UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

SEMINAR FOR THE "COMPUTER FORENSICS" COURSE
[2018/19]

Bitcoin Transaction Forensics

Juraj Juričić

Zagreb, January 2019.

Contents

1. Introduction

Bitcoin is a decentralised digital currency invented by an unknown entity using the name *Satoshi Nakamoto* released in 2009. Bitcoin has no central repository and no single administrator. Being a cryptocurrency, it relies on cryptographic protocols to generate the currency, validate transactions and prove ownership. Bitcoin utilises a public peer-to-peer distributed ledger, called *blockchain*, to store the transactions and prevent double-spending [12].

Since its inception, Bitcoin has been used for various usages, some of which are illicit, illegal, or the users simply don't want to leave traces of the transactions behind. The mechanisms Bitcoin uses seem to provide ways to conceal the traces, making it an obvious tool in such transactions. However, Bitcoin is often mistakenly thought of as an anonymous and secure platform, whose transactions are untraceable. This article will show that Bitcoin as a platform is not completely anonymous and the transactions are not completely untraceable.

The goal of Bitcoin transaction forensics is to determine a couple of things. First, looking at an arbitrary transaction, one would like to know which amount was actually meant to be transacted, and which was just the *change*[1]. Furthermore, one would like to figure out if different addresses are related (e.g. they belong to the same user) [2]. Finally, the goal of transaction forensics might be to determine who is the real-world user owning an address (that is, to determine the user-address mapping). This last task is much more complex because it requires off-network information.

Since the forensics of bitcoin transactions is a complex task and still a very much researched field, this article provides just an overview of the subject. It provides an overview of the methods used to determine the flow of value (bitcoins) and some basic tools or methods that one can use to further anonymise transactions.

[1] In Bitcoin, a single coin cannot be spent partially – to send just a portion of a coin, one must make two new coins, one of which they send back to themselves. This coin sent back to self is called *change*.

[2] In Bitcoin, an address is the only identifier of a user.

2. Elements of the Bitcoin Platform

2.1. Bitcoin Adresses

In Bitcoin, users are identified with their public cryptographic key-pair. A user uses their private key to authorise transactions and prove ownership of value. In Bitcoin, the only user identifier used is their *address*. A Bitcoin address is an identifier of 26-35 alphanumeric characters that represents a possible destination for a payment [2]. Therefore, Bitcoin is not anonymous but pseudonymous – a user is associated with a pseudonym (an address in this case).

The simplest and most commonly used type of address is *P2PKH* (Pay-to-PubkeyHash) address, which is just a cryptographic hash of the recipient's public key. There are other types of bitcoin addresses that won't be considered in this article.

In an average transaction, a user sends some value of bitcoins from one or multiple addresses (these addresses are called input addresses) to one or multiple output addresses (recipient's addresses).

Every user may have many of addresses. Generating an address is free and as simple as generating a new key-pair. It is recommended to use an address only once (and as we'll see, that is the main way of *anonymising* user's bitcoins). Specialised software called *wallet* can be used for managing multiple addresses of a single user.

2.2. Blockchain and Transactions

Every Bitcoin transaction ever finalised is securely recorded in the public Bitcoin blockchain. A blockchain is a growing list of records, called *blocks*, which are linked with hash pointers. Each block contains a cryptographic hash of the previous block (inside the hash pointer), a timestamp, and transaction data [14]. The blocks, once created, cannot be modified without modifying all the successive blocks, making this method secure for storing the transactions because once recorded transactions cannot be easily modified.

To spend some value (and perform a transaction), a user has to prove ownership

by signing the transaction with the private key that corresponds to the public key that hashes to the address containing the input bitcoins. These proofs of ownership are called **witnesses**.

When a transaction has multiple inputs, there will be multiple witnesses – one is required for each of the inputs.

2.3. Transaction Outputs

The result of each transaction is one or more outputs. Outputs can be thought of as individual *coins*[1]. An address does not, in fact, have any balance value assigned to it – an address' balance is the sum of all unspent outputs ever sent to that address.

An output is identified by transaction id and its position in that transaction.

Outputs are used up in their entirety – one cannot spend half of an output, or any other arbitrary portion of an output. If a user wants to send a smaller amount of value in a transaction, they can create multiple outputs, one of which will be the *change*.

