HOCHSCHULE MITTWEIDA
University of Applied Sciences

BACHELORARBEIT

Herr
Lukas Schöne

Anonymität im Bitcoin Netzwerk: Eine umfassende Untersuchung der Entitäts- und Transaktionsnachverfolgungstechniken auf der Bitcoin Blockchain

Mittweida, Februar 2024

Fakultät **Angewandte Computer- und Biowissenschaften**

BACHELORARBEIT

Anonymität im Bitcoin Netzwerk: Eine umfassende Untersuchung der Entitäts- und Transaktionsnachverfolgungstechniken auf der Bitcoin Blockchain

Autor:
Lukas Schöne

Studiengang:
Angewandte Informatik

Seminargruppe:
IF19WI2B

Erstprüfer:
Prof. Dr.-Ing. Andreas Ittner

Zweitprüfer:
Mario Oettler, Dipl.-Volkswirt

Einreichung:
Mittweida, 23.02.2024

Verteidigung/Bewertung:
Mittweida, 2024

Faculty of **Applied Computer Sciences and Biosciences**

BACHELOR THESIS

Anonymity on the Bitcoin Network: A comprehensive examination of entity and transaction tracking techniques on the Bitcoin blockchain.

Author:
Lukas Schöne

Course of Study:
Applied Computer Science

Seminar Group:
IF19WI2B

First Examiner:
Prof. Dr.-Ing. Andreas Ittner

Second Examiner:
Mario Oettler, Dipl.-Volkswirt

Submission:
Mittweida, 23.02.2024

Defense/Evaluation:
Mittweida, 2024

Bibliografische Beschreibung

Schöne, Lukas:
Anonymität im Bitcoin Netzwerk: Eine umfassende Untersuchung der Entitäts- und Transaktionsnachverfolgungstechniken auf der Bitcoin Blockchain. – 2024. – 62 S.
Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2024.

Referat

Das Ziel dieser Bachelorarbeit ist die Darstellung von Techniken zur Verfolgung von Transaktionen und Entitäten auf der Bitcoin Blockchain. Die zugrundeliegende Forschungsfrage lautet wie folgt: Welche Aussagekraft und Grenzen haben Heuristiken und Analysemethoden zur Nachverfolgung von Bitcointransaktionen und wie wirken sie sich auf deren Anonymität aus?

Zur Beantwortung der Forschungsfrage wird eine Literaturrecherche durchgeführt. Die Funktionsweise der Methoden wird dargestellt, Stärken und Schwächen der Methoden werden aufgezeigt und gegebenenfalls Verbesserungsvorschläge gemacht.

Es zeigt sich, dass die Anonymität des Bitcoin-Systems aus verschiedenen Richtungen eingeschränkt werden kann. Die vorgestellten Analysemethoden haben jedoch ihre Grenzen und können die Anonymität des Bitcoinsystems nicht endgültig überwinden.

Letztendlich bleibt festzuhalten, dass die Anonymität eines Nutzers von seiner Bitcoin-Nutzung und der Nutzung seines Umfelds abhängt. Ein bewusster, anonymer Umgang mit der Kryptowährung ist die Grundlage für eine solide Privatsphäre.

Inhaltsverzeichnis

Inhaltsverzeichnis I

Abbildungs- und Tabellenverzeichnis IV

1 Einleitung 1

2 Technische Grundlagen 3
2.1 Grundlagen von Bitcoin . . . 3
2.1.1 Adressen und Adresstypen . . . 3
2.1.2 Transaktionen . . . 4
2.1.3 Blöcke, Blockchain und Mining . . . 5
2.1.4 Bitcoin Peer-to-Peer Netzwerk . . . 6
2.2 Anonymität und Pseudonymität . . . 7
2.3 Graphentheoretische Konzepte . . . 8
2.3.1 Graphenstruktur . . . 8
2.3.2 Spezielle Elemente in Graphen . . . 8

3 Identifizierung und Zuordnung von Transaktionen auf der Blockchain 10
3.1 Ähnlichkeit der Transaktionszeiten . . . 10
3.2 Ähnlichkeit der Transaktionswerte . . . 11

4 Heuristiken und Methoden auf Basis der Transaktionseigenschaften 12
4.1 Multi-Input Heuristik . . . 12
4.1.1 Allgemeine Multi-Input Heuristik . . . 12
4.1.2 Multi-Input Heuristik Fehlervermeidung bei CoinJoin und PayJoin-Transaktionen 15
4.1.3 Erweiterung der Multi-Input Heuristik um Community-Detection . . . 18
4.2 Wechselgeldheuristik . . . 20
4.2.1 Allgemeine Wechselgeldheuristik . . . 20
4.2.2 Universell anwendbare Wechselgeldhinweise . . . 22
4.2.3 Gleichartige Weiterverwendung von Transaktionsoutputs - Fingerprinting . . . 28
4.2.4 Wechselgelderkennung auf Basis späterer Adresswiederverwendungen . . . 33
4.3 Adressclustering bei speziellen Transaktionen . . . 34
4.3.1 Coinbase-Transaktionen . . . 34
4.3.2 Single-Output Transaktionen . . . 34
4.4 Aktive Methoden und Angriffe . . . 35
4.4.1 Informationsgewinn durch Handel . . . 35
4.4.2 Erzwungene Wiederverwendung von Adressen . . . 35

5 Statistische Analyse und Qualitative Bewertung von Clusteringmechanismen 37
5.1 Datensatzerzeugung . . . 37
5.2 Kennzahlen und Vergleichbarkeit . . . 38
5.2.1 Anwendbarkeit, Effizienz und Adressreduktionsrate . . . 38
5.2.2 Genauigkeitsmaße . . . 39
5.2.3 NMI und aNMI . . . 39

5.3 Qualität der Multi-Input Heuristik . . . 40
5.3.1 Anwendbarkeit und Adressreduktionsrate . . . 40
5.3.2 Genauigkeit der Heuristik . . . 40
5.3.3 NMI und aNMI . . . 41
5.4 Qualität der Wechselgeldheuristik . . . 42
5.4.1 Anwendbarkeit und Adressreduktionsrate . . . 42
5.4.2 Genauigkeit der Heuristik und ihrer Hinweise . . . 43
5.4.3 NMI und aNMI . . . 45

6 Methoden basierend auf Informationen abseits des Bitcoin Netzes 46
6.1 Frei verfügbare Informationen . . . 46
6.1.1 Motivation für die Veröffentlichung . . . 46
6.1.2 Informationsgewinn durch Scraping und Crawling . . . 46
6.1.3 Erkennung einer Bitcoinadresse . . . 47
6.2 Informationsgewinn durch Bitcoinforks . . . 47
6.2.1 Der Fork von Bitcoin-Cash . . . 48
6.2.2 Chainübergreifendes Adressclustering . . . 48
6.2.3 Mächtigkeit der Chainübergreifenden Clusteringmethodik . . . 49
6.3 Verknüpfung von Transaktionen mit IP-Adressen . . . 50
6.3.1 Technische Realisierbarkeit . . . 50
6.3.2 Übertragung durch Tornetzwerk . . . 51
6.3.3 Ausnahmen und Fehler . . . 52

7 Analyse von Finanzströmen 53
7.1 Darstellung von Transaktionen in Graphen . . . 53
7.1.1 Transaktionsgraphen . . . 53
7.1.2 Adressgraphen . . . 54
7.1.3 Nutzergraphen . . . 54
7.2 Breitensuche . . . 55
7.3 Taintanalyse . . . 56
7.3.1 Poison Methode . . . 56
7.3.2 Haircut Methode . . . 57
7.3.3 Taintmethoden auf Basis der Input Reihenfolge . . . 58
7.3.4 Wertbasierte Taint Methode . . . 59

8 Diskussion und Ausblick 60
8.1 Darstellung der Methoden . . . 60
8.2 Qualität der Methoden . . . 60
8.3 Einfluss auf die Anonymität . . . 60
8.4 Ausblick . . . 61

9 Fazit 62

Anhang 63

A Dekodierte Bitcoin Transaktion 63

B Statistische Untersuchung der Wechselgeldhinweise 64

Glossar **65**

Literaturverzeichnis **67**

Eidesstattliche Erklärung **73**

Abbildungs- und Tabellenverzeichnis

Abbildungsverzeichnis

3.1 Korrelation von Transaktionswerten 11

4.1 Graphische Darstellung der Multi-Input Heuristik. 12
4.2 Zusammenführung von Adressclustern. 13
4.3 Bruch der Multi-Input Heuristik durch CoinJoin Transaktion 15
4.4 Graphische Darstellung des PayJoin Konzepts. 17
4.5 Markierung einer Community im Entitätsgraphen entsprechend der Community-Detection Heuristik 18
4.6 Graphische Darstellung der allgemeinen Wechselgeldheuristik. 20
4.7 Peeling-Chain Phänomen 21
4.8 Wechselgelderkennung bei Transaktionen zu anderen Adressclustern 23
4.9 Optimale Wechselgeldheuristik 24
4.10 Mehrere Wechselgeldkandidaten bei Optimaler Wechselgeldheuristik. 25
4.11 Wechselgelderkennung anhand unterschiedlicher Adresstypen. 26
4.12 Wechselgelderkennung durch gleichartige Transaktionen. 28
4.13 Wechselgelderkennung durch späteren, gemeinsamen Input. 33

5.1 Gegenüberstellung von Multi-Input Heuristik und darauf aufbauender Community-Detection Heuristik [21, S. 174 f.] 41
5.2 Veränderung der Aussagekraft von Wechselgeldhinweisen [31, S. 10] 45

6.1 Darstellung einer Blockchain nach einem Hardfork 48
6.2 Visualisierung des chainübergreifenden Clusterings 49

7.1 Graphische Darstellung eines Transaktionsgraphen 53
7.2 Graphische Darstellung eines Adressgraphen 54
7.3 Überführung eines Adressgraphen in einen Nutzergraph 54
7.4 Ebenendarstellung des BFS-Suchalgorithmus 55
7.5 Verbreitung des Taintscores bei der Poison Methode 57
7.6 Verbreitung des Taintscores bei der Haircut Methode 57
7.7 Verbreitung des Taintscores bei FIFO und LIFO 58
7.8 Verbreitung des Taintscores bei der TIHO Methode 59

Tabellenverzeichnis

5.1 Ausschnitt aus Anhang B bezüglich der Anwendbarkeit universeller Wechselgeldhinweise 42
5.2 Ausschnitt aus Anhang B bzgl. der Adressreduktionsrate der universellen Wechselgeldhinweise 43

5.3 Ausschnitt aus Anhang B bezüglich der Genauigkeit der universellen Wechselgeldhinweise 44
5.4 Ausschnitt aus Anhang B bezüglich der Genauigkeit der Fingerprinting Wechselgeldhinweise 44

Abkürzungsverzeichnis

DER . . . . . Distinguished Encoding Rules

ECDSA . . . . . Elliptic Curve Digital Signature Algorithm

P2WPKH . . . . . Pay to Witness Public Key Hash

P2WSH . . . . . Pay to Witness Script Hash

SegWit . . . . . Segregated Witness

1 Einleitung

Die kürzliche Genehmigung von börsengehandelten Bitcoin-Fonds durch die US-Börsenaufsicht SEC stellt die Kryptowährung einmal mehr in den Mittelpunkt der gesellschaftlichen Aufmerksamkeit. [1] Während Investoren eine neue Kapitalanlage wittern, stellt sich für andere die Frage: Was für ein System sie damit eigentlich unterstützen. In der Wahrnehmung vieler Menschen ist Bitcoin noch immer eine mysteriöse digitale Währung, mit der unbeobachtet gehandelt werden kann. Doch inwiefern trifft diese Wahrnehmung zu? Ist Bitcoin tatsächlich so intransparent und anonym, wie oft angenommen wird, oder gibt es Möglichkeiten, seine Nutzer zu verfolgen und zu identifizieren?

Bitcoin als digitale Währung unterliegt keiner zentralen Kontrolle und entzieht sich daher staatlichen Interventionsmaßnamen. Illegitime Aktivitäten wie Geldwäsche oder Terrorfinanzierung werden daher häufig über Kryptowährungen wie Bitcoin realisiert. Um diese Aktivitäten erkennen, verfolgen und verhindern zu können, werden Nachverfolgungstechniken benötigt. In der Vergangenheit wurden daher Forschungsprojekte ins Leben gerufen, um Methoden zur Identifizierung und Überwachung verdächtiger Transaktionen in Kryptowährungen zu entwickeln. In internationaler Zusammenarbeit wurden bereits verschiedene solcher Projekte, namentlich Titanium [2] und Bitcrime[3], durchgeführt.

Im Rahmen dieser Bachelorarbeit soll der Frage nachgegangen werden, welche Aussagekraft und welche Grenzen Analysemethoden zur Nachverfolgung von Bitcointransaktionen haben und wie sie sich auf die Anonymität auswirken. Das Ziel dieser Arbeit ist es, einen Überblick zu geben, welche Techniken für Transaktions- und Entitätsverfolgung es gibt, wie und wann sie eingesetzt werden können, wie sie möglicherweise verbessert werden könnten und wo ihre Grenzen liegen.

Zur Beantwortung der gestellten Forschungsfrage wird in dieser Arbeit eine Literaturrecherche durchgeführt. Alle verwendeten Quellen dieser Recherche sind im Literaturverzeichnis aufgeführt. Neben den literarischen Quellen wird in dieser Arbeit auf ein Interview mit Jakob Hasse verwiesen. Hasse ist Mitentwickler des Analyseprogramms *dence blockchain investigator* [4], welches im Rahmen des EU-Projekts Titanium und Bitcrime entwickelt wurde. Das Interview wurde am 05.01.2024 durchgeführt. Sinngemäße Zitate daraus werden in dieser Arbeit verwendet.

Die notwendigen technischen Grundlagen werden im zweiten Kapitel erläutert. Das beinhaltet die grobe technische Funktionsweise von Bitcoin, einige Begriffsklärungen, sowie die graphentheoretischen Grundlagen, die für diese Arbeit vorausgesetzt werden. Das erste Kernthema beschäftigt sich mit der Identifizierung von Transaktionen auf der Blockchain. Es beschreibt, wie mithilfe von Metainformationen einer Transaktion diese auf der Bitcoin Blockchain gefunden werden kann. Im zweiten Kernthema werden Clusteringheu„0ristiken beleuchtet, die es ermöglichen, verschiedene Adressen einem gemeinsamen Eigentümer zuzuordnen. Speziell wird dabei auf die Multi-Input Heuristik und die Wechselgeldheuristik eingegangen. Weiterhin werden Methoden vorgestellt, die auf speziellen Transaktionen beruhen oder eine aktive Teilname am Bitcoinnetzwerk erfordern. Im folgenden Kapitel wird auf die Qualität dieser Methoden eingegangen. Es wird zunächst aufgezeigt, wie ein Datensatz erstellt werden muss, damit eine statistische Untersuchung überhaupt möglich und sinnvoll ist. Außerdem werden die Kennzahlen erläutert, anhand derer die Heuristiken verglichen werden können. Die statistische Analyse und die damit verbundene qualitative Einschätzung der Methoden erfolgt anschließend. In Kapitel 6 werden Nachverfolgungsmethoden vorgestellt, die auf

Informationen abseits des Bitcoin-Systems basieren. Dazu gehören Identifikationsmethoden durch Webcrawling und -scraping, Clusteringmethoden auf Basis von Bitcoinforks sowie die Identifikation von Transaktionsteilnehmern durch IP-Adressen. Das letzte Kernthema beschäftigt sich mit der Analyse von Finanzströmen. Hier wird dargestellt, wie mithilfe von Graphen Muster in Finanzströmen entdeckt werden können und wie Transaktionswerte über mehrere Transaktionen hinweg verfolgt werden können.

2 Technische Grundlagen

2.1 Grundlagen von Bitcoin

Bitcoin ist eine verteilte, dezentrale Kryptowährung, welche 2008 in einem Paper [5] vom Pseudonym Satoshi Nakamoto vorgestellt wurde. Sie ermöglicht Nutzern den Austausch von Werten ohne eine zentral kontrollierte Instanz wie eine Bank. Bitcoin verbindet verschiedene technische Konzepte miteinander und realisiert dadurch ein sicheres und unabhängiges Zahlungssystem.

Die Übertragung von Werten wird in Bitcoin über Transaktionen realisiert. Diese Transaktionen bilden einen Werttransfer von Adressen zu Adressen ab. Transaktionen werden von einem Peer-to-Peer Netzwerk validiert und in Blöcken gespeichert. Durch die Verkettung dieser Blöcke werden sie, sowie die Transaktionen darin, unveränderbar.[5]

2.1.1 Adressen und Adresstypen

Bitcoinadressen sind alphanumerische Zeichenketten, die den Empfänger einer Transaktion darstellen. Technisch betrachtet sind sie eine Kombination aus dem Identifikator eines Adresstyps, einem Hashwert und einer Prüfsumme. Der Identifikator zeigt an, wie der darauf folgende Hashwert zu interpretieren ist. Abhängig vom Adresstyp kann dieser unterschiedlichen Ursprungs sein.

P2PKH - Pay to Public Key Hash

P2PKH hat den Identifikator 1, weshalb auch alle P2PKH-Adressen mit einer 1 beginnen. Dieser Adresstyp wird genutzt für einfache Transaktionen an einen anderen Nutzer. Bei P2PKH stammt das Urbild des Hashwerts von einem öffentlichen Schlüssel. Möchte ein Nutzer die Coins verwenden, die an seine P2PKH-Adresse gesendet wurden, muss er durch eine digitale Signatur mit dem zugehörigen privaten Schlüssel nachweisen, dass er der Eigentümer des öffentlichen Schlüssels ist, dessen Hashwert nach dem Identifikator steht. [6, S. 136]

P2SH - Pay to Script Hash

P2SH hat den Identifikator 3. Dementsprechend beginnen alle P2SH-Adressen mit diesem Identifikator. Dieser Adresstyp weist an, den nachfolgenden Hashwert als den eines Skriptes zu interpretieren. Dieses Skript ist eine Ansammlung von Bedingungen, die erfüllt sein müssen, damit die Coins, die an die Adresse gesendet wurden, ausgegeben werden können. Mit diesem Adresstyp können komplexe Transaktionsbedingungen realisiert werden. [6, S. 81 f.]

Über ein solches Skript ließe sich zum Beispiel eine Multisignatur bzw. Threshhold-Multisignatur erzwingen. Es könnte lauten: Überwiesen wird an die öffentlichen Schlüssel a, b und c. Um den Transaktionswert einzulösen, müssen mindestens 2 dieser 3 Schlüssel ihre Signatur bereitstellen.

Möchte ein Nutzer die Coins verwenden, die an eine P2SH-Adresse gesendet wurden, muss er das entsprechende Skript sowie die Lösung des Skriptes bereitstellen. [6, S. 81 f.] Es gibt neben Multisignatur noch weitere Anwendungsbereiche für P2SH-Adressen, zum Beispiel als Brücke für SegWit. [6, S. 336 f.] Die Bedeutung des Skripts offenbart sich allerdings erst bei Weiterverwendung der Coins. [6, S. 152 f.]

P2WPKH/P2WSH - Native SegWit

Native SegWit Adressen sind dazu da, die Transaktionsgröße zu reduzieren. Das Konzept ermöglicht es, Signaturen und Skripte außerhalb der Transaktion zu speichern. [6, S. 329 f.] Diese Adresstypen nutzen das Präfix: bc1q. Funktionell und Anwendungstechnisch ähneln P2WPKH und P2WSH aber ihren NonSegWit-Äquivalenten.

P2TR - Pay to Taproot

P2TR ist die neueste Entwicklung der Bitcoinadressen. Es behält alle Vorteile von Native Segwit, ermöglicht jedoch eine erheblich gesteigerte Performance durch einen neuen Signaturalgorithmus sowie eine verbesserte Privatsphäre, die einige der im Folgenden beschriebenen Heuristiken bricht. Dieser Adresstyp verwendet das Präfix bc1p. [7],[8] P2TR kann seit Ende 2021 genutzt werden, fand jedoch lange Zeit nur sehr begrenzt Adoption. Erst seit Mitte 2023 findet P2TR signifikante Anwendung, weshalb die Quellenlage diesbezüglich stark begrenzt ist. [9] Aus diesen Gründen wird Taproot in dieser Arbeit nicht betrachtet.

2.1.2 Transaktionen

Transaktionen (TX) repräsentieren Übertragungen von Werten. Sie bestehen aus einer Menge von Transaktionseingängen (Inputs) und einer Menge von Transaktionsausgängen (Outputs). Die Inputs einer Transaktion beziehen sich auf Transaktionsoutputs vorangegangener Transaktionen. Aus diesem Grund werden Transaktionsoutputs, die noch nicht in einer Transaktion verwendet wurden, auch als UTXOs (Unspent Transaction Output) bezeichnet. [6, S. 117 ff.]

Multi-Input / Multi-Output Transaktionen

Im einfachsten Fall besitzt ein Nutzer eine UTXO, die genau den Wert hat, den er auch übertragen möchte. In einem solchen Fall setzt er diese UTXO als Input und erstellt einen Output für den Empfänger. In der Realität ist dieser Fall jedoch häufig nicht gegeben.

Besitzt ein Nutzer keine UTXO, die den vollständigen Wert der Überweisung abdeckt, muss er mehrere UTXOs als Input einsetzen. Dabei ist es möglich, dass die UTXOs von verschiedenen Bitcoinadressen stammen. Wichtig ist nur, dass die Werte der UTXOs in Summe den Überweisungsbetrag abdecken. [6, S. 121–129]

Es ist einem Nutzer außerdem möglich, mehrere Überweisungen in einer Bitcoin-Transaktion abzuwickeln. Dafür erstellt er mehrere Outputs und legt fest wieviele Coins an welchen Output gehen sollen.

UTXOs müssen in einer Transaktion jedoch immer vollständig ausgegeben werden. Anders ausgedrückt: Es ist nicht möglich, nur einen Teil einer UTXO auszugeben. Um den überschüssigen Wert einer UTXO nicht zu verlieren, können Nutzer einen Output an eine eigene Adresse erstellen. So erhält der Nutzer eine neue UTXO mit seinem Wechselgeld. Ein solcher Output wird darum auch als Wechselgeldoutput bezeichnet.

Werden die Inputs nicht vollständig durch die Outputs aufgebraucht, gilt diese Differenz als Transaktionsgebühr. Diese Transaktionsgebühr ist ein Anreiz für das System, diese Transaktion möglichst schnell zu verarbeiten.[6, S. 121–129]

Technischer Aufbau einer Transaktion

Innerhalb des Bitcoin Peer-to-Peer Netzwerks werden Transaktionen als rohe Daten (Rohtransaktionen) verbreitet. Rohtransaktionen sind serialisierte Datenblöcke, die alle Transaktionsinformationen enthalten. Um diese Informationen für Menschen lesbar zu machen, können sie dekodiert werden. [6, S. 118] Eine beispielhafte dekodierte Transaktion (JSON) ist in Anhang A vorhanden. Nachfolgend wird erläutert, welche relevanten Informationen in einer Bitcoin Transaktion enthalten sind.

version : Ist die Versionsnummer des Transaktionsformats. Für mögliche Änderungen am Transaktionsaufbau.

locktime : Die Locktime ist ein Wert, über den festgelegt werden kann, dass eine Transaktion erst ab einem gewissen Zeitpunkt verarbeitet werden darf.

vin : Ist eine Liste von Transaktionsinputs, die angeben, woher die Bitcoins stammen. Jeder Eintrag enthält eine Transaktions-ID (txid). Die Transaktions-ID dient als Identifikator und entspricht dem Hashwert der Transaktion auf dessen Output sich dieser Input bezieht. Mit dem Ausgabeindex (vout) wird spezifiziert, auf welchen der Outputs der Transaktion sich der Input bezieht. In scriptSig befindet sich das *Redeemscript*. Dieses enthält die erforderlichen Daten zur Autorisierung der Transaktion. Bezogen auf P2PKH-Adressen umfasst das Redemmscript typischerweise die Signatur und den öffentlichen Schlüssel. Außerdem enthält jeder Transaktionsinput eine Sequenznummer (sequence). Diese kann abhängig von der Version unterschiedliche Zwecke erfüllen.

vout : Ist eine Liste der Transaktionsoutputs. Jeder Eintrag enthält den Wert, der auf diesem Output ausgezahlt wird (value), sowie ein Lockingscript (scriptPubKey). Dieses definiert die Bedingungen, unter denen die Bitcoins weiterverwendet werden dürfen.

[6, S. 118 ff.]

2.1.3 Blöcke, Blockchain und Mining

Blöcke bestehen aus einem Blockheader und einer Sammlung von Transaktionen. [6, S. 196 f.] Jedem Knoten im Bitcoin Netzwerk ist es möglich, solche Blöcke zu erstellen. Die erste Transaktion ist dabei die sogenannte Coinbase-Transaktion. In dieser überträgt der Ersteller des Blockes die Transaktionsgebühren und eine pauschale Blockbelohnung an sich selbst. [6, S. 221 f.]

Die Blockchain ist eine Datenstruktur ähnlich einer “Linked List“, bei der ein Element auf seinen Vorgänger verweist. Im Fall der Blockchain wird jedoch nicht nur auf das vorherige Element verwiesen, sondern jedes Element ist in seiner Gültigkeit abhängig vom vorherigen. Jeder Block ist durch einen Hashwert identifiziert. Ein neuer Block enthält den Hashwert des vorherigen Blockes in seinem Header, wodurch sein eigener Hashwert von dem des vorherigen Blockes abhängt. Würde also ein Block und damit einhergehend sein Hashwert verändert, würden alle nachfolgenden Blöcke ebenfalls ihren Hashwert ändern. Diese rekursive Abhängigkeit garantiert die Integrität und Unveränderlichkeit der Blöcke und damit der Transaktionen. [6, S. 195 ff.]

Bitcoin verwendet einen Konsensmechanismus namens "Proof of Work"(PoW), um Einigkeit darüber zu erlangen, welche Blöcke der Blockchain angehören. Dieser Konsens wird durch den Hashwert eines Blocks erreicht, der kleiner sein muss als ein vorgegebener Grenzwert, damit das Bitcoin-Netzwerk den Block akzeptiert. Wenn ein Block diesen Grenzwert nicht erfüllt, wird er von den anderen Teilnehmern des Netzwerks verworfen, und die Suche wird fortgesetzt. Erfüllt ein Block diese Bedingung jedoch, wird überprüft, ob alle Transaktionen im Block gültig sind. Wenn auch das der Fall ist, wird der Block als Teil der Blockchain akzeptiert, und die Suche nach dem nächsten Block beginnt. [6, S. 242 f.]

Finden mehrere Teilnehmer des Netzwerkes gleichzeitig einen validen Block, kommt es zu einem sogenannten Fork. In einem solchen Fall bestehen zwei gleichermaßen valide Stränge der Blockchain, die miteinander in Konkurrenz stehen. Auf Dauer wird sich einer der beiden Stränge durchsetzen, weil Miner sich für einen der beiden Stränge entscheiden. Indem sie ihre Rechenleistung auf die Weiterentwicklung eines Strangs konzentrieren, entwickelt sich ein Strang schneller als der andere. Der längere Strang wird als die gültige Version der Blockchain angesehen und der andere verworfen. [6, S. 213 f.]

Die Gefahr, dass Transaktionen durch Forks ungültig werden, besteht immer. Sie wird jedoch geringer je mehr Blöcke an den Block angehangen werden, der die Transaktion beinhaltet. Einfach ausgedrückt: Ein Block wird durch seine nachfolgenden Blöcke bestätigt.

2.1.4 Bitcoin Peer-to-Peer Netzwerk

Bitcoins Netzwerkstruktur basiert auf einer Peer-to-Peer Architektur. Es besteht aus einzelnen Netzwerkknoten, die untereinander kommunizieren, ohne dass es einer zentralen Instanz bedarf. Das Peer-to-Peer Netzwerk dient dazu, Informationen, wie Transaktionen oder Blöcke, zu kontrollieren und unter allen Knoten zu verbreiten. [6, S. 171 ff.]

Verbreitung von Transaktionen

Wenn ein Nutzer eine Transaktion erstellt, sendet er sie an seine Nachbarknoten. Erhält ein Knoten eine Transaktion, überprüft dieser die Transaktion auf Aktualität sowie Validität und leitet sie gegebenenfalls an seine Nachbarknoten weiter. Auf diese Weise verbreiten sich Transaktionen nach und nach im gesamten Netzwerk. Ein Knoten des Netzwerks leitet eine Transaktion nicht weiter, wenn eine der folgenden Bedingungen erfüllt ist.

Aktualität:

- Der Knoten hat diese Transaktion erst kürzlich weitergeleitet
- Die Transaktion ist bereits Teil der Blockchain.

Validität:

- Die Transaktion versucht UTXOs zu verwenden, die bereits in einer anderen Transaktion verwendet wurden.
- Die Signatur eines oder mehrerer Inputs ist ungültig.
- Ein oder mehrere Input(s) der Transaktion referenzieren einen Output, der nicht existiert.

[10, S. 471 f.]

Verbreitung von Blöcken

Analog zu Transaktionen werden auch Blöcke von einem Knoten erstellt und an seine Nachbarknoten gesendet. Erhält ein Knoten einen Block, muss er diesen auf Validität prüfen. Dazu gehört eine Prüfung der Transaktionen im Block sowie eine Prüfung des Blockes selbst.

Transaktionen:

- Alle Transaktionen sind valide.
- Es finden keine Wiederholungen von Transaktionen statt, weder innerhalb des Blocks noch Wiederholungen aus einem vorherigen Block.

Block:

- Der Block ist syntaktisch korrekt.
- Der Block muss die Maximalgröße von einem Megabyte einhalten.
- Der Hashwert des Blockes darf den aktuellen Grenzwert nicht überschreiten.
- Der Zeitstempel des Blockes ist zulässig.
- Die erste Transaktion des Blockes ist eine Coinbase Transaktion, und es existiert nur eine.

[6, S. 238 f.]

