

**Behind the Chain of Obscurity:
Methodologies for Cryptocurrency
Forensic Analysis**

Tin Tironsakkul

SUBMITTED FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

HERIOT-WATT UNIVERSITY



DEPARTMENT OF MATHEMATICS,
SCHOOL OF MATHEMATICAL AND COMPUTER SCIENCES.

September, 2022

The copyright in this thesis is owned by the author. Any quotation from the thesis or use of any of the information contained in it must acknowledge this thesis as the source of the quotation or information.

Abstract

Bitcoin and alternative cryptocurrencies are decentralised digital currencies that allow users to anonymously exchange money without requiring the presence of a trusted third party. The privacy components of cryptocurrency can facilitate illegal activities and present new challenges for cybercrime forensic analysis. Tackling such challenges motivates new research interest in cryptocurrency tracking. This thesis explores and proposes novel methodologies and improvements to existing cryptocurrency tracking and analysis methodologies.

Our first contribution explores the most commonly used cryptocurrency tracking methodology named *Taint Analysis* and investigates a potential improvement to the methodology's tracking precision with the implementation of address profiling. We also introduce two context-based taint analysis strategies and hypothesise behaviours related to the tracked Bitcoins context to create a set of evaluation metrics. We conducted an experiment using sample data from known illegal Bitcoin cases to illustrate and evaluate the methodology, and the results reveal distinct transaction behaviours in tracking between the results with and without address profiling for all of the metrics. Our second contribution proposes a cryptocurrency tracking methodology named *Address Taint Analysis* that is capable of tracking zero-taint coins created by Privacy-Enhancing Technologies (PETs) called centralised mixer services, which are untrackable with taint analysis tracking. Our results indicate that our proposed address taint analysis can trace the zero-taint Bitcoins from nine well-known mixer services back to the original Bitcoins. Our third contribution investigates and proposes a detection method for Wasabi Wallet's CoinJoin transactions, which is one of the most recent well-known PET services. Our fourth contribution introduces an open-source library for cryptocurrency tracking and analysis named, *TaintedTX*, that we utilised to perform our research experiments. The library supports a variety of taint analysis strategies that users can select to track targeted transactions or addresses. The library also includes a compilation of utility functions for address clustering, website scraping, transaction and address classifications.

*I dedicate this thesis to my parents,
Narong and Ornthip, for their never-ending support since the beginning of my life.*

Acknowledgements

I would like to express my greatest gratitude to all of my three supervisors, Manuel Maarek, Andrea Eross and Mike Just, who provided education and research guidance and assisted me throughout my research study. The work in this thesis would not be possible without their contributions and advice. Thanks also to Sasa Radomirovic, who provided valuable feedback on our work.

I would like to extend my gratitude to Malte Möser, Rainer Böhme, Dominic Breuker, Sean Foley, Thibault de Balthasar, Julio Hernandez-Castro, Rolf van Wegberg, Jan-Jaap Oerlemans, and Oskar van Deventer for their inspirational work and for providing valuable research data.

Additionally, I thank every researcher who provides a contribution to the progress of cryptocurrency research. Without their research and finding, it would be impossible to establish the foundation in this work. And naturally, I also express my admiration for Satoshi Nakamoto, who provides an ingenious invention of a financial instrument without ever claiming the real-life fame and also riches from most of their Bitcoin treasure that still sits idle until today.

The country of Scotland and Heriot-Watt University also provide me with a superb learning resource, experiment instruments, work experience, and of course, an exceptional research environment. I quite like the weather here, even if many seem to complain about it.

Finally, I would like to express my utmost love to my parents, Narong Tiron-sakkul and Ornthip Ruangkul, my two younger brothers, Tham and Tat Tiron-sakkul, for their support and encouragement in both my study and time of need.

Research Thesis Submission

Please note this form should be bound into the submitted thesis.

Name:	Tin Tironsakkul		
School:	School of Mathematical and Computer Sciences		
Version:	Final	Degree Sought:	PhD

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1. The thesis embodies the results of my own work and has been composed by myself
2. Where appropriate, I have made acknowledgement of the work of others
3. The thesis is the correct version for submission and is the same version as any electronic versions submitted*.
4. My thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5. I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6. I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

ONLY for submissions including published works

Please note you are only required to complete the Inclusion of Published Works Form (page 2) if your thesis contains published works)

7. Where the thesis contains published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) these are accompanied by a critical review which accurately describes my contribution to the research and, for multi-author outputs, a signed declaration indicating the contribution of each author (complete)
8. Inclusion of published outputs under Regulation 6 (9.1.2) or Regulation 43 (9) shall not constitute plagiarism.

* Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.

Signature of Candidate:	Tin Tironsakkul	Date:	30/9/2022
-------------------------	-----------------	-------	-----------

Submission

Submitted By (<i>name in capitals</i>):	TIN TIRONSAKKUL
Signature of Individual Submitting:	Tin Tironsakkul
Date Submitted:	30/9/2022

For Completion in the Student Service Centre (SSC)

Limited Access	Requested	Yes		No		Approved	Yes		No	
<i>E-thesis Submitted (mandatory for final theses)</i>										
Received in the SSC by (<i>name in capitals</i>):						Date:				

Inclusion of Published Works

Please note you are only required to complete the Inclusion of Published Works Form if your thesis contains published works under Regulation 6 (9.1.2)

Declaration

This thesis contains one or more multi-author published works. In accordance with Regulation 6 (9.1.2) I hereby declare that the contributions of each author to these publications is as follows:

Citation details	Tin Tironsakkul, Manuel Maarek, Andrea Eross, Mike Just, Tracking Mixed Bitcoins., Data Privacy Management, Cryptocurrencies and Blockchain Technology. DPM 2020, CBT 2020 (pp. 447-457). (Lecture Notes in Computer Science; Vol. 12484). Springer. (2020)
Author 1	Main author
Author 2	Supervisor authors
Signature:	Tin Tironsakkul
Date:	30/9/2022

Citation details	Tin Tironsakkul, Manuel Maarek, Andrea Eross, and Mike Just. 2022. The Unique Dressing of Transactions: Wasabi CoinJoin Transaction Detection. In Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference (EICC '22). Association for Computing Machinery, New York, NY, USA, 21–28.
Author 1	Main author
Author 2	Supervisor authors
Signature:	Tin Tironsakkul
Date:	30/9/2022

Citation details	Tin Tironsakkul, Manuel Maarek, Andrea Eross, Mike Just, Context Matters: Methods for Bitcoin Tracking (Under Review)
Author 1	Main author
Author 2	Supervisor authors
Signature:	Tin Tironsakkul
Date:	30/9/2022

Please included additional citations as required.

Internal Examiners Declaration Form
 (This form must be typed and all sections completed)

Candidate's Name:	Tin Tironsakkul	Heriot-Watt Person ID:	H00209479
School:	MACS	Degree Sought:	PhD
Campus: <i>(If off-campus please state location)</i>	Edinburgh		

Declaration

1. I confirm that the corrections to the thesis of the above named have been carried out to the satisfaction of the examiners Yes No N/A
2. I confirm that the Joint Examiners Report Form states recommendation (c) - 'Award degree following satisfactory completion of significant corrections to the satisfaction of the Internal Examiner' Yes No

If yes,


- Please provide details below to demonstrate that the particular corrections are satisfactory.
- Confirm that the corrections have been completed within the period of time given, if not please give an explanation.

3. I confirm that the Joint Examiners Recommendation was originally: Re-submit (decision (d or e) on the previous Joint Examiners Form) Yes No
4. I confirm that thesis title has changed since the temporary thesis was submitted. **If yes,** please provide details of amended title here: Yes No

Behind the Chain of Obscurity: Methodologies for Cryptocurrency Forensic Analysis

5. I confirm that I have seen the final version of this thesis and it has been presented in accordance with University regulations. Yes

Internal Examiner

Print Name:	Michael Lones	Date:	30/09/2022
Signature:		School:	MACS

Notes

1. The Internal Examiner's Declaration Form should be submitted along with the final electronic copy of the thesis through the [Postgraduate Research Thesis Submission site](#)

Contents

List of Tables	vi
List of Figures	viii
1 Introduction	1
1.1 Research Aim, Questions, and Objectives	2
1.1.1 Research Aim	2
1.1.2 Research Questions	3
1.1.3 Research Objectives	4
1.2 Contributions	5
1.3 Ethical Considerations	7
1.4 Thesis Outline	9
2 Literature Review	12
2.1 Bitcoin Historical Overview	12
2.2 Bitcoin Internal System	13
2.2.1 Blockchain	14
2.2.2 Blocks	16
2.2.3 Addresses	17
2.2.4 Transactions	19
2.2.5 Bitcoin Decentralised Network	23
2.2.6 Mining	24
2.3 Cryptocurrency and Illegal Activities	26
2.3.1 Cryptocurrency Thefts	26
2.3.2 Illegal Goods and Services	27
2.3.3 Ransomware Attacks	28

2.3.4	Scams	29
2.3.5	Money Laundering	31
2.4	Bitcoin Tracking and Deanonymisation	31
2.4.1	Taint Analysis	31
2.4.2	Address Clustering and Deanonymisation	36
2.4.3	Existing Tracking Implementations	41
2.5	Bitcoin and Privacy Techniques	42
2.5.1	Privacy Techniques	42
2.5.2	Cryptocurrency Privacy-Enhancing Technologies (PETs)	46
2.5.3	Alternative Privacy Cryptocurrencies	49
2.6	Conclusion	50
3	Context-based Tracking	51
3.1	Context-based Bitcoin Tracking Methodology	52
3.1.1	Address Profiling	54
3.1.2	Transaction Profiling	58
3.1.3	Context-based Taint Analysis Strategy	61
3.1.4	Tracking Evaluation Metrics	64
3.2	Sample and Control Groups Collection	67
3.2.1	Theft Case Sample Selection	67
3.2.2	Control Groups Criteria and Selection	67
3.3	Results and Discussion	70
3.3.1	Address Profiling Results	71
3.3.2	Transaction Frequency (H1) Results	75
3.3.3	PETs Detection (H2) Results	77
3.3.4	Reused Address (H3) and Fresh Address (H4) Results	80
3.3.5	Number of Addresses per Transaction (H5) Results	82
3.3.6	Transaction Fee (H6) Results	83
3.3.7	Results Summary and Discussion	85
3.3.8	Reflection on Alternative Attempts	89
3.3.9	Limitations	90
3.4	Conclusion and Future Work	91

4	Zero-taint Bitcoin Tracking	93
4.1	Methodology	94
4.1.1	Address Taint Analysis	95
4.1.2	Backward Address Taint Analysis	97
4.1.3	Filtering Criteria and Address Profiling	98
4.2	Sample Cases	102
4.3	Results and Discussion	103
4.3.1	Address Taint Analysis	103
4.3.2	Backward Address Taint Analysis	105
4.3.3	Filtering Criteria	106
4.3.4	Service Address Profiling	108
4.3.5	Reflection on Alternative Attempts	110
4.3.6	Limitations	110
4.4	Conclusion and Future Work	112
5	Wasabi CoinJoin Transaction Detection	114
5.1	Wasabi Wallet	115
5.2	Data Sources and Collection	117
5.2.1	Sources and Collection	117
5.2.2	Reliability of Data per Period	119
5.3	Identification of Criteria	120
5.3.1	Examining Wasabi CoinJoin Transaction Patterns	120
5.3.2	Defining Criteria	122
5.4	Detection Results and Discussion	124
5.4.1	Detection Results	124
5.4.2	Analysis of Detection Results for the Published Transaction Periods	127
5.4.3	Analysis of Detection Results for the Sourceless Periods	128
5.4.4	Application of the Detection Method	130
5.5	Reflection on Alternative Attempts	131
5.6	Conclusion and Future Work	132

6	Conclusions	134
6.1	Summary of Contributions	134
6.2	Future Work	138
6.2.1	Change of Ownership Detection	138
6.2.2	Tainted Proportion Application	138
6.2.3	Blockchain-wide Transaction Analysis	139
6.2.4	Illegal Activities Detection	139
6.2.5	PETs Reverse-Engineering Investigation	140
6.2.6	External Blockchain Data Investigation	140
6.3	Concluding Remarks	141
A	Additional Information for Bitcoin Internal System	142
A.1	Bitcoin Transactions	142
A.2	Proof-of-work	143
A.3	Peer-to-Peer Network	143
B	Cryptocurrency Tracking Tool	146
B.1	TaintedTX Overview	146
B.1.1	System Requirements	147
B.1.2	Libraries Requirements	148
B.2	Data Schemas	148
B.2.1	Blockchain Data	149
B.2.2	Classification Data	152
B.2.3	Utility Data	153
B.2.4	Data Preparing	154
B.3	Taint Analysis Functions	154
B.3.1	Taint Analysis	154
B.3.2	Taint Analysis Strategies	159
B.3.3	Address Taint Analysis	167
B.4	Classification Functions	169
B.4.1	CoinJoin Transaction Classification	169
B.4.2	Mixer Transaction Classification	171
B.4.3	Lightning Network Transaction Classification	172

B.5 Utility Functions 173

 B.5.1 Address Clustering 174

 B.5.2 Transaction Analysis 174

 B.5.3 Control Groups Finding 176

 B.5.4 Export for Neo4J 176

 B.5.5 Website Scraping 178

B.6 Sample Use Cases 180

 B.6.1 10,000 BTC For Two Pizzas 181

 B.6.2 Miners' Bitcoin Spending Observation 182

B.7 Conclusion and Future Work 183

Bibliography **187**

List of Tables

2.1	Block data structure	16
2.2	Bitcoin transaction data structure inside a block	20
2.3	Bitcoin transaction input data structure inside a block	22
2.4	Bitcoin transaction output data structure inside a block	22
3.1	Identified service address data	56
3.2	Identified transaction profile data	58
3.3	Sample cases and control group number	68
4.1	Sample cases	102
4.2	Address tainting results	104
4.3	Sample mixer services and calibration of the filtering criteria	106
4.4	Address tainting results with filtering criteria	107
4.5	Address tainting results with filtering criteria and service address pro- filing	109
5.1	Published transaction data sets	117
5.2	General Criteria and Period-specific Criteria	123
5.3	Detection method results	124
B.1	Pseudo-code objects and functions description	147
B.2	Block data dictionary	149
B.3	Transaction height data dictionary	150
B.4	Transaction hash data dictionary	150
B.5	Address hash data dictionary	151
B.6	Transaction output data dictionary	151
B.7	Transaction input data dictionary	151

B.8	Identified address entity data dictionary	152
B.9	Identified Privacy-Enhancing Technology (PET) transaction data dictionary	153
B.10	In-Out strategy variants	159
B.11	<i>record</i> data dictionary	163
B.12	In-Out sorting (<i>variant_sorting</i>)	165
B.13	Withdrawn transaction filtering	169

List of Figures

2.1	Blockchain and fork	15
2.2	Merkle Tree of Bitcoin transactions	17
2.3	A simplified example of a Bitcoin transaction	19
2.4	Silk Road market home page [129]	28
2.5	Bitcoin payment screen of CryptoLocker ransomware [74]	29
2.6	Poison strategy	33
2.7	Haircut strategy	34
2.8	FIFO strategy	35
2.9	LIFO strategy	36
2.10	Multi-input address clustering heuristic	37
3.1	Methodology process	54
3.2	Dirty-First strategy	61
3.3	TIHO strategy	63
3.4	Percentage of the tainted Bitcoins reaching identified service and PETs	71
3.5	Proportion of identified service types.	72
3.6	Number of transactions	74
3.7	Transaction frequency (H1)	76
3.8	Percentage of transactions reaching identified PETs (H2)	78
3.9	Percentage of potential PET transactions (H2)	79
3.10	Proportion of reused and fresh addresses (H3 and H4)	81
3.11	Average number of addresses per transaction (H5)	82
3.12	Average difference between transaction fee size ratio to daily average (H6)	84
3.13	Summary of TC^{AP} results for each evaluation metric.	86

4.1	Bitcoin transfer using normal transaction and using mixer service . . .	93
4.2	Transaction taint analysis and address taint analysis	95
4.3	Address taint analysis and backward address taint analysis	98
4.4	Example of a peeling chain	100
5.1	Example of Wasabi CoinJoin transaction	116
5.2	Collected Wasabi CoinJoin transactions per day	119
5.3	Wasabi CoinJoin transaction value	120
5.4	Wasabi CoinJoin transaction inputs and outputs	121
5.5	Detection method evaluation	127
5.6	Detected Wasabi CoinJoin transactions per day	129
A.1	Bitcoin peer connection and address propagation	144
A.2	Bitcoin blockchain propagation for full nodes	145
A.3	Bitcoin block headers propagation for lightweight nodes	145
B.1	Blockchain data schemas	149
B.2	A simplified architecture diagram for the <i>TaintedTX</i> library	156
B.3	Transaction depth example	157
B.4	Haircut transaction fee distribution	159
B.5	Example of the <i>TaintedTX</i> library’s taint analysis operation	161
B.6	An example of the LIFO strategy distribution and record	162
B.7	Difference between Dirty-First and Pure Dirty-First	165
B.8	Example of a Neo4j transaction network graph with the <i>simple</i> export option	179
B.9	A Screenshot of Jupyter notebook running the taint analysis function on the Pizza payment transaction	181
B.10	Type of address entities that received portion of Pizza Bitcoins	182
B.11	Miners’ Bitcoins spending	183
B.12	Network graph of the case 3’s Dirty-First results created with Python NetworkX module	184

Glossary

ADR/ADDR	Address.
AMD	Advanced Micro Devices.
API	Application programming interface.
BIP	Bitcoin improvement proposal.
BTC	Bitcoin (as currency unit).
CPU	Central processing unit.
DFS	Depth-First Search.
ECSDA	Elliptic curve digital signature algorithm.
FBI	Federal Bureau of Investigation.
FIFO	First-In, First-Out.
HTTP	Hypertext transfer protocol.
HTTPS	Hypertext transfer protocol secure.
IP	Internet protocol.
LIFO	Last-In, First-Out.
LN	Lightning Network.
MIT	Massachusetts Institute of Technology.

P2EP	Pay-to-End-Point.
P2PKH	Pay-to-Public-Key-Hash.
P2SH	Pay-to-Script-Hash.
P2WPKH	Pay-to-Witness-Public-Key-Hash.
P2WSH	Pay-to-Witness-Script-Hash.
PETs	Privacy-Enhancing Technologies.
RAM	Random access memory.
RIPEND	RACE Integrity Primitives Evaluation Message Digest.
SegWit	Segregated Witness.
SHA	Secure hash algorithm.
SIG	Signature.
SPV	Simplified Payment Verification.
SQL	Structured query language.
TCP	Transmission control protocol.
TX	Transaction.
URL	Uniform resource locator.
USD	United States dollar.
UTXO	Unspent transaction output.

Chapter 1

Introduction

Since its first launch in 2009, Bitcoin has grown to become the most valuable electronic currency and has extraordinary effects on not only the technology industry but also the financial sector. Bitcoin's innovative implementation of a blockchain database system operated by a decentralised network without requiring a central intermediary, such as a central bank or government body to oversee its system, presents a new approach to monetary and database systems.

While many people regard Bitcoin and other cryptocurrencies as an innovative digital currency, their association with illegal activities on the Internet, such as darknet market tradings, ransomware attacks, thefts, and scams, diminishes cryptocurrencies' value and credibility to become alternatives to traditional money or to be integrated into the real-world economy. Bitcoin and other cryptocurrencies have been predominantly utilised as instruments of cybercrimes. One reason for cryptocurrencies to appeal to criminals is their privacy systems that provide anonymity to their users, for instance, in the form of pseudonymous addresses [58]. Cybercrimes related to cryptocurrency can also significantly affect the economy of the cryptocurrency market itself [73, 203]. For example, security issues of cryptocurrency service platforms that result in hacking and theft incidents – such as the hacking of the Coinrail exchange platform in 2018 caused the price of Bitcoin and other cryptocurrencies to drop by almost 10% in one hour [105].

The pseudonymous address system in Bitcoin does not provide perfect anonymity because its blockchain data is transparent and does not conceal exchanges of Bitcoins between addresses. This exposure in the Bitcoin blockchain privacy opens up

opportunities for efforts into the development of cryptocurrency forensic analysis and deanonymisation techniques either by utilising information obtained from the blockchain data or external sources [8, 51, 185]. However, research into cryptocurrency forensic analysis is still in its early stage. While there were studies [6, 128] that proposed tracking methodologies in the early days of cryptocurrency, there have been no recent significant progress or improvement to the methodologies, including the most commonly used method named taint analysis. There are two crucial challenges that affect the accuracy and practicality of taint analysis. First, taint analysis typically produces tracking results with a large number of unrelated transactions because its tracking methodology does not take into account the change in Bitcoins' ownership. Second, individuals can evade taint analysis tracking by employing Privacy-Enhancing Technologies (PETs) such as mixer services that can create zero-taint Bitcoins¹.

In this thesis, we focus on investigating solutions to these challenges and proposing novel cryptocurrency tracking methodologies. Ultimately, the methodology presented in this work can assist future research and cybersecurity in combating cybercrimes. As the number of cryptocurrencies is extensively larger than a thesis can investigate², the scope of this thesis focuses primarily on the forensic analysis of Bitcoin, which is the most valuable and utilised cryptocurrency at present.

For the remainder of this chapter, we describe our research aim, objectives, and questions in Section 1.1. Subsequently, we outline the contributions of our work in Section 1.2, discuss ethical considerations in Section 1.3, and provide the overall thesis structure in Section 1.4.

1.1 Research Aim, Questions, and Objectives

1.1.1 Research Aim

The aim of this thesis is to investigate and propose novel methodologies for the forensic analysis of Bitcoins and other similar alternative cryptocurrencies. Therefore, the research objectives focus on the investigation and development of Bitcoin

¹Bitcoins from a mixing process that have no connection back to the original deposited Bitcoins.

²As of the year 2021, there are more than 10,000 Cryptocurrencies in market [42].

forensic analysis methodology with the intention to address the research questions described in Section 1.1.2.

1.1.2 Research Questions

We define the research questions that this thesis intends to address as follows:

RQ 1

How to determine when a change of hands of Bitcoins occurs?

Transactions in Bitcoin contain only the information of Bitcoin exchanges from one pseudonymous address to another. Bitcoin addresses typically contain no personal information that can shed some light on the owners' identity. The exchange of Bitcoins' ownership can be challenging to determine because of the lack of such information in the blockchain data. This issue can significantly affect the accuracy of cryptocurrency tracking and analysis methodologies since the results can contain transactions unrelated to the targeted users. This issue raises the question of how can we then determine when Bitcoins are exchanged from one user to another.

RQ 2

How to harvest evidence for cryptocurrency forensics?

While it is possible to obtain transaction and address data from the Bitcoin blockchain, such data do not contain information that can conclusively provide us with the context of coin exchanges. This issue raises a significant question of what other sources of information can serve as evidence and valuable data for forensic processes, in addition to using the immutable data inside the cryptocurrency blockchain, to consolidate the tracking analysis and how can we harvest them.

RQ 3

How can Bitcoins PETs obscuring be deciphered?

Cryptocurrency PETs are one of the crucial obstructions to the cryptocurrency forensic progress as their primary purpose is to increase the tracking difficulty of obscured Bitcoins. For cryptocurrency tracking methodologies to be able to provide accurate results, they need to be able to decipher the actual movement and destination of obscured Bitcoins. Therefore, this issue raises an important question of how

to reverse-engineer PETs to reveal their mixing mechanism, which can be utilised as a method to track Bitcoins.

1.1.3 Research Objectives

In order to address the research questions above, we formulate our research objectives as follows:

RO 1

Develop a method to identify transaction behaviours of Bitcoin users.

As Bitcoin blockchain does not contain information that can indicate the change of Bitcoin ownerships, we hypothesise that each type of Bitcoin user (e.g., illegal users and cryptocurrency services) typically have different Bitcoin usage behaviours from the others, which would affect their transaction behaviours. Therefore, we investigate and characterise transaction behaviours to build an evaluation framework for determining the change of Bitcoin ownership to address RQ 1.

RO 2

Develop approaches for harvesting address and transaction data.

While we can harvest and utilise Bitcoin blockchain data to observe Bitcoin movement between addresses and analyse transaction behaviour to identify potential Bitcoin change of ownership, these data can not provide conclusive evidence. Hence, we investigate other sources of information, both internal (blockchain) and external (such as previous research, websites, and knowledge of common cryptocurrency practices), that can be utilised as evidence for cryptocurrency forensics to address RQ 2. The additional data will also serve as valuable information for the creation of a transaction behaviour framework to detect a change in Bitcoin ownership for RQ 1 and the creation of Bitcoin PETs “demixing” methodologies for RQ 3.

RO 3

Develop taint analysis refined with profiling of addresses and transactions.

The state-of-the-art taint analysis methodology typically produces a large number of unrelated transactions since its tracking process keeps following tainted Bit-

coins regardless of signs of ownership changes. Therefore, we investigate and experiment with taint analysis methodologies that make use of behaviour knowledge in their tracking processes. The method would provide evidence for cryptocurrency forensics by identifying transactions that change Bitcoins' ownership, which address RQ 1 and 2.

RO 4

Develop detection methods for Bitcoin PET transactions and demixing procedures.

In order for the Bitcoin tracking methodologies to be able to provide practical and accurate results for cryptocurrency forensics, the methodologies must be capable of identifying transactions or addresses involving PETs and utilising suitable solutions to discern the movement of obscured Bitcoins. Therefore, we investigate methodologies to track (demix) Bitcoins obscured by PETs for RQ 3. As Bitcoin PETs obscure actual Bitcoins' movement and exchanges, the PETs demixing methodologies will also improve the tracking methodologies in detecting change of Bitcoin ownership for RQ 1.

RO 5

Develop an open-source Python library for adaptive Bitcoin tracking operation.

The methodologies we develop to address the research questions are those that have not been presented in any published cryptocurrency tracking tool as far as we know. Therefore, we intend to collect the proposed methodologies and publish them in a Python library that is easy to use, access and modify.

1.2 Contributions

We summarise the contributions of this thesis which are the outcomes of the research objectives as follows:

- We propose several data-gathering methodologies to harvest address and transaction profile data from inside and outside the blockchain (Chapter 3), which contribute to RO 2 and 4. We utilised the harvested data to develop a new approach for Bitcoin tracking by tailoring taint analysis strategies to track tainted Bitcoins until they reach potential exit points (e.g., service and PET

entities) or change hands, which contributes to RO 1 and 3. We developed two new context-based taint analysis strategies that use the background (stolen and publicly known) of the targeted Bitcoins to track their movement instead of the arbitrary distribution ruleset in strategies previously proposed by the literature (see Section 2.4.1), which contribute to RO 3. Additionally, we designed a set of metrics to evaluate tracking accuracy based on transaction and address indicators reflecting specific behaviours (e.g., the distribution of the stolen Bitcoins and PETs' usage). The evaluation metrics show potential for detecting changes in Bitcoin ownership unidentified by the address profile data, which contribute to RO 1. The tracking and profiling methodologies presented in this chapter are compatible with most cryptocurrencies that utilise blockchain systems similar to Bitcoins. With some appropriate modification, the methodologies can be applicable for alternative cryptocurrencies that implement a different transaction system. However, the methodologies are less likely to be practical for cryptocurrencies that implement additional privacy features and obscure information utilised in the methodologies, such as Zcash's blockchain that obscure transaction amount and address³.

- We developed Bitcoin demixing methodologies called *Address Taint Analysis* that can solve the issue of tracking evasion with centralised mixer services (Chapter 4). The address taint analysis methods are capable of deciphering obscured “zero-taint” Bitcoins by well-known PET services, which contribute to RO 4. The demixing results of the proposed method also provide evidence of Bitcoin movement for forensics, which contribute to RO 2. We also introduced and calibrated withdrawal properties that can reduce a significant number of false positive results. The address taint analysis methodology is applicable for cryptocurrencies that utilise a transparent blockchain system similar to Bitcoin. The method is also usable for mixing services that employ a similar mixing mechanism as the mixer service investigated in this work.
- We performed a transaction analysis on Wasabi Wallet's CoinJoin transactions using transaction and address data harvested from the service's documentation and API as evidence (Chapter 5). To the best of our knowledge, the detection

³<https://z.cash>

method we proposed is the first to be capable of producing highly precise detection results with a minimal number of false positive transactions for all time periods. The results from this method can be utilised as evidence of Bitcoin obscuring activities, which contribute to RO 2. The detection method is also the first crucial step in the development of the demixing method for Wasabi CoinJoin mixing, which contributes to RO 4. Although we focus primarily on the Wasabi service detection in this work, the methodology and proposed detection method should be adaptable for identifying PET transactions of any type that possess a unique transaction pattern in cryptocurrencies with a transparent blockchain system.

- We developed a Python library for cryptocurrency tracking that use to run the experiments in this thesis. We published the source code of the library on a public repository. The library contains functions that are an implementation of the methodologies presented in this thesis, such as adaptive taint analysis, address taint analysis, address clustering heuristics, and PETs detection methods (Appendix B), which contributes to RO 5. Although the library is currently designed explicitly for the Bitcoin blockchain and transaction data structure, the library’s functions should be adaptable for alternative cryptocurrencies with adjustments appropriate to their data structure.

1.3 Ethical Considerations

We discuss the ethical considerations regarding our work’s experiment sample and result data in this section. We can categorise the data in our work into four main categories as follows:

Bitcoin blockchain data The Bitcoin blockchain data we use to perform all of the experiments contain transaction and pseudonymous address data of all of its users. This data is publicly available to anyone via Bitcoin clients or external blockchain explorer services. The Bitcoin blockchain data is not deanonymisable without knowledge of the physical identity behind the pseudonymous address.

