zero bits determined by the block difficulty. In order to find such a hash, miners typically alter the value of the

nonce until the resulting hash aligns with the difficulty[4]. The difficulty value is adjusted by the protocol every 2016 blocks to such a level, that the previous 2016 blocks would have been generated in exactly two weeks (this leads to a rate of 6 blocks per hour or one block every 10 minutes). The first block received by participants to match these criteria is accepted and further blocks will be calculated with this block as their base. Naturally, other participants will not only check whether the hash value meets the criteria set by the current difficulty, but will also perform additional verifications, such as whether any transaction uses an input which has already been used in previous transactions (double spending) [NBF+16].

In order to incentivise this so-called *mining* of new blocks, the *miner* that finds a new block first will be awarded the *coinbase* transaction, the first transaction in a block which transfers the block reward to an address specified by the miner. The block reward started at 50 BTC and is halved every 210,000 blocks (roughly every four years) until it reaches 0, at which point no further bitcoins can be generated. This results in a total of $\frac{\sum_{i=0}^{32} 210000 * \lfloor \frac{50*10^8}{2^i} \rfloor}{10^8} \approx 2.1 * 10^7$ or 21,000,000 (twenty-one million) total bitcoins, which is expected to be reached around the year 2140 [NBF+16].

As the mining of new blocks will still be required even as block rewards approach and reach zero, mining is additionally incentivised by fees which are payed with every transaction. These fees, called miner fees, are the difference between the sum of all input values of a transaction and the sum of all output values, and is collected by whoever mines the block which includes the transaction similar to the coinbase transaction. As miners in general are naturally interested in collecting the highest amount of fees possible, they will gravitate towards including transactions which feature a higher fee [NBF+16].

Due to the decentralized nature of Bitcoin, it is possible for two or more different (but valid) blocks to be broadcast at a similar time, with participants receiving different blocks first. In this case, participants may start the proof-of-work process for new blocks with different blocks as base until a longer chain is received [Nak08].

This proof-of-work based mining mechanism has also attracted criticism, however, as it requires a large amount of electricity. Li et al. estimate a minimum annual power consumption of 23.38 TWh for Bitcoin mining alone in [LLP+19]. This enormous energy consumption leads to a significant carbon footprint as discussed by Truby in [Tru18].

2.3 Anonymity and Privacy in Bitcoin

Pfitzmann and Köhntopp define anonymity in [PK01] as "the state of being not identifiable within a set of subjects, the anonymity set". In terms of unlinkability, also defined in [PK01] as a third party not being able to increase their knowledge of the relation of two items within a system, this means that an item cannot be linked to an individual.

[4]Miners could also change a special parameter in the *coinbase* transaction, but this would require recalculating the Merkle tree and therefore consume more resources.

Pseudonymity, defined as "the use of pseudonyms as IDs", in contrast allows items to be linked to pseudonyms, but not to an individual as long as the pseudonym itself cannot be linked to the individual [PK01].

A basic example of anonymity would be a guestbook - a single person is able to write multiple entries which cannot be linked to each other, or to the individual. An example for pseudonymity could be a bulletin board or forum, where a person is able to create a pseudonym by specifying a user name and to contribute posts. While it is not possible to link these posts to the individual, they are linked to the specified username, i.e., the pseudonym. This is a crucial difference as, should the pseudonym become linkable to the individual, all actions the individual has taken under the guise of their pseudonym (i.e., all posts) become linkable to the individual as well.

In Bitcoin, all transactions are linked to the public/private key pair used to sign the transaction. While individuals may generate as many addresses (hashes of public keys) as they like without having to reveal any identification, all actions taken by these addresses can be linked. Bitcoin therefore does not provide users with anonymity, but rather with pseudonymity. This becomes an issue if it is possible to cluster multiple addresses (i.e., link multiple addresses together) and link them to an individual - the privacy granted by Bitcoins pseudonymity would then be broken and in the worst case all transactions ever conducted by the individual would become linkable [NBF+16]. How this clustering of addresses is achievable is discussed further in Section 2.4, with Section 2.5 providing insights into Bitcoin mixing, which aims to improve the anonymity of users by *unlinking* clustered coins.

Another important aspect to consider when dealing with technologies that provide anonymity are ethical considerations, in particular the right to privacy vs. the fact that anonymity can be exploited for nefarious purposes. This is true not only for cryptocurrencies, but also for e.g., end-to-end encryption or the Tor network.

2.4 De-anonymization Techniques

In classic payment schemes, the anonymity of the participants of transactions is based on the principle that only entities which are directly involved in transactions are aware of their contents. Obviously, this model of guaranteeing anonymity is in direct contrast to the public nature of the blockchain, where all participants are able to see and verify all transactions ever conducted. However, all Bitcoin transactions are conducted by public/private key pairs which can be generated on the fly. This creates a certain pseudonymity for participants similar to the pseudonymity of a user name in a public forum as described in Section 2.3. It is possible to see the transactions conducted by an entity identified by a public key, but there is no direct link to the corresponding real-world user of this key [Nak08].

As an additional security mechanism to preserve privacy, the original paper by Satoshi Nakamoto suggests to generate new key pairs for every transaction in order to prevent

the entire transaction history of a user being linked if a single key pair is compromised (i.e., a link between a key pair and a real-world entity could be established) [Nak08]. This linking of public addresses to real-world characteristics has been discussed in various publications, such as [GKRN18] by Goldfeder et al., which aims to de-anonymize users through the use of third-party trackers when items are purchased using cryptocurrencies, or [BKP14] by Biryukov et al., which explores how users can be de-anonymized via network traffic analysis.

2.4.1 Clustering Heuristics

Due to the previously stated mechanisms, many participants of the Bitcoin network as well as institutions and even law enforcement agencies such as the FBI consider Bitcoin transactions to be difficult to link to one another, or to real world users [MPJ+13]. However, Meiklejohn et al. discussed a way to cluster and tag Bitcoin addresses in [MPJ+13]. As a first step, the authors attempted to tag as many addresses as possible with publicly available information. This was achieved by:

- Participating in mining pools, tagging every input address of payout transactions
- Depositing and withdrawing coins using different wallet services, as well as from bank and non-bank exchanges
- Conducting purchases from vendors accepting Bitcoin as payment scheme
- Participating in gambling services
- Interacting with other services, such as mixing services, or advertisement services
- Searching for publicly advertised addresses e.g., from charities accepting Bitcoin donations

As a next step, Meiklejohn et al. defined two heuristics in order to cluster addresses belonging to the same entity. The first heuristic, which has been previously proposed by e.g., [RH13] and [AKR+13], exploits the fact that in order for a transaction to be valid, all inputs must sign the transaction with their respective private key. As the private key is typically not shared among users, the heuristic therefore concludes that **all public keys used as an input for a transaction belong to the same entity**. This heuristic can be applied transitively. As all input addresses of one transaction are clustered, the input addresses of all other transactions which use any of the already clustered addresses as an input belong to the same cluster. As an example, if addresses a_1 and a_2 are used as inputs for transaction t_1, and a_1 and a_3 are used as inputs for transaction t_2, then a_1, a_2, and a_3 belong to the same entity [MPJ+13].

The second heuristic proposed in [MPJ+13] deals with change addresses used to receive the spare change (the difference between the input value of a transaction and the output value which is to be transferred). More specifically, it exploits the idiom of use proposed

by Nakamoto in [Nak08] that fresh addresses (i.e., newly created addresses which receive the change and are never re-used) should receive any change of a transaction. In order to limit the amount of false positive matches, [MPJ+13] considers transaction outputs to be change addresses if the address was never the target of any transaction before, the transaction is not a coinbase transaction, the address is not also part of the transactions inputs, and it is disambiguous (i.e., there is only one output address which matches these criteria). If such an output is present in a transaction, the second heuristic assumes that **the entity in control of all inputs is also in control of this change address** [MPJ+13].

2.5 Privacy-preserving Techniques

As clustering poses a direct threat to the anonymity and privacy of the users of cryptocurrencies in general, several proposals and even entire new currencies have been published which aim to improve user privacy. One such privacy-centric currency is Zcash[5], which is based on the Zerocash proposal and makes use of zero knowledge proofs in order to create shielded transactions which hide all participants as well as the amount transmitted in a transaction[SCG+14]. Another prominent example is Monero[6], which uses Ring Confidential Transactions (which in turn are based on cryptographic ring signatures) in order to achieve similar goals [Noe15].

However, both zero knowledge proofs and ring signatures are incompatible with Bitcoin and therefore cannot be introduced into Bitcoin without a hard fork of the software itself. In order to improve user privacy of Bitcoin participants, various concepts of **coin mixing** have been discussed and implemented. Broadly speaking, coin mixing allows Bitcoin users to mix their UTXOs with the UTXOs of other users, thereby disrupting previously established links between UTXOs. Coin mixing can be separated into two distinct classes:

- Third party based **centralized** services, which facilitate the mixing of coins sent to them by users
- **Decentralized**, peer-to-peer based protocols, in which coins are mixed by users themselves without requiring a third party (other than for coordinating the mix)

Centralized or dedicated mixing services essentially work by accepting transactions from users and sending back the same amount (minus a fee) to a different address specified by the user. These services have to be trusted by users, both that they do not keep any records of which input is mapped to which output, and that users receive back their coins in the first place [NBF+16]. The function of centralized services, how they can improve their accountability and guarantee anonymity, as well as their place in the

[5] https://z.cash
[6] https://www.getmonero.org

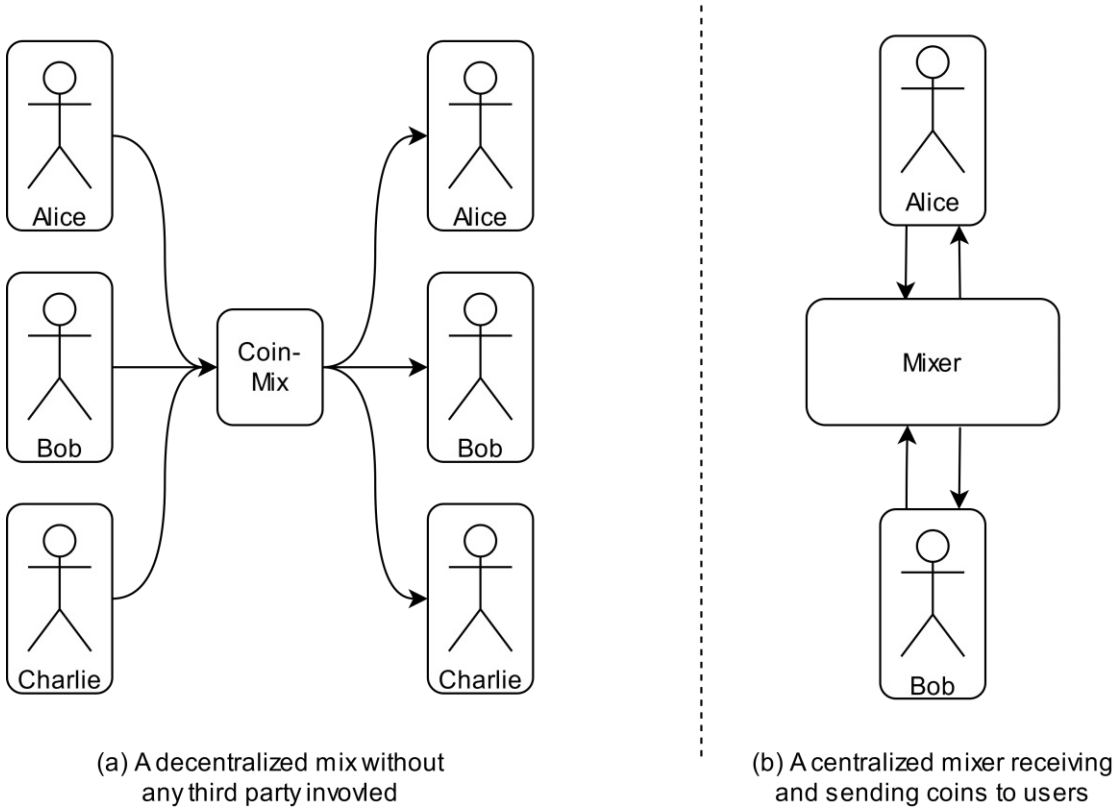

Figure 2.3: Decentralized vs. centralized coin mixing.

greater Bitcoin ecosystem, have all been discussed in detail in a number of publications, such as [BNM+14], [MBB13], [dBHC17], [VWOvD18], and [CG20].

In decentralized or peer-to-peer mixes on the other hand, trust in a third party is unnecessary as participants will exchange coins between each other. Figure 2.3 shows the basic premise of both centralized and decentralized coin mixing.

2.5.1 CoinJoins

The concept of CoinJoins was first introduced by Bitcoin core developer Gregory Maxwell and published as a post on the Bitcoin Forum [Max13]. The basic concept of a CoinJoin transaction is that instead of conducting separate transactions, multiple users could combine their inputs and outputs into a single transaction, thereby disrupting the assumption that all inputs of a transaction belong to the same entity [Max13]. A simplified CoinJoin transaction is illustrated in Figure 2.4.

CoinJoin transactions are possible because the signatures of the inputs of a transaction are independent of each other, allowing participants to specify an agreed upon set of input and output addresses, which can then be subsequently signed by all participants.

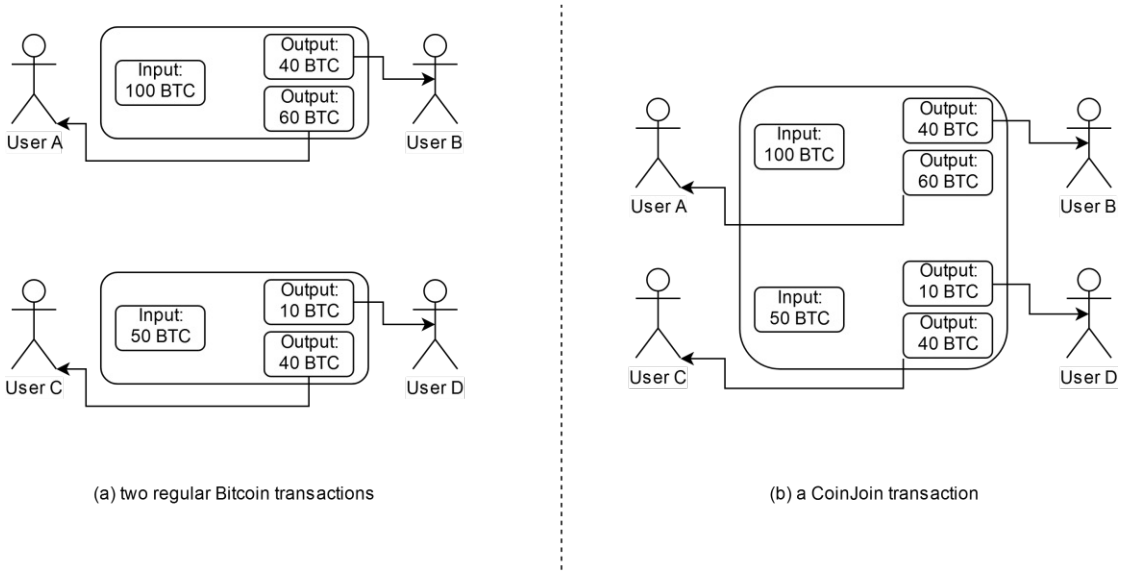

Figure 2.4: The basic idea of a CoinJoin based on [Max13]. (a) shows two distinct Bitcoin transactions while (b) shows a single CoinJoin transaction. To a third party, it becomes increasingly complex to link the various outputs to individual users as the number of participants in a CoinJoin transaction increases.

This also provides robustness against coin theft, as participants can simply refuse to sign the transaction if the input or output addresses of a transaction are not what was previously agreed on [Max13].

In order to truly improve the privacy of CoinJoin transactions, participants could further agree on a specific value of their outputs, provide at least $n * v$ BTC (where n is the number of participants and v is the agreed upon value) as inputs and specify n different output addresses, each receiving v BTC. This way, a third party would be unable to link any output address to any input address, providing the participants with an anonymity set of n [Max13]. As an example, consider 5 participants which aim to mix their coins using a CoinJoin transaction. The participants would first agree on a common value such as 1 BTC, and then provide inputs with a combined value of at least $5 * 1 = 5$ BTC (in practice, every user would likely provide one or more UTXOs with a (combined) value of at least 1 BTC as input for the CoinJoin transaction). Subsequently, every user specifies one output address, with each such address receiving 1 BTC. An onlooker would then be unable to link any of these 1 BTC UTXOs to any input.

While the basic CoinJoin concept is generally resistant to malicious participants attempting to steal other participants coins as stated above, it does suffer from some vulnerabilities against attackers which participate in the CoinJoin transaction. Depending on how the participants of a CoinJoin transaction are coordinated, it could be possible for an attacker to create a large number of addresses (so-called Sybil attacks) and queue

to participate in CoinJoin transactions. The attacker could then issue a denial-of-service attack by simply refusing to sign the final transaction, thereby disrupting the process for all other participants. The impact of this attack increases with the amount of times an attacker controlled address is selected to participate in the CoinJoin transaction [BOLL14], [QC19].

In a similar fashion, an attacker could perform a de-anonymization attack, again by creating a large number of addresses. The goal of the attacker in this scenario is to participate in a CoinJoin transaction with a large number of their own addresses, thereby reducing the anonymity set of all other participants. In an extreme example, consider 5 participants of a CoinJoin, 4 of which are controlled by an attacker. In such a scenario, it would be trivial for the attacker to de-anonymize the single remaining address [BOLL14], [QC19].

The CoinJoin proposal has subsequently inspired and serves as a basis for additional protocols such as CoinShuffle [RMSK14] and CoinShuffle++ [RMSK17].

2.6 Wasabi Wallet & Samourai Wallet

Wasabi Wallet and Samourai Wallet are both open source Bitcoin wallets that focus primarily on enhancing the privacy of their users. Alongside standard features expected from a Bitcoin wallet, both wallets offer a variety of privacy-preserving features, the most prominent of which are their respective CoinJoin implementations which are based on the ZeroLink framework.

2.6.1 The ZeroLink Framework & Chaumian CoinJoins

The ZeroLink framework, co-authored by Ádám Ficsór of Wasabi Wallet and *TDevD* of Samourai Wallet serves as the basis for the mixing services offered by both Wasabi Wallet and Samourai Wallet. ZeroLink defines a pre-mix wallet, which consists of UTXOs that have not yet been mixed, a post-mix wallet, consisting of UTXOs which have been mixed, and a mixing technique, which moves coins from a pre-mix wallet to a post-mix wallet. While the framework is compatible with most on-chain mixing protocols such as CoinShuffle, it also defines the Chaumian CoinJoin as a mixing technique [dFT17].

Chaumian CoinJoins are based on CoinJoins, discussed in Section 2.5.1, and blind signatures[7], first described by David Chaum in [Cha83]. Chaumian CoinJoins also make use of a tumbler, which acts as a coordinator of the CoinJoin transaction. A Chaumian CoinJoin transaction has the following phases [dFT17]:

1. Input Registration Phase, in which a user registers their confirmed UTXOs to be used as inputs of the CoinJoin transaction along with proofs that the coins are

[7] A blind signature is a digital signature where the content of what is signed is hidden from the signer but which is able to correctly verify the *unblinded* or plain text content.

indeed owned by the user, the desired change output addresses, as well as the blinded outputs which are to receive the mixed coins. Once all information has been received by the tumbler, it will verify the validity of the input transactions and proofs, sign the blinded output, and return this signed blinded output to the user. The user can then unblind the signed output.

2. Connection Confirmation Phase, which is triggered once enough inputs have been registered in order to achieve the desired anonymity set of the CoinJoin. This phase is intended to confirm whether enough participants are still available for the desired anonymity set to be upheld. All users must confirm their intent to participate in the CoinJoin in this phase and will be temporarily banned from participating in CoinJoins should they not proceed in subsequent phases. In the event that not enough users confirm their intent to participate, the tumbler will fall back to the input registration phase.

3. Output Registration Phase, in which the unblinded signed outputs are registered with the tumbler. This can be done by the same user that originally registered the input, but ideally using a different communication path (e.g., a new Tor connection[8]). The tumbler will then verify the signature on the unblinded output and construct the CoinJoin transaction.

4. Signing Phase, in which the unsigned CoinJoin transaction is transmitted to every user which has previously registered an input in phase (1). Once the tumbler has collected all signatures, it will combine them and create the finalized CoinJoin transaction which is then propagated on to the Bitcoin network at large.

Due to the use of blind signatures, the tumbler is unaware of the contents of the blinded outputs registered in phase (1) and is therefore unable to link any input address provided to the unblinded outputs registered in phase (2) [dFT17].

While potential security implications and threat models against the ZeroLink framework are further analyzed in Chapter 5, the ZeroLink framework does state that a common denomination must be used in the mixing technique in order to prevent an attacker from potentially de-anonymizing users based on amount analysis [Atl14] [dFT17]. Users may also participate in multiple rounds of CoinJoins in order to increase the anonymity set of their coins further (*remix*), or if the common denomination is smaller than the amount of coins the user wants to mix[9] [dFT17].