Ist der Block valide, fügt der Knoten den Block seiner Kopie der Blockchain hinzu und leitet den Block an seine Nachbarknoten weiter. Auch diese überprüfen die Korrektheit des Blockes, fügen ihn gegebenenfalls ihrer Blockchain hinzu und leiten ihn weiter.

2.2 Anonymität und Pseudonymität

Nach der Definition von Pfitzer und Hansen ist Anonymität, die Unfähigkeit eines Angreifers, ein Subjekt innerhalb einer Gruppe von Subjekten, zu identifizieren. [11, S. 10] Dies trifft im Bitcoinsystem nicht zu. Nutzer des Bitcoinnetzwerks interagieren im Bitcoinsystem mithilfe von individuellen Adressen, und diese sind identifizierbar. Diese Dynamik trifft vielmehr auf die Definition der Pseudonymität

zu. Ein Pseudonym ist nach der Definition von Pfitzer und Hansen: Ein Identifikator eines Subjekts, der nicht mit seinem wirklichen Namen identisch ist. [11, S. 21] Bitcoin ist also per se kein anonymes sondern vielmehr ein pseudonymes Zahlungssystem.

2.3 Graphentheoretische Konzepte

Graphen sind abstrakte Strukturen, die aus einer Menge von Knoten und Kanten bestehen, die die Verbindungen zwischen diesen Knoten repräsentieren. [12, S. 5] In Zahlungssystemen können Transaktionen über Graphen dargestellt werden, um Finanzflüsse oder zusammengehörige Knoten zu identifizieren.

2.3.1 Graphenstruktur

Ein Graph ist eine abstrakte Struktur, die eine Menge von Objekten zusammen mit den zwischen diesen Objekten bestehenden Beziehungen repräsentiert. Ein Graph besteht dabei aus einer Menge von Knoten und einer Menge von Kanten. Die Knoten werden untereinander durch Kanten verbunden, wobei eine Kante immer genau zwei Knoten miteinander verknüpft.

In der Graphentheorie wird zwischen gerichteten und ungerichteten Graphen unterschieden. Bei ungerichteten Graphen sind alle Kanten symmetrisch. Das bedeutet: Eine Beziehung zwischen zwei Knoten besteht beidseitig. Bei gerichteten Graphen sind Kanten einseitig. Solche Beziehungen sind gerichtet und bestehen nicht zwangsläufig in beiden Richtungen. [12, S. 5–10]

Bei der Darstellung von Zahlungssystemen werden im Wesentlichen gerichtete Graphen verwendet, da Transaktionen gerichtete Geldflüsse sind. Auch speziell bezogen auf das Bitcoinsystem gibt es Strukturen, die sich ideal durch diese Art von Graphen abbilden lassen.

Beispielsweise lassen sich Bitcointransaktionen als Graph darstellen, dessen Knoten Adressen und Kanten Transaktionen sind. Ein solcher Graph kann für die Analyse von Finanzflüssen genutzt werden.

2.3.2 Spezielle Elemente in Graphen

In Graphen gibt es spezielle Elemente und Muster, die verschiedene Eigenschaften und Strukturen des Graphen repräsentieren. Im Folgenden werden Graphenstrukturen beschrieben, die in dieser Arbeit Anwendung finden.

Knoten werden durch Kanten miteinander verbunden. Eine Kante besteht dabei immer zwischen genau zwei Knoten. Das bedeutet aber nicht, dass zwischen zwei Knoten nur eine Kante bestehen kann. Bestehen zwischen zwei Knoten mehrere Kanten, werden diese als *Mehrfachkanten* bezeichnet. [13, S. 245]

Ein *Untergraph* eines Graphen ist ein Graph, der durch Auswahl einer Teilmenge von Knoten und den dazugehörigen Kanten aus dem Ursprünglichen entsteht. Mit anderen Worten: Ein Untergraph enthält eine Teilmenge der Knoten und Kanten des ursprünglichen Graphen, wobei die Beziehungen zwischen den ausgewählten Knoten erhalten bleiben. [13, S. 133]

Ein Weg oder Pfad in einem gerichteten Graphen ist eine Sequenz von Knoten und gerichteten Kanten, die es erlaubt, von einem Startknoten zu einem Zielknoten zu gelangen. Ein *Zyklus oder Kreis* in einem gerichteten Graphen ist ein Weg, bei dem der Start- und Endknoten gleich ist. [13, S. 133]

Eine *Community* bezieht sich auf eine Gruppe von Knoten, die untereinander stärker verbunden sind als mit Knoten außerhalb der Gruppe. Zwischen den Knoten der Community existieren viele Kanten, aber es existieren nur wenige Kanten zu Knoten außerhalb der Community. [14, S. 490] Extremformen von Communities sind *Cliquen oder vollständige Untergraphen*. Bei solchen Graphen sind alle Knoten des Untergraphen miteinander verbunden. [13, S. 151]

3 Identifizierung und Zuordnung von Transaktionen auf der Blockchain

Um eine Transaktions- oder Entitätsverfolgung auf der Bitcoin-Blockchain durchzuführen, ist es von entscheidender Bedeutung, einen Ausgangspunkt zu finden, von dem aus die Analyse beginnen kann. Im Folgenden wird erläutert, wie Transaktionen auf der Bitcoin-Blockchain identifiziert und zugeordnet werden können, um einen Startpunkt festzulegen.

3.1 Ähnlichkeit der Transaktionszeiten

Die Transaktionszeit ist ein entscheidender Aspekt bei der Identifizierung von Transaktionen auf der Bitcoin-Blockchain. Eine Transaktion enthält jedoch selbst keinen Zeitstempel, der den Zeitpunkt festhält, zu dem sie ins Bitcoin Netzwerk geschickt wurde. Der einzige verfügbare Zeitstempel ist der des Blockes, in dem die Transaktion der Blockchain hinzugefügt wird. Die Zeit, die zwischen Sendung der Transaktion in das Bitcoinnetzwerk und Aufnahme in einen Block vergeht, hängt maßgeblich von der Auslastung des Bitcoinnetzwerks und den angehängten Transaktionsgebühren ab. Bei hoher Auslastung des Netzwerks können Transaktionen mit geringen Gebühren erst stark verzögert in einen Block aufgenommen werden. Teilweise werden sie sogar verworfen. [6, S. 126 f.] Der Zeitstempel des Blockes sagt daher nicht zwangsläufig etwas über die Transaktionszeit der Transaktion aus. Das gilt insbesondere deshalb, weil der Zeitstempel für Blöcke relativ frei gewählt werden kann. Vereinfacht ausgedrückt kann der Miner eines Blockes den Zeitstempel in einem Rahmen von etwa 2 Stunden frei wählen. [15, S. 1]

Einem Knoten im Peer-to-Peer Netzwerk ist es jedoch möglich, Transaktionen mit dem Zeitstempel zu verknüpfen, an dem er erstmalig von der Transaktion erfährt. Diese Methode ermöglicht zwar weder eine exakte zeitliche Einordnung noch eine garantierte Reihenfolge von Transaktionen. Dieser Zeitstempel ist aber erheblich genauer als der des Blockes, in den eine Transaktion aufgenommen wird.

In einer Untersuchung stellen Decker u.A. die Dauer der Propagation von Informationen im Bitcoinnetzwerk dar. Sie zeigen, dass es durchschnittlich 12,6 Sekunden dauert, bis ein Knoten eine Information empfängt. 50% aller Knoten erhalten die Information bereits nach 6,5 Sekunden. [16, S. 5 f.] Diese Daten beziehen sich auf Informationen im Allgemeinen. Es ist anzunehmen, dass Transaktionsinformationen sich erheblich schneller im Netzwerk verbreiten als Blockinformationen. Schließlich müssen für die Blockverifizierung statt einer zum Teil mehrere 1000 Transaktionen geprüft werden. Davon ausgehend ist anzunehmen, dass ein Knoten unabhängig von seiner Position im Netzwerk Transaktionen innerhalb weniger Sekunden erhält. Eine Untersuchung von Miŝić u.A. geht sogar von einer Transaktionspropagation an 99% aller Knoten in einer halben Sekunde aus. [17, S. 5 f.]

Ist einem potentiellen Angreifer der ungefähre Zeitraum einer Transaktion bekannt, lässt sie sich von diesem identifizieren oder zumindest eingrenzen.

3.2 Ähnlichkeit der Transaktionswerte

Neben der Transaktionszeit ist auch der Wert einer Transaktion ein wichtiger Faktor bei der Identifizierung und Analyse von Transaktionen. Ist einem Angreifer der genaue Transaktionsbetrag bekannt, lassen sich die möglichen Transaktionen auf der Blockchain eingrenzen. [18, S. 5] [19, S. 93] Selbst wenn keine Transaktion mit einem exakten Wert gefunden wird, können durch die Analyse von Transaktionswerten Korrelationen identifiziert werden. Das gilt insbesondere dann, wenn Transaktionen in Teiltransaktionen aufgeteilt werden.

Ein Beispiel für diese Beobachtung ist das Folgende. Angenommen Alice überweist eine Summe von Bitcoins an Bob. Alice möchte jedoch ihre Privatsphäre schützen und den Transaktionsbetrag verschleiern. Dies kann durch die Aufteilung der Transaktionssumme auf mehrere Adressen erreicht werden, bevor sie schließlich auf Bobs Adresse zusammengeführt wird. Siehe Abbildung 3.1.

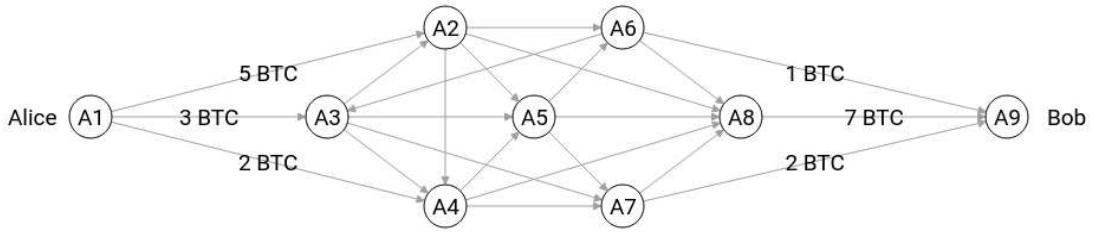


Abbildung 3.1: Korrelation von Transaktionswerten

Der genaue Pfad, den die Bitcoins dabei über die verschiedenen Adressen nehmen, ist unwichtig. Entscheidend ist lediglich, dass die Gesamtsumme letztendlich auf einer einzigen Adresse ankommt.

Indirekt ergibt sich daraus eine mögliche Mitigationsmaßnahme. Überweist Alice über einen komplexen Weg an mehrere Adressen von Bob, tritt der ursprüngliche Transaktionswert nicht mehr auf. Sofern Bob diese Adressen nicht selbst zusammenführt, lässt sich diese Transaktion für einen unbeteiligten Beobachter nur schwer zusammenführen.

4 Heuristiken und Methoden auf Basis der Transaktionseigenschaften

In diesem Kapitel werden Heuristiken und Methoden vorgestellt, die dazu dienen, Bitcoinadressen einem gemeinsamen Eigentümer, einer *Entität*, zuzuordnen. Die im Folgenden beschriebenen Heuristiken haben ihren Ursprung in einem Paper von Reid und Harrigan. Die Grundannahmen beider Heuristiken gehen auf eine Untersuchung [20] zurück, die sich erstmals mit der Nachverfolgbarkeit von Bitcointransaktionen beschäftigte. Neben den passiven Heuristiken werden zwei aktive Methoden bzw. Angriffe vorgestellt, die dem gleichen Zweck dienen.

4.1 Multi-Input Heuristik

Die primäre Heuristik, mit der Adressclustering betrieben werden kann, ist die sogenannte Multi-Input oder auch Common-Input-Ownership Heuristik. Wie in den technischen Grundlagen dargelegt, können für eine Transaktion mehrere UTXOs als Input verwendet werden. Die einzelnen UTXOs, die für eine Transaktion verwendet werden, müssen dafür nicht zwangsläufig auf der gleichen Adresse liegen. Wie schon im ursprünglichen Whitepaper von Satoshi Nakamoto beschrieben, sollte sogar für jede empfangene Transaktion ein neues Schlüsselpaar und damit eine neue Adresse generiert und verwendet werden. Ein Verbinden von mehreren Adressen, durch die gemeinsame Verwendung als Inputs einer Transaktion, lässt sich aber nicht vermeiden. [5, S. 6]

4.1.1 Allgemeine Multi-Input Heuristik

Funktionsweise

Die Grundannahme der Multi-Input Heuristik besagt, dass bei Transaktionen mit mehreren Inputs davon ausgegangen werden kann, dass alle Inputadressen zur gleichen Entität gehören. [20, S. 207] Diese Annahme ist deshalb naheliegend, da alle privaten Schlüssel der Inputadressen für die vollständige Signatur der Transaktion benötigt werden. [21, S. 168]

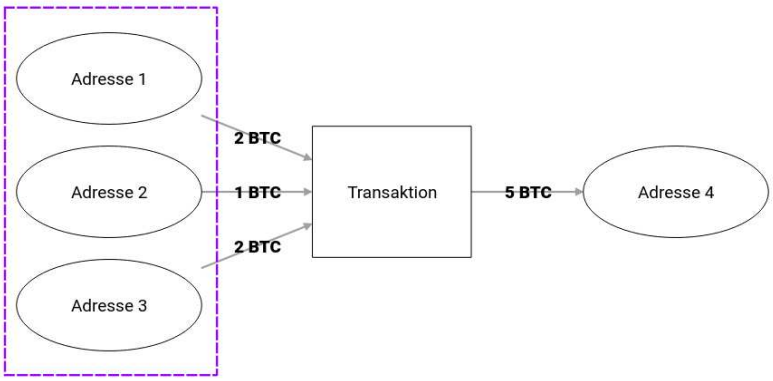


Abbildung 4.1: Graphische Darstellung der Multi-Input Heuristik.

Wie in Abbildung 4.1 dargestellt, lassen sich alle Transaktionsinputs und damit einhergehend alle Inputadressen einer gemeinsamen Entität zuordnen. Die Zusammenführung von Clustern in dieser Heuristik folgt einer Äquivalenzrelation. Das bedeutet, dass die Eigenschaften der Reflexivität, Symmetrie und Transitivität erfüllt sind.

Reflexivität: Eine Adresse A befindet sich immer mit sich selbst in einem Adresscluster.

Symmetrie: Werden Adresse A und B als Input einer Transaktion verwendet, befinden sich die Adressen A und B sowie umgekehrt auch B und A im selben Cluster.

Transitivität: Befinden sich in Transaktion $T1$ die Inputadressen A und B und in $T2$ die Inputadressen B und C, so sind auch die Adressen A und C im gleichen Adresscluster.

Durch die Eigenschaft der Transitivität wachsen die Adresscluster mit zunehmender Transaktionszahl immer weiter an. [22, S. 11] Die Transitivität impliziert außerdem die Möglichkeit, Cluster miteinander zu verbinden. Beim Zusammenführen von Clustern muss jedoch speziell auf die Größen geachtet werden.

Wenn Adresscluster durch diese Heuristik verbunden werden, lässt sich das Wachstum des Clusters messen. [22, S. 8]

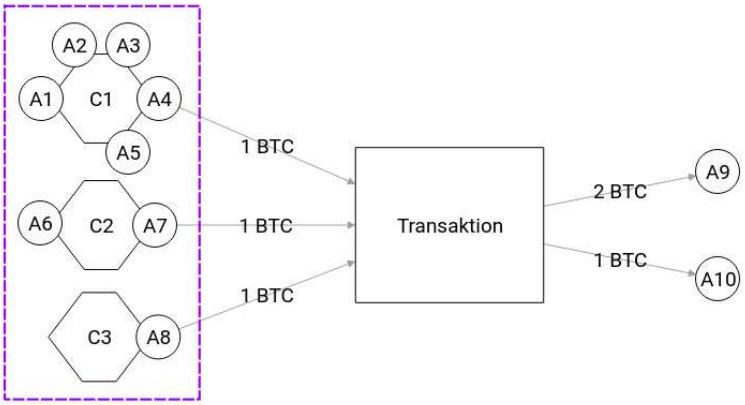


Abbildung 4.2: Zusammenführung von Adressclustern.

Werden durch eine Transaktion beispielsweise drei Adresscluster der Größe 5, 2 und 1 zusammengeführt, wächst das 5-Adresscluster um 1 und 2. Eine graphische Darstellung dieser Zusammenführung von Clustern verschiedener Größe ist zu sehen in Abbildung 4.2. Die Cluster sind als Hexagone dargestellt. Ihre zugehörigen Adressen A1-A8 sind an deren Ecken. Obwohl nur die Adressen A4, A7 und A8 in einer Transaktion als Input gemeinsam verwendet werden, lassen sich die Adressen aller Cluster zu einem Gesamten verbinden.

Nach einer Analyse von Harrigan und Fretter konzentriert sich die Wachstumsverteilung im Wesentlichen auf einzelne Adressen oder kleine Cluster. [22, S. 7–9] Ein kleines Clusterwachstum ist typisch für Privatpersonen, die auf einem Wallet Bitcoins senden und empfangen. Ein großes Wachstum tritt zum Beispiel auf, wenn zwei Wallets zusammengeführt werden. Abhängig von der Größe der Wallets könnten dabei Cluster mit einer jeweiligen Größe von >10 zusammengeführt werden. Dafür reicht es, entsprechend der Transitivität, eine Transaktion mit zwei Inputs (eine vom jeweiligen Wallet)

auszuführen. Ähnlich einer solchen Walletzusammenführung mit noch größerem Wachstum der Cluster wäre ein Zusammenschluss von nicht personellen Entitäten wie Unternehmen. Angenommen Kryptobörse A und B fusionieren und haben jeweils ein Adresscluster von 2000 Adressen. In einer Transaktion verwenden sie jeweils eine Adresse ihrer Cluster. Das Clusterwachstum, verursacht durch diese Transaktion, ist dementsprechend hoch.

Nach einer Analyse von Harrigan und Fretter tritt ein Clusterwachstum von mehr als 10 lediglich in 0,1% und ein Wachstum von mehr als 100 in 0,01% aller Fälle auf. Es ist also sehr selten, dass zwei große Adresscluster zusammengeführt werden. Und es wird seltener, je größer das Clusterwachstum ist. [22, S. 8–11] Diese Seltenheit sollte nicht mit der Wahrscheinlichkeit verwechselt werden, dass die Verbindung von zwei großen Clustern korrekt ist. Allerdings kann das Zusammenführen von zwei großen Adressclustern ein Indiz dafür sein, dass die Heuristik mit ihrer Annahme in diesem Fall falsch liegt. [22, S. 11] Auf die Gründe für einen solchen Fehler der Multi-Input Heuristik wird im folgenden Abschnitt eingegangen.

Unabhängig davon, ob korrekte Zusammenführung oder nicht, können große Clusterfusionen als außergewöhnlich markiert werden. Adresscluster mit solchen Markierungen beinhalten womöglich Adressen von mehr als einer Entität. Als Beispiel führen Harrigan und Fretter das Cluster von Mt. Gox an. Die Outputs der Wachstumsförderndsten 0,1% der Bitcointransaktionen zwischen Juli 2010 und Februar 2014 konnten demnach Mt.Gox zugeordnet werden. [22, S. 11]

Ausnahmen und Fehler

Auch wenn die zugrundeliegende Annahme dieser Heuristik sehr naheliegend scheint, ist sie dennoch nicht immer korrekt. Die Verwendung verschiedener Inputadressen ist zwar ein starker Indikator für eine gemeinsame Entität, sie ist aber kein Beweis. [23] Im Folgenden wird aufgezeigt, wie ein Fehler der Heuristik provoziert werden kann.

Da die Signaturen einer Transaktion von einander nicht abhängen, ist das folgende Szenario nicht auszuschließen. Zwei Entitäten einigen sich auf eine Transaktion. Besteht Einigkeit über Struktur und Werte der Transaktion, versehen die Entitäten sie unabhängig voneinander mit einer Signatur und führen die Signaturen zusammen. [23] In einem einfachen Beispiel würden Alice und Bob dafür eine Transaktion erstellen, in der sie beide 2 Bitcoin als Input bereitstellen, und 2 Bitcoin als Output erhalten.

Das obige Beispiel beschreibt ein Konzept, das erstmals von Maxwell beschrieben wurde und CoinJoin genannt wird. [23] Dieses Konzept ist nicht nur auf die Kopplung von zwei Entitäten beschränkt. Es ermöglicht auch die Abwicklung multipler Transaktionen unterschiedlicher Entitäten in einer einzigen Bitcoin-Transaktion. Diese Praxis, mehrere Transaktionen zu bündeln, wirkt sich dabei nicht nur auf die gewonnene Anonymität aus. Sie resultiert ebenso in Einsparungen bei den Transaktionsgebühren durch die reduzierte Gesamtgröße einer Transaktion im Vergleich zu mehreren einzelnen. [23]

Ein konkretes Beispiel ist das Folgende: Alice möchte Bob 3 Bitcoin senden. Etwa zur gleichen Zeit, möchte auch Carol 4 Bitcoin an Dan senden. Um ihre Anonymität zu schützen und einen Fehler der Multi-Input Heuristik herbeizuführen, entschließen sich Alice und Carol, ihre Überweisungen in

einer gemeinsamen Bitcoin Transaktion zu kombinieren. Dazu entwerfen sie eine Transaktion wie in Abbildung 4.3 dargestellt. Während die blau markierten Adressen korrekt als zusammengehörig erkannt werden, wird die rot markierte Adresse fälschlicherweise als ebenfalls zugehörig erkannt.

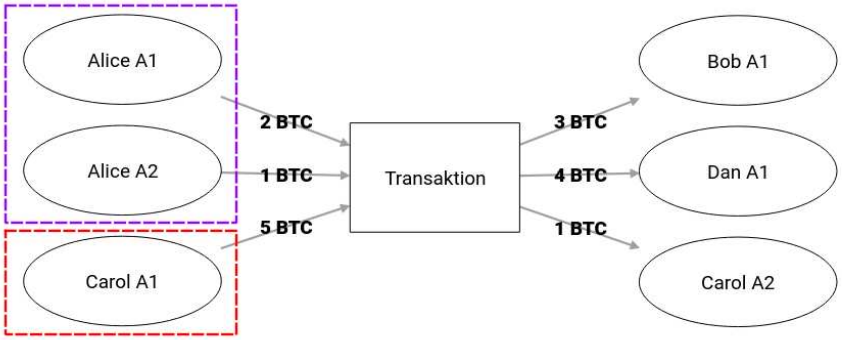


Abbildung 4.3: Bruch der Multi-Input Heuristik durch CoinJoin Transaktion

Da Alice keine UTXO besitzt, die einen Wert von 3 Bitcoin darstellt, muss sie Transaktionsinputs von 2 verschiedenen Adressen stellen: Alice A1 und Alice A2. Carol hingegen hat nur eine UTXO des Wertes 5 Bitcoin. Um ihren überschüssigen Bitcoin nicht zu verlieren, erstellt sie in der Transaktion neben den Outputs von Bob A1 und Dan A1 noch einen dritten Output Carol A2 für sich selbst, um ihr Wechselgeld zurück zu erhalten. Haben sich Alice und Carol auf den Transaktionsaufbau geeinigt, signieren beide die Transaktion.

Die Transaktion kann erst dann vom Bitcoinnetzwerk bearbeitet werden, sobald die Signaturen aller Inputs vorhanden sind. So besteht zu keiner Zeit die Möglichkeit für Alice oder Carol, die Coins der jeweils anderen zu stehlen. [23]

4.1.2 Multi-Input Heuristik Fehlervermeidung bei CoinJoin und PayJoin-Transaktionen

Die Verwendung des CoinJoin Konzepts erschwert das Adressclustering anhand der Multi-Input Heuristik. Oder anders ausgedrückt: Ohne weitere Anpassung der Heuristik wären jeder CoinJoin sowie alle nach dem Konzept von CoinJoin kombinierten Transaktionen falschpositive Ergebnisse der Heuristik.

Fehlervermeidung bei CoinJoin

In einigen Fällen von CoinJoins und kombinierten Transaktionen können die ursprünglichen Teiltransaktionen getrennt werden. Dies wird durch die übertragenen Beträge oder durch die Anordnung der In- und Outputs ermöglicht. [24, S. 5]

Im Beispiel, dargestellt in Abbildung 4.3, werden durch die Anordnung und die Werte der In- und Outputs die Teiltransaktionen unterscheidbar. Die Werte der Gesamttransaktion lassen zwei Interpretationen für eine Aufteilung in Teiltransaktionen zu:

Interpretation 1: Die Inputs Alice A1 und Carol A1 stehen in Verbindung mit den Outputs Bob A1 und Dan A1. Die andere Teiltransaktion wäre dementsprechend: Alice A2 zu Carol A2.

Interpretation 2: Die Inputs Alice A1 und A2 korrelieren mit dem Output Bob A1. Die andere Teiltransaktion wäre dann Carol A1 zu Dan A1 und Carol A2.

Interpretation 2 ist dabei wahrscheinlicher. In der gegebenen graphischen Darstellung der Transaktion addieren sich die ersten beiden Inputs zum ersten Output der Transaktion. Oder anders ausgedrückt: Die Teiltransaktionen nach Interpretation 2 lassen sich durch eine Linie voneinander trennen. Gleiches ist für Interpretation 1 nicht möglich. Allgemein ist dieses Indiz aber nur selten anwendbar, da eine einfache, nicht triviale Permutation der Inputs und Outputs diese Eigenschaft unbrauchbar macht. [24, S. 5]

Sind Teiltransaktionen unterscheidbar, wie im vorliegenden Fall, kann auf den Teiltransaktionen die Multi-Input Heuristik weiterhin ausgeführt werden. Häufig ist es jedoch ausreichend, eine CoinJoin Transaktion als solche zu identifizieren, und sie aus diesem Grund für die Multi-Input Heuristik zu sperren.

In einer Untersuchung zeigen Schnoering und Vazirgiannis, wie Heuristiken zur Erkennung von CoinJoins verwendet werden können. Das gelingt trotz unterschiedlicher Implementationen von CoinJoin in verschiedenen Wallets. [24, S. 6–15] Sie beschreiben, dass ihre Heuristiken eine niedrige Falsch-Negativ-Rate aufweisen und Transaktionen zuverlässig als CoinJoin identifizieren. Es besteht jedoch ein erhöhtes Risiko für falschpositive Ergebnisse, bei denen Transaktionen fälschlicherweise als CoinJoin-Transaktionen erkannt werden, obwohl sie keine sind. [24, S. 15]

Das bedeutet: Obwohl Teiltransaktionen nicht immer unterscheidbar sind, ist es häufig möglich, CoinJoin Transaktionen als solche zu identifizieren. [25, S. 3],[26, S. 15] So lassen sich die falschpositiven Ergebnisse der Multi-Input Heuristik zumindest reduzieren.

Fehlervermeidung bei PayJoin

Eine spezielle Form des CoinJoin ist das sogenannte PayJoin oder P2EP-Pay-to-EndPoint. Dieses Konzept wurde von Matthew Haywood 2018 erstmals vorgeschlagen. [27] Dabei wird die eigentliche Transaktion mit einer Transaktion von Empfänger zu Empfänger zusammengeführt. Im Kern ist es also ein CoinJoin kombiniert mit einer tatsächlichen Transaktion (siehe Abbildung 4.4). Auf diese Weise lässt sich wie schon beim CoinJoin die Multi-Input Heuristik brechen. Zusätzlich wird die Unterscheidbarkeit der Einzeltransaktionen durch ihre Übertragungswerte verhindert. [28, S. 2 f.]

In PayJoin-Transaktionen lassen sich Teiltransaktionen nicht identifizieren. Es gibt jedoch eine Heuristik, bei der PayJoin-Transaktionen durch die Verwendung von nicht benötigten Inputs als solche erkannt werden können. [28, S. 2]

Es gibt zwei Typen von nicht benötigten Inputs. Der erste Typ, der im Folgenden beschrieben wird, kann auf eine PayJoin-Transaktion hinweisen. Der andere Typ kann auf Wechselgeldoutputs hinweisen und wird daher in Kapitel 4.2 erläutert.[28, S. 3 f.]

PayJoin-Transaktionen bestehen im Allgemeinen aus mehreren Inputs und genau 2 Outputs. Entsprechend gehört einer der Outputs dem Empfänger der Transaktion, und ein Output besteht als Wechselgeldadresse. Allgemein erkennt die Heuristik Transaktionen mit nicht benötigten Inputs daran, dass der größte Outputwert ohne den kleinsten Inputwert gezahlt werden kann. [28, S. 4] Dieser

kleinste Input kann dann als nicht benötigt markiert werden. Zu erwähnen ist, dass auch mehrere nicht benötigte Inputs in einer Transaktion verwendet werden können. Das kann dann passieren, wenn der Empfänger einer PayJoin-Transaktion mehrere Inputs an sich selbst schickt.

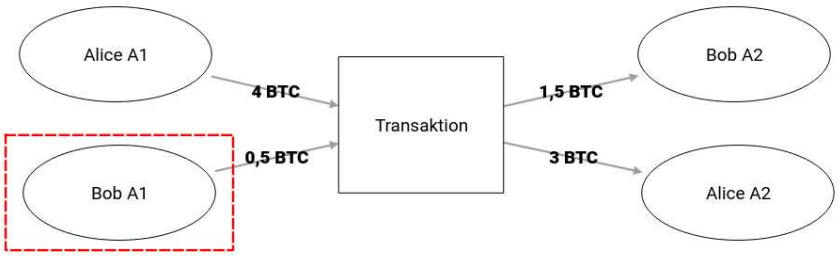


Abbildung 4.4: Graphische Darstellung des PayJoin Konzepts.

Bei der in Abbildung 4.4 dargestellten Transaktion handelt es sich um eine PayJoin-Transaktion. Der Aufbau ähnelt dem CoinJoin-Konzept, in diesem Spezialfall wird aber gleichzeitig ein Wert an Bob übertragen. Die rote Markierung von Bob A1 zeigt einen nicht benötigten Input. Im abgebildeten Beispiel sind die Eigentumsverhältnisse bekannt. Dadurch ist die Interpretation als PayJoin-Transaktion naheliegend. Diese Beispieltransaktion ließe sich aber auch bei unbekannten Eigentumsverhältnissen als PayJoin-Transaktion identifizieren. Dazu wird die Heuristik zur Erkennung von nicht benötigten Inputs verwendet.