External address and transaction data The service address and transaction pro-

filing data we used in Chapter 3, 4 and 5 is external information that contains the ownership identification information of associated addresses or transactions, which are the identification hash and the service entity's name. The entity information in this data holds only the name of the service and not the name of the individuals behind it. Although the profiling data is obtained from publicly available information, it can still be used to help reveal the personal information of the anonymised addresses.

Previous research data A significant portion of the address and transaction profiling data in Chapter 3 and 4 is data that we obtained from various previous research, most of which are made publicly available by the authors. We requested and received the unpublished experiments' data from the authors of previous related studies via contacts in a confidential manner in Chapter 4.

Experiment result data All of the experiment results in this work generate data that contain a list of transactions and addresses. The resulting taint analysis tracking data in Chapter 3 contains a list of transactions and addresses that potentially received Bitcoins associated with the illegal activities, whether the addresses belong to the original illegal Bitcoin owner(s) or unrelated users. The result data in Chapter 3, 4, and 5 contains a list of transactions performed by the targeted PET services and its users' addresses.

We do not hold and publish address and transaction information that can serve as identification data in our work. None of the data except for a portion of Bitcoin blockchain data in the public repository (see Appendix B) is available publicly or to anyone else outside of the team of researchers.

While there has not been any contact to request for the research data yet, we set a guideline on how to respond to such requests in the future as follows:

Blockchain data As the storage size of the blockchain data is too large to transfer, we will decline any request for a complete or a portion of blockchain data from our storage. We will redirect and assist requesters to the blockchain parsing process, external blockchain parsing tools and the data structure of our blockchain data instead.

Address and transaction profiling data We will decline any request for address and transaction profile data that we used in the experiment. As the data is

publicly available and we sufficiently describe the methods and processes we used to obtain it in the published work and respective chapters in this thesis, we will provide suggestions or clarifications on the data gathering process instead of providing the data itself. The example transaction and address identifications we provided as examples in this thesis are imitations and not related to the real data in the blockchain.

Experiment result data We will consider any request for our experiment data depending on the aspect of the requested data. We will, however, not share tracking results that contain transaction or address data either publicly or in private requests as such data contains address identifier information.

Experiment algorithm code data We published the complete source code of our Python library on a public repository that contains functions for all of the methodologies we used to perform the experiments in this work. We do not include any information that can potentially be used as identification information in the public repository (including the commit histories).

We store the above-mentioned data in one of the Linux machines located at the School of Mathematical and Computer Science in Heriot-Watt University Edinburgh campus. The data is stored in the personal folder with the read and write permission that allows only to researchers (Tin Tironsakkul, Manuel Maarek, Mike Just, Andrea Eross).

1.4 Thesis Outline

The thesis is structured into seven chapters, including this introduction chapter as follows:

- Chapter 2, we introduce and describe the internal working of the Bitcoin system that are relevant to the research area of this thesis in sufficient detail, which are blockchain, transactions, addresses, proof-of-work and nodes. We discuss illegal activities in the Bitcoin ecosystem and cover the methodologies of Bitcoin tracking and deanonymisation proposed by related works. We discuss the privacy concern involving Bitcoin and existing privacy-enhancing technologies that are created to address these concerns. Lastly, we discuss

other alternative cryptocurrencies that improved on the privacy issues presented in Bitcoin.

- Chapter 3, we investigate and present an improvement to Bitcoin tracking methodology by utilising external information for address profiling. The purpose of incorporating address profiling into the taint analysis process is to improve the precision of tracking results by reducing unessential tracking results unrelated to the targeted users' Bitcoin activities. The methodology and work in this chapter is currently under review [173]. A previous version of this work was presented at Cryptocurrency Research Conference 2019 (CRC)⁴ with the title “Probing the Mystery of Cryptocurrency Theft: An Investigation into Methods for Taint Analysis” [171].
- Chapter 4, we propose a novel tracking solution for zero-taint Bitcoins from centralised mixer/tumbling services. We demonstrate that zero-taint Bitcoins, which are immune to taint analysis tracking, can still be tracked with the tracking method we introduce called *address taint analysis*. We published and presented the work in this chapter at Cryptocurrencies and Blockchain Technology (CBT) Workshop 2021 with the title “Tracking Mixed Bitcoins” [172]. Additionally, the methodology presented is expanded from the published version by incorporating the address profiling presented in Chapter 3 to address one of the previous fundamental limitations and improve the results of address taint analysis tracking.
- Chapter 5, we propose a detection method for one of the most recent well-known PET services, Wasabi Wallet. We analyse the service's CoinJoin transactions using the published transactions of different time periods obtained from various external sources. We derive potential transaction patterns from the published transactions to create sets of criteria for the Wasabi CoinJoin transactions detection method. We published and presented this work at European Interdisciplinary Cybersecurity Conference (EICC) 2022 with the title “The Unique Dressing of Transactions: Wasabi CoinJoin Transaction Detection” [174]. We received the “Best Paper Award” from the conference with this paper.

⁴<https://cryptorc.org/project/crc2019>

- Chapter 6, we conclude the thesis with a summary of the contributions. We then discuss potential future works that improve upon the present limitations and investigate the area we have yet to cover in our current work. Finally, we provide a conclusion of our work.
- Appendix, we present the library that we develop to perform the experiments shown in Chapter 3, 4, and 5. We describe the data structure, software architecture, and script algorithm of the library's functions in detail. We also discuss the limitations in the current implementation of the tool and potential improvements that can solve these limitations or expand the tool's functions and features further.

Chapter 2

Literature Review

We first provide a historical overview of Bitcoin and cryptocurrency in Section 2.1, follows by the internal system of Bitcoin in section 2.2. Subsequently, we discuss the illegal activities related to Bitcoin and cryptocurrency in Section 2.3 and detail past literature on cryptocurrency tracking and deanonymisation methodologies in Section 2.4. Finally, we discuss privacy concerns involving Bitcoin and privacy techniques, including PETs and alternative cryptocurrencies in Section 2.5.

2.1 Bitcoin Historical Overview

Cryptocurrency is a medium of exchange that uses cryptography to conduct and secure its operation. It is a peer-to-peer electronic currency using database records called *blockchain* that can be accessed and maintained by anyone on the Internet. As most of the cryptocurrencies have no connection to either gold or traditional fiat currency and are not controlled by any central third-party entity, this means that their value is controlled solely by the value that people assign to it according to the supply and demand of the market.

The cryptocurrency concept was first discussed by Wei Dai in 1998 [46]. Additionally, Nick Szabo designed a decentralised digital currency called Bit Gold using cryptography to maintain a chain of currency data similar to Bitcoin blockchain in the same year [166]. In 2008, a person or group of individuals with the “Satoshi Nakamoto” pseudonym developed and created the first decentralised cryptocurrency called Bitcoin [133] by using Wei Dai’s cryptocurrency concept, the blockchain con-

cept by Haber and Stornetta [78], and the proof of work concept by Dwork and Naor [52]. Bitcoin was then launched in 2009 with its first block (genesis block) mined on 2009-01-03 with the message “The Times 03/Jan/2009 Chancellor on brink of second bailout for banks”.

Bitcoin is a cryptocurrency without physical representation in the real world. Unlike other electronic currencies, Bitcoin does not use an account balance system where Bitcoin amounts are added or subtracted from a balance with each transaction. Each Bitcoin’s “coin” exists only as data in the form of transaction output attached with a password script that its owner needs to unlock in order to spend the Bitcoins inside.

Bitcoin is the first electronic currency that resolves the fundamental challenge of *double-spending* without requiring a trusted third party to verify the transactions. Double-spending is one of the critical issues for the implementation of electronic currency, which involve attackers using the same “money” multiple times. Bitcoin overcomes the double-spending issue by implementing the blockchain database system to record all transactions and the proof-of-work protocol to strengthen against tampering of the transaction record data [69, 176].

Since its inception, the Bitcoin protocol has gone through various updates, such as Segregated Witness (SegWit) in 2017 that changed the block data size limit to accommodate the increasing growth in transaction activity and caused a hard “fork” that created a new cryptocurrency named Bitcoin Cash [196]. However, the core principle and mechanism of the Bitcoin protocol remain unchanged to this day.

2.2 Bitcoin Internal System

We describe in detail the architecture of the Bitcoin blockchain (Section 2.2.1 and 2.2.2), each type of Bitcoin address (Section 2.2.3), and how Bitcoin transactions are performed (Section 2.2.4). We then describe Bitcoin network protocol and the mining process along with the proof of work process (Section 2.2.5), which are crucial for the understanding of this thesis.

2.2.1 Blockchain

Blockchain or block chain is a database containing a list of continuously growing records (blocks) linked together with cryptographic hashing¹ that can be stored in multiple systems at once. The primary purpose of a blockchain database is to reduce the data integrity risk from data tampering and corruption. In Bitcoin, blockchain is used as a linear chain of transaction records distributed across a decentralised network of its users called *nodes*. Bitcoin blockchain data is unencrypted, which means that all of the Bitcoin transaction data can be accessed and interacted with by anyone on the Internet.

The Bitcoin blockchain contains a continuous chain of blocks starting from the first block called the genesis block. Every block that comes after the genesis block contains a hash created from its previous block, which effectively forms a blockchain. Any attempt to alter the data of any previous block will also have to change the hash of every successive block in existence, or the “counterfeited” blockchain will become invalid and rejected by the Bitcoin network. Furthermore, a chain is considered valid only if all blocks and transactions in the chain are valid [202].

There are many circumstances that can cause the Bitcoin blockchain to split into two or more forks (see Figure 2.1). One of such circumstances is when multiple *miners* (see Section 2.2.6) finish mining the same block and broadcast it to the Bitcoin network at roughly a similar time. In this case, some Bitcoin network nodes will receive one or another block first, and they will consider the first broadcast block they receive to be valid and keep comparing the length of blockchain data with other nodes. The shorter chain will be abandoned (orphan) by Bitcoin network nodes as Bitcoin protocol always considers the longest chain to be the only valid one [10]. Transactions inside an orphaned block are considered to be invalid and do not exist in the Bitcoin blockchain data. However, if there are enough nodes that decide to follow the orphan chain, it can result in a chain split that creates a new cryptocurrency [82].

The change of Bitcoin protocol can also result in a blockchain split. There are two types of changes in the Bitcoin protocol, which are soft and hard forks.

¹Cryptographic hash is an algorithm that maps or converts any data into a fixed-size hash value.

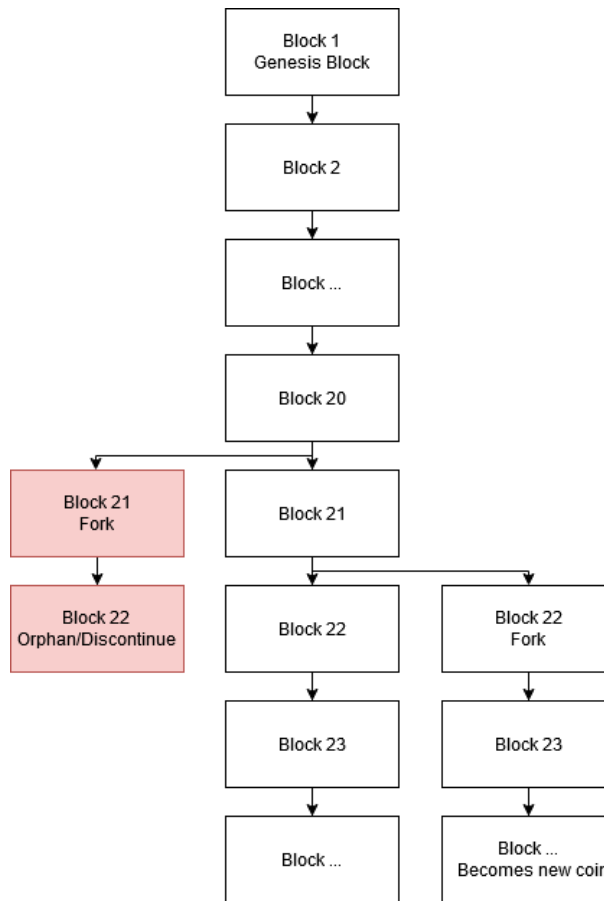


Figure 2.1: Blockchain and fork

Soft Forks Soft forks are changes of protocol ruleset that will cause a new and valid block in the old ruleset to be invalid in the new change. A soft fork does not necessarily result in a chain split as the new block ruleset does not conflict with the old ruleset [163]. For example, reducing the maximum block size from 1 megabyte to 500 kilobytes is considered a soft fork change as the new block is still within the old 1-megabyte limit rule.

Hard Forks Hard forks are changes in Bitcoin protocol that will cause a new block that is invalid in the old ruleset to be valid in the new change. A hard fork change will result in a chain split as nodes with the old ruleset will reject new blocks after the change. This change will also result in a permanent split as the only way to change back is to revert the blockchain history to before the change occurred. Hence, a hard fork usually only happens when almost every node agrees to the change [119]. For example, increasing the maximum block limit to 2 megabytes will cause a new block to violate the previous rule and makes it invalid.

2.2.2 Blocks

Blocks are simply data files that can contain any information. In Bitcoin, blocks are used to store transactions and other necessary data to form a chain of data that is easy to verify but difficult to tamper with [168]. During the early years of Bitcoin, each block in Bitcoin had a maximum size limit of 1 megabyte. The size limit was increased to approximately 4 megabytes in the SegWit upgrade [94].

Data	Description	Size
Magic numbers	A static unique value “0xD9B4BEF9” for indicating the type of data structure	4 bytes
Blocksize	Number of total block size in bytes	4 bytes
Blockheader	A set of specific block data	80 bytes
Transaction counter	Total number of transactions	1 - 9 bytes
Transactions	A list of transactions	Depending on the number of transactions

Table 2.1: Block data structure

A Bitcoin block contains several types of stored data, as can be seen in Table 2.1. Two of which make up the majority of the block data size: the block header and the transactions. Block header is a set of data related to the block itself and the main component to create a hash of the block, which serves as the block identification [139]. The data that composes a block header are as follows:

Block version A number that indicates the block validation rules it follows, currently at version 4.

Hash A hash string of the previous block header for linking the block into the continuous chain and validating the data integrity of the chain.

Timestamp A timestamp of the block creation time in Unix Epoch time format². The timestamp is used to help to prevent potential blockchain data tampering, where full nodes will reject a block with a timestamp that is two hours ahead of their system clock or behind the median time of the previous 11 blocks.

Targeted difficulty An encoded 256-bit target number that a block hash value needs to be lower than, where the lower the target value equates to higher

²Unix time is time system in a second format that has passed since 00:00:00 Coordinated Universal Time (UTC), 1970-01-01.

difficulty. The target can start from and be no less than difficulty of one³.

Nonce A random 32-bit number for the purpose of finding block hash that satisfies the target difficulty (see Section 2.2.6).

Merkle root A 256-bit hash of every transaction in the block [49]. A Merkle root is created from a Merkle or hash tree that combines multiple hashes of all transactions in the block into a hash tree structure, as shown in Figure 2.2.

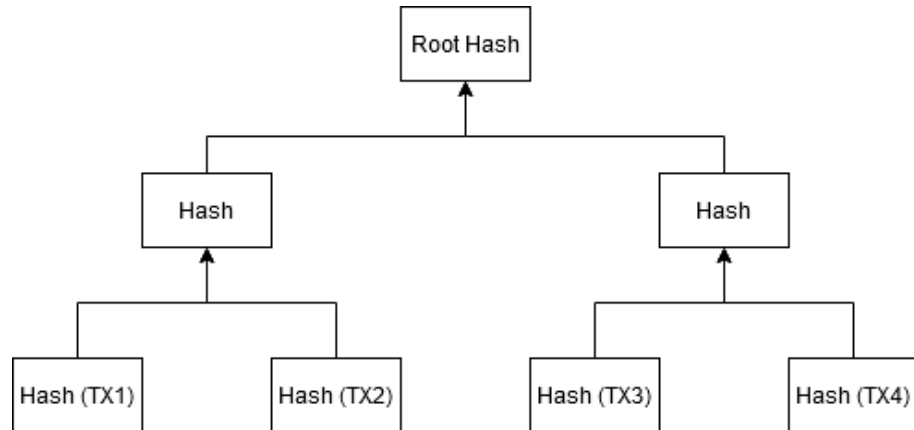


Figure 2.2: Merkle Tree of Bitcoin transactions

The root hash at the top is commonly referred to as Merkle root, which is a compact hashed of all transactions in the block.

2.2.3 Addresses

Bitcoin addresses are unique identifiers of Bitcoins ownership that are used for sending or receiving Bitcoins in transactions. In order to create an address, users have to generate a pair of private-public keys with the help of an automated address generator software, such as a *wallet* client. This process can be easily performed to create multiple addresses without any limitations. Moreover, Bitcoin addresses typically do not contain the personal information of the address owner(s)⁴.

Each Bitcoin address has a 256-bit number private key, which functions as a password for signing the “signature” to verify that the coins in transaction outputs belong to the address and can be spent by the owners. A private key is used to generate a unique identifier string called *public key* via ECDSA (Elliptic Curve

³At value `0x00000000FFFF000`.

⁴One exception is *vanity addresses* which are addresses that contain a set of specific characters that can serve as identification, such as `1SMITHaC2qOBkABeQ4csEPFiWJf2MJGgm`. Users can create vanity addresses in the same way as common addresses but typically require numerous creation attempts to find the right combination.

Digital Signature Algorithm) cryptographic hashing algorithm. The purpose of a public key is to be part of the transaction data for verification by the Bitcoin network to confirm that the transaction is valid. A public key is also used for generating a unique base58 string public key hash with SHA-256 and RIPEMD-160 cryptographic hash functions, which served as identification and recipient address information for receiving Bitcoins [24].

There are several types or formats of Bitcoin address which are Pay-to-Pubkey-Hash (P2PKH), Pay-to-Script-Hash (P2SH), Bech32, OP_RETURN, and Non-standard addresses.

Pay-to-Pubkey-Hash Addresses P2PKH addresses or commonly known as legacy addresses, are addresses that have a public key hash starting with the number “1”. P2PKH addresses are the first version of Bitcoin addresses and have their public key hash created from only their public key.

Pay-to-Script-Hash Addresses P2SH addresses have a public key hash that starts with the number “3”. Introduced in 2012, P2SH Addresses use a hash of any valid script instead of a public key to generate its public key hash. The script used to create the hash serves as the address’s unlocking condition that requires a matching script to access and spend the Bitcoins belonging to the addresses. One example of a P2SH address type is multi-signature (Multisig) addresses that allow the address to possess multiple private keys instead of only one. A multi-signature address typically requires a specific number of its private keys to unlock and spend Bitcoins in a transaction, such as one of two or two of three keys [22].

Bech32 Addresses Bech32 addresses or commonly known as bc1 addresses, have a public key hash that starts with “bc1”. Bech32 address is an address format introduced in the SegWit protocol upgrade. A unique feature of Bech32 Addresses is that the public key hash does not contain mixed case characters, unlike the previous address formats, to improve readability for the users. Additionally, the public key hash of Bech32 addresses can be created from either public key or other code script [23].

OP_RETURN Addresses OP_Return addresses are a unique type of addresses with a script that indicates the addresses as invalid. Bitcoin network nodes will still consider transactions that contain OP_Return addresses as valid and can be confirmed into the blockchain. However, any Bitcoin sent to an OP_Return address will become completely unspendable or irredeemable afterwards [14]. The primary purposes of OP_Return addresses are for the destruction of Bitcoins or storing specific signed data, such as images, proof-of-ownership signatures, messages, and passwords [116, 153, 207].

Non-standard Addresses Non-standard Addresses are addresses that are not approved or can not be recognised by the Bitcoin protocol as any of the standard address types mentioned above. Similar to OP_Return addresses, non-standard addresses may contain arbitrary data or script. Transactions with non-standard addresses are typically rejected by Bitcoin network nodes but can still be confirmed into the blockchain. Some examples of non-standard addresses are *UnLocked* addresses that contain an empty locking script, and *OnlyHash* addresses that contain a hash of specific data in the locking script [20, 153].

2.2.4 Transactions

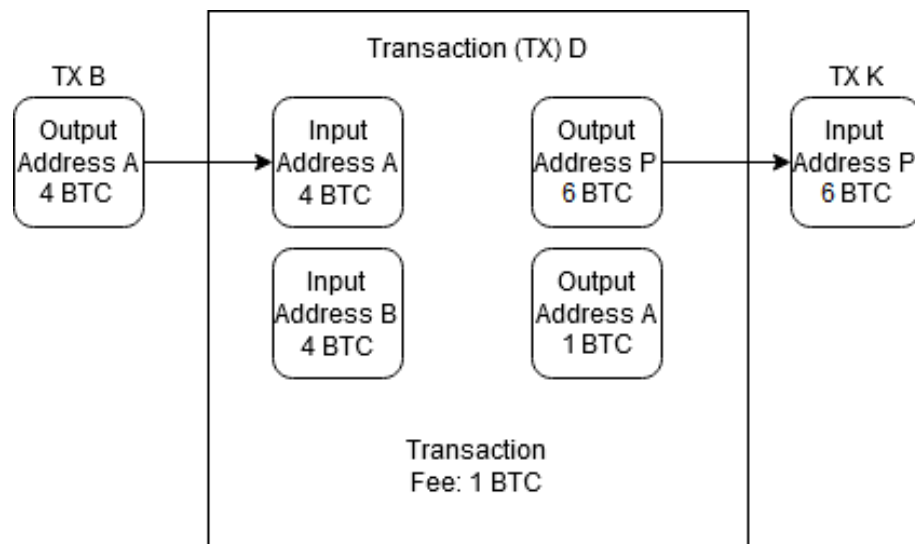


Figure 2.3: A simplified example of a Bitcoin transaction
TX is an acronym for transaction.

Data	Description	Size
Version number	The version of the transaction protocol, currently at 1	4 bytes
Input counter	Total number of transaction inputs	1 - 9 bytes
List of inputs	A list of transaction inputs	Depending on the number of transaction inputs
Output counter	A Total number of transaction outputs	1 - 9 bytes
List of outputs	A list of transaction outputs	Depending on the number of transaction outputs
Flag	A string 0001 if there is witness data, introduced in the SegWit upgrade	optional 2 byte array
Witnesses	A list of witnesses with one for each input, introduced in the SegWit upgrade	Optional variable
Locktime	A Block height or time when the transaction is first valid (can be mined)	4 bytes

Table 2.2: Bitcoin transaction data structure inside a block

Bitcoin transactions are exchanges of Bitcoins between one (with itself) or more addresses. Bitcoin transactions are initiated by the transaction sender(s) to send Bitcoins to the designated addresses [12]. As shown in Figure 2.3 and Table 2.2, Bitcoin transaction mainly consists of data related to transaction inputs (Table 2.3) and transaction outputs (Table 2.4).

Transaction senders confirm their transactions by broadcasting them to other Bitcoin network nodes for verification into a database called Mempool (Memory Pool), where all of the unconfirmed transactions wait for miners to confirm them into a new block in the blockchain. Transactions can be considered complete only when the block that contains them is successfully mined into the blockchain [147].

There are three standard types of transactions according to the types of addresses in transaction outputs that the Bitcoin system supports, which are Pay-to-Pubkey-Hash (P2PKH), Pay-to-Script-Hash (P2SH), Pay-to-Witness-Public-Key-Hash (P2WPKH), and Pay-to-Witness-Script-Hash (P2WSH) transactions. For more information on the code script of each transaction type, see Appendix A.

Pay-to-Pubkey-Hash Transactions P2PKH transactions are standard transactions that create transaction outputs to P2PKH addresses. The resulted transaction outputs specify the condition for future spending as inputs in the subsequent trans-

action with a private key signature and a public key of the receiving address.

Pay-to-Script-Hash Transactions P2SH transactions are transactions that create transaction outputs to P2SH addresses. Instead of requiring a public key to spend the transaction outputs, any valid script function can be used. Additionally, the signature required to validate the script can be any script that is sufficient to satisfy the condition and the transaction outputs from P2SH Transaction can only be used when the verification process of the serialized script with the input signature returns a Boolean value True.

For a P2SH Transaction with multi-signature addresses, the conditions specified in the resulting transaction outputs are slightly different and require a specific number of private key signatures instead of one. Transactions outputs of a multi-signature address are commonly referred to as m-of-n multi-signature transaction outputs, with m being the minimum number of required signatures to use the transaction outputs and n being the total number of signatures of the output address.

Pay-to-Witness-Public-Key-Hash and Pay-to-Witness-Script-Hash Transactions The SegWit upgrade introduces two new types of transactions, which are P2WPKH and P2WSH transactions. The two transaction types are similar to their previous transaction type counterparts (P2PKH and P2SH, respectively). The main difference is that transaction outputs' scripts include only a version byte (currently zero at the time of this thesis) and 20 bytes witness script containing a public key hash for P2WPKH outputs and 32 bytes witness script containing a script hash for P2WSH outputs.

2.2.4.1 Transaction Inputs

A transaction input is a reference to the transaction output from the previous transaction. Each Bitcoin transaction requires at least one transaction input that indicates the Bitcoins that are being spent. A transaction can have more than one transaction input from either single or multiple addresses. All of the Bitcoins in transaction inputs are added up into a total value and are distributed to the transaction outputs [146]. If the combined value of transaction inputs are more than that of transaction outputs, the remaining value will serve as a transaction fee for

Data	Description	Size
Previous Transaction hash	A transaction hash of the input's previous transaction	32 bytes
Previous transaction output index	Index of the input's previous transaction	4 bytes
Script length	Length of the input script	1 - 9 bytes
scriptPubKey or ScriptSig	Input script	Variable
Sequence number	A transaction level relative to the transaction's Locktime, generally at maximum (0xFFFFFFFF) and only relevant when Locktime is higher than 0	4 bytes

Table 2.3: Bitcoin transaction input data structure inside a block

the transaction's miners. It should be noted that there is no specific distribution method or record that indicate how individual input is distributed to the resulting outputs in a transaction.

Data	Description	Size
Value	The value of Bitcoins transferred to the output in Satoshis*	8 bytes
Script length	Length of the output script	1 - 9 bytes
scriptPubKey or ScriptSig	Output script	Variable

Table 2.4: Bitcoin transaction output data structure inside a block

*Sat or Satoshi is the smallest unit of the Bitcoin (1 Bitcoin is equal to 100,000,000 Satoshis).

2.2.4.2 Transaction Outputs

Each Bitcoin transaction requires at least one recipient transaction output to receive Bitcoins from the transaction input(s). Transaction output data consists of how many Bitcoins the output contains and the condition required to be spent in the subsequent transaction with a specific script depending on the transaction type [178]. Transaction outputs that are already spent in a transaction and confirmed into a block are called *spent output* and can no longer be used again in other transactions. Spent outputs serve as links to their previous transactions back to the point when they were created (mined) in a Coinbase transaction and effectively establish a chain of transactions.

Transaction outputs that have not yet been spent in a transaction are commonly referred to as *unspent transaction outputs* or UTXO. The number of Bitcoins that each address currently owns and can spend is calculated from the total Bitcoin value in the unspent transaction outputs. The Bitcoin protocol validates transaction outputs of recently broadcasted transactions with the unspent transaction output database maintained by Bitcoin network nodes, and the transaction will be considered valid only if all transaction inputs are unspent transaction outputs [184].

Bitcoin transactions may, in addition, contain transaction outputs called *change outputs*. As Bitcoins exist in the form of transaction outputs and not the balance of an address, the value inside transaction inputs can exceed the intended transferring value in a transaction. Transaction senders would require to create one or more transaction outputs to receive back the remaining change Bitcoins. The address of change output is called *change address* and belong to the transaction senders. Users can choose to either receive their Bitcoins back to the input addresses, other used addresses, or entirely new addresses [61]. Using the example from Figure 2.3, the change output in this transaction would be the second output to address A. It must be noted that there is no indication in transaction or output data that specifically distinguishes which transaction outputs in a transaction are change outputs.

2.2.5 Bitcoin Decentralised Network

As Bitcoin operates without a central third-party entity, all of the operations in Bitcoin depend on the consensus from the majority of the entities called *nodes* in the Bitcoin network. There are two types of bitcoin nodes, which are full nodes and lightweight nodes. We discuss more detail on the Bitcoin peer-to-peer network operation of each Bitcoin node type in Appendix A.

Full Nodes Anyone who participates in maintaining, verifying, and relaying the blockchain data is called a full node. In order to become full nodes in the Bitcoin network, users have to actively operate a Bitcoin wallet client that requires whole blockchain data to function and communicate with other full nodes in the network [48]. It is worth noting that most Bitcoin wallet clients do not necessarily require their users to generate a Bitcoin address or possess any Bitcoin to become

a full node. Additionally, a single user can operate multiple full nodes at the same time. Miners are also full nodes as they have to maintain, verify and relay blockchain data similar to other full nodes.

Lightweight Nodes Unlike full nodes, lightweight nodes are nodes that do not keep and maintain the whole blockchain data but can still send Bitcoin transactions. Lightweight Nodes only keep the block header data to validate the transactions with a process called Simplified Payment Verification (SPV) [114, 208]. Lightweight Nodes also do not participate in the verification process of other transactions or the blockchain. However, lightweight nodes are still required to connect to their parent full nodes to receive and send transaction data [108].