[8] Note that in order to prevent the tumbler from identifying the user based on network artifacts such as IP addresses, ZeroLink requires the use of Tor or similar mechanisms that obfuscate the users' identities.

[9] Although it is theoretically possible in the ZeroLink framework for a user to register multiple inputs in a single round, therefore foregoing the need to participate in multiple rounds, this results in a lower anonymity set [dFT17].

2.6.2 Wasabi Wallet

Wasabi Wallet is developed by *zkSNACKs* and published on GitHub [zkS18]. Based on ZeroLink, it provides an implementation of the Chaumian CoinJoin alongside other privacy-preserving techniques. The lowest possible denomination for Chaumian CoinJoins supported by Wasabi Wallet is 0.1 BTC with a small alteration (up to ±0.02 BTC) each round in order to improve the anonymity set for coins which are remixed[10]. While 0.1 BTC is the lowest possible amount to register for the input registration phase in Wasabi Wallet, larger denominations of the form $0.1 * 2^n$ where n is a positive integer, are possible as well [Was19].

Wasabi Wallet charges a coordinator fee in addition to the mining fees for every CoinJoin, regardless of whether the input is an unmixed *premix* input, or a *remix* input. The coordinator fees are calculated as $0.003\% * a$ of the mixed amount, where a is the desired anonymity set. For example, if the desired anonymity set is 50, then the fees would amount to $0.003\% * 50 = 0.15\%$. The exact amount may vary, for example *remix* input may not have to pay the full coordinator fee if not enough funds are available. Conversely, if the left over change amount of a transaction is too small to be paid out to participants (smaller than 0.3% of the ~0.1 BTC base denomination or ~0.0003 BTC), it may be added to the coordinator fee [Was19].

Another privacy-preserving feature offered by Wasabi Wallet is PayJoin, which is a coordinated transaction that aims at breaking the common input ownership heuristic by obscuring the exact amount transferred between users. This is achieved by creating a transaction in which both users transfer funds to each other. For example, if Alice wants to transfer 2 BTC to Bob, she might transfer 5 BTC to Bob instead, with Bob transferring 3 BTC to Alice in the same transaction [Was19].

Wasabi Wallet also differentiates itself from most other Bitcoin wallets through the mandatory use of labels for received UTXOs and spent coins (target addresses). These labels are used to track which external entity is aware of which coin and would subsequently be aware of potential further uses of this coin. This mandatory usage of labels assists in establishing which coins should be "CoinJoined" before further use [Was19].

In order to correctly verify transactions, Wasabi Wallet automatically attempts to detect whether a full Bitcoin node is installed on the system, and tries to connect to the Bitcoin background service[11] from which it retrieves necessary block information. Should a full node not be available to Wasabi Wallet, it can also fall back to querying information from the Bitcoin network itself using block filters based on BIP158 [Was19].

Wasabi Wallet routes all traffic through the Tor network in order to thwart network level attacks on a users' anonymity. When retrieving UTXOs without a full Bitcoin node installed, the Wasabi Wallet backend provides the aforementioned block filters to

[10] Note that the actual range is typically between 0.095 and 0.105 BTC [Was19].

[11] If such a service is not installed locally but on a remote machine, it is also possible to configure the IP address or Tor onion service.

all Wasabi Wallet clients over Tor, from which clients can identify and retrieve blocks of interest. Every block is retrieved using a new Tor stream to a new peer. If a full node is available, it will retrieve all Bitcoin blocks with Wasabi Wallet simply retrieving information from the installed node itself. New transactions are similarly broadcast only to nodes connected to the Tor network, with every transaction being broadcast to a new Bitcoin peer using a new Tor stream. This is true even if a full node is available [Was19].

2.6.3 Samourai Wallet

Like Wasabi Wallet, Samourai Wallet also offers an implementation of the Chaumian CoinJoin based on the ZeroLink framework called Samourai Whirlpool. The underlying framework and concepts are therefore identical for both Wasabi Wallet and Samourai Wallet. Both wallets make use of a central tumbler which coordinates the CoinJoin transactions between users while being unable to link input and output addresses due to blinded signatures.

In contrast to Wasabi Wallet, Samourai Whirlpool features three distinct denominations for its CoinJoins which are called *pools*: 0.01 BTC, 0.05 BTC, and 0.5 BTC. Support for an additional pool of size 0.001 BTC was published with version *v0.99.96e* which was released on 05.03.2021.

In practice, when users join a pool, Samourai Wallet generates a special transaction named *Tx0*. This transaction splits the input into n different outputs (called *premix*) whose size equals the chosen pool denomination (plus a small amount which will be used to pay for subsequent mining fees), one output which carries the fee paid for joining the pool ($5\% * s$ where s is the pool size), and one output which carries potential change. For example, if a user were to join the 0.01 BTC pool with an input of 0.123 BTC, the outputs of the *Tx0* transactions would be [Sam19]:

- 1 output of 0.0005 BTC, the fee to be paid ($0.05 * 0.01$)
- 12 outputs of 0.01 BTC, the *premix* outputs to be mixed (as mentioned before, the outputs would be slightly higher to compensate for the mining fees)
- 1 output of 0.0025 BTC, the remaining change

These *premix* outputs can now be registered by the user to be used in the actual CoinJoin transactions. The CoinJoin transactions themselves always feature exactly 5 inputs, of which at least 1 is a *remix* (i.e., a transaction which is being mixed for at least a second round) and 5 outputs. The very first mix of a pool, the *genesis mix*, does not feature a *remix* input for obvious reasons. The value of the inputs is slightly above the pool denomination for *premix* transactions (with the difference being used to pay for mining fees) while the value of the outputs is exactly equal to the pool denomination. The input value of *remix* transactions is obviously exactly equal to the pool denomination as they themselves are outputs of a CoinJoin transaction [Sam19].

The fact that at least one input of a transaction is a *remix* transaction leads to the observation that every Whirlpool CoinJoin has a link to its genesis mix. This further leads to the conclusion that the backward-looking anonymity set for every CoinJoin output reaches back to the genesis mix and increases for every future CoinJoin (although the backward-looking anonymity set of every transaction itself is fixed). In contrast, the forward-looking anonymity set for a Whirlpool output depends on whether it, or one of its 4 peer-outputs, participates in further Whirlpool CoinJoins as a *remix* input [Sam19].

Alongside Samourai Whirlpool, Samourai Wallet also features a number of additional privacy features, such as:

- Ricochets, which causes outgoing transactions to *bounce* over a number of destinations before reaching the final target.
- Stonewall, which is a transaction that mimicks a CoinJoin transaction in that there are multiple outputs with a common denomination, but with all inputs being supplied by the same user. One of these outputs will be the actual transfer a user conducts to a different entity, while the other outputs remain under control of the user, thereby increasing the entropy of the transaction.
- Stonewallx2, which is a Stonewall transaction with two participants.
- Stowaway, which is similar to the PayJoin feature of Wasabi Wallet.

Another key difference between Samourai Wallet and Wasabi Wallet is that Wasabi Wallet has clients for popular desktop operating systems, whereas Samourai Wallet is only available as an Android APK. By default, Samourai Wallet is a light client that connects to the Samourai Wallet backend infrastructure in order to query and broadcast transactions. In a similar fashion to Wasabi Wallet, Samourai Wallet uses Tor to obfuscate a user's IP address. As an alternative to Tor, it is also possible for users to specify their own OpenVPN configuration. The developers of Samourai Wallet have published a compatible full node wallet server called **Samourai Dojo** to allow users to run their own full node.

CHAPTER 3

Analysis of Decentralized Mixing Services

This chapter explores how Wasabi Wallet and Samourai Wallet CoinJoin transactions can be detected. Section 3.1 and Section 3.2 introduce heuristics which aim at detecting Wasabi Wallet CoinJoin and Samourai Whirlpool transactions respectively.

The part of the Bitcoin ledger analyzed for this thesis stretches from block 1 to block 658738.

3.1 Detecting Wasabi Wallet CoinJoin Transactions

Ádám Ficsór published a Github repository[1] in August 2019, which aims at detecting both Wasabi Wallet and Samourai Wallet transactions with the goal of comparing various statistics of both wallets. The detection mechanism for Wasabi Wallet transactions exploits the historically fixed coordinator addresses used by Wasabi Wallet until block 610000 and tags transactions as Wasabi Wallet transactions if both of the following conditions are satisfied [dF19]:

- At least one of the following coordinator addresses is in the list of output addresses of the transaction:
  - **bc1qa24tsgchvuxsaccp8vrnkfd85hrcpafg20kmjw**
  - **bc1qs604c7jv6amk4cxqlnvuxv26hv3e48cds4m0ew**
- There are at least 3 indistinguishable output values (i.e., there are at least 3 outputs with the same value).

[1] https://github.com/nopara73/WasabiVsSamourai

While this method is accurate in its detection of Wasabi Wallet transactions, Wasabi Wallet began creating new coordinator addresses to collect fees starting with January 31, 2020[2], or block 610000 [dF20].

Ground Truth In order to evaluate the effectiveness of subsequent heuristics which are not based on the static coordinator heuristic described above, we now establish a *ground truth*. To do so, we use the static coordinator address from [dF19] in order to detect all Wasabi Wallet CoinJoin transactions from block 530500 to block 609999. We can further extend our ground truth data due to the fact that all potential detections before block 530500, the first block to feature Wasabi Wallet CoinJoin transactions [dF20], must be false positive identifications. Doing so yields 7406 Wasabi Wallet CoinJoin transactions between blocks 1 and 609999.

Ficsór's Heuristic Ficsór published a new repository[3] in May 2020, which continues to analyze and compare CoinJoin transactions of Wasabi Wallet, Samourai Wallet, and other CoinJoin transactions. This repository includes a new heuristic to identify Wasabi Wallet CoinJoin transactions after block 610000 by evaluating if transactions meet the following conditions [dF20]:

- The transaction has at least 10 outputs of equal value
- The most frequent equal output value is 0.1 ± 0.02 BTC
- There are more inputs than the outputs of the most frequent equal value

Comparing this new heuristic from [dF20] with the established ground truth yields the following results:

- 2367 transactions were identified by [dF20], but are not part of the ground truth (i.e., ~24.4% of the identified transactions are false positives)
  - 2269 of these transactions occurred before, and 98 after block 530500
- 81 transactions were not identified by [dF20], but are part of the ground truth (i.e., ~1.1% of the transactions in the ground truth were not identified)
- 7325 transactions were identified by [dF20] and are indeed part of the ground truth (i.e., ~75.6% of the identified transactions are true positives)

[2] https://docs.wasabiwallet.io/FAQ/FAQ-UseWasabi.html#what-is-the-coordinator-address

[3] https://github.com/nopara73/Dumplings

Our Heuristic We now propose an improved heuristic for detecting Wasabi Wallet transactions, which uses the work of Ficsór as its base. A transaction is a Wasabi Wallet CoinJoin transaction if all of the following conditions are true:

1. There are at least 10 outputs of equal value
2. The most frequent output value is 0.1 ± 0.02
3. There are at least as many inputs as occurrences of the most frequent output
4. There is at least one unique output value
5. There are at least 3 distinct output values

The first three aspects of this heuristic (1) - (3) are as proposed by Ficsór in [dF20]. The reasoning behind (4) is that the fee collected by the coordinator address is very likely to be distinct from any other output produced by the transaction. The third distinct output value (5) is due to the fact that it is highly likely that at least one change output will be produced by the transaction (i.e., there are at least distinct values for the CoinJoin itself, the coordinator fee, and one change output).

Analyzing the results of our proposed heuristic against the established ground truth data revealed that:

- 110 transactions identified as Wasabi Wallet transactions were known to be addresses used by gambling services, namely *LuckyBit*[4] (addresses starting with *1Lucky*) and *SatoshiDice*[5] (addresses starting with *1dice*). As Wasabi Wallet generates the output addresses on the fly (and only uses *Bech32* addresses), addresses matching these schemes will not be valid.
- In 149 transactions at least one address appeared multiple times in the list of output addresses. Due to the generation of recipient addresses for CoinJoins in Wasabi Wallet, this should not occur[6].
- 1118 transactions featured output values of exactly 0.08, 0.09, 0.1, 0.11, or 0.12 BTC. Such precise values are unlikely to occur in actual CoinJoins due to the slight discrepancy the CoinJoin values have to the base denomination as explained in Section 2.6.1.
- 37 transactions featured output values between $0.08 - 0.085$ or $0.115 - 0.12$ BTC. Such edge cases are unlikely to occur as the actual denomination should be closer to 0.1 BTC in the vast majority of cases [Was19].

[4] `https://luckyb.it`
[5] `https://satoshidice.com`
[6] Note that while the ZeroLink protocol itself does not explicitly forbid this, it would likely be detrimental to a users' anonymity.

- There were 73 transactions identified as Wasabi Wallet transactions by the static coordinator address heuristic which featured output values outside the range of $0.08 - 0.12$, e.g., 0.05. We believe that these transactions are not real Wasabi Wallet CoinJoins conducted in a productive environment, but rather tests in order to e.g., determine how changing the denomination of CoinJoins impact the fees paid for by users.

In order to increase the accuracy of our heuristic, the following additional filters were introduced upon inspection of the false positive results. Transactions are to be discarded if they are detected by our heuristic and:

- Address schemes used by known gambling services appear in its output
- An address appears multiple times in the list of output addresses
- CoinJoin output values are exactly 0.08, 0.09, 0.1, 0.11, or 0.12 BTC
- CoinJoin output values are between $0.08 - 0.085$ or $0.115 - 0.12$ BTC

Furthermore, discard transactions found by the static coordinator address heuristic if CoinJoin output values are not between 0.08 and 0.12 BTC.

Evaluating the refined heuristic against the ground truth established from blocks 1 to 609999 (the final block before Wasabi introduced fresh coordinator addresses [dF20]) results in:

- 153 transactions detected as false positive (transactions were identified Wasabi Wallet CoinJoin transactions, even though they did not feature a coordinator address)
- 8 transactions detected as false negative (transactions were identified with the static coordinator address heuristic, but not with our proposed heuristic)
- 7325 transactions detected as true positive (transactions identified by both heuristics)

Table 3.1 compares the results of both heuristics against the established ground truth data (i.e., from block 1 to block 609999). Using this heuristic and discarding the 153 transactions known to be false positives, we have identified 18,840 Wasabi Wallet CoinJoin transactions from block 1 to block 658738.

Metric	Heuristic proposed in [dF20]	Our heuristic	Delta
Precision	~0.756	~0.980	~0.224
Recall	~0.989	~0.999	~0.1
F1-Score	~0.857	~0.989	~0.132

Table 3.1: Precision, recall, and F1-score for the heuristic proposed in [dF20], our proposed heuristic, and the delta between both evaluated against the ground truth data established using the static coordinator heuristic from [dF19] & [dF20] for blocks 1 to 609999.

3.2 Detecting Samourai Whirlpool Transactions

As already mentioned in Section 3.1, Ficsór published not only a detection heuristic for Wasabi Wallet transactions, but also one for Samourai Wallet in order to compare the transaction count and volume mixed for both wallets. According to the published heuristic, a transaction is a Samourai Whirlpool CoinJoin if [dF19]:

- The number of inputs of the transaction is equal to 5
- The number of outputs of the transaction is equal to 5
- All outputs have the same value and this value equals one of the Samourai Whirlpool sizes (0.01, 0.05, or 0.5 BTC) ±0.0011 BTC

This heuristic in general has remained the same in [dF20] with two differences:

- Samourai Wallet started to support a pool size of 0.001 BTC with version *v0.99.96e*, this pool size was subsequently added to the heuristic. As this was only implemented after our last recorded block (658738), this pool size is not considered in our analysis going forward.
- The maximum difference of the output value to the pool sizes changed from 0.0011 BTC to 0.01 BTC.

While it is not possible to establish a similar *ground truth* as was achieved for Wasabi Wallet CoinJoin transactions and described in Section 3.1, all detected transactions before block 570000, which was the first block to feature Samourai Whirlpool transactions [dF20], are obviously false positives. Our implementation of the heuristic described in [dF20] (i.e., with a maximum difference of the output value compared to pool sizes of 0.01 BTC) has detected **39** transactions between blocks 1 and 569999, all of which must be false positive results, and 84619 transactions from block 570000 to block 658738. Using a maximum difference of 0.0011 BTC as described in [dF19] changed the number of detected transactions before block 570000 to **16** while 84,610 transactions were detected from block 570000 to block 658738. It should also be noted that all transactions detected

by [dF19] were detected by [dF20], i.e., the same 84,610 transactions were detected by both heuristics. [dF20] detected the 9 additional potential Samourai Whirlpool CoinJoin transactions which, after manual inspection, were revealed to not be valid Samourai Wallet Whirlpool CoinJoin transactions, i.e., are false positive detections.

However, both heuristics suffer from an inaccuracy, namely that the output values of a Samourai Whirlpool CoinJoin transaction are always uniform and equal exactly the Whirlpool pool size (i.e., the outputs of a CoinJoin are equal to exactly 0.01, 0.05, or 0.5 BTC). Moreover, the input values of a Samourai Whirlpool CoinJoin are between n and $n + m$, where n is the size of the Whirlpool pool, and m is the transaction fee. Naturally, the value of n is constant with all inputs of a CoinJoin transaction, and the value of m depends on whether the transaction is a *remix* or *premix* input.

In order to improve the accuracy of these heuristics, we propose the following heuristic:

- The number of inputs and outputs of the transaction is equal to 5.
- At least one and at most three inputs are *remix* addresses, i.e., there are 1, 2, or 3 inputs with a value exactly equal to a Samourai Whirlpool pool size.
- At least two and at most four inputs are *premix* addresses, i.e., there are 2, 3, or 4 inputs with a value between a Samourai Whirlpool pool size and the pool size plus a certain amount which makes up the transaction fee (with a maximum difference of 0.0011 BTC as per [dF19]).
- The uniform value of all outputs is exactly equal to a Samourai Whirlpool pool size.

Obviously, the pool size has to remain stable for all inputs of the transaction, i.e., all remix and premix inputs of a single transaction should be based on the same pool size. Using the proposed heuristic, we were able to identify 84,596 Samourai Whirlpool CoinJoin transactions from block 570000 to block 658738. We have also tested our heuristic against all transactions from block 1 to block 569999 and did not detect any transactions, i.e., we did not find any false positives before block 570000. Table 3.2 shows a comparison of the detected results for all three heuristics.

Note that, according to [Sam19], the number of *remix* inputs for each Samourai Whirlpool transaction is 1 - 2, and the number of *premix* inputs is 3 - 4 (i.e., 1 *remix* and 4 *premix* inputs, or 2 *remix* and 3 *premix* inputs). Applying these constraints, however, reduced the number of detected transactions to 14,604. Manually analyzing transactions which were detected by our heuristic (i.e., 1 - 3 *remix* and 2 - 4 *premix* inputs) but not by these stricter constraints shows that the detected transactions still appear to be Samourai Whirlpool CoinJoins. The constraints were therefore relaxed as described above.

Our heuristic still suffers from an inaccuracy, however, as the inputs for the first transaction for every pool can only be *premix* addresses. The transaction IDs for the genesis mixes according to [Lau19] and further refined through our own analysis are:

Heuristic	**Block #** < 570000	**Block #** ≥ 570000
[dF19]	16	84610
[dF20]	39	84619
Our proposal	0	84596

Table 3.2: Comparison of found Samourai Wallet Whirlpool transactions for [dF19], [dF20] (with the 0.001 BTC pool excluded) and our proposal. Note that the 6 genesis genesis mixes are not included in our result.