Im Beispiel möchte Alice 1 BTC an Bob senden. Wäre das ihr einziges Ziel, wäre es auch möglich, ausschließlich diesen Input zu verwenden. In diesem Fall würden 1 BTC an Bob A2 und 3 BTC an Alice A2 ausgezahlt. Um aber die Multi-Input Heuristik zu täuschen, stellt auch Bob einen Input von 0,5 Bitcoin bereit, welcher ebenfalls an Bob A2 ausgezahlt wird. Dieser Input von Bob A1 kann nach der Grundannahme der Heuristik als nicht benötigt klassifiziert, und damit die Gesamttransaktion als PayJoin für die Multi-Input Heuristik gesperrt werden.

Sind die Eigentumsverhältnisse der Outputadressen unbekannt, ist nicht erkennbar, ob Alice 1 BTC an Bob überweist und 3 BTC als Wechselgeld erhält - oder ob Alice 2,5 BTC an Bob überweist und 1,5 BTC als Wechselgeld erhält. Die Teiltransaktionen lassen sich also nicht einfach separieren. Dafür würden Wechselgeldindikatoren benötigt. Mit diesem Problem setzt sich die Wechselgeldheuristik im nächsten Kapitel auseinander.

Es besteht jedoch ein wichtiges Problem bei dieser Heuristik. Nur weil in einer Transaktion nicht benötigte Transaktionsinputs vorkommen, macht sie das nicht zu einer PayJoin-Transaktion. Ist einer Entität die Anwendung dieser Heuristik (und die damit einhergehende Sperrung der Transaktion für die Multi-Input Heuristik) bewusst, kann diese Entität jeder Transaktion einen nicht benötigten Input hinzufügen. Das erhöht zwar die Transaktionsgebühren, führt aber dazu, dass alle Transaktionen von einer PayJoin-Erkennung abgefangen werden.

4.1.3 Erweiterung der Multi-Input Heuristik um Community-Detection

Die Funktionsweise der Multi-Input Heuristik lässt sich erweitern beziehungsweise kombinieren mit der sogenannten Community-Detection. [21, S. 170] Community-Detection selbst ist eine Methodik bekannt aus der Graphentheorie. Es gibt verschiedene Algorithmen, die dabei versuchen, mehrfach untereinander verbundene Knoten in einem Graphen auszumachen.

Funktionsweise

Das Konzept dieser Heuristik baut auf den bereits durch die Multi-Input Heuristik geclusterten Entitätsknoten auf. Die Knoten entsprechen somit schon möglichen Entitäten. Die Idee der Community-Detection Heuristik ist nun, dass verschiedene von der Multi-Input Heuristik erkannte Entitäten einer gemeinsamen Entität angehören, wenn sie untereinander Transaktionen ohne Wechselgeld versenden. [21, S. 170]

Aus graphentheoretischer Sicht betrachtet, übernimmt diese Heuristik die geclusterten Adressen und fügt diese als Knoten einem Graphen hinzu. Im Anschluss daran werden anhand der folgenden Kantenregeln gerichtete Kanten zwischen den Knoten gelegt. Zwei Knoten werden durch eine gerichtete Kante verbunden, wenn zwischen ihnen eine Transaktion besteht, die:

1. an weniger als 10 andere Entitätsknoten ausschüttet.

2. keine Ausschüttung an die Entität selbst (also eine bereits bekannte Wechselgeldadresse) vornimmt.

Auf diesem Graphen wird ein Community-Detection Algorithmus ausgeführt, um Communities zu erkennen. Diese Untergraphen sind nach Grundannahme der Heuristik als eine Entität aufzufassen. [21, S. 170] Eine visuelle Darstellung einer solchen Community-Detection in einem Graphen ist in Abbildung 4.5 zu sehen. Nach Grundannahme der Heuristik sind die blau markierten Entitäten E13, E5, E7, E3 und E6 als eine Entität aufzufassen.

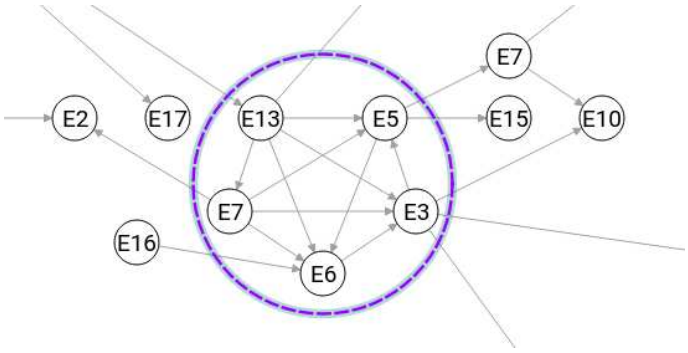


Abbildung 4.5: Markierung einer Community im Entitätsgraphen entsprechend der Community-Detection Heuristik

Die Ausführung des Community-Detection Algorithmus kann des Weiteren in einer höheren Aggregationsstufe oder auch rekursiv auf seinem eigenen Ergebnis erneut ausgeführt werden. Auf diese Weise verbindet diese Heuristik mehrere kleine Cluster zu wenigen großen. Vermehrt sieht diese Methode aber auch Zusammenhänge zwischen Clustern, wo tatsächlich gar keine bestehen. [21, S. 170–175]

Ausnahmen und Fehler

Die gerade genannte Erkennung von Zusammenhängen bei Clustern, zwischen denen tatsächlich keine bestehen, verstärkt sich, je höher die Aggregationsstufe bzw. je häufiger der Community-Detection Algorithmus rekursiv aufgerufen wird.

Entsprechend den Kantenregeln entsteht dieser Fehler genau dann, wenn zwischen zwei oder mehreren Entitäten Transaktionen ohne Wechselgeld stattfinden. Das kann passieren, wenn eine Entität eine Rohtransaktion erstellt und sein Wechselgeld vergisst. [6, S. 129 f.]

Man stelle sich zudem das folgende Szenario vor: Die Multi-Input Heuristik konnte den Entitiäten A, B und C jeweils einige Adressen zuordnen, sie jedoch nicht miteinander verbinden. Tatsächlich sind diese 3 Entitäten verschiedene Kryptobörsen und somit keine gemeinsame Entität. A, B und C senden einander Transaktionen ohne Wechselgeldoutput (A an B, B an C und C an A). Aus graphentheoretischer Sicht entsteht entsprechend der Kantenregeln ein vollständiger Untergraph. Da ein 3-vollständiger Untergraph als Community zählt, reichen diese 3 Transaktionen auf der niedrigsten Aggregationsebene aus, um alle 3 Kryptobörsen einer gemeinsamen Entität zuzuordnen.

Mögliche Verbesserungen

Diese Heuristik hat ebenfalls eine nachvollziehbare und auch häufig zutreffende Grundannahme. Sie lässt sich jedoch dahingehend verbessern, die Zahl der falschpositiven Ergebnisse zu reduzieren.

In der Untersuchung von Remy u.A. wird der Louvain-Algorithmus zur Erkennung von Communities verwendet. Dieser Algorithmus ist für ungerichtete Graphen konzipiert und berücksichtigt daher keine Mehrfachkanten. Das mehrfache Versenden von Transaktionen ohne Wechselgeld deutet auf einen Zusammenhang zwischen den Adressen und Entitäten mit höherer Wahrscheinlichkeit hin. Konkret ließe sich also ein anderer Community-Detection Algorithmus verwenden, welcher Mehrfachkanten in einer Community berücksichtigt.

Weiterhin ließe sich berücksichtigen, ob bei einer Transaktion eine übermäßig hohe Transaktionsgebühr anfällt. Das wäre ein Indiz dafür, dass ein Wechselgeldoutput in der Transaktion lediglich vergessen wurde. Dies tritt besonders häufig bei Transaktionen auf, die manuell erstellt wurden. Bei der Verwendung von Walletsoftware kommt ein solcher Fehler eher selten vor, da sie von selbst Wechselgeldadressen generieren und vorschlagen. [6, S. 129 f.] Es stellt sich die Frage, welche Transaktionsgebühr erforderlich ist, um die Transaktion für die Community-Detection Heuristik zu sperren. Eine erhöhte Transaktionsgebühr kann auch eine hohe Priorität einer Transaktion bedeuten. [6, S. 127]

Es muss also abgewogen werden, wie hoch die Transaktionsgebühr im Vergleich zu den typischen Gebühren für hochprioritäre Transaktionen sein muss, damit sie als Transaktion mit vergessener Wechselgeldadresse gilt. Es ist wichtig, dass die Transaktionsgebühr zum Zeitpunkt der Transaktion angemessen ist. Die üblichen Transaktionsgebühren schwanken historisch und tageszeitlich. Das hängt mit Angebot und Nachfrage nach Blockplatz beziehungsweise mit dem Transaktionsaufkommen zu einem gegebenen Zeitpunkt zusammen. [29]

Je niedriger diese Grenze angesetzt wird, desto wahrscheinlicher ist es, Transaktionen mit vergessenem Wechselgeldoutput zu finden. Dadurch steigt die Erkennungsrate. Parallel steigt jedoch die Anzahl der fälschlicherweise erkannten Transaktionen und die Präzision sinkt.

4.2 Wechselgeldheuristik

Eine weitere Heuristik für Adressclustering auf Basis der Transaktionseigenschaften ist die sogenannte Wechselgeldheuristik. Wie in den technischen Grundlagen dargestellt, lassen sich Werte (UTXOs) nur vollständig übertragen. Hat eine Entität A nur UTXOs mit hohem Wert, möchte aber nur einen kleineren Wert an Entität B übertragen, kann A sich das Wechselgeld an eine von ihr selbst kontrollierte Adresse senden. Teilweise lassen sich die Wechselgeldadressen sehr einfach zuordnen. Bei Fällen, in denen die Wechseladresse nicht offensichtlich ist, kann es helfen, den folgenden Indizien und Hinweisen zu folgen.

4.2.1 Allgemeine Wechselgeldheuristik

Technische Grundlage

Die Grundannahme der Wechselgeldheuristik besagt, dass in einer Transaktion, bei der mindestens 2 Outputs bestehen, einer der Outputs eine Wechselgeldadresse darstellen kann. [30, S. 41] Eine beispielhafte Darstellung einer solchen Transaktion ist zu sehen in Abbildung 4.6. Einer der beiden grün markierten Transaktionsoutputs gehört nach Annahme der Wechselgeldheuristik mit dem Transaktionsinput (blau) einer gemeinsamen Entität an.

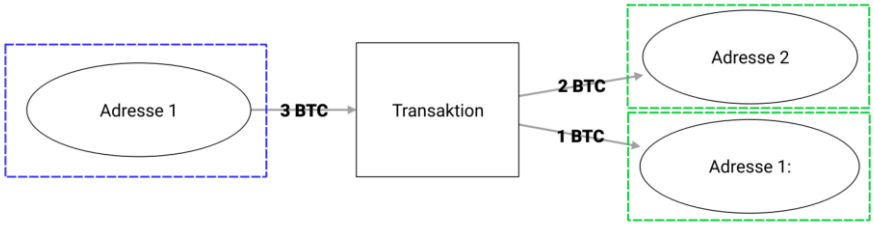


Abbildung 4.6: Graphische Darstellung der allgemeinen Wechselgeldheuristik.

Eine Zuordnung des Wechselgeldoutputs ist hier ohne weitere Informationen nicht möglich. Nehmen wir die Parteien der Transaktion als Alice (Sender) und Bob (Empfänger) an. Es ist möglich, dass Alice 2 BTC an Bob sendet und 1 BTC als Wechselgeld erhält. Andererseits wäre möglich, dass Alice 1 BTC an Bob sendet und 2 BTC als Wechselgeld erhält. Um die Wechselgeldadresse zuzuordnen, bedarf es weiterer Indizien oder Hinweisen.

Es existieren verschiedene Arten von Hinweisen, die eine Wechselgeldoutputerkennung ermöglichen. Universell anwendbare Wechselgeldhinweise nutzen ausschließlich Eigenschaften der Transaktion selbst, um den Wechselgeldoutput zu bestimmen. Dazu gehören die verwendeten Adressen, die Werte der Transaktionsinputs und -outputs oder auch die verwendeten Transaktionstypen. [31, S. 8]

Nicht-universell anwendbare Wechselgeldhinweise überprüfen nicht nur die Eigenschaften der zu untersuchenden Transaktion selbst. Sie untersuchen zusätzlich die Transaktionseigenschaften von Folgetransaktionen, welche die Outputs der zu untersuchenden Transaktion ausgeben. [31, S. 8]

Anders ausgedrückt: Universell anwendbare Wechselgeldhinweise lassen sich unmittelbar auf alle Transaktionen anwenden, bei denen mindestens 2 Outputs existieren. Nicht-universell anwendbare Wechselgeldhinweise lassen sich erst dann anwenden, wenn die Outputs einer zu untersuchenden Transaktion in einer anderen Transaktion als Inputs verwendet wurden.

Um den Wechselgeldoutput mit möglichst hoher Sicherheit zu bestimmen, verwenden Moeser u.A. mehrere Hinweise in Kombination. Die einzelnen Wechselgeldhinweise geben den Output an, den sie als Wechselgeldoutput identifizieren würden. Wenn eine Mehrheit der Hinweise den gleichen Output als Wechselgeldoutput identifiziert, ist dieser mit hoher Warscheinlichkeit auch der tatsächliche Wechselgeldoutput. Noch einmal steigern lässt sich diese Wahrscheinlichkeit mit einem Schwellenwert. Anders ausgedrückt: Ein Output wird als Wechselgeldoutput angenommen, wenn ein Output mindestens X Stimmen mehr auf sich vereint als alle anderen. Je höher X gewählt wird, desto seltener treten falschpositive Wechselgeldoutputs auf. [31, S. 10]

Ein häufig auftretendes Phänomen ist das der sogenannten *Peeling-Chain*. Bei einer Peeling-Chain startet eine Adresse mit einem hochwertigen UTXO. Der Eigentümer dieser Adresse verwendet diesen hohen Wert, um eine Transaktion auszuführen, und schickt den Wertüberschuss zurück an eine von ihm selbst kontrollierte Adresse. Teilweise sogar an die Inputadresse wie in Abbildung 4.7 dargestellt.

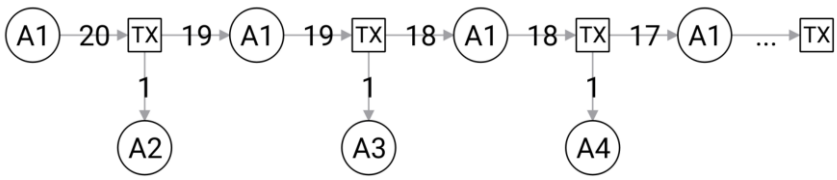


Abbildung 4.7: Peeling-Chain Phänomen

Anders ausgedrückt: Eine Entität gibt immer wieder Teile von einem großen Wert ab, indem sie einen kleinen Wert als Zahlung verschickt und den Großteil als Wechselgeld zurückerhält. Dieser Vorgang kann solange wiederholt werden, bis der große Ausgangswert auf einen kleinen Wert geschrumpft ist. Mehrere kleine Werte können anschließend in einer Transaktion kombiniert werden, um wieder einen großen Output zu generieren, und der Prozess startet erneut. [32, S. 9] Besonders leicht zu erkennen ist eine solche Peeling-Chain, wenn die Wechselgeldadresse gleich der Inputadresse ist.

Ausnahmen und Fehler

Die Indizien und Hinweise, die in den Unterkapiteln 4.2.2 bis 4.2.4 dargestellt werden, sind nicht fehlerfrei und können falschpositive Ergebnisse liefern. Da diese Fehler aber indizspezifisch sind, werden diese in den entsprechenden Abschnitten aufgezeigt und erläutert.

Es existieren jedoch auch allgemeine Fehler der Wechselgeldheuristik. Die Grundannahme, dass bei multiplen Outputs eine Wechselgeldadresse besteht, ist nicht immer erfüllt. Wie schon in den Verbesserungsvorschlägen zur Community-Detection Heuristik aufgezeigt, kommt es vor, dass Wechselgeldoutputs lediglich vergessen werden. Abgesehen davon gibt es Transaktionen, bei denen allgemein kein Wechselgeld benötigt wird. Es ist möglich, dass ein Wechselgeldoutput nicht erforderlich oder nicht ökonomisch ist, weil das anfallende Wechselgeld niedriger ist als die für den zusätzlichen Output benötigten Transaktionsgebühren. Solche Restbeträge werden auch als *Dust* bezeichnet. [33]

Weiterhin ist es möglich, dass mehrere Outputs einer Transaktion Wechselgeldoutputs sind. In verschiedenen Implementationen wird unterschiedlich damit umgegangen. Einige Umsetzungen der Wechselgeldheuristik nehmen allgemein an, dass in einer Transaktion nur ein Wechselgeldoutput existiert. Werden dennoch mehrere Kandidaten für Wechselgeldoutputs gefunden, wird keiner der Outputs als Wechselgeld angenommen. Diesem Prinzip folgen sowohl Möser und Narayanan als auch Meiklejohn u.A. in ihren Untersuchungen. [31, S. 8], [32, S. 6]

Wird die Heuristik statisch implementiert, sodass sie maximal einen Wechselgeldoutput erkennen darf, aber mindestens einen erkennen muss, bedeutet das Folgendes: Selbst wenn die Heuristik einen richtigen Wechselgeldoutput erkennt (korrektpositiv), kann sie falschnegative Ergebnisse nicht ausschließen. Andersherum: Bei einer fehlerhaften Erkennung des Wechselgeldoutputs macht die Heuristik nicht nur den dementsprechenden falschpositiven Fehler. Sofern ein Wechselgeldoutput existiert, erkennt sie den korrekten Wechselgeldoutput nicht, und produziert damit zusätzlich einen falschnegativen Fehler.

4.2.2 Universell anwendbare Wechselgeldhinweise

Self-Change und Wechselgeld anhand bestehender Cluster

Bei manchen Transaktionen sind die Wechselgeldoutputs eindeutig ersichtlich. Zum Beispiel wenn eine Entität eine von ihr bereits verwendete Adresse als Wechselgeldadresse verwendet. [31, S. 16] [32, S. 6] Ein Spezialfall, der sogenannte *Self-Change*, liegt vor, wenn die Adresse, von der ein Input stammt, gleichzeitig als Output verwendet wird. [32, S. 6 f.] Auch Outputadressen, die schon dem Inputadresscluster angehören, sind als Wechselgeldoutput wahrscheinlich. Diese Fälle sind jedoch eher selten, da viele gängige Wallets sich daran halten, Adressen nicht wiederzuverwenden. [31, S. 16]

Dieser Indikator für einen Wechselgeldoutput ist eindeutig. Er ist jedoch nur zur Erkennung des Wechselgeldoutputs geeignet. Es besteht kein Vorteil für das Adressclustering, da keine Adresse verwendet wird, die dem Adresscluster nicht bereits angehört.

Wie bereits im Abschnitt zur Multi-Input Heuristik dargestellt, konzentriert sich die Wachstumsverteilung von Adressclustern sehr stark auf das Wachstum um kleine Cluster oder einzelne Adressen. [22, S. 8–11] Diese Feststellung von Harrigan und Fretter bezieht sich nicht nur auf das Wachstum der Cluster bei Anwendung der Multi-Input Heuristik. Outputadressen, die bereits zu einem anderen großen Cluster gehören, sind als Wechselgeldadressen sehr unwahrscheinlich.

Ein Beispiel: Aufgrund der Multi-Input Heuristik konnten der Entität Alice (C1) die Adressen A1-A25 und der Entität Bob(C2) die Adressen A26-A50 zugeordnet werden. Bei einer Transaktion von Alice an Bob, wie der in Abbildung 4.8, kann eine Adresse als Wechselgeld mit hoher Wahrscheinlichkeit ausgeschlossen werden.

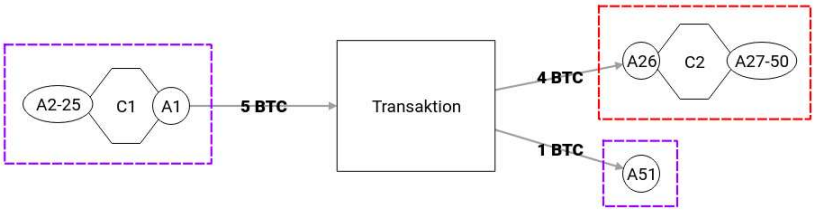


Abbildung 4.8: Wechselgelderkennung bei Transaktionen zu anderen Adressclustern

Würde man A26 als Wechselgeldadresse ansehen, müsste man die Cluster C1 und C2 miteinander verbinden und als einer gemeinsamen Entität zugehörig ansehen. Das Clusterwachstum in diesem Beispiel würde 25 betragen. Viel wahrscheinlicher ist jedoch, dass A51, die noch keinem Cluster angehört, die Wechselgeldadresse ist. Das Clusterwachstum hier entspricht nur 1.

Shadow-Address - Wechselgeldoutputs bei neuen Adressen

Von den meisten Wallets wird die Wechselgeldadresse neu erstellt. Solche, explizit für Wechselgeld erstellte Adressen werden häufig auch als *Shadow-Address* bezeichnet. [30, S. 41] [32, S. 6] Neu generierte Adressen sind solche, die in der Transaktionshistorie vor der zu untersuchenden Transaktion noch nicht aufgetreten sind. Die Aussage, die anhand von neu generierten bzw. alten, bereits aufgetretenen Outputadressen getroffen werden kann, ist die Folgende: Bei einer Transaktion mit Outputadressen A und B, wobei A bereits in der Bitcoinhistorie vorkam, B jedoch nicht, kann angenommen werden, dass B eine Shadow-Address ist und damit als Wechselgeldoutput in Frage kommt. [30, S. 41]

Die Annahme dieses Indikators ist jedoch weder für jede Transaktion anwendbar, noch ist sie immer korrekt. Angenommen: Alice möchte Bob 2 BTC senden. Dafür erstellt Bob eine neue Adresse, die bisher nicht in der Bitcoinhistorie vorkommt, und teilt diese Adresse Alice mit. Alice sendet 2 BTC an Bobs Adresse und ihr Wechselgeld an eine von ihr selbst neu generierte Shadow-Address. Bei einer solchen Transaktion kann dieser Indikator keine Aussage treffen, da beide Adressen zuvor noch nie verwendet wurden. Jetzt sei angenommen, dass Alice ihr Wechselgeld an eine ihrer alten Adressen sendet. In diesem Fall geht dieser Wechselgeldindikator vom falschen Output als Wechselgeld aus.

Optimale Wechselgeldheuristik

Die optimale Wechselgeldheuristik bezieht die Transaktionsinputs in die Wechselgeldanalyse mit ein. Sie geht davon aus, dass in einer Transaktion keine nicht benötigten Inputs verwendet werden. [34, S. 6] Dies ist der zweite Typ nicht benötigter Inputs. Dieser ist nicht mit dem ersten Typ zur Erkennung von PayJoin-Transaktionen gleichzusetzen.

Ein Wallet stellt die Inputs einer Transaktion genau so zusammen, dass der Wert, der übertragen werden soll, abgedeckt ist und der überschüssige Wert als Wechselgeld zurückerhalten werden kann. Das bedeutet: Der Wert des Wechselgeldoutputs einer Transaktion sollte vom Wert niedriger sein als der wertgeringste Input. Andernfalls hätte der wertgeringste Input weggelassen werden können und das Wechselgeld um dessen Wert reduziert. [34, S. 6]

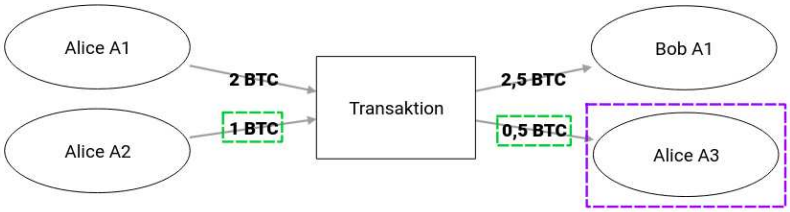


Abbildung 4.9: Optimale Wechselgeldheuristik

In der beispielhaften Transaktion, dargestellt in Abbildung 4.9, möchte Alice 2,5 BTC an Bob übertragen. Dazu benutzt sie 2 UTXOs der Werte 2 BTC und 1 BTC und verwendet diese als Input. Ihr Wechselgeld entspricht also einem Output von 0,5 BTC. Denkt man sich diesen Wechselgeldoutput weg, sind dennoch alle Inputs für die Wertstellung erforderlich. Angenommen, sie wolle nur 0,5 Bitcoin an Bob senden, wäre dafür nur der Input von 1 BTC nötig gewesen. Die zusätzliche Größe der Transaktion würde höhere Kosten für Transaktionsgebühren bedeuten. Daher ist davon auszugehen, dass der 2,5 BTC Output der Zahlung und der 0,5 BTC Output dem Wechselgeld entspricht.

Zusammengefasst besagt die Annahme der optimalen Wechselgeldheuristik also: Der Output einer Transaktion kann dann als Wechselgeldoutput angenommen werden, wenn er geringer ist als der geringste Input einer Transaktion.

Dieser Indikator ist nicht immer anwendbar. Hat eine Transaktion nur einen Input, ist dieser Hinweis trivial, da alle Outputs wertgeringer oder maximal wertgleich zum Input sein können. [35] Entsprechend der Annahme ist dieser Indikator auch nur dann geeignet, wenn es einen Output gibt, der kleiner ist als der geringste Input. Gleichzeitig besagt dieser Hinweis nur, dass der Wechselgeldoutput niedriger sein muss als der geringste Input. Das bedeutet aber nicht, dass es keinen noch geringeren Output geben kann.

Im eben genannten Beispiel existiert nur ein Output, der geringer ist als der niedrigste Input. Aber was geschieht, wenn Alice nicht nur 2,5 BTC an Bob, sondern auch 0,2 BTC an Carol überweisen möchte. Das Wechselgeld entspricht in dem Fall 0,3 BTC. Bei gleichbleibenden Inputs entfällt ein Wechselgeld von 0,3 BTC und es ergibt sich die Transaktion, abgebildet in Abbildung 4.10.

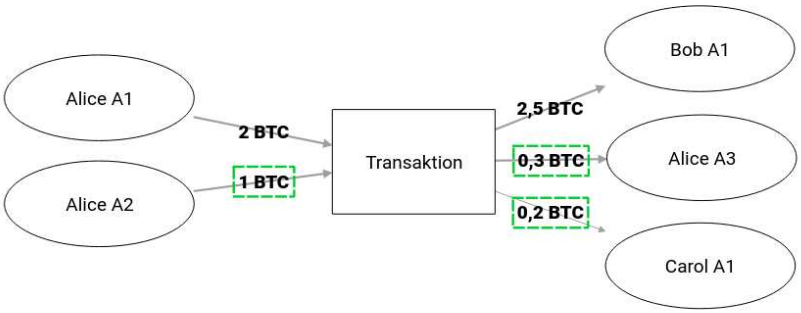


Abbildung 4.10: Mehrere Wechselgeldkandidaten bei Optimaler Wechselgeldheuristik.

In diesem Fall gibt es mehrere Transaktionsoutputs, die geringer sind als der kleinste Transaktionsinput. Es resultiert eine Transaktion, bei der es mehrere Wechselgeldkandidaten gibt, beziehungsweise eine eindeutige Zuordnung des Wechselgeldoutputs nicht möglich ist.

In einer Untersuchung von Ghesmati u.A. wird aufgezeigt, dass es verschiedene Implementationsvariationen dieses Indikators gibt. [28, S. 4] In der Implementation von BlockSci wird angenommen, dass, wenn ein Output geringer ist als jeder Input, es mit hoher Wahrscheinlichkeit ein Wechselgeldoutput ist. Dass auch mehrere Outputs geringer sein können, wird berücksichtigt. In diesem Fall gibt die Funktion mehrere Wechselgeldkandidaten zurück. Gleichzeitig bietet diese Implementation die Möglichkeit, nur dann einen Wechselgeldoutput vorzuschlagen, wenn auch tatsächlich nur einer existiert. [35] Wie die anderen von Ghesmati u.A. zitierten Implementationen mit mehreren Wechselgeldkandidaten umgehen, lässt sich nicht prüfen, da die Quellenangaben nicht mehr erreichbar sind.

Die Annahme des Indikators basiert darauf, dass ein Wallet eine Transaktion möglichst effizient und klein gestaltet, um dem Nutzer Transaktionsgebühren zu sparen. Ähnlich der Argumentation bei nicht benötigten Inputs des ersten Typs (siehe Kapitel 4.1.2) ist auch hier das künstliche Vergrößern der Transaktion, um diesen Indikator zu umgehen, nicht auszuschließen. Anders formuliert: Eine Entität, die sich der Anwendung dieses Indikators bewusst ist, kann absichtlich die In- und Outputs der Transaktion so wählen, dass der Indikator einen falschen Output als Wechselgeldoutput erkennt. Damit ließe sich dieser Hinweis zu einem falschpositiven Ergebnis zwingen.

Reicht es der Entität, seinen tatsächlichen Output zu verschleiern, erzeugt sie 2 Wechselgeldoutputs. Wie bereits in der Fehlerbetrachtung der allgemeinen Wechselgeldheuristik dargelegt: Viele Analysen verwerfen die Suche nach dem Wechselgeldoutput, wenn mehrere Kandidaten existieren.

Wechselgelderkennung bei verschiedenen Adresstypen

Ein weiterer Indikator für Wechselgeldadressen können die Adresstypen der Outputs sein. Wie in den technischen Grundlagen dargelegt, lassen sich verschiedene Adresstypen für Transaktionen verwenden. Sie beschreiben dabei, was getan werden muss, um einen auf diese Adresse übertragenen Wert nutzbar zu machen.