Lightweight wallet clients may instead use an external wallet system that belongs to the wallet service. In this scenario, the wallet service can have absolute control of their users' addresses and the users may not have access to the private key. The wallet addresses can also exist only as balance data in the service's external database. Therefore, these types of wallets can be restricted or entirely seized by the services [68, 201].

2.2.6 Mining

Bitcoin mining is a process of encrypting the newly created transaction block header and transaction data into a hash code to create a *proof of work* and secure the block's data. This process can be participated in by anyone on the Internet that operates a full Bitcoin node. The person involved in the mining process is called a miner, and a group of people that join together to contribute to the mining process is called a mining pool.

The mining process involves miners continuously obtaining the newest block information and validating the blockchain and transaction data. Miners would attempt to create a new block using the hash of the previous block and a list of unconfirmed transactions from the Mempool database. The new block that miners attempt to create must satisfy the current target difficulty by finding a hash or proof of work with a value lower or equal to the target using their system processing power. Once a block is successfully created, the block miners need to broadcast the

new blocks to the network for verification by other full nodes [107].

The Bitcoin system provides incentives to miners by giving rewards to the first participant that completes the block mining with transaction fees provided by transaction senders and newly created coins at a specific number. The Bitcoin system rewards miners each time a block is mined with newly created coins starting from 50 Bitcoins (BTC) and decreases in half every 210,000 blocks. The system will stop creating and rewarding new Bitcoins once there are 21 million Bitcoins in circulation. This way, Bitcoin can grow without depending on any centralised organisation but entirely by the users of Bitcoin themselves, while also limiting the supply of the coins to make Bitcoin a scarce commodity [91].

When Miners successfully mine a new block, they will receive newly mined coins and transaction fees in a transaction called *Coinbase* transaction. Coinbase transactions are the first transaction of a block generated by the miners and different from other transactions in that they have only one transaction input called generation transaction input, and its content is entirely ignored [81]. The Bitcoin output in a Coinbase transaction contains completely new Bitcoins that have no direct connection to the previous transactions, unlike other transaction outputs.

Bitcoin miners can explicitly choose which transaction they want to put in the block they create⁵. Miners typically calculate recommended transaction fees from all of the current unconfirmed transactions waiting in the Mempool database and prioritise transactions with the highest ratio of transaction fee value to transaction data size per byte due to the maximum block size limit. In this way, an incentive is created that allows transaction senders to make a decision between confirmation time and transaction cost. Transaction senders can pay a higher than the recommended transaction fee to shorten confirmation time, or they can save money by paying less transaction fee and waiting for longer confirmation time [53, 101].

Proof-of-work The mining process involves a proof of work process to create a countermeasure against mass mining and data tampering by making the mining process more difficult to accomplish according to the competing computational power. Bitcoin implements a difficulty system that constantly adjusts the difficulty of the mining process. For the block to become valid, miners have to find a hash combi-

⁵It is also possible for miners to create a block with only a Coinbase transaction.

nation that is lower than the difficulty target hash. The difficulty of Bitcoin mining changes every 2,016 blocks based on the computing power of all miners in the network to keep block mining time at a specific average rate, which is around one block every 10 minutes. Hence, the more miners compete in the mining process, the higher the difficulty is [64]. For more detail on the Bitcoin proof-of-work difficulty target calculation and block hashing requirement, see Appendix A.

The proof of work in the mining process serves as a security measure against transaction history tampering. As each block requires the previous block's hash, attackers must calculate a new hash of the block that they want to tamper along with the hash of the next blocks. As the blockchain will be continuously growing, this makes it highly challenging for anyone to alter or delete any block data once it is put into the blockchain, making the blockchain a semi-permanent record [70]. Furthermore, the difficulty and time-consuming aspects of the mining process make it computationally expensive for attackers to change or create a predictable hash for potential fake hash creation, as every hash is arbitrarily unique from each other [17].

2.3 Cryptocurrency and Illegal Activities

Bitcoin and other cryptocurrencies are frequently associated with illegal activities in cyberspace due to their privacy-centric nature that allows users to exchange money with pseudonymous identities and no restriction. There are several types of illegal activities that cryptocurrencies can facilitate in their operation either directly or indirectly, which are cryptocurrency thefts (Section 2.3.1), illegal goods and service tradings (Section 2.3.2), ransomware attacks (Section 2.3.3), and scams (Section 2.3.4).

2.3.1 Cryptocurrency Thefts

As cryptocurrencies hold monetary value, this, in turn, creates interest for individuals to steal cryptocurrencies for personal gain. The most common cause of cryptocurrency theft is the compromise in the victim's system that operates their wallet client(s), which allows attackers to gain access to their addresses' private key and steal the Bitcoins. As addresses in cryptocurrencies, including Bitcoin, typically

do not have a secondary protection system, such as secondary passwords, attackers can gain absolute control over addresses once they obtain the private keys [143].

Individual Thefts Bitcoin users themselves are often a victim of cryptocurrency thefts. The first major cryptocurrency theft occurred in 2011, in which a Bitcoin user lost 25,000 BTC (roughly 500,000 USD at the time) due to the security compromises in their mining system. The user reported the theft on the Bitcointalk forum⁶ [4]. The incident became one of the most studied sample cases in cryptocurrency cybercrime research [124, 128, 150].

Service Thefts Cryptocurrency services are also common victims of cryptocurrency thefts. Since the creation of Bitcoin, there have been many cryptocurrency service thefts nearly every year. From the first Mt.Gox exchange incident in 2011 [131] to the latest Hotbit exchange hack in 2021 [90]. The thefts that occurred at cryptocurrency services can affect both the service and its direct users. They also often cause a negative impact on the economy of the cryptocurrency market, which in turn can affect other cryptocurrency users and services to a degree [29, 41].

There are two common protections against cryptocurrency thefts for both Bitcoin users and services, which are cold storage wallets and multi-signature addresses. Cold storage wallets are data storing methods that involve users storing their addresses' private keys in location(s) that are unconnected to the Internet and can not be accessed from active systems, such as a paper wallet where users write down their addresses' private key on a paper or a hardware wallet like a USB drive [76, 109]. Meanwhile, multi-signature addresses (see Section 2.2.3) allow users to keep their addresses' private keys in separate places and systems [112].

2.3.2 Illegal Goods and Services

The anonymity in cryptocurrencies can facilitate users in tradings of illegal goods and services on the Internet, whether it be illegal drugs, prostitution, weapons, gambling, or even assassinations [56, 100]. Most of the illegal trading activities occurred on service entities called *Darknet Markets*.

⁶<https://bitcointalk.org>

Darknet Markets Darknet Markets are dark web⁷ commercial sites that provide a black market service for users to trade goods and services similar to online marketplaces, such as eBay and Amazon. Many online dark markets have adopted cryptocurrencies as their primary method of exchange [5, 63]. One prominent example is the Silk Road darknet market, which was one of the most popular darknet black markets that used Bitcoins for transactions on its website. Silk Road darknet made over 1.2 billion USD from its operation before its shut down by the FBI in 2013 [83, 156].

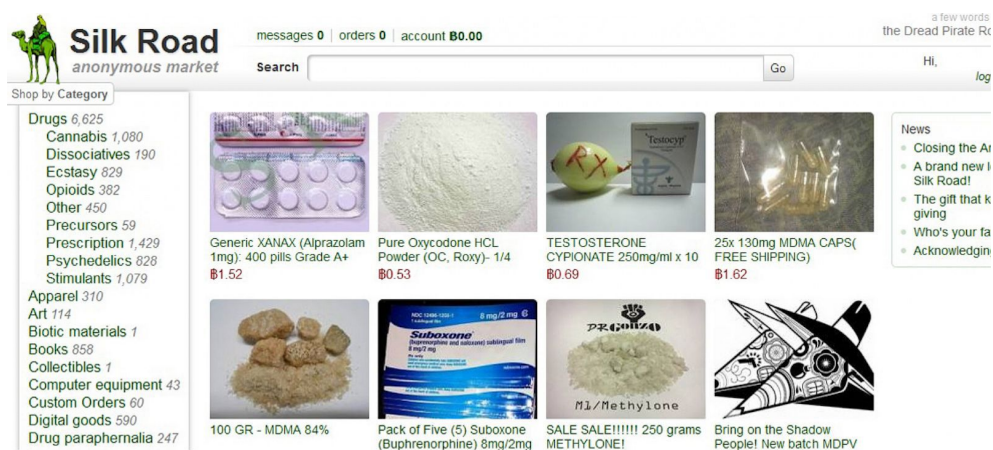


Figure 2.4: Silk Road market home page [129]

The closure of the Silk Road darknet market did not deter the rise of newer darknet markets. Since then, many darknet markets have taken place as the most popular trading platform, and many were also taken down by governments and legal forces. Some well-known examples are Agora, Alphabay, and Dream Market, all of which use Bitcoins or other cryptocurrencies for their payment system [3, 33].

2.3.3 Ransomware Attacks

Ransomware is a type of malware that steals and encrypts its victims' data for ransom. The victims can either choose to pay the ransom to unlock and retrieve the data back or risk losing the data permanently or having the data published by the attackers. Cryptocurrencies are typically utilised as the core component in ransomware attacks as they allow attackers to obtain the ransom money anonymously

⁷Dark web sites are websites that are not indexed by common search engines and can only be accessed by specific software or authorisation.

compared to other money transfer systems [102, 142, 145, 151].

Many public and private organisations were under the attacks of ransomware, which resulted in a significant loss. For example, CryptoLocker ransomware attacks managed to extort around 3 million USD in 2013 before its operation was taken down, and the ransom Bitcoins were seized by law enforcement agencies [110]. Another well-known example of ransomware is the Wannacry ransomware attacks that affected around 200,000 computer systems in 150 countries, including the National Health Service (NHS) in the United Kingdom. The attack can be linked to the cybercrime group named “Lazarus Group” which is involved in various other cyberattacks and allegedly sponsored by the North Korean government [21, 38].

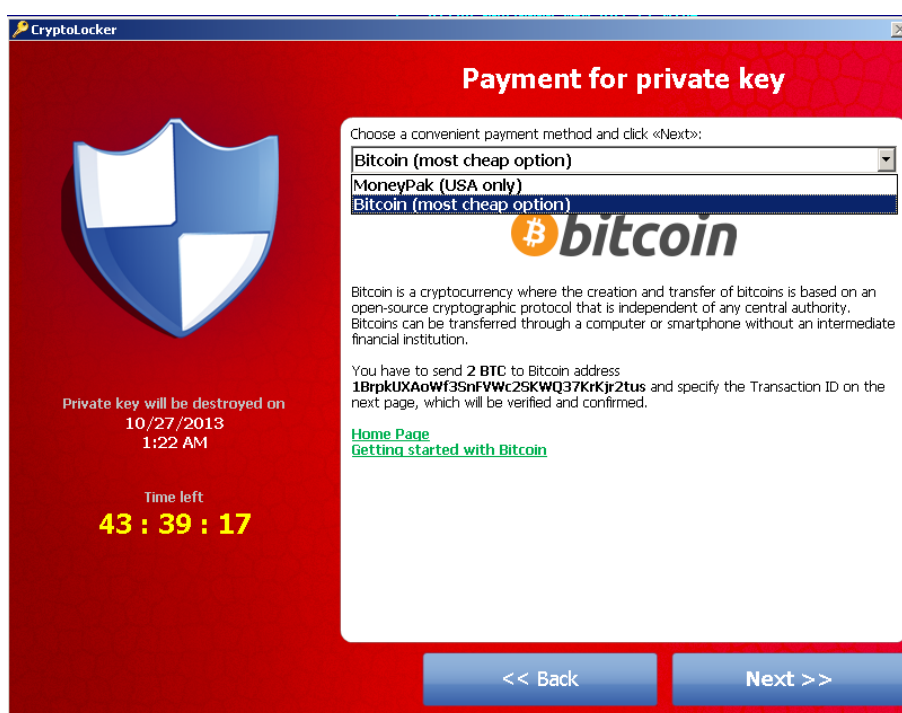


Figure 2.5: Bitcoin payment screen of CryptoLocker ransomware [74]

2.3.4 Scams

The cryptocurrency market itself is still one of the most unregulated financial technologies (Fintech) [45, 126, 162]. Almost anyone can create a cryptocurrency service or business without requiring official registration. The lack of clear regulation provides an opportunity for individuals to easily create a scam service or website to obtain cryptocurrency coins from its victim users [125, 165]. Some of the common

types of cryptocurrency scams are service scams, phishing websites, and High Yield Investment Programs (HYIPs).

Service Scams As cryptocurrency is still a new and growing market, new services are regularly created to fill the market demands. Many such services can be fake/s-cam services that typically have one purpose, to receive money or cryptocurrency coins from its victim without providing the promised goods or services in return. Cryptocurrency service scams exist in many forms, such as wallet scams, cryptocurrency exchange scams, mining pool scams, and even PET service scams [182].

Even the most commonly used services are not without the risk of being scam services. For example, Mt.Gox exchange, which was the largest exchange service during its time, is widely believed to fake their own hacking and theft as part of the exit scam⁸. Mt.Gox exchange was also allegedly involved in illegal acts by employing trading bots⁹ named “Willy” and “Markus” to artificially increase trade activity within the Mt.Gox market system by purchasing a large number of Bitcoin throughout their operations. As a result, the price of Bitcoin was raised from 150 USD to more than 1,000 USD within two months span before the price collapsed [66].

Phishing Websites Phishing or spoofed websites are also prevalent in the cryptocurrency market. Similar to other types of Phishing websites, cryptocurrency phishing websites often mimic a well-known service in an attempt to trick users into paying cryptocurrency coins to the scammers’ addresses or providing cryptocurrency addresses’ private keys or other information [200].

High Yield Investment Programs (HYIPs) HYIPs or online Ponzi schemes are scams that involve scammers advertising or promising investment opportunities with unrealistically high return profit. Since Ponzi schemes are typically unsustainable (even if created with a legitimate purpose), they eventually collapse unless maintained by perpetual exponential growth [99, 135]. Ponzi schemes are typically operated by criminals who frequently disappear with the investment funds before

⁸An exit scam is a type of scam where a business or service creates a good reputation as a legitimate entity first before stealing money from their users before its disappearance or closure.

⁹Bitcoin trading bot is a type of program that automatically buys or sells Bitcoins at certain intervals.

the victims catch on to the scheme or the inevitable collapse. There are many cryptocurrency services that offer HYIPs where users receive interest percentage of invested cryptocurrency coins after a specific number of days [183]. There are also alternative cryptocurrencies that are created with the purpose of being Ponzi schemes themselves, such as Bitconnect¹⁰ or OneCoin¹¹.

2.3.5 Money Laundering

Cryptocurrencies can also be used as laundering tools for real-world currencies either for tax evasion or assisting criminal activities [31]. The anonymity that Bitcoin and other cryptocurrencies provide for their users makes them an effective medium for removing any connection between the source money that criminals exchange to cryptocurrency coins and the laundered money they exchange back [32, 149].

2.4 Bitcoin Tracking and Deanonymisation

The illegal activities in Bitcoin and other cryptocurrencies create interests in the development of tracking and deanonymisation techniques from cryptocurrency market participants and organisations – such as government, regulatory agencies, private blockchain analytics companies – as well as researchers to decipher and track the transaction network of illegal cryptocurrency coins, whether it be for research, crime forensic, law enforcement, or personal interest purposes.

We review in this section the Bitcoin tracking and deanonymisation literature that contributes to cryptocurrency forensic analysis methodologies.

2.4.1 Taint Analysis

As mentioned in Section 2.1, Bitcoins exist in the form of unspent transaction outputs that are newly created from the sum of inputs in transactions. As a result, when there are multiple transaction inputs and transaction outputs in the same transaction, the blockchain data does not contain any information that indicates the exact flow of Bitcoins exchange between addresses in transaction inputs and

¹⁰<https://bitconnect.com>

¹¹<https://web.archive.org/web/20150207021001/http://onecoin.eu/>

those in transaction outputs. Hence, it can be challenging to identify or differentiate the distribution and destination of the interested Bitcoins without utilising a precise methodology for Bitcoin tracking.

Taint analysis or taint checking is a well-known data analysis concept that assigns a “tainted” value to a specific data source of interest and analyses the flow of information by following possible paths that the tainted data can propagate. The concept is commonly used for security exploit detection (e.g., [123], programming code analysis [9], and information flow analysis [28]).

Taint analysis is utilised as a tracking method in cryptocurrency that tracks targeted cryptocurrency coins using transaction information in the blockchain. The primary purpose of cryptocurrency taint analysis is to determine the association between the addresses in a transaction [127], which can be used to classify the targeted cryptocurrency coins (e.g., stolen Bitcoins resulting from a known theft transaction) as tainted (or “dirty”) and any address that uses or transfers them will be considered a tainted address. Meanwhile, coins that are unrelated to tainted coins are considered clean coins. Each taint analysis strategy applies a specific rule-set to estimate how the targeted cryptocurrency coins are distributed in the following transactions.

The taint analysis method can be employed for tracking cryptocurrencies that utilise transparent blockchain systems similar to Bitcoin. For example, several studies have implemented the taint analysis concept to track the movement of Ethereum coins and smart contracts tokens [39, 67]. However, privacy-oriented cryptocurrencies that obscure transaction and address information, like Zcash, Monero¹², and Zcoin (Firo)¹³, are typically immune to taint analysis tracking because they are explicitly designed to strengthen privacy against blockchain-based tracking.

We identify three taint analysis strategies proposed in the literature, which have been implemented for Bitcoin tracking in various studies [1, 40, 47, 164, 181].

Poison and Haircut The Poison strategy is a taint analysis strategy that classifies every transaction output within the transactions as a fully tainted output, regardless of the number of tainted Bitcoins involved [128]. The number of tainted

¹²<https://www.getmonero.org>

¹³<https://firo.org>

Bitcoins will exponentially increase when tainted and clean Bitcoins are used together in the same transaction, which is a drawback that makes the strategy unable to provide precise tracking results. Using the example in Figure 2.6, in a transaction with a 7 BTC clean input and a 3 BTC tainted input, both of the resulting 9 BTC and 1 BTC outputs will be classified as entirely tainted. This tainting strategy results in the total number of tainted Bitcoins at 10 BTC from initially 3 BTC. It is worth noting that we do not account for the transaction fee in this transaction example.

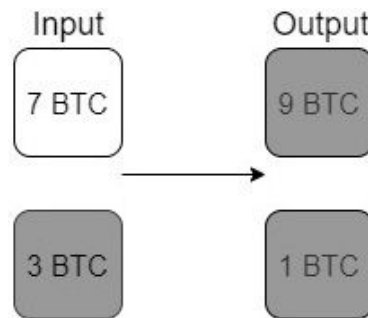


Figure 2.6: Poison strategy

White rectangles represent clean inputs or outputs, and dark grey rectangles represent fully tainted ones.

The Haircut strategy shares the same tainting methodology as the Poison strategy, though implements an additional rule: instead of being classified as tainted entirely, each output in the transaction will receive a proportion of the tainted inputs according to their proportions [128]. Using the example in Figure 2.7, instead of being tainted entirely, both outputs will receive the same proportion of the tainted Bitcoins to the total input value ($3/10$ proportion), which in this case would be 2.7 tainted BTC for the 9 BTC output and 0.3 tainted BTC for the 1 BTC output.

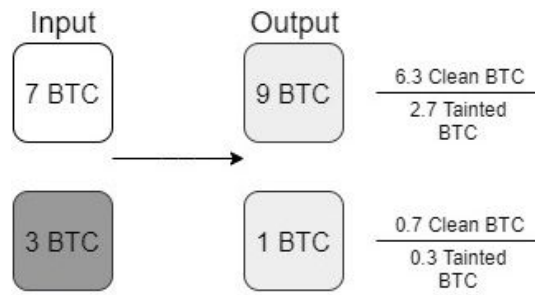


Figure 2.7: Haircut strategy

White rectangles represent clean inputs or outputs, dark grey rectangles represent fully tainted ones, and light grey rectangles represent partly tainted ones.

While the Poison and Haircut strategies are the most common tracking strategies used in the previous Bitcoin tracking research, we argue that both strategies typically produce a very large number of tainted transactions due to their tainting methodology, especially when tainted Bitcoins get combined with other clean Bitcoins, which makes them impractical for both tracking and analysis purposes.

FIFO (First-In, First-Out) The FIFO strategy is a concept of asset inventory management for sorting the order of items via distribution. The concept of FIFO is essentially that the first item that goes in is also the first one that goes out [6]. Using the example shown in Figure 2.8, the FIFO strategy starts by distributing the first 350 BTC input to the first and second outputs. Next, the FIFO strategy will distribute the 100 and 500 BTC inputs to the third output. Finally, the remaining 500 BTC and the last 60 BTC inputs are distributed to the last output. As a result, the 200 and 150 BTC outputs will contain the full proportion of the tainted Bitcoins. The third output will contain no tainted Bitcoins, and the last output will be partly tainted with 60 BTC.

The FIFO strategy is implemented as a taint analysis strategy based on the argument that it has already been established for official law enforcement for tracking stolen traditional currency and can provide more precise results compared to the Poison and Haircut strategies, as the FIFO strategy does not consider every resulting output as tainted. This would allow governments or relevant organisations to implement more practical law enforcement and blacklisting systems that can constrain illegal Bitcoins from a smaller number of transaction outputs and addresses [6].

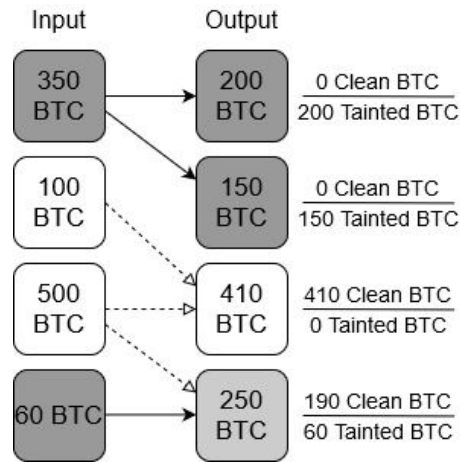


Figure 2.8: FIFO strategy

White rectangles represent clean inputs or outputs, dark grey rectangles represent fully tainted ones, and light grey rectangles represent partly tainted ones.

However, as the FIFO strategy distributes Bitcoins based on a uniform predetermined way, it is possible for the FIFO strategy tracking to produce inaccurate tracking results. The FIFO strategy can distribute tainted Bitcoins to the transaction output(s) that are not their intended destination (e.g., distribute tainted Bitcoins to other unrelated users in a PET transaction). Using the example in Figure 2.8, if the 410 BTC output is the intended destination for the tainted Bitcoins, the FIFO strategy will produce inaccurate tracking results afterwards. Therefore, it is impractical to implement the FIFO strategy independently for tracking purposes and should be instead implemented in combination with other taint analysis strategies.

LIFO (Last-In, First-Out) A strategy that is a natural alternative to the FIFO strategy is the LIFO strategy which operates in the opposite ordering of the FIFO strategy. The LIFO strategy assumes that the last item that goes in is always the first to go out. Using the example shown in Figure 2.9, the LIFO strategy starts by distributing the last 60 BTC input to the first output. Next, the LIFO strategy distributes the 500 BTC input to fill up the first, second and third outputs followed by the second 100 BTC input. Lastly, the first 350 BTC input is distributed to fill up the remaining 410 BTC and the last 250 BTC outputs. As a result, the first 200 BTC output will contain 60 BTC tainted Bitcoins, the third output will contain 100 BTC tainted Bitcoins, and the last 250 BTC output will be entirely tainted.

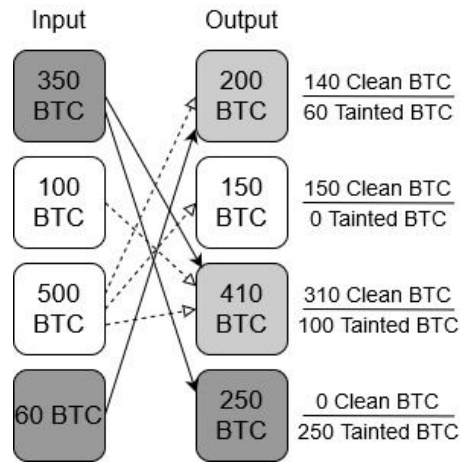


Figure 2.9: LIFO strategy

White rectangles represent clean inputs or outputs, dark grey rectangles represent fully tainted ones, and light grey rectangles represent partly tainted ones.

It should be mentioned that the LIFO strategy also shares the same weakness as mentioned for the FIFO strategy since both strategies distribute tainted Bitcoins in a uniform predetermined way based on transaction orders.

2.4.2 Address Clustering and Deanonimisation

Although taint analysis provides a foundation to Bitcoin tracking, simply employing taint analysis strategies to track individual user's Bitcoins from one address to another is inefficient due to the fact that taint analysis on its own does not take the transaction purpose and address ownership into account, which often produces unessential tracking results regardless of the strategy employed. For example, tracking Bitcoins after they reach addresses that belong to a cryptocurrency service indicates that the tainted Bitcoins are already exchanged with the service and are no longer in the possession of the targeted users. Both address clustering and deanonymisation can assist the cryptocurrency forensic analysis by providing the information of address ownership to the taint analysis and assisting the analysis process of tainted Bitcoins' spending and movement.

2.4.2.1 Address Clustering

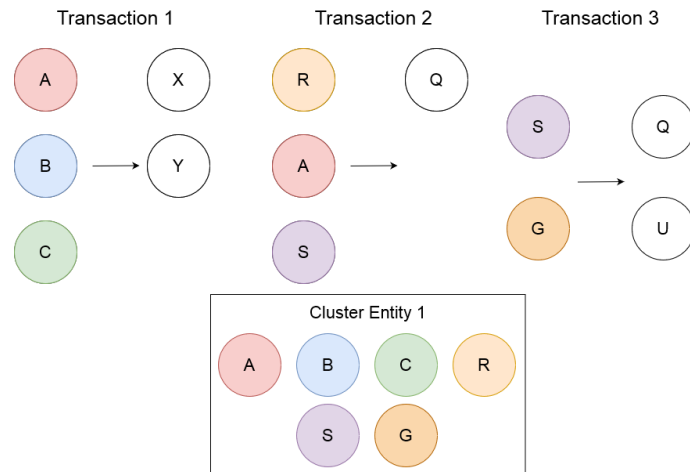


Figure 2.10: Multi-input address clustering heuristic

Circles represent addresses, and various colours are indications that the addresses are clustered by the multi-input address clustering heuristic.

Bitcoin address clustering is the process of linking and classifying Bitcoin addresses that likely belong to the same user/entity into a group or cluster based on the information of the transaction and predefined clustering heuristic. The address clustering process in Bitcoin utilises a similar clustering procedure as the machine learning counterpart, where the process groups unlabeled input data based on specific connections. As Bitcoin address clustering heuristics are typically created based on a specific assumption of transaction pattern due to the lack of verifiable ground-truth information, the proposed heuristics are not guaranteed to be capable of providing completely accurate results.

The Bitcoin address clustering process starts with every address classified in a cluster of one address. The Bitcoin address cluster procedure processes all transactions in the Bitcoin blockchain in chronological order from the first block to the latest, and if multiple clusters intersect with each other, then all of the clusters involved are merged into one cluster. After the clustering process, each address cluster can be labelled with an identification of the potential owners. To this day, there are two most commonly used Bitcoin address clustering heuristics as follows:

Multi-input Heuristic The most commonly used Bitcoin address clustering heuristic is the multi-input heuristic or input sharing clustering, which classifies all the input addresses in the same transactions as belonging to the same entity [36, 103, 209]. The procedure of the multi-input heuristic is as follows; for any transaction (tx) in transaction data (TX), if the number of transaction inputs is more than one, then all addresses in the transaction inputs ($input_adr$) belong to the same user cluster (c_i) where i is the unique identification of the cluster, as depicted in the below algorithm.

```

for  $tx \in TX$  do
  | if COUNT( $tx.input$ ) > 1 then
  | | APPEND  $c_i \leftarrow \{tx.input\_adr \in tx.input\}$ 
  | end
end

```

The multi-input heuristic originated from the fact that in order to perform a multi-input transaction, the owner(s) of all the input addresses have to agree to the transaction by signing the addresses' private keys, which often implies that the input addresses in the same transaction would belong to the same person as they need to possess all of the addresses' private keys. However, while the multi-input address clustering heuristic is one of the most utilised, in addition to reducing the tracking effectiveness, the CoinJoin method we discuss in Section 2.5.2 can significantly reduce the effectiveness of this address clustering heuristic.

Change Address Heuristic Another well-known address clustering method is the change address clustering heuristic which identifies which transaction output is the change output or address that belongs to the transaction input user (transaction sender). The change address clustering heuristic operates based on the four specific criteria as follows: [121]

1. The change address has never been used before in any other transaction (i.e., fresh address).
2. The transaction is not a Coinbase transaction.
3. The transaction has no common address in transaction inputs and outputs (i.e., self-change address).

4. The other output addresses other than the fresh change address have been reused in other transactions before.

The procedure of change address clustering heuristic is depicted in the below algorithm.

```

for  $tx \in TX$  do
  if  $\text{COUNT}(tx.output) > 1 \wedge tx \neq \text{Coinbase} \wedge tx.output\_adr \notin$ 
     $tx.input\_adr \wedge \text{COUNT}(tx.fresh\_ADR) = 1$  then
    for  $output\_adr \in tx.output$  do
      if  $\text{COUNT}(output\_adr.tx) = 1$  then
        | APPEND  $c_i \leftarrow output\_adr$ 
      end
    end
  end
end

```

The change address clustering heuristic can identify which output addresses are change addresses belonging to the transaction initiators, which allows for more extensive transaction tracking and deanonymisation [54, 209]. However, the change address clustering heuristic can be evaded by either sending coins to more than one change output or avoiding using change output. Additionally, the heuristic can not identify change addresses in transactions with multiple change addresses due to the fourth criterion.