- **c6c27bef217583cca5f89de86e0cd7d8b546844f800da91d91a74039c3b40fba** for the 0.01 BTC pool
- **94b0da89431d8bd74f1134d8152ed1c7c4f83375e63bc79f19cf293800a83f52** for the 0.05 BTC pool
- **b42df707a3d876b24a22b0199e18dc39aba2eafa6dbeaaf9dd23d925bb379c59** for the 0.5 BTC pool

We are now able to use these genesis mixes in order to further validate our heuristic, as every Samourai Whirlpool CoinJoin other than the genesis mixes will have at least one *remix* address as described in Section 2.6.3. Therefore, all transactions identified by the heuristic should have a *link* to a genesis mix. Indeed, we were able to trace 84,590 out of the 84,596 identified transactions to an appropriate genesis pool, with 55,383 transactions being traced to the 0.01 BTC pool genesis mix, 25,597 transactions being traced to the 0.05 BTC genesis mix, 3610 transactions being traced to the 0.5 BTC genesis mix and the following 6 transactions not being traceable to a genesis mix:

1. **148e84427ff117ed15332fb905a6059a43561965c3eecf81b8e7072143e44dc2**
2. **904c932350daa03e960e7e9db22488794a55ac3ca5b5031a71d924f53f9be700**
3. **451e09ac808ab8b75ad7a948c33d70ef3bfc27aa36dd84a65444bd41ef2bec86**
4. **f195770bf0a077453546a42a4d44263286b8c2ad3fbf29852241e857ef8bc849**
5. **41852e2758d9dedaadaf8644af784b39028bdb417837a5ecf7ec32c44891d2db**
6. **57ee479d2fd48cc10430d24c7d5efda3ed2fd484c7d054ac05922bcba1dbff08**

Manual inspection of these six transactions shows that (1) belongs to the 0.01 BTC pool and features 2 *remix* transactions which themselves are structured like genesis mixes (i.e., 5 *premix* inputs and uniform outputs of 0.01 BTC). Transactions (2) - (4) belong to the 0.05 BTC pool and feature 1 *input* transaction which is structured like a genesis mix. Transaction (5) is very similar, and in fact is traceable to the same "genesis" mix as (2) - (4), but only after 2 hops. Transaction (6) also belongs to the 0.05 BTC pool but

Pool	Genesis TX ID
0.01	**c6c27bef217583cca5f89de86e0cd7d8b546844f800da91d91a74039c3b40fba**
0.01	4c906f897467c7ed8690576edfcaf8b1fb516d154ef6506a2c4cab2c48821728
0.01	a4259682535205584194 9a8270eda6fb37566a8780b2aec6b49d8035955d060e
0.05	**94b0da89431d8bd74f1134d8152ed1c7c4f83375e63bc79f19cf293800a83f52**
0.05	a554db794560458c102bab0af99773883df13bc66ad287c29610ad9bac138926
0.05	792c0bfde7f6bf023ff239660fb876315826a0a52fd32e78ea732057789b2be0
0.5	**b42df707a3d876b24a22b0199e18dc39aba2eafa6dbeaaf9dd23d925bb379c59**

Table 3.3: All discovered Samourai Whirlpool genesis mix transactions with the "main" genesis mixes (i.e., the genesis mix for the vast majority of Samourai Whirlpool transactions) being listed in **bold**.

features a different "genesis" mix as (2) - (5). All of these discovered genesis mixes are also listed by [Lau19] and may have been used for testing. Table 3.3 lists the previously described "main" genesis mixes as well as these 4 additional genesis-like mixes.

CHAPTER 4

Analysis of Mixing Schemes in the Bitcoin Ecosystem

Building on the work of Chapter 3, this chapter explores the role of Wasabi Wallet and Samourai Wallet within the greater Bitcoin ecosystem. To that end, the chapter first presents various metrics and statistics such as the amount mixed and the activity (i.e., number of CoinJoin transactions) of both services in Section 4.1. Furthermore, the amount of entities participating in these mixes, as well as possible relations of these entities to each other, and to known services such as Bitcoin exchanges is analyzed in Section .

4.1 Longitudinal Analysis

This Section provides a general overview of various metrics and statistics for both Wasabi Wallet CoinJoin, as well as Samourai Whirlpool CoinJoin transactions. The analyzed activity period for Wasabi Wallet stretches from block 530500 to block 658738, while the period for Samourai stretches from block 570000 to block 658738.

The converted fiat values are taken from the *GraphSense Cryptoasset Analytics Platform*[1] [HSRK21] and reflect the historic exchange rates of Bitcoin.

4.1.1 Wasabi Wallet

Number of CoinJoin Transactions

From blocks 530500 to block 658738, there were a total of 18,687 Wasabi Wallet CoinJoin transactions with an average of 79 inputs and 128 outputs per transaction. Grouping the

[1] https://graphsense.info

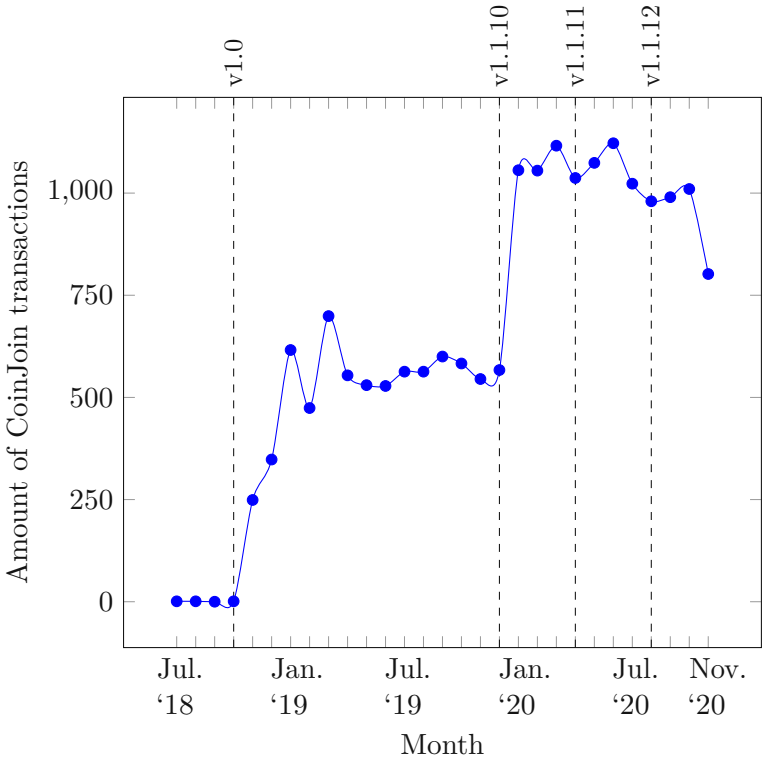

Figure 4.1: Amount of Wasabi Wallet CoinJoin transactions per month from July 2018 to November 2020.

transactions by month shows that Wasabi CoinJoins didn't truly start until November 2018, which isn't surprising considering that the official release date for Wasabi Wallet was 31.10.2018[2]. The three CoinJoin transactions before this date were likely final tests by the Wasabi Wallet developers. Figure 4.1 provides a graphical representation of the number of transactions every month from July 2018 until November 2020.

The number of Wasabi Wallet CoinJoin transactions appears to remain fairly constant, slowly increasing to almost 700 transactions in March 2019 before settling at roughly 500-600 transactions throughout the remainder of 2019. Curiously, in January 2020 the amount of Wasabi Wallet CoinJoin transactions rose to over 1000 transactions and continued to remain at this level throughout the rest of 2020. This spike in activity coincides with the release of Wasabi Wallet v1.1.10[3], which featured a number of improvements to the core functionality of Wasabi due to a refactoring of the block-, transaction-, and coin-processing, as well as updates to the GUI.

Whether this release is the cause for the increase of Wasabi Wallet CoinJoin transactions

[2] https://github.com/zkSNACKs/WalletWasabi/releases/tag/v1.0
[3] https://github.com/zkSNACKs/WalletWasabi/releases/tag/v1.1.10

is unclear, however, as the release dates of other major versions such as v1.1.11[4] (released on 05.04.2020) and v1.1.12[5] (released on 05.08.2020) do not correlate with an increased amount of CoinJoin transactions.

Amount Mixed by Wasabi

Wasabi Wallet CoinJoin transactions feature a total output volume of ~606,019.18 BTC. In regard to fiat currency, Wasabi Wallet CoinJoin outputs have totaled ~4.66 billion EUR or ~5.26 billion USD. Note that these numbers still include the coordinator fees and change outputs and therefore do not equal the amount of mixed bitcoins.

The amount of outgoing bitcoins correlates roughly with the number of CoinJoin transactions, with two deviations:

1. In August 2019, the number of transactions remained fairly constant (563 transactions vs. 563 in July and 600 in September 2019) while the amount of BTC spiked sharply to ~35,362.61 BTC (with the values for July and September being ~13,358.33 BTC and ~26,200.61 BTC respectively).

2. In March and April 2020, the output volume of CoinJoin transactions rose to ~45,646.05 BTC and ~47,006.22 BTC (versus ~31,907.63 in February and ~35,696.03 in May 2020), while the number of CoinJoin transactions again remained fairly constant. This spike was followed by a rapid drop to ~26,769.56 BTC in June 2020. After a small increase to ~32,021.67 BTC in July 2020, the output volume has been steadily decreasing until November 2020.

These output volumes of Wasabi Wallet CoinJoins, graphically represented in Figure 4.2, fluctuate much more notably than the total number of Wasabi Wallet CoinJoin transactions. One of the reasons for this might be the high volatility of the Bitcoin price when compared to fiat currencies. Indeed, when considering the output volume in Euro, the spike during March/April 2020 is no longer notable: ~281.84 million EUR in February, ~282.68 million EUR in March, ~303.53 million EUR in April, and ~301.52 million EUR in May 2020. This spike therefore seems to correlate with the fluctuating BTC/EUR conversion rate.

This correlation does not hold for the first spike in August 2019, however. The output volume for Wasabi CoinJoin transactions in EUR for July, August, and September 2019 was ~127.09 million EUR, ~338.80 million EUR, and ~239.75 million EUR respectively. Figure 4.3 shows the output volume for Wasabi Wallet CoinJoin transactions in millions of EUR.

[4] https://github.com/zkSNACKs/WalletWasabi/releases/tag/v1.1.11
[5] https://github.com/zkSNACKs/WalletWasabi/releases/tag/v1.1.12

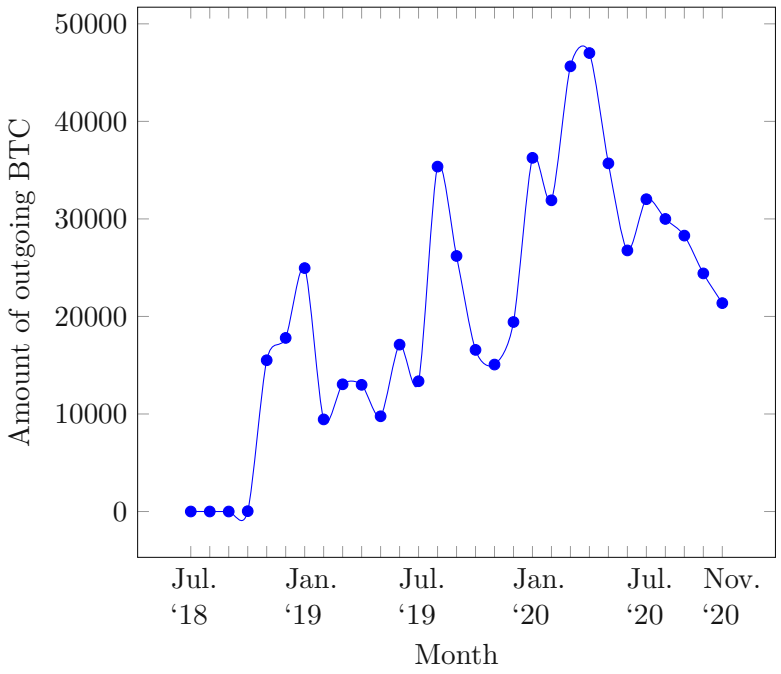

Figure 4.2: Amount of outgoing BTC of Wasabi Wallet CoinJoins per month from July 2018 to November 2020.

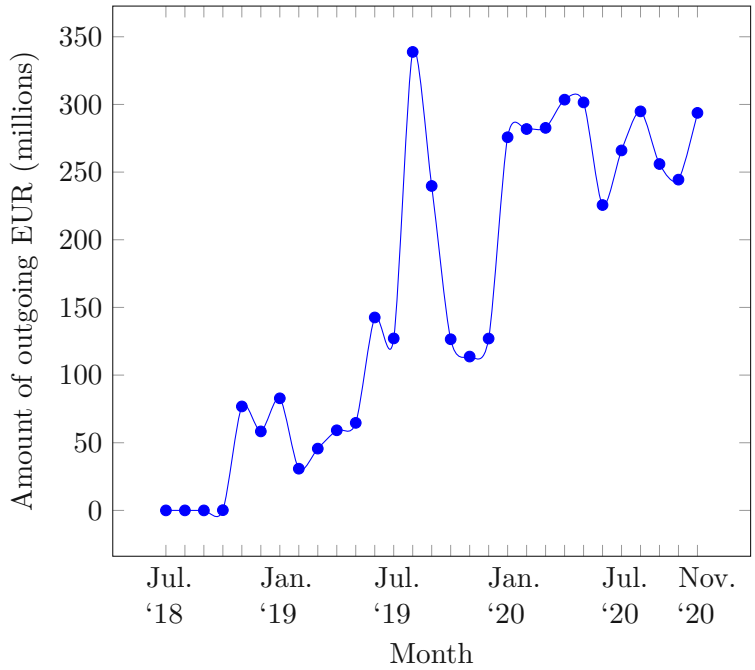

Figure 4.3: Amount of outgoing EUR (in millions) of Wasabi Wallet CoinJoins per month from July 2018 to November 2020.

Amount of Mixed Outputs Leaving Wasabi Mixes

In order to detect the actual amount of mixed bitcoins and their associated addresses, the following factors have to be considered:

- An output must leave the Wasabi Wallet ecosystem, i.e., it must not be a *remix*.
- It must be a mix output, not a coordinator fee or a change output.

In order to satisfy the first requirement, an output must not serve as an input for a future Wasabi Wallet CoinJoin transaction, while the second condition can be considered satisfied if there are at least two outputs with the same value[6]. Parsing all identified Wasabi Wallet CoinJoin transactions for outputs matching these criteria, we have discovered that, in total, 953,508 addresses have received coins which have left the Wasabi CoinJoin ecosystem, of which 922,568 were unique (i.e., 30,940 addresses have received bitcoins leaving Wasabi CoinJoins multiple times). These addresses, as well as their associated entities, are explored further in Section 4.2. The total amount of mixed coins to leave the ecosystem equals ~143,713.24 BTC (~1.19 bn EUR or ~1.35 bn USD). Figure 4.4 shows the amount of mixed BTC leaving the Wasabi Wallet ecosystem per month from July 2018 to November 2020.

While the amount of mixed BTC leaving Wasabi rises steadily throughout the analyzed time frame, August and September 2019 feature a drastic increase. While ~3350.93 mixed BTC have left Wasabi in July 2019, the numbers for August and September 2019 are ~17,405.95 and ~12,145.88 respectively. A spike in this time frame was also present in the previously analyzed amount of outgoing EUR. The reason for this spike is unclear, but it appears that a large amount of coins have left the Wasabi Wallet ecosystem in August and September 2019.

Amount of Fresh Inputs Entering Wasabi Mixes

In similar fashion, it is possible to determine the amount of fresh Wasabi Wallet CoinJoin inputs, i.e., CoinJoin inputs which are not *remix* inputs. An input address can be considered fresh if the address does not occur as a mix output of any previous Wasabi Wallet CoinJoin transaction. In total, 1,467,519 addresses were used as inputs for Wasabi Wallet CoinJoin transactions, 647,935 of which were *remix* inputs while the remaining 819,584 input addresses were fresh, i.e., they have not been used as output addresses of previous mixes. Of these fresh input addresses, 789,441 input addresses were unique, with 18,503 addresses being re-used at least once. While the vast majority, 18,394 of the re-used input addresses were used less than 10 times, 55 were re-used between 10 and 24 times, 27 were re-used between 25 and 49 times, and another 27 were re-used over 50

[6]Recall that while we have used the Wasabi Wallet base denomination of ~0.1 BTC to identify Wasabi Wallet CoinJoin transactions, a CoinJoin can feature outputs of multiple denominations as described in Section 2.6.2.

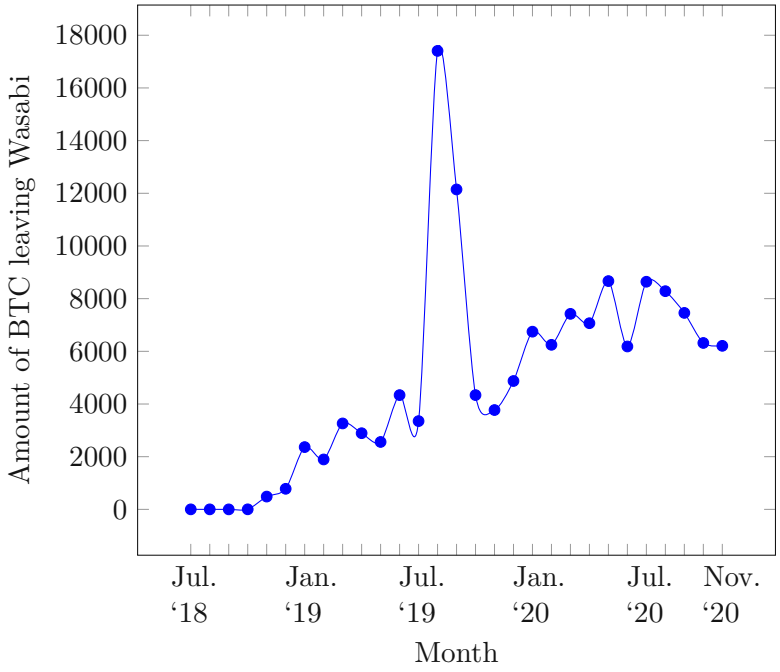

Figure 4.4: Amount of mixed BTC leaving the Wasabi Wallet ecosystem per month from July 2018 to November 2020.

times. The exact nature of the entities behind these input addresses is explored further in Section 4.2.

Figure 4.5 shows the amount of fresh BTC put into the Wasabi Wallet ecosystem per month. This figure reveals three separate spikes in fresh inputs - adjusting for the fluctuating BTC price, Figure 4.6 shows the amount of fresh EUR (in millions). The spikes in November 2018 and January 2020 fit the release dates of Wasabi Wallet v1.0 and v1.1.10 respectively, which further indicates that these releases may have caused an increased usage of Wasabi Wallet. The third spike, which occurred in August and September 2019, has also been observed in the amount of mixed BTC leaving the Wasabi Wallet ecosystem. This could therefore indicate a *one-off* mixing of a large amount of coins by some entities.

Wasabi Income & Fees Paid by Users

As described in Section 2.6.2, Wasabi Wallet charges a coordinator fee for every CoinJoin transaction, which participants have to pay in addition to miner fees. Before Wasabi Wallet introduced freshly generated coordinator addresses for every transaction, as explained in Section 3.1, two static coordinator addresses were used to collect all fees which are listed in Table 4.1. Since the switch to freshly generated coordinator addresses per CoinJoin transaction, it has become increasingly difficult to determine the coordinator

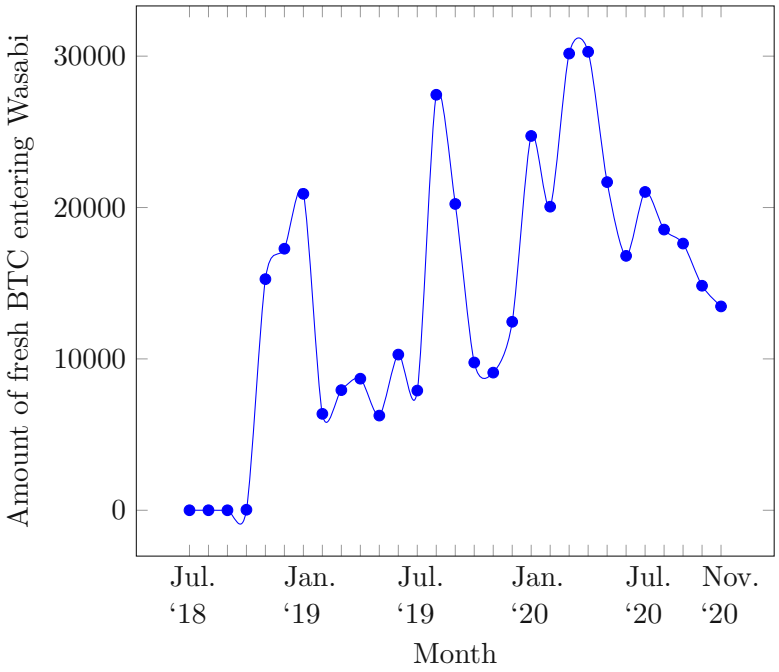

Figure 4.5: Amount of fresh BTC entering the Wasabi Wallet ecosystem per month from July 2018 to November 2020.

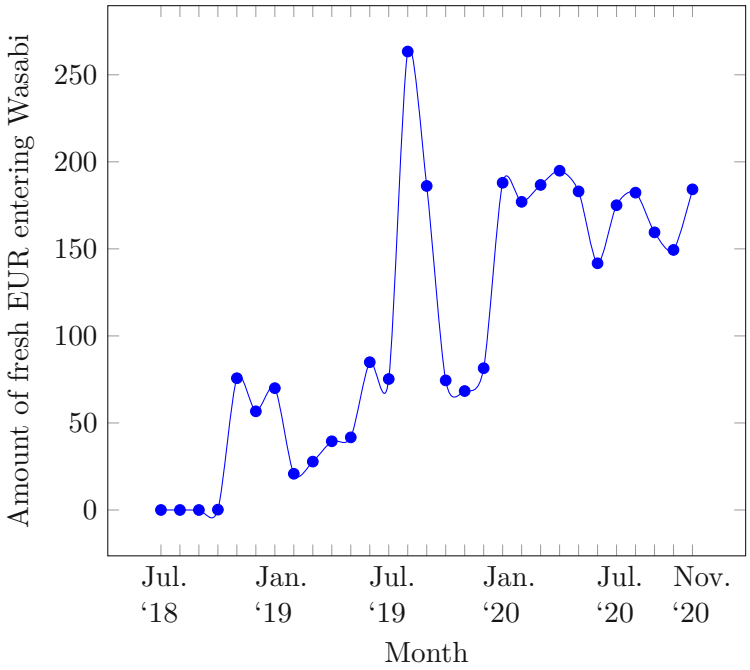

Figure 4.6: Amount of fresh EUR (in millions) entering the Wasabi Wallet ecosystem per month from July 2018 to November 2020.