Die Grundidee dieses Idikators ist die folgende: Eine Entität, die häufig Transaktionen über Adressen des gleichen Typs empfängt, verwendet wahrscheinlich den gleichen Adresstyp für seine Wechselgeldoutputs. [31, S. 9] Es sollte also angenommen werden, dass ein Output, der den gleichen Adresstyp wie die Inputs verwendet, dem Wechselgeldoutput entspricht. Diese Annahme stützt sich auf typische Walletsoftware, die für Transaktionen häufig gleiche Adresstypen verwendet. In der aktuellen Version des Electrum Wallet (4.4.6) [36] zum Beispiel werden bei Erstellung eines neuen Wallets 30 P2WPKH-Adressen generiert. Diese sollen für den Empfang von Transaktionen und Wechselgeld des Nutzers verwendet werden.

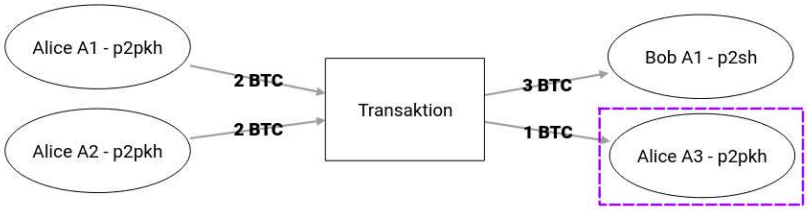


Abbildung 4.11: Wechselgelderkennung anhand unterschiedlicher Adresstypen.

Ein Beispiel: Alice hat alle ihre bisherigen UTXOs über Pay-to-Public-Key-Hash (P2PKH) erhalten. In einer Überweisung an Bob über 3 BTC nutzt sie 2 dieser UTXOs der Werte 2 BTC. Bob möchte seine Bitcoins per Pay-to-Script-Hash (P2SH) erhalten. Alice erstellt also eine Transaktion mit einem P2SH-Output (3 BTC) für Bob und einem P2PKH-Output (1 BTC) für ihr Wechselgeld. Diese Transaktion ist abgebildet in Abbildung 4.11.

Dieser Indikator ist entsprechend der Annahme nicht immer anwendbar. Er ist nur dann einsetzbar, wenn bei einer Transaktion:

1. die Inputadressen vom gleichen Typ sind.

2. genau eine Outputadresse, dem Inputadresstyp entspricht.

Ist eine dieser Bedingungen nicht zutreffend, kann dieser Indikator nicht verwendet werden. Bei typverschiedenen Inputadressen ist nicht eindeutig, welcher Adresstyp als Wechselgeldoutput zu erwarten ist. Existiert keine Outputadresse, die dem Adresstypen der Inputs entspricht, gibt es keinen Wechselgeldkandidaten. Existieren mehrere Outputadressen, die dem Adresstypen der Inputs entsprechen, gibt es mehrere Wechselgeldkandidaten.

Eine Entität kann sich der Identifikation seiner Wechselgeldoutputs entziehen oder den Indikator zu falschen Ergebnissen zwingen. Eine Identifikation des Wechselgeldoutputs lässt sich durch die Verwendung von verschiedenen Adresstypen vermeiden. Es ist auch möglich, mehrere Wechselgeldoutputs zu erstellen. Bei mehreren Kandidaten sollte auch bei diesem Indikator die Wechselgelderkennung abgebrochen werden. Man zwingt den Indikator zu einem falschpositiven Ergebnis, wenn der Zieloutput dem gleichen Adresstyp der Inputadressen entspricht, der Wechselgeldoutput jedoch einen anderen Adresstyp verwendet. [31, S. 9]

Wechselgelderkennung anhand von Transaktionswerten

Wechselgeldoutputs lassen sich ebenfalls anhand ihrer Transaktionswerte identifizieren. Kaufbeträge sind allgemein eher runde Beträge. Das bedeutet: Der Betrag hat viele nachgestellte Nullen oder liegt in der Nähe eines solchen Betrages. Diese Eigenschaft macht damit die Identifikation von Zahlungsoutputs möglich. Indirekt lassen sich so aber auch Wechselgeldoutputs identifizieren. [31, S. 9]

Diese Eigenschaft ist nicht nur anwendbar auf die Werte dargestellt in Bitcoin. Sie lässt sich übertragen in andere Währungen wie US-Dollar oder Euro. Dabei ist jedoch der Zeitpunkt der Transaktion entscheidend, da der Bitcoinkurs ebenso wie Währungen relativ zueinander stark schwanken können. [37] Während ein Bitcoinwert von 0,077163 BTC nicht als runder Betrag erscheint, bedeutete er zum Transaktionszeitpunkt möglicherweise einen Wert von 3000 Euro.

Dieser Indikator kann in verschiedenen Stufen redundant verwendet werden. Das bedeutet: Ein Output wird als Zahlung aufgefasst, wenn eine gewisse Anzahl von nachgestellten Nullen vorhanden ist. In der Untersuchung von Möser und Narayanan wurden 6 Dezimalstufen (2-7 nachgestellte Nullen) definiert. Die runden Werte wurden dabei ausschließlich in der Währung Bitcoin betrachtet. [31, S. 24]

Die Übertragbarkeit dieses Indikators in verschiedene Währungen kann unter Umständen zu falschpositiven Ergebnissen führen. In einigen Fällen können bei Transaktionen je nach Währungsperspektive gerade und ungerade Werte wechseln. Aus der Bitcoin-Perspektive könnte Output A gerade und Output B ungerade sein, aus der Euro-Perspektive wechselt jedoch diese Eigenschaft, sodass Output A ungerade und Output B gerade wird.

Außerdem kann dieser Indikator durch die variable Verwendung von Transaktionsgebühren umgangen werden. Angenommen, Alice hat eine UTXO vom Wert 1,55 BTC und möchte 1,2 BTC an Bob senden. Um den Wechselgeldoutput nicht durch diesen Indikator sichtbar zu machen, sendet sie ihrer Wechselgeldadresse nur 0,3 BTC. Die restlichen 0,05 BTC werden vom Bitcoinnetzwerk implizit als Transaktionsgebühr aufgefasst. Eine solche Vorgehensweise ließe sich durch die möglicherweise deutlich erhöhte Transaktionsgebühr identifizieren. Eine Zuordnung des Zahlungs- und Wechselgeldoutputs bleibt jedoch ausgeschlossen.

Eine andere Möglichkeit, anhand von Beträgen einer Transaktion den Wechselgeldoutput zu identifizieren, ist das sogenannte *Fee Bumping*. Es ist möglich, dass Transaktionen vom Bitcoinnetzwerk nicht verarbeitet werden, weil sie eine zu geringe Transaktionsgebühr bereitstellen. In dieser Situation kann sich der Sender dieser Transaktion dafür entscheiden, die Transaktionsgebühr seiner Transaktion zu erhöhen, damit ein Miner die Transaktion verarbeitet. [38] Die Erhöhung der Transaktionsgebühren lässt sich unter anderem durch die Reduktion der Outputwerte erreichen. Da der Zahlungswert meist nicht verringert werden kann, muss der Wechselgeldwert reduziert werden. Eine Entität, die eine Transaktion erst mit geringer und später mit erhöhter Transaktionsgebühr beobachtet, kann den Wechselgeldoutput durch die Wertveränderung identifizieren. Ausnahmefälle dieses Wechselgeldindikators entstehen, wenn das Wechselgeld durch die Reduktion von Zahlungs- und Wechselgeldoutput entsteht (Aufteilung der Transaktionsgebühren auf Sender und Empfänger), oder die Outputadressen verändert werden. [39]

4.2.3 Gleichartige Weiterverwendung von Transaktionsoutputs - Fingerprinting

Eine Form von nicht-universell anwendbaren Wechselgeldhinweisen sind Hinweise auf Basis der Weiterverwendung von Transaktionsoutputs. Bei dieser Art von Wechselgeldhinweisen werden die Eigenschaften einer Transaktion und deren Folgetransaktionen beobachtet. Verschiedene Eigenschaften einer Transaktion machen den Fingerabdruck einer Transaktion aus. Dieser Fingerabdruck wird dann mit den Fingerabdrücken der Folgetransaktionen verglichen, welche die Outputs der zu untersuchenden Transaktion ausgeben. [31, S. 9]

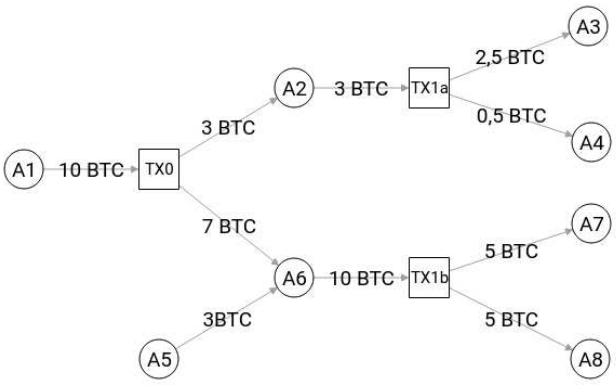


Abbildung 4.12: Wechselgelderkennung durch gleichartige Transaktionen.

Abbildung 4.12 zeigt die zu untersuchende Transaktion TX0. Es soll herausgefunden werden, welcher der beiden Outputs von TX0 der Wechselgeldoutput ist. Dazu wird untersucht, ob die Transaktionen TX1a bzw. TX1b sich in ihren Transaktionseigenschaften TX0 ähneln. Ähnelt eine der Transaktionen TX1* der Transaktion TX0 sehr stark und die andere ist sehr verschieden, kann das auf den Wechselgeldoutput hinweisen. Angenommen TX1a ähnelt der Transaktion TX0. Dann ist der Output von TX0, der in TX1a als Input verwendet wird, mit hoher Wahrscheinlichkeit der Wechselgeldoutput von TX0.

Es wird also überprüft, ob der Output einer Transaktion in ähnlicher Art und Weise weiterverwendet wird. Dazu lassen sich verschiedene Eigenschaften von Transaktionen verwenden. [31, S. 8] Teilweise werden auch Eigenschaften verwendet, die schon im Kapitel zu universell anwendbaren Wechselgeldhinweisen aufgegriffen wurden. In den folgenden Erläuterungen dieser Eigenschaften steht TX0 für die zu untersuchende Transaktion, und TX1 steht für eine Transaktion, die einen Output von TX0 als Input verwendet.

Die Gleichheit dieser Eigenschaften kann unter anderem dann entstehen, wenn die Transaktionen von der gleichen Walletsoftware kreiert wurden. Dass die Identifizierung einer Clientsoftware möglicherweise auch die Unterscheidung zwischen Zahlung und Wechselgeld ermöglichen könnte, wurde bereits von Reid und Harrigan festgestellt. [20, S. 220] Sie stellen in ihrer Untersuchung jedoch keine Merkmale fest, anhand derer eine Walletsoftware erkennbar wäre.

Ausnahmen und Fehler sind auch bei dieser Methode nicht auszuschließen. So können Softwareupdates von Wallet-Software zu grundlegend anderen Transaktionseigenschaften führen. Außerdem ist es möglich, dass eine Entität die Walletsoftware wechselt und so den Fingerabdruck verändert. [31, S. 9] Ebenfalls kann eine Entität Transaktionen jederzeit manuell erstellen und damit ein Fingerprinting der Transaktion erheblich erschweren.

Anzahl der Transaktionsin- und outputs

Die Anzahl der Transaktionsinputs und -outputs kann auf das Verhalten einer Wallet-Software hinweisen. Die Anzahl der Inputs richtet sich danach, welche UTXOs einer Entität zu Verfügung stehen. Jedoch weisen Muster wie Peeling-Chains eine gleichbleibende Anzahl von Inputs und Outputs auf. [31, S. 21–23] Ein Transaktionsmuster wie eine Peeling-Chain kann auf einen Wallet und damit auf eine gemeinsame Entität hinweisen.

Andersherum kann ein Transaktionsmuster in TX1 einen Wechselgeldoutput von TX0 mit hoher Wahrscheinlichkeit ausschließen. Wird zum Beispiel ein Output von TX0 in einer Batching-Transaktion verwendet, ist dieser wahrscheinlich kein Wechselgeldoutput. [39]

Reihenfolge von Transaktionsin- und outputs

Die Reihenfolge von Transaktionsinputs und -outputs ist in Bitcoin nicht vorgegeben und wird daher von verschiedenen Wallets unterschiedlich implementiert. So kann ein Wallet die UTXOs nach Empfangsdatum, ein anderes alphabetisch nach der Adresse und wieder ein anderes nach Wert sortieren. [40] Es besteht zwar die Möglichkeit, die In- und Outputs zu permutieren, sie ist aber nicht von allen Wallets implementiert. So werden in manchen Wallets die Inputs geordnet oder die Wechselgeldoutputs allgemein als letztes angeordnet. [39]

Sind also die Transaktionsinputs und -outputs in TX0 und TX1 nach dem gleichen Schema angeordnet, weist das auf den Wechselgeldoutput hin. [31, S. 23] Auch eine Permutation der In- und Outputs in TX0 und TX1 kann auf den Wechselgeldoutput hinweisen. Sind die In- und Outputs in TX0 und TX1a permutiert, in TX1b aber auffällig geordnet, ist auch hier der Wechselgeldoutput naheliegend.

Locktime

Fee-Sniping ist eine Methode von Bitcoin-Minern, nur Transaktionen mit sehr hohen Transaktionsgebühren in ihre Blöcke aufzunehmen. Entitäten, die Transaktionen in das Bitcoinnetz senden, können das verhindern, indem sie ihre Transaktionen mit einer *Locktime* versehen. Vereinfacht gesagt gibt die Locktime an, ab welchem Zeitpunkt die Transaktion in einen Block aufgenommen werden darf.

Diese Sperre kann als Blockanzahl, die bestehen muss, oder mit einem Zeitstempel angegeben werden. [6, S. 157] Diese zeitliche Sperre kann in jede Bitcointransaktion eingebaut werden, aber nicht alle Wallets ermöglichen das Setzen dieser Sperre. Wiederum andere wie z.B. Bitcoin Core [41] setzen die Sperre standardmäßig. Werden in TX0 und TX1 Locktimes verwendet, deutet das auf einen Wechselgeldoutput hin. [31, S. 23] Auch eine unterschiedliche Verwendung nach Art oder Dauer kann auf einen Wechselgeldoutput hindeuten.

Version

Durch die eben beschriebene Locktime lassen sich eigene Transaktionen bis zu einem gewissen Zeitpunkt verzögern. Weiterhin ist es möglich, Transaktionen relativ zu den verwendeten Inputs zu verzögern. Ist die Transaktionsversion auf 2 gesetzt, wird die Sequenznummer eines Inputs dafür genutzt, Transaktionen bis zu einem bestimmten Alter des ausgegebenen Outputs zu sperren. [42]

Anders ausgedrückt: Alice überweist 1 BTC an Bob. Bob kann nun diesen Input in einer Transaktion verwenden. Mit der Sequenznummer kann Bob festlegen, wie lange die Überweisung von Alice her sein muss, damit Bobs Transaktion in einen Block aufgenommen werden kann.

Auch diese Funktionalität wird von manchen Wallets standardmäßig verwendet und von anderen nicht. Setzen TX0 und TX1a die Transaktionsversion 2 TX1b aber nicht, ist das ein Indiz dafür, dass TX1a den Wechselgeldoutput von TX0 verwendet. [31, S. 23]

Opt-In Replace by Fee

Eine weitere Eigenschaft von Transaktionen, die zum Fingerprinting verwendet werden kann, ist das Nutzen der *Opt-In Replace by Fee* Funktionalität. Replace by Fee allgemein ist ein Konzept, das es ermöglicht, Transaktionen nach dem Versand in das Bitcoin Netzwerk zu verändern, bzw. zu ersetzen. Beim Opt-In Replace by fee, kündigt die Transaktion bereits an, dass sie womöglich ersetzt wird. Ermöglicht wird diese Funktionalität durch die Sequenznummer. Ein Wallet behält sich Änderungen an einer Transaktion vor, wenn es die Sequenznummer niedriger wählt, als 0xFFFFFFFF-1. In der ersetzenden Transaktion verwendet das Wallet dann eine höhere Sequenznummer und zahlt häufig eine höhere, mindestens jedoch die gleiche Transaktionsgebühr. [43] Würde das Wallet eine niedrigere Transaktionsgebühr zahlen, gäbe es für das Miner-Netzwerk keinen Grund, nicht trotzdem die ursprüngliche Transaktion zu verwenden.

Häufig wird diese Funktionalität genutzt, um Transaktionen eine höhere Transaktionsgebühr zu geben, damit sie schneller in einem Block verarbeitet werden. Manche Wallets lassen sich die Möglichkeit der Ersetzung einer Transaktion offen, andere unterstützen diese Methodik nicht. Behalten sich TX0 und TX1 eine Transaktionsersetzung vor, kann das ein Indiz für einen Wechselgeldoutput sein. [31, S. 23]

SegWit

SegWit oder Segregated Witness ist ein Transaktionskonzept, welches einige Probleme traditioneller Transaktionen löst bzw. reduziert. Unter anderem wird eine Speicherreduktion von Transaktionen ermöglicht. Konkret werden Skripte und Signaturen nicht innerhalb der Transaktionen gespeichert. Dadurch reduziert sich die Größe der einzelnen Transaktionen, und mehr Transaktionen können in einem Block untergebracht werden. [44]

Es gibt verschiedene Arten, das SegWit-Konzept zu nutzen. Einerseits gibt es native SegWit-Adressen. Diese werden analog zu den traditionellen P2WPKH und P2WSH genannt und beginnen statt mit 1 bzw. 3 mit bc1q. Auf der anderen Seite gibt es P2SH-Adressen die das SegWit-Konzept in ihrem Skript tragen. [44] Veraltete Versionen von Walletsoftware hätten es blockieren können, an native SegWit-Adressen zu zahlen, da sie nur Adressen akzeptieren, die mit 1 oder 3 beginnen. Diese P2SH-Adressen bilden also eine Brücke zwischen Entitäten, die einen Wallet nutzen, der das Konzept nicht unterstützt, und einer Entität, die Transaktionen auf einer SegWit-Adresse empfangen möchte.

Die Vorteile von Segregated Witness Transaktionen greifen erst, wenn mindestens ein Input von einer SegWit Adresse stammt. Kommt keiner der Inputs von einer SegWit-Adresse, können Signaturen bzw. Skripte nicht ausgelagert werden. [45] Ob eine Transaktion TX0 und dessen nachfolgende Transaktionen TX1, TX2... native SegWit, Brücken SegWit oder nicht SegWit-Adressen nutzt, kann Aufschluss über den Wechselgeldoutput geben. [31, S. 23]

Verwendung unbestätigter Transaktionsoutputs

Wie in den Grundlagen erwähnt, werden die Transaktionen in einem Block durch das Anhängen von weiteren Blöcken bestätigt. Es ist sogar empfohlen, 6 Bestätigungen eines Blockes abzuwarten, bevor man eine Zahlung akzeptiert, um das Risiko eines Double Spendings zu reduzieren. Mit der unmittelbaren Weiterverwendung von Transaktionsoutputs signalisiert eine Entität die Bereitschaft das Risiko eines Double Spendings eingehen zu wollen. Dieses Risiko einzugehen, ist laut Möser und Narayanan typisch für manche Intermediäre; weshalb sie in ihrer Untersuchung ein solches Verhalten einem Zahlungsoutput zuschreiben und den jeweils anderen als Wechselgeldoutput auffassen. [31, S. 23]

Für Wechselgeldoutputs ist das Warten auf eine Bestätigung der Transaktion durch die weitere Blockchainentwicklung aber ebenfalls überflüssig. Nimmt man eine Transaktion TX0 an, in der Alice an Bob 2 BTC überweist und sich ein Wechselgeld von 1 BTC zahlt, dann kann Alice den 1 BTC für eine andere Transaktion TX1 weiterverwenden. Sie muss nicht auf Bestätigungen der Transaktion warten, da nur sie selbst versuchen könnte, den Input von TX0 erneut auszugeben. Würde sie das tun, und dieser Angriff funktioniert, würde das ihre Transaktion TX1 ebenfalls ungültig machen. Vermutlich ist das auch ein Grund für die schwache Aussagekraft dieses speziellen Hinweises, beschrieben in Kapitel 5.

Absolute und relative Transaktionsgebühr

Ein Fingerprinting ist ebenfalls anhand der Transaktionsgebühren möglich. Einerseits gibt es Wallets, die für jede Transaktion die gleiche Transaktionsgebühr zahlen, unabhängig von ihrer Größe. Andere Wallets zahlen immer die gleiche Transaktionsgebühr in Relation zur Transaktionsgröße. [31, S. 23] Wiederum andere Wallets wie zum Beispiel Wasabi Wallet passen ihre relativen Transaktionsgebühren an die derzeitig typischen Transaktionsgebühren an. [46] Gleichen sich die relativen oder absoluten Höhen der Transaktionsgebühren in TX0 und TX1, kann das auf den Wechselgeldoutput hinweisen. [31, S. 23] Auch unterschiedliche dynamische Gebührenempfehlungen von Wallets könnten Rückschlüsse auf den Wechselgeldoutput zulassen, dies bedarf jedoch noch genauerer Untersuchung. [47]

Wie alle anderen Hinweise des Fingerprintings auch, kann dieser Hinweis durch Walletänderung oder ein Softwareupdate getäuscht werden. Auch eine gezielte Verschleierung durch einen fortgeschrittenen Nutzer ist nicht auszuschließen. [47] Eine Änderung am Transaktionsgebührverhalten tritt aber auch dann auf, wenn eine Entität eine Transaktion schnell verarbeitet haben möchte. [6]

Multisignatur

Wie in den Grundlagen erwähnt lassen sich mit P2SH-Adressen Multisignatur-Transaktionen realisieren. Eine Entität, die einen Multisignaturwallet nutzt, kann an jeden Adresstyp senden, der vom Wallet unterstützt wird. Einen Wechselgeldoutput wird diese Entität aber immer an eine gleichartige P2SH-Multisignaturadresse senden. Anders ausgedrückt: Der Wechselgeldoutput wird zur Ausgabe des Werts dieselbe Anzahl von Signaturen bzw. den selben Threshhold erfordern.

Angenommen, Alice und Bob haben einen Multisignatur-Wallet und senden eine Transaktion an Carol und wollen das Wechselgeld zurückerhalten. Damit Alice und Bob den Wechselgeldoutput nur gemeinsam verwenden können, müssen sie das Wechselgeld an eine P2SH-Multisignatur-Adresse senden.

Verwendet Carol eine P2PKH-Adresse, ist eine Zuordnung des Wechselgeldoutputs durch den universell anwendbaren Wechselgeldhinweis anhand verschiedener Adresstypen möglich. Verwendet sie jedoch auch eine P2SH-Adresse, funktioniert das nicht.

Was genau das Skript einer P2SH-Adresse bedeutet, erschließt sich erst bei der Weiterverwendung des Outputs. [6] Stellt sich also durch TX1 heraus, dass TX0 genau einen Multisignaturoutput des gleichen Typs hat, ist eben dieser der Wechselgeldoutput.

Adresstypen

Ein universell anwendbarer Indikator für Wechselgeldoutputs war die Erkennung anhand verschiedener Outputadresstypen. Wird an unterschiedliche Adresstypen ausgezahlt, ist ein Output, der den gleichen Adresstyp verwendet wie die Inputs, ein Wechselgeldkandidat. Dieser Indikator wird beim Fingerprinting lediglich auf TX1 und folgende Transaktionen angewendet. Verwendet also TX0 Inputs und Outputs vom gleichen Typ, kann in TX1a und TX1b geprüft werden, ob diese Outputs auch wieder an Adressen dieses Typs gezahlt werden. Verwendet eine Folgetransaktion weiterhin Outputs des gleichen Typs und die andere nicht, deutet das auf einen Wechselgeldoutput hin. [31, S. 23]

Kurze Signaturen

Abgesehen vom neuen P2TR-System werden in Bitcoin ECDSA-Signaturen in der DER Kodierung verwendet. Diese Signaturen variieren in ihrer Länge zwischen 71 und 73 Byte. [48] Ist der erste Versuch einer Signatur 73 Byte lang, kann ein Wallet durch das Ausprobieren verschiedener Nonces eine Verkürzung erreichen. Um Nutzern des BitcoinCore-Wallets Transaktionsgebühren zu sparen, schlägt Chow vor, die Signatur so oft mit unterschiedlichen Nonces zu generieren, bis eine Signatur 71 Byte groß ist. Dieser Vorschlag ist inzwischen im Bitcoin-Core Wallet implementiert. [49]

Da noch immer Transaktionen mit Signaturen der Größe 72 oder 73 Byte in das Bitcoinnetzwerk gesendet werden, wird diese Methode von manchen Wallets verwendet und von anderen nicht. Das macht diese Transaktionseigenschaft geeignet für Fingerprinting. Verwenden TX0 und TX1 Signaturen mit der verkürzten Länge, kann das auf einen Wechselgeldoutput hindeuten.

Dieser Hinweis funktioniert jedoch nur als Implikation. Das bedeutet: Alle Transaktionen, die von einem Wallet (der diesen Vorschlag implementiert) gesendet werden, haben Signaturen von 71 Byte. Aber nur weil eine Signatur 71 Byte hat, muss sie deswegen nicht von einem solchen Wallet stammen. [49] Eine Signatur kann auch zufällig 71 Byte lang sein.

4.2.4 Wechselgelderkennung auf Basis späterer Adresswiederverwendungen

Wiederverwendung einer Adresse als Input

Neben universellen und Fingerprinting-Hinweisen auf den Wechselgeldoutput kann ein Wechselgeldoutput durch einen späteren gemeinsamen Input identifiziert werden. Anders ausgedrückt: Der Wechselgeldoutput einer Transaktion kann durch die Multi-Input Heuristik identifiziert werden. [31, S. 4] Nehmen wir als Beispiel 2 Transaktionen TX1 und TX2, wie dargestellt in Abbildung 4.13 an.

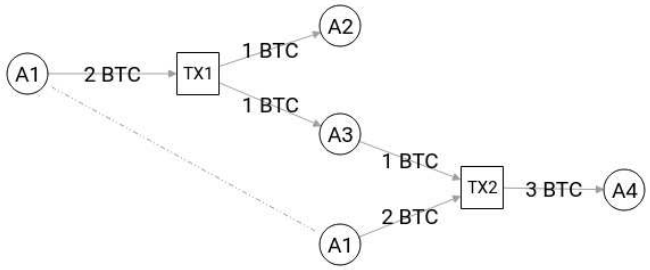


Abbildung 4.13: Wechselgelderkennung durch späteren, gemeinsamen Input.

Es soll untersucht werden, welcher der Outputs von TX1 der Wechselgeldoutput ist. Die Analyse von TX1 allein liefert kein eindeutiges Ergebnis. Betrachtet man jedoch zusätzlich TX2, ergibt sich gemäß der Multi-Input Heuristik, dass A1 und A3 der gleichen Entität angehören. Geht man mit diesem Wissen an die Analyse von TX1 heran, ist A3 als Wechselgeldoutput naheliegend.

Dieser Wechselgeldhinweis ist sehr deutlich, und liegt äußerst selten falsch. Die einzige Möglichkeit, ein falschpositives Ergebnis dieses Indikators zu provozieren, wäre ein CoinJoin. Dieser müsste außerdem mit genau den Adressen des Zahlungsoutputs und einem Transaktionsinput bestehen. Konkret am Beispiel aus Abbildung 4.13: Damit der Transaktionsoutput von TX1 falsch erkannt wird, muss TX2 eine CoinJoin-Transaktion aus dem Zahlungsoutput und einem Input von TX1 sein.

Wiederverwendung von Adressen als Output

Wie im Kapitel 4.2.2 dargelegt, gibt es einen Wechselgeldhinweis (Shadow-Address), der anhand der Neuheit einer Adresse den Wechselgeldoutput identifiziert. Die Begründung für diesen Hinweis lag darin, dass Wallets neue Adressen für ihre Wechselgeldoutputs nutzen. Ein Ausnahmefall dieses Hinweises war dann gegeben, wenn mehrere Outputadressen noch nicht aufgetreten waren. In einer Untersuchung Zhang u.A. wird angenommen, dass Shadowadressen nicht mehrfach als Outputs vorkommen, auch nicht zu einem späteren Zeitpunkt. [50]

Dementsprechend schlagen Zhang u.A. eine nachträgliche Wechselgeldidentifikation vor. Wenn alle Outputadressen einer Transaktion bis auf eine mehrfach als Output verwendet werden, kann diese als Wechselgeldadresse identifiziert werden. [50, S. 210586]

4.3 Adressclustering bei speziellen Transaktionen

Die im Folgenden beschriebenen Adressclusteringmethoden beschränken sich auf spezielle Transaktionstypen. Diese haben nur begrenzte Anwendungsfälle, und können daher Hinweise auf Adressen geben, die von der gleichen Entität kontrolliert werden.

4.3.1 Coinbase-Transaktionen

Technische Grundlage

Wie in den Grundlagen dargelegt, sind Coinbase-Transaktionen dazu da, die Blockbelohnung sowie die Transaktionsgebühren an den Miner eines Blockes auszuschütten. Der Miner eines Blockes wird also eine seiner eigenen Adressen als Empfängeradresse angeben. Wie alle anderen Transaktionen sind auch Coinbase-Transaktionen nicht in ihrer Outputanzahl beschränkt. Es ist aber anzunehmen, dass ein Miner die Blockbelohnung und die Transaktionsgebühren für sich behält. Daher sind alle Outputadressen einer Coinbase-Transaktion einer gemeinsamen Entität zuzuordnen. [51, S. 84]

Ausnahmen und Fehler

Bei einem Miningpool wird die Rechenleistung von mehreren Rechnern gebündelt, um einen Block der bestehenden Blockchain anzuhängen. Die Blockbelohnung und Transaktionsgebühren werden anschließend unter den Teilnehmern des Miningpools aufgeteilt. Die Aufteilung der Blockbelohnung wird unterschiedlich realisiert. Bei vielen Miningpools wird die Rechenleistung über mehrere Blöcke genutzt und ein Guthaben errechnet, das ab einer gewissen Höhe ausgezahlt werden kann. [32, S. 4] In einem solchen Fall gehören multiple Outputs einer Coinbase-Transaktion einer gemeinsamen Entität, nämlich dem Miningpool.