Address clustering can also combine with transaction behaviour analysis. For example, when one address distributes its Bitcoins to various other addresses and those receiving addresses send all of the distributed Bitcoins to the same single address later, it can be assumed that all of the receiving addresses belong to the same entity [79].

2.4.2.2 Address Deanonymisation

While the easy to generate pseudonymous address function in Bitcoin and alternative cryptocurrencies provide anonymity to its user to some degree, Harrigan and Fretter' study [79] discovered that a significant number of Bitcoin users reuse their addresses multiple times, and many of the reused addresses can be formed into a super-cluster with high centrality. Therefore, it is still possible to identify the entity of the address owners via the analysis of its public blockchain record and address

clustering heuristics.

Bitcoin addresses can be deanonymised and profiled using the information from transaction data in the blockchain. For example, Dorit and Adi [154] performed statistical analysis of Bitcoin transaction history until May 2012 using the *Union-Find algorithm*¹⁴ to find the owner or a group of owners that has control over the associated Bitcoin addresses from transaction input sharing similar to the multi-input address clustering heuristic. The researchers created 2,460,814 entities out of 3,120,948 addresses and identified three out of nineteen entities with the highest number of incoming coins, which are Mt.Gox exchange, Instawallet¹⁵, and Deepbit¹⁶.

There are also other studies that integrate external off-blockchain information, such as public personal information, geo-location, and business data, to improve the address profiling process. For example, Fleder et al. [62] examined the level of anonymity of Bitcoin users by combining external information and blockchain analysis to deanonymise and profile Bitcoin addresses using publicly available information from websites, forums, and social networks with web scraping and transaction fingerprinting methods. The researchers used a web scraping tool to scrape address data from the Bitcointalk forum and “eavesdrop” on transaction information, such as time and number of Bitcoins exchanged in the transaction to match with information in the blockchain for identification from the same forum’s posts. The research’s network graph analysis result reveals that several identified forum users conducted direct transactions to addresses belonging to SatoshiDICE¹⁷, Silk Road market, and Wikileaks¹⁸. The methods employed by Fleder et al. [62] can indirectly overcome the anonymity function in Bitcoin and are beneficial for the deanonymisation process. On the other hand, these methods can not be reliably used exclusively in the long term as users can fake the information or employ multiple online identities. Hence, these methods are best suited for combining with other address clustering heuristics.

Two other pieces of research by Jordan et al. [95, 96] implement the multi-input address clustering heuristic to create an address cluster data set and use the external

¹⁴Union-Find Algorithm is a network analysis method used to track related subsets within a network into groups.

¹⁵<https://instawallet.org> is a wallet service closed in 2013 due to the theft that occurred.

¹⁶<https://deepbit.net> is a defunct mining pool.

¹⁷<https://satoshidice.com> is a gambling website that uses Bitcoin as a betting currency.

¹⁸<https://wikileaks.org> is a website that publishes secret, classified and leaked information and documents.

address profile data scraped from Wallet Explorer¹⁹ to identify cryptocurrency service entity addresses. The research identified 30,331,700 addresses belonging to 272 unique entities and discovered common transaction patterns of entities belonging to the same service type that can be applied for the classification of other unidentified addresses.

2.4.3 Existing Tracking Implementations

There have been some research works that developed a cryptocurrency tracking library and made it publicly available that we know of as follows:

- *Bitiodine* [164] is an open-source²⁰ blockchain parser and visualiser software that allows users to parse cryptocurrency blockchain data into SQLite databases. *Bitiodine* also includes an address clustering heuristic function to visualise a graph of address clusters. *Bitiodine* itself does not have a taint analysis tracking function, but the SQLite blockchain databases can be exported to perform external taint analysis. The *Bitiodine*'s developers ended the tool's development and support after the last update on 2020-07-13.
- *Bitconeview* [50] is a blockchain visualiser software that allows users to perform cryptocurrency tracking similar to taint analysis. The developers also provide an online service for transaction directed graph network visualising with the tool²¹.
- *BlockSci* [97] is an open-source blockchain analysis software that allows users to directly parse blockchain data from a full node client of Bitcoin or other similar cryptocurrencies. *BlockSci* analysis library includes transaction analysis and address clustering functions and can be used with other Python and C++ language programming modules. *BlockSci* also offers a simple taint analysis function that produces a chain of tainted transactions from a specific starting transaction. *BlockSci* has been used in several cryptocurrency studies [18, 27, 72] for blockchain parsing and analysis purposes. The *BlockSci*'s developers ended the development and support of the tool after its latest version 0.7 release on 2020-07-30.

¹⁹<https://www.walletexplorer.com>

²⁰<https://github.com/tzarskyz/bitiodine>

²¹<http://www.bitconeview.info>

- *Taintchain* [7] is an open-source taint analysis and visualiser Rust module²². The tool supports only the FIFO taint analysis strategy and has no other additional functions for transaction analysis. The FIFO taint analysis function in this tool can taint the miner’s transaction fee. The tool’s first and only version (at the time of this thesis’s writing) was released on 2020-06-19.

2.5 Bitcoin and Privacy Techniques

As the Bitcoin blockchain is completely transparent, the privacy level in Bitcoin depends on the discretion of the users themselves. If users are not cautious in their Bitcoin activities, their address and transactions can become vulnerable to tracking and deanonymisation attempts. Bitcoin users can improve their privacy by employing privacy techniques (Section 2.5.1), utilising external PETs (Section 2.5.2), or exchange their Bitcoins with alternative cryptocurrencies that have better privacy features (Section 2.5.3).

2.5.1 Privacy Techniques

There are several privacy techniques that Bitcoin users can utilise to improve their privacy.

Wallet Selection As mentioned in Section 2.2.5, lightweight node wallets require a connection to a parent full node to be operational and send transactions. Parent full nodes can obtain the address and transaction data of the child lightweight nodes, which can potentially be linked to the user information on the application side, such as the IP address of the lightweight node system [19, 98]. Therefore, full node wallets provide better privacy but come at the cost of the requirement to maintain the growing blockchain data. Some wallet services may require users to provide personal identification information to access their service. For example, Binance, which is one of the most well-known exchange services, requires their users to supply them with a government-provided ID document in order to use their exchange service

²²<https://github.com/TaintChain/RustyTaintChain>

that includes wallet account service ²³.

Tor Connection While full node wallets can provide better privacy than lightweight node wallets, they are still not without faults. As described in Section 2.2.5, full nodes require an IP address to establish a connection to the Bitcoin network. Although such information is not recorded in the blockchain data, it is possible for neighbour nodes to keep track and publish such information [59, 210]. It is then a simple task to link the address and transaction information to the IP address of the node that broadcasts the targeted transactions [15, 87]. As most Bitcoin wallet clients are compatible with the Tor network, users can connect to the Bitcoin network and broadcast transactions through Tor to solve this privacy issue [130].

Wallet Fingerprinting Prevention Each Bitcoin wallet client typically has a default address and transaction setting that can be used to determine which client software an address used to perform transactions. Such information can be examined to help with the analysis of the ownership of addresses in the targeted transactions. There are several types of information that can be used to identify whether transactions are created with the same wallet client settings [2].

- The type or format of Bitcoin address (see Section 2.2.3) can be the source of privacy compromise as most wallet clients use only one specific address type by default. If all addresses in transaction inputs are of the same address types, it can indicate that the input addresses are from similar wallet clients. Additionally, if a transaction has output addresses of different types, the output addresses that have the same address type as the transaction inputs are likely to be the change outputs.
- Transaction inputs and outputs selection algorithm is different for each wallet client and typically not randomised by default. The transaction pattern can then be analysed to determine whether the prior or subsequent transactions are performed by the same wallet client or not.
- Each wallet client generally use only one script for P2SH and P2WSH addresses. The attackers can determine address ownership by the address script

²³<https://www.binance.com/en/support/announcement/51bf294e26324211a4731ca998e110ca>

in the transaction inputs and outputs. For example, a transaction input from a 2-of-3 multi-signature address to a transaction output with a 3-of-4 multi-signature address indicates that the two addresses are less likely to be from the same wallet client.

- Each wallet client typically has a specific algorithm to calculate and assign the transaction fee, which can potentially be analysed for further tracking analysis.

These wallet related privacy issues can be mitigated by utilising multiple wallet clients and randomising the transaction setting to reduce the tractability.

Avoiding Address Reuse One of the most common privacy practices in Bitcoin is to avoid reusing the same address multiple times. Address reusing allows other people to effortlessly observe the history of all previous transactions and analyse the Bitcoin activity of the address's owner(s) [71, 134]. As Bitcoin allows users to create new addresses with zero cost, users should create and use new addresses for every transaction. This technique increases the difficulty of the transaction network analysis since different addresses are more difficult to analyse compared to a single address, especially if they have no direct connection to each other. However, it is possible for attackers to force address reuse on the targeted users by simply sending some Bitcoin to the targeted address, which would create an incentive for the users to use the Bitcoins and potentially reveal more address and transaction connections.

Transaction Privacy Practices Most of the Bitcoin wallet clients allow their users to control or adjust the transaction structure in some ways. This function can be used to improve the users' privacy by setting up the transactions in specific ways that make them more challenging to analyse.

Due to the address deanonymisation with the multi-input address clustering heuristic, sending Bitcoins from multiple addresses in the same transaction can severely compromise the user's privacy. Users can avoid transaction input sharing between their addresses and send Bitcoins from each address separately to preserve their privacy. Change outputs can also be the source of privacy vulnerability as their purpose indicate that the change addresses belong to the same entity as the transaction senders, as mentioned in Section 2.4.2.1. It is possible to avoid this issue by setting up transactions to have the same Bitcoin input value as the required

spending value (plus transaction fee). Users can also set up their transactions to have more than one change output, although this will increase the transaction fee cost due to the increase in transaction size from more transaction outputs.

Off-exchange Tradings As Bitcoin offer a transaction system that allows users to exchange Bitcoins without requiring a centralised third-party entity to validate the transactions, Bitcoin users do not necessarily have to use a cryptocurrency service in order to purchase products or services or exchange Bitcoins to other currencies. For example, buyers can agree on Bitcoin exchanges with sellers via external communication channels, such as forum websites, private text messages, or physical meetings and send Bitcoins to sellers' addresses directly. This type of Bitcoin exchange is significantly difficult to identify with the blockchain data and taint analysis tracking because the transactions involved are generally not different from any other transactions where Bitcoins move from pseudonymous addresses to others.

However, there is a severe risk for off-exchange trading transactions due to the lack of a refund system for Bitcoin senders, except for circumstances such as when a blockchain fork occurs and causes the transactions to become invalid in the shorter chain. Bitcoin receivers can refuse to adhere to the agreement of exchange after they receive Bitcoins, which means that the senders would lose their Bitcoins without a reasonable way to obtain them back [137]. Hence, off-exchange trading transactions can be easily performed but possess a considerable risk, unlike Bitcoins exchanges through a cryptocurrency service.

There are many public and private decentralised cryptocurrency off-exchange trading services that are created to lessen the risk of off-exchange trading transactions. Unlike other centralised cryptocurrency services, decentralised off-exchange trading services typically do not have a central service address that users have to first send Bitcoins to before they can access the service. For example, Bitcoin-otc²⁴ provides a public over-the-counter marketplace service where users register with a pseudonymous user name and optional Bitcoin addresses on the service. Users can then post buy or sell order advertisements on the service's order book board. The service also provides a reputation rating system where users can be rated a score by other users with a positive or negative rating that indicates the degree of trust.

²⁴<https://bitcoin-otc.com>

2.5.2 Cryptocurrency Privacy-Enhancing Technologies (PETs)

While the privacy techniques described in Section 2.5.1 increase tracking difficulty and Bitcoin users' privacy, those techniques still do not provide a complete tracking immunity. This issue creates demands for further privacy improvement with assistance from external systems or services in the form of PETs. There are three prominent PETs explicitly designed to facilitate Bitcoin tracking evasion, which Bitcoin users can use to obscure their transactions.

2.5.2.1 Bitcoin Mixing

Bitcoin mixing is a PET performed by a cryptocurrency mixer service (also often referred to as laundering or tumbling service), which facilitates its users with the mixing process [13]. The most successful mixing process would produce what is commonly called “zero-taint” Bitcoins by completely removing any transaction connection between the original and the resulting mixed Bitcoins, thereby rendering taint analysis tracking completely ineffective [60]. There are two major types of mixer services, which are centralised and decentralised mixer services.

Centralised Mixer Services Centralised Mixer services are mixer services that rely on central mixing addresses to perform the mixing operation. Centralised mixer services typically operate by having users deposit their Bitcoins to one of its deposited addresses and mix the deposited Bitcoins with Bitcoins from other users [84, 100]. Subsequently, the service would send unrelated Bitcoins to the user's destined address(es) in one or multiple transactions [60, 199]. The majority of well-known mixer services, such as Bitcoin Fog, Helix, Bestmixer, and Bitlaunder are centralised.

Decentralised Mixer Services Decentralised mixer services are mixer services that do not have central mixing addresses to perform the obscuring operation. They typically serve as an intermediary by providing an automated mixing protocol for its users to connect with each other and perform the mixing without having to transfer the Bitcoins through service addresses. A decentralised mixer service typically operates by using the CoinJoin mixing concept as its mixing operation [113]. The

users would generally pay the mixing fee to the service's address(es) in the mixing transactions instead of sending Bitcoins to a deposited address. Wasabi Wallet's CoinJoin mixing, JoinMarket, and Samurai Wallet's Whirlpool are examples of well-known decentralised Mixer services.

2.5.2.2 CoinJoin

CoinJoin is a PET proposed in 2013 by Gregory Maxwell, one of the developers of Bitcoin, where multiple Bitcoin users join the same transaction to obscure their transaction activity [118]. The primary objective of the CoinJoin method is to reduce the effectiveness of the deanonymisation process with a multi-input address clustering heuristic by combining unrelated transactions from multiple users in a single transaction, thereby causing the multi-input address clustering heuristic to inaccurately cluster a large number of Bitcoin users into the same entity. The CoinJoin method can also reduce the accuracy of transaction tracking by taint analysis due to a large number of transaction outputs unrelated to the tainted Bitcoins inputs [117, 120].

The CoinJoin method can be performed manually between a group of users or via a service. However, there is a privacy issue involving the CoinJoin service where the service can observe and record the exact distribution of Bitcoins in the transactions they create. Chaumian CoinJoin is a subtype of the CoinJoin protocol, which incorporates the Chaumian or Schnorr Zerolink protocol that prevents CoinJoin coordinators (services) from observing the connection between transaction inputs and transaction outputs [37]. The Chaumian CoinJoin allows users to connect to the CoinJoin coordinator server and submit their transaction inputs and outputs under different identities with blind signatures. The coordinator verifies the submitted outputs while not being able to link them to the transaction inputs.

2.5.2.3 Off-chain transactions

Off-chain transactions are an external mechanism that allows Bitcoin users to exchange Bitcoins outside of the blockchain. One example is the Lightning Network protocol. As exchanges of Bitcoins in the off-chain transaction system are not recorded inside the blockchain, users can evade blockchain transaction tracking by

spending their Bitcoins via the off-chain transaction system.

The Lightning Network is an off-chain transaction that allows two or more Bitcoin users to exchange their Bitcoins without requiring any confirmation within the Bitcoin blockchain. The Lightning Network channel can be created by any Bitcoin user, which appears in the Bitcoin blockchain in the form of a P2WSH output to a multi-signature address (bech32). Bitcoin users first set up a Bitcoin Lightning node and send Bitcoin funds to the Lightning Network multi-signature address to create a network channel [206]. This transaction is typically referred to as a funding or opening transaction. Next, the users can connect to other lightning nodes, which will allow them to exchange Bitcoins with other users in the network.

The Lightning Network channel has a maximum Bitcoin capacity limit per channel. The first version of the Lightning Network has a maximum channel capacity of 0.042 BTC [157], while the maximum capacity was increased to 0.167 BTC in the version 0.10 update in 2020 [158]. Since its beta launch in 2018, the maximum capacity of the Lightning Network channel has become varied as more alternative Lightning Network protocols are introduced. For example, Bitrefill²⁵ introduced the first Lightning Network node with 1 BTC capacity limit in 2019 [26] and Bitfinex introduced a Lightning Network node with 2 BTC capacity limit in 2020 [25]. It is possible to filter the Lightning Network transactions using the Bitcoin amount limit on the potential outputs to reduce the false positive results further.

Bitcoin exchanges between users inside a Lightning Network channel are completely invisible from the Bitcoin blockchain and can be performed without any limitation until the channel is closed. Upon closing, the channel's address will distribute the Bitcoins back to users' addresses according to the closing balance in a transaction called settlement or closing transaction that will appear in the Bitcoin blockchain [148].

Therefore, transactions within Lightning Network channels are immune to blockchain tracking techniques, such as taint analysis. However, the Lightning Network is not without fault as the privacy of Bitcoin exchanges inside Lightning Network channels can be compromised because every transaction activity inside the network can be observed by anyone that joins the network [170, 179]. As such, outsiders can conduct

²⁵<https://www.bitrefill.com>

a probing attack to obtain transaction information with minimal cost [85, 86].

2.5.3 Alternative Privacy Cryptocurrencies

Another common privacy technique that Bitcoin users can utilise is cross-currencies trading, where users exchange their Bitcoins to other alternative cryptocurrencies that have better privacy features, either with cryptocurrency exchange services or with other cryptocurrency users instead of spending Bitcoins directly. Users can then anonymously spend their exchanged coins and improve their online privacy.

There have been many new developments of blockchain protocol and alternative cryptocurrencies to solve privacy issues stemming from the transparent blockchain in Bitcoin. Two of the most notable alternative cryptocurrency examples are Zcoin and Monero.

Zerocoin protocol is a cryptocurrency privacy protocol introduced by Misers et al. [122] that proposes a cryptographic extension for the blockchain protocol to prevent transaction tracking with blockchain analysis. The Zerocoin protocol “removes” the coins when they are spent in a transaction and creates entirely new coins, similar to when new Bitcoins are mined. The Zerocoin transactions are verified by *zero-knowledge proof* protocol that allows users to prove a spending condition without revealing any other information. The Zerocoin protocol also provides an improvement to users’ privacy by hiding the transaction amounts and address balances in the blockchain data. There are many alternative cryptocurrencies that implement the Zerocoin protocol, with some of the most popular Zerocoins being Zcoin and Zcash.

Cryptonote protocol is another cryptocurrency privacy protocol proposed by Saberhagen [180] that aims to solve the various issues in Bitcoin protocol, including transaction traceability. Cryptonote protocol utilises the Diffie–Hellman key exchange method, where the transaction senders and recipients exchange a “secret key” to sign the transactions. Only users with the secret key can access and view the transaction data. Therefore, the protocol allows blockchain data to hide transaction information from those without the secret key. Cryptonote protocol replaces the Bitcoin address signature protocol with a one-time ring signature protocol that allows a group of users to hold keys to a single address and anonymously sign transactions.

Monero is one of the alternatives cryptocurrencies that implement the Cryptonote protocol [138].

2.6 Conclusion

As detailed in this chapter, Bitcoin is an electric currency that provides anonymity to its users in the form of pseudonymous addresses, which facilitate illegal activities on the Internet. However, since Bitcoin utilises a transparent blockchain system that allows third-party observers to identify and track any Bitcoin and address of interest, this creates an opportunity for the creation of tracking and deanonymisation methods to combat crimes.

Although there have been various studies that propose Bitcoin tracking solutions, there are also newly developed PETs that reduce their practicality or render them obsolete. The following chapters aim to improve upon the established cryptocurrency tracking methodologies and propose novel methodologies of our own to overcome the challenges that are present in the current state-of-the-art methodologies.

Chapter 3

Context-based Tracking

Research into Bitcoin tracking remains a relevant subject due to the need to identify and trace Bitcoins related to illegal activities, such as ransomware, sales of illicit goods, tax evasion, and cryptocurrency theft. For example, Singaporean cryptocurrency exchange, KuCoin, was hacked in September 2020 and lost around 1,000 Bitcoins (66 million USD) [93]. While Bitcoin is no longer the cryptocurrency with the most effective *tracking resistance*, compared to newer cryptocurrencies with additional privacy protocols (e.g., cryptocurrencies like Zcoin [122] and Monero [138]), it remains the most prominent and valuable cryptocurrency in use today due to its high acceptability [88] and pseudonymity system to protect its users' identities [30, 43, 133]. This makes Bitcoin attractive to individuals who are looking for a less traceable currency, compared to traditional currencies.

There have been a few studies that propose methodologies for Bitcoin tracking named taint analysis, which we discussed in Section 2.4.1. Taint analysis in its current state is still the state-of-the-art approach that various recent studies adopt for Bitcoin tracking [16, 144, 198] and analysis [47, 198]. However, the tracking of Bitcoins is still challenging due to the current tracking methodology only following Bitcoins' movement from one address to another even if they are long exchanged to other unrelated users and the rise of new PETs like the CoinJoin method or mixer services that allow individuals to evade Bitcoin tracking [120]. There has been no significant improvement in the precision of the tracking of the movement trail of individual users' Bitcoins. Therefore, our work contributes to the development progress of cryptocurrency tracking, which can assist cybersecurity in combating

cryptocurrency cybercrimes.

In this chapter, we propose a methodology to improve the precision of Bitcoin tracking by making the tracking process adaptable to the context of address ownership and tracking evasion. The tracking process will stop tracking Bitcoins that are considered to not be in the hands of the targeted users (e.g., illegal Bitcoin users that steal targeted Bitcoins) any longer, and thereby continuing the tracking on these Bitcoins would not provide meaningful information (we refer to this as *unessential tracking*). We conduct an experiment to illustrate the application of our methodology using historic Bitcoin theft incidents as sample cases. The tracking and PETs profiling in this work also includes recently developed PETs that previous tracking studies did not consider, such as decentralised mixer services and Lightning Network transactions.

The remainder of the chapter proceeds as follows. We detail our context-based tracking methodology in Section 3.1. We present the sample cases we investigate and the criteria we use to build control groups for our experimentation in Section 3.2. We then discuss the results we obtained in Section 3.3. Lastly, we conclude in Section 3.4.

3.1 Context-based Bitcoin Tracking Methodology

We discuss the data gathering process for address profiling in Section 3.1.1 and transaction profiling in Section 3.1.2. We then introduce the context-based taint analysis strategies in Section 3.1.3 and propose the address and transaction metric for evaluation in Section 3.1.4.

The taint analysis operates by tracking Bitcoins with a specific tracking strategy (e.g., Haircut or FIFO). The tracking process typically produces transaction trails unrelated to the targeted users' activities as it does not differentiate the ownership of addresses that received the tainted Bitcoins. The effectiveness of Bitcoin tracking can be improved by integrating the context information of the targeted Bitcoins, transactions and addresses involved. By context we mean information external to the blockchain that informs on the nature of some transactions and addresses (i.e., transactions known to be illegal acts or addresses identified to be cryptocurrency

services), as well as knowledge of practices inside the Bitcoin ecosystem which could be recognised (patterns of PETs). Therefore, the key principle for our methodology is that the tracking process should take into consideration the background of the targeted Bitcoins, purposes of transactions, and ownership of addresses that are being tracked and adapt its tracking operation accordingly. We formulate the methodology with three main aspects as follows.

1. The modelling of the Bitcoin tracking using the context of address profiling based on identified service and mixer addresses (Section 3.1.1) and identified PET transactions, such as CoinJoin and Mixer Services, using the identified transaction patterns, and our hypothesised properties of potential PET transactions (Section 3.1.2). The purpose of the address profile is to determine the tracking scope and influence taint analysis methodology, while the PET transaction profile is for tracking results' evaluation.
2. The introduction of two context-based taint analysis strategies that we compare as part of our evaluation (Section 3.1.3).
3. The evaluation of the tracking outcomes with a set of address and transaction metrics. The evaluation metrics are potential characteristics based on the background of the targeted Bitcoins (Section 3.1.4).

This tracking methodology is specifically designed for cryptocurrencies with an open blockchain system, such as Bitcoins and other similar cryptocurrency coins, and is less applicable to cryptocurrencies with obscured blockchain like Zcoin that use Zerocoin protocol [122] and Monero [138] because the tracking process relies on the transaction and address information from within the blockchain.

The context-based tracking methodology process can be summarised as illustrated in Figure 3.1. First, we gather address profile data of identified service and mixer addresses and incorporate it into the tracking algorithm to tailor the taint analysis process. Second, we gather transaction profiling for PET service transactions using identified transaction patterns, which we use only for evaluation purposes. Third, we collect transaction data of known Bitcoin theft cases from publicly available sources for sample cases and select control groups with similar transaction characteristics. Fourth, we perform taint analysis with two established taint analysis strategies, namely FIFO and LIFO, and two context-based taint analysis strategies,

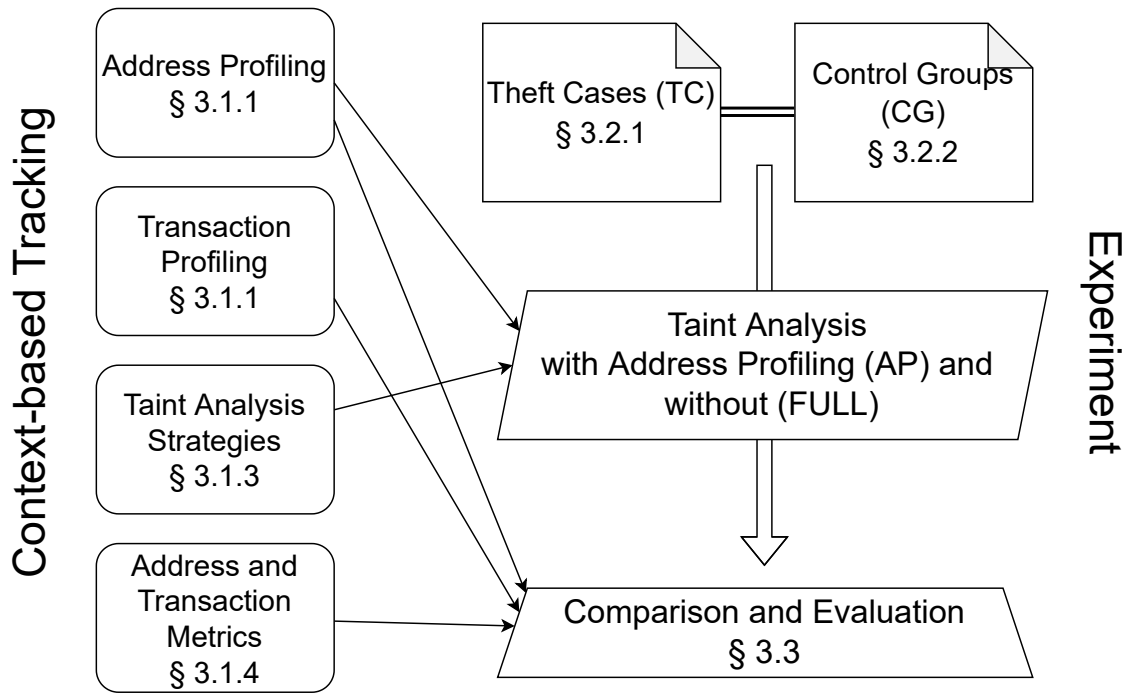


Figure 3.1: Methodology process

namely *Dirty-First* and TIHO (Taint-In, Highest-Out) strategies. We choose to implement several taint analysis strategies to perform Bitcoin tracking instead of choosing one strategy to accomplish such a task. Our rationale for employing multiple strategies is that each strategy has its own strengths and weaknesses that can affect the accuracy of the tracking results. We choose not to implement the Poison and Haircut tracking strategies due to the fact that these two methods render an enormous number of tainted transactions, as mentioned in Section 2.4.1. The taint analysis process algorithm we utilise in this work is shown in Algorithm 2, and the algorithm for each taint analysis strategy is shown in Section B.3.2. Lastly, we compare and evaluate the taint analysis results of sample cases and control groups with the evaluation metrics that are based on address and transaction behaviour.

3.1.1 Address Profiling

The tracking process can be improved with the implementation of address profiling into the tracking algorithm to prevent the process from following unrelated transactions. In order to achieve this, we require information that can indicate the entities behind pseudonymous addresses and the type of such entities (service or PETs). As such information typically does not exist in the blockchain, we obtain this informa-

tion from external sources.

The gathering process for the address profile data can be classified into three stages. First, we retrieve address profile data from previous studies on Bitcoin address classification and mixer service analysis. Second, we implement a web scraping process on a public Bitcoin address tagging website to obtain more cryptocurrency service address profile data. Third, we employ the multi-input address clustering heuristic using the algorithm and parameter shown in Algorithm 14 on the scraped service address data and the mixer deposited address data obtained from the previous mixer analysis studies. Therefore, the address profile data consists of addresses belonging to various types of cryptocurrency services and mixer services.

There are three limitations and risks in the gathering process we described above. First, the address profile data that is publicly available is likely to account for only a small proportion of all addresses that belong to cryptocurrency services, which means that the tracking process can still track tainted Bitcoins that pass through unidentified service addresses (false negative). Second, there is a possibility that the address profile data we retrieve is inaccurate and contains false positive results. We address the second risk by including additional data verification in the second gathering step (our gathering via the web scraping) to ensure that the address data we scraped does not contain false positive results (see Section 3.1.1.1). Third, the accuracy of the multi-input address clustering heuristic that we employ in the third gathering stage can be significantly reduced because of the CoinJoin method. We argue that the heuristic can still be reliably used to cluster addresses that belong to a cryptocurrency service because such services have less need to use additional privacy protections. Furthermore, the findings of the studies that analyse the mixing mechanism of the profiled mixer services indicate no evidence of employing the CoinJoin method in their mixing operations.