Address	BTC in	EUR in	USD in
bc1qa24tsgchvuxsaccp8vrnkfd85hrcpafg20kmjw	~40.43	~304,953.69	~337,634.97
bc1qs604c7jv6amk4cxqlnvuxv26hv3e48cds4m0ew	~71.66	~519,514.72	~582,151.00
Total	~112.09	~824,468.41	~919,785.97

Table 4.1: Fees collected by the historic static coordinator addresses of Wasabi Wallet.

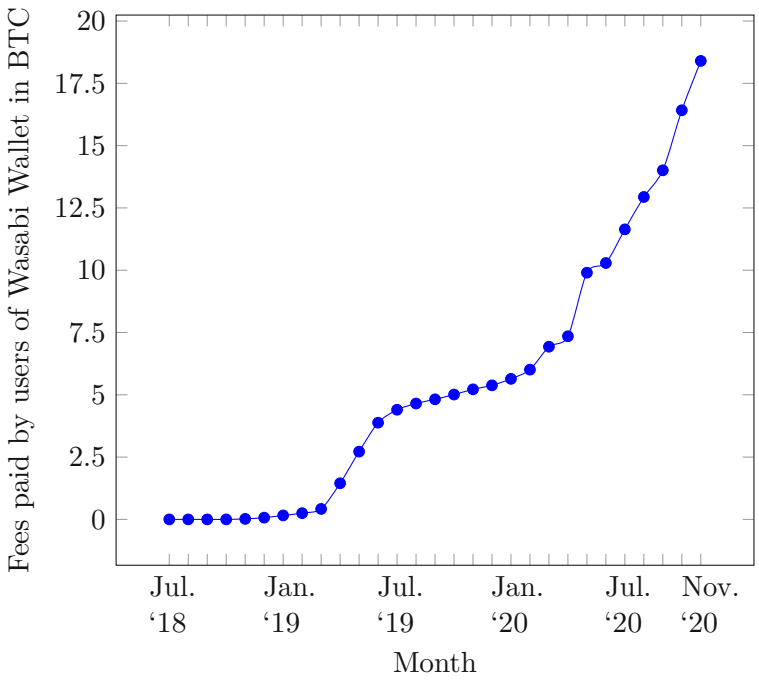

Figure 4.7: Cumulative miner fees paid by Wasabi Wallet CoinJoin users per month from July 2018 to November 2020 in BTC.

output of a Wasabi Wallet CoinJoin. Figure 4.7 displays the cumulative amount of miner fees paid by users.

4.1.2 Samourai Wallet

Number of Samourai Whirlpool Transactions

A total of 84603 Samourai Whirlpool transactions (including genesis mixes) have been discovered between blocks 570000 and 658738, with every transaction featuring exactly 5 input and 5 output addresses. By pool size, 55,387 transactions have occurred in the 0.01 BTC pool, while 25,605 and 3611 transactions have been observed in the 0.05 BTC and 0.5 BTC pools respectively. Figure 4.8 provides a graphical representation of the number of transactions for each pool grouped by month from April 2019 to November 2020.

The amount of Samourai Whirlpool transactions feature a notable spike in March 2020.

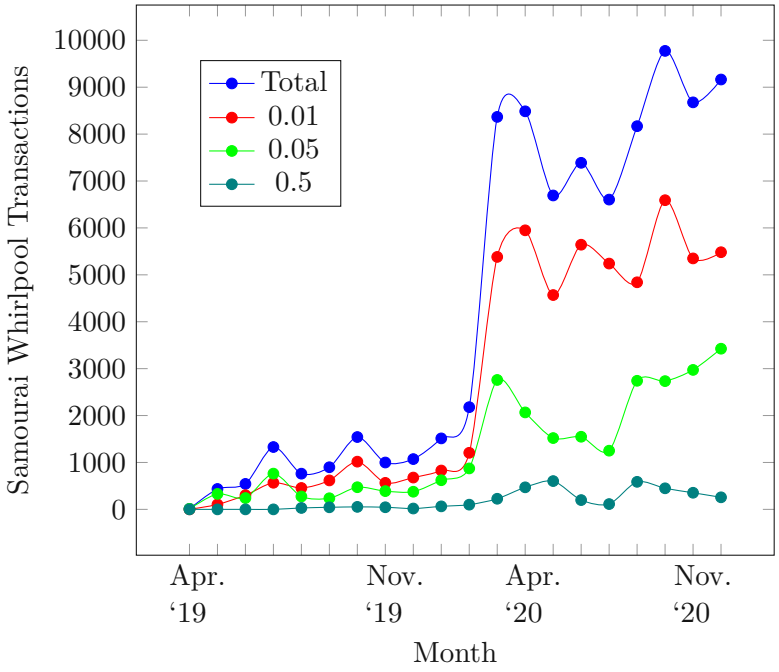

Figure 4.8: Amount of Samourai Whirlpool transactions per month from April 2019 to November 2020.

Pool	# Transactions (Share of Total)	BTC output (Share of Total)
0.01	55387 (~65.47%)	~2769.35 (~15.22%)
0.05	25605 (~30.26%)	~6401.25 (~35.18%)
0.5	3611 (~4.27%)	~9027.5 (~49.61%)
Total	84603 (100%)	~18,198.1 (100%)

Table 4.2: Number of Samourai Whirlpool transactions and outgoing BTC per pool size.

The reason for this sudden increase in transactions is unclear.

Amount Mixed by Samourai Whirlpool

Samourai Whirlpool transactions have mixed a total of ~18,198.1 BTC (~162.6 million EUR or ~186.35 million USD). While the 0.01 BTC pool is used for the majority of all Whirlpool transactions (~65.47%) while only accounting for ~15.22% of BTC outputs. The 0.05 pool is responsible for roughly a third of all transactions and outputs (~30.26% and ~35.18% respectively), while the 0.5 BTC pool is only used in ~4.27% of all Samourai Whirlpool transactions but features almost half of all outgoing BTC (~49.61%). Table 4.2 shows the relation of transactions and outputs for each pool size while Figure 4.9 provides a graphical representation.

Figures 4.10 and 4.11 represent the output volume of Samourai Whirlpool per month in

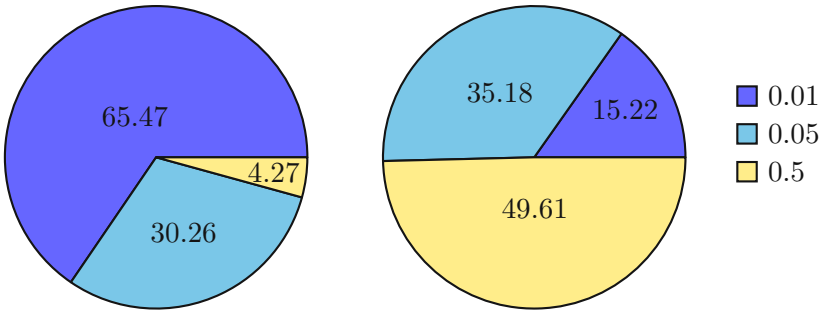

Figure 4.9: Share of each Samourai Whirlpool pool in the number of transactions (left) and output volume (right).

BTC and EUR (millions) respectively. The spike in the amount of transactions in March 2020 is also present in the output volume, regardless of whether the output is interpreted in BTC or EUR. Interestingly, while the amount of transactions slightly drops after April 2020, the drop in output volume is much higher, as is the subsequent rise in output volume after July 2020.

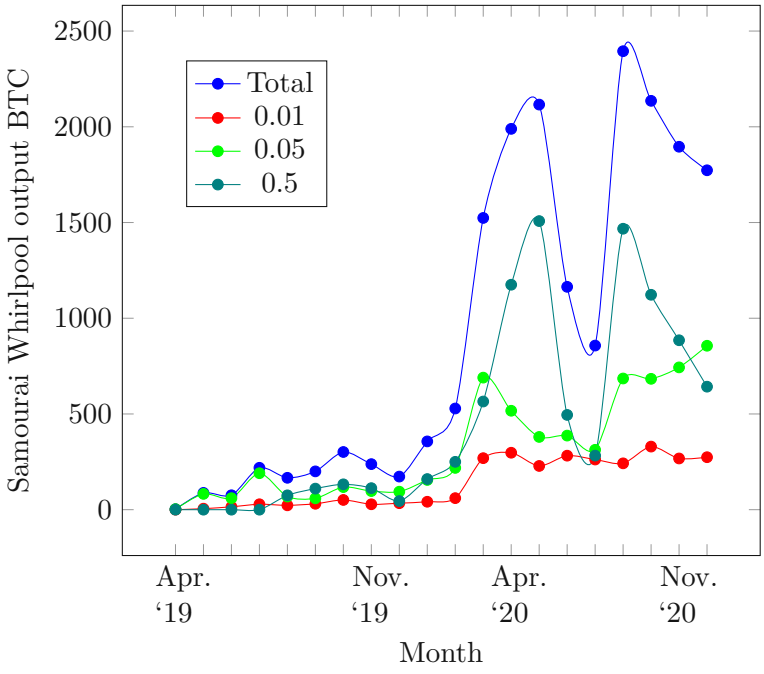

Figure 4.10: Amount of BTC put out by Samourai Whirlpool per month from April 2019 to November 2020.

Another interesting aspect arises when the individual pools are compared. As the 0.01 BTC pool is responsible for over 65% of all Whirlpool transactions, it heavily influences

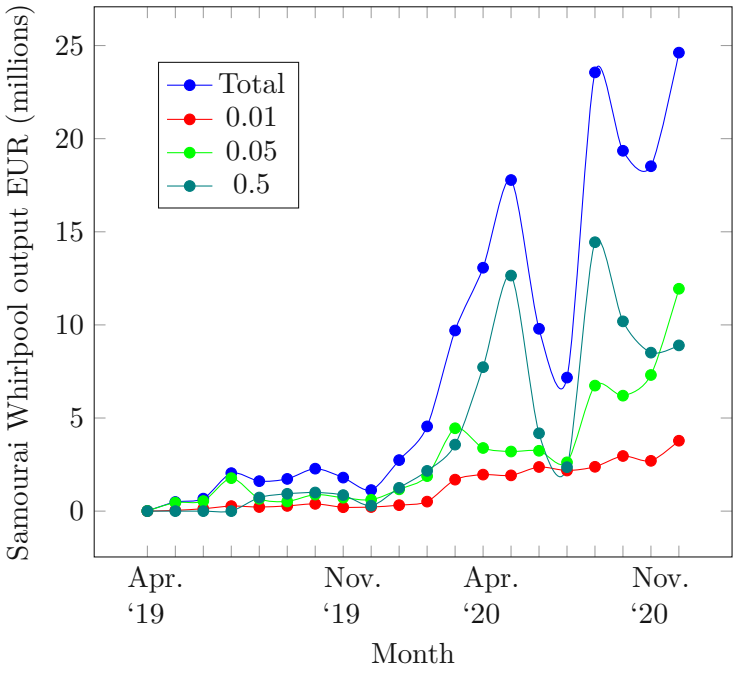

Figure 4.11: Amount of EUR (in millions) put out by Samourai Whirlpool per month from April 2019 to November 2020.

the overall amount of CoinJoin transactions, while the 0.5 BTC pool, which is responsible for almost 50% of all output volume, heavily influences the overall output volume. Even more interesting is the fact that the output volume for the 0.01 and 0.05 BTC pools is much more steadily rising, with peaks being much less pronounced than for the 0.5 BTC pool.

Amount of Mixed Outputs Leaving Samourai Whirlpool

In contrast to Wasabi Wallet CoinJoins, Samourai Whirlpool CoinJoin transaction outputs are always either being used as a *remix* input in a future Whirlpool transaction, or they are leaving the Samourai Whirlpool ecosystem. Therefore, it suffices to enumerate all Samourai Whirlpool outputs which are not re-used as an input in future Samourai Whirlpool CoinJoin transactions (i.e., all non-*remix* outputs of a Whirlpool CoinJoin).

By pool size, the 0.01 BTC pool features 276,935 outputs, all of which are unique. This is hardly surprising, as 5 fresh output addresses are generated per CoinJoin transaction and the 0.01 BTC pool has been used by 55,387 transactions ($55387 * 5 = 276935$). The same holds for the 0.05 BTC pool which features 128,025 unique outputs and the 0.5 BTC pool which features 18,055 unique outputs. Filtering outputs which are being used as inputs in subsequent Whirlpool CoinJoin transactions yields 120,356 output addresses for the 0.01 BTC pool, 56,616 output addresses for the 0.05 pool, and 7585 output addresses

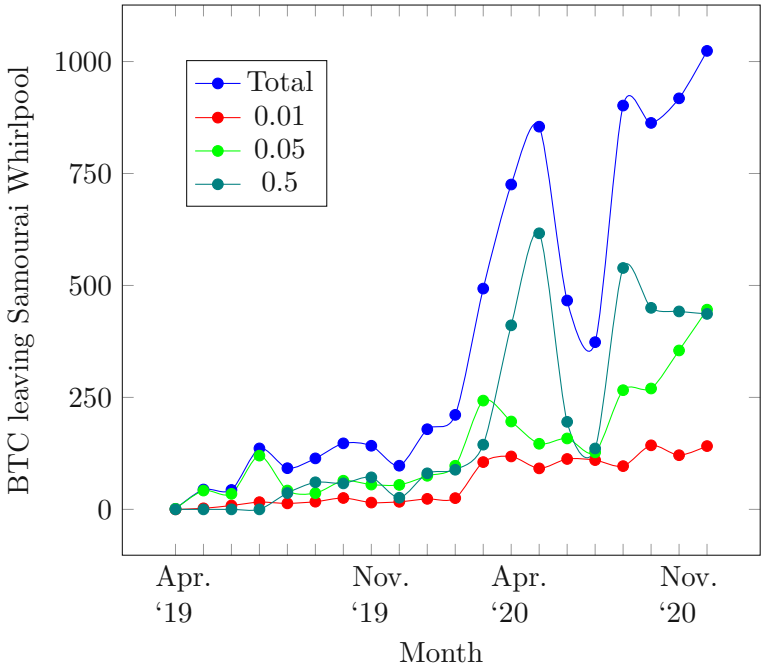

Figure 4.12: Amount of mixed BTC leaving Samourai Whirlpool per month from April 2019 to November 2020.

for the 0.5 BTC pool which are not used as future *remix* inputs.

Figure 4.12 shows the amount of BTC leaving Samourai Whirlpool per month from April 2019 to November 2020. These values co-evolve very strongly with the overall outputs seen in Figure 4.10.

Amount of Fresh Inputs Entering Samourai Whirlpool

As with Wasabi Wallet, it is possible to detect fresh CoinJoin inputs by detecting all relevant Samourai Whirlpool transaction inputs which have not received coins by Samourai Whirlpool CoinJoins themselves. Doing so reveals that 238,458 of the 423,015 Samourai Whirlpool input addresses are *remix* inputs, while the remaining 184,557 input addresses are fresh inputs, 183,759 of which are unique. The number of re-used input addresses is lower than for Wasabi Wallet CoinJoin transactions, with 776 addresses being used twice and 11 addresses being re-used three times.

Figure 4.13 shows the amount of fresh bitcoins entering the Samourai Whirlpool ecosystem per month and per pool. The number of fresh inputs co-evolves with the number of mix outputs leaving Samourai Whirlpool. In October and November 2020, however, the number of total fresh inputs appears to be dropping while the number of mix outputs leaving seems to rise.

It is also possible to analyze the *Tx0* transactions themselves, and look for *Tx0* outputs

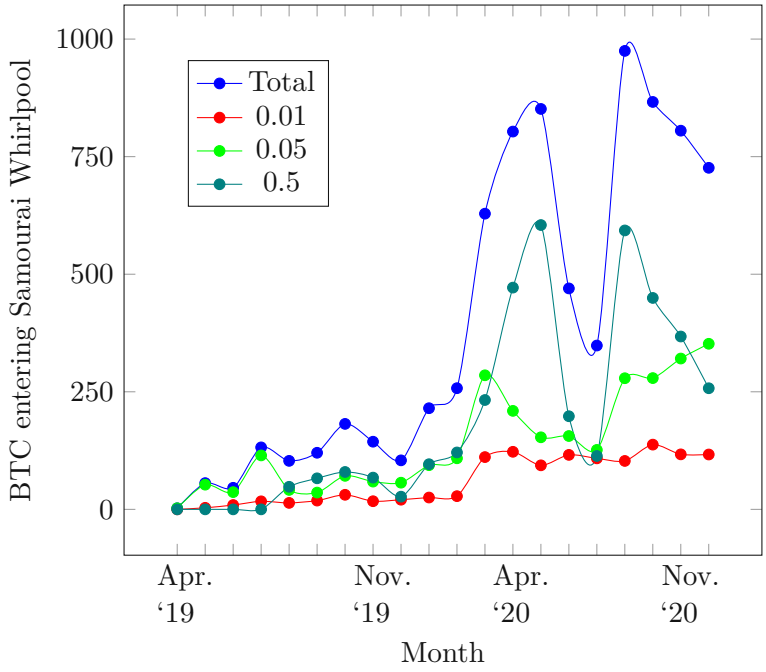

Figure 4.13: Amount of fresh BTC entering Samourai Whirlpool per month from April 2019 to November 2020.

which not (yet) been spent in Samourai Whirlpool transactions. In total, 16,635 such outputs have been observed for the 0.01 BTC pool, 5141 for the 0.05 BTC pool, and 710 for the 0.5 BTC pool.

Samourai Whirlpool Income & Fees Paid by Users

As Samourai Wallet charges fees for *Tx0* transactions and not for the CoinJoin transactions themselves, the income of Samourai Wallet can be calculated by analyzing all previously discovered *Tx0* transactions. As stated in Section 2.6.3, Samourai Wallet charges a fee of $5\% * s$, where s is the pool size[7]. The total fees for all pools are therefore 0.0005, 0.0025, and 0.025 BTC for the 0.01, 0.05, and 0.5 BTC pools respectively. The total income of Samourai Whirlpool, and by extension the coordinator fees paid by users, is therefore the sum of all *Tx0* outputs that feature their respective pool fee and equals ~40.08 BTC. Table 4.3 shows the coordinator fees collected for *Tx0* transactions, the miner fees paid in CoinJoin transactions, as well as the total amount of fees paid for by users per pool.

It appears that a new collector address is generated by Samourai Whirlpool for the vast majority of all *Tx0* transactions, across all pools. On rare occasions, however, addresses

[7]The fees were reduced by 30% for the 0.05 and 0.5 BTC pools on March 17th, 2021. This date is outside the scope of the collected transactions however, and is therefore not relevant for this thesis [Sam21].

Pool	Coordinator Fees	Miner Fees	Total Fees
0.01	~8.62 BTC (~21.51%)	~6.03 BTC (~66.48%)	~14.65 BTC (~29.8%)
0.05	~13.54 BTC (~33.78%)	~2.68 BTC (~29.55%)	~16.22 BTC (~32.99%)
0.5	~17.93 BTC (~44.74%)	~0.36 BTC (~3.97%)	~18.29 BTC (~37.21%)
Total	~40.08 BTC (100%)	~9.07 BTC (100%)	~49.16 BTC (100%)

Table 4.3: Coordinator, miner, and total fees per Samourai Whirlpool pool.

have been reused. Moreover, 2929 *Tx0* transactions have been identified whose outputs have been used in Samourai Whirlpool CoinJoin transactions, but where the fees did not match any official amount. These transactions, as well as those whose collector addresses were reused, may have been due to tests conducted by the developers of Samourai.

4.2 Entity Network Analysis

As shown in Section 2.4, Bitcoin addresses can be clustered into entities using various heuristics. While we assume that the CoinJoin mixes for both Wasabi and Samourai cannot be undone, this Section explores which entities send coins into these mixes, and which entities receive mixed coins. GraphSense is used to extract entity information from input and output addresses, the ID assigned to entities is therefore the same as the entity ID in GraphSense. Also note that, as an entity analysis for remix transaction will not yield relevant results, only fresh inputs entering, and mixed outputs leaving the ecosystem of both wallets are considered. Furthermore, entities which feature an in-degree or out-degree of at least 100 are categorized as services. The in-degree of an entity refers to the number of entities than send coins to this entity, while the out-degree is the number of entities that receive coins from the entity.