Table 2.1 provides a real-world[2] example of a transaction with multiple inputs and outputs. The two inputs consumed have a value of ≈0.026 and ≈0.086, while the two generated outputs have a value of ≈0.021 and ≈0.091.

Type	**ID**	**Value**
Input	582ec73cf11cf...ec068dd3e833fbb5d8a8**:1**	0.0259349
	7b42073588708...dc255342652766cfe0ff**:0**	0.0855901
Output	c138bb3f6d306...7f0b6791dfcc5c5f5425**:0**	0.02054611
	c138bb3f6d306...7f0b6791dfcc5c5f5425**:1**	0.0908708

Table 2.1: Inputs and outputs of a sample transaction [6]

2.4. Wallets

Although not an integral part of the Bitcoin platform, Bitcoin wallets play a very important role in transacting. A Bitcoin wallet is essentially a collection of a single user's private keys. Most Bitcoin wallets are client software used to manage these keys and to make transactions on the Bitcoin network. A wallet keeps track of addresses associated with the private keys, parses and creates transaction data, and keeps track of all available outputs (*coins*).

[1]Throughout this article, term *coin* will be used to signify a single *transaction output*.

[2]Transaction ID: c138bb3f6d3060d6ce7bd1733cdc377b96caf485c5c77f0b6791dfcc5c5f5425

3. Bitcoin Transactions

This chapter will provide an outlook on different variations of Bitcoin transactions. In all following examples, transactions' inputs and outputs will be shown as addresses, in order to declutter the format. In reality, inputs and outputs are actually not addresses but transaction outputs (as explained in 2.3).

The goal of this chapter is to uncover some truth about these transactions and addresses.

3.1. Change

Every Bitcoin transaction creates outputs, also called coins. A single coin cannot be spent partially (i.e. to spend a coin, one must completely use up the coin). A good analogy is having a cash bill: if one has a 50€ bill, they can't split it into two equal parts that are worth 25€ each – they have to give away the 50€ bill, and if they want to buy something worth 20€, the seller will return the change – 30€.

A similar concept applies to Bitcoin transactions: when a user wants to send 1 BTC to someone else, but has only a 3 BTC output (coin) available, they create a transaction with one input (3 BTC) and 2 outputs (1 BTC to the recipient and 2 BTC back to themselves).

3.2. Single-input-single-output transactions

The most common type of Bitcoin transaction is sending money from one user to another.

The most basic of these transactions takes one input and creates one output. An example of this transaction is shown in table 3.1. In this transaction, a user sent about 0.3 bitcoins from address 1Jbtmu3h... to address 1WHz729X...[1]. The output's value is almost the same as the input's value (i.e. there is no change), and this transaction

[1]The difference between inputs' and outputs' values is the miner's fee, which is not important for our analysis

represents either a payment to someone else's address or transfer of value from the user's one address to another.

This is the case when there is no change address – the user had a coin of size equal to the amount they wanted to send, so there is no need for a change.

Type	**Address**	**Value**
Input	1Jbtmu3hJ3ya3mUTHhBMzYPWFLkK7VRXxH	0.30133789
Output	1WHz729XYCjUi2LRxDyrqfmEhADQeV8ww	0.30023789

Table 3.1: Sample transaction [7]

3.3. Single-input-multiple-output transactions with change sent back to sending address

Following the previous example, the most common type of transactions that send money from one user to another has multiple (often two) outputs. An example of a transaction with only one input and two outputs is shown in table 3.2.

In this transaction, a user sent about 45 bitcoins from address 1NW56pjn... to address 1BTCorgH...[2] (≈3 BTC) and back to 1NW56pjn... (≈42 BTC).

It is obvious that the user sent 3 BTC to the Bitcoin Foundation, keeping 42 BTC to themselves. This happens when a user wants to spend a coin that is larger than the amount they want to send. The remaining value is called *change* and is sent back to the user.

In this transaction, the user sent the change to the same address, making it obvious what was the intended transactional value, and what was just the change.

As we will see in the coming chapter, this is most likely a donation user made to the Bitcoin Foundation.