3.1.1.1 Identified Service Addresses

One aim of our tracking methodology is to identify addresses that involve a change of ownership or exit points of targeted Bitcoins. We set the assumption that the ultimate purpose of illegal Bitcoins is to be exchanged for other currencies, goods and services in either virtual or physical form, as indicated in various tracking analyses

Table 3.1: Identified service address data

<i>Service Type</i>	<i>Total Addresses</i>	<i>Total Entities</i>	<i>% Addresses Added</i>
Exchange	9,721,443	155	7.4%
Darknet Market	2,325,733	17	0%
Payment	13,247,057	14	0%
Gambling	2,822,760	124	0.82%
Mining	625,931	46	0.88%
Wallet	1,923,853	16	0.005%
Mixer	1,565,880	10	85%
Other	262,660	272	90.61%
Total	32,495,417	644	7%

The “% Addresses Added” column indicates the proportion of addresses that we add to the existing datasets from our data gathering. The other service category consists of cryptocurrency faucets, E-commerce businesses, service donation addresses, and other types of cryptocurrency services.

from various public research and blockchain analysis organisations’ reports [58, 92, 152, 186]. Therefore, we consider service addresses to be the end goal or exit point of the targeted Bitcoins and will stop tracking for those specific Bitcoin outputs. We classify any address that belongs to cryptocurrency services such as cryptocurrency exchange services, online gambling services, e-commerce businesses, marketplace services (including dark-net market), or payment services that users can exchange their Bitcoins for other currencies or goods to be service addresses.

We retrieve address profile data from studies [95, 96], which publish their Bitcoin service address profile data that are labelled into six different service categories, which are exchange, service, gambling, mining, darknet market, and historic addresses (no longer operational). We classify cryptocurrency faucets, e-commerce businesses, service donation addresses, and other kinds of cryptocurrency services as other services. We also re-categorise the historic type services to their appropriate service type, as shown in Table 3.1. The total number of service addresses and entities we obtained from these two studies are shown in the mentioned table.

We use a similar address profile data gathering methodology as in the previous studies [95, 96] to obtain more address profile data, but with an additional verification process. We first utilise a web scraping script on the CheckBitcoinAddress website¹ to obtain data of addresses that are reported as belonging to a cryptocurrency service. Additionally, we verify scraped address data to ensure that the addresses belong to a cryptocurrency service by manually searching the scraped addresses on public websites and removing addresses that we can not find public evidence of

¹<https://checkbitcoinaddress.com> is a reporting and labelling website that allows users to report Bitcoin addresses with an identity profile.

ownership by the associated service entity. We subsequently perform multi-input address clustering heuristics on the scraped addresses to expand the address data. The total number of service addresses we obtained with the above method is shown in Table 3.1 at the “% Addresses Added” column.

3.1.1.2 Identified PET Addresses

The presence of a mixer service can indicate the points where taint analysis tracking is no longer effective because of the creation of zero-taint Bitcoins, as mentioned in Section 2.5.2.1. Therefore, we can consider the targeted Bitcoins that reach identified mixer service addresses no longer traceable with the taint analysis strategies and will stop tracking for those specific Bitcoin outputs.

The results of the reverse-engineering experiments in the previous studies [47, 127, 172, 181] discovered that the majority of mixer services typically utilise deposited addresses to receive Bitcoins from users before transferring the deposited Bitcoins to their central address(es) for further mixing.

There are two types of information sources of identified mixer address data which we use for address profiling as follows:

1. The address data from the two studies [95, 96], which contains addresses belonging to three prominent mixer services; Bitcoin Fog, BitLaunder and HelixMixer.
2. The deposited address data of Bitcoin Fog, Bitlaundry [127], BitLaunder, DarkLaunder, Alphabay [47], Helix Light [47, 89], BestMixer [136], Bitmix.biz [199] and two unnamed mixer services [181].

The total number of mixer service addresses we obtained from the previous studies is shown in Table 3.1.

The studies of the second source type analyse the mixing mechanism of the mixer services by using the services, and their findings indicate no evidence of CoinJoin method in their mixing operations. We performed multi-input address clustering heuristics on the deposited addresses we retrieved to obtain other deposited addresses belonging to the mixer services. The total number of mixer service addresses we expanded with multi-input address clustering heuristics in this experiment is shown in Table 3.1 at the “% Addresses Added” column.

3.1.2 Transaction Profiling

In order to accurately analyse the movement of stolen Bitcoins, we require identification or a method that can indicate the purpose of the transactions, especially for those that involve PETs for tracking evasion.

For the PET transaction data gathering process, we first derive transaction classification methods that can identify PET transactions based on their unique transaction patterns. Subsequently, we employ the transaction classification methods for each PET on every transaction in the blockchain and label any transaction that matches transaction patterns with the classification methods as a PET transaction.

The transaction profiling process has one crucial limitation, which is the lack of ground-truth data to verify the PET transactions and ensure that the classification methods will not produce false positive and false negative results. Therefore, we do not stop taint analysis operation for tainted Bitcoins that move through either identified or potential PET transactions and utilise the transaction profiling primarily for the evaluation metric in this experiment. The table below shows the total number of transactions for each identified PET.

Table 3.2: Identified transaction profile data

<i>PETs</i>	<i>Total Transactions</i>
ChipMixer	85,950
JoinMarket CoinJoin	763,827
Wasabi CoinJoin	12,192
Samourai Whirlpool CoinJoin	108,824
Lightning Network Channel	58,708
Total	1,029,495

3.1.2.1 ChipMixer Transactions

ChipMixer² is a well-known mixer service that is different from the mixer services previously mentioned in Section 3.1.1.2. The reverse-engineering experiment on the ChipMixer service indicates that the service’s mixing transaction has a unique but static transaction characteristic³ that is distinct from common transactions [199].

According to the ChipMixer analysis findings of the previous research [199], the ChipMixer’s mixing protocol always distributes Bitcoins to transaction outputs

²<https://chipmixer.com>

³The service always performs the mixing transactions in one specific way.

(referred to as “chips”) of exactly the same value and in a round number with three decimal places (e.g., 0.005 BTC and not 0.0055 BTC). The chip outputs also can not have their value lower than 0.001 BTC or higher than 4.096 BTC. However, each mixing transaction can have one transaction output that is an exception to the mentioned rule, which is a transaction output that receives the mixing fee or donation from users to the service. The Algorithm 12 demonstrates the algorithm we use to identify the ChipMixer’s mixing transactions in this work.

3.1.2.2 CoinJoin Transactions

We utilise the CoinJoin transaction classification provided by the BlockSci library tool to detect CoinJoin transactions performed by JoinMarket⁴, which is one of the most prominent mixing services that allows users to engage in CoinJoin mixing together. The CoinJoin classification is based on the JoinMarket’s CoinJoin transaction detection algorithm presented in a previous study [75].

We also utilise a classification of CoinJoin transactions performed by two other well-known CoinJoin services, namely Wasabi Wallet⁵ and Samurai Wallet⁶. For the Wasabi CoinJoin transaction detection, we use the static coordinator address belonging to the service as a classification method [195]. However, the Wasabi wallet service no longer uses static coordinator addresses to perform CoinJoin transactions as of February 2020 and uses fresh coordinator addresses for every transaction. Hence, we obtained additional Wasabi CoinJoin transaction data from the previous study that retrieved more transactions directly from the service’s public API [199]. We discuss the Wasabi service later in Chapter 5 in more detail.

For the Samurai CoinJoin transaction detection, we use the transaction characteristics that the whirlpool mechanism employs. The whirlpool protocol always performs the CoinJoin mixing with five input addresses to five output addresses. The Samurai CoinJoin transactions must have five transaction outputs with the exact same value of either 0.01, 0.05 or 0.5 BTC, as well as five transaction inputs with a Bitcoin value no less than the transaction outputs value [159]. The Algorithm 11 illustrates how we apply the above characteristics to identify the Samurai

⁴<https://github.com/JoinMarket-Org/joinmarket-clientserver>

⁵<https://wasabiwallet.io>

⁶<https://samuraiwallet.com>

Wallet’s mixing transactions.

The total number of identified CoinJoin transactions we obtained in this experiment is shown in Table 3.2.

3.1.2.3 Lightning Network Transactions

The Lightning Network transactions consist of a funding transaction and a closing transaction that appear in the blockchain data, as mentioned in Section 2.5.2.3. These two transactions typically have unique transaction characteristics that can be used to differentiate Lightning Network transactions from other transactions [77, 141]. Therefore, it is possible to identify Lightning Network channel transactions using only blockchain data.

A Lightning Network funding transaction can be potentially identified from the existence of at least one multi-signature address output, and the closing transaction always has only one transaction input from the funding transaction and two transaction outputs. The transaction output value must also be in the channel capacity limit of 0.042 BTC for transactions that occurred before the capacity update on 2020-04-29 and 0.167 BTC for transactions that occurred after the update. Therefore, we classify potential Lightning Network transactions based on this transaction pattern. The algorithm that we use to detect lightning network transactions is shown in Algorithm 13.

3.1.2.4 Potential PET transactions

As our identified PET profile data is likely to contain only a fraction of all PET addresses and transactions in the Bitcoin blockchain, we design a classification method to identify transactions that may involve PETs based on the pattern recognition of transaction characteristics. Typically, the most distinguishing characteristic of the transactions that involve PETs is the inclusion of other unrelated or clean Bitcoins in the transaction inputs.

We propose a classification method for transactions that potentially involve PETs as follows: the tainted transaction will be classified as a potential PET transaction if there are transaction input(s) with completely clean Bitcoins (i.e., Bitcoins unrelated to the tainted Bitcoins), which can be an indication that Bitcoins belonging to

other users are being mixed with tainted Bitcoins by either mixer services or the CoinJoin method. The transaction must also have no transaction inputs from service addresses. The algorithm that we use to detect potential PET transactions is shown in Algorithm 16.

3.1.3 Context-based Taint Analysis Strategy

To make taint analysis more efficient, we include into the taint analysis transaction characteristics that are relevant to the targeted Bitcoins. As such, we propose two additional strategies, namely Dirty-First and TIHO, in this experiment.

3.1.3.1 Dirty-First Strategy

As mentioned in Section 3.1.2.4, when tainted Bitcoins are used in a transaction with clean Bitcoins, this event may indicate that the tainted Bitcoins were obscured by PETs, especially in the case of illegal activities where illegal Bitcoin users are less likely to combine stolen Bitcoins with their other clean Bitcoins since this would expose their other Bitcoin activities⁷. Meanwhile, if there is no clean Bitcoin involved, there is high certainty that the stolen Bitcoins still belong to the illegal Bitcoin users. The assumption is based on what we previously mentioned that illegal Bitcoin users are more likely to utilise PETs [35, 160] and that the majority of cryptocurrency PETs, including CoinJoin and centralised mixer services, operate by combining multiple unrelated Bitcoins together to obscure their movement [47, 199].

To illustrate and analyse the tracking results of fully tainted Bitcoins, we propose a taint analysis strategy named Dirty-First.

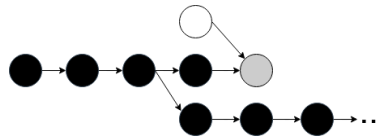


Figure 3.2: Dirty-First strategy

White circles represent clean Bitcoin outputs, black circles represent fully tainted Bitcoin outputs, and grey circles represent partly tainted Bitcoin outputs that contain clean Bitcoins of any amount. Black arrow lines indicate transactions that are being tracked, and three dots represent further tracking.

⁷It should be noted that this assumption is less applicable for cases unrelated to illegal activities as it is not completely uncommon for Bitcoin users to combine their Bitcoins.

The Dirty-First strategy has a similar concept as the Depth-First Search algorithm⁸ where it tracks only fully tainted Bitcoins. The Dirty-First strategy stops tracking tainted Bitcoin outputs if there are clean Bitcoins in transaction inputs regardless of their number, as illustrated in Figure 3.2. It is worth noting that the Dirty-First strategy produces tracking results that are a subset of other taint analysis strategies' results (i.e., the results of the Poison, Haircut, FIFO, LIFO, and TIHO strategies would contain fully tainted Bitcoin transactions in the Dirty-First strategy's results for the same tracking case).

The Dirty-First strategy has an advantage in that the strategy can create a network of transactions that are most likely to be performed by the targeted illegal Bitcoin users due to the lack of clean Bitcoin mixing. The Dirty-First strategy's tracking results should be able to illustrate the transaction behaviour of illegal Bitcoin users with the least number of false positive results. However, the Dirty-First strategy also has a disadvantage in that it may be possible for illegal Bitcoin users to mix their tainted Bitcoins with their own clean Bitcoins of any amount, which would cause the Dirty-First strategy to misclassify those transactions as false negative and stop tracking even if the transactions afterwards are still performed by the illegal Bitcoin users.

3.1.3.2 TIHO Strategy

We introduce a taint analysis strategy named TIHO (Taint-In, Highest-Out), which prioritises the distribution of the tainted inputs to the highest value outputs, as shown in Figure 3.3.

The Taint-In strategy possesses an advantage in that the strategy performs a targeted tracking on tainted Bitcoins based on specific transaction patterns rather than purely on the arbitrary transaction order like the FIFO and LIFO strategies.

The TIHO strategy is based on the fact that the primary purpose of PETs like the CoinJoin method is to make it difficult to identify and prove the receiving addresses of obscured Bitcoins. Illegal Bitcoin users who utilise non-zero-taint PETs instead of those that can produce zero-taint Bitcoins (completely immune to taint analysis tracking) are likely to trust these PETs enough that they can safely ex-

⁸Depth-First Search (DFS) is a graph analysis algorithm that performs a search from a starting graph node to the subsequent nodes as far as possible before backtracking.

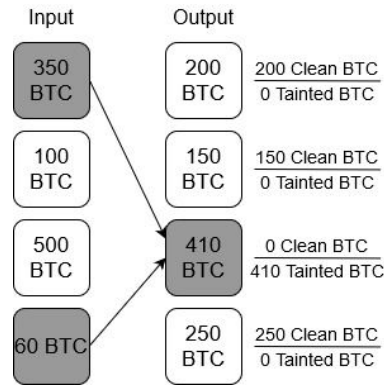


Figure 3.3: TIHO strategy

White rectangles represent clean inputs or outputs, and dark rectangles represent fully tainted ones.

change their stolen Bitcoins to other values without requiring to distribute stolen Bitcoins into smaller proportions. Additionally, previous research [160] shows that most of the well-known Bitcoin theft and ransomware incidents typically involve a significantly large amount of Bitcoins. Therefore, we set the assumption that when stolen Bitcoins are obscured by the CoinJoin method, the highest value outputs are most likely to be stolen Bitcoin outputs because the number of stolen Bitcoins is typically higher than average users' Bitcoins.

Using the example shown in Figure 3.3, the TIHO strategy starts by distributing the tainted 350 BTC and 60 BTC inputs to the highest value output of 410 BTC. Then, the TIHO strategy distributes the remaining clean inputs to the rest of the other lower value outputs. As a result, only the highest value 410 BTC output will be considered as tainted, and the other three outputs will contain no tainted Bitcoins. It should be noted that we do not account for the transaction fee in this transaction example.

The Taint-In strategy should be beneficial for tracking Bitcoins that pass through PETs like the CoinJoin method by distributing tainted Bitcoins to transaction outputs that are likely to belong to the illegal Bitcoin users. However, the Taint-In strategy has a disadvantage similar to the other taint analysis strategies where it can not track zero-taint Bitcoins produced by mixer services.

3.1.4 Tracking Evaluation Metrics

We hypothesise that the characteristics of transactions and addresses in Bitcoin theft cases are distinguishable from those involved in non-illegal Bitcoin activities due to the attempts to evade tracking and legal enforcement. As illustrated in previous studies that Bitcoin users typically are not privacy-conscious in their Bitcoin activities [65, 79], while illegal Bitcoin users are more privacy-conscious and make use of PETs to obscure their Bitcoins [35, 160]. Therefore, it may be possible to build evaluation metrics that measure the performance of tracking results based on the potential characteristics of Bitcoin theft cases. We propose evaluation metrics and present a corresponding hypothesis based on the behaviour of Bitcoin privacy practices and PETs.

We define six evaluation metrics in total, and the hypothesis for each of those metrics are as follows.

H1 (Transaction Frequency) The number of transactions (per day) for Bitcoin theft cases is significantly high.

H2 (PETs Detection) There are a significant percentage of PET transactions in Bitcoin theft cases.

H3 (Reused Address) The proportion of reused addresses is low for Bitcoin theft cases.

H4 (Fresh Address) The majority of tainted addresses in Bitcoin theft cases are fresh addresses.

H5 (Number of Addresses per Transaction) The majority of transactions in Bitcoin theft cases contain a large number of addresses.

H6 (Transaction Fee) The transaction fee of the majority of transactions in Bitcoin theft cases is high.

3.1.4.1 Transaction Frequency (H1)

We expect the number of tainted transactions per day to be high for Bitcoin theft cases because of the common transaction obscuring or privacy technique that involves distributing Bitcoins in multiple transactions and addresses to increase the difficulty of tracking. It is unlikely for non-illegal Bitcoin cases to employ this tech-

nique because transaction senders typically have to pay a transaction fee for each transaction (see Section 3.1.4.6), which can incur a significant loss of Bitcoins due to a large number of transactions.

3.1.4.2 PETs Detection (H2)

We include the identified PETs' profile data (both addresses and transactions) and potential PETs classifications as an evaluation metric to identify PETs' usage and strategies that obscure stolen Bitcoins. We anticipate that the number of transactions involving PETs such as the CoinJoin method or a mixer service is different depending on the privacy requirement. Hence, there should be a significantly large number of transactions involving PETs for Bitcoin theft cases as illegal Bitcoin users would likely utilise PETs several times to obscure the transaction trails.

3.1.4.3 Reused Address (H3)

While privacy protection is often considered to be one of the most important aspects of Bitcoin among its user-base, many Bitcoin users do not seem to be privacy-conscious, as can be observed from a large number of reused addresses discovered in the previous research [65, 79, 155]. These findings provide us with a valid reason to assume that there is a high chance that the number of reused addresses involved in transactions with stolen Bitcoins, which benefit the most from privacy measures, is minimal, compared to non-illegal Bitcoin cases. Hence, we propose a *reused address*, which is an address that has been used in transactions more than once as one of the evaluation metrics.

3.1.4.4 Fresh Address (H4)

Following the reused address metric, we assume that illegal Bitcoin users would create new addresses every time they distribute stolen Bitcoins to avoid reusing previous addresses. Thus, we expect that the significant majority of tainted addresses in Bitcoin theft cases to be *fresh addresses*, which are addresses that do not have any transaction activity before receiving any stolen Bitcoin. It is worth noting that both reused and fresh address metrics do not include identified service and mixer addresses.

3.1.4.5 Number of Addresses per Transaction (H5)

Based on the privacy technique mentioned in the transaction frequency metric (Section 3.1.4.1), we anticipate the majority of transactions involving stolen Bitcoins to be distribution transactions for obscuring. We expect distribution transactions in Bitcoin theft cases to have a large number of addresses per transaction in order to distribute stolen Bitcoins to multiple addresses and make tracking more difficult. It should be noted that the number of addresses per transaction metric includes both the input and output addresses in the transaction. For example, a 1-to-2 addresses transaction (a transaction with one input address and two output addresses) is equal to three addresses per transaction.

3.1.4.6 Transaction Fee (H6)

A transaction fee is an incentive provided by transaction initiator(s) to miners to prioritise confirming the transaction into the blockchain. A transaction fee is calculated from the difference between the total number of Bitcoins in transaction inputs and transaction outputs in a transaction [133] (e.g., a transaction with 2 BTC input and 1 BTC output has a transaction fee value of 1 BTC). Typically, the recommended transaction fee rate that Bitcoin miners charge is calculated from the data size of the transaction and the number of transactions that are currently waiting for confirmation at the time.

We implement the transaction fee as one of the evaluation metrics based on the assumption that privacy practices utilised in Bitcoin theft cases can influence the transaction fee value. For example, illegal Bitcoin users may try to obscure their transaction trail by rapidly moving the stolen Bitcoins. Therefore, they need to pay a sufficient transaction fee to accomplish this strategy⁹. The transaction fee variable we use is the ratio of the transaction fee value to transaction data size.

⁹This may not apply in case miners and illegal Bitcoin users are the same individuals or accomplices.

3.2 Sample and Control Groups Collection

We evaluate the methodology presented in Section 3.1 by applying it to known cases of transactions involving illegally-acquired Bitcoins. We explain the selection process of the sample cases for the experiment in Section 3.2.1. Section 3.2.2 provides details on the control group criteria and selection.

3.2.1 Theft Case Sample Selection

We selected a total of 26 historical Bitcoin theft cases from the year 2012 to 2021. The cases of cryptocurrency service thefts and ransomware attacks were reported either in Bitcoin news websites or on Bitcoin forums and included details of the theft transactions or the suspects' Bitcoin addresses. Such details are public information.

It should be clarified that we exclude the affected service's addresses from the address profiling implementation and evaluation of the related theft case (i.e., if a sample case involved illegal Bitcoin users stealing Bitcoins from service A, we excluded all identified addresses of service A from the address profile data when we track and evaluate that sample case). The purpose of this exclusion is to avoid potential service misclassification due to illegal Bitcoin users sharing service addresses with their addresses as transaction inputs in the same transactions since some of the sample cases involved illegal Bitcoin users hacking into the service's computer system and gaining control of the service's addresses to steal the Bitcoins. This scenario can cause the multi-input address clustering heuristic of the service address profile data to misclassify illegal Bitcoin users' addresses as service addresses.

3.2.2 Control Groups Criteria and Selection

For each sample case, we build a control group of non-illegal Bitcoin transactions that bears enough similarity to allow comparison. However, there is no reliable information to guarantee that the control transactions are not related to illegal activities. To mitigate this risk, we select multiple control transactions per sample case using the following steps. We first identify potential control transactions from all transactions in the blockchain that possess similar characteristics as the sample cases based on a set of criteria (see Section 3.2.2.1, 3.2.2.2 and 3.2.2.3). We discard

Table 3.3: Sample cases and control group number

Sample Case	Control Selection		
	Matching	Discarded	Selected
TC1	682	664	10 (18)
TC2	1,038	967	10 (71)
TC3	1,226	941	10 (285)
TC4	2	–	2 (2)
TC5	3,271	2,633	10 (638)
TC6	11	2	9 (9)
TC7	883	504	10 (379)
TC8	1	–	1 (1)
TC9	89	59	10 (30)
TC10	19	12	7 (7)
TC11	8	–	8 (8)
TC12	63,074	13,612	10 (49,462)
TC13	18,222	16,058	10 (2,164)
TC14	6,855	5,763	10 (1,092)
TC15	59,390	36,122	10 (23,268)
TC16	923	811	10 (112)
TC17	574	53	10 (521)
TC18	26	24	2 (2)
TC19	166	28	10 (13)
TC20	4	–	4 (4)
TC21	4,525	3,378	10 (1,147)
TC22	116	11	10 (105)
TC23	1,017	560	10 (457)
TC24	1	–	1 (1)
TC25	339	144	10 (195)
TC26	1,759	860	10 (899)

The numbers in parentheses in the Selected column is the total number of remaining transactions after discard.

matching control transactions that belong to the same transaction chain (i.e., we keep only the first transaction and discard the following transactions) to prevent the control groups from sharing identical results. We subsequently select from the remaining transactions the first ten that have the transaction value closest to the sample case. The limit of ten control transactions per case is chosen to reduce the computational cost while retaining a sufficient size for the analysis and evaluation and ensuring the control groups do not disproportionately represent only the sample cases with a significantly larger number of control transactions. We finally discard transactions from the control group if after applying the four taint analysis strategies, the results reach a transaction that is already included in the tainting results of the theft cases. We repeat the process until we find unrelated transactions to avoid the risk of control groups being related to the sample cases.

There are three transaction characteristic criteria that we use to identify potential

control transactions for each sample case, as we define below.

3.2.2.1 Time

To avoid selecting control transactions that can be either directly or indirectly related to the sample case, we set the time criteria to be within 60 days prior to the day when the sample cases' first distribution transaction occurred. We select 60 days periods to ensure the control transactions can be obtained in a sufficient number while still sharing similar conditions of the cryptocurrency market, PETs, and privacy practices to the respective sample case as close as possible since such factors can influence transaction behaviours in significant ways. For example, the average transaction fee rate at a specific time affects transaction fee payment, which in turn can increase or decrease the willingness of users to send transactions (transaction frequency).

3.2.2.2 Transaction value

We set the transaction value criteria to be in the 10% range of the sample value, e.g., if the sample case's distribution transaction involves 5,000 BTC, the transaction value criteria for the control transactions selection will be at between 4,500 and 5,500 BTC for that particular sample case. If the sample case is involved in multiple transactions, we will select the transaction with the highest number of Bitcoins. If the criteria of transaction with the highest value result in zero control matching, we will instead select transactions with the next highest value.

3.2.2.3 Transaction type

The transaction type refers to the number of addresses in the transaction inputs and outputs. For example, if the sample case's distribution transaction is a 1-to-2 transaction (one input address to two output addresses), the control transactions we select will also be a 1-to-2 addresses transaction. Similar to the transaction value criteria, we will use the distribution transaction with the highest value of Bitcoins.

The number of the control sample transactions that match all of the above-mentioned criteria for each TC is shown in Table 3.3 at the Control Selection/-Matching column. The number of control sample transactions that are discarded

is shown in the Control Selection/Discarded column. The total number of control samples we selected for each TC and the remaining number of matched transactions after transactions are discarded is shown in the Control Selection/Selected column (see in the parenthesis). In total, we selected 224 transactions as control groups.

3.3 Results and Discussion

In this section, we present and discuss the results of our tracking methodology for the sample cases and control groups. We performed tracking on each sample case and control case for 15 days with the FIFO, LIFO, Dirty-First, and TIHO strategies. For simplicity, we refer to the results of each sample theft and ransomware case as ‘TC’ (Theft Case). We present the results of the control group of each sample case together and refer to their results as ‘CG’ (Control Groups). We indicate tracking results with the inclusion of address profiling described in Section 3.1.1 with ‘^{AP}’ (short for Address profiling) for sample cases (TC^{AP}) and control groups (CG^{AP}). We also indicate results without address profiling with ‘^{Full}’ (short for Full results) for sample cases (TC^{Full}) and control groups (CG^{Full}). We indicate the taint analysis strategy’s results with address profiling as ‘Dirty-First^{AP}’, ‘FIFO^{AP}’, ‘LIFO^{AP}’, and ‘TIHO^{AP}’ and the full results as ‘Dirty-First^{Full}’, ‘FIFO^{Full}’, ‘LIFO^{Full}’, and ‘TIHO^{Full}’. It is also worth noting that we subtract the transaction fee from every transaction input proportionally (similar to the Haircut strategy) for all taint analysis strategies and do not taint transaction fee outputs.

The results of the control groups for each taint analysis strategy shown in this section are derived from the weighted average of all control groups’ results except for the transaction frequency (H1) metric. We use the transaction number as the weight for the transaction-related metrics which are PET transactions (H2), number of addresses per transaction (H5) and transaction fee (H6). We use the address number for the address related metrics, which are reused addresses (H3) and fresh addresses (H4).

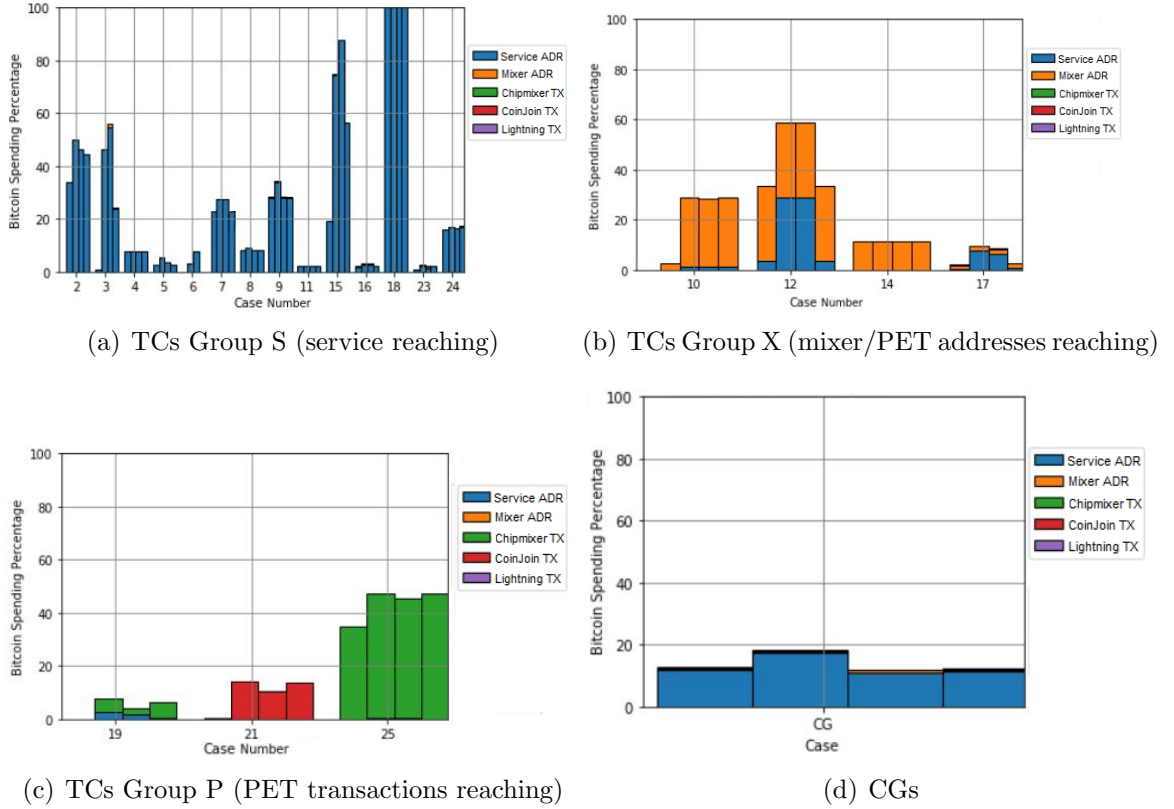


Figure 3.4: Percentage of the tainted Bitcoins reaching identified service and PETs. The bar in each case represents the results for each taint analysis strategy in the order as follows: Dirty-First, FIFO, LIFO, and TIHO. TX stands for transaction and ADR stands for address. Group U (unidentified spending) is not shown in the figure due to the lack of apparent results.