After mapping the input and output addresses to entities, GraphSense will be utilized to further find all neighbors, and neighbors' neighbors, of these directly participating entities, unless the entity has been categorized as a service. Entities which directly participate in CoinJoin transactions are labeled *Level 0*, while their neighbors and neighbors' neighbors are labeled *Level 1* and *Level 2* respectively. To be more precise, the entity analysis for all CoinJoin transactions entails:

- For all fresh inputs:
  - Identify which entities send fresh inputs directly into CoinJoin transactions (*Level 0*)
  - Identify which entities send inputs to *Level 0* entities (*Level 1*)
  - Identify which entities send inputs to *Level 1* entities (*Level 2*)
- For all mixed outputs:

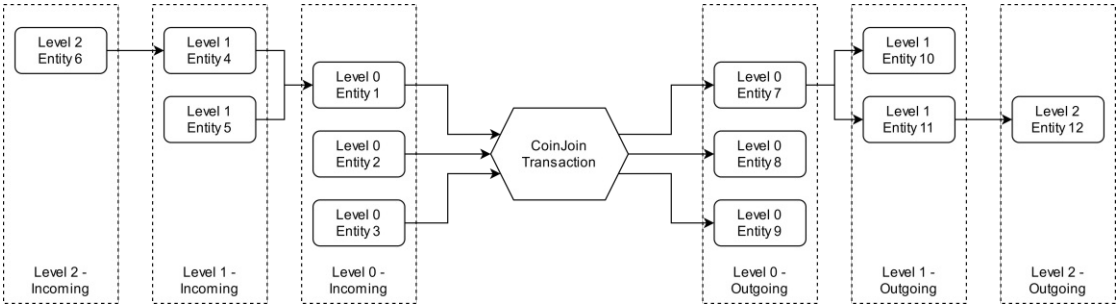

Figure 4.14: Incoming and outgoing entity levels *Level 0*, *Level 1*, and *Level 2* visualized.

- Identify the entities which receive mixed outputs directly from CoinJoin transactions (*Level 0*)
- Identify which entities receive coins from *Level 0* entities (*Level 1*)
- Identify which entities receive coins from *Level 1* entities (*Level 2*)

Figure 4.14 visualizes this process. The discovered *Level 0* entities are subsequently ranked based on how many addresses of an entity deposit fresh inputs directly into CoinJoin transactions and how many addresses of an entity receive outputs directly from CoinJoins. This ranking is referred to as *Level 0 Score*.

Entities will be grouped into the following categories, with the categorization being supplied by GraphSense:

- *Mixing Service* - Mixing services, both centralized and decentralized
- *Exchange* - Exchange services
- *Crime* - Entities related to service hacks, scams, ransomware and other extortion
- *Wallet Service* - Bitcoin wallet services
- *Mining Pool* - Bitcoin miners and mining pools
- *Ponzi Schemes* - Entities related to known Ponzi schemes
- *Gambling* - Known gambling services
- *Market* - Bitcoin markets
- *Organization* - Organizations
- *Service* - Other services

Distance	Direction	# Entities	# Tagged Entities	# Sanitized Entities
Level 0	Incoming	45,251	2	45,251
	Outgoing	298,059	10	298,059
Level 1	Incoming	74,488	18	46,322
	Outgoing	221,096	72	205,145
Level 2	Incoming	147,276	141	77,407
	Outgoing	312,862	854	236,403

Table 4.4: The number of identified entities, those which have been assigned at least one tag in GraphSense, and the number of sanitized entities that are participating in Wasabi Wallet CoinJoins within a maximum of two hops.

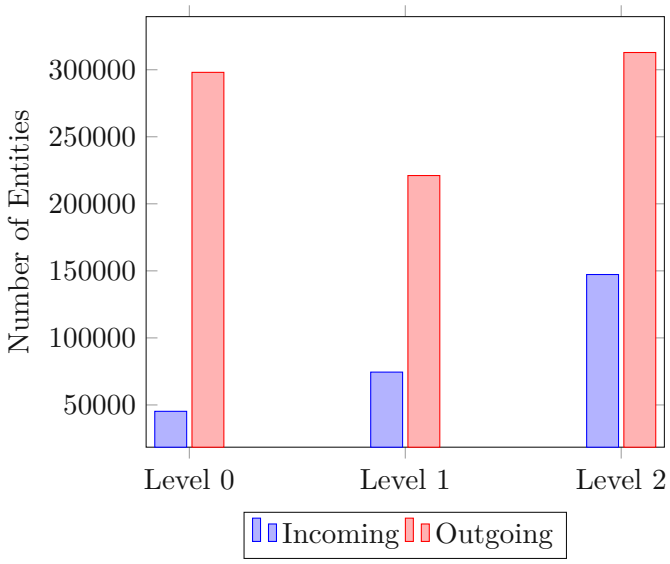

Figure 4.15: The number of entities participating in Wasabi Wallet CoinJoin transactions.

4.2.1 Entities Participating in Wasabi Wallet CoinJoins

In total, 876,696 different entities have participated in Wasabi Wallet CoinJoins either directly (*Level 0*), or within one to two hops (*Level 1* and *Level 2*). Of these, 739,607 entities have received coins, i.e., are outgoing entities, and 168,980 have sent coins, i.e., are incoming entities. While the majority of the discovered entities have not been assigned a tag in GraphSense, 864 outgoing and 141 incoming entities have been tagged. Figure 4.15 provides a graphical visualization of the number of Wasabi Wallet CoinJoin transaction participants.

The number of detected outgoing entities is larger than the number of incoming entities at every distance. This is not necessarily a surprise, especially at *Level 0*, as the purpose of CoinJoins is to disrupt clustering techniques. Moreover, it is clear that some entities are multiple distances away from, or may be in either direction of the CoinJoin, e.g., a

Entity	Level 0 Score	Tags	Incoming CoinJoin Address Percentage
597931459	696612	Wasabi Fee	696612/1713222 (~40.66%)
447051516	375	-	375/482 (~77.8%)
508051014	358	bittrex	358/1621941 (~0.02%)
446758067	289	-	289/400 (~72.25%)
511745626	270	-	270/514 (~52.53%)

Table 4.5: The five highest scored identified incoming Wasabi Wallet CoinJoin *Level 0* entities.

single entity could be a *Level 1* and *Level 0* incoming entity, as well as a *Level 2* outgoing entity.

Analyzing these intersecting entities by testing whether the same entity appears in multiple sets reveals that 28,166 incoming entities have been classified as both *Level 0* and *Level 1*, 31,048 entities have been classified as both *Level 0* and *Level 2*, and 64,208 entities have been classified as both *Level 1* and *Level 2*. When excluding incoming entities which have already been classified as a lower level, 46,322 entities remain at *Level 1* and 77,407 entities remain at *Level 2*.

Reviewing outgoing entities in an analog fashion shows 15,951 redundant entities between *Level 0* and *Level 1*, 39,990 redundant entities between *Level 0* and *Level 2*, and 47,193 redundant entities between *Level 1* and *Level 2*. Excluding these redundant entities results in 205,145 entities at *Level 1* and 236,403 entities at *Level 2*. Table 4.4 provides an overview of the number of participating entities in Wasabi Wallet CoinJoin transactions.

Another interesting question is whether entities appear in both directions from a CoinJoin transaction, and at what frequency. Comparing the set of all incoming entities with the set of all outgoing entities results in 31,891 intersecting entities, i.e., 31,891 entities appear as both incoming and outgoing entities.

Incoming Entities

As previously discussed, 168,980 distinct entities have been discovered sending coins into Wasbai Wallet CoinJoin transactions either directly, or within two hops. Of these, GraphSense has assigned tags to 141 entities while 168,839 remain untagged.

The five highest ranked entities at *Level 0* are entities **597931459**, **447051516**, **508051014**, **446758067**, and **511745626**. Table 4.5 shows these entities with their score and tags. No incoming entities at *Level 0* other than **597931459** and **508051014** have been assigned a tag in GraphSense.

Table 4.6 groups all discovered incoming entities across *Level 0*, *Level 1*, and *Level 2* for which at least one tag has been assigned in GraphSense into categories.

Category	Level 0	Level 1	Level 2	Total (distinct)
Mixing Service	1	2	2	2
Exchange	1	10	102	102
Crime	0	3	12	12
Wallet Service	0	2	5	5
Mining Pool	0	2	7	7
Ponzi Scheme	0	1	4	4
Gambling	0	0	3	3
Market	0	0	0	0
Organization	0	0	0	0
Service	0	0	4	4

Table 4.6: Incoming Wasabi Wallet CoinJoin entities as categorized by GraphSense.

Entity	Level 0 Score	Tags	Incoming CoinJoin Address Percentage
597931459	157848	Wasabi Fee	157569/1713222 (~9.21%)
600678446	753	-	753/773 (~97.41%)
600349759	487	-	487/488 (~99.8%)
801074967	432	-	432/873 (~49.48%)
642055290	385	-	385/391 (~98.47%)

Table 4.7: The five highest scored outgoing Wasabi Wallet CoinJoin *Level 0* entities.

Outgoing Entities

Of the 739,607 distinct outgoing entities receiving coins from Wasabi Wallet CoinJoin transactions across a maximum of two hops, GraphSense has tagged a total of 864 entities.

The highest ranked outgoing *Level 0* entity is once again entity **597931459** (*Wasabi Fee*). This is hardly surprising, as the coordinator fee is paid in every Wasabi Wallet CoinJoin transaction. Entities **600678446**, **600349759**, **801074967**, and **642055290** are the next highest rated outgoing *Level 0* entities and are shown in Table 4.7 together with their achieved scores. Of these top 5 entities, only **597931459** has been tagged in GraphSense. In fact, only a total of 10 outgoing entities at *Level 0* have been assigned a tag. These tagged entities, together with all other tagged entities for *Level 1* and *Level 2* are grouped into categories and listed in Table 4.8.

Entities Sending and Receiving Coins from Wasabi Wallet

Comparing the incoming and outgoing entities, it is apparent that some entities both send and receive coins from Wasabi Wallet CoinJoin transactions, i.e., some entities are both incoming and outgoing entities. As previously stated, the number of these entities is 31,891, of which 41 have been assigned at least one tag in GraphSense. As seen from tables 4.5 and 4.7, the highest scoring *Level 0* entity in both directions is entity **597931459**, tagged as *Wasabi Fee*. As stated before, this is not surprising

Category	Level 0	Level 1	Level 2	Total (distinct)
Mixing Service	2	3	5	5
Exchange	4	25	80	80
Crime	5	18	24	24
Wallet Service	0	4	4	5
Mining Pool	1	4	7	7
Ponzi Scheme	0	3	5	5
Gambling	0	4	10	10
Market	0	0	4	4
Organization	0	1	1	1
Service	0	3	10	10

Table 4.8: Outgoing Wasabi Wallet CoinJoin entities as categorized by GraphSense.

Category	Incoming only	Outgoing only	Incoming & Outgoing
Mixing Service	0	3	2
Exchange	80	58	22
Crime	1	13	11
Wallet Service	1	1	4
Mining Pool	2	2	5
Ponzi Scheme	0	1	4
Gambling	0	7	3
Market	0	4	0
Organization	0	1	0
Service	3	9	1

Table 4.9: The intersection of incoming and outgoing Wasabi Wallet CoinJoin entities as categorized by GraphSense.

for the outgoing direction when considering that the coordinator fee is paid in every CoinJoin transaction. Interestingly, the same entity is also the most dominant incoming entity. This is due to the fact that Wasabi Wallet may inflate the anonymity set of CoinJoin transactions. Specifically, if only a low number of users register their coins for a CoinJoin, Wasabi Wallet may register their own coins to increase the number of CoinJoin participants, thereby increasing the perceived anonymity set [LE20c]. The implications of this are further discussed in Section 5.3.1.

There are 41 participating entities across *Level 0*, *Level 1*, and *Level 2* which are present in both directions of a Wasabi Wallet CoinJoin transaction for which tags are available in GraphSense. Table 4.9 provides an overview over the categories of identified and tagged entities that are both incoming and outgoing Wasabi Wallet CoinJoin participants.

Summary

While most identified entities have not been assigned any tags in GraphSense and are therefore reduced to nameless clusters, it is still possible to infer information from analyzing the participants of Wasabi Wallet CoinJoin transactions. Figure 4.16 shows the relationship between the 10 highest ranked incoming and outgoing *Level 0* entities, except entity **597931459** (*Wasabi Fee*), with their *Level 1* and *Level 2* neighbors. It is interesting to note that almost all entities are connected after at least two hops by entity **597931459**, although many entities are also connected through other neighbors.

According to the tags provided by GraphSense, addresses of 5 mixing services (including the two *Wasabi Fee* entities), 160 exchange services, 6 wallet services, 9 mining pools, 1 organization, 4 markets, 10 gambling services, as well as 13 other tagged services have either directly participated in Wasabi Wallet CoinJoin transactions, or are within close proximity. Furthermore, entities which have been connected to illicit activities have also used Wasabi Wallet CoinJoin transactions, namely 5 entities related to Ponzi schemes, and 25 entities directly tied to criminal activities (20 entities tagged as service hack, 2 as sextortion[8], and 3 as ransomware).

It is especially interesting to see that 31,891 entities have been identified as both incoming and outgoing entities, including 6 entities connected to a major security breach of the Binance exchange, and 2 entities tied to the Lazarus group, a "U.S.-designated North Korean state-sponsored malicious cyber group" [U.S20].

4.2.2 Entities Participating in Samourai Whirlpool

Using the clustering heuristics of GraphSense, 343,463 entities can be observed participating in Samourai Whirlpool transactions at either *Level 0*, *Level 1*, or *Level 2*. Of these, a total of 152,695 entities can be classified as incoming and 213,016 entities can be classified as outgoing entities, with 7332 incoming and 816 outgoing entities having been assigned at least one tag in GraphSense.

As with Wasabi Wallet CoinJoins, the total number of distinct outgoing entities is larger than the number of incoming entities, although the difference is smaller than in Wasabi Wallet. However, looking at the number of distinct incoming and outgoing entities per *Level* reveals that the number of incoming entities at *Level 1* and *Level 2* is actually greater than the number of respective outgoing entities. This suggests that a large number of entities are simultaneously multiple distances from a Whirlpool transaction. Indeed, when filtering out incoming entities which have been classified as belonging to a lower level, 28,112 entities remain at *Level 1* and 40,864 entities remain at *Level 2*. Doing the same for outgoing entities gives 52,035 entities at *Level 1* and 69,054 entities at *Level 2*. Table 4.10 displays the number of incoming and outgoing entities while Figure 4.17 provides a graphical visualization.

[8] *Sextortion* refers to sexual extortion as described in [PRHC19].

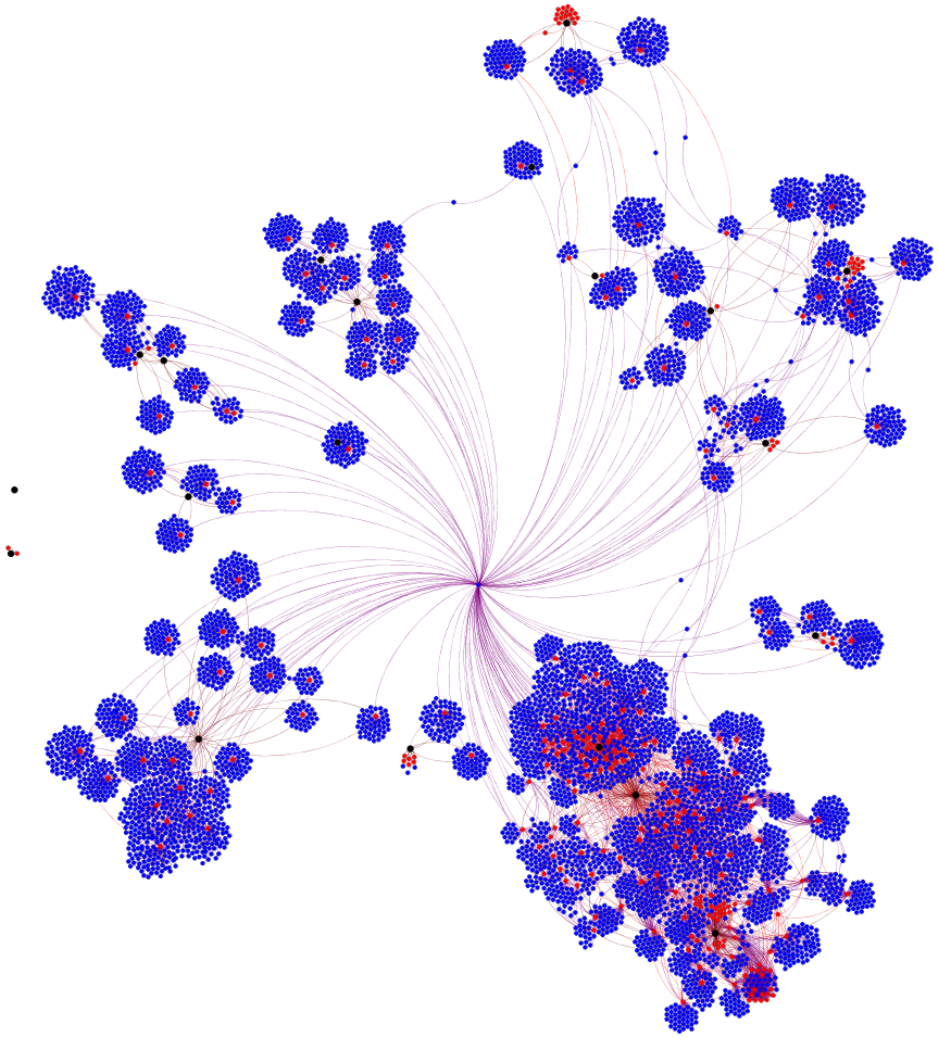

Figure 4.16: The relationships between the 10 highest ranked incoming and outgoing Wasabi Wallet CoinJoin *Level 0* (black) entities with their *Level 1* (red) and *Level 2* (blue) neighbors.

Distance	Direction	# Entities	# Tagged Entities	# Sanitized Entities
Level 0	Incoming	83,719	7308	83,719
	Outgoing	91,927	10	91,927
Level 1	Incoming	110,806	7091	28,112
	Outgoing	58,323	61	52,035
Level 2	Incoming	151,255	7077	40,864
	Outgoing	102,387	811	69,054

Table 4.10: The number of identified entities, those which have been assigned at least one tag in GraphSense, and the number of sanitized entities that are participating in Samourai Whirlpool transactions within a maximum of two hops.

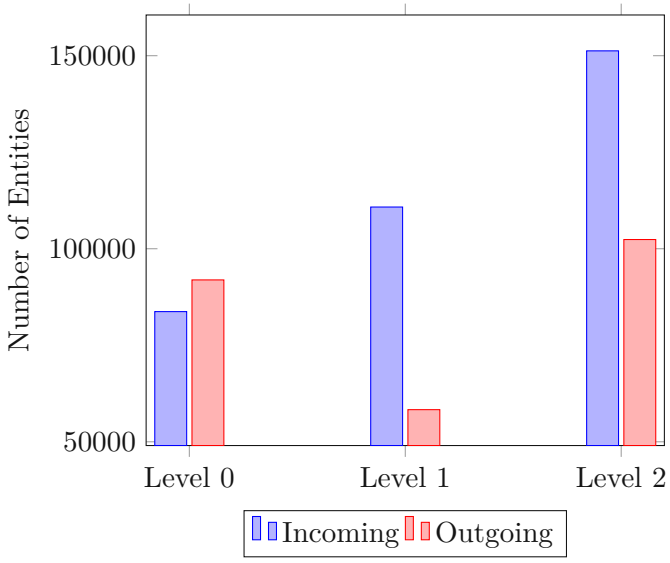

Figure 4.17: The number of entities participating in Samourai Whirlpool transactions.

The number of entities that can be classified as both incoming and outgoing entities is 22,219.

Incoming Entities

A total of 152,695 unique entities have been classified as incoming across *Level 0*, *Level 1*, and *Level 2*. 7332 of these entities have been assigned at least one tag in GraphSense, while the rest have not been tagged.

At *Level 0*, i.e., entities directly participating with Whirlpool transactions, 83,719 entities have been observed, of which 7308 entities have been tagged. It should be noted, however, that all of these tagged entities bear the tag *Samourai Wallet*, which has been assigned based on the original heuristics of Ficsór [dF19]. The five highest ranked entities, none of which have been assigned a tag, are listed in Table 4.11.

Entity	Level 0 Score	Tags	Incoming CoinJoin Address Percentage
708033617	115	-	115/291 (~39.52%)
730252977	14	-	14/67 (~20.9%)
689579035	8	-	8/39 (~20.51%)
685658470	8	-	8/67 (~11.94%)
724359066	7	-	7/25 (~28%)

Table 4.11: The five highest scored incoming Samourai Whirlpool *Level 0* entities.

Category	Level 0	Level 1	Level 2	Total (distinct)
Mixing Service	0	1	1	1
Exchange	0	0	14	14
Crime	0	0	10	10
Wallet Service	0	0	2	2
Mining Pool	0	0	4	4
Ponzi Scheme	0	0	3	3
Gambling	0	0	0	0
Market	0	0	0	0
Organization	0	0	0	0
Service	0	0	0	0

Table 4.12: Incoming Samourai Whirlpool entities as categorized by GraphSense.

Entity	Level 0 Score	Tags	Incoming CoinJoin Address Percentage
641862176	468	-	468/478 (~97.9%)
735477944	399	-	399/401 (~99.5%)
671300966	370	-	370/371 (~99.7%)
731723348	347	-	347/348 (~99.7%)
637768989	344	-	344/347 (~99.1%)

Table 4.13: The five highest scored outgoing Samourai Whirlpool *Level 0* entities.