Meiklejohn u.A. stießen in ihrer Untersuchung jedoch auch auf einen Miningpool, welcher die Auszahlung der Belohungen in der Coinbase-Transaktion durchführte. [32, S. 4] In diesem Fall gehören die Outputadressen der Coinbase-Transaktion nicht einer gemeinsamen Entität, sondern den Teilnehmern dieses Miningpools.

4.3.2 Single-Output Transaktionen

Technische Grundlage

Etwa 14% aller Bitcointransaktionen haben genau einen Output. [31, S. 5 f.],[52, S. 191] Dabei können Transaktionen aus verschiedenen Gründen genau einen Output haben. Eine grobe Unterscheidung kann anhand der Inputanzahl erfolgen. So bezeichnen Gong u.A. Single-Output Transaktionen mit mehreren Inputs als *Konsolidierung* (4,6%) und mit einem Input als *Transfer* (9,2%). [52, S. 191] Kaldoner u.A. sagen, es läge bei Transfertransaktionen nahe, dass Input- und Outputadresse der gleichen Entität gehören. [53, S. 10] Für diese Clusteringmethode sprechen einige Anwendungsfälle, bei denen beide Adressen tatsächlich einer Entität angehören. Mögliche Anwendungsfälle solcher Transaktionen sind der Wechsel auf eine andere Wallet-Software oder der Wechsel von nonSegWit

zu SegWit. Auch ein Eigentumsnachweis über einer Summe Bitcoins ist durch eine Transfertransaktion realisierbar. Bei genauerer Betrachtung basiert auch die Community-Detection Heuristik auf dieser Annahme.

Ausnahmen und Fehler

Diese Clusteringmethode erfährt in der Literatur jedoch keine breite Unterstützung. So formulieren Möser u.A., dass es bei Transfertransaktionen keinen guten Indikator gäbe, der direkt und zuverlässig bestimmen könne, ob die Outputadresse der gleichen Entität angehört. [31, S. 4] Neben solchen Anwendungsfällen, bei denen die Heuristik richtig liegt, gibt es andere Anwendungsfälle, bei denen diese Heuristik falschpositive Ergebnisse liefert. Wie bereits dargelegt gibt es Transaktionen, bei denen ein Wechselgeldoutput überflüssig bzw. unökonomisch wäre oder teilweise lediglich vergessen wird.

4.4 Aktive Methoden und Angriffe

Das Adressclustering mit Hilfe der bisher beschriebenen Heuristiken erfordert keine aktive Teilnahme am Bitcoin Netzwerk. Es gibt aber auch Methoden bzw. Angriffe, die eine aktive Teilnahme am Bitcoinnetzwerk erfordern. So kann ein Angreifer mit Diensten interagieren und dadurch Adressinformation erhalten. Ein Angreifer kann Opfer außerdem dazu drängen Adressen mehrfach bzw. gemeinsam als Input einer Transaktion zu verwenden.

4.4.1 Informationsgewinn durch Handel

Technische Grundlage

Damit eine Entität eine Transaktion empfangen kann, muss sie dem Sender der Transaktion eine seiner Adressen mitteilen. Im Fall von nichtpersonellen Entitäten ist es also schwer, die Pseudonymität ihrer Adressen zu wahren. Beispielsweise können bei Einzahlungen und Abhebungen von Werten bei Kryptobörsen oder Glücksspielanbietern Adressen dieser Dienste offengelegt werden. In abgewandelter Form lassen sich so ebenfalls Informationen über Miningpools, Marktplätze oder Mixing-Dienste sammeln. [32, S. 4 f.]

Zuverlässigkeit

Die durch diese Methode gewonnene Adresszuordnung ist als sehr verlässlich anzunehmen, da es keine Möglichkeit gibt, sie zu umgehen. Aus diesem Grund eignet sich diese Methode auch zur Datensatzerzeugung. [32, S. 4]

4.4.2 Erzwungene Wiederverwendung von Adressen

Wie bereits erwähnt, wird schon im ursprünglichen Whitepaper von Satoshi Nakamoto empfohlen, für jede Bitcointransaktion eine neue Adresse zu verwenden. [5, S. 6] Eine Entität, die dieser Empfehlung nicht folgt, lässt zu, dass ihre Transaktionen miteinander in Verbindung gebracht werden können.

Taucht eine Adresse in mehreren Transaktionen auf, können alle Transaktionen, die diese Adresse enthalten, miteinander in Verbindung gebracht werden. Auch eine Entität, die der Empfehlung folgt, kann zu einer Adresswiederverwendung gezwungen oder zumindest gedrängt werden. [19, S. 92]

Technische Grundlage

Bei diesem Angriff zahlt der Angreifer einen kleinen Betrag an eine Adresse der Zielentität. Danach beobachtet der Angreifer, wie dieser Transaktionsoutput weiterverwendet wird. [19, S. 92] Wird er gemeinsam mit anderen Adressen als Input einer Transaktion verwendet, gehören diese gemäß der Multi-Input Heuristik zur gleichen Entität. Dieser Angriff funktioniert deshalb, da ein Wallet die Zahlungseingänge auf allen Adressen überwacht, auch auf solchen, die bereits durch den Nutzer verwendet wurden. In vielen Fällen werden die UTXOs für Transaktionsinputs nicht manuell durch den Nutzer, sondern vom Wallet zusammengestellt. Eignet sich der UTXO des Angreifers besser als andere UTXOs für eine Transaktion, wird er verwendet. [36],[41],[46] Aus diesem Grund fällt einem Nutzer dieser Angriff womöglich gar nicht auf.

Zu beachten ist bei diesem Angriff der Wert, der an die Zielentität gezahlt wird. Es ist abzuwägen, wie hoch bzw. niedrig er gewählt wird; je niedriger der Wert, desto günstiger ist der Angriff für den Angreifer. Wird er jedoch zu niedrig gewählt, kann es passieren, dass das Ausgeben des Outputs für das Opfer unökonomisch wird; [33] je höher der Wert, desto höher ist die Wahrscheinlichkeit, dass der Wallet den UTXO in einer Transaktion verwendet. Wird er jedoch zu hoch gewählt, kann es passieren, dass der UTXO als einziger Input einer Transaktion verwendet wird. In diesem Fall hat der Angriff keine Auswirkungen.

Zuverlässigkeit

Die Entwickler des Softwarewallets Samourai Wallet [54] warnten 2018 erstmals vor diesem Angriff und empfahlen Nutzern, solche UTXOs für die Inputzusammenstellung zu sperren. [55] Wird der Angriffs-UTXO durch den Nutzer nicht verwendet, hat der Angriff keine Auswirkungen. Möchte das Opfer des Angriffs den UTXO nutzen, sollte es den UTXO mit keiner ihrer UTXOs gemeinsam verwenden. Möglichkeiten des Opfers, den UTXO zu verwenden, ohne Informationen preiszugeben, wären: Das Spenden des vollständigen UTXOs oder ein CoinJoin mit gleichen Werten.

5 Statistische Analyse und Qualitative Bewertung von Clusteringmechanismen

5.1 Datensatzerzeugung

Um eine statistische Untersuchung zu ermöglichen, wird ein verlässlicher Datensatz benötigt, der die tatsächlichen Zugehörigkeitsverhältnisse der Adressen abdeckt. Wichtig dabei zu beachten ist, dass die Datenbasis nicht mit den Heuristiken erzeugt werden kann, deren Qualität überprüft werden soll.

Ein solcher Datensatz kann synthetisch unter Verwendung kontrollierter Testnetze und künstlich erzeugter Transaktionen generiert werden. Allerdings ist in diesem Fall eine komplexe Dynamik der Transaktionsarten wie im tatsächlichen Bitcoinnetzwerks nicht erreichbar. [52, S. 194 f.],[56, S. 8–11] Dadurch wären die gewonnenen Erkenntnisse kaum repräsentativ. Eine andere Möglichkeit, einen Datensatz zu erzeugen, ist die Verwendung von frei zugänglichen Informationen über Besitzverhältnisse von Adressen. Zusätzlich kann mit Bitcoinnutzern und Dienstleistern kooperiert werden, die bereit sind, ihre Adressen preiszugeben. Auch hier ist es nicht möglich, die Dynamik des gesamten Bitcoinnetzes abzubilden. Diese Methode kommt der Komplexität näher als ein synthetischer Datensatz, jedoch auf Kosten der Sicherheit über die Besitzverhältnisse der Adressen. [31, S. 5],[32, S. 4] Die Genauigkeit und Qualität einer Adressclusteringheuristik lässt sich also allgemein nur begrenzt statistisch nachweisen. Das liegt daran, dass auf das reale Bitcoinnetz bezogen nur selten absolut verlässliche Aussagen darüber getroffen werden können, ob Adressen tatsächlich einer Entität angehören oder nicht. [31, S. 5], [57, S. 5]

Vergleichbar ist dieses Problem mit einer Arzneimittelstudie. In der ersten beschriebenen Möglichkeit lässt sich genau sagen, wer ein Medikament und wer ein Placebo erhalten hat. Jedoch befinden sich in den Testgruppen nur Männer zwischen 20 und 25, die allgemein einen gesunden Lebensstil pflegen. Bei der zweiten beschriebenen Möglichkeit sind Männer, Frauen und Kinder in den Testgruppen vertreten, man weiß aber nicht mit Sicherheit, wer das Medikament und wer das Placebo erhalten hat.

Bei einer Untersuchung von Gong u.A. wurde die erste beschriebene Methode verwendet. Dabei wurden 300 Entitäten mit einem anfänglichen Besitz von jeweils 1000 Bitcoin simuliert. Dieser synthetische Datensatz wurde im Folgenden für die statistische Untersuchung der Multi-Input Heuristik, einer einfachen Version der Wechselgeldheuristik und deren Kombination genutzt. [52, S. 194 f.]

In einer anderen Untersuchung von Meiklejohn u.A. wurde die zweite beschriebene Methode verwendet, um so viele Adressen wie möglich verschiedenen Entitäten zuzuordnen, ohne dabei die Heuristiken zu verwenden, die überprüft werden sollten. Zusätzlich erweiterten sie ihren Datensatz mit Adressen von Dienstleistern, die sie mit der in 4.4.1 beschriebenen Methode gewinnen konnten. [32, S. 4 f.] Weiterhin wurden frei zugängliche Informationen genutzt, um diesen Datensatz zu erweitern. So wurden Entitäten Adressen zugeordnet, die sie öffentlich als ihre eigenen angaben. [32, S. 5]

Auf Basis solcher Datensätze lassen sich statistische Untersuchungen zur Qualität von verschiedenen Heuristiken durchführen. Über die Aussagekraft und Repräsentation dieser statistischen Maße auf das real existierende Bitcoinnetz lässt sich jedoch aufgrund der möglichen Ungenauigkeiten des Datensatzes streiten. Auch im Interview mit Hasse wird klar zum Ausdruck gebracht, dass eine statistische Analyse auf Basis eines solchen Datensatzes nur aussagen kann, wie gut eine Heuristik oder ein Hinweis zum vorliegenden Datensatz passt; diese statistischen Maße jedoch nicht auf die gesamte Blockchain projeziert werden können.

Ebenfalls hängt der Ansatz zur Datensatzerzeugung von der zu untersuchenden Heuristik ab. In einer Untersuchung von Möser und Narayanan wurde der Datensatz folgendermaßen erstellt. Allgemein wurden nur Transaktionen mit genau 2 Transaktionsoutputs in den Datensatz aufgenommen - mit der Grundannahme, dass einer dieser Outputs der Zahlung und der jeweils andere als Wechselgeldoutput dient. Bei dieser Transaktionsstruktur kommt es gemäß der Heuristik, beschrieben in 4.2.4, zu Offenbarungen der tatsächlichen Wechselgeldadresse. Wie bereits dargelegt, sind falschpositive Ergebnisse dieses Hinweises äußerst selten. Mit einem Datensatz, in dem nur Transaktionen mit später offenbarter Wechselgeldadresse verwendet werden, lassen sich die verschiedenen Hinweise auf Wechselgeldoutputs statistisch untersuchen. Untersuchungen zur Multi-Input Heuristik sind auf Basis des gegebenen Datensatzes nur wenig sinnvoll, weil ein großer Teil der Transaktionen nur einen Input hat. Auch bei dieser Methode lassen sich Fehler nicht gänzlich ausschließen. Die Ungenauigkeit der Datenbasis ist aber minimal [31, S. 3 f.]

5.2 Kennzahlen und Vergleichbarkeit

5.2.1 Anwendbarkeit, Effizienz und Adressreduktionsrate

Die Anwendbarkeit einer Heuristik beschreibt, bei wie vielen Fällen die Heuristik einsatzfähig ist. Während manche Heuristiken nur einen sehr spezifischen Anwendungsfall haben, können andere auf beinahe alle Bitcointransaktionen angewendet werden. Die Anwendbarkeit wird dabei in % angegeben.

Die Effizienz eines Algorithmus oder einer Heuristik beschreibt, wie aufwändig es ist, zu einem Ergebnis zu gelangen. Speziell weil manche Heuristiken die Betrachtung von mehreren Transaktionen erfordern, sind unterschiedliche Zeit- und Platzkomplexitäten der Heuristiken zu erwarten. Die Effizienz der Heuristiken ist meines Wissens noch nicht untersucht worden. Sie ist auch wenig relevant, da der wesentliche limitierende Faktor in der Datensatzerzeugung und -aufbereitung liegt und bis auf wenige Ausnahmen alle Zeit- und Platzkomplexitäten der Algorithmen linear sind.

In den Untersuchungen von Liu u.A. und Zhang u.A. wird das Maß der Adressreduktionsrate vorgeschlagen. Diese setzt die Zahl der anfänglichen Adressen (A) in Relation zu den gruppierten Clustern (C), nach Anwendung einer Heuristik. [50, S. 210588],[57, S. 5] Die entsprechende Formel zur Berechnung der Adressreduktionsrate ist:

$$Addressreductionrate = r = \frac{|A| - |C|}{|A|}$$

5.2.2 Genauigkeitsmaße

Die Qualitätskennzahlen Precision, Recall und Accuracy sind Metriken, die bei der Bewertung von Klassifikationsalgorithmen, insbesondere im Zusammenhang mit binären Klassifikationen, verwendet werden. Diese Metriken basieren auf den Begriffen True Positive (TP), True Negative (TN), False Positive (FP) und False Negative (FN), die sich aus den Ergebnissen einer Klassifikation ergeben. Die Definitionen sind wie folgt

True Positive (TP): Fälle die korrekt als positiv klassifiziert wurden.

True Negative (TN): Fälle die korrekt als negativ klassifiziert wurden.

False Positive (FP): Fälle die fälschlicherweise als positiv klassifiziert wurden.

False Negative (FN): Fälle die fälschlicherweise als negativ klassifiziert wurden.

Aus dieser Fallunterscheidung lassen sich weitere Qualitätsmaße ableiten. *Accuracy* ist die Genauigkeit eines Klassifikators, gemessen an seinen korrekten Zuordnungen in Relation zu allen getätigten Zuordnungen. Accuracy beschreibt also die Wahrscheinlichkeit einer korrekten Klassifikation. *Precision* ist die Genauigkeit, gemessen als Anteil der korrekt positiv klassifizierten Fälle in Relation zur Gesamtzahl der als positiv klassifizierten Fälle. Precision sagt also aus, wie viele der als positiv markierten Fälle tatsächlich positiv sind. *Recall(TPR)* ist der Anteil der Fälle, die korrekt als positiv vorhergesagt wurden, in Relation zur Gesamtzahl der tatsächlich positiven Fälle. Recall sagt also aus, wie viele der tatsächlich positiven Fälle erkannt wurden. *False Alarm (FPR)* ist der Anteil der Fälle, die falsch als positiv vorhergesagt wurden, in Relation zur Gesamtzahl der tatsächlich negativen Fälle. [58, S. 1–4]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}\ ,\ Precision = \frac{TP}{TP + FP}\ ,$$
$$Recall = \frac{TP}{TP + FN}\ ,\ FalseAlarm = \frac{FP}{TN + FP}$$

In diesem Kontext ebenfalls zu nennen ist der F1Score. Dieser entspricht dem harmonischen Mittel aus Precision und Recall. [58, S. 1–4]

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

5.2.3 NMI und aNMI

Normalized Mutual Information (*NMI*) ist eine Kennzahl aus der Informationstheorie. Sie beschreibt die gegenseitige Ähnlichkeit bzw. Abhängigkeit von zwei verglichenen Zuordnungen. Je größer die Ähnlichkeit zweier Zuordnungen ist, desto größer auch der NMI. Der NMI wird auf das Intervall 0;1 abgebildet. 0 bedeutet dabei, dass es keinen Zusammenhang der Zuordnungen gibt. 1 wiederum bedeutet einen eineindeutigen Zusammenhang. [21, S. 171]

Gehen wir in einem Beispiel von einem Datensatz mit Autos aus. Im ersten Abhängigkeitstest werden die Automarke und Autofarbe verglichen. 100 Autos werden also auf zwei verschiedene Arten in Cluster eingeteilt. Da diese Eigenschaften nicht stark zusammenhängen, ist der NMI eher gering. Im zweiten Abhängigkeitstest wird Automarke mit dem Preis verglichen. Zwischen diesen Eigenschaften besteht ein stärkerer Zusammenhang, speziell weil manche Automarken sich auf Autos in gewissen Preissegmenten beschränken. Der NMI ist also höher als im ersten Test. Ein NMI von 0 wird genau dann erreicht, wenn eine Eigenschaft im Datensatz immer die gleiche Ausprägung hat, unabhängig von der jeweils anderen.

Analog ist dieser Wert auch auf die Clusterzuordnung durch Heuristiken anwendbar. Hierbei kann die Abhängigkeit zwischen tatsächlicher Clusterzuordnung und der Clusterzuordnung einer Heuristik gemessen werden. Sind die Clusteringergebnisse ähnlich, ist der NMI hoch, sind sie verschieden, ist er niedrig.

Haben zwei unabhängige Zuordnungen jeweils viele Ausprägungen, kann ein hoher NMI bestehen. Um diesem Problem entgegenzuwirken, wird der Adjusted Normalized Mutual Information *aNMI* genutzt. Der aNMI berücksichtigt die Häufigkeiten der verschiedenen Ausprägungen um Zufallszusammenhänge herabzusetzen. Diese Anpassung behält die Abbildung auf das Intervall 0;1 bei, korrigiert zufällige Zusammenhänge aber nach unten. [21]

5.3 Qualität der Multi-Input Heuristik

5.3.1 Anwendbarkeit und Adressreduktionsrate

Anwendbar ist die Multi-Input Heuristik auf alle Transaktionen, die zwei oder mehr Transaktionsinputs haben. Die Häufigkeit von Multi-Input gegenüber Single-Input Transaktionen schwankt dabei nur wenig. [57, S. 7] Sie beträgt laut Gong u.A. etwa 27,2% aller Bitcointransaktionen. [52, S. 192] Liu u.A. können diese Häufigkeit, trotz eines anderen Datensatzes, bestätigen. [59]

Je nach Implementation der Multi-Input Heuristik und dem Umgang mit Coin- und PayJoin-Transaktionen lassen sich viele Adressen auf wenige Cluster reduzieren. In der Untersuchung von Liu u.A. wird die Adressreduktionsrate der Multi-Input Heuristik mit 42,6% angegeben. [57, S. 8] Im Gegensatz dazu geben Zhang u.A. die Adressreduktionsrate mit 52,05% an. [50, S. 210588]

Die Adressreduktionsrate der Community-Detection Heuristik auf Basis der Multi-Input Heuristik wurde meines Wissens noch nicht untersucht. Das Ziel der Heuristik ist es jedoch, bestehende Cluster und Adressen weiter zu aggregieren. Dementsprechend ist eine hohe Adressreduktionsrate anzunehmen. Je höher dabei die Aggregationsstufe bzw. die Rekursionstiefe gewählt wird, desto höher ist auch die Adressreduktionsrate.

5.3.2 Genauigkeit der Heuristik

Die Multi-Input Heuristik kann mit einer Precision von 0,98 als relativ sicher angenommen werden. Die Annahme, dass Adressen, die gemeinsam als Input verwendet werden, zur gleichen Entität gehören, ist also in den meisten Fällen korrekt. [21, S. 171–173] Ausnahmen stellen lediglich CoinJoin-artige

Transaktionen dar. Auch der Recall ist laut Remy u.A. mit 0,77 [21] bzw. laut Nick 0,68 [34, S. 26] relativ hoch. Das bedeutet, dass zwei Adressen, die tatsächlich der gleichen Entität angehören, sehr häufig auch zusammengeführt werden. Nach den Daten von Remy u.A. ergibt sich ein F1 Score von 0,86. [21, S. 171–173]

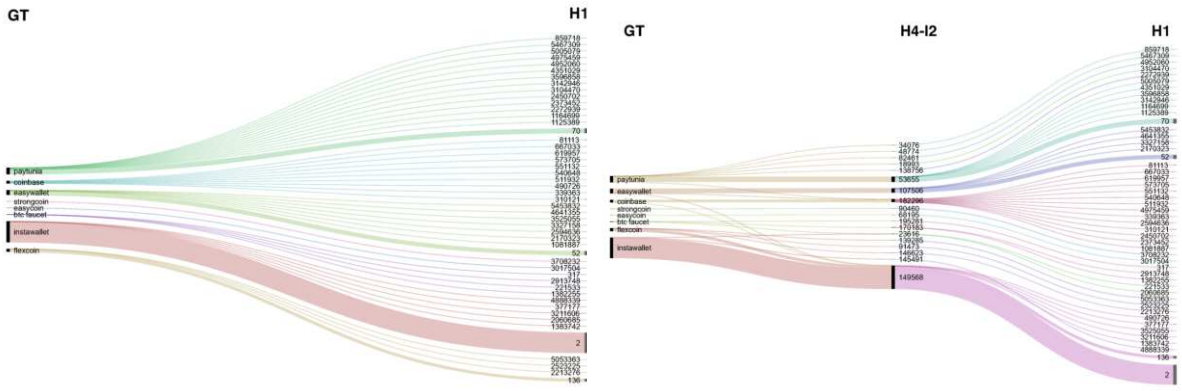
Eine Community-Detection auf Basis der Multi-Input Heuristik auszuführen, reduziert die Precision wesentlich. Sie steigert aber den Recall. Mit steigender Aggregationsstufe bzw. höherer Rekursionstiefe findet diese Heuristik immer mehr Zusammenhänge zwischen Adressen. Die korrekten Zuordnungen steigern dabei den Recall. Die falschen Zuordnungen schwächen die Precision. In Aggregationsstufe 1 sinkt die Precision von 0,98 auf 0,75. Der Recall hingegen steigt von 0,77 auf 0,79. Mit den höheren Aggregationsstufen sinkt die Precision auf 0,25. Der Recall steigt dafür auf 0,91. [21, S. 173]

5.3.3 NMI und aNMI

In der Untersuchung von Remy u.A. wird auch die Ähnlichkeit des Datensatzes von Meiklejohn [32], mit den Heuristikergebnissen untersucht. Die Multi-Input Heuristik allein erreicht einen NMI von 0,89. Durch die vielen verschiedenen Cluster ist die jeweilige Auftretenswahrscheinlichkeit eines Clusters aber gering. In Bezug auf den aNMI bedeutet das eine Absenkung auf 0,65. [21, S. 173]

In den niedrigen Aggregationsstufen der Community-Detection Heuristik sinkt zwar die Ähnlichkeit und damit der NMI. Durch die erhöhte Auftretenswahrscheinlichkeit der einzelnen Cluster kann der aNMI dennoch steigen. So ist in der niedrigsten Aggregationsstufe der NMI mit 0,86 geringer als bei der Multi-Input Heuristik, der aNMI liegt aber mit 0,66 über dem der Multi-Input Heuristik. In der nächsthöheren zweiten Aggregationsstufe sinkt der NMI weiter auf 0,81, der aNMI steigt aber auf 0,67. Bei den höheren Aggregationsstufen wird die Zahl der Cluster zwar weiter reduziert, die vielen Falschzuordnungen sorgen aber für ein Absinken von NMI und aNMI. [21, S. 173]

Die Ähnlichkeitsmaße sind visualisiert in Abbildung 5.1. GT steht dabei für die tatsächliche Zuordnung (Ground Truth), H4-l2 steht für die Community-Detection Heuristik in der Aggregationsstufe 2 und H1 für die Zuordnung durch die Multi-Input Heuristik.



(a) Multi-Input Heuristik **(b)** Community-Detection Heuristik

Abbildung 5.1: Gegenüberstellung von Multi-Input Heuristik und darauf aufbauender Community-Detection Heuristik [21, S. 174 f.]

5.4 Qualität der Wechselgeldheuristik

5.4.1 Anwendbarkeit und Adressreduktionsrate

Die Wechselgeldheuristik im Allgemeinen, also die Annahme, dass bei mehreren Outputs ein Wechselgeldoutput besteht, kann definitionsgemäß auf alle Transaktionen mit mehr als einem Output angewendet werden. Moeser und Gong geben den Anteil von Multi-Output Transaktionen bei verschiedenen Datensätzen mit 87,1% an. Ebenso stellen beide die Dominanz von Transaktionen mit genau 2 Outputs dar. Wie beim Prozentsatz der Transaktionen mit mehr als einem Input, unterliegen auch diese Werte einer eher geringen Schwankung. [52, S. 192 f.], [31, S. 6], [57, S. 7]

Die Hinweise auf einen Wechselgeldoutput sind jedoch nicht auf alle Multi-Output Transaktionen anwendbar. Ihre Anwendbarkeit bestimmt sich anhand der Kriterien, die sie zur Klassifikation nutzen. Obwohl Liu u.A. und Moeser u.A. nicht den gleichen Datensatz verwenden, sind sich ihre Ergebnisse ähnlich. In der Untersuchung von Liu u.A. und Moeser u.A. wird die Anwendbarkeit der Universellen Wechselgeldhinweise untersucht.

Anwendbarkeit	Liu [59]	Moeser [31]
Shadow-Address	28,29%	k.A
Wiederverwendung von Adressen als Output	30,62%	k.A
Adresstypen	39,53%	36,9%
Runde Beträge [n=6]	12,84%	10,4%
Optimale Wechselgeldheuristik	11,08%	13,3%

Tabelle 5.1: Ausschnitt aus Anhang B bezüglich der Anwendbarkeit universeller Wechselgeldhinweise

Die Anwendbarkeit des Hinweises der runden Beträge wurde von Moeser u.A. darüber hinaus in verschiedenen Stufen analysiert. Wie sich zeigt, sinkt die Anwendbarkeit des Hinweises, je stärker die Rundung vorausgesetzt wird. Während die Rundung von n=2, also mindestens 2 nachstehende Nullen, eine Anwendbarkeit von 38,3% aufweist, sinkt diese bis auf 4,8% bei einer Rundung von n=7. [31, S. 24]

Die Anwendbarkeit der Fingerprinting-Hinweise wurde von Moeser u.A. untersucht. Abhängig vom Hinweis variiert die Anwendbarkeit zwischen 11% und 57%. [31, S. 24] Die genauen Untersuchungsergebnisse sind zusammengefasst in Anhang B. Die höchsten Anwendbarkeiten weisen die Anzahl und Reihenfolge der Transaktionsinputs und -outputs auf.

Für die Berechnung der Adressreduktionsrate der Wechselgeldheuristik muss sie mit der Multi-Input Heuristik kombiniert werden, da die Wechselgeldheuristik allein kein Adressclustering ermöglicht. Die vom Wechselgeldhinweis identifizierte Wechselgeldadresse wird mit den Inputadressen der Transaktion geclustert. Dementsprechend kann die Adressreduktionsrate eines Wechselgeldhinweises nicht geringer sein als die der Multi-Input Heuristik allein. Zu beachten ist also die Veränderung des Wertes nach oben. Diese beschreibt nämlich den Effekt des Hinweises.

Die diesbezüglichen Untersuchungen von Liu u.A. und Zhang u.A. basieren auf unterschiedlichen Datensätzen, und es ist anzunehmen, dass sich ihre Implementation der Multi-Input Heuristik, bezüglich des Umgangs mit Coin- und PayJoin Transaktionen, unterscheiden. Aus diesem Grund unter-

scheiden sich die Adressreduktionsraten ihrer Multi-Input Heuristiken. Die Adressreduktionsrate der Multi-Input Heuristik allein wird bei Zhang u.A. mit 52,05% und bei Liu u.A. mit 42,6% angegeben. [50, S. 210588] [57, S. 8]

Adressreduktionsrate	Liu [59]	Zhang [50]
Shadowaddress	46,2% [+3,6]	53,02% [+0,97]
Wiederverwendung von Adressen als Output	46,4% [+3,8]	53,23% [+1,18]
Adresstypen	46,5% [+3,9]	k.A
Runde Beträge [n=6]	43,9% [+1,3]	k.A
Optimale Wechselgeldheuristik	44,0% [+1,4]	k.A

Tabelle 5.2: Ausschnitt aus Anhang B bzgl. der Adressreduktionsrate der universellen Wechselgeldhinweise

Von allen universellen Wechselgeldhinweisen beeinflusst die Erkennung anhand von Adresstypen die Adressreduktionsrate am stärksten. Die Adressreduktionsrate der Wechselgeldheuristik wurde nach meinem Kenntnisstand bisher nur für universell anwendbare Hinweise untersucht. Für Fingerprinting-Hinweise liegen noch keine Untersuchungen vor.

5.4.2 Genauigkeit der Heuristik und ihrer Hinweise

Die Genauigkeit der Wechselgeldheuristik wird von Moeser u.A. [31] und Remy u.A. [21] näher untersucht. Sie verfolgen jedoch unterschiedliche Ansätze in ihrer Genauigkeitsbetrachtung, sodass ihre Ergebnisse nicht vergleichbar sind. Zusätzlich wird von Remy u.A. [21] lediglich die Wechselgeldheuristik auf basis des Shadow-Address Hinweises betrachtet, während Moeser u.A. [31] die Genauigkeit der einzelnen Wechselgeldhinweise untersucht.