3.3.1 Address Profiling Results

The percentage shown in Figure 3.4 is the proportion of the exchanged/obscured stolen Bitcoins reaching addresses and transactions identified as belonging to a cryptocurrency service or PET, compared to the total number of stolen Bitcoins when we start tracking. The results of the sample cases can be categorised into four groups, which are sample cases that spend stolen Bitcoins with services (Group S), sample cases that obscure stolen Bitcoins with PETs that are identified in address profiling (Group X), sample cases that obscure stolen Bitcoins with identified PETs in transaction profiling (Group P), and sample cases that have a minimal number (less than 1%) of stolen Bitcoins reaching identified addresses and transactions (Group U), which are not shown in the Figure 3.4. The reason for the division between Group X and Group P is because of the lack of ground-truth data for the transaction profiling in this work, unlike address profiling. For the rest of the results, we will present the results of the sample cases based on the spending group

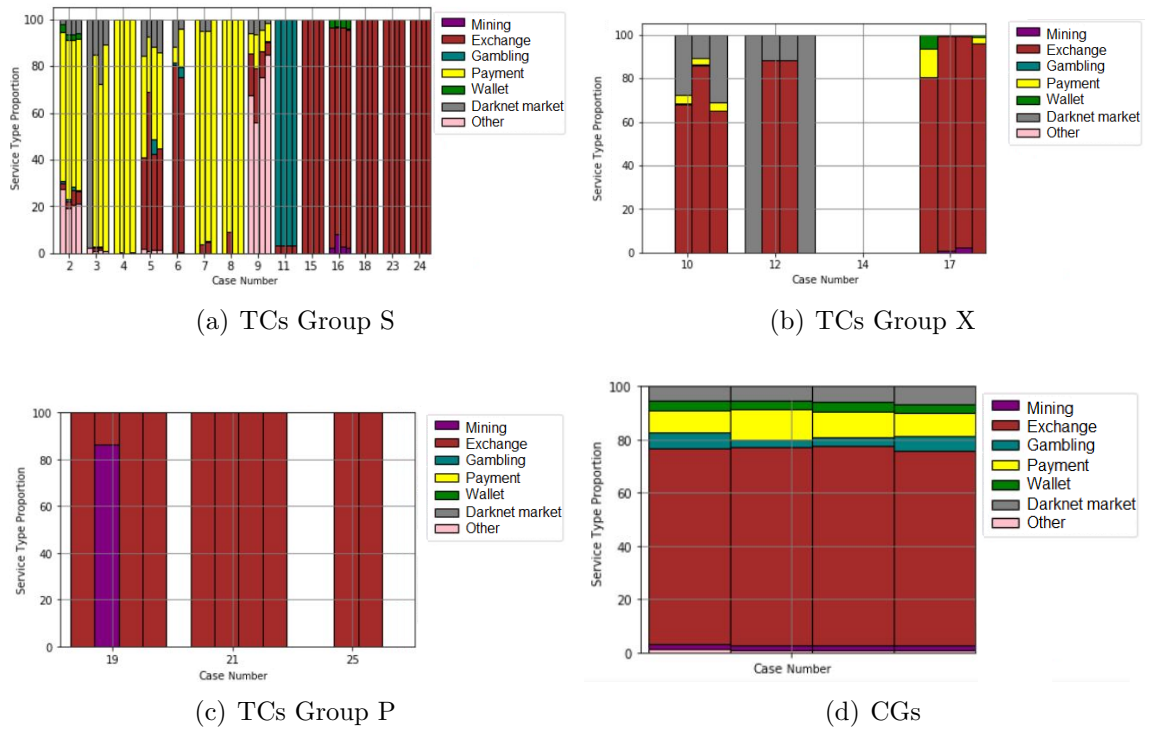


Figure 3.5: Proportion of identified service types.

The bar in each case represents the results for each taint analysis strategy in the order as follows: Dirty-First, FIFO, LIFO, and TIHO.

classification.

Unexpectedly, the majority of the sample cases show a significant percentage of the stolen Bitcoins reaching cryptocurrency services without passing through PETs (Group S), as can be seen in the Dirty-First results in Figure 3.4. For example, the Dirty-First results of cases TC2, TC7, TC9, TC15, and TC24 show around 20% of the stolen Bitcoins reaching service addresses, while case TC18's Dirty-First results show 100% of the stolen Bitcoins reaching service addresses within the first 15 days. The results of sample cases in Group S indicate that the majority of sample cases we observe may not rely on PETs to obscure the stolen Bitcoins, as indicated by the significantly high Bitcoin spending in the Dirty-First results.

The majority of sample cases in Group S typically have similar patterns for which the FIFO, LIFO, and TIHO strategies' results show a marginally different percentage of stolen Bitcoins reaching service addresses. However, two sample cases, TC3 and TC15, show significantly different results between the three strategies. Furthermore, the results of some of the sample cases in Group S show a significant increase in the number of stolen Bitcoins reaching service addresses for the FIFO,

LIFO, and TIHO strategies, compared to the Dirty-First strategy. As the sample cases in Group S contain no visible PET transactions in the results, it is possible to assume that the difference between each taint analysis strategy's results is because the stolen Bitcoins passed through unidentified service addresses. Subsequently, the unidentified service addresses combined the stolen Bitcoins with clean Bitcoins and sent them to identified service addresses afterwards. The majority of stolen Bitcoins from the sample cases in Group S reach either payment or exchange services, as shown in Figure 3.5(a). There are only four sample cases, namely TC2, TC3, TC5, and TC9, that show a substantial number of stolen Bitcoins reaching darknet markets in the Dirty-First results. The results suggest that most of the illegal Bitcoin users of these sample cases may prefer to exchange stolen Bitcoins with reliable services rather than illegal channels, despite the risk of the Bitcoins being seized by the receiving services or law enforcement.

The results of four sample cases in Group X show a significant number of stolen Bitcoins reaching identified mixer addresses, namely cases TC10, TC12, TC14, and TC17, as shown in Figure 3.4(b). On the other hand, the results of three sample cases in Group P show either the stolen Bitcoins reaching CoinJoin transactions (TC21) or ChipMixer transactions (TC19 and TC25), as shown in Figure 3.4(c). The results of these two groups indicate the difference in the illegal Bitcoin users' obscuring and spending strategies, compared to the sample cases in Group S. Intriguingly, the results of two sample cases (TC12 and TC17) in Group X and the other two (TC19 and TC21) in Group P show a small number of stolen Bitcoins directly reaching both service and mixer addresses in the Dirty-First results, as shown in Figure 3.5. These results may suggest that illegal Bitcoin users intend to spend the stolen Bitcoins in several ways. For example, illegal Bitcoin users may obscure some of the stolen Bitcoins before exchanging them with exchange services that require personal information in a large number and directly spend the rest on darknet market tradings or small number exchanges, which do not require obscuring measures.

There are some sample cases' results that show a very small number of stolen Bitcoins reaching identified cryptocurrency services or PETs, which are cases TC1, TC13, TC20, TC22, and TC26 (Group U). There is no sample case that transfers the stolen Bitcoins through Lightning Network channels in this experiment.

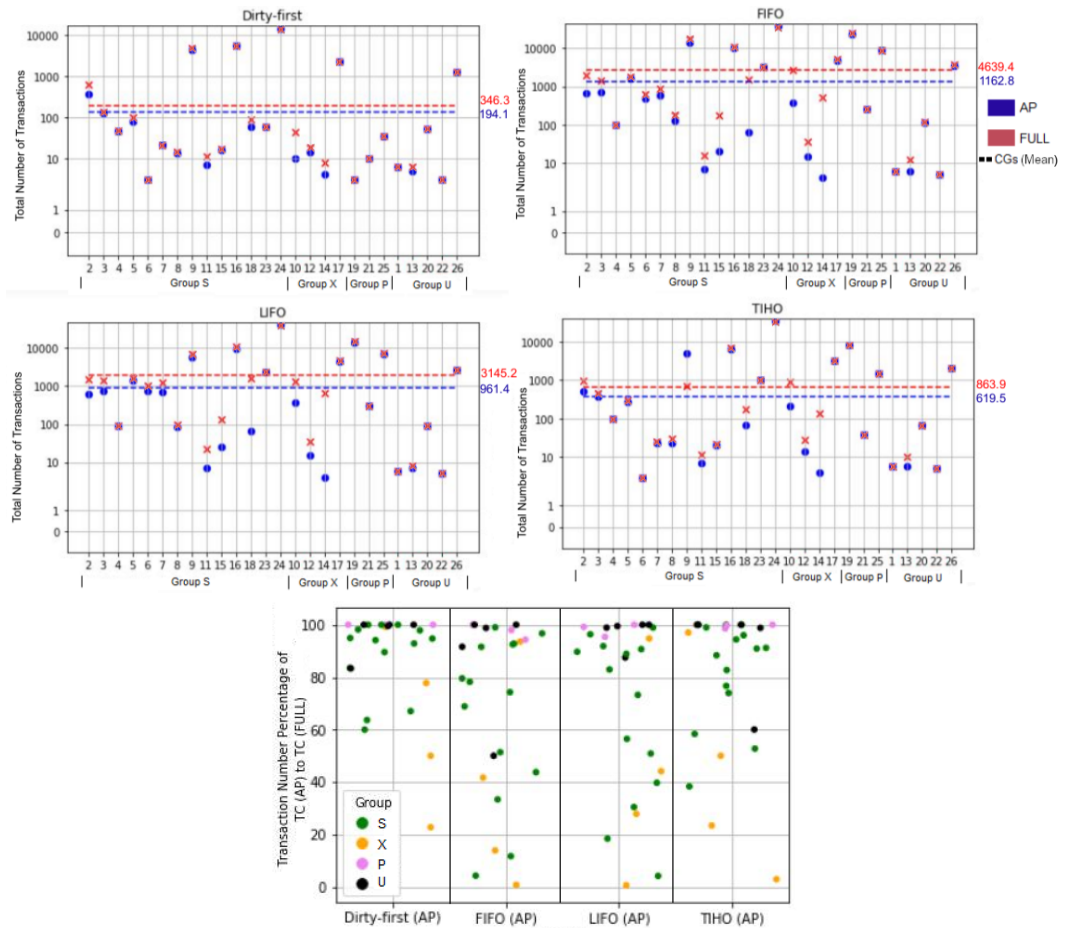


Figure 3.6: Number of transactions
Each dot represents a sample case.

The results of the control groups (CGs) and the sample cases in Group S are most similar, where both show mostly tainted Bitcoins reaching service addresses. However, when looking at the service types (Figure 3.4(d)), the results reveal the difference in the Bitcoin spending methods between the sample cases and control groups. Unsurprisingly, the majority of services in the control groups' results are exchange services followed by payment services. Interestingly, the control groups' results also show a noticeable number of Bitcoins reaching gambling and darknet market services, which are not outside our expectations as both types of services are widely reported as a significant part of the Bitcoin ecosystem [34, 44, 169] and do not necessarily indicate that the control groups are related to illegal activities since it is possible for the Bitcoins to be exchanged with other users via other unidentified services first before reaching gambling and darknet market services.

The inclusion of address profiling shows a considerable high reduction in the

number of tainted transactions for the majority of the TC^{AP} results, especially for the FIFO, LIFO and TIHO strategies, as can be seen in Figure 3.6. The Dirty-First strategy shows the least number of sample cases with a significant change in the transaction number, which suggests that cryptocurrency and mixer services typically combine the stolen Bitcoins they receive with other Bitcoins shortly after the exchanges.

The $FIFO^{AP}$, $LIFO^{AP}$, and $TIHO^{AP}$ results show a considerably distinct pattern where sample cases show a reduction in the number of tainted transactions from lower than 10% to as high as 90%. These results suggest that the illegal Bitcoin users have different spending strategies where some try to quickly spend the stolen Bitcoins to lessen the risk of the Bitcoins being blacklisted by cryptocurrency services, while the others are more cautious and likely to wait for the interest of tracking the stolen Bitcoins decline before spending them.

Intriguingly, the $TIHO^{AP}$ results show an overall lower reduction in tainted transaction numbers, compared to the $FIFO^{AP}$ and $LIFO^{AP}$ results for most sample cases. One explanation for this pattern is that the lower value outputs are utilised more as spending outputs, compared to the higher value ones that the TIHO strategy prioritises.

The total number of transactions' results for sample cases in Group U show that three out of five sample cases have a very small number of transactions, which explains the lack of Bitcoin spending. However, the results of cases TC20 and TC26 show a large number of transactions, which indicates that our address and transaction profile data are unable to identify the spending and obscuring strategies for these two sample cases.

The address profiling results demonstrate that the taint analysis tracking can benefit from the implementation of address profiling, as can be seen from the significant reduction in the unessential tracking results for multiple sample cases.

3.3.2 Transaction Frequency (H1) Results

The results of the transaction frequency metric (defined in Section 3.1.4.1) in Figure 3.7 are shown as the average number of tainted transactions per day. The results of each sample case seem to yield a considerably diverse pattern, ranging from the

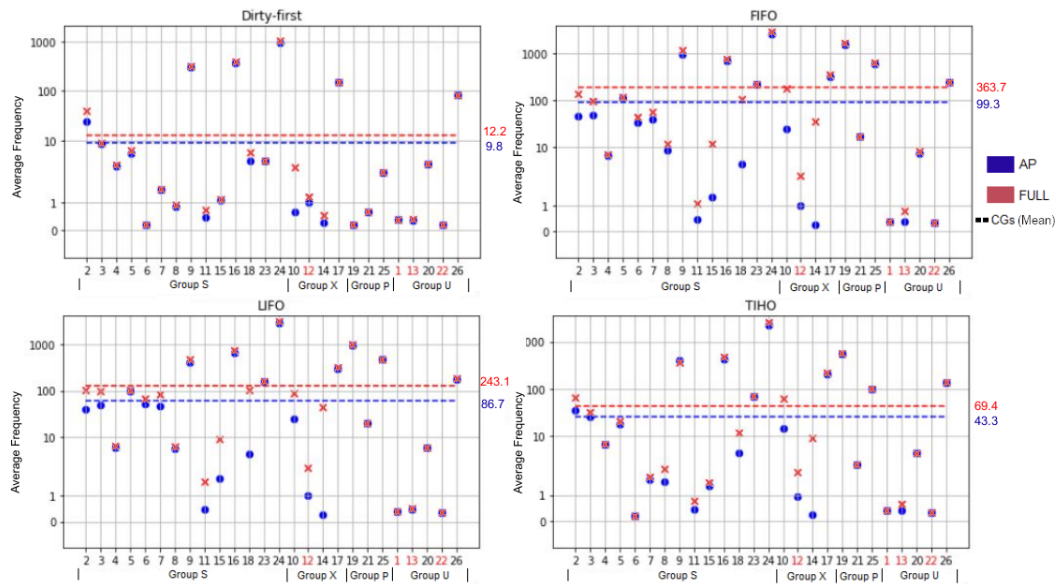


Figure 3.7: Transaction frequency (H1)

Sample cases with a red colour number are cases with low transaction activity (less than average of one transaction per day) for any taint analysis strategy.

average of one transaction to as high as 1,000 transactions per day. There are four sample cases that have an average number of transactions of less than one per day in both of the TC^{AP} and TC^{Full} results for all four taint analysis strategies, which are cases TC1, TC12, TC13, and TC22. The results suggest suggests that illegal Bitcoin users did not use the distribution technique described in Section 3.1.4.1 possibly to maximise their profit from the theft. Meanwhile, the results with high transaction activity, such as cases TC9, TC16, and TC24, indicate that illegal Bitcoin users rapidly distributed their stolen Bitcoins to increase the number of transactions that needed to be tracked and analysed.

For this work, we mainly focus on the results of the sample cases with high transaction activity as the sample cases with very low transaction activity (less than an average of one transaction per day for all results) do not provide meaningful transaction behaviour information for analysis and comparison.

Similar to the total number of transactions results (shown in Figure 3.6), the inclusion of address profiling shows a significant reduction of transaction frequency in the TC^{AP} results for the FIFO, LIFO, and TIHO strategies, as can be seen in Figure 3.7. For example, case TC10's $FIFO^{AP}$ results show an average of 24.3 transactions per day, while the $FIFO^{Full}$ results show as high as 175.1 transactions per day. However, there are few sample cases that do not show as much difference,

such as case TC24's FIFO^{AP} results, which show an average of 2,592.9 transactions per day, compared to the FIFO^{Full} results at an average of 2,794 transactions per day. This pattern is similar to the service address results presented in Section 3.3.1, where sample cases with a higher number of stolen Bitcoins reaching identified service or mixer addresses (such as case TC18) also show a higher reduction in the transaction number.

The TC^{AP} and TC^{Full} results show an overall lower number of transactions, compared to the control groups' results (CG^{AP} and CG^{Full}, respectively), especially for the Dirty-First strategy. There are a few exceptions that show remarkably high transaction frequency for all four taint analysis strategies' results, such as cases TC9, TC16, and TC24, as can be seen in Figure 3.7. This pattern indicates that for most theft cases, the illegal Bitcoin users do not distribute the stolen Bitcoins rapidly in a large number of transactions as we expected, possibly to avoid unnecessarily losing their profits because of the transaction fee. Hence, the majority of the sample cases' results do not support our H1 hypothesis that the Bitcoin theft cases would have a high transaction frequency.

Nevertheless, the transaction frequency metric shows the potential for further analysis that can assist in the effort to investigate illegal Bitcoin users' strategies. Additionally, the lack of transaction activity for some sample cases may be due to our tracking period of 15 days from the first distribution transaction. This issue can be alleviated by extending the tracking timeframe further to reveal more transaction activity that we have not yet captured for these sample cases.

3.3.3 PETs Detection (H2) Results

As discussed in Section 3.3.1, we observed seven sample cases in Groups X and P that utilise identified PETs at a considerable level. The identified PET transaction proportion results reveal further insights into the obscuring strategies employed by illegal Bitcoin users, as shown in Figure 3.8. The results suggest that illegal Bitcoin users typically employ only one type of PET to obscure the stolen Bitcoins. The proportion of transactions involving identified PETs in the sample cases' results is not substantially different from the results of the control groups except for the sample cases in Groups X and P. Interestingly, case TC14's Dirty-First^{AP}, FIFO^{AP},

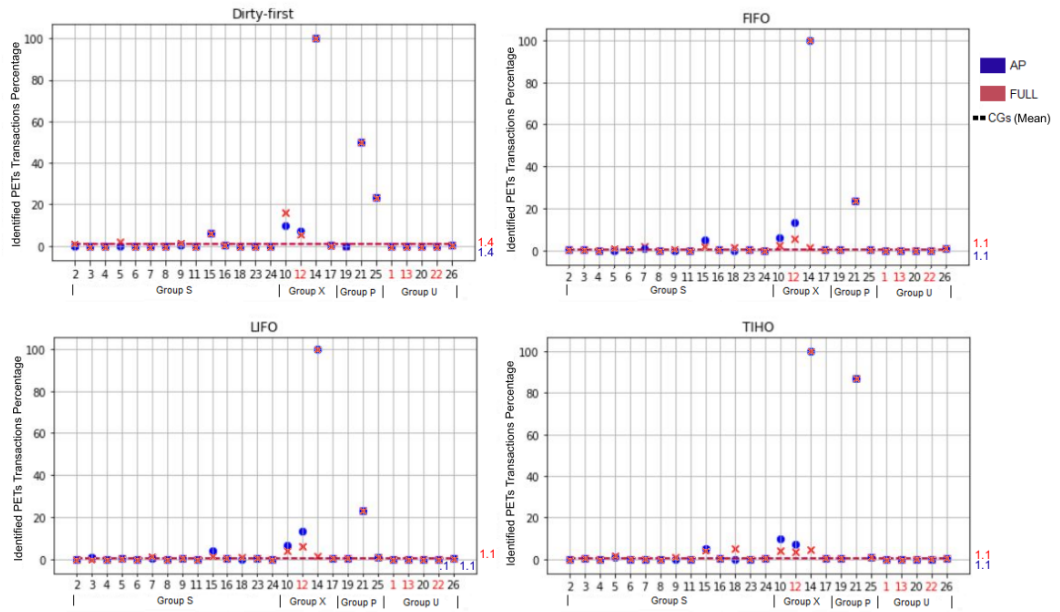


Figure 3.8: Percentage of transactions reaching identified PETs (H2)

LIFO^{AP}, and TIHO^{AP} results show 100% of transactions involving identified PETs, but fewer than 10% of transactions for the FIFO^{Full}, LIFO^{Full}, and TIHO^{Full} results. These results indicate that the illegal Bitcoin users of this sample case sent the stolen Bitcoins to a PET service in every transaction starting from the first transaction. Additionally, case TC15, which show an insignificant number of Bitcoins reaching PETs in Figure 3.4(a), have almost 10% of transactions involving identified PETs for all four taint analysis strategies' results. These results may be an indication of the illegal Bitcoin users changing their stolen Bitcoins' spending strategies.

The results in Figure 3.9 are shown as the proportion of tainted transactions that are classified as a potential PET transaction by the classification method described in Section 3.1.2.4. It is also worth noting that the potential PET transaction results do not include the identified PET transactions or transactions with an identified service or mixer address.

The Dirty-First strategy shows a significantly large number of potential PET transactions from 10% to 80% of transactions, while the FIFO, LIFO, and TIHO strategies show a small number of potential PET transactions of fewer than 10% for most sample cases in both of the TC^{Full} and TC^{AP} results, including those that employ identified PETs in Groups X and P, which are not much different from the CG^{Full} and CG^{AP} results. Additionally, the TC^{Full} results generally show either an

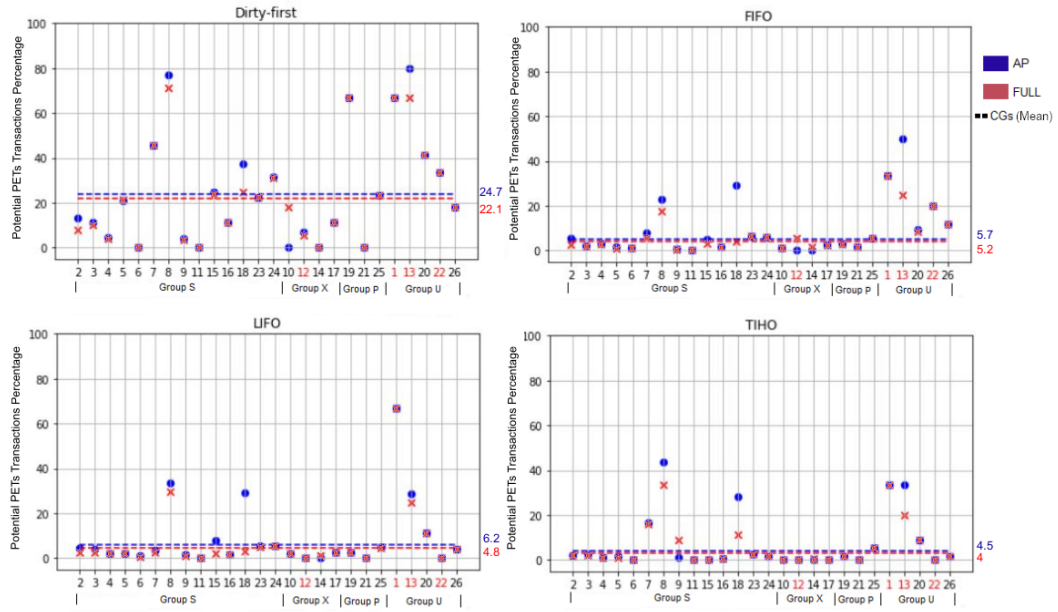


Figure 3.9: Percentage of potential PET transactions (H2)

equal or smaller proportion of potential PET transactions, compared to the TC^{AP} results for the majority of the sample cases. Since the potential PET transaction classification method relies on the presence of completely clean Bitcoins in transaction inputs, these results imply that clean Bitcoin mixing occurs mainly when stolen Bitcoins reach either services or PETs but not in subsequent transactions.

As we discovered that the sample cases in Group S show the stolen Bitcoins reaching service addresses directly in the Dirty-First results, the potential PET transactions in the results of these sample cases are more likely to be transactions involving unidentified cryptocurrency services rather than PETs. Meanwhile, there is a high possibility that the potential PET transactions identified in the results of Groups X and P' cases are transactions involving unidentified PETs. Intriguingly, the results of Group U cases show a considerably high proportion of potential PET transactions for all four taint analysis strategies despite showing a remarkably small number of identified addresses and transactions, including the sample cases with a small total number of transactions like cases TC1 and TC13. The results of sample cases in Group U suggest that the lack of Bitcoin spending in the sample cases of this group may not be due to the limitation of a small tracking timeframe, but rather the incompleteness of our address and transaction profile data.

Therefore, our H2 hypothesis that there would be a significant number of PET

transactions in Bitcoin theft cases is not supported by the results. Nevertheless, the identified PETs profiling and the potential PETs classification method reveal insights into the obscuring strategy, or lack thereof, employed by the illegal Bitcoin users. The PET address and transaction' profiling can be expanded further to assist the taint analysis algorithm in detecting and adapting its tracking process for PET transactions. It would also be possible to expand PETs' profile data by identifying common patterns such as transaction shape that may indicate when tainted Bitcoins reach transactions with similarity as identified PET transactions.

3.3.4 Reused Address (H3) and Fresh Address (H4) Results

The results of reused address (defined in Section 3.1.4.3) and fresh address (defined in Section 3.1.4.4) metrics in Figure 3.10 are shown as the proportion of addresses that are either old and reused, fresh but reused later, or fresh and never reused. It should be noted that we exclude addresses identified as belonging to either a service or PET in reused and fresh address results.

The results of the reused and fresh address metrics reveal a consistent pattern for most sample cases. The Dirty-First strategy generally shows a higher number of fresh and not reused addresses, compared to the FIFO, LIFO, and TIHO strategies for the TC^{AP} results. The presence of reused addresses in the Dirty-First strategy's results may indicate that the illegal Bitcoin users tend to avoid reusing addresses but not always since the Dirty-First^{AP} results still show a substantial reused address proportion. Meanwhile, the FIFO, LIFO, and TIHO strategies generally show an increasing number of reused addresses for most of the TC^{AP} results. Considering that the address profile data is still likely to contain only a fraction of service and mixer addresses in existence, the increase in the number of reused addresses for these three strategies in the TC^{AP} results can be from addresses belonging to unidentified services, PETs, or other Bitcoin users that receive the stolen Bitcoins.

The TC^{Full} results generally show an increase in the number of reused addresses and a decrease in the number of fresh addresses, compared to the TC^{AP} results for most sample cases. The increase in the number of reused addresses in the TC^{Full} results supports our hypothesis in Section 3.1.1.1 that cryptocurrency services have lower privacy requirements to perform privacy techniques.



Figure 3.10: Proportion of reused and fresh addresses (H3 and H4)

Intriguingly, the Dirty-First^{AP} results of cases TC5, TC10, and TC20 show a more significant proportion of reused addresses, compared to the control groups. These results reveal intriguing insights that even illegal Bitcoin users may not completely avoid reusing their addresses, which is one of the most common privacy techniques that any Bitcoin user can costlessly perform without requiring any PET. It would be possible to analyse the illegal Bitcoin users' transaction activity outside of the tainted transactions with these previously used addresses, which can ultimately help unveil their personal information. The results of the sample cases generally show a lower number of reused addresses and a higher number of fresh

addresses, compared to the control groups. Therefore, the results of reused and fresh addresses support our H3 and H4 hypotheses.