Table 4.12 shows all tagged entities for *Level 1* and *Level 2* (excluding those tagged as *Samourai Wallet*) grouped into categories.

Outgoing Entities

As previously discussed, 213,016 entities have been discovered which receive coins from Samourai Whirlpool transactions, with 816 entities having been tagged in GraphSense, 743 of which bear the tag *Samourai Wallet*.

Only one of the 91,927 outgoing *Level 0* entities has been assigned a tag in GraphSense that is not *Samourai Wallet*, namely entity **597931459** (*Wasabi Fee*) which achieved a score of 14. Table 4.13 shows the five highest ranked outgoing *Level 0* entities.

Category	Level 0	Level 1	Level 2	Total (distinct)
Mixing Service	1	1	2	2
Exchange	0	16	39	39
Crime	0	16	20	20
Wallet Service	0	2	5	5
Mining Pool	0	4	5	5
Ponzi Scheme	0	3	3	3
Gambling	0	2	3	3
Market	0	0	0	0
Organization	0	0	0	0
Service	0	1	6	6

Table 4.14: Outgoing Samourai Whirlpool entities as categorized by GraphSense.

Category	Incoming only	Outgoing only	Incoming & Outgoing
Mixing Service	0	1	1
Exchange	0	25	14
Crime	0	10	10
Wallet Service	0	3	2
Mining Pool	1	2	3
Ponzi Scheme	0	0	3
Gambling	0	3	0
Market	0	0	0
Organization	0	0	0
Service	0	6	0

Table 4.15: Samourai Whirlpool entities as categorized by GraphSense that participate in CoinJoin transactions as incoming and outgoing entities.

Other than the aforementioned entity **597931459**, 72 additional entities have been assigned a tag that is not *Samourai Wallet*. Table 4.14 shows the categories these entities are assigned to.

Entities Sending and Receiving Coins from Samourai Whirlpool

As with Wasabi Wallet, a number of entities participate in Samourai Whirlpool transactions both as incoming and outgoing entity. To be more precise, 22,248 entities have been observed to either directly participate in Whirlpool transactions, or appear as neighbors, 766 of which have been assigned at least one tag in GraphSense. Discounting the 743 entities which have been tagged as *Samourai Wallet*, 23 tagged entities remain and are categorized in Table 4.15.

Summary

As with Wasabi Wallet, most identified entities have not been assigned any tags in GraphSense. Figure 4.18 shows the relationship between the 10 highest ranked *Level 0* entities with their *Level 1* and *Level 2* neighbors. In contrast to the relationship graph of Wasabi Wallet participants, the graph for Samourai Whirlpool participants appears much more scattered. It is still interesting to note that 9 entities are connected to each other over at most two hops. Furthermore, entities **674422225** and **661985180** are also connected via entity **668044737** at *Level 2*.

The tags provided by GraphSense show that addresses of 2 mixing services (including the main *Wasabi Fee* entity), 39 exchange services, 5 wallet services, 6 mining pools, 3 gambling services, as well as 6 other tagged services have either directly participated in Samourai Whirlpool transactions, or are within close proximity. Furthermore, entities which have been connected to illicit activities have also used Wasabi Wallet CoinJoin transactions, namely 3 entities tied to Ponzi schemes, and 20 entities directly related to criminal activities (15 entities tagged as service hack, 2 as sextortion, 1 as scam, and 2 as ransomware).

Finally, 22,248 entities are both incoming and outgoing Samourai Whirlpool entities, 10 of which are directly related to criminal activities, and all of which are also within two hops of Wasabi Wallet CoinJoin transactions.

4.2.3 Entities Participating in Wasabi Wallet CoinJoins and Samourai Whirlpool

Some entities have been observed participating in both Wasabi Wallet CoinJoin and Samourai Whirlpool transactions across *Level 0*, *Level 1*, and *Level 2*. In total, 427 entities can be classified as incoming entities to both services, while 7803 entities can be classified as outgoing entities. Excluding those tagged as *Samourai Wallet*, 24 incoming and 63 outgoing entities have been assigned at least one tag in GraphSense. Moreover, 111 entities have participated in both services as both an incoming and outgoing entity, 22 of which have been tagged. Table 4.16 shows the categories these entities belong to.

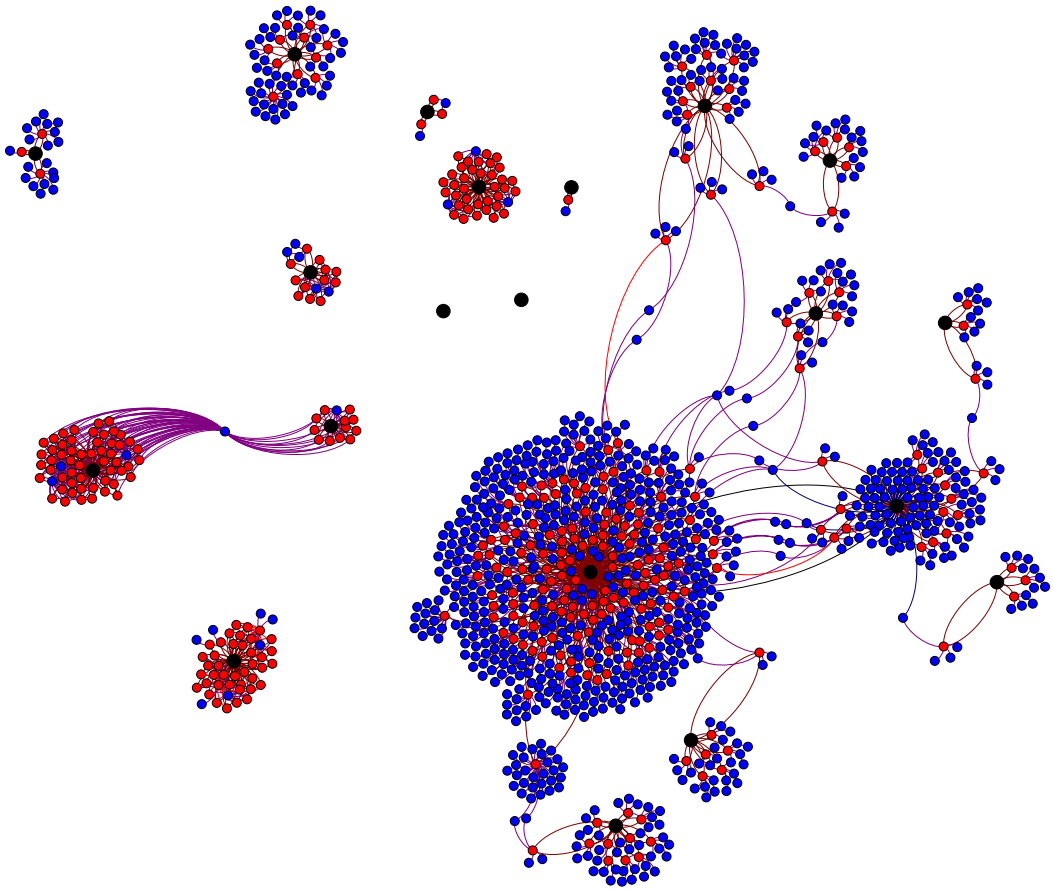

Figure 4.18: The relationships between the 10 highest ranked incoming and outgoing Samourai Whirlpool *Level 0* (black) entities with their *Level 1* (red) and *Level 2* (blue) neighbors.

Category	Wasabi only	Samourai only	Wasabi & Samourai
Mixing Service	3	0	2
Exchange	129	8	31
Crime	6	1	19
Wallet Service	1	0	5
Mining Pool	3	0	6
Ponzi Scheme	2	0	3
Gambling	7	0	3
Market	4	0	0
Organization	1	0	0
Service	7	0	6

Table 4.16: The categories of tagged entities participating in Wasabi Wallet CoinJoin transactions, Samourai Whirlpool transactions, or both.

CHAPTER 5

Security & Privacy Review

This chapter discusses and reviews both Wasabi Wallet and Samourai Wallet for potential security related weaknesses. Section 5.1 provides an overview of the fundamentals of information security and how they relate to Bitcoin wallets. Building upon these fundamentals, Section 5.2 explores the security of the ZeroLink framework, the theoretical foundation for the CoinJoin implementations of both Wasabi Wallet and Samourai Wallet, in depth. Finally, Section 5.3 analyzes known public advisories concerning the security of both wallets.

5.1 Fundamentals & Attack Vectors

The fundamental elements of information security are often stated to be confidentiality, integrity, and availability (abbreviated as CIA, or the CIA triad). While the CIA triad has sometimes been criticized for missing various aspects relevant to information security such as authenticity and non-repudiation, it is still widely used when establishing or auditing information systems [WM14].

Obviously, while the overarching goals of information security may often be very similar, they should be accurately specified when dealing with concrete systems. In the case of Bitcoin coin mixing (e.g., the ZeroLink framework), a malicious adversary must not be able to:

1. Gain any information on which inputs are related to which mixing outputs (i.e., reverse the mixing process, de-anonymizing participants)
2. Access any funds of other participants (i.e., steal coins of other participants)
3. Interfere with the mixing process, preventing participants from completing the mix

These goals can roughly be mapped to the CIA triad, with (1) being mapped to confidentiality (an adversary is unable to access protected information), (2) being mapped to integrity (an adversary is unable to manipulate information in an undetected manner), and (3) being mapped to availability (users are able to access information).

Another aspect to consider in information security is whether an adversary is passive, i.e., only observes information emitted by a system, or whether they actively accesses and manipulate the system in question. In regards to Bitcoin coin mixing, this would translate into an attacker merely observing CoinJoin transactions broadcast in the network versus an attacker actively participating in the mixing process. A passive attacker could furthermore be able to observe non-Bitcoin information, such as IP-network traffic, while an active adversary could participate in the mixing process in any role.

5.2 Security of the ZeroLink Framework

With the information security goals of Bitcoin coin mixing specified in Section 5.1, this section examines the ZeroLink framework, explained in Section 2.6.1 in regards to the threats of de-anonymization, coin theft, and denial-of-service for both a passive and an active adversary. Note, however, that both coin theft and denial-of-service are inherently active attacks and a passive adversary can, by definition, only observe. Therefore, Section 5.2.2 and Section 5.2.3 will only consider an active attacker.

5.2.1 De-anonymization

When attempting to de-anonymize participants, an active adversary can see all inputs and all outputs of a CoinJoin transaction, as well as the past history of all input addresses and the potential future of all output addresses. Moreover, an adversary might be able to access internet traffic including payloads in general. Recall that the Chaumian CoinJoin protocol of the ZeroLink framework consists of the following phases: the input registration phase, the output registration phase, and the signing phase, as well as a connection confirmation phase. As the connection confirmation phase exists to confirm registered users, no data concerning the CoinJoin itself is transmitted [dFT17].

During the input registration phase, however, an adversary may be able to access the inputs, blinded CoinJoin output, and the change output sent by users, as well as the signed blinded CoinJoin output returned by the tumbler to every user. As the CoinJoin output is cryptographically blinded by the user and only unblinded locally, a passive adversary would only be able to access the blinded change outputs and inputs. In the output registration phase, users who transmit the unblinded CoinJoin outputs to the tumbler should do so as another identity (e.g., using a different Tor circuit or a different VPN connection). An adversary would then be able to access these unblinded outputs, but can no longer link them to the original identity used in the input registration phase. Finally, in the signing phase, the tumbler sends the finalized, unsigned CoinJoin transaction to every user that registered an input, while users return the signed CoinJoin

transaction back to the tumbler who proceeds to broadcast the CoinJoin transaction into the Bitcoin network [dFT17]. A passive attacker would therefore be able to access the unsigned CoinJoin output, the individual signatures of observed users, as well as the finalized CoinJoin transaction.

An adversary observing all messages sent between users and the tumbler would be able to establish an association between input addresses, blinded CoinJoin outputs, change outputs, the signatures for all users registering in the input registration phase, as well as an association between the unblinded CoinJoin outputs and all users registering in the output registration phase. However, provided that the blinding of the CoinJoin output is reasonably secure, and that users establish two separate channels for the input and output registration phase, an adversary would only be able to link the inputs to the unblinded CoinJoin outputs if they were able to de-anonymize both channels used by the users. E.g., if a user establishes two distinct Tor connections for the input and output registration phases, an adversary would only be able to link inputs to unblinded outputs if they were able to de-anonymize both Tor connections.

An active adversary could either intercept messages sent by the users and the tumbler by impersonating the tumbler, or act as the tumbler itself. In both cases, an active attacker would not learn any additional information than a passive adversary.

As described in Section 2.5.1 and Section 2.6.1, it becomes increasingly difficult for an observer to establish a link between CoinJoin output addresses and input addresses as more users participate in the mix. While it may be possible to conduct an analysis of the amounts of transaction inputs and outputs in order to de-anonymize users as presented by [Atl14], ZeroLink establishes that common denominations must be used for CoinJoin mix outputs [dFT17]. A purely passive adversary can therefore not gain any knowledge in regards to which input belongs to which output.

Active attackers, on the other hand, are capable of conducting Sybil attacks against any kind of CoinJoin implementation as described in Section 2.5.1. For regular CoinJoin implementations, it is trivial to de-anonymize a user if the adversary participates in the mix and controls all but one input/output pair. The same holds true for the Chaumian CoinJoin. The challenge for an attacker lies in preventing other legitimate users from participating in the mix. If the attacker is not in control of the tumbler, they would need to monitor all messages sent during input registration phases for addresses they wish to de-anonymize, and actively intercept all registrations which feature other inputs. Once a desired input has registered in the mix, they would need to register as many addresses as required to reach the target anonymity set.

Such an attack would likely be detected, however, as all other users would no longer be able to register their inputs. Even if the tumbler was impersonated for all remaining users, the operators of the legitimate tumbler would likely detect the attack as they would not see any input registrations, nor see their own registrations should they participate in a CoinJoin themselves.

Should an attacker be in control of the tumbler, and an address they wish to de-anonymize

registers in the mix, they could refuse to register any further addresses while deregister all previously registered inputs. This deregistration of inputs would be detected by users whose inputs have been deregistered, resulting in a detection of the malicious tumbler. While the tumbler would be able to de-anonymize the user, it would likely no longer be trusted by all other parties and the de-anonymized user could conduct an additional CoinJoin using a different, trusted tumbler [dFT17]. If the tumbler were to refuse to register any non-target inputs from the beginning, there would be no deregistration for other users to detect. As non-target users will never be able to register their inputs, this will likely still lead to suspicion against the tumbler.

Another attack vector for a malicious tumbler would be to create separate input registration phases for every address they wish to de-anonymize. As no users would need to be deregistered, other parties would not detect any anomalous behavior by the tumbler. However, this attack vector is not feasible due to the separate output registration phase - the malicious tumbler would be unable to determine which unblinded output should be mapped to the separated input.

Therefore, the only remaining attack vector for an adversarial tumbler is to de-anonymize every input individually. In order to do so, once any input is registered, the tumbler will refuse to register any other inputs and instead register their own Sybil addresses. This is repeated any time an input is registered. Such an attack will lead to a higher latency for users, however, as their inputs might be refused a number of times before they are registered to participate in a mix, which may lead to suspicion similar to the attack vector in which non-target users are never able to register.

In conclusion, it is only possible to de-anonymize users if a malicious tumbler actively manipulates input registrations. While such an attack may succeed, it would likely lead to retroactive suspicion, which in turn may cause users to mix their coins again using a different tumbler.

5.2.2 Coin Theft

Coin theft in the Chaumian CoinJoin can only occur if the registered outputs of a user are manipulated in a manner that is not evident for the user. Recall that users register their potential change outputs in the input registration phase, their CoinJoin outputs in the output registration phase, and sign the finalized transaction in the signing phase. An adversary that is not the tumbler could intercept network packets in all three phases and manipulate the content, i.e., they could provide different change addresses, different CoinJoin outputs, and forge the unsigned transaction sent from the tumbler to the user.

Provided all cryptographic operations are reasonably secure, the attacker would not be able to forge a CoinJoin output, as these outputs have been digitally signed by the tumbler. Moreover, users are able to verify all outputs of the finalized transaction before they sign it. Due to the nature of digital signatures, an attacker would also not be able to modify the signed CoinJoin transaction. Similarly, even if the attacker is in control of the tumbler, they would be unable to modify any output addresses as users would

discover the modifications and refuse to sign the CoinJoin transaction and, once signed, the transaction can no longer be modified without invalidating all signatures.

It can therefore be concluded that the Chaumian CoinJoin is resistant to coin theft from active adversaries.

5.2.3 Denial-of-Service

Denial-of-Service (or DoS) attacks can be implemented using a wide variety of actions. Obviously, an attacker may conduct network level attacks where network packets are dropped, or the tumbler is overwhelmed by a large number of distributed connection attempts (Distributed Denial-of-Service, or DDoS). These attacks, however, are theoretically possible against every networked service or appliance and are not specific attacks against the Chaumian CoinJoin. On a related note, the tumbler may refuse to register certain inputs, effectively denying service to them. Again, this is not unique to the Chaumian CoinJoin, as most service providers will be able to refuse their service to individual consumers. Therefore, both network-level attacks and denial-of-service by the tumbler itself will not be considered further in the remainder of this section.

[dFT17] lists the following potential DoS attack vectors against the Chaumian CoinJoin:

1. A user registers their input, but spends it in a different transaction before the CoinJoin is finalized.
2. A user registers their input, but refuses to sign the finalized CoinJoin in the signing phase.
3. An input is registered, and a blinded output is returned to the user, but the unblinded output is never registered in the output registration phase
4. An output is never registered in the output registration phase (see (3)), but this output is registered in a future output registration round

Against (1), the ZeroLink framework suggests to ban the malicious input if it is prematurely spent while the mix is still in its input registration phase. Should (1) occur in a later phase, the tumbler must additionally ban all registered outputs and fall back to the input registration phase. Should a user refuse to sign the finalized CoinJoin as in (2), the tumbler should also ban the malicious input and all outputs, and fall back to the input registration phase.

In order to mitigate (3), a blame phase is entered in which all registered inputs must unblind and reveal their original outputs. This way, the malicious input that refused to provide an output is detected and banned, while the tumbler falls back into the input registration phase again. For obvious reasons, users must generate and provide new outputs in the subsequent output registration phase. In fact, [dFT17] states that users must never register the same output in multiple rounds.

Attack vector (4) may follow attack vector (3) if an input is registered, but the output is never provided. While the input may be banned after the blame phase, the signed output acquired by the attacker is never revealed and may be registered in a future output registration round, even if no input is provided. In order to safeguard against this vector, the ZeroLink framework requires tumblers to reject an output if it has already been registered in a previous mix. Therefore, a user can only disrupt a single round with their "malicious" output. Furthermore, the use of a *roundHash* was introduced as a round identifier to further protect against this attack. This *roundHash* is a hash of all inputs, and is provided to all inputs during the connection confirmation phase. The same hash must be provided to the tumbler when registering the output and is once again returned during the signing phase for users to verify. An adversary wanting to supply a wrong output is unaware of the correct value of the *roundHash* and can therefore not register their output [dFT17].

The use of the *roundHash* may not be effective against attackers which are able to monitor network traffic, however, as they may be able to extract the *roundHash* from network payloads and proceed to register outputs in incorrect mixes.

In regards to the banning of malicious inputs against attack vectors (1), (2), and (3), an attacker is able to transfer their funds to a different address and register this new address again, in an attempt to continue disrupting mixing rounds. The cost of this attack would be equal to the transaction fees paid by the attacker as they transfer their funds. A potential defense against such an attack entails using clustering techniques in order to ban new addresses generated by the malicious entity. The ZeroLink framework expands on this by suggesting that a tumbler may ban related UTXO inputs (all other outputs of the same transaction which funded the malicious input), as well as subsequent transaction outputs (i.e., all future outputs funded by transactions in which the malicious input was used). Similarly, a tumbler may ban all other outputs of the parent transactions of the malicious inputs. This, however, is prone to ban potential legitimate users as well [dFT17].

The ZeroLink framework works under the assumption that keeping up DoS attacks will become economically infeasible, as the transaction costs paid for by attackers to circumvent bans rises as the number of banned addresses related to the malicious inputs rises as well. This has two obvious flaws which are partially acknowledged by the authors [dFT17]:

- Some legitimate users may be banned from participating in the mix if their UTXOs are related to those of the attacker. This is especially relevant if the attackers use a large service such as an exchange or a mixing service themselves before starting their attack. Indeed, if an attacker were to participate in a previous round of the Chaumian CoinJoin, they could then execute a DoS attack with malicious inputs which would make the tumbler unable to ban related UTXOs without also banning all potential remix inputs that participated in the same CoinJoin round as the attacker.

- An attacker with a large amount of bitcoins may keep attacking the service before it becomes economically infeasible.

5.3 Implementation & Public Advisories

While both Wasabi Wallet and Samourai Wallet are based on the ZeroLink framework, this section takes a look at publicly available security advisories and related concerns regarding individual design choices and implementations.