Kombinierte Genauigkeitsbetrachtung

In der Betrachtungsweise von Remy u.A. werden die Multi-Input Heuristik und die Wechselgeldheuristik in Kombination betrachtet. Adressen, die laut Datensatz und Heuristik einer Entität gehören, werden als True-Positive aufgefasst. Dementsprechend kann die Kombination aus Multi-Input und Wechselgeldheuristik keinen geringeren Recall aufweisen als die Multi-Input Heuristik allein. [21, S. 171–173]

Die Multi-Input Heuristik wird von Remy u.A. mit einem Recall von 0,77 angegeben. Die Kombination aus Multi-Input Heuristik und Wechselgelderkennung anhand des Shadow-Address Hinweises kommt auf einen Recall von 0,83. Allerdings beeinflusst die Kombination mit der Wechselgelderkennung nicht nur den Recall positiv, sondern auch die Precision in noch höherem Maße negativ. Die Multi-Input Heuristik allein kommt auf eine Precision von 0,98. In Kombination mit der Wechselgeldheuristik wird nur noch eine Precision von 0,09 erreicht. Der resultierende F1-Score sinkt damit von 0,86 auf 0,16. [21, S. 171–173]

Isolierte Genauigkeitsbetrachtung

Moeser u.A [31] betrachten die Wechselgeldhinweise isoliert von der Multi-Input Heuristik. Ein True Positive ist genau dann erreicht, wenn die Wechselgeldadresse von einem Hinweis korrekt erkannt wird. Über den gesamten Datensatz betrachtet erreichen die universellen Wechselgeldhinweise

einen Recall (TPR) zwischen 0,237 und 0,306 bei einem False Alarm (FPR) von 0,020 bis 0,031. [31, S. 24] In der Datenbasis kommen nur Transaktionen mit neuen Adressen vor, daher wird der Shadow-Address Hinweis nicht untersucht. [31, S. 9] Die exakten Werte der einzelnen Wechselgeldhinweise sind abgebildet in Tabelle 5.3.

Genauigkeit	TPR(Recall)	FPR(False Alarm)
Shadowaddress	k.A	k.A
Wiederverwendung von Adressen als Output	k.A	k.A
Adresstypen	0,237	0,031
Runde Beträge [n=6]	0,211	0,005
Optimale Wechselgeldheuristik	0,306	0,026

Tabelle 5.3: Ausschnitt aus Anhang B bezüglich der Genauigkeit der universellen Wechselgeldhinweise

Beim Wechselgeldhinweis anhand runder Beträge wurde, neben der Anwendbarkeit, auch die Genauigkeit stufenweise untersucht. Analog zur Anwendbarkeit zeigt sich, dass die TPR sowie die FPR mit steigender Rundung absinken. Während bei einer Rundung von n=2 eine TPR von 0,467 und FPR von 0,012 bestehen, sinkt diese bei einer Rundung von n=7 auf eine TPR von 0,107 und FPR von 0,001. [31, S. 24] Je mehr nachgestellte Nullen gefordert werden, desto seltener ist der Hinweis anwendbar, aber desto höher ist seine Aussagekraft.

Die Genauigkeit der Fingerprinting-Wechselgeldhinweise wird von Moeser u.A. mit den in Tabelle 5.4 abgebildeten TPR- und FPR-Werten angegeben. Die Genauigkeit der Wechselgeldhinweise anhand von Locktime, Version und Multisignatur, stechen durch ihre hohe TPR und niedrige FPR hervor. Diese genannten Wechselgeldhinweise sind also besonders aussagekräftig.

Besonders wenig Aussagekraft besitzt der Wechselgeldhinweis anhand der Verwendung unbestätigter Transaktionsoutputs. Dieser Hinweis besitzt eine relativ hohe FPR im Vergleich zu seiner TPR.

Genauigkeit	TPR(Recall)	FPR(False Alarm)
Anzahl der Transaktionsin- und Outputs	0,263	0,107
Reihenfolge der Transaktionsin- und Outputs	0,262	0,053
Locktime	0,307	0,003
Version	0,245	0,004
Opt-In Replace by Fee	0,075	0,003
SegWit	0,191	0,021
Verwendung unbestätigter Transaktionsoutputs	0,100	0,061
Absolute Transaktionsgebühr	0,117	0,025
Relative Transaktionsgebühr	0,042	0,008
Multisignatur	0,140	0,001
Adresstypen	0,294	0,023

Tabelle 5.4: Ausschnitt aus Anhang B bezüglich der Genauigkeit der Fingerprinting Wechselgeldhinweise

Die bisher dargestellte Genauigkeit der einzelnen Wechselgeldhinweise bezieht sich auf den gesamten Datensatz. Sie zeigt jedoch nicht, dass sich die Genauigkeit und damit die Aussagekraft eines Hinweises über die Zeit verändern kann. Im Interview stellt Hasse klar, dass die Hinweise

immer im zeitlichen Kontext betrachtet werden müssen. So können Hinweise in den ersten Jahren der Blockchain besonders gut funktionieren, zu einem späteren Zeitpunkt jedoch an Bedeutung und Aussagekraft verlieren.

Moeser u.A. stellen zudem fest, dass die Gruppe der universell anwendbaren Hinweise mehr und mehr an Genauigkeit verlieren, während die Fingerprinting-Hinweise immer mehr an Bedeutung gewinnen.

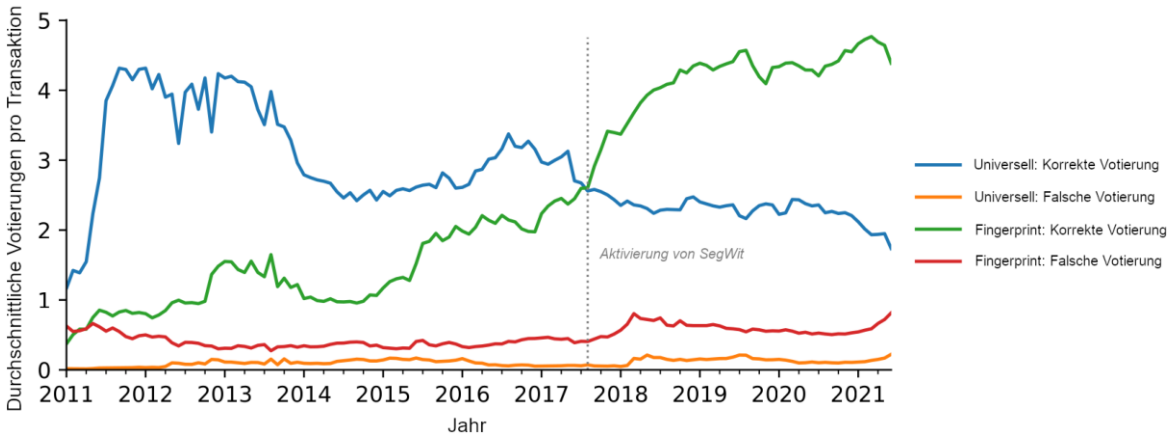


Abbildung 5.2: Veränderung der Aussagekraft von Wechselgeldhinweisen [31, S. 10]

Wie die Graphik in Abbildung 5.2 zeigt, sind die universellen Hinweise zu Beginn sehr effektiv und verlieren über die Jahre an Anwendbarkeit und Aussagekraft. Die Fingerprinting-Hinweise sind anfangs kaum anwendbar und nur selten korrekt, gewinnen aber über die Jahre immer mehr an Anwendbarkeit und Aussagekraft. Dies begründet sich durch die stetig anwachsende Zahl der beobachtbaren Transaktionsmerkmale. Ebenfalls lässt sich ein starker Anstieg der korrekten und falschen Zuordnungen der Fingerprinting Hinweise seit 2017 beobachten. Dieser ist der Verbreitung des Segregated Witness Systems geschuldet. [31, S. 8 ff.]

5.4.3 NMI und aNMI

Damit der NMI bzw. der aNMI angewendet werden kann, muss ein Adressclustering vorliegen, welches mit dem Datensatz verglichen werden kann. Auch hier muss also die Wechselgeldheuristik in Kombination mit der Multi-Input Heuristik betrachtet werden. Neben der Genauigkeit reduzieren sich aber auch NMI und aNMI deutlich. Während die Multi-Input Heuristik auf einen NMI von 0,89 bzw. einem aNMI von 0,65 kommt, reduzieren sich diese Kennzahlen bei der Kombination aus Multi-Input und Wechselgeldheuristik auf 0,47 bzw. 0,15. [21, S. 173]

6 Methoden basierend auf Informationen abseits des Bitcoin Netzes

6.1 Frei verfügbare Informationen

Über manche Bitcoinadressen sind Eigentümerinformationen frei verfügbar. Nicht selten werden Bitcoinadressen von ihren Eigentümern selbst veröffentlicht. Die Veröffentlichung von Bitcoinadressen kann dabei aus verschiedenen Motiven erfolgen. Unabhängig vom Motiv können diese Informationen durch Scraper gefunden werden.

6.1.1 Motivation für die Veröffentlichung

Einer der offensichtlichsten Gründe ist die Transparenz. Wohltätigkeitsorganisationen oder Crowdfunding-Plattformen veröffentlichen Bitcoinadressen, um Spenden zu erhalten und dabei gleichzeitig die Transaktionen für ihre Unterstützer nachvollziehbar zu machen. Andererseits veröffentlichen einige Entitäten ihre Adressen, um Trinkgeld oder finanzielle Unterstützung für Projekte oder Ideen zu erhalten. Häufig werden dazu Bitcoinadressen in der Signatur von E-Mails oder Forumposts eingebunden. [60, S. 3]

Andere Motivationen können Betrugsmaschen sein. Bei einem Angriff auf Twitter wurde sich Zugang auf mehrere prominente Accounts verschafft. Von diesen wurde die betrügerische Nachricht verbreitet, dass jeder, der auf eine gewisse Bitcoinadresse überweist, den doppelten Betrag zurückerhält. [61] Auch Ransomware kann eine Quelle für Eigentümerinformationen von Bitcoinadressen sein. Bei Ransomware werden alle Dateien auf einem Rechner verschlüsselt und erst gegen die Zahlung von Lösegeld freigegeben. Es ist naheliegend, dass die Adresse, an die das Lösegeld gezahlt werden soll, dem Erpresser gehört.

6.1.2 Informationsgewinn durch Scraping und Crawling

Scraping bzw. Crawling im Allgemeinen ist das automatisierte Extrahieren von Inhalten und Daten aus Webseiten mithilfe von Software. Typischerweise beinhaltet Scraping das Abrufen von Webseiten und das Extrahieren von relevanten Informationen, die dann in einem strukturierten Format gespeichert werden. [62, S. 1]

Im Anwendungsfall der Bitcoinadresssuche bezieht sich Scraping auf die Suche nach Bitcoinadressen in Foren, sozialen Medien und anderen Plattformen, die einem Eigentümer zugeordnet werden können. Eine Entität, die eine Bitcoinadresse in einem Forum oder einem sozialen Medium veröffentlicht, ist in vielen Fällen auch der Eigentümer der Adresse. In manchen Fällen lässt sich nicht nur die Entität herausfinden, sondern auch Standorte, E-Mails, Benutzernamen etc. [19, S. 95] Spagnuolo u.A. setzen in ihrer Untersuchung Scraper auf die Foren BitcoinTalk und Bitcoin-OTC an und stellen weiterhin dar, dass diese Scraper leicht auf andere Plattformen erweiterbar sind. [63, S. 460]

6.1.3 Erkennung einer Bitcoinadresse

Der Kern eines Scrapers oder Crawlers ist das Muster, nach dem er sucht. Auch wenn Bitcoinadressen auf den ersten Blick nach willkürlichen Zeichenkombinationen aussehen, folgen sie Mustern, die für die Suche genutzt werden können. Beispielsweise verwendet die Base58-Kodierung für Bitcoinadressen lediglich einen beschränkten Zeichensatz, der eine Verwechselung von I und l bzw. O und 0 ausschließt. Entsprechend können reguläre Ausdrücke verwendet werden, um nach diesen Mustern zu suchen. Bitcoinadressen implementieren zusätzlich Prüfsummen, um Fehleingaben zu vermeiden. Diese können durch ein Skript überprüft werden, um die Validität der Bitcoinadresse zu bestätigen. [64]

In der Untersuchung von Fleder u.A. wird Scrapy genutzt, um nach Strings zu suchen, die dem folgenden regulären Ausdruck entsprechen.

```
1.{26,33}
```

Anschließend werden die Prüfsummen der Adressen getestet und dadurch überprüft, ob es sich um eine valide Adresse handelt. [60, S. 3]

Der reguläre Ausdruck, der von Kester u.A. verwendet wird, findet alle Zeichenketten, die mit einer 1 beginnen und 27-34 Zeichen lang sind. Modernere Adresstypen wie P2SH oder SegWit-Adressen bleiben jedoch unerkannt. Um alle aktuellen Adresstypen, inklusive P2WSH und P2TR, abzudecken, könnte der folgende reguläre Ausdruck verwendet werden.

```
^(bc1|[13])[a-km-zA-HJ-NP-Z1-9]{25,59}$
```

Dieser reguläre Ausdruck deckt alle Adressen ab, die mit 1, 3 oder bc1 starten, 26 bis 62 Zeichen haben und dabei nur Zeichen der Base58- bzw. Bech32-Kodierung nutzen. Eine Bestätigung durch die Prüfsummen bleibt dennoch erforderlich.

6.2 Informationsgewinn durch Bitcoinforks

Grundlegende Änderungen am Bitcoinprotokoll können zu sogenannten *Hardforks* führen. Hardforks sind Spaltungen der Blockchain, bei der beide resultierenden Stränge fortgeführt werden. Ein Strang der Blockchain wendet die originalen Regeln weiterhin an. Der andere wendet die geänderten Bedingungen an, wodurch eine alternative Blockchain entsteht. Infolgedessen entstehen zwei voneinander unabhängige und parallel verlaufende Blockchain-Zweige, wobei jeder seine eigene Transaktionshistorie und Regelsetzung hat. Dies führt zu einer dauerhaften Spaltung der ursprünglichen Blockchain in zwei separate und eigenständige Ketten. [65]

Die Bitcoin-Blockchain erlebte mehrere solcher Hardforks, die in anderen Kryptowährungen wie Bitcoin-Cash, Bitcoin-Gold, Bitcoin-Diamond etc. resultierten. [65] Die Transaktionen auf den abgespaltenen Blockchains können jedoch Rückschlüsse auf Adresscluster auf der Bitcoin Blockchain

zulassen. Die im Folgenden beschriebene Methode zum chainübergreifenden Adressclustering ist auf alle Forks von Bitcoin anwendbar, welche das Transaktions- und Adressierungssystem beibehalten. Zum einfachen Verständnis wird dieser Vorgang am Beispiel von Bitcoin-Cash dargestellt. Die Methodik ist aber analog auf alle anderen Forks anwendbar.

6.2.1 Der Fork von Bitcoin-Cash

Eine langanhaltende Diskussion über die Skalierbarkeit von Bitcoin resultierte 2017 in einem Hardfork der Bitcoin Blockchain. Bei diesem Fork wurden die Rahmenbedingungen des Minings hinsichtlich der Blockgrößen verändert. Bitcoin-Cash erhöhte die Blockgröße von ursprünglich 1 MB über 8 MB auf 32 MB, um eine erhöhte Transaktionsfrequenz zu erreichen. [66]

Ein Block, der nicht der Maximalgröße vom Bitcoinprotokoll entspricht, wird von den Bitcoin-Minern verworfen, und es wird weiterhin versucht, einen Block an den vorherigen anzuhängen. Es fanden sich jedoch Anhänger dieses Konzepts, die den Block akzeptierten und diese Kette fortführten. Aus diesem Grund entwickeln sich die Blockchains von Bitcoin bzw. Bitcoin-Cash in unterschiedliche Richtungen. Die Veränderung dieser Miningbedingungen veränderte jedoch nicht die allgemeine Funktionsweise. Die Transaktions- und Adressregularien blieben unberührt.

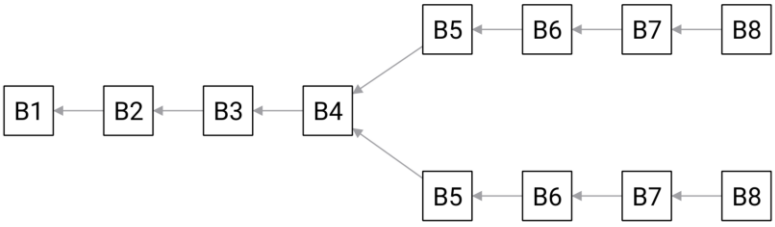


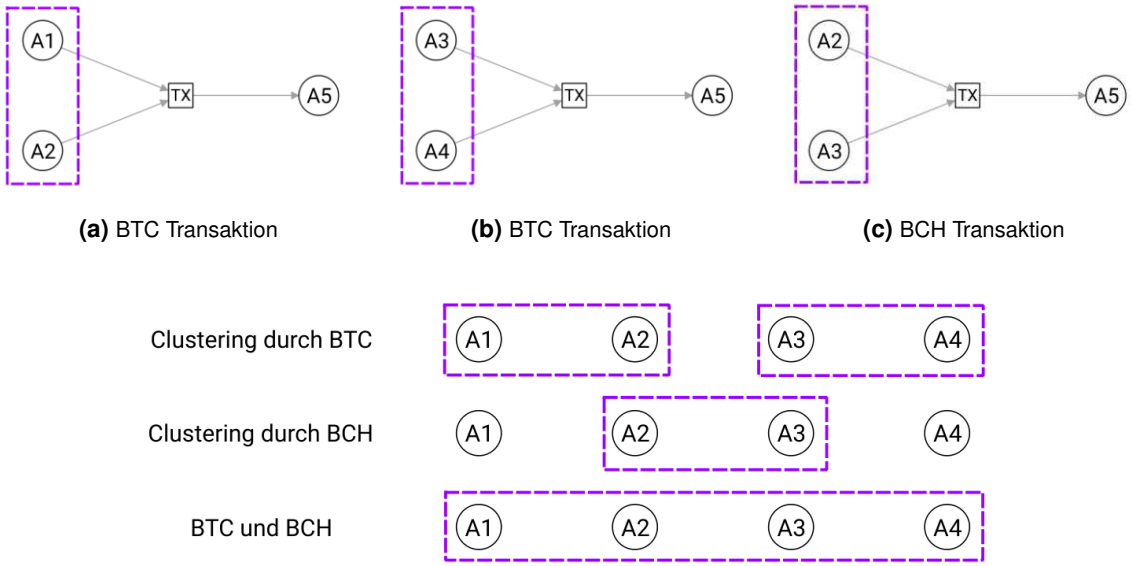
Abbildung 6.1: Darstellung einer Blockchain nach einem Hardfork

Wie in Abbildung 6.1 dargestellt, bleiben beide Blockchains bestehen und werden weiter fortgesetzt. Gleichzeitig fassen beide die Blockchain vor dem Fork als ihre Historie auf. Die Coins, welche vor dem Fork auf einer Adresse lagen, können also sowohl auf der Bitcoin-Blockchain, als auch auf der Bitcoin-Cash-Blockchain ausgegeben werden. [67, S. 240–242] [68, S. 1 f.]

Adressen können danach eingeteilt werden, wann und auf welcher Blockchain sie eingesetzt werden. Besonders interessant sind solche Adressen, die nach dem Fork auf beiden Blockchains verwendet werden. Bei gleichen Adressen ist davon auszugehen, dass sie unabhängig von der Blockchain der gleichen Entität gehören.

6.2.2 Chainübergreifendes Adressclustering

Die Heuristiken beschrieben in 4.1 und 4.2 sind ebenso auf Transaktionen des Bitcoin-Cash Netzwerks anwendbar. Speziell die Multi-Input Heuristik angewandt auf Bitcoin-Cash Transaktionen kann wesentliche Clusteringergebnisse aufdecken, die allein durch die Analyse der Bitcointransaktionen nicht realisierbar wären. [53, S. 2731 f.] Ein Beispiel, wie chainübergreifendes Clustering zusätzliche Informationen enthüllen kann, ist abgebildet in Abbildung 6.2.



(a) BTC Transaktion **(b)** BTC Transaktion **(c)** BCH Transaktion

(d) Übersicht Single-Chain und chainübergreifendes Clustering

Abbildung 6.2: Visualisierung des chainübergreifenden Clusterings

Angenommen, Alice hält vor dem Fork auf den Adressen A1, A2, A3 und A4 Bitcoin. Nach dem Fork besitzt sie die Coins auf der Bitcoin und Bitcoin-Cash Blockchain. Eve beobachtet das Bitcoinnetzwerk und konnte bisher zwei kleine Cluster A1;A2 (6.2a) und A3;A4 (6.2b) feststellen, diese aber nicht miteinander verbinden. Durch Beobachtung des Bitcoin-Cash-Netzwerks stellt Eve das Cluster A2;A3 (6.2c) fest. Durch das chainübergreifende Clustering kann sie nun alle Adressen A1 bis A4 einem Cluster zuordnen (6.2d).

6.2.3 Mächtigkeit der Chainübergreifenden Clusteringmethodik

Um die Qualität und Mächtigkeit dieser Strategie zu prüfen, untersuchen Kaldoner u.A. die kurzzeitigen und langzeitigen Auswirkungen dieser Methode. Die kurzfristige Analyse der Bitcoin-Cash Blockchain kann im Wesentlichen korrekte Clusteringergebnisse aufdecken, bevor die Analyse der Bitcoinblockchain allein diese Ergebnisse bestätigt. Die langfristige parallele Analyse beider Chains kann die Cluster in vielen Fällen weiter reduzieren und deckt zu großen Teilen Beziehungen zwischen Clustern auf, die durch die Analyse der Bitcoinblockchain allein nicht zu erkennen wären. [53, S. 2731 f.]

Kurzzeitige Auswirkungen

Bei der kurzzeitigen Auswirkungsanalyse untersuchen Kaldoner u.A. die ersten 5 Monate der Blockchain. Sie können zeigen, dass ein chainübergreifendes Clustering etwa 1,05 Millionen zusätzliche Clustermerges, gegenüber der Analyse der Bitcoinblockchain allein, verursacht. Beobachtet man die Bitcoinblockchain über diesen Zeitraum hinaus, zeigt sich, dass 75,5 % der Clustermerges zu einem späteren Zeitpunkt auch durch die Beobachtung der Bitcoinblockchain allein möglich gewesen wären. Im Durchschnitt zeigen sich die Clustermerges auf der Bitcoinblockchain aber erst 9 Monate später. Dass eine hohe Überlappung der Clustermerges besteht, ist ein Indiz dafür, dass die Clustermer-

ges korrekt sind. Die restlichen Clustermerges werden ausschließlich durch das chainübergreifende Adressclustering gefunden. Das bedeutet, die Cluster wären, basierend auf Transaktionen der Bitcoinblockchain allein, nicht zusammengeführt worden. [53, S. 2731]

Langzeitige Auswirkungen

Um die langzeitigen Auswirkungen abzuschätzen, werden die nachfolgenden 2 Jahre der Bitcoin-Cash Blockchain analysiert. In diesem Zeitraum können durch das chainübergreifende Clustering etwa 570.000 zusätzliche Clustermerges identifiziert werden. Die Analyse der Bitcoinblockchain führt mehr als 30 Millionen Adressen zu etwas mehr als 750.000 Entitäten zusammen. Diese Clustermerges der chainübergreifenden Analyse reduziert die Zahl der Entitäten weiter, auf weniger als 200.000. [53, S. 2731 f.]

Zusammenfassend lässt sich sagen, dass das chainübergreifende Adressclustering sowohl auf kurze als auch auf lange Sicht einen Einfluss auf die Entitätsnachverfolgung von Bitcoinnutzern hat. Insbesondere hinsichtlich der Identifizierung von Verbindungen zwischen Bitcoinadressen aufgrund deren Aktivitäten auf der Bitcoin-Cash-Blockchain.

6.3 Verknüpfung von Transaktionen mit IP-Adressen

Das Verknüpfen von IP-Adressen mit Transaktionen ist eine weitere Methode, den identifizierten Adressclustern die realweltliche Entität zuzuordnen. Einige Entitäten geben diese Information freiwillig preis. Siehe Kapitel 6.1. Andere Entitäten wie Marktplätze oder Kryptobörsen können ihre Entität nur schwer verschleiern. Siehe Kapitel 4.4.1. Durch die Verknüpfung von Transaktionen mit IP-Adressen lassen sich einige Entitäten dennoch identifizieren.

6.3.1 Technische Realisierbarkeit

Die technische Grundlage dieser Methode liegt in der Verbreitung von Transaktionen im Bitcoinnetzwerk. Wie in den technischen Grundlagen dargelegt, werden Transaktionen von den Knoten im Peer-to-Peer Netzwerk weitergeleitet. Bei der Verknüpfung von Transaktionen mit IP-Adressen wird angenommen, dass der Knoten, der eine Transaktion als Erstes in das Peer-to-Peer Netzwerk sendet, der Initiator der Transaktion ist. Es ist also erforderlich, ein Peer dieses Knotens zu sein, um den Initiator einer Transaktion zu identifizieren [69].

Auch wenn das Bitcoinsystem nicht explizit zwischen Client und Server unterscheidet, lassen sich Knoten des Bitcoinnetzwerks in zwei Kategorien einteilen. Öffentliche Knoten entsprechen dabei Servern, also Dienstleistern, die Transaktionen empfangen und weiterleiten. Private Knoten entsprechen Clients, also Nutzern, die Transaktionen lediglich versenden, jedoch weder empfangen noch weiterleiten. [70, S. 3]

Öffentliche Bitcoin Knoten

Im Peer-to-Peer-Netzwerk kann mittels der *GETADDR*-Nachricht jeder Knoten die Nachbarn seiner eigenen Nachbarknoten ermitteln. [70, S. 3] Man sendet also diese Nachricht an alle seine Nachbarknoten und erhält von jedem Nachbarn eine Liste von anderen Knoten, die man wiederum nach ihren

Nachbarknoten fragt. Rekursiv führt man diesen Vorgang aus, bis man alle Knoten des Netzwerks identifiziert hat. [69], [70, S. 6] Die Gesamtzahl der öffentlich erreichbaren Bitcoinknoten schwankt leicht und lag im Jahr 2023 zwischen 14.000 und 18.000. [71]

Private Bitcoin Knoten

Viele Nutzer von Bitcoin beschränken sich darauf, ihre Transaktionen in das Bitcoin Peer-to-Peer-Netzwerk zu senden. Diese privaten Bitcoinknoten sind nur bedingt Teil des Peer-to-Peer Netzwerks, da sie keine Transaktionen empfangen oder weiterleiten. [70, S. 3] Um auch die Transaktionen dieser Teilnehmer zu erfassen, werden von einem Angreifer möglichst viele öffentlich erreichbare Knoten erstellt. Angesichts der Tatsache, dass jeder Bitcoin-Teilnehmer in der Regel sieben oder acht andere Knoten als Nachbarn auswählt, sollte der Angreifer mindestens jeden siebten öffentlichen Knoten kontrollieren. [69] [70] Je mehr öffentliche Knoten der Angreifer bereitstellt, desto höher ist die Wahrscheinlichkeit, dass ein privater Knoten mindestens einen der vom Angreifer kontrollierten Knoten als Nachbarknoten auswählt.

Aussagekraft und IP Verknüpfung

Bei Empfang einer Transaktion auf einem seiner öffentlichen Knoten kann der Angreifer prüfen, ob der Ursprungsknoten, von dem die Transaktion gesendet wurde, öffentlich ist und damit Transaktionen weiterleitet oder nicht. Im Fall eines privaten Knotens ist die IP-Adresse des Senders mit hoher Wahrscheinlichkeit die des Transaktionsinitiators. Diese Zuordnung ist wahrscheinlich, da es sich bei der Transaktion nicht um eine Weiterleitung handeln kann.

Für den Fall, dass es sich bei dem Ursprungsknoten um einen öffentlichen Knoten handelt, kann nicht ausgeschlossen werden, dass es sich um eine weitergeleitete Transaktion handelt. Jedoch steigt mit zunehmender Anzahl Angreifer kontrollierter öffentlicher Knoten die Wahrscheinlichkeit, dass der öffentliche Knoten tatsächlich der Initiator der Transaktion ist.

Diese Vorgehensweise beschreibt auch der Bitcoin-Entwickler 0xB1C. Gleichzeitig beobachtet er dieses Verhalten bei einer Entität, die er als „Linking Lion“ bezeichnet. Diese Entität operiert seit 2018 etwa 812 Bitcoinknoten aus verschiedenen Adressbereichen. Nach seiner Vermutung steht hinter dieser Entität ein Analyseunternehmen. [72]

6.3.2 Übertragung durch Tornetzwerk

Nun ließe sich einwenden, dass ein Nutzer sich der Identifizierung seiner IP-Adresse entziehen kann, indem er Transaktionen über das Tor-Netzwerk an öffentliche Knoten schickt. Selbst wenn einer der Nachbarknoten eine Zuordnung von IP-Adresse und Transaktion vornimmt, erhält dieser schließlich nicht die IP-Adresse des Nutzers, sondern eine Tor-IP-Adresse. [69]

Dieses Argument ist zutreffend. Allerdings lässt sich die Nutzung von Tor für Bitcoin Transaktionen temporär unterbinden. Der in Bitcoin implementierte Schutz vor Denial of Service Angriffen lässt sich dazu verwenden, Tor-IP-Adressen zu sperren. Sendet eine IP-Adresse eine schadhafte Nachricht (fehlerhafte Transaktion oder Block), erhält diese IP-Adresse Strafpunkte. Erreicht dieser einen Wert

von 100, wird jegliche Kommunikation von dieser IP-Adresse für 24 Stunden ignoriert. So lassen sich alle IP-Adressen vom Tor-Netzwerk, aber auch von anderen Diensten, wie zum Beispiel prominenten VPN-Anbietern, sperren. [70, S. 5], [73, S. 5]

6.3.3 Ausnahmen und Fehler

Mit der dargestellten Methode lassen sich Transaktionen mit IP-Adressen in Verbindung bringen. Typischer Weise werden Transaktionen vom Sender in das Bitcoin-Netzwerk geschickt. Es existieren jedoch auch Anwendungsfälle wie z. B. PayJoin, bei denen die Transaktion vom Empfänger in das Bitcoinsystem geschickt wird. Diese Unterscheidung und damit die Zuordnung von IP-Adressen zu Bitcoinadressen ist nur bedingt möglich.