3.3.5 Number of Addresses per Transaction (H5) Results

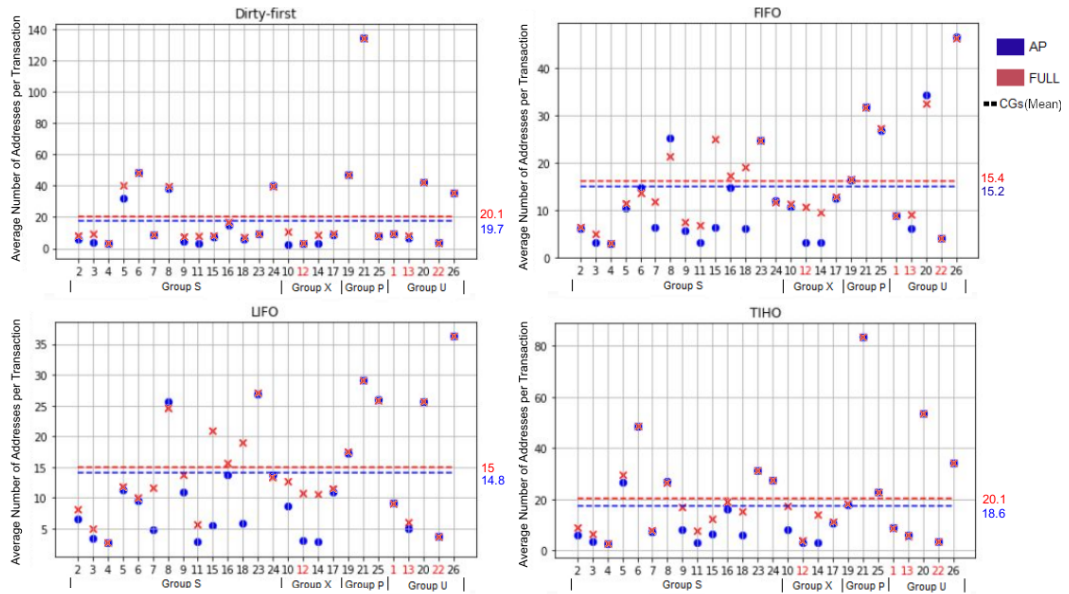


Figure 3.11: Average number of addresses per transaction (H5)

The results of the number of addresses per transaction metric as defined in Section 3.1.4.5 are shown as the weighted average from all tainted transactions, as shown in Figure 3.11.

The results of the number of addresses per transaction reveal an unexpected pattern where the majority of the TC^{AP} results show a small average number of addresses per transaction (lower than 20) for the Dirty-First strategy. There are a few exceptions, like cases TC21 and TC24, which show a much higher average number of addresses per transaction (around 140 and 40 addresses per transaction). The results suggest that the illegal Bitcoin users in most sample cases prefer to send the stolen Bitcoins in small transactions, possibly to avoid making their transactions distinct from other transactions. The small number of addresses per transaction for the sample cases' Dirty-First^{AP} results suggests that the transactions that are likely to be performed by the illegal Bitcoin users do not typically have a large number of addresses because of the Bitcoin privacy technique mentioned in Section 3.1.4.5. There is also no clear difference between each group that indicates either a common

or unique pattern.

The majority of the sample cases show an increase in the average number of addresses per transaction for the TC^{Full} results. This pattern indicates that the transactions that occurred after the stolen Bitcoins reached service or mixer addresses generally have a higher number of addresses, compared to the transactions in the TC^{AP} results. As such, the results illustrate that the transactions that occur after the stolen Bitcoins reach a service or PET address are substantially different from the transactions in TC^{AP} results for all four taint analysis strategies.

The TC^{AP} results show an increasing average number of addresses per transaction for the FIFO, LIFO, and TIHO strategies, compared to the Dirty-First strategy. Based on our previously mentioned hypothesis that the presence of clean Bitcoins indicates the possibility of PETs' usage, the increase in the number of addresses per transaction in TC^{AP} results for the three strategies may be due to the transactions that involve unidentified cryptocurrency services or PETs. However, the CG^{AP} results show an overall higher number of addresses per transaction, compared to most sample cases in the TC^{AP} results for all four taint analysis strategies.

The results of the number of addresses per transaction metric do not support our H5 hypothesis that the majority of transactions in Bitcoin theft cases would be large transactions. Nevertheless, the number of addresses per transaction metric shows potential for revealing a change in transaction behaviour between the Dirty-First strategies and the FIFO, LIFO, and TIHO strategies, which can be an indicator for changes in the stolen Bitcoins' ownership.

3.3.6 Transaction Fee (H6) Results

The results of the transaction fee metric as defined in Section 3.1.4.6 are shown as the weighted average of the difference between the transaction fee size ratio (Satoshis¹⁰ per byte) in tainted transactions and all of the transactions on the same day, as shown in Figure 3.12.

The transaction fee size ratio in the TC^{AP} results shows a considerably diverse pattern for the Dirty-First strategy, where the sample cases show a transaction fee size ratio of either lower than, equal to, or higher than the daily average. There

¹⁰Satoshis is the smallest unit of Bitcoin, 1 BTC is equal to 100,000,000 Satoshis.

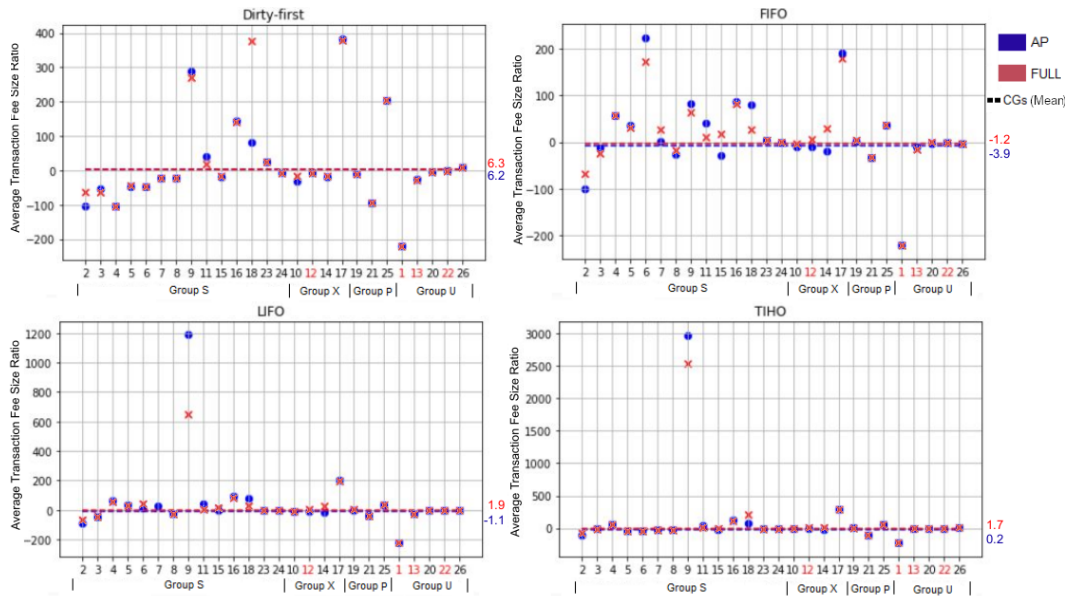


Figure 3.12: Average difference between transaction fee size ratio to daily average (H6)

are four sample cases, TC1, TC2, TC4, and TC21, which have an average of 100 transaction fee size ratio lower than the daily average. Meanwhile, five sample cases, namely TC9, TC16, TC17, TC18, and TC25, show a transaction fee size ratio of 100 Sat per byte higher than the daily average. The varied results in the Dirty-First^{AP} results indicate that there seems to be no standard practice that illegal Bitcoin users employ for transaction fee payment, and each user typically pays according to their preference.

Meanwhile, the FIFO, LIFO, and TIHO strategies in the TC^{AP} results show a substantial change in the transaction fee size ratio, compared to the Dirty-First strategy. The FIFO^{AP} results generally show an increase in the transaction fee size ratio and exceed the daily average for many sample cases. Intriguingly, the LIFO^{AP} and TIHO^{AP} results are significantly different from the FIFO^{AP} results, as they seem to exhibit a transaction fee size ratio remarkably close to the daily average and the CG^{AP} results for most sample cases. These results may indicate that the results of the two strategies contain a large number of transactions performed by a similar type of entity, which we assume can be either unidentified services or PETs. The reasoning for this assumption is that services and mixers (as mentioned in Section 3.1.1.2) tend to combine their Bitcoins into transaction outputs with a large

number of Bitcoins and transfer them to their users in a “peeling chain¹¹”. Hence, the TIHO strategy that prioritises distributing tainted Bitcoin to the output with the highest value would keep following change outputs that belong to the services. Change outputs are also often the last outputs in the transactions as many wallet clients create transactions by putting change outputs after spending outputs by default [11].

The TC^{Full} results are considerably different from the TC^{AP} results, especially for the Dirty-First and FIFO strategies, where the transaction fee size ratio are closer to the daily average for most sample cases. This pattern supports the assumption in the previous paragraph that the services and PETs typically pay transaction fees close to the daily average. Although similar to the results of the number of the addresses per transaction (H5), there seems to be no obvious pattern in each group that can differentiate the sample cases in the same group from the others.

The transaction fee metric results do not support our H6 hypothesis that the transactions in Bitcoin theft cases would have a high transaction fee in this experiment. Nevertheless, the transaction fee metric results illustrate a clear change in transaction fee behaviour, especially between the Dirty-First and the FIFO, LIFO, and TIHO strategies. The changes in transaction fee behaviour after clean Bitcoins mixing are likely an indication that the transactions with clean Bitcoins are performed by different entities, which support our hypothesis of clean Bitcoin mixing.

3.3.7 Results Summary and Discussion

As shown in Figure 3.13, two out of six evaluation metrics’ hypotheses are supported by the experiment’s results, namely reused address (H3) and fresh address (H4) metric. The four other metrics’ results do not support their hypothesis but illustrate a significant change in transaction behaviour that may also indicate a change in the ownership of the stolen Bitcoins. We summarise the key findings of each metric as follows.

Theft cases do not have higher transaction frequency (contradicting H1)

The results of the theft cases do not support the hypothesis of H1) but the metric

¹¹Bitcoin peeling or peeling chain is the act of continuously spending a small number of Bitcoins from the same large transaction output in a continuous chain of transactions [121]

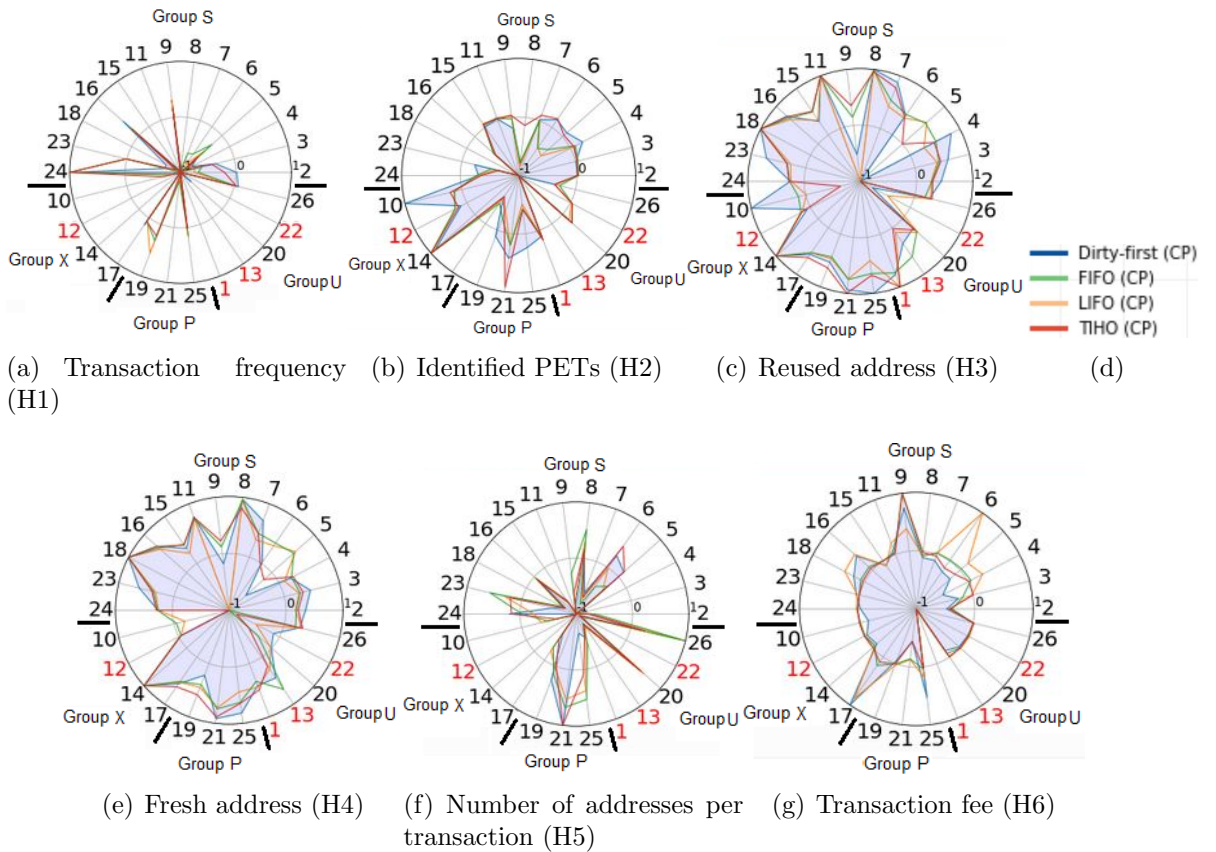


Figure 3.13: Summary of TC^{AP} results for each evaluation metric.

The scale is based on the difference between the TC^{AP} results and the CG^{AP} results of the same taint analysis strategy, where 0 is the CG^{AP} value, -1 (inner) is the value of TC^{AP} that contradicts the hypothesis the most and further than the CG^{AP} , while 1 (outer) is the value of TC^{AP} that supports the hypothesis the most and further than the CG^{AP} .

manages to reveal the difference in illegal Bitcoin users' spending strategy, whereas most spent their stolen Bitcoins in a small number of transactions, while there are also those who rapidly distribute their stolen Bitcoins in a large number of transactions (as high as 1,000 transactions per day).

Theft cases do not have a higher number of PET transactions (contradicting H2) The results of the theft cases do not support the hypothesis of H2 but suggest that most of the illegal Bitcoin users we observed do not utilise PETs to obscure their stolen Bitcoins before spending them, and those that utilise PETs tend to employ only one type of PETs.

Theft cases have a lower number of reused addresses (supporting H3) The results of the theft cases support the hypothesis of H3 and illustrate shifts in the behaviour when the tainted Bitcoins were exchanged with identified services.

Theft cases have a higher number of fresh addresses (supporting H4)

Similar to the H3, the results of the theft cases support the hypothesis of H4 and indicate that most of the illegal Bitcoin users tend to use fresh addresses that are typically not reused afterwards.

Theft cases have transactions with a lower number of addresses per transaction (contradicting H5)

The results of the theft cases support the hypothesis of H5 by illustrating that the illegal Bitcoin users perform mostly small transactions. However, the results indicate a significant increase after the stolen Bitcoins reach identified service addresses.

Theft cases have transactions with diverse transaction fees (contradicting H6)

The results of the theft cases support the hypothesis of H6 by revealing diverse transaction fee spending behaviours between the sample cases. However, the results illustrate a general shift toward the day average when including transactions after reaching identified service addresses.

While some of the evaluation metrics hypotheses are not supported by the results, the majority of the evaluation metrics show distinct results between the sample cases and the control groups, which suggest that the evaluation metrics might be useful for further contextualising Bitcoin tracking solutions. We summarise the significant key findings that the evaluation metric illustrates as follows:

Theft cases have distinctly different transaction and address behaviour from the control groups

The results of evaluation metrics illustrate that the transactions and addresses in both TC^{AP} and TC^{Full} are significantly different from the CG^{AP} and CG^{Full} , respectively, including the metrics that contradict our hypothesis. The only remarkable exception is the PETs usage metric, whereas most theft cases show minimal PET transactions similar to the control groups. Nonetheless, this is likely to be because of the incompleteness of the current address profile data, which makes the tracking process unable to identify transactions that involve a PET service.

Theft cases' transaction and address behaviours show significant change after reaching identified service addresses The results of evaluation metrics (as shown in Figures 3.7, 3.9, 3.10 3.11, 3.12) show that the transactions in TC^{AP} and TC^{Full} typically have clearly different transaction behaviours except for the Dirty-First strategy¹². The grouping of sample cases based on their Bitcoins spending does not seem to illustrate a common pattern of those in the same group or difference from the other groups outside of the Bitcoin spending results. However, this may be due to the limitation of using unrelated theft and ransomware attack cases.

The introduction of context-based taint analysis strategies (Section 3.1.3) and compilation of multiple taint analysis strategies reveal insights into transaction behaviour patterns that would be elusive for tracking results of an individual taint analysis strategy. The Dirty-First strategy illustrates multiple occasions where fully tainted stolen Bitcoins managed to directly reach addresses that are likely to belong to a cryptocurrency service without relying on PETs. The strategy also shows the capability to reveal various behavioural changes when compared to the FIFO, LIFO, and TIHO results that may indicate the change of stolen Bitcoins' ownership.

Meanwhile, the TIHO strategy's results typically show the lowest number of stolen Bitcoins reaching service and PET addresses, compared to the FIFO and LIFO strategies. Additionally, the TIHO results for case TC21 that show usage of the CoinJoin method do not seem to exhibit a higher service address reaching or a substantial difference, compared to the other two strategies. Therefore, the TIHO strategy does not illustrate a clear benefit of providing more accurate tracking over the FIFO and LIFO strategies in this experiment.

Instead of continuing to track tainted Bitcoins after they reach a service address, the tracking process should adapt its operation to track the targeted users' activity outside of the Bitcoin system. For example, when targeted users exchange tainted Bitcoins for other cryptocurrency coins via an exchange service, then the tracking process should attempt to identify the other cryptocurrency coins that the targeted users receive using information obtained from the exchange service involved. Subsequently, the tracking process can continue tracking using the blockchain data of the exchanged cryptocurrency coins [204].

¹²This is because of its tracking methodology of stopping tracking after clean Bitcoins combine with tainted Bitcoins.

Similar to Bitcoins that reach a service address, the tracking process should adapt its algorithm for tracking obscured Bitcoins like zero-taint Bitcoins with specialised tracking strategies. There are two strategies proposed by previous studies to track zero-taint Bitcoins as far as we know. The first strategy operates by matching every transaction in the blockchain that occurs during the mixing period with a set of criteria and filtering the potential transaction outputs that may contain the targeted mixed Bitcoins [89]. The second strategy involves a method called *Address taint analysis*, which is a variant of taint analysis designed to identify the mixer service address network and produce a transaction network that may be involved with the mixing operation. Then, the outputs of the targeted Bitcoins can be pinpointed using a set of criteria similar to the tracking strategy above [172].

3.3.8 Reflection on Alternative Attempts

In earlier experiments, we incorporated a different address profiling method into the taint analysis by using transaction traffic to classify service addresses with the assumption that high transaction traffic often implies the address is a point of central exchange for many users, similar to how businesses operate in the real world. The classification process operates by comparing the number of transactions of every address that appears in the blockchain within six months of the theft (three months before and after the theft transaction) and classifies addresses with a significantly higher number of total transactions as service addresses.

We chose the classification of service addresses to be at the top percentile of all addresses in the time limit at 99th percentile (higher than 18 transactions) with the reasoning that choosing the lower percentile would mean a higher chance to include services that employ transaction obscuring techniques, such as laundering service addresses; moreover, it would also increase the chance of false classification of normal addresses.

We reconsidered the approach as the transaction traffic profiling method can only detect a small subset of service addresses and can not detect service addresses that have small transaction activity during the targeted time. Additionally, it is possible that there are still many individuals who reuse their addresses, as pointed out in a previous study [79]. The current profiling method makes use of ground-truth data

in the form of seed addresses (i.e., addresses known to belong to services or publicly displayed on the service’s website) to identify clusters of service addresses.

Earlier experiments also made use of only the potential PET transaction detection method (see Section 3.1.2.4) to detect PET transactions. While the method can also detect PET transactions that are by the identified PET service in this study, the potential PET transaction method on its own can not determine the PET service involved, which is a piece of crucial information for cryptocurrency forensic analysis to track obscured Bitcoins. Hence, we introduce various detection methods to identify transactions that are potentially related to known PETs based on their mixing mechanism and transaction characteristics.

3.3.9 Limitations

Although context-based tracking demonstrates potential benefit in reducing a relatively large number of unessential transactions, there are limitations of our approach and experiment that we discuss below.

As context-based tracking is designed with the assumption that service addresses are exit points of targeted Bitcoins, the approach has a limitation where service address profiling data may contain false positive results. This limitation can stem from users setting up false service addresses to trick the address profiling or disreputable services sharing their transactions with other addresses via the CoinJoin method. It is possible to mitigate this limitation by utilising more thorough address profile data gathering and verification methods, which will also ensure that context-based tracking is most effective and accurate.

The address and transaction profiling methodology and data we utilise in this work are likely incomplete, and many addresses that belong to cryptocurrency services and PETs remain unidentified (false negative). This limitation can be improved with more address and transaction profile data. One example of data sources that can help strengthen context-based tracking is blockchain analysis companies, which typically possess more extensive profile databases compared to the public sources we utilise in this work. Expanded address profile data will also allow us to analyse the Bitcoin spending of each sample theft case with more accuracy.

The hypothesis for the evaluation metrics in this experiment is based on the

assumption that the illegal sample cases would generally follow the privacy practice to increase the tracking difficulty. However, as illustrated in the results that the sample cases typically do not share common behaviours as we hypothesised. This issue is likely due to the use of unrelated theft cases as samples and the limited number of sample cases we study. A higher number of sample cases may help illustrate more common behaviours among theft cases.

Additionally, the current work lacks the data of the sample theft cases that can verify the actual movement of the stolen Bitcoins. As such, we could not thoroughly analyse the theft cases in this work. While it is likely that some of the theft cases we examined are already solved by law enforcement, this information typically is not publicly available because of the nature of the information itself. This information will allow us to compare and evaluate each taint analysis strategy and the profiling data.

3.4 Conclusion and Future Work

In an attempt to precisely track Bitcoins and other similar cryptocurrency coins, tracing the targeted Bitcoins to the end of the blockchain would only show which pseudonymous addresses are the last holders of the targeted Bitcoins chosen by the tracking process. The methodology we presented in this chapter proposes to make the tracking process adaptive to the change in Bitcoin ownership with address profiling. The results of our experiment involving the analysis of 26 historical Bitcoin theft cases compared to a set of controls show benefits in incorporating address profiling to taint analysis process and confirm the relevance of the set of metrics we defined. One of the context-based strategies we introduced, Dirty-First, allows us to observe the spending and obscuring strategies of the stolen Bitcoins used by illegal Bitcoin users. However, the TIHO strategy does not show distinct outcomes, compared to existing taint analysis strategies.

The integration of address profiling into the taint analysis process demonstrate that it can reduce a substantial number of unessential transactions that also affect the overall transaction behaviour in the analysis results. The integration can become more effective by expanding the address profile data and classifying other types of

entities. Additionally, the tracking process can use transaction profiling to recognise PET transactions and adapt its tracking operation once future work can thoroughly verify that the transaction classifications do not produce false results.

The evaluation metrics can be expanded to include other behaviours that we have not yet investigated in this work. For example, future work could implement other wallet fingerprinting methodologies discussed in Section 2.5.1 like address type as metrics to evaluate changes in Bitcoin ownership.

Just as the privacy in Bitcoin and other cryptocurrencies continue to evolve to protect its users from tracking attempts, so too must the tracking methodology. Our context-based tracking methodology presents the necessary improvements for cryptocurrency tracking effort and provides the next step for future cyber forensics research to assist in understanding practices within cryptocurrencies and combating cybercrimes.

Chapter 4

Zero-taint Bitcoin Tracking

A Bitcoin *mixer service* (also commonly known as *tumbler* or *laundering service*) is a PET in the form of cryptocurrency service that allows users to “anonymise” their Bitcoins by eliminating any possible connection between their original deposited Bitcoins and the *mixed* Bitcoins that they withdraw later from the service [13, 84]. This mixing process can make the tracking of Bitcoin movements between addresses challenging, such as when using techniques like taint analysis [127]. Mixer services are also frequently used as one of the core components in transaction obscuring for illicit activities, such as theft, ransomware, and dark market trade [161, 177].

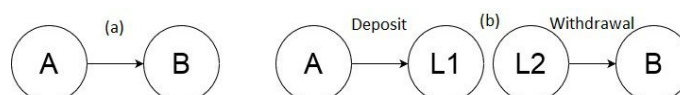


Figure 4.1: Bitcoin transfer using normal transaction and using mixer service
The figure depicts the transfer of Bitcoins from address A to address B using a normal Bitcoin transaction on the left side (a), and using a mixer service (involving addresses L1 and L2) to send Bitcoins to address B on the right side (b).

As shown in Figure 4.1, address *A* would send Bitcoins directly to address *B* in a normal Bitcoin transaction. However, this interaction establishes a connection between the two addresses in the blockchain, allowing anyone to observe the movement of Bitcoins [133]. Mixer services attempt to prevent this traceability by serving as an intermediary between the two addresses where address *A* deposits Bitcoins to a mixer service address (named the receiver address) for mixing purposes. Next, the mixer service uses another address(es) (named the delivery address) to deliver completely unrelated (zero-taint) Bitcoins to address *B* in withdrawn transactions.

As a result, the interaction between address A and B is obscured in the blockchain, as there is no direct connection or transaction between the two end-point addresses. Furthermore, simple transaction tracking methods are incapable of tracking the actual exchange of Bitcoins between the two addresses. One method used to track the mixed Bitcoins is to calculate every possible combination on every transaction within the mixing time for the potential withdrawn transaction outputs [89], which typically produce a large number of unrelated transactions.

Few studies have investigated and reverse-engineered mixer services to discover their mixing pattern [47, 127, 181]. We are aware of only one study that proposed a tracking method for mixed Bitcoins, which adapted the aforementioned approach and evaluated their method on a single mixer service [89]. In particular, we are not aware of any proposed tracking method to overcome the transaction obscuring feature of mixer services. Hence, in this chapter, we introduce a novel tracking method called *address taint analysis* that focuses on tainting at the address level, whereas previous taint analysis approaches have focused on tainting at the transaction level. We investigate this method, both on its own and in combination with other tracking methods, such as address clustering and *backward tainting*. We also introduce a set of filtering criteria that we use in combination with cryptocurrency service address profiling (discussed in Chapter 3) in an attempt to reduce the number of false positive results. We evaluate our solutions with verifiable mixing transactions of nine centralised mixer services used in previous reverse-engineering studies.

The remainder of the chapter is structured as follows. We define our new methods and filtering criteria in Section 4.1. Using the sample cases presented in Section 4.2, we evaluate the results of these methods and discuss the results in Section 4.3. In Section 4.4, we conclude and discuss improvements we envision.

4.1 Methodology

In this section, we describe the address taint analysis method and its combination with other tracking methods (address clustering in Section 4.1.1 and backward tainting in Section 4.1.2). Subsequently, we discuss the filtering criteria we developed and the rationale behind them and the service address profiling of cryptocurrency

services in Section 4.1.3.

To evaluate the effectiveness of our tracking methods and filtering criteria, we compare the number of tainted transaction outputs of each method to the baseline of all outputs occurring in the same time frame. Our definition of the baseline is based on work from a previous study [89].

Baseline

All outputs of every transaction recorded in the blockchain within the tainting time frame of a given sample case.

4.1.1 Address Taint Analysis

The majority of centralised mixer services usually utilise a group of central addresses in order to combine and mix deposited Bitcoins from their users [47, 127, 181]. We assume that the receiver and delivery addresses within the centralised mixer services are both likely to interact with the central addresses at some point in time.

Our taint analysis method, *address taint analysis*, operates at the address connection level, where any address that receives Bitcoins from tainted addresses will be considered as a tainted address, including every Bitcoin it possesses at any point in time. (the algorithm of the address taint analysis method is shown at Algorithm 9) Existing taint analysis methods operate at the transaction level, where the tainted Bitcoins of a received address do not affect other Bitcoins belonging to that address unless they are used together in the same transactions, as shown in Figure 4.2.

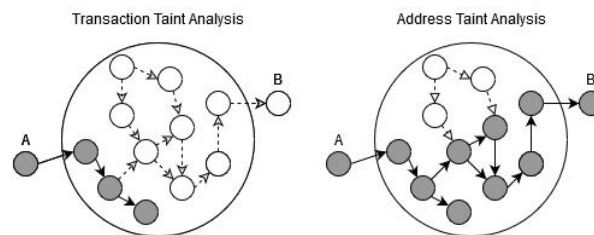


Figure 4.2: Transaction taint analysis and address taint analysis

The figure depicts the difference between the transaction taint analysis and address taint analysis methods on an example mixing case that shows the deposited transaction from address *A* and the withdrawn transaction to address *B*. Grey arrows and circles indicate a transaction and address that involves tainted Bitcoins, while white ones indicate they are clean.

The assumption for address taint analysis is that any transaction and address that can be connected to the receiver addresses at any point in time, whether directly

or indirectly, may be related to the mixer service in some way. Therefore, the objective of address taint analysis is not only to track the mixed Bitcoins but also to map the network of address clusters and their transactions that may involve the mixer service operation, as in Bitcoin network analysis [111]. Hence, address taint analysis tracking should be able to discover a relationship between the deposited and withdrawn transactions that the transaction taint analysis is unable to accomplish, as shown in Figure 4.2.

We describe three methods below for using address taint analysis (one further method is described in Section 4.1.2). The first method uses only address taint analysis. For the second and third methods, we investigate the potential of incorporating address clustering heuristics into the address taint analysis in order to improve the tracking results. As deanonymisation is not our primary objective, we utilise the address clustering heuristic to assist the address taint analysis algorithm for capturing relationships between addresses that are outside the scope of address taint analysis, which regard only the Bitcoin movement (address A sends Bitcoins to address B).

Method 1

Address taint analysis only.