5.3.1 Wasabi Wallet

As described in Section 2.6.2, Wasabi Wallet follows the ZeroLink framework very closely. Wasabi Wallet CoinJoins are therefore resistant against coin theft due to cryptographic signatures as explored in Section 5.2.2. Similarly, denial-of-service attack vectors and defenses as listed in Section 5.2.3 also hold for Wasabi Wallet.

In regards to de-anonymization attacks, Section 5.2.1 describes that such an attack requires the tumbler to actively engage in Sybil attacks in order to isolate participants. However, this isolation is only required if the attacker wants to de-anonymize a user with absolute certainty. *ErgoBTC* of OXT Research discusses how entities can be tracked even after using Wasabi Wallet CoinJoins in [LE20c]. Primarily, the following vectors may be exploited to link output to input addresses:

1. Paying the change output of a mix to the same address that receives a mixed output
2. Analyzing volume and timing in order to identify entities
3. Low anonymity set due to low participation

(1) may lead to the de-anonymization of at least the mixed output that also receives the change output. This can be done by the tumbler, who is aware of the connection between the unblinded change address and the input, by an attacker who intercepts the initial input registration, as well as parties that are able to deduce the connection between input and change address through volume analysis of the transaction output. While (1) may initially only lead to the de-anonymization of the one mix output, should this output be used in a joint transaction with other mix outputs (that is not a CoinJoin transaction) it may further de-anonymize these outputs. This de-anonymization of outputs also has an effect on all other outputs (of the same denomination) of the same CoinJoin transaction, as the total anonymity set decreases accordingly [LE20c]. This is true for all CoinJoin transactions.

Attack vector (2) targets entities that mix a much larger amount of BTC than the tumbler usually mixes. While the outputs of this large mix input[1] are initially hidden within the

[1]The input may also be split into several individual inputs.

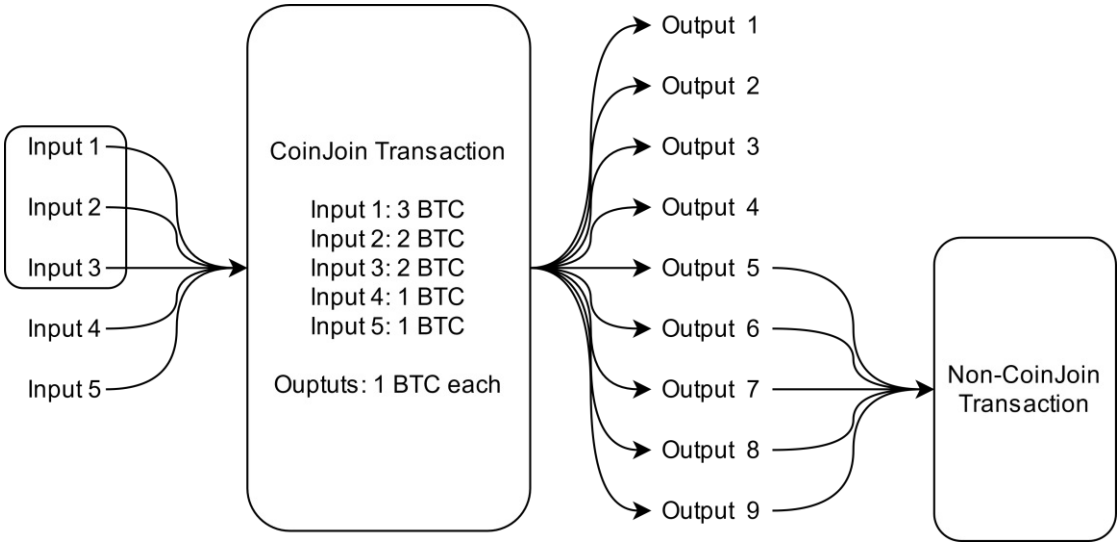

Figure 5.1: A simplified illustration of how CoinJoin outputs can be linked to inputs based on volume. Assuming an observer has already clustered inputs 1-3 into a single entity, they can conclude that 7 outputs holding 1 BTC each must belong to that same entity. If subsequently 5 outputs are jointly used in a non-CoinJoin transaction, they can further deduce that these 5 outputs belong to the same entity as well. This further reduces the effective anonymity set of the remaining outputs to 4 instead of the expected 9.

anonymity set of the mix outputs, if a large number of outputs is subsequently used in a joint non-CoinJoin transaction it can be deduced that these outputs are linked to the large input [LE20c]. Figure 5.1 illustrates this in a simplified fashion.

Finally, (3) concerns the fact that a Wasabi Wallet CoinJoin is conducted if either the target anonymity set is reached (i.e., 100 peers have registered), or an hour has elapsed since the input registration phase started. If the number of participants is low, the anonymity set for all CoinJoin outputs will be low as well. Related to this, Wasabi Wallet can inflate the number of participants by registering inputs under control of the Wasabi Wallet coordinator which increases the anonymity set of all outputs, as explained in Section 4.2. This increase however is not valid for the tumbler who is aware of which addresses were used to inflate the anonymity set. Furthermore, if the outputs of the inputs used for inflating the anonymity set are subsequently linked together the anonymity set for all other output suffers accordingly. As a side note, as Wasabi Wallet collects fees based on the anonymity set of a CoinJoin, users may pay a higher price for a potentially weaker than advertised anonymity set (at least in regards to the tumbler) [LE20c].

Public Advisories

Following Gregory Maxwell's initial CoinJoin proposal, Maxwell, Michael Marquart (*Theymos*), and Pieter Wuille started an escrow bounty fund to reward notable achievements in the field of Bitcoin privacy. From this fund, Wasabi Wallet was awarded 10 BTC for being "the first wallet that implements CoinJoin in both a highly-usable and sound way" [Max13].

However, some parties have voiced their concerns regarding some design issues of the Wasabi Wallet CoinJoin implementation. *LaurentMT* and *ErgoBTC* of OXT Research describe two alleged vulnerabilities in [LE20a], which supposedly cancel out the effect of remixing if an adversary has knowledge of all coins in a users wallet. It should be noted that *zkSnacks*, the developers of Wasabi Wallet disagree with these findings and whether they are truly exploitable [LE20a], [LE20b].

5.3.2 Samourai Wallet

Samourai Whirlpool differs more strongly from the ZeroLink framework than the CoinJoin implementation of Wasabi Wallet does. As described in Section 2.6.3, users conduct an initial transaction (*Tx0*) in order to split their coins into chunks equal to the chosen pool denomination. These individual chunks are then mixed together with at least one remix transaction in rounds with exactly 5 inputs and 5 outputs.

This has little to no effect on the attack vectors regarding coin theft as cryptographic signatures that prevent coin theft are still in place. Denial-of service attack vectors and defenses are also very similar to those explained in Section 5.2.3. The only difference is the initial *Tx0* transaction an attacker would need to conduct before participating in a CoinJoin. As Samourai Whirlpool operators would likely (temporarily) ban all outputs of a *Tx0* transaction if a single output were to conduct a DoS attack, adversaries would need to conduct a new *Tx0* transaction for every attack.

Concerning de-anonymization attacks by the Samourai Whirlpool tumbler, the situation changes slightly due to the fixed number of 5 participants per CoinJoin transaction. A tumbler wishing to de-anonymize a user would therefore only need to join a mix with four Sybil addresses. However, the situation is more complex due to the fact that each CoinJoin transaction requires at least one remix input and at least two premix inputs. Depending on whether the tumbler wants to attack a premix or a remix input, it would need to use one to two fresh premix inputs. Moreover, Samourai Wallet only charges fees for the initial *Tx0* transaction, but not for individual Whirlpool mix transactions. Users are therefore encouraged to remix their coins multiple times, resulting in an increased anonymity set. An attacker aiming to de-anonymize users would therefore need to keep participating in these mixes for as long as the user participates.

Concerning adversaries that do not control the tumbler, a user would still benefit from remixing their outputs for multiple CoinJoin rounds. While this is also true for Wasabi (or any CoinJoin implementation), it is even more important for Samourai Whirlpool as

the forward-looking anonymity set for every CoinJoin round is initially only 5. Similar to Wasabi Wallet, users may still be de-anonymized if they make a joint non-CoinJoin transaction with CoinJoin outputs and non-CoinJoin ouptuts. An analysis based on timing and volume may also be used to de-anonymize users.

CHAPTER 6

Discussion

This chapter discusses the results derived from Chapter 3, Chapter 4, and Chapter 5. Section 6.1 summarizes the findings and their implications, while Section 6.2 discusses the limitations of this thesis. Finally, Section 6.3 gives suggestions for potential future work.

6.1 Summary of Findings

Using the heuristics discussed in Chapter 3 we have shown that the number of CoinJoin transactions, as well as the number of mixed coins for both Wasabi Wallet and Samourai Whirlpool seem to be steadily increasing. We have also identified entities linked to criminal activities within a short distance of CoinJoin transactions, and have discussed the security of both the underlying ZeroLink framework the two wallet services themselves.

6.1.1 Wasabi Wallet

Going by the number of CoinJoin transactions, the popularity of Wasabi Wallet in general seems to increase with some phases of stagnation between jumps in the amount of transactions conducted. The amount of fresh BTC entering Wasabi and the amount of mixed BTC leaving the Wasabi ecosystem correlate with this observation, showing a steady increase albeit with some fluctuations.

A common theme throughout the entire analysis of Wasabi Wallet CoinJoins was a significant spike in fresh inputs and mixed outputs in August and September 2019 which might indicate a *one-off* mixing of a large amount of coins.

The participating entities feature a number of entities associated with addresses related to criminal actors and activities, such as the Binance exchange incident in 2019 [De19] or the allegedly North Korean Lazarus group. Exchange services, or their users, were also

active participants, with a great number of participating entities being related to various exchanges. To a lesser degree, mining pools, wallet services, gambling services, and other legitimate services were also associated with Wasabi Wallet CoinJoin transactions.

The relationship graph 4.16 between the top 10 participating entities also shows, that 5 of these entities are connected with each other across at most two hops.

6.1.2 Samourai Whirlpool

Similar to Wasabi Wallet, Samourai Whirlpool's popularity appears to be increasing. This is again true for the number of transactions as well as the amount of fresh coins entering and mixed coins leaving the Samourai Whirlpool ecosystem. The amount of fresh BTC entering Samourai Whirlpool and mixed BTC leaving Samourai Whirlpool correlate to a high degree, except for October and November 2020. During these months, the number of fresh BTC seems to drop while the number of mixed outgoing BTC has risen to its all-time high. This could, however, also be due to November 2020 not being fully present in the analyzed data set.

Of the three analyzed pools (0.01 BTC, 0.05 BTC, and 0.5 BTC), the 0.01 BTC pool dominates the total number of transactions, while the 0.5 BTC pool encompasses almost half the total output volume. Looking at fresh inputs and mixed outputs, the 0.5 BTC pool correlates very precisely with the total amount of fresh inputs/mixed outputs, including the drop in fresh inputs in October/November 2020. In contrast, the mixed outputs leaving the 0.5 BTC pool only featured a very slight drop during these two months.

The 0.05 BTC pool, however, appears to be gaining popularity, with both the amount of fresh inputs and mixed outputs increasing during October and November 2020. The values for the 0.01 BTC pool also appear to be increasing slowly but steadily.

As with Wasabi Wallet, entities associated with legitimate and criminal activities have been observed participating in Samourai Whirlpool transactions. Analyzing the entities related to criminal actors, most of the detected entities appear to be within two hops of participants of both Wasabi Wallet and Samourai Whirlpool. The same is true for 31 exchange services, and a number of other legitimate Bitcoin services. In fact, except for 8 entities related to exchange services and one related to a scam, all entities tagged by GraphSense that are within two hops of Samourai Whirlpool participants are also within two hops of Wasabi Wallet CoinJoin participants.

6.1.3 Security

The ZeroLink framework which serves as the basis for both Wasabi Wallet and Samourai Whirlpool appears to be sound in general. It is robust against coin theft and features measures that increase the cost of denial-of-service attacks, aimed at making such attacks infeasible for longer durations. Furthermore, the only truly feasible way to de-anonymize participants appears to be a rogue coordinator conducting Sybil attacks, which could

lead to retroactive suspicion. These Sybil attacks are likely easier to conduct in Samourai Whirlpool than in Wasabi Wallet, as every Samourai Whirlpool transaction is only comprised of 5 inputs whereas Wasabi Wallet typically features a much larger number of inputs.

Analyzing the timing and volume of inputs and outputs, however, could de-anonymize users if they behave in a manner different from normal users. If users are de-anonymized, the anonymity set of all remaining participants is also reduced.

Concerning Wasabi Wallet's inflation of the anonymity set through the coordinator, the effective anonymity set of participants against said coordinator may be lower than advertised, and could allow a rogue coordinator to better trace participants. While there are no indicators so far that Samourai Whirlpool engages in the same practice (perhaps due to fees not being based on the achieved anonymity set), it would in theory also be possible for the Samourai Whirlpool coordinator.

6.2 Limitations

The heuristics used to identify transactions of both Samourai Whirlpool and Wasabi Wallet feature limitations. The heuristic for identifying Wasabi Wallet CoinJoin transactions was evaluated against a ground truth established by the static coordinator address heuristic with good results. The accuracy of the heuristic may be improved by using additional restrictions. For Samourai Whirlpool, our heuristic has not detected any false positive transactions in blocks with a height lower than 570000, but we were unable to establish a ground truth to evaluate our heuristic. However, using the fact that all transactions must have a link to a genesis mix, we were able to show that all identified transactions did indeed feature such a link.

Both heuristics could be verified further by conducting CoinJoin transactions using both services on the main Bitcoin network, and check whether these transactions would be reliably identified. Concerning Samourai Whirlpool, another path for identification would be to use the link of each transaction to the genesis mix. Once known, such a heuristic could recursively check every transaction output if the subsequent transaction is also a Samourai Whirlpool CoinJoin. This heuristic should reliably identify all Samourai Whirlpool transactions, provided the genesis mixes are valid. Table 3.3 provides an exhaustive list of genesis mixes for the 0.01 BTC, 0.05 BTC, and 0.5 BTC pools. The 0.001 BTC pool was introduced in March 2021 and is not included in this thesis.

The analysis of participating entities in Section 4.2 was limited to neighbors with a maximum distance of two hops to direct participants. Furthermore, neighbors were only considered if their in degree / out degree was less than 100 in order to distinguish wallets from services. Obviously, increasing the amount of hops analyzed as well as removing the constraints concerning the in/out degrees (provided a different mechanism was in place to identify services) would result in a more complete picture of the place of Wasbai Wallet and Samourai Wallet in the larger Bitcoin ecosystem.

Additionally, Section 4.2 only ranks direct participants of CoinJoin transactions (*Level 0* entities), while *Level 1* and *Level 2* entities have not been ranked. Combining clustering techniques such as provided by GraphSense with analyzing neighboring transactions would allow *Level 1* and *Level 2* addresses to be ranked as well, providing additional details regarding participants of CoinJoin transactions.

6.3 Future Work

Other than improving the limitations described in Section 6.2, possible ideas for future work in analyzing Wasabi Wallet and Samourai Wallet can be found in the realm of security. While Chapter 5 provides a theoretical overview, we did not conduct actual attacks against either wallet service. Attempting to de-anonymize users while running a rogue coordinator could be an avenue for future research.

6.3.1 Post-Mix Security

As discussed in Section 5.3, even without considering possible Sybil attacks, the outputs of CoinJoin transactions may still be linkable in certain scenarios. The following aspects in particular may lead to de-anonymization:

- If a CoinJoin output is spent together with a non-CoinJoin output in a non-CoinJoin transaction, the CoinJoin output can be attributed to the same entity as the non-CoinJoin output due to the common input ownership heuristic (see Section 2.4).
- If a large amount of coins is being used as inputs for CoinJoin transactions in a rather short period of time, and a large amount of coins is subsequently merged together again following the CoinJoin, this large amount of merged coins is likely attributable to the same entity that deposited the coins into the CoinJoin transaction.

Moreover, if some outputs of a CoinJoin transaction are de-anonymized, the anonymity set for all remaining coins suffers as well. Wasabi Wallet, and especially Samourai Wallet, therefore offer a number of post-mix tools to improve privacy as described in Section 2.6.2 and Section 2.6.3. Of particular interest is the PayJoin feature of Wasabi Wallet, and the Stowaway, Stonewall, and Stonewallx2 features of Samourai Wallet.

Future work could analyze how well users practice post-mix security, and how well participants can be de-anonymized by analyzing the timing and volume of mixed coins.

6.3.2 WabiSabi & Wasabi Wallet Fees

Building on the ZeroLink framework, Ficsór et al. have proposed *WabiSabi: Centrally Coordinated CoinJoins with Variable Amounts*. The goal of *WabiSabi* is to replace the ZeroLink framework in Wasabi Wallet 2.0, which is currently in development. *WabiWabi*

replaces the blind signatures used in Chaumian CoinJoins with keyed-verification anonymous credentials (KVAC) schemes. The use of KVAC results in verifiable but potentially hidden transaction amounts, i.e., the coordinator of the CoinJoin will be able to verify that the outputs of a CoinJoin do not exceed the inputs, but will not learn the exact values which have been mixed. This also eliminates the need for a common minimum denomination, while improving mixing performance [dFKOS21].

As a concrete implementation for *WabiSabi* (i.e., Wasabi Wallet 2.0) has not yet been released, the heuristics proposed in Section 3.1 are unlikely to detect *WabiSabi* CoinJoin transactions. Disregarding other potential changes, the hidden transaction amounts and lack of a common minimum denomination will make the discussed heuristics obsolete. Once released, developing heuristics for *WabiSabi* will be required to further analyze the future of Wasabi Wallet.

Another interesting topic is the amount of coordinator fees collected by Wasabi Wallet since the coordinator fee collector addresses are no longer static. As Wasabi Wallet charges a fee of $0.003\% * a$, with a being the anonymity set, the anonymity set of each CoinJoin output denomination could be calculated and used to find the fee output. This could be worthwhile even with the imminent release of *WabiSabi* in order to calculate the amount of fees collected by Wasabi Wallet since January 31st, 2020.

CHAPTER 7

Conclusion

As the popularity of Bitcoin rises, so do the requirements for anonymous transactions and privacy preserving techniques. Wasabi Wallet and Samourai Wallet are two often recommended wallet services based on decentralized CoinJoins, both promising the robust and secure mixing of bitcoins.

We have improved commonly used heuristics for identifying the CoinJoin transactions of both wallets, and evaluated their accuracy as far as possible. In the case of Wasabi Wallet, the implementation was tested against an established ground truth and yielded improved results than other publicly available heuristics. The heuristic for Samourai Whirlpool found no false positive detections before the release of Samourai Wallet.

Analyzing the detected transactions shows that the amount of transactions and mixed coins for both wallet services is steadily increasing, indicating a growing user base. The analysis of participating entities, as well as their neighbors and neighbors' neighbors shows that a number of legitimate services are present in the near vicinity of CoinJoin transactions. On the other hand, as is common with services providing anonymity, they can be abused for criminal and illicit activities. As shown, entities connected to e.g. service hacks, ransomware, and extortion are also within a short distance of CoinJoin transactions conducted by both Wasabi Wallet and Samourai Wallet.

The theoretical framework underlying both wallets appears to be sound and secure against coin theft, denial-of-service, and de-anonymization. While it could be possible for an adversarial coordinator to de-anonymize users, such an action could lead to retroactive suspicion and cause the de-anonymized user to additionally mix their coins using a different coordinator or a different service. For an adversary without access to the coordinator, de-anonymization becomes increasingly difficult as the anonymity set of output addresses grows.

However, without employing proper post-mix security, it is still possible to de-anonymize participants though traffic and timing analyses. Furthermore, when the Wasabi Wallet

coordinator inflates the number of participants, the effective anonymity set users have against the coordinator suffers accordingly.