Type	**ID**	**Value**
Input	1NW56pjnaddNRzaFnJPd9s2kg1QHbP5pT3	44.9999
Output	1BTCorgHwCg6u2YSAWKgS17qUad6kHmtQW	3
	1NW56pjnaddNRzaFnJPd9s2kg1QHbP5pT3	41.9998

Table 3.2: Sample transaction [8]

[2]This is a known address of Bitcoin Foundation

3.4. Single input multiple output transaction with a change address

The transactional approach in the previous section allows for easy tracking of the value flow and is usually discouraged within the community. It is generally recommended for **a single address to be used only once**. To implement this, when sending a change, wallet software usually sends the change output to a newly minted address (but still belonging to the same user). Since an address can easily be generated (by creating a new key-pair), this process is done seamlessly.

Determining the actual transacted value in a transaction that uses a change address is more complex than in the previous example. Table 3.3 demonstrates one such transaction.

In this transaction, we cannot instantaneously determine which of the outputs are change. There is no property recorded in the blockchain to do this. Both the first and second output could be the intended recipient and the sender's change address. Furthermore, there is also a possibility that both of these addresses are actual recipients (meaning that there is no change address).

However, there is a heuristic used to determine what amount was meant to be sent and what is the change. In this transaction, one of the inputs has a *round* value of **0.04**, while the other has an arbitrary value (**0.03835675**). When this is the case, it is most likely that the arbitrary value is the change and the *round* value is the intended transaction amount. It is important to remember that the users of Bitcoin are people, and people "like" *round* (or *nice*) numbers.

This heuristic can be applied when the amount sent is determined by the user (usually for donations or simple money transfer). However, when bitcoins are used for paying a product, the value usually looks more random (or, more often, it is a fiat[3] value converted to Bitcoin value). A similar approach can be used then, but using a fiat value as a value that is being *rounded*.

Type	**ID**	**Value**
Input	1FH8Krv8khfzbxoMqJ9SDshCb5VDeWPDJ4	0.0788
Output	17Jk1XSB87AEWCeVoBwoPEykyEYYYMZaqU	0.04
	13Bz7AmarC4fsUCePiojERbtCTNsKDoCjs	0.03835675

Table 3.3: Sample transaction [9]

[3]Fiat is a real-world currency, like EUR or USD

3.5. Other kinds of transactions

3.5.1. Multiple input single output

This is the kind of transaction that is most commonly used to combine multiple coins into one, where both the sender and the receiver is the same person (but not necessarily the same address). Another use case might be sending the exact amount to a single entity from a single user's multiple addresses. Finally, it might mean that multiple users combined their payment to a single entity, but that is usually less common.

An example of this transaction is shown in table 3.4.

Type	**ID**	**Value**
Input	142xkmNCHs8KHcdd7f3DiFWijgc9vskJz6	0.04880665
	1PQN3DytkkfuqHyvYc1snFfPjLfUzfhyUT	0.11899419
Output	1NS9KjA9NUrXcfd637kdjStP1B2vZhHWx4	0.16716317

Table 3.4: Sample transaction [10]

3.5.2. Multiple input multiple output

This is oftentimes the most complex kind of transaction to uncover. Uses cases are several, but the most interesting one is *covering the tracks*. Multiple unrelated users may combine their inputs in a single transaction and generate multiple outputs, thus making it harder to determine where someone's coin went.

Since these transactions are often big, no example will be provided here, but readers are encouraged to explore them on their own in the Bitcoin public blockchain. A good example may be [11].

This idea can be improved upon using so-called *Bitcoin tumblers*.

4. Bitcoin Adresses

This chapter will overview different addresses based on what purpose they serve. It is important to understand that Bitcoin addresses are recommended to be used as *burners*, meaning that each should be used only once in order to preserve anonymity and ambiguity.

True burner addresses are addresses that have only one incoming and one outgoing transaction, where the outgoing transaction always uses up all available input.

4.1. Donation address

This address kind was already shown in [8] – the receiving address 1BTCorgH... is a known address of Bitcoin Foundation.

The simplest way to implement a donation address is for an organisation to use a single address where all willing donators can donate to. Therefore, these addresses usually have many incoming transactions and are therefore not *burner*. Once in a while, all the value will be transferred to another address, often a burner one. In this way, the single non-burner address is linked to multiple burner addresses, but it takes additional steps to link them.