The operation of address taint analysis used in this chapter is conceptually similar to the Poison strategy of taint analysis [128] as the Bitcoins are considered completely tainted, regardless of the number of tainted Bitcoins involved but goes further by affecting every Bitcoin possessed by the address throughout time. Since our main priority is to discover the connection between the deposited and withdrawn Bitcoins, other transaction taint analysis strategies, which generally emphasise the distribution of taint value proportions, would not provide further information for this purpose.

As centralised mixer services typically perform the mixing operation continuously, it is possible for the service to deliver Bitcoins that are already mixed prior to the time of the deposited transactions. As such, address taint analysis will also need to taint from the time period before the deposited transactions occurred. To put it simply, address taint analysis will taint all Bitcoins that the tainted addresses send, both before and after the deposited transactions time.

Method 2

Address taint analysis with multi-input address clustering heuristic.

We use the multi-input address clustering heuristic (see Algorithm 14) coupled with address taint analysis to taint any address that shares inputs with the tainted addresses. We use the same hypothesis as the original multi-input address clustering heuristic for our adaptation – any address that shares input in the same transaction with any tainted address is also likely to be one of the mixer service addresses and will be classified as a tainted address.

Method 3

Address taint analysis with multi-input and multi-output address clustering heuristic.

As an augmentation to Method 2, here we also incorporate the multi-output address clustering heuristic (see Algorithm 14) with the assumption that in the case of the mixing operation, the central addresses would often distribute the mixed Bitcoins to other mixer addresses first before delivering them to the users. Consequently, we expect that the multi-output address clustering heuristic should improve the chance of tracking such scenarios, even if the delivery addresses of the mixer service never send mixed Bitcoins to one another or share input in the transaction.

4.1.2 Backward Address Taint Analysis

The address taint analysis method operates with the assumption that the deposited and withdrawn mixer addresses may have a connection with each other via the central addresses, the analysis will not connect deposited inputs to the withdrawn outputs if there is no connection between the addresses involved, as shown in Figure 4.3.

In such situations, address taint analysis from the deposited address can not reach the withdrawal address. However, the knowledge of pre-existing withdrawal addresses could be used to identify the targeted withdrawal address. The search would consist of tainting backward from this known withdrawal address and then forward towards potential withdrawal addresses.

Therefore, we introduce another method for this scenario by applying backward tainting to the address taint analysis to create another tracking method called *Back-*

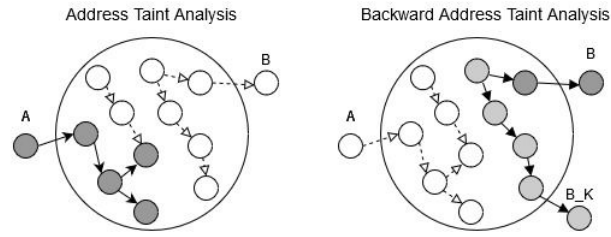


Figure 4.3: Address taint analysis and backward address taint analysis
 The figure depicts a centralised mixer service with two separate central groups tainted without and with backward address taint analysis. Notice the lack of any interaction between the address A and B groups. B_K represents the withdrawn transaction output(s) from a known case used for backward address taint analysis in Method 4. Lighter grey colour circles represent taint analysis results of backward address taint analysis, and darker grey colour circles represent taint analysis result of address taint analysis performed after backward address taint analysis.

ward Address Taint Analysis. This method operates by tainting any address that sends Bitcoins to a tainted address. Rather than attempting to discover the connection between the mixed Bitcoins, the purpose of this method is to investigate whether it's possible to discover the address clusters used for withdrawn transactions. The idea would be that these addresses could subsequently be used to find the targeted withdrawn transaction outputs. Thus, this method operates in two steps, as described in the example below.

Method 4

Perform Method 3 on the results of backward address taint analysis on the known pre-existing withdrawn transactions from the same mixer service.

Using the example from Figure 4.3, Method 4 starts by performing the backward address taint analysis variation from the withdrawn transactions of a case from the same mixer service (B_K) for three days to trace the mixed Bitcoins back to the central address clusters. Next, we use the results of the backward address taint analysis to perform address taint analysis at the time of the deposited transactions of the targeted sample case (A).

4.1.3 Filtering Criteria and Address Profiling

To further reduce the number of false positive results, we define five filtering criteria based on the information of the withdrawn transactions obtained from reverse-engineering experiments used in previous studies [47, 127, 181] and the service ad-

dress profile data we utilise to filter transactions that are unlikely to be related to the targeted mixing operation. The algorithm that we use to apply the filtering criteria is shown in Algorithm B.13.

The criteria can be applied for mixed Bitcoins in general with appropriate calibration. The calibration of the criteria parameters can also be specified to be stricter to reduce the false positive results even further, but this can increase the risk of missing the target. The parameters used in this experiment are obtained from observing the sample cases provided by the studies mentioned above. We set the parameters conservatively to reduce the risk of losing the targeted withdrawn transactions for this experiment. In establishing the filtering criteria for our investigation, we had the advantage of knowing the target withdrawn transaction outputs that we were searching for. For future studies, we plan to investigate the criteria on data with unknown target values.

Criterion 1 (Value of Withdrawn Bitcoins)

The transaction output value of the targeted withdrawn transaction outputs can not be higher than the deposited input value minus the mixing fee.

As mixer services typically subtract a specific mixer service fee¹ from the initial deposited Bitcoins, the amount of the withdrawn Bitcoins would be lower than the original deposited amount. Depending on the mixer service, the mixing fee can vary in a specific range, such as between 1-2% of the deposited Bitcoins. For this experiment, we use a minimum mixing fee for this criterion to lessen the risk of missing positive results.

This criterion does have at least one limitation, as it may be possible for the mixer services to combine the withdrawal of multiple deposited Bitcoins, which can make the withdrawn transaction outputs larger than the deposited input.

Criterion 2 (Withdrawn Transaction's Shape)

The number of transaction inputs and outputs of the targeted withdrawn transactions must be in the same pattern as the other withdrawn transactions by the same mixer service.

Reverse-engineering examples used in the literature [47, 127, 181] show that the

¹Note that mixer service fee is different from Bitcoin transaction fee, which is mentioned in Criterion 5.

mixer services usually perform withdrawn transactions in a specific pattern. For example, one of the most common shapes of withdrawn transactions is in the form of a one-to-two addresses transaction where a single transaction output is sent to two addresses, one belonging to the user and the other to the mixer service. The number of transaction inputs and outputs of the targeted withdrawn transactions must be in the same pattern as the other withdrawn transactions by the same mixer service.

A limitation of this criterion is that it is also possible for the mixer service to randomise the shape pattern or have an exception scenario (e.g., the withdrawn Bitcoins are in large value so that the service needs to combine other inputs in a withdrawn transaction) that can make the targeted withdrawn transaction different from the common pattern.

Criterion 3 (Withdrawn Transaction Chain’s Shape)

If the mixing algorithm has a continuous withdrawn transaction chain pattern (e.g., peeling chain shown in Figure 4.4), either the transaction before or after the targeted withdrawn transactions must have the same number of transaction inputs and outputs as the common pattern.

Following from Criterion 2, the reverse-engineering results of the mixing sample cases indicate that multiple mixer services usually perform the withdrawn transactions in a continuous peeling chain, where a single transaction input with a large amount of Bitcoins is continuously peeled into two transaction outputs with one typically much smaller than the other [47].

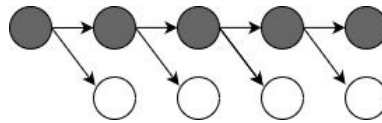


Figure 4.4: Example of a peeling chain

The figure depicts the peeling chain of a transaction chain that is commonly used by centralised mixer services. Black circles represent delivery addresses that belong to the mixer service and white circles represent users’ addresses that receive withdrawn Bitcoins in a transaction output.

As such, either the previous or next transaction of the targeted withdrawn transactions must also follow a similar pattern, accounting for the possibility that the targeted withdrawn transactions can be at the start or the end of the withdrawn

transaction chain.

Similar to a limitation for Criterion 2, the mixer service can randomise the transaction chain shape or simply does not have one, which can increase the risk of missing the targeted transaction outputs or make this criterion inapplicable.

Criterion 4 (Reused Input Address)

The input address in the targeted withdrawn transactions is not used as transaction input more than once in its lifetime.

Our analysis of the verifiable mixing transactions from the previous studies [47, 127, 181] shows that the majority of the centralised mixer services never reuse their delivery addresses before and after the withdrawn transaction. Therefore, we can utilise this information as a criterion to exclude any transaction with input addresses that have been reused at any point in time.

A limitation of this criterion is that although generally mixer services do avoid reusing the same address more than once, which is one of the most common Bitcoin privacy practices [65], it is possible for a mixer service to disregard this practice.

Criterion 5 (Withdrawn Transaction Fee)

The transaction fee value of the targeted withdrawn transactions must be the same as in other withdrawn transactions in the same time period.

From our own analysis of the verifiable mixing transactions from the previous studies, we also detect a specific pattern in the transaction fee values of the withdrawn transactions. In particular, the transaction fee values are of the same specific amount, such as 0.0005 BTC or 0.0001 BTC, even with a different transaction fee per byte ratio and at a different time and day. This suggests that mixer services generally do not automatically adjust the transaction fee setting in real-time but in a specific amount of time. As such, if the transaction fee always remains constant for the other withdrawn transactions at a similar time by the same mixer service, we can use the transaction fee as a criterion to exclude unrelated transactions.

Similar to the other criteria, it is possible for a mixer service to not have the practice of using a constant transaction fee for a period of time, which would make this criterion inapplicable.

Service Address Profiling We incorporate the external address information of identified cryptocurrency services (excluding mixer services) described in Section 3.1.1 to exclude transactions that are unlikely to be a part of the targeted mixer service’s operation from the tainting results. The filtering process with service address profiling operates by first identifying tainted addresses that belong to a cryptocurrency service. Subsequently, the process removes transactions that have any identified service address in the transaction inputs and their subsequent transactions from the results.

4.2 Sample Cases

We use 15 mixing transaction samples from previous studies [47, 127, 181] which have shown that transaction taint analysis could not taint the withdrawn Bitcoins from the deposited Bitcoins. These studies perform reverse engineering on prominent centralised mixer services: Blockchain.info’s Shared Send function, Bitcoin Fog, Bitlaundry, BitLauder, DarkLauder, Alphabay and Helix Light, as shown in Table 4.1.

<i>Case</i>	<i>Service</i>	<i>Mixing Time*</i>	<i>#Withdrawn TX</i>
1	Blockchain.info	0 confirmation	1
2	Bitcoin Fog	176,177 confirmations	2
3	Bitcoin Fog	1,114, 1,326 confirmations	2
4	BitLaundry	34 confirmations	1
5	BitLaundry	154 confirmations	1
6	Unnamed Mixer 1	7 confirmations	1
7	Unnamed Mixer 2	4, 6, 6, 6, 6 confirmations	5
8	Bitlaunder	60 confirmations	1
9	Bitlaunder	41 confirmations	1
10	Darklaunder	64 confirmations	1
11	Darklaunder	404 confirmations	1
12	Alphabay	27 confirmations	1
13	Alphabay	42 confirmations	1
14	Helix Light	7 confirmations	1
15	Helix Light	2, 2, 2 confirmations	3

Table 4.1: Sample cases

* Mixing time is presented in the number of confirmations in the blockchain between the deposited and withdrawn transactions. If the deposited and withdrawn transactions occur within the same block, the number will be zero. If there are multiple withdrawn transactions in multiple blocks, the number will be shown in the number list as “10, 45, 60” for example.

As one study [181] chose to not publicly name their tested mixer services, we exclude any identifiable information of the services and transactions and refer to the mixer services from that study as “Unnamed”. It should be noted that one of the

mixing sample cases, namely case 5, has two withdrawn transactions, but only one of the transactions produce a zero-taint withdrawn output. As we use only zero-taint transactions as sample cases for this experiment, the number of the withdrawn transaction is listed as one for case 5.

For the address taint analysis experiment, we use the transaction hash of deposited transactions to perform the address taint analysis, and the transaction hash of withdrawn transactions are used to verify whether the address taint analysis can successfully reconnect the withdrawn Bitcoins back to the original deposited Bitcoins. If all of the targeted withdrawn transactions appear in the taint analysis results, we consider the experiment successful for that sample case.

In some of the mixing sample cases, a *change address* that belongs to the user is reused to interact with the withdrawn Bitcoins later on. This type of scenario can severely decrease the effectiveness of mixer services and make the mixed Bitcoins easily traceable. As user error is an extraneous variable that is not related to the mixer services and can affect the results of our experiment, we exclude any such change addresses from the deposited transactions.

4.3 Results and Discussion

4.3.1 Address Taint Analysis

As centralised mixer services typically perform the mixing operation continuously, it is possible for the service to deliver Bitcoins that are already mixed prior to the time of the deposited transactions. We set the time limit for the address taint analysis operation to begin taint analysis from five days before the deposited transactions until the maximum amount of mixing time allowed by the mixer service (e.g., Bit-Laundry allows up to a maximum of 10 days mixing time). If the mixer service did not have a mixing time setting, we set the time limit to three days.

As shown in Table 4.4, the results of our experiment demonstrate that even zero-taint mixed Bitcoins are not always perfectly immune to tracking. The majority of the sample cases show successful results overall except for the Blockchain.info and Bitcoin Fog cases, where the targeted withdrawn transaction outputs could not be found. The address taint analysis methods manage to accomplish the experiment's

Case & Service	Baseline	Method (%)			
		1	2	3	4
1 Blockchain.info	485,155	—	—	93	n/a
2 Bitcoin Fog	713,899	—	—	—	95
3 Bitcoin Fog	1,525,276	—	—	—	98
4 BitLaundry	1,013,374	83	83	96	97
5 BitLaundry	1,016,043	82	83	96	45
6 Unnamed 1	1,337,727	83	84	92	n/a
7 Unnamed 2	1,264,966	84	85	92	n/a
8 Bitlauder	1,867,536	80	81	93	89
9 Bitlauder	2,156,487	79	80	93	89
10 Darklauder	1,712,521	82	83	94	95
11 Darklauder	1,845,130	81	83	93	94
12 Alphabay	1,949,670	81	83	93	96
13 Alphabay	2,175,263	81	83	94	94
14 Helix Light	1,858,540	75	77	93	94
15 Helix Light	1,777,542	74	76	94	94

Table 4.2: Address tainting results

We indicate with — that the method’s experiment for the sample case was unsuccessful and with n/a the absence of an experiment. The percentage result represents the method’s transaction output number compared to the baseline before and after applying the filtering criteria, where the lower percentage means the fewer false positive results.

main objective, which is to reconnect the original deposited Bitcoins to the mixed Bitcoins, albeit with the extensive spreading of the tainted results. It should be noted that the number of transaction outputs in Table 4.2 (and later in Table 4.4 and 4.5) only count from when the deposited transactions occurred until the end of the mixing time limit.

The majority of the sample cases show successful results overall except for the Blockchain.info and Bitcoin Fog cases. For the majority of sample cases, Method 1 yields the lowest number of transaction outputs, compared to the other three methods and the Baseline method, followed by Method 2 and lastly Method 3. The number of transaction outputs for Method 1 is considerably lower than those of the Baseline method at roughly 20%. For example, Method 1 has 21% (443,816) fewer transactions than the Baseline method results for case 9, and 17% (171,811) fewer transactions for case 4.

The results of Method 2 are generally similar to those of Method 1. For example, Method 2 has only 1% (8,676) more transactions than Method 1 for case 7, and 2% (35,939) more for case 12. Meanwhile, Method 3 produces a greater number of

transaction outputs, compared to the first two methods, and is much closer to the results of the Baseline method. For example, Method 3 has 12% (199,707) more transaction outputs than Method 1 for case 10, and 6% (105,791) fewer than the results of the Baseline method. As such, our results suggest that the incorporation of address clustering and backward address taint analysis methods is not always necessary for the tracking of centralised mixer services.

4.3.2 Backward Address Taint Analysis

As shown in Table 4.2, the address taint analysis experiment on the Bitcoin Fog cases (2 and 3) produces unsuccessful results. This is because the mixer service keeps the deposited Bitcoins idle for as long as six months, which is outside the time period verification for our experiments.

While the initial deposited transaction for case 2 occurred on 2013-04-29, the deposited Bitcoins were not used at all until 2013-11-07, even though the withdrawn transactions occurred on 2013-04-30. This is similar to the situation for case 3. This type of scenario indicates that the central address clusters used for deposited and withdrawn transactions are separate and can not be connected because of the time limit constraint in this experiment.

Method 4 shows successful results for all sample cases as shown in Table 4.4. Although, aside from the Bitcoin Fog cases, the Method 4 taint analysis results (and the results after applying filtering criteria – see Section 4.3.3) generally do not provide improved results, compared to the other three methods. In particular, the number of transaction outputs resulting from Method 4 is higher than those of Method 3 for most cases. For example, Method 4 has 2% (30,480) more transaction outputs than Method 3 for case 10 which is only 4% (75,311) lower than the Baseline method results. However, there are some exceptions where Method 4 performs better than Method 3 such as for case 5 and 9, where the number of transaction outputs are 53% and 5% lower than those of Method 3, respectively.

Nevertheless, the results of the backward address taint analysis of Method 4 shows that it is possible to defeat the centralised mixer service operation with separate central address clusters. If one can initiate the mixing transactions at the same time as the targeted mixing transactions so as to perform backward address taint

analysis, one also can discover the central address clusters that are being used for the withdrawal of targeted mixed Bitcoins.

4.3.3 Filtering Criteria

<i>Service</i>	<i>C1</i>	<i>C2</i>	<i>C3</i>	<i>C4</i>	<i>C5</i>
Blockchain.info	0.5%	one-to-one	one-to-two	Y	10,000 Sat
Bitcoin Fog	1%	one-to-two	one-to-two	Y	50,000 Sat
BitLaundry	2.49%	one-to-two	one-to-two	Y	50,000 Sat
Unnamed 1	1%	one-to-two	one-to-two	Y	10,000 Sat
Unnamed 2	1%	one-to-two	one-to-two	Y	10,000 Sat
Bitlaunder	2%	N	N	Y	N
Darklaunder	2%	N	N	Y	N
Alphabay	10,000 Sat	one-to-two	N	N	N
Helix Light	2%	one-to-many	N	Y	50,000 Sat

Table 4.3: Sample mixer services and calibration of the filtering criteria. Letter “C” in the column headers is an acronym for Criterion. Letter “Y” indicates that the criteria can be applied to the mixer services and letter “N” indicates otherwise. The Bitcoin value is presented in Sat or Satoshis (the smallest unit of Bitcoin).

After performing address taint analysis on each sample case, we applied the filtering criteria listed in Section 4.1.3 on each method’s results for every case, as shown in Table 4.3. While the majority of the sample mixer services employ a one-to-two peeling chain method (continuous one-to-two transaction), there are some exceptions.

- The Blockchain.info’s shared send function operates slightly differently than the other mixer services. Instead of peeling the withdrawn Bitcoins and sending them to the users directly, the service always peels off the withdrawn Bitcoins and transfers them to one of its addresses first before sending them to the users in a one-to-one address transaction type. As such, Criteria 2 and 3 can still be applied for this mixer service case.
- The BitLaunder, DarkLaunder and Helix Light cases use a different version of a peeling technique. Instead of continuous one-to-two address transactions, the mixer services’ algorithm peels a single large value transaction input to multiple transaction outputs (one-to-many). Additionally, the mixing algorithm of the BitLaunder and DarkLaunder cases do not always perform the withdrawal transactions in one specific pattern. Hence, we can not apply Cri-

teria 2 and 3 for these two mixer services' samples. Moreover, we can not apply transaction fee Criterion 5 as the mentioned mixer services regularly adjust the transaction fee based on the transaction size.

Case & Service	Baseline Criteria	Method Criteria (%)			
		1	2	3	4
1 Blockchain.info	87	—	—	96	—
2 Bitcoin Fog	9,804	—	—	—	96
3 Bitcoin Fog	12,945	—	—	—	99
4 BitLaundry	24,885	95	95	99	99
5 BitLaundry	22,712	96	96	98	45
6 Unnamed 1	51,099	73	74	75	—
7 Unnamed 2	48,626	78	79	80	—
8 Bitlaunder	385,811	87	87	96	90
9 Bitlaunder	428,042	86	86	96	89
10 Darklaunder	333,400	84	86	96	96
11 Darklaunder	367,516	81	85	96	96
12 Alphabay	181,512	85	86	96	96
13 Alphabay	227,718	78	79	94	94
14 Helix Light	6,329	91	91	97	97
15 Helix Light	6,160	93	94	97	97

Table 4.4: Address tainting results with filtering criteria

We indicate with — that the method's experiment for the sample case was unsuccessful and with n/a the absence of an experiment. The percentage result represents the method's transaction output number compared to the baseline before and after applying the filtering criteria, where the lower percentage means the fewer false positive results.

The address taint analysis results show significant improvement in terms of the number of transaction outputs for all of the methods, including the Baseline method after applying the filtering criteria, as can be seen in the extensive reduction in the transaction outputs number shown in Table 4.4. Assuming that our assumptions are correct and the filtering criteria are correctly adjusted, this would mean that we've reduced the number of false positive transaction outputs.

For the sample cases for which we can apply more filtering criteria, namely case 1 to 7, 14, and 15, the number of false positive transaction outputs is reduced by 90% to as high as 99%. However, the transaction output number after applying filtering criteria for the first three methods is closer to the Baseline method outputs at around 10% lower. For example, the number of transaction outputs for Method 1 for case 5 is reduced by 97% (821,957), but when compared to the Baseline method's results, the difference in transaction output number becomes less after applying the filtering

criteria from 17% to only 6% lower.

While the sample cases that have less applicable filtering criteria, which are case 8 to 13, generally have a lower reduction in the number of transaction outputs at around 80%. When compared to the results of the Baseline method after applying filtering criteria, the number of transaction outputs show an increased reduction than for the other cases at around 20% lower. For example, the number of transaction outputs for Method 1 for case 11 is reduced by 80% (1,196,509) after applying the filtering criteria but is 18% (27,086) lower than the Baseline method.

However, there are cases where the results yield different result patterns. For example, for case 7 and 8, the number of transaction outputs is much lower for the three methods, compared to those of the baseline method, unlike the other cases with less applicable filtering criteria. Further, Method 1 for case 6 has the number of transaction outputs (after applying filtering criteria) that are 27% lower, and case 7 has 22% lower than the baseline. Interestingly, Helix light cases (case 14 and 15) show the highest reductions in the number of transaction outputs. We hypothesise this is due to the constant 50,000 Satoshis transaction fee used in the one-to-many transaction type that makes the withdrawn transactions extremely unusual, compared to other transactions.

The differences in the results may be because the exploitable transaction patterns of centralised mixer services have exceedingly unique patterns that make their transactions have characteristics that are considerably different from other transactions. Thus, this makes them less difficult to distinguish. We hypothesise that the fewer filtering criteria that can be applied to reduce the number of false positive results, the more of an advantage the address taint analysis can provide over the Baseline method. Nevertheless, the significant reduction in transaction outputs suggests that the filtering criteria can be adopted for other tracking methods of mixer services in general.

4.3.4 Service Address Profiling

The results shown in Table 4.5 are the results of each method applied with the filtering criteria first and followed by the cryptocurrency service address profiling.

The incorporation of service address profiling shows a further significant reduc-

Case & Service		Baseline Criteria + Profiling	Method Criteria + Profiling (%)			
			1	2	3	4
1	Blockchain.info	78	—	—	91	—
2	Bitcoin Fog	7,402	—	—	—	95
3	Bitcoin Fog	10,356	—	—	—	98
4	BitLaundry	19,532	92	93	93	96
5	BitLaundry	17,942	92	91	96	49
6	Unnamed 1	38,257	77	79	80	—
7	Unnamed 2	37,149	80	81	81	—
8	Bitlaunder	165,898	78	78	83	80
9	Bitlaunder	185,587	76	77	85	83
10	Darklaunder	170,034	75	82	87	88
11	Darklaunder	180,082	73	81	88	88
12	Alphabay	147,020	80	80	85	86
13	Alphabay	189,005	76	77	84	84
14	Helix Light	2,451	77	78	84	85
15	Helix Light	2,145	85	86	88	89

Table 4.5: Address tainting results with filtering criteria and service address profiling. We indicate with — that the method’s experiment for the sample case was unsuccessful and with n/a the absence of an experiment. The percentage result represents the method’s transaction output number compared to the baseline before and after applying the filtering criteria, where the lower percentage means the fewer false positive results.

tion in the number of false positive transaction outputs for all of the sample cases’ results without losing the targeted withdrawn outputs. The number of false positive transaction outputs in the baseline results are generally reduced by around 20 to 50% for most sample cases. For example, the number of transaction outputs for the baseline method is reduced by 52% (199,913) for case 8 and 19% (34,492) for case 12.

The address profiling results of address taint analysis methods show a further reduction in the number of transaction outputs, compared to the baseline method. The reduction pattern in false positive transaction outputs is similar to the filtering criteria results where the results of address taint analysis methods show around 10 to 20% lower number of transaction outputs, compared to the baseline method. The difference in the transaction output number after reduction between each method is typically not as significant, compared to the baseline method, at around 1 to 6% for most sample cases with Method 1 showing the highest reduction percentage, followed by Method 2, 3 and 4. Therefore, the results show that address profiling can remarkably improve the accuracy of both the baseline and address taint analysis

methods.

4.3.5 Reflection on Alternative Attempts

The motivation for this work stems from earlier experiments of the work in Chapter 3 where we could not track zero-tainted Bitcoins obscured by a mixer service with the transaction taint analysis strategies. This issue was an exceptionally difficult challenge since there had been no method capable of tracking zero-tainted Bitcoins. Earlier attempts of our demixing experiments used a similar approach as the baseline method that applies filtering criteria to create a list of potential withdrawn transactions. As illustrated in the experiment results, the number of false positive transactions for the baseline method is impractically large for most sample cases. Hence, we attempt to develop a new method that can track zero-tainted Bitcoins and provide more accurate demixing results in this work, which resulted in the address taint analysis method.

During earlier experiments, we discovered that the address taint analysis method is not applicable for every centralised mixer service (as shown in the results for BitcoinFog cases). Hence, we changed our experiment approach for the unsuccessful cases to discover whether it is possible to compromise the mixing operation by using a discovery attack on the mixer service to identify central address groups used for the mixing operation. This change in the approach resulted in the development of the backward address taint analysis method.

4.3.6 Limitations

Despite the successful results and potential of the address taint analysis and filtering criteria, there are limitations of our approach that we discuss below.

The number of tainted transaction outputs with and without the filtering criteria is still relatively large when compared to the number of targeted withdrawn transaction outputs, as can be seen in Table 4.2. The address taint analysis presented in this chapter taints the whole address, similar to the Poison strategy for transaction taint analysis, and does not utilise any other additional information besides the information of the deposited transactions. Future research might attempt to further reduce the number of potential outputs.

It may be possible to alleviate this issue by implementing more withdrawal criteria or adjusting the criteria' parameters to be more calibrated. For example, Criteria 3 can be adjusted to include more than a single previous or next transaction for some specific cases where the mixer algorithm uses a long peeling chain.

As this work examines and obtains sample mixing transactions only from published studies, it is essential to discuss the potential issue of publication bias. Outside of the nine sample mixer services investigated in this study, there are still many other mixer services that have been uninvestigated and may utilise different mixing mechanisms. Although there is a possibility that there are other types of mixing mechanisms immune to the address taint analysis method, the sample mixer services are among the most well-known Bitcoin mixer services, as mentioned in the previous studies [47, 127, 181]. Therefore, the sample data should be representative of the significant majority of centralised mixer services PET and the methodology presented in this work should be applicable to other centralised mixer services that utilise similar mixing mechanisms as the sample services.

Nevertheless, address taint analysis can be counteracted by the mixer services or the development of new PETs that defeat the taint analysis algorithm, similar to other transaction tracking methodologies. This is similar to how the CoinJoin method is introduced to oppose the multi-input address clustering heuristic or mixer services to prevent transaction taint analysis tracking. Hence, the address taint analysis method requires continuous development and improvement to remain applicable to new transaction obscuring techniques.

Additionally, the risk increases if the mixer service uses a randomised mixing algorithm to obscure any exploitable pattern. As the filtering criteria are currently designed based on the common transaction pattern found in the withdrawn transactions, the current filtering criteria would be less effective, as shown in the results of Table 4.4. This issue ultimately has a high probability of producing inaccurate results if the criteria are applied incorrectly. Thus, to avoid the risk of false incrimination of innocent users, the tracking method should always be utilised with caution and should only be implemented after a thorough exploration of the mixing algorithm involved.

The backward address taint analysis approach is also not without challenges.

As shown in the Bitcoin Fog cases, the receiver addresses are not reused addresses and have a remarkably long idle time after receiving the deposited Bitcoins. The approach operates with the requirement that the attackers identify which mixer service is used for the targeted mixed Bitcoins to perform backward address taint analysis within a similar time frame. The approach can be accomplished if the attackers can identify the mixer service with other means in time or attempt to perform backward address taint analysis attacks on every mixer service that employs this type of mixing algorithm.

There is also one potential limitation of the address profiling filtering that there may be a possibility that the targeted users set up the withdrawn transaction to directly send mixed Bitcoins to service addresses directly instead of their own addresses. In this scenario, the address profiling can cause the filtered tracking results to lose the targeted withdrawn outputs.

4.4 Conclusion and Future Work

As transaction obscuring