List of Figures

List of Tables

Appendix

The 9 false positive Samourai Whirlpool CoinJoin transactions additionally identified by [dF20]:

- **c481189505440ef826e11764ea5736d0c2bf6a45197a81c46f3eb4d41c9fc756**
- **8e4de55bce765d52200e52e4d7c101927f22b4e96107bac4de39f4c20775bc71**
- **68adf9f836967764b7606cbe47513dbe586d30bb59b2d0079b4da92306ecdab6**
- **37888dcecfefa974463769eff17f9fb760c875531fd2add802c394b462983526**
- **839241535e7091dc18d80cfda1d0e9cf7f5188ecb7646faec3ec59ad25869915**
- **020b0151ded22fc746338c2c61d77a11cf4880885f5e6191a4a08db82df51a10**
- **47c82113ca5ea32539cd34844f99919ad81ab18832d78b6eb4d801a4b4b77f7b**
- **f29f2387cf5222a51fe5f70e19d92d0916db69a6d48445ddac1a303573f434cf**
- **53dda92fde52dccc1c5af0fbd80389ad2864ea15df5a87d3205fddc2ec3a31bd**

Entity	Tags	Entity	Tags
110007994	btc-multiplier.fr, x-bitcoins.com, btc-multiplier.de	630681827	blockchaininfo
332183403	coinhako.com	388050037	Shapeshift
535913751	bitstamp, binance hack	52434720	happycoins.com
351536180	Shapeshift	118559766	changetip.com
53234354	btcc.com	384634133	Shapeshift
410609776	Shapeshift	334164653	Shapeshift
393627424	Shapeshift	443994777	Shapeshift
382344586	Shapeshift	1397230	bitcoin reddit, theymos
601591779	bitcoindoubler.fund	418407752	Shapeshift
10070355	bips hack	118861337	lootool.com payment 160103, lootool 20151220, lootool.com payment 160115
386796768	Shapeshift	414957832	Shapeshift
19475088	luckybit red, luckybit green, luckybit yellow, luckyb.it	285337226	vaultoro.com
50570976	agoramarket	376666079	Shapeshift
238245736	miningkings	360398906	Shapeshift
497795675	xapo	436147231	Shapeshift
410455613	Shapeshift	7443477	thepiachu
362029005	Shapeshift	276739156	btc-e.com
492026685	dragonex hack	149034608	paralelni polis donations
274408440	holytransaction.com	430797172	Shapeshift
361717366	Shapeshift	165710957	faucetbox.com
701978702	bitcoinfog, unknown ransomware	731889450	natasha, battlesrc bitcoin, coinbase, jaredkaragen, Shapeshift User 25, c01nc3, hyip monitor investspot, bittoclick, trinick, kurph, karlzt, wegfan no.2, coin academy, www.investspot.biz
337928264	Shapeshift User 93	407284804	Shapeshift
498024182	binance hack	449280499	Shapeshift
76884227	bitkonan.com	430799839	Shapeshift
416326262	Shapeshift	455062906	moonbit.co.in
317493452	zaif, zaif hack	182143020	coinjar.com
360938950	Shapeshift	449939870	Shapeshift
333456568	Shapeshift	372482749	Shapeshift
415673881	Shapeshift	12356252	eobot
488001697	binance hack	371788579	Shapeshift
360644142	Shapeshift	757187617	bitfinex
384856798	Shapeshift	348529534	Shapeshift User 30, Shapeshift User 33
753281798	okpool.top, okex	338004169	bitcoincopy.site123.me, poloniex
407439038	Shapeshift	376449085	Shapeshift
133843155	vbtc hot wallet	448046161	Shapeshift

Table A1.1: *Level 0*, *Level 1*, and *Level 2* entities participating in Wasabi Wallet CoinJoin transactions which have been assigned at least one tag in GraphSense other than *Samourai Wallet* (Part 1/4).

Entity	Tags	Entity	Tags
430736416	Shapeshift	637504149	cryptonator
372575153	Shapeshift	138063629	bitpay.com
418834441	Shapeshift	390811503	Shapeshift
382533278	Shapeshift	63813064	spectrocoin.com
386778299	Shapeshift	508051014	bittrex
177955953	bitpay.com	239324496	helixmixer
27790268	mane salon wellington, nz	403080239	yabtcl.com
417473138	Shapeshift	60849888	evolutionmarket
371967548	Shapeshift	129350440	cointrader.net, localbitcoins.com, anxpro.com, telco 214
485629147	Jiadong Li	407126516	cubits.com
383923782	Shapeshift	483725680	binance hack, binance
424694364	Shapeshift	393779867	Shapeshift
375026536	Shapeshift	391348234	Shapeshift
382691826	localbitcoins.com	21223514	fybsg.com
391621574	Shapeshift	317849903	coinspot.com.au
259096273	*149 distinct tags*	124176400	bitoex.com
415223610	Shapeshift	151488823	bitcointoyou cold wallet
417960787	Shapeshift	420205315	Shapeshift
387528155	Shapeshift	415210227	Shapeshift
414518766	Shapeshift	421068419	Shapeshift
580219218	cloudbet.com	597931459	Wasabi Fee
422153895	Shapeshift	120642452	alphabaymarket
356019657	okcoin.com	154973769	bitcointoyou hot wallet
759393443	betmoose.com	401273698	Shapeshift
662700621	kraken	384068103	Shapeshift
537704035	binance	612284741	Jiadong Li, huobi.com, Yinyin Tian Lazarus Group, huobi mining pool
803810591	therocktrading.com	414970858	Shapeshift
15384687	iosp, alexrussel1980, btc-e.com	335094971	Shapeshift
60360799	bitstamp, the_thing	418984053	Shapeshift
261728438	blocktrades.us	416881093	Shapeshift
15457089	campbx.com	270334661	bleutrade.com
397889998	Shapeshift	337288772	bitbargain.co.uk
378002289	safedice.com	413878849	Wasabi Fee
61718555	bter.com	406629379	Shapeshift
28092820	masterxchange.com	479492696	monolit, sanoshi, miner, ivsoft, bitkoin, kripta

Table A1.2: *Level 0*, *Level 1*, and *Level 2* entities participating in Wasabi Wallet CoinJoin transactions which have been assigned at least one tag in GraphSense other than *Samourai Wallet* (Part 2/4).

Entity	Tags	Entity	Tags
418766772	Shapeshift	416348890	Shapeshift
407211702	Jiadong Li	70502936	coingaming.io
48884391	slushpool	39659651	bitpay.com
354704087	Shapeshift	145208378	crypt, k.a.t, virwox.com, pirate party of austria
77982384	cryptsy.com	76513046	veracrypt
421359011	Shapeshift	345703284	Shapeshift
400016262	Shapeshift	774872354	coinmotion.com
414862006	Shapeshift	632235802	bittrex
788605564	binance hack	383683125	Shapeshift
93283494	coinkite.com	111940416	btc.com, 7pool, haobtc.com, bixin
395681314	Shapeshift	531154632	binance hack
421023285	Shapeshift	431657820	mercadobitcoin.com.br
45409647	bitsquare.io donations	62222685	bitclub network
376026893	Shapeshift	418610875	Shapeshift
732227093	binance hack	634544901	binance hack
66821636	Coinjoin Bounty	212781523	bestdoubler.eu
414065888	Shapeshift	128622392	nitrogensports.eu
360477002	Shapeshift	267439854	bitcoinbon.at - Unique sending out address
234735136	helixmixer	383830511	Shapeshift
221060919	bit-x.com	493430436	cryptopay.me
406038888	binance	658562378	binance hack
430294784	Shapeshift	411721180	Shapeshift
754722475	bitcoin.de	430776812	Shapeshift
30816384	kraken	420636831	Shapeshift
389973417	Shapeshift	38812836	Protonmail
89276054	paymium.com	99758175	bitpay.com
350928564	hitbtc.com	233468564	Sextortion Spam, Yinyin Tian Lazarus Group
433661739	Shapeshift	383685998	Shapeshift
9861443	strongcoin.com-fee	31823339	cex.io
17642138	Internet Archive	14316969	prism-break.org
89192626	bitcoinwallet.com	372938165	Shapeshift
336926848	Jiadong Li, Yinyin Tian Lazarus Group	414515119	Shapeshift
339806540	Shapeshift	335099468	Shapeshift
43289794	coin-swap.net	617877125	coinpayments.net
376097954	Shapeshift	393746465	Shapeshift
430804180	Shapeshift	388978576	Shapeshift

Table A1.3: *Level 0*, *Level 1*, and *Level 2* entities participating in Wasabi Wallet CoinJoin transactions which have been assigned at least one tag in GraphSense other than *Samourai Wallet* (Part 3/4).

Entity	Tags	Entity	Tags
520909478	binance hack	22288441	cryptosplit, cryptsy.com
341062387	Shapeshift	66987273	bitbond.com
346288946	Shapeshift	410663555	Shapeshift
421458509	Shapeshift	414947803	Shapeshift
1134488	taypeinternational	418613067	Shapeshift
3525967	*various Satoshidice tags*	344851605	Shapeshift
376312489	Shapeshift	380319978	Sextortion Spam, luno.com
420152539	Shapeshift	419890639	Shapeshift
18672110	razy	761763764	bitpanda
18584312	bitmit.net	361899532	Shapeshift
414812421	Shapeshift	71644140	satoshidice.com
18614117	silkroadmarket	769437183	binance hack
415904389	Shapeshift	420529601	Shapeshift

Table A1.4: *Level 0*, *Level 1*, and *Level 2* entities participating in Wasabi Wallet CoinJoin transactions which have been assigned at least one tag in GraphSense other than *Samourai Wallet* (Part 4/4).

Entity	Tags	Entity	Tags
788605564	binance hack	732227093	binance hack
658562378	binance hack	531154632	binance hack
634544901	binance hack	769437183	binance hack
483725680	binance hack, binance	535913751	bitstamp, binance hack
488001697	binance hack	406038888	binance
537704035	binance	177955953	bitpay.com
233468564	Sextortion Spam, Yinyin Tian Lazarus Group	612284741	Jiadong Li, huobi.com, Yinyin Tian Lazarus Group, huobi mining pool
336926848	Jiadong Li, Yinyin Tian Lazarus Group	485629147	Jiadong Li
407211702	Jiadong Li	380319978	Sextortion Spam, luno.com
408704438	Shapeshift	382890465	Shapeshift
382983234	Shapeshift	398481330	Shapeshift
414431866	Shapeshift	404230810	Shapeshift
413163170	Shapeshift	337928264	Shapeshift User 93
348529534	Shapeshift User 30, Shapeshift User 33		
449280499	Shapeshift	597931459	Wasabi Fee
129350440	cointrader.net, localbitcoins.com, anxpro.com, telco 214	479492696	monolit, sanoshi, miner, ivsoft, bitkoin, kripta
90504395	hashnest.com	493430436	cryptopay.me
761763764	bitpanda	753281798	okpool.top, okex
508051014	bittrex	111940416	btc.com, 7pool, haobtc.com, bixin
144006213	999dice.com	182143020	coinjar.com
759393443	betmoose.com	662700621	kraken
99758175	bitpay.com	332183403	coinhako.com
138063629	bitpay.com	317849903	coinspot.com.au
48884391	slushpool	39659651	bitpay.com
754722475	bitcoin.de	407126516	cubits.com
21223514	fybsg.com	774872354	coinmotion.com
617877125	coinpayments.net	45409647	bitsquare.io donations
497795675	xapo	38812836	Protonmail
601591779	bitcoindoubler.fund	261728438	blocktrades.us
276739156	btc-e.com	676195970	Twitter Hack Scam
637504149	cryptonator	803810591	therocktrading.com
31823339	cex.io	89192626	bitcoinwallet.com
630681827	blockchaininfo	757187617	bitfinex
259096273	*149 distinct entities*	274408440	holytransaction.com
701978702	bitcoinfog, unknown ransomware	338004169	bitcoincopy.site123.me, poloniex
270334661	bleutrade.com	431657820	mercadobitcoin.com.br
580219218	cloudbet.com	632235802	bittrex
731889450	natasha, battlesrc bitcoin, coinbase, jaredkaragen, Shapeshift User 25, c01nc3, hyip monitor investspot, bittoclick, trinick, kurph, karlzt, wegfan no.2, coin academy, www.investspot.biz		

Table A2: *Level 0*, *Level 1*, and *Level 2* entities participating Samourai Whirlpool transactions which have been assigned at least one tag in GraphSense other than *Samourai Wallet*.

Entity	Tags
761763764	bitpanda
233468564	Sextortion Spam, Yinyin Tian Lazarus Group
259096273	*149 distinct tags*
637504149	cryptonator
488001697	binance hack
535913751	bitstamp, binance hack
754722475	bitcoin.de
757187617	bitfinex
788605564	binance hack
601591779	bitcoindoubler.fund
274408440	holytransaction.com
338004169	bitcoincopy.site123.me, poloniex
483725680	binance hack, binance
731889450	natasha, battlesrc bitcoin, coinbase, jaredkaragen, Shapeshift User 25, c01nc3, hyip monitor investspot, bitoclick, trinick, kurph, karlzt, wegfan no.2, coin academy, www.investspot.biz
753281798	okpool.top, okex
634544901	binance hack
508051014	bittrex
662700621	kraken
497795675	xapo
617877125	coinpayments.net
612284741	Jiadong Li, huobi.com, Yinyin Tian Lazarus Group, huobi mining pool
597931459	Wasabi Fee

Table A3: *Level 0*, *Level 1*, and *Level 2* entities participating in both Samourai Whirlpool and Wasabi Wallet CoinJoin transactions which have been assigned at least one tag in GraphSense other than *Samourai Wallet*.

Bibliography

[AKR+13] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating User Privacy in Bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 34–51. Springer, 2013.

[Atl14] Kristov Atlas. Weak Privacy Guarantees for SharedCoin Mixing Service. `http://www.coinjoinsudoku.com/advisory/`, June 2014. [Online; accessed 07.03.2021].

[BKP14] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. Deanonymisation of clients in bitcoin p2p network. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 15–29, New York, NY, USA, 2014. Association for Computing Machinery.

[BNM+14] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A. Kroll, and Edward W. Felten. Mixcoin: Anonymity for Bitcoin with Accountable Mixes. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security*, pages 486–504, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[BOLL14] George Bissias, A. Pinar Ozisik, Brian N. Levine, and Marc Liberatore. Sybil-Resistant Mixing for Bitcoin. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, WPES '14, page 149–158, New York, NY, USA, 2014. Association for Computing Machinery.

[CG20] J. Crawford and Y. Guan. Knowing your Bitcoin Customer: Money Laundering in the Bitcoin Economy. In *2020 13th International Conference on Systematic Approaches to Digital Forensic Engineering (SADFE)*, pages 38–45, 2020.

[Cha83] David Chaum. Blind Signatures for Untraceable Payments. In David Chaum, Ronald L. Rivest, and Alan T. Sherman, editors, *Advances in Cryptology*, pages 199–203, Boston, MA, 1983. Springer US.

[dBHC17] Thibault de Balthasar and Julio Hernandez-Castro. An Analysis of Bitcoin Laundry Services. In *Nordic Conference on Secure IT Systems*, pages 297–312. Springer, 2017.

[De19] Nikhilesh De. Hackers Steal $40.7 Million in Bitcoin From Crypto Exchange Binance. Article, available: https://www.coindesk.com/hackers-steal-40-7-million-in-bitcoin-from-crypto-exchange-binance, May 2019. [Online; accessed 10.04.2021].

[dF19] Ádám Ficsór. Wasabi vs Samourai Various Stats. https://github.com/nopara73/WasabiVsSamourai, 2019. [Online; accessed 22.03.2021; commit f8174e607363007ddf212ca17e93608c0d2e42f2].

[dF20] Ádám Ficsór. Dumplings. Github, available: https://github.com/nopara73/Dumplings, 2020. [Online; accessed 07.03.2021; commit fdedc57458988d20341f89b215249f53e2768ba1].

[dFKOS21] Ádám Ficsór, Yuval Kogman, Lucas Ontivero, and István András Seres. WabiSabi: Centrally Coordinated CoinJoins with Variable Amounts. Cryptology ePrint Archive, Report 2021/206, 2021. https://eprint.iacr.org/2021/206.

[dFT17] Ádám Ficsór and TDevD. ZeroLink: The Bitcoin Fungibility Framework. Github, available: https://github.com/nopara73/ZeroLink, October 2017. [Online; accessed 07.03.2021; commit f5491dbc6cf7d3783ebaefcc3469111870e05745].

[GKRN18] Steven Goldfeder, Harry Kalodner, Dillon Reisman, and Arvind Narayanan. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *Proceedings on Privacy Enhancing Technologies*, 2018(4):179 – 199, 2018.

[HSRK21] Bernhard Haslhofer, Rainer Stütz, Matteo Romiti, and Ross King. GraphSense: A General-Purpose Cryptoasset Analytics Platform. *Arxiv pre-print*, 2021.

[Lau19] LaurentMT. whirlpool_stats. Gitlab repository, available: https://code.samourai.io/whirlpool/whirlpool_stats, October 2019. [Online; accessed 07.03.2021; commit: 97fb33c5d0e6843abcc644be274303a61269c506].

[LE20a] LaurentMT and ErgoBTC. An Analysis and Disclosure Regarding the Deterministic Nature of the Wasabi Wallet CoinJoin Algorithm. OXT Research, available: https://research.oxt.me/alerts/2020/08/21/Wasabi-Wallet/full, August 2020. [Online; accessed 24.03.2021].

[LE20b] LaurentMT and ErgoBTC. An update on the disclosed vulnerabilities in Wasabi Wallet. Medium, available: https://medium.com/oxt-research/an-update-on-the-disclosed-vulnerabilities-in-wasabi-wallet-4ac0e228acb9, August 2020. [Online; accessed 07.04.2021].

[LE20c] Stephan Livera and ErgoBTC. SLP179 Ergo - Unwinding Bitcoin CoinJoins: Tumblers, Wasabi, JoinMarket. Podcast, available: https://stephanlivera.com/episode/179/, June 2020. [Online; accessed 24.03.2021].

[LLP+19] Jingming Li, Nianping Li, Jinqing Peng, Haijiao Cui, and Zhibin Wu. Energy consumption of cryptocurrency mining: A study of electricity consumption in mining cryptocurrencies. *Energy*, 168:160–168, 2019.

[Max13] Gregory Maxwell. CoinJoin: Bitcoin privacy for the real world. Bitcoin Forum, available: https://bitcointalk.org/index.php?topic=279249.0, August 2013. [Online; accessed 07.03.2021].

[MBB13] M. Möser, R. Böhme, and D. Breuker. An inquiry into money laundering tools in the Bitcoin ecosystem. In *2013 APWG eCrime Researchers Summit*, pages 1–14, 2013.

[MPJ+13] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 127–140, 2013.

[Nak08] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Report, Manubot, 2008.

[NBF+16] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[Noe15] Shen Noether. Ring Signature Confidential Transactions for Monero. Cryptology ePrint Archive, Report 2015/1098, 2015. https://eprint.iacr.org/2015/1098.

[Pec12] M. E. Peck. Bitcoin: The Cryptoanarchists' Answer to Cash. *IEEE Spectrum*, 49(6):50–56, 2012.

[PK01] Andreas Pfitzmann and Marit Köhntopp. *Anonymity, Unobservability, and Pseudonymity — A Proposal for Terminology*, pages 1–9. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[PRHC19] Masarah Paquet-Clouston, Matteo Romiti, Bernhard Haslhofer, and Thomas Charvat. Spams meet Cryptocurrencies: Sextortion in the Bitcoin Ecosystem. *CoRR*, abs/1908.01051, 2019.

[QC19] Mikerah Quintyne-Collins. Short Paper: Towards Characterizing Sybil Attacks in Cryptocurrency Mixers. *IACR Cryptol. ePrint Arch.*, 2019:1111, 2019.

[RH13] Fergal Reid and Martin Harrigan. An Analysis of Anonymity in the Bitcoin System. In *Security and Privacy in Social Networks*, pages 197–223. Springer, 2013.

[RKB15] Chris Richter, Sascha Kraus, and Ricarda B. Bouncken. Virtual currencies like Bitcoin as a paradigm shift in the field of transactions. *International Business & Economics Research Journal (IBER)*, 14(4):575–586, 2015.

[RMSK14] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. Coinshuffle: Practical Decentralized Coin Mixing for Bitcoin. In *European Symposium on Research in Computer Security*, pages 345–364. Springer, 2014.

[RMSK17] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. P2P Mixing and Unlinkable Bitcoin Transactions. In *NDSS*, pages 1–15, 2017.

[Sam19] Samourai Wallet. Diving head first into Whirlpool Anonymity Sets. Medium, available: `https://medium.com/samourai-wallet/diving-head-first-into-whirlpool-anonymity-sets-4156a54b0bc7`, October 2019. [Online; accessed 07.03.2021].

[Sam21] Samourai Wallet. Changes to Whirlpool mixing fees effective March 2021. Medium, available: `https://medium.com/@SamouraiWallet/changes-to-whirlpool-mixing-fees-effective-march-2021-30c8a2a59aed`, March 2021. [Online; accessed 07.03.2021].

[SCG+14] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized Anonymous Payments from Bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE, 2014.

[Tru18] Jon Truby. Decarbonizing Bitcoin: Law and policy choices for reducing the energy consumption of Blockchain technologies and digital currencies. *Energy Research & Social Science*, 44:399–410, 2018.

[U.S20] U.S Department of the Treasury. Treasury Sanctions Individuals Laundering Cryptocurrency for Lazarus Group. Press Release, available: `https://home.treasury.gov/news/press-releases/sm924`, March 2020. [Online; accessed 10.04.2021].

[VWOvD18] Rolf Van Wegberg, Jan-Jaap Oerlemans, and Oskar van Deventer. Bitcoin Money Laundering: Mixed Results? *Journal of Financial Crime*, 2018.

[Was19] Wasabi Documentation. `https://github.com/zkSNACKs/WasabiDoc/tree/master/docs`, July 2019. [Online; accessed 05.04.2021; commit: eeb8ef5e798f11f73981a0d0d17ae828ca8d75d1].

[WM14] Michael E. Whitman and Herbert J. Mattord. *Principles of Information Security, 4th edition.* Cengage Learning, 11 2014.

[zkS18] zkSNACKs. Wasabi Wallet. `https://github.com/zkSNACKs/WalletWasabi`, October 2018. [Online; accessed 24.03.2021; commit: 2cdbb513c6dd1b222e8b31e6369c858f332f4dea].