A good example of this address is the aforementioned Bitcoin Foundation donation address – one can explore the transactions associated with this address at [1].

4.2. Payment address

When a user wants to use bitcoins to buy a product or service, the seller usually uses some kind of payment gateway[3]. The role of a payment gateway is accepting a cryptocurrency from buyers and usually giving a fiat currency to sellers. When a user places an order, the payment gateway generates a new single-use address.

All the transactions outputting to that address are associated with that purchase, and once the funds exceed the required amount, the value is usually transferred to one of the payment gateway's other addresses. Most of the payment addresses have a single

incoming and a single outgoing transaction. Unfortunately, that is also true for most burner addresses, which makes payment addresses harder to recognise.

4.3. Personal address

Personal adresses are the regular addresses most users use. They are most often generated by a wallet software, and kept track of through it. However, people being people, they are most often not completely burner – a user will more often than not reuse a single address, usually because they are not aware of the possibility of using a burner address.

4.3.1. Change address

A special subtype of personal addresses are change addresses, already mentioned in 3.1 and 3.4 – they are addresses used to keep the transactional change. However, since they are almost exclusively managed by wallet software, they will be *true burner*, meaning they will always have a single incoming transaction and a single outgoing transaction (just like payment addresses in 4.2).

Generally, addresses that are completely managed by software are more likely to be *true burner* addresses since the software is more predictable and consistent than human behaviour. Of course, humans can also attempt to always use burner addresses for manual transacting.

5. Increasing anonymity

5.1. Burner addresses

As mentioned multiple times, using an address only once, although not enough, is very important in order to prevent associating users with transactions. By using an address only once, advanced methods must be used in order to determine which addresses are related.

5.2. Bitcoin Tumblers

Looking back to 3.5.2, multiple input multiple output transactions were a good way to obfuscate value flow. However, the inputs and outputs are still directly linked one to another, and a careful examiner might be able to figure out which output is connected to which input.

By taking this idea a step further, one gets to the idea of *tumblers*. Bitcoin tumbler is a service that mixes potentially identifiable bitcoins with others, so as to obscure the trail back to the fund's original source [13]. A tumbler takes multiple users' bitcoins and mixes them with other users' bitcoins, giving the user a *fresh* batch of bitcoins that cannot be traced back to them. Tumblers also take an arbitrary fee, usually between 1% and 3%. Figure 5.1 shows how a tumbler combines different bitcoins to forward some amount to a user.

Tumblers are, to some degree, an exchange of Bitcoins between users in an anonymous manner so someone ends up getting the (almost) same amount but composed of bitcoins from several different sources [5].

5.2.1. Risks involved

Legality Although tumblers themselves are legal, bitcoin tumblers are often associated with illegal activities. Mixing can be used for money laundering by mixing illegally obtained funs. A user that uses a tumbler to obfuscate the flow of legally obtained funds might get some of the illegally obtained funds in the process.

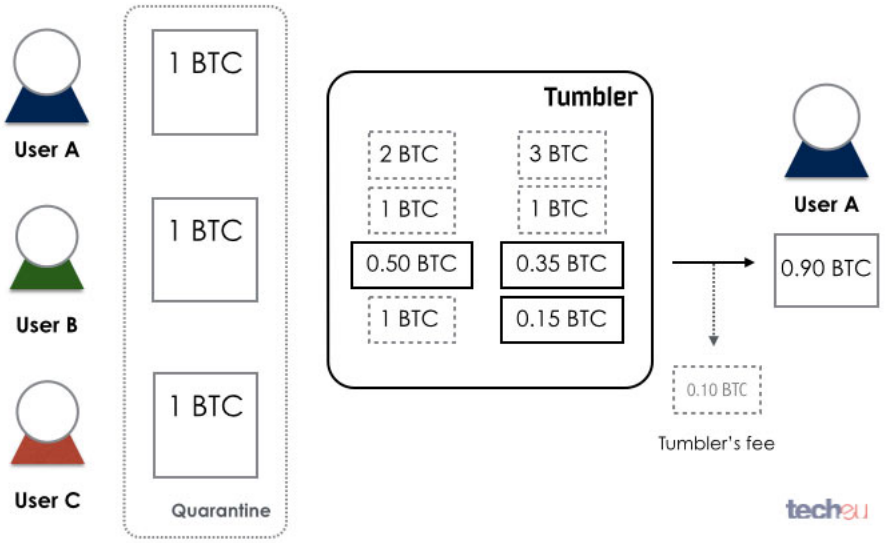

Figure 5.1: Bitcoin tumbler [5]