Einen weiteren Ausnahmefall stellen Transaktionsverbreitungsdienste dar. Bei solchen Diensten lassen sich Bitcointransaktionen über ein Webinterface eingeben. [10, S. 483] [74] [75] Diese werden nach einer Prüfung an das Bitcoinnetzwerk weitergeleitet. So entsteht ein Intermediär, der als solcher identifiziert werden kann, eine Identifikation der tatsächlichen Transaktionsinitiatoren aber nicht zulässt. Da die Transaktionen von solchen Diensten vor der Weiterleitung geprüft werden, ist eine Sperre nach 6.3.2 ausgeschlossen.

7 Analyse von Finanzströmen

Alle Bitcointransaktionen sind für jeden durch die Blockchain einsehbar. Da Transaktionsinputs Referenzen auf frühere Transaktionsoutputs sind, lassen sich diese Transaktionen verketten. Durch die Verkettung von Transaktionen lassen sich Finanzströme identifizieren. Die Finanzstromanalyse fokussiert sich dabei auf die Weiterverwendung einer spezifischen Geldsumme. Außerdem kann umgekehrt nach der Herkunft eines Wertes gesucht werden. Bei der Analyse von Finanzströmen wird nach Auffälligkeiten und Mustern gesucht, die auf gewisse Aktivitäten hindeuten können.

7.1 Darstellung von Transaktionen in Graphen

Um Finanzströme zu identifizieren, werden die Transaktionen verkettet und als Graph betrachtet. Es gibt jedoch verschiedene Herangehensweisen, Transaktionen im Bitcoinsystem in einem Graphen darzustellen. Die verschiedenen Graphenkonzepte eignen sich dabei für verschiedene Analysemethoden.

7.1.1 Transaktionsgraphen

Transaktionsgraphen sind eine Möglichkeit, Transaktionen ohne Gruppierung darzustellen. Sie stellen lediglich dar, auf welche vorhergehenden Transaktionen eine neue Transaktion Bezug nimmt. Die Knoten des Graphen sind Transaktionen und die gerichteten Kanten zwischen den Knoten entsprechen den Transaktionsinputs bzw. Outputs. [20, S. 204] Eine Darstellung eines Transaktionsgraphen ist zu sehen in Abbildung 7.1.

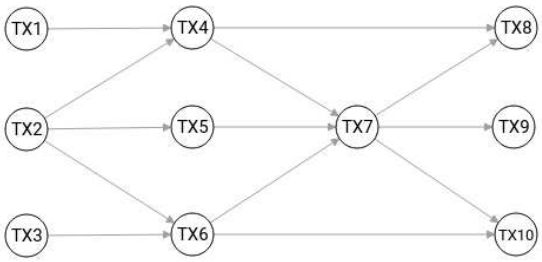


Abbildung 7.1: Graphische Darstellung eines Transaktionsgraphen

Hat eine Transaktion mehrere Inputs bzw. Outputs, so hat dessen Knoten im Transaktionsgraph mehrere eingehende bzw. ausgehende Kanten. Die Adressen spielen in diesem Graphenkonzept keine Rolle. Da der Output einer Transaktion erst in einer späteren Transaktion als Input verwendet werden kann, verlaufen alle Kanten des Transaktionsgraphen in eine Richtung. Aus diesem Grund sind auch Kreise in einem Transaktionsgraphen ausgeschlossen.

7.1.2 Adressgraphen

Es sei angemerkt, dass die Begriffe Transaktionsgraph und Adressgraph in der Literatur zum Teil synonym verwendet werden. Der in [60, S. 5] beschriebene Transaktionsgraph entspricht vielmehr der Definition eines Adressgraphen. Beim Adressgraph entsprechen Knoten den Bitcoinadressen, und die gerichteten Kanten Transaktionen von einer Quelladresse zu einer Zieladresse. [76, S. 463] Eine beispielhafte Darstellung eines Adressgraphen ist abgebildet in Abbildung 7.2.

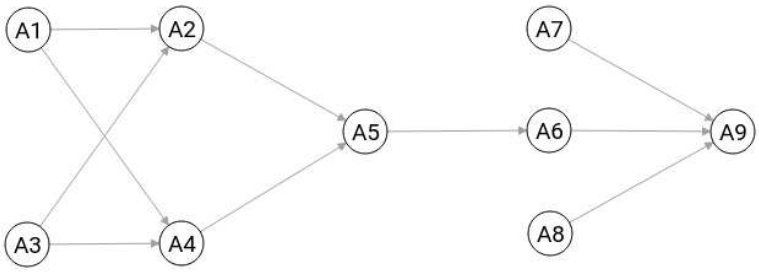


Abbildung 7.2: Graphische Darstellung eines Adressgraphen

Diese Darstellung ermöglicht es, Bitcoins von einer Adresse zu einer anderen zu verfolgen, über mehrere Transaktionen hinweg. Umgekehrt lässt sich so auch die Herkunft von Bitcoins feststellen. Zusammengefasst: Es lässt sich nachvollziehen, woher die Coins einer Adresse stammen und wie sie weiterverwendet werden.

7.1.3 Nutzergraphen

Der Nutzergraph oder auch Clustergraph stellt Transaktionen nicht von Adresse zu Adresse, sondern von Entität zu Entität dar. Die Knoten eines Nutzergraphen sind also Entitäten und die gerichteten Kanten zwischen den Knoten stellen die Transaktionen zwischen ihnen dar. Der Nutzergraph nutzt den Adressgraph als Basis und verbindet mehrere Adressknoten zu einem Entitätsknoten. [20, S. 205 ff.] Der Nutzergraph ist also eine Aggregation von Knoten des Adressgraphs. Eine Visualisierung eines Nutzergraphen ist zu sehen in Abbildung 7.3.

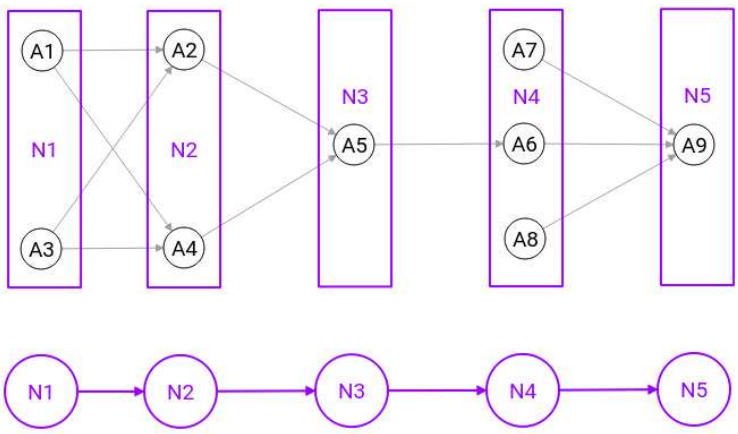


Abbildung 7.3: Überführung eines Adressgraphen in einen Nutzergraph

Welche Adressen einer Entität zugehören, bestimmt sich dabei aus den genutzten Heuristiken zum Adressclustering. So kann ein Nutzergraph beispielsweise ein Adressclustering nur auf Basis der Multi-Input Heuristik vornehmen. [20] Ebenfalls kann ein Clustering anhand einer Kombination von Multi-Input und Wechselgeldheuristik erfolgen. [51]

7.2 Breitensuche

In einer Untersuchung von Zhao und Guan wird ein Breitensuchalgorithmus auf Adress- und Nutzergraphen angewandt. Dazu geht man zunächst von einem oder mehreren Knoten als Ausgangspunkt aus. Als Erstes werden alle Knoten besucht, die eine Kante von einem Ausgangsknoten entfernt sind. Der Algorithmus speichert die Anzahl aller Knoten, die einen Abstand von 1 haben (Knotenzahl). Rekursiv wird den Kanten in tiefere Ebenen gefolgt, bis keine weiteren Knoten mehr erreichbar sind. [51, S. 88 f.]

Als Beispiel soll der Graph in Abbildung 7.4 dienen.

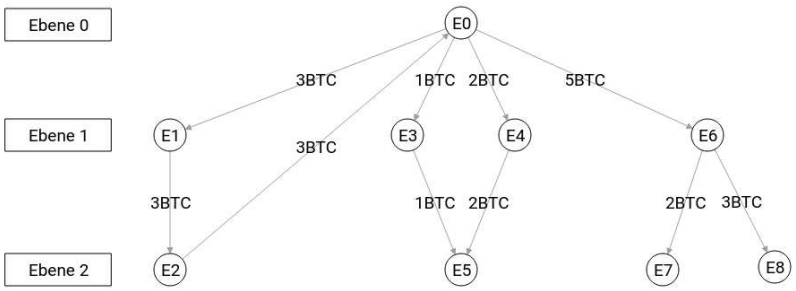


Abbildung 7.4: Ebenendarstellung des BFS-Suchalgorithmus

E0 wird als Ausgangsknoten festgelegt. E0 tätigt Transaktionen an E1, E3, E4 und E6. In Ebene 1 befinden sich also 4 Knoten. In der nächsten Iteration des Algorithmus werden die Knoten in Ebene 1 jeweils als Ausgangspunkt betrachtet. Diese tätigen Transaktionen an E2, E5, E7 und E8. Die Knotenanzahl in Ebene 2 ist also 4. In der folgenden Iteration werden die Knoten aus Ebene 2 als Ausgangspunkte gewählt. Hier findet nur noch eine Transaktion von E2 zu E0 statt. Die Knotenzahl in Ebene 3 ist also 1.

Mit der Breitensuche auf dem Nutzergraphen können Transaktionsmuster erkannt werden. Ein Baummuster wie in Abbildung 7.4 rechtsseitig zu sehen ist eher unauffällig. Ein Transaktionsmuster, bei dem Werte erst aufgetrennt und später zum gleichen Wert summierend zusammengeführt werden (Abbildung 7.4 mittig), ist ebenso verdächtig wie Kreise (Abbildung 7.4 linksseitig). Beide Transaktionsmuster können auf einen Geldwäscheversuch hindeuten. [51, S. 89]

Es muss jedoch anerkannt werden, dass dies nur Indizien bzw. Hinweise sein können. Die genannten Transaktionsmuster können auch unter normalen Umständen auftreten. [51, S. 93] Kreise in Nutzergraphen können zum Beispiel bei Nutzern auftreten, die untereinander Bitcoins senden. Ein Trennen und späteres Zusammenführen von Bitcoinwerten tritt auch dann auf, wenn zwei Nutzer Coins von der gleichen Börse kaufen und anschließend auf dem gleichen Handelsplatz einkaufen.

7.3 Taintanalyse

Weiterhin können Methoden der Taintanalyse auf den Transaktionsgraphen angewendet werden. Die Taintanalyse ist eine Methode, bei der markierte Geldmengen auf ihrem Weg durch das Transaktionsnetzwerk beobachtet werden. [77, S. 13]

Ein mögliches Ziel einer Taintanalyse ist das Sperren bzw. *Blacklisten* von nicht legitimen Coins. So könnten Händler markierte Coins als Zahlungsmittel ablehnen. [78, S. 21] Auch wäre die Prüfung von Transaktionen durch Miner denkbar, indem Transaktionen mit markierten Coins nicht in Blöcke aufgenommen werden, bzw. Blöcke mit markierten Transaktionen nicht akzeptiert werden.

Dabei ist jedoch die Zeitspanne zwischen dem Vorfall, der zur Sperrung führt, und der tatsächlichen Sperrung des markierten Geldes zu beachten. Wenn diese Ereignisse zeitlich auseinander liegen, könnten unbeteiligte Dritte den wirtschaftlichen Schaden tragen. [78, S. 20]

Im Bitcoinsystem können Werte durch Transaktionen geteilt bzw. kombiniert werden. Um Transaktionswerte dennoch verfolgen zu können, werden unterschiedliche Ansätze zur Verfolgung von markierten Geldmengen angewandt. Manche Ansätze nutzen eine binäre Unterscheidung von markierten und nicht markierten Coins. Andere wiederum lassen einen Markierungsanteil (Taintscore) zu. [77, S. 13] Diese Teilmarkierung kann dazu genutzt werden, um den illegitimen bzw. legitimen Wertanteil eines Transaktionsoutputs zu bestimmen. Das könnte bei der Umsetzung eines Coinblacklistings dazu genutzt werden, den Restwert eines Transaktionsoutputs zu bestimmen. [78, S. 21]

Zur einfachen Darstellung erfolgen die Markierungen an den Adressen, die an der Transaktion beteiligt sind. Die Markierungen beziehen sich jedoch nicht auf Adressen, sondern nur auf spezifische Transaktionsoutputs. Würden stattdessen die Adressen markiert, ließen sich alle Transaktionsoutputs, die legitim auf einer Adresse sind, durch eine Transaktion mit markierten Coins unbrauchbar machen. [78, S. 21]

7.3.1 Poison Methode

Bei der Poison Methode werden alle Outputs einer Transaktion binär markiert. Hat eine Transaktion keinen markierten Input, so bleiben auch die Outputs unmarkiert. Hat eine Transaktion jedoch einen oder mehrere markierte Inputs, so werden alle Transaktionsoutputs ebenfalls vollständig markiert. Das geschieht unabhängig davon, ob die Transaktion unmarkierte Inputs besitzt oder nicht. Bei dieser Methode kann sich die Anzahl der markierten Coins erhöhen, wenn markierte gemeinsam mit nicht markierten Coins als Input genutzt werden. [78, S. 21 f.] Eine visuelle Darstellung der Poison Methode ist abgebildet in Abbildung 7.5

Die Poison Methode eignet sich gut, um alle Outputs zu markieren, die mit der ursprünglich markierten Geldmenge in Verbindung stehen. Allerdings kann innerhalb weniger Transaktionen die markierte Geldmenge auf ein Vielfaches ihrer selbst ansteigen. [77, S. 13] Würde sich eine Politik der Ablehnung von markierten Outputs durchsetzen und die Menge der markierten Coins erhöhen, würde sich die Menge der verwendbaren Coins immer weiter reduzieren. [78]

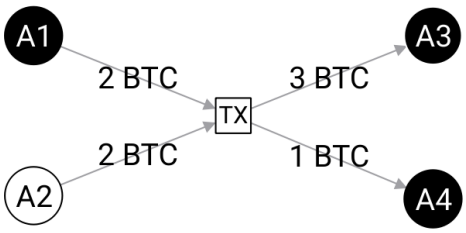


Abbildung 7.5: Verbreitung des Taintscores bei der Poison Methode

7.3.2 Haircut Methode

Anstatt eine Transaktion bzw. deren Outputs vollständig zu markieren, erfolgt bei der Haircut Methode eine relative Markierung der Coins. Die Outputs werden dabei anteilig in der Proportion markiert, in der markierte und nicht markierte Coins in den Inputs auftreten. [77, S. 13] Der Taintscore der Outputs berechnet sich also folgendermaßen aus den Inputs i:

$$\text{Taintscore} = \frac{\sum_i (\text{Wert}_i \times \text{Taintscore}_i)}{\sum_i \text{Wert}_i}$$

Diese anteilige Markierung kann ebenso fortlaufend angewandt werden wie die Poison Methode.

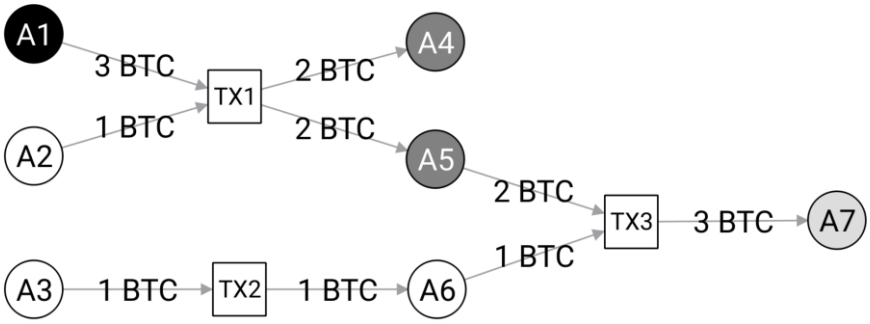


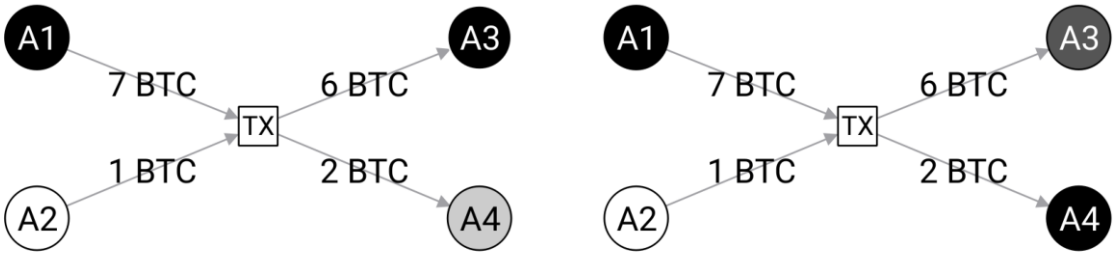
Abbildung 7.6: Verbreitung des Taintscores bei der Haircut Methode

Die beispielhafte Darstellung der Haircut Methode, abgebildet in Abbildung 7.6, zeigt den Taintscore eines Outputs anhand seiner Einfärbung. In diesem Beispiel soll der markierte Wert von A1 durch den Transaktionsgraphen verfolgt werden. A1 ist daher vollständig markiert, das entspricht einem Taintscore von 1. Die 3 markierten Coins von A1 werden in TX1 mit einem nicht markierten Coin (Taintscore von 0) zusammengeführt. Die Outputs A4 und A5 haben daher einen Taintscore von 0,75. In TX2 wird ein nicht markierter Coin transferiert. Bei TX3 hat einer der Inputs einen Wert von 2 BTC bei einem Taintscore von 0,75 und der andere einen Wert von 1 BTC bei einem Taintscore von 0. Die Outputs dieser Transaktion haben dementsprechend einen Taintscore von 0,5.

Ebenso wie bei der Poison Methode, werden alle Transaktionen, die mit der ursprünglich markierten Geldmenge in Verbindung stehen, mindestens teilweise markiert. Der Taintscore verrät jedoch den relativen Anteil des ursprünglich markierten Geldes in jedem Transaktionsoutput. Die Summe der markierten Coins bleibt also zu jedem Zeitpunkt der Betrachtung gleich. [77, S. 13]

7.3.3 Taintmethoden auf Basis der Input Reihenfolge

FIFO und *LIFO* sind die Kurzformen für First In First Out bzw. Last In First Out. Sie beschreiben, in welcher Reihenfolge Elemente bzw. Ressourcen verwendet werden. In diesem Fall bedeuten sie, dass bei mehreren Transaktionsinputs die Werte in dieser Reihenfolge auf die Transaktionsoutputs angerechnet werden. [79, S. 245] Konkret werden bei der FIFO Methode die zuerst auftretenden Inputs auf die zuerst auftretenden Outputs angerechnet. Analog werden bei der LIFO Methode die zuletzt auftretenden Inputs auf die zuerst auftretenden Outputs angerechnet.



(a) First In First Out Methode **(b)** Last In First Out Methode

Abbildung 7.7: Verbreitung des Taintscores bei FIFO und LIFO

Beispielhafte Darstellungen der FIFO und LIFO Methoden sind in der Abbildung 7.7 zu sehen. Bei der FIFO Methode wird der erste Input (A1) zuerst auf den ersten Output (A3) angerechnet. Da der erste Input den ersten Output vollständig abdeckt, wird dieser vollständig markiert. Zusätzlich deckt der erste Input eine Hälfte des zweiten Outputs (A4) ab. Da die andere Hälfte von einem nicht markierten Output (A2) stammt, erhält A4 einen Taintscore von 0,5.

Bei der LIFO Methode wird der letzte Output (A2) zuerst auf den ersten Output (A3) angerechnet. Daher besteht A3 aus einem nichtmarkierten und 5 markierten Coins. Entsprechend erhält er einen Taintscore von 0,83. Der letzte Output (A4) wird ausschließlich durch den markierten Input gedeckt, weshalb dieser vollständig markiert wird.

Der Vorteil, der sich aus der Verwendung von FIFO bzw. LIFO gegenüber der Haircut Methode ergibt, liegt in der absoluten Anzahl der markierten Knoten. Während die Haircut Methode alle Outputs zumindest teilweise markiert, können bei FIFO bzw. LIFO Outputs ohne Markierung verbleiben. [77, S. 13], [80, S. 3] In einer Untersuchung von Anderson u.A. wurden 132 Werte markiert und von der FIFO bzw. Haircut Methode langfristig durch das Transaktionsnetzwerk verfolgt. Die Ergebnisse dieser Untersuchung zeigen, dass die Markierungen der Haircut Methode die Markierungen der FIFO Methode um ein Vielfaches übersteigen. [79]

Dennoch scheint die Anrechnungsreihenfolge willkürlich, da die Transaktionsinputs und -outputs bei Bitcoin beliebig permutiert werden können. Begründet werden kann die Verwendung von FIFO mit der britischen Rechtsprechung. Nach einem Urteil sind die legitimen bzw. illegitimen Einzahlungen auf ein Konto nach dem FIFO-Prinzip auf die Zahlungen anzurechnen. [79]

7.3.4 Wertbasierte Taint Methode

Die wertbasierte Taint Methode TIHO (Taint In, Highest Out) rechnet alle markierten Inputwerte auf die werthöchsten Outputs an. [80, S. 4] Um die Gesamtanzahl der markierten Coins aller Inputs zu erhalten, werden der Wert und Taintscore der Inputs multipliziert. Anschließend werden die markierten Coins aller Inputs aufsummiert. Die Summe aller markierten Coins wird dann auf die Outputs in der Reihenfolge ihrer Wertigkeit angerechnet.

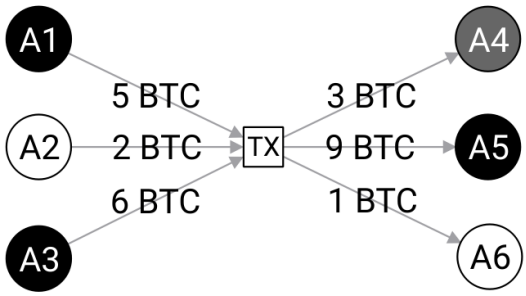


Abbildung 7.8: Verbreitung des Taintscores bei der TIHO Methode

Im Beispiel, dargestellt in Abbildung 7.8, werden 2 vollständig markierte und ein nicht markierter Input für eine Transaktion verwendet. Die Gesamtzahl der markierten Coins liegt also bei 11. Diese markierten Coins werden auf die Outputs ihrer Größe nach angerechnet. Der größte Output der Beispieltransaktion ist A5 mit 9 BTC. Da dieser Output vollständig von den 11 markierten Coins gedeckt wird, erhält dieser Output eine vollständige Markierung. Die restlichen 2 markierten Coins werden auf den nächstkleineren Output (A4) angerechnet. Dementsprechend erhält der Output A4 eine Markierung von 0,66. Der kleinste Output A6 bleibt in diesem Beispiel unmarkiert.

Laut Tironsakkul u.A. begründet sich diese Methode auf der Annahme, mit dem werthöchsten Output regelmäßig den Hauptzweck einer Transaktion zu identifizieren. Dass diese Annahme nicht auf jede Transaktion zutrifft, stellen sie ebenfalls heraus. [80, S. 4 f.] Bei wie vielen Fällen angenommen werden kann, dass der größte Output den Hauptzweck einer Transaktion darstellt, bleibt unklar. Speziell im Fall von Peelingchains oder allgemein Zahlungen mit großen Inputs, aber geringen Kaufbeträgen, ist diese Annahme nicht zutreffend.

8 Diskussion und Ausblick

8.1 Darstellung der Methoden

Es werden stetig Abwandlungen der Anonymisierungstechnologien entwickelt. Die Verwendung der Multi-Input Heuristik führte zur Entwicklung von CoinJoin, was wiederum die Grundlage für PayJoin darstellt. Hier endet die Entwicklung jedoch nicht. Entwicklungen wie z.B CoinShuffle [81] basieren auf der gleichen Grundlage und erweitern den Funktionsumfang.

Die stetige Entwicklung abgewandelter oder neuer Anonymisierungstechnologien führt dazu, dass die Methoden immer wieder auf Gültigkeit überprüft werden müssen. Schließlich können neue Anonymisierungstechnologien bei breiter Anwendung die Annahmen der unterschiedlichen Heuristiken brechen.

Ein konkretes Beispiel dafür ist das Adressformat P2TR. Durch die Verwendung dieses Adresstyps lassen sich Transaktionen an Entitätsgruppen realisieren. Um den UTXO zu nutzen, muss der Nutzer aber nur noch nachweisen, dass er einer der Empfänger ist, nicht jedoch welcher. [7], [8] Anders ausgedrückt: Bitcoinadressen referenzieren keine individuellen Entitäten mehr. Diese Änderung kann Clusteringheuristiken erheblich beeinflussen. Andere Methoden, wie zum Beispiel die Analyse von Transaktionsströmen, bleibt davon völlig unberührt. [82]

8.2 Qualität der Methoden

Neben der Funktionsweise wurden die Methoden mit verschiedenen Metriken auf ihre Qualität überprüft. Konkret wurden die Eigenschaften Anwendbarkeit und Genauigkeit mit verschiedenen Kennzahlen beschrieben. Es besteht jedoch ein entscheidendes Problem bei der Erzeugung eines verlässlichen Datensatzes, der die Besitzverhältnisse von Adressen korrekt abdeckt. Verwendet man einen synthetischen Ansatz, ist der Datensatz nicht repräsentativ für das Bitcoinsystem. Erstellt man einen Datensatz auf Basis tatsächlicher Transaktionen im Bitcoinsystem, wird immer eine Restunsicherheit über die Zuordnungen bestehen.

Unabhängig davon, welchen Ansatz man zur Datensatzerzeugung verwendet: Eine statistische Untersuchung auf dessen Basis kann keine Genauigkeitsdaten liefern, die unmittelbar auf das gesamte Bitcoinsystem anwendbar sind. Noch etwas deutlicher wird dies von Hasse im Interview ausgedrückt. Er sagt, bei der statistischen Untersuchung auf Basis eines solchen Datensatzes ließe sich lediglich bewerten, wie gut die Heuristik mit dem Datensatz übereinstimmt, und nicht die reale Genauigkeit der Heuristik einschätzen. Um verlässliche Aussagen treffen zu können, müsse der Einzelfall geprüft werden.

8.3 Einfluss auf die Anonymität

Diese Arbeit stellt dar, dass mithilfe von verschiedenen Nachverfolgungsmethoden die Anonymität der Nutzer von Bitcoin beschnitten werden kann. Durch eine bewusste anonyme Verwendung kann ein Nutzer jedoch seine Identität verbergen und schützen.

Werkzeuge und Dienste zur Anonymisierung entwickeln sich weiter. Die Einschätzung der Anonymität der Kryptowährung Bitcoin ist daher nur eine Momentaufnahme. Konzepte wie P2TR könnten auf Dauer die Methoden zur Deanonymisierung schwächen. Doch obwohl die Nutzung von Taproot seit mehr als 2 Jahren unterstützt wird, wird die Funktion erst seit kurzer Zeit signifikant eingesetzt. [9]

Eine mögliche Erklärung dafür ist, dass die frühe Anwendung dieser Privatsphäre-Funktionalität im ersten Moment zu einer Schwächung der eigenen Privatsphäre führen kann; speziell durch die Verwendung des Wechselgeldhinweises anhand unterschiedlicher Adresstypen. Es wird also abgewogen, ob der Privatsphäregewinn, der durch Taproot erreichbar ist, den kurzfristigen Privatsphäreverlust wert ist.

8.4 Ausblick

Eine Empfehlung für weitere Forschung ist daher, die Auswirkungen von Taproot auf die Anonymität des Bitcoinsystems zu untersuchen. Es sollte aufgezeigt werden, welche Implikationen die Einführung von Taproot für die Privatsphäre der Nutzer hat, welche Anonymisierungsfunktionalitäten sie bieten kann und inwiefern der langfristige Zugewinn an Privatsphäre einen zwischenzeitlichen Verlust an Privatsphäre rechtfertigen kann.

Es wäre weiterhin zu untersuchen, welchen Einfluss künstliche Intelligenz bzw. maschinelles Lernen auf die Analyse von Transaktionsdaten hat. Künstliche Intelligenz erkennt Muster und Zusammenhänge in großen Datenmengen und zeigt daher möglicherweise neue Ansätze für Adressclusteringmethoden und die Identifizierung von Transaktionsbeziehungen auf. Bei einer solchen Untersuchung sollte jedoch besonders auf die Korrektheit des Datensatzes geachtet werden, auf dessen Basis die künstliche Intelligenz trainiert wird.

9 Fazit

Das Ziel dieser Arbeit war es herauszufinden, inwiefern Nutzer und Transaktionen des Bitcoinsystems nachverfolgt werden können. Dazu wurde in der vorliegenden Arbeit auf verschiedene Methoden eingegangen, welche die Anonymität von Bitcointransaktionen beschneiden. Diese Methoden lassen sich in vier Kategorien einteilen.

- Die Wiedererkennung und Zuordnung von Transaktionen auf der Bitcoin-Blockchain anhand von Transaktionsmerkmalen.
- Die Gruppierung von Adressen anhand ihrer Verwendung und Transaktionseigenschaften.
- Methoden, die Informationen abseits des Bitcoinsystems in die Analyse einfließen lassen.
- Die Analyse von Finanzströmen zur Mustererkennung und um Gelder über Transaktionen hinweg zu verfolgen.

Die Wiedererkennung von Transaktionen auf der Blockchain, die in Kapitel 3 dargestellt wird, zeigt, wie Transaktionen initial identifiziert und zugeordnet werden können. Es wird eindeutig aufgezeigt, dass Transaktionen identifizierbar sind, teilweise auch dann, wenn sie obfuskiert werden.

In Kapitel 4 wird dargestellt, wie durch Heuristiken und aktive Methoden Adressen entsprechend ihrer Eigentümer gruppiert werden können. Passive Methoden, wie die beschriebene Multi-Input oder Wechselgeldheuristik, ermöglichen die Zusammenführung von Adressen und damit die Zusammengehörigkeit von Transaktionen und deren Teilnehmern.

Kapitel 6 stellt im Folgenden dar, wie diese Gruppen real-weltlichen Entitäten zugeordnet werden können. Es zeigt sich, dass Selbstidentifikation durch soziale Medien und Foren und die Zuordnung von IP-Adressen zu Transaktionen die Pseudonymität des Bitcoinsystems gefährden können.

Schlussendlich wird über die Analyse von Finanzströmen in Kapitel 7 gezeigt, dass nicht nur Nutzer, sondern auch Transaktionen bzw. deren Werte im Bitcoinsystem verfolgt werden können.

Zusammenfassend lässt sich sagen: Die Anonymität des Bitcoinsystems kann aus verschiedenen Richtungen eingeschränkt werden. Die Analysemethoden haben jedoch ihre Grenzen und können die Anonymität des Bitcoinsystems nicht endgültig überwinden. Die Anonymität eines Nutzers hängt insbesondere von seiner Bitcoin-Nutzung und der Nutzung seines Umfelds ab. Ein bewusster anonymer Umgang mit der Kryptowährung ist die Grundlage für eine solide Privatsphäre.

Anhang A: Dekodierte Bitcoin Transaktion

```
1   {
2      "version": 1,
3      "locktime": 0,
4      "vin": [
5      {
6         "txid":
7         "7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18",
8         "vout": 0,
9         "scriptSig" :
10        "3045022100884d142d86652a3f47ba4746ec719bbfbd040a570b1deccbb6498c7
11        5c4ae24cb02204b9f039ff08df09cbe9f6addac960298cad530a863ea8f53982c0
12        9db8f6e3813[ALL]0484ecc0d46f1918b30928fa0e4ed99f16a0fb4fde0735e7ad
13        e8416ab9fe423cc5412336376789d172787ec3457eee41c04f4938de5cc17b4a10
14        fa336a8d752adf",
15        "sequence": 4294967295
16     }
17     ],
18     "vout": [
19     {
20        "value": 0.01500000,
21        "scriptPubKey": "OP_DUP OP_HASH160
22        ab68025513c3dbd2f7b92a94e0581f5d50f654e7 OP_EQUALVERIFY OP_CHECKSIG"
23     },
24     {
25        "value": 0.08450000,
26        "scriptPubKey": "OP_DUP OP_HASH160
27        7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY OP_CHECKSIG",
28     }
29     ]
30     }
31  ]
```

[6, S. 118 f.]

Anhang B: Statistische Untersuchung der Wechselgeldhinweise

Statistische Untersuchungen der Wechselgeldhinweise	Anwendbarkeit		Adressreduktionssrate		Genauigkeit		Ähnlichkeit	
	Liu [59]	Moeser [31]	Liu [59]	Zhang [50]	Moeser [31] TPR	Moeser [31] FPR	Remy [21] NMI	Remy [21] aNMI
Universell Anwendbar								
Shadow Address	28,29%		46,2%	53,02%			0,47	0,15
Wiederverwendung von Adressen als Output	30,62%		46,4%	53,23%				
Adresstypen	39,53%	36,9%	46,5%		0,237	0,031		
Runde Beträge (N=6)	12,84%	10,4%	43,9%		0,211	0,005		
Optimale Wechselgeldheuristik	11,08%	13,3%	44,0%		0,306	0,026		
Fingerprinting								
Anzahl der Transaktionsin- und Outputs		56,8%			0,263	0,107		
Reihenfolge der Transaktionsin- und Outputs		44,3%			0,262	0,053		
Locktime		36,3%			0,307	0,003		
Version		32,0%			0,245	0,004		
Opt-In Replace by Fee		11,4%			0,075	0,003		
SegWit		26,0%			0,191	0,021		
Verwendung unbestätigter Transaktionsoutputs		21,4%			0,100	0,061		
Absolute Transaktionsgebühr		30,5%			0,117	0,025		
Relative Transaktionsgebühr		20,4%			0,042	0,008		
Multisignatur		15,4%			0,140	0,001		
Adresstypen		39,2%			0,294	0,023		
Kurze Signaturen								

Glossar

Base58 Ein Datenkodierungsformat. Es nutzt alphanumerische Zeichen, verzichtet jedoch auf Zeichen die verwechselt werde könnten. Es wird häufig für die Kodierung von P2PKH und P2SH Adressen verwendet.

Batching-Transaktion Batching-Transaktionen beziehen sich auf die Praxis des Bündelns mehrerer kleiner Überweisungen zu einer einzigen Bitcointransaktion. Diese Methode wird oft von Kryptowährungsbörsen und -dienstleistern verwendet, um die Anzahl der Transaktionen auf der Blockchain zu reduzieren und die Transaktionsgebühren zu senken.

Bech32 Ein Datenkodierungsformat. Es nutzt einen kleineren alphanumerischen Zeichensatz als Base58 und ist daher besser für QR-Codes geeignet. Diese Kodierung wird häufig für SegWit bzw. Taproot Adressen verwendet.

digitale Signatur Eine digitale Signatur sind Daten, die zur Authentizitätsprüfung anderer Daten dienen. Digitale Signaturen sind ein Anwendungsfall asymmetrischer Kryptographie. Die Signatur wird mithilfe des privaten Schlüssels erstellt und kann mit dem öffentlichen Schlüssel geprüft werden.

Distinguished Encoding Rules Ein definierter Standard, den Bitcoin verwendet, um ECDSA-Signaturen zu kodieren [83].

Double Spending Double Spending ist das mehrfache Ausgeben eines Geldbetrages. Bitcoin schützt vor "Double Spending", indem es jede Transaktion verifiziert, um sicherzustellen, dass der Input für die Transaktion nicht bereits ausgegeben wurde [6, S. XXV].

Elliptic Curve Digital Signature Algorithm Ein kryptographischer Algorithmus, der von Bitcoin verwendet wird, um sicherzustellen, dass Gelder nur von ihren rechtmäßigen Eigentümern ausgegeben werden können.[6, S. XXV].

Hashwert Ein Hashwert ist das Ergebnis einer Hashfunktion, welche aus Daten beliebiger Länge (Urbild) einen Wert fester Länge errechnet.

Heuristik Eine Heuristik ist eine auf Erfahrung basierende Methode oder Technik, die dazu dient, Probleme zu lösen oder Entscheidungen zu treffen, indem sie eine Näherungslösung liefert, die möglicherweise nicht optimal, aber für den gegebenen Kontext praktikabel und effizient genug ist.

Miner Ein Netzwerkknoten des Bitcoin Peer-to-Peer Netzwerks, der das Ziel verfolgt, gültige Blöcke der Blockchain anzufügen.

Miningpool Ein Zusammenschluss von mehreren individuellen Bitcoin-Minern, die ihre Rechenleistung kombinieren, um die Wahrscheinlichkeit zu erhöhen, einen Block erfolgreich zu finden.

Mt. Gox Die Kryptowährungsbörse Mt. Gox, die einst zu den größten der Welt gehörte, brach 2014 nach einem massiven Hack zusammen. Mt. Gox war für Schlüsselimportfeature bekannt. Dies führte im folgenden zu einem 'Superclustering', bei dem ein einziges Unternehmen einen beträchtlichen Teil der Bitcoin-Bestände kontrollierte.

Multisignatur Multisignatur bedeutet, dass mehr als ein Schlüssel benötigt wird, um eine Bitcoin-Transaktion zu autorisieren [6, S. XXVII].

Pay to Witness Public Key Hash Ist das Segregated Witness Äquivalent zu P2PKH. Dabei werden Daten zur Validierung (hier Öffentlicher Schlüssel und Signatur) der Transaktion aus der Transaktion ausgelagert.

Pay to Witness Script Hash Ist das Segregated Witness Äquivalent zu P2SH. Dabei werden Daten zur Validierung der Transaktion (hier Skripte, öffentliche Schlüssel und Signaturen) aus der Transaktion ausgelagert.

Peer-to-Peer Eine Netzwerkarchitektur, bei der alle Rechner des Netzwerks ohne zentrale Instanz untereinander kommunizieren. Peer-to-Peer stellt ein Gegenstück der Client-Server-Architektur dar.

Segregated Witness Ein vorgeschlagenes Upgrade des Bitcoin-Protokolls, das durch eine technologische Innovation die Signaturdaten von den Bitcoin-Transaktionen trennt. Segregated Witness ist ein vorgeschlagener Soft Fork; eine Änderung, die Bitcoins Protokollregeln technisch restriktiver macht [6, S. XXX].

Tor-Netzwerk Das Tor-Netzwerk ist ein anonymisierendes Netzwerk, das es Nutzern ermöglicht, ihre Online-Aktivitäten zu verschleiern und ihre Privatsphäre zu schützen, indem es den Datenverkehr über mehrere Server leitet, um die Herkunft und Zieladresse zu verschleiern..

Transaktionsgebühr Der Absender einer Transaktion bezahlt oft eine Gebühr an das Netz für die Bearbeitung der angeforderten Transaktion. Die Transaktionsgebühr wird dem Miner des Blocks in der Coinbase-Transaktion übertragen [6, S. XXV].

Wallet Software zur Verwaltung von Bitcoin-Vermögensbeständen, die private und öffentliche Schlüssel speichert und eine benutzerfreundliche Erstellung von Transaktionen ermöglicht.

Zeit- und Platzkomplexität Die Platzkomplexität bezeichnet den für die Ausführung benötigten Platz und die Zeitkomplexität die Anzahl der für die Ausführung erforderlichen Operationen. Die Zeitkomplexität wird anhand der Anzahl der Iterationen gemessen, die ein Algorithmus zur Ausführung benötigt, und die Platzkomplexität wird anhand des für die Ausführung des Algorithmus benötigten Speichers oder Platzes gemessen [84, S. 2].

Literaturverzeichnis

[1] A. Dörner und A. Neuhaus. „Was Sie zu den neuen Bitcoin-ETFs jetzt wissen müssen“. (2024), Adresse: https://www.handelsblatt.com/finanzen/maerkte/devisen-rohstoffe/bitcoin-etf-genehmigt-was-sie-wissen-muessen-02/100005937.html (besucht am 10. 02. 2024).

[2] BITCRIME. „Verfolgung und Prävention organisierter Finanzkriminalität mit virtuellen Währungen (BITCRIME)“. (2017), Adresse: https://www.bitcrime.de/ (besucht am 10. 02. 2023).

[3] T. PROJECT. „TITANIUM PROJECT“. (2020), Adresse: https://titanium-project.eu/ (besucht am 10. 02. 2023).

[4] dence GmbH, *dence blockchain investigator*, 5. Jan. 2023. Adresse: https://www.dence.de/de/products/virtual_currencies.

[5] P. Nakamoto, „Bitcoin: A Peer-to-Peer Electronic Cash System“, 2008.

[6] A. M. Antonopoulos, *Mastering Bitcoin: Programming the Open Blockchain*, 2nd. O'Reilly Media, Inc., 2017, ISBN: 1491954388.

[7] P. Wuille, J. Nick und A. Towns. „Taproot: SegWit version 1 spending rules“. (2020), Adresse: https://github.com/bitcoin/bips/blob/master/bip-0341.mediawiki (besucht am 13. 01. 2023).

[8] P. Wuille, J. Nick und A. Towns. „Validation of Taproot Scripts“. (2020), Adresse: https://github.com/bitcoin/bips/blob/master/bip-0342.mediawiki (besucht am 13. 01. 2023).

[9] Pseudonym:0xB10C und Bitrefill. „Taproot spending Transactions“. (2024), Adresse: https://transactionfee.info/charts/transactions-spending-taproot/ (besucht am 09. 02. 2024).

[10] P. Koshy, D. Koshy und P. McDaniel, „An Analysis of Anonymity in Bitcoin Using P2P Network Traffic“, in *Financial Cryptography and Data Security*, N. Christin und R. Safavi-Naini, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, S. 469–485, ISBN: 978-3-662-45472-5.

[11] A. Pfitzmann und M. Hansen, *A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management*, http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf, v0.34, Aug. 2010. Adresse: http://dud.inf.tu-dresden.de/literatur/Anon%5C_Terminology%5C_v0.34.pdf.

[12] P. Läuchli, „Grundlagen der Graphentheorie“, in *Algorithmische Graphentheorie*. Basel: Birkhäuser Basel, 1991, S. 5–10, ISBN: 978-3-0348-5635-5. DOI: 10.1007/978-3-0348-5635-5_2. Adresse: https://doi.org/10.1007/978-3-0348-5635-5_2.

[13] P. Tittmann, „Graphen“, in *Einführung in die Kombinatorik*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, S. 131–164, ISBN: 978-3-662-58921-2. DOI: 10.1007/978-3-662-58921-2_5. Adresse: https://doi.org/10.1007/978-3-662-58921-2_5.

[14] S. Fortunato und C. Castellano, „Community Structure in Graphs“, in *Computational Complexity: Theory, Techniques, and Applications*, R. A. Meyers, Hrsg. New York, NY: Springer New York, 2012, S. 490–512, ISBN: 978-1-4614-1800-9. DOI: 10.1007/978-1-4614-1800-9_33. Adresse: https://doi.org/10.1007/978-1-4614-1800-9_33.

[15] P. Szalachowski, *(Short Paper) Towards More Reliable Bitcoin Timestamps*, 2018. arXiv: 1803.09028 [cs.CR].

[16] C. Decker und R. Wattenhofer, „Information propagation in the Bitcoin network“, in *IEEE P2P 2013 Proceedings*, 2013, S. 1–10. DOI: 10.1109/P2P.2013.6688704.

[17] V. B. Mišić, J. Mišić und X. Chang, „Making Transaction Propagation More Efficient: Deferred Transaction Relay in Bitcoin“, in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, S. 1–6. DOI: 10.1109/GLOBECOM42002.2020.9322130.

[18] H. Yousaf, G. Kappos und S. Meiklejohn, *Tracing Transactions Across Cryptocurrency Ledgers*, 2019. arXiv: 1810.12786 [cs.CR].

[19] S. Ghesmati, W. Fdhila und E. Weippl, „Studying Bitcoin Privacy Attacks and Their Impact on Bitcoin-Based Identity Methods“, in *Business Process Management: Blockchain and Robotic Process Automation Forum*, J. González Enríquez, S. Debois, P. Fettke, P. Plebani, I. van de Weerd und I. Weber, Hrsg., Cham: Springer International Publishing, 2021, S. 85–101, ISBN: 978-3-030-85867-4.

[20] F. Reid und M. Harrigan, *An Analysis of Anonymity in the Bitcoin System*, 2012. arXiv: 1107.4524 [physics.soc-ph].

[21] C. Remy, B. Rym und L. Matthieu, „Tracking Bitcoin Users Activity Using Community Detection on a Network of Weak Signals“, in *Complex Networks & Their Applications VI*, C. Cherifi, H. Cherifi, M. Karsai und M. Musolesi, Hrsg., Cham: Springer International Publishing, 2018, S. 166–177, ISBN: 978-3-319-72150-7.

[22] M. Harrigan und C. Fretter, „The Unreasonable Effectiveness of Address Clustering“, S. 368–373, 2016. DOI: 10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0071.

[23] G. Maxwell. „CoinJoin: Bitcoin privacy for the real world“. (2013), Adresse: https://bitcointalk.org/index.php?topic=279249.0 (besucht am 13. 12. 2023).

[24] H. Schnoering und M. Vazirgiannis, *Heuristics for Detecting CoinJoin Transactions on the Bitcoin Blockchain*, 2023. arXiv: 2311.12491 [cs.CR].

[25] S. Goldfeder, H. Kalodner, D. Reisman und A. Narayanan, *When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies*, 2017. arXiv: 1708.04748 [cs.CR].

[26] M. Möser und R. Böhme, „Join Me on a Market for Anonymity“, 2016. Adresse: https://api.semanticscholar.org/CorpusID:14471821.

[27] M. Haywood, „Improving Privacy Using Pay-to-EndPoint (P2EP)“, Aug. 2018. Adresse: https://blog.blockstream.com/en-improving-privacy-using-pay-to-endpoint/ (besucht am 13. 12. 2023).

[28] S. Ghesmati, A. Kern, A. Judmayer und N. S. and, *Unnecessary Input Heuristics & PayJoin Transactions*, Cryptology ePrint Archive, Paper 2022/589, https://eprint.iacr.org/2022/589, 2022. DOI: 10.1007/978-3-030-78642-7_56. Adresse: https://eprint.iacr.org/2022/589.

[29] N. Ilk, G. Shang, S. Fan und J. Zhao, „Stability of Transaction Fees in Bitcoin: A Supply and Demand Perspective“, *MIS Quarterly*, Jg. 45, S. 563–692, Juni 2021. DOI: 10.25300/MISQ/2021/15718.

[30] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer und S. Capkun, „Evaluating User Privacy in Bitcoin“, in *Financial Cryptography and Data Security*, A.-R. Sadeghi, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, S. 34–51, ISBN: 978-3-642-39884-1.

[31] M. Möser und A. Narayanan, *Resurrecting Address Clustering in Bitcoin*, 2022. arXiv: 2107.05749 [cs.CR].

[32] S. Meiklejohn u. a., „A fistful of bitcoins: characterizing payments among men with no names“, *Proceedings of the 2013 conference on Internet measurement conference*, 2013. Adresse: https://api.semanticscholar.org/CorpusID:2224198.

[33] P. Todd. „Blocking uneconomical UTXO creation“. (2013), Adresse: https://bitcoin-development.narkive.com/X6QeWq8w/elopment-blocking-uneconomical-utxo-creation (besucht am 09. 02. 2024).

[34] J. D. Nick, „Data-driven de-anonymization in bitcoin“, Magisterarb., ETH-Zürich, 2015.

[35] H. Kaldoner u. a. „BlockSci: Design and applications of a blockchainanalysis platform“. (2020), Adresse: https://citp.github.io/BlockSci/reference/heuristics/change.html#static-heuristics (besucht am 13. 12. 2023).

[36] Electrum Technologies GmbH, *Electrum Wallet*, Version 4.4.6, 20. Dez. 2023. Adresse: https://electrum.org/.

[37] N. Reiff, S. Anderson und T. Li. „Why Is Bitcoin Volatile?“ (2024), Adresse: https://www.investopedia.com/articles/investing/052014/why-bitcoins-value-so-volatile.asp (besucht am 18. 02. 2024).

[38] D. A. Harding und P. Todd, *Opt-in Full Replace-by-Fee Signaling*, 2015. Adresse: https://github.com/bitcoin/bips/wiki/Comments:BIP-0125 (besucht am 05. 01. 2023).

[39] *Privacy-Bitcoin Wiki*. Adresse: https://en.bitcoin.it/Privacy (besucht am 13. 12. 2023).

[40] K. Atalas. „Lexicographical Indexing of Transaction Inputs and Outputs“. (2015), Adresse: https://github.com/bitcoin/bips/blob/master/bip-0069.mediawiki (besucht am 14. 12. 2023).

[41] Bitcoin Project, *BitcoinCore*, Version 25.0, 31. Dez. 2023. Adresse: https://bitcoin.org/en/bitcoin-core/.

[42] M. Friedenbach, N. Dorier und J. Kinoshita. „Relative lock-time using consensus-enforced sequence numbers“. (2015), Adresse: https://github.com/bitcoin/bips/blob/master/bip-0068.mediawiki (besucht am 14. 12. 2023).

[43] D. Harding und P. Todd. „Opt-in Full Replace-by-Fee Signaling“. (2015), Adresse: https://github.com/bitcoin/bips/blob/master/bip-0125.mediawiki (besucht am 14. 12. 2023).

[44] E. Lombrozo, J. Lau und P. Wuille. „Segregated Witness (Consensus layer)“. (2015), Adresse: https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki (besucht am 14. 12. 2023).

[45] „Segregated Witness Wallet Development Guide“, in *Bitcoin Developer Guide*, 2023. Adresse: https://bitcoincore.org/en/segwit_wallet_dev/ (besucht am 19. 12. 2023).

[46] zkSNACKs Ltd., *Wasabi Wallet*, Version 2.0.4.1, 20. Dez. 2023. Adresse: https://wasabiwallet.io/.

[47] Pseudonym:0xB10C. „Following the Blockchain.com feerate recommendations“. (2020), Adresse: https://b10c.me/observations/03-blockchaincom-recommendations/ (besucht am 21. 12. 2023).

[48] Pseudonym:0xB10C. „Evolution of the signature size in Bitcoin“. (2020), Adresse: https://b10c.me/blog/006-evolution-of-the-bitcoin-signature-length/ (besucht am 21. 12. 2023).

[49] A. A. Chow, P. Wuille und G. Maxwell. „Always create signatures with Low R values“. (2018), Adresse: https://github.com/bitcoin/bitcoin/pull/13666 (besucht am 21. 12. 2023).

[50] Y. Zhang, J. Wang und J. Luo, „Heuristic-Based Address Clustering in Bitcoin“, *IEEE Access*, Jg. 8, S. 210 582–210 591, 2020. DOI: 10.1109/ACCESS.2020.3039570.

[51] C. Zhao und Y. Guan, „A GRAPH-BASED INVESTIGATION OF BITCOIN TRANSACTIONS“, in *Advances in Digital Forensics XI*, G. Peterson und S. Shenoi, Hrsg., Cham: Springer International Publishing, 2015, S. 79–95, ISBN: 978-3-319-24123-4.

[52] Y. Gong, K.-P. Chow, H.-F. Ting und S.-M. Yiu, „Analyzing the Error Rates of Bitcoin Clustering Heuristics“, in *Advances in Digital Forensics XVIII*, G. Peterson und S. Shenoi, Hrsg., Cham: Springer International Publishing, 2022, S. 187–205, ISBN: 978-3-031-10078-9.

[53] H. Kalodner u. a., „BlockSci: Design and applications of a blockchain analysis platform“, in *29th USENIX Security Symposium (USENIX Security 20)*, USENIX Association, Aug. 2020, S. 2721–2738, ISBN: 978-1-939133-17-5. Adresse: https://www.usenix.org/conference/usenixsecurity20/presentation/kalodner.

[54] Streetside Development, LLC, *Samourai Wallet*, Version 00.99.98i, 31. Dez. 2023. Adresse: https://samouraiwallet.com/.

[55] „dusting attack“. eprint: Twitter. (25. Okt. 2018), Adresse: https://twitter.com/SamouraiWallet/status/1055345822076936192 (besucht am 29. 06. 2018).

[56] K. Lee, S. Ulkuatam, P. Beling und B. Scherer, „Generating Synthetic Bitcoin Transactions and Predicting Market Price Movement Via Inverse Reinforcement Learning and Agent-Based Modeling“, *Journal of Artificial Societies and Social Simulation*, Jg. 21, Juni 2018. DOI: 10.18564/jasss.3733.

[57] F. Liu u. a., „Bitcoin Address Clustering Based on Change Address Improvement“, *IEEE Transactions on Computational Social Systems*, S. 1–12, 2023. DOI: 10.1109/TCSS.2023.3239031.

[58] D. Powers, „Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation“, *Mach. Learn. Technol.*, Jg. 2, Jan. 2008.

[59] F. Liu, *Bitcoin Address Clustering Based on Change Address Improvement*, 2022. DOI: 10.21227/apzc-be43. Adresse: https://dx.doi.org/10.21227/apzc-be43.

[60] M. Fleder, M. S. Kester und S. Pillai, *Bitcoin Transaction Graph Analysis*, 2015. arXiv: 1502.01657 [cs.CR].

[61] F. Schräer. „Twitter-Hack mit Bitcoin-Betrug: 18-Jähriger bekennt sich schuldig“. (2021), Adresse: https://www.heise.de/news/Twitter-Hack-mit-Bitcoin-Betrug-18-Jaehriger-bekennt-sich-schuldig-5989917.html (besucht am 31. 01. 2023).

[62] P. H. Cording und K. Lyngby, „Algorithms for web scraping“, *Lyngby: Technical University of Denmark.[Consulta: 14 Junio 2017]*, 2011.

[63] M. Spagnuolo, F. Maggi und S. Zanero, „BitIodine: Extracting Intelligence from the Bitcoin Network“, in *Financial Cryptography and Data Security*, N. Christin und R. Safavi-Naini, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, S. 457–468, ISBN: 978-3-662-45472-5.

[64] G. Andresen. „Python code for validating bitcoin address“. (2010), Adresse: https://bitcointalk.org/index.php?topic=1026.0 (besucht am 06. 02. 2024).

[65] K. Peters, E. Rasure und V. Velasquez. „A History of Bitcoin Hard Forks“. (2023), Adresse: https://www.investopedia.com/tech/history-bitcoin-hard-forks/ (besucht am 06. 02. 2024).

[66] J. Frankenfield, E. Rasure und A. Courage. „A History of Bitcoin Hard Forks“. (2023), Adresse: https://www.investopedia.com/terms/b/bitcoin-cash.asp (besucht am 06. 02. 2024).

[67] W. Bazán-Palomino, „Bitcoin and Its Offspring: A Volatility Risk Approach“, in *Advanced Studies of Financial Technologies and Cryptocurrency Markets*, L. Pichl, C. Eom, E. Scalas und T. Kaizoji, Hrsg. Singapore: Springer Singapore, 2020, S. 233–256, ISBN: 978-981-15-4498-9. DOI: 10.1007/978-981-15-4498-9_13. Adresse: https://doi.org/10.1007/978-981-15-4498-9_13.

[68] Y. Kwon, H. Kim, J. Shin und Y. Kim, *Bitcoin vs. Bitcoin Cash: Coexistence or Downfall of Bitcoin Cash?*, 2019. arXiv: 1902.11064 [cs.CR].

[69] D. Kaminsky. „Black Ops of TCP/IP 2011“, Chaos Computer Club e.V. (2011), Adresse: https://media.ccc.de/v/cccamp11-4555-black_ops_of_tcpip_2011-en.

[70] A. Biryukov, D. Khovratovich und I. Pustogarov, *Deanonymisation of clients in Bitcoin P2P network*, 2014. arXiv: 1405.7418 [cs.CR].

[71] BITNODES. „Reachable Bitcoin Nodes“. (2024), Adresse: https://bitnodes.io/dashboard/1y/ (besucht am 05. 02. 2024).

[72] Pseudonym:0xB10C. „LinkingLion: An entity linking Bitcoin transactions to IPs?“ (2023), Adresse: https://b10c.me/observations/06-linkinglion/ (besucht am 06. 02. 2023).

[73] A. Biryukov, D. Khovratovich und I. Pustogarov, *Deanonymisation of clients in Bitcoin P2P network*, 2014. arXiv: 1405.7418 [cs.CR].

[74] B. W. Services. „Broadcast Your Transaction“. (2023), Adresse: https://live.blockcypher.com/btc/pushtx/ (besucht am 06. 02. 2024).

[75] Blockchain.com. „Broadcast a transaction“. (2024), Adresse: https://www.blockchain.com/explorer/assets/btc/broadcast-transaction (besucht am 06. 02. 2024).

[76] A. Sharma, A. Agrawal, A. Bhatia und K. Tiwari, „Bitcoin's Blockchain Data Analytics: A Graph Theoretic Perspective“, in *Advanced Information Networking and Applications*, L. Barolli, F. Hussain und T. Enokido, Hrsg., Cham: Springer International Publishing, 2022, S. 459–470, ISBN: 978-3-030-99584-3.

[77] J. Wu, J. Liu, Y. Zhao und Z. Zheng, „Analysis of cryptocurrency transactions from a network perspective: An overview“, *Journal of Network and Computer Applications*, Jg. 190, S. 103 139, 2021, ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2021.103139. Adresse: https://www.sciencedirect.com/science/article/pii/S1084804521001557.

[78] M. Möser, R. Böhme und D. Breuker, „Towards Risk Scoring of Bitcoin Transactions“, in *Financial Cryptography and Data Security*, R. Böhme, M. Brenner, T. Moore und M. Smith, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, S. 16–32, ISBN: 978-3-662-44774-1.

[79] R. Anderson, I. Shumailov und M. Ahmed, „Making Bitcoin Legal“, in *Security Protocols XXVI*, V. Matyáš, P. Švenda, F. Stajano, B. Christianson und J. Anderson, Hrsg., Cham: Springer International Publishing, 2018, S. 243–253, ISBN: 978-3-030-03251-7.

[80] T. Tironsakkul, M. Maarek, A. Eross und M. Just, *Probing the Mystery of Cryptocurrency Theft: An Investigation into Methods for Taint Analysis*, 2019. arXiv: 1906.05754 [cs.CR].

[81] T. Ruffing, P. Moreno-Sanchez und A. Kate, „CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin“, in *Computer Security - ESORICS 2014*, M. Kutyłowski und J. Vaidya, Hrsg., Cham: Springer International Publishing, 2014, S. 345–364, ISBN: 978-3-319-11212-1.

[82] F. Kleinwort, W. Posdorfer und J. Edinger, „Analyzing the Effect of Taproot on Bitcoin Deanonymization“, in *2023 IEEE 43rd International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2023, S. 25–30. DOI: 10.1109/ICDCSW60045.2023.00011.

[83] R. LEARN. „DER Format“. (2024), Adresse: https://river.com/learn/terms/d/der/ (besucht am 18. 01. 2023).

[84] A. O. Ngu, „Dimensional Complexity & Algorithmic Efficiency“, *International Journal of Modern Nonlinear Theory and Application*, Jg. 11, Nr. 01, S. 1–10, 2022, ISSN: 2167-9487. DOI: 10.4236/ijmnta.2022.111001. Adresse: http://dx.doi.org/10.4236/ijmnta.2022.111001.

Eidesstattliche Erklärung

Hiermit versichere ich – Lukas Schöne – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.



Mittweida, 23. Februar 2024 — Lukas Schöne

Ort, Datum — Lukas Schöne