Stolen funds Another risk is that of having bitcoins stolen. Since a user is essentially sending their funds to someone else's address, they have to make sure they trust the owner of tumbler. The Bitcoin platform does not provide any assurance that the requested bitcoins will actually be delivered.

Service transparency In order to be effective, tumblers must not keep track of the requested *tumbles*. Most of the tumblers claim not to keep any logs, but there are logs on many different levels, from the user's ISP (which could prove a user used the tumbler, but not how) to the user's machine (in case of a seizure) to actual tumbler logs (if the tumbler service does not actually delete request history).

6. Conclusion

Bitcoin transactions are completely public and transparent by design. Bitcoin as a platform does not make users anonymous, but is rather pseudonymous. Addresses are meant to be used only once, and doing so ensures the user's privacy on at least some levels.

The way an address is used can help determine what it is used for. Various transactions offer different levels of privacy and knowing when to use which can help users obfuscate their tracks. Likewise, knowing different kinds of transaction usage can help analysts determine which addresses might be related. Finally, there are various tools that can help explore the flow of bitcoins on Bitcoin's blockchain, one of which is blockchain.com.

For further reading, the reader is directed to [4, 15].

7. Bibliography

[1] Bitcoin address 1BTCorgHwCg6u2YSAWKgS17qUad6kHmtQW. https://bit.ly/2C53MRW. Accessed: January 6, 2019.

[2] Bitcoin Wiki: Address. https://en.bitcoin.it/wiki/Address, . Accessed: January 7, 2019.

[3] BitPay documentation. https://bitpay.com/docs/, . Accessed: January 6, 2019.

[4] Sciencemag: Why criminals can't hide behind Bitcoin. https://www.sciencemag.org/news/2016/03/why-criminals-cant-hide-behind-bitcoin. Accessed: January 7, 2019.

[5] Tech.eu: Bitcoin Wallets. https://tech.eu/features/1283/bitcoin-wallets/. Accessed: January 7, 2019.

[6] Blockchain Transaction c138bb3f6d3060d6ce7bd1733cdc377b96caf485c5c77f0b6791dfcc5c5f5425. https://bit.ly/2QrmwQG, . Accessed: January 6, 2019.

[7] Blockchain Transaction 534a9a04b4c9161f73013ed6a01fa000071f4e5d62ddfae9d5b45bed1666a7cb. https://bit.ly/2Fg54xk, . Accessed: January 6, 2019.

[8] Blockchain Transaction b345c51064074f5d6c0e11187326567765eb4fa3236ca1be440ccb5745775238. https://bit.ly/2RBycF5, . Accessed: January 6, 2019.

[9] Blockchain Transaction fae25185116d2274abb4087be9257d5a991dcc3cbdc0d66dddadee5bbf772e27. https://bit.ly/2ABoTLx, . Accessed: January 6, 2019.

[10] Blockchain Transaction 4f72770fb51984d33f4633e36828a31f31058964a980abc7d3f0c79538174efa. `https://bit.ly/2GYXDwi`, . Accessed: January 6, 2019.

[11] Blockchain Transaction d160cea33cb4653928b828c461683f12cab7707e18d9ef9ea3c46c2f389948eb. `https://bit.ly/2Fd96qj`, . Accessed: January 6, 2019.

[12] Jennifer Shasky Calvery. Statement of Jennifer Shasky Calvery, Director Financial Crimes Enforcement Network United States Department of the Treasury. *Vienna, Virginia, United States: Financial Crimes Enforcement Network*, 2013.

[13] Usman Chohan. The Cryptocurrency Tumblers: Risks, Legality and Oversight. 2017.

[14] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, i Steven Goldfeder. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[15] Fergal Reid i Martin Harrigan. An analysis of anonymity in the bitcoin system. U *Security and privacy in social networks*, stranice 197–223. Springer, 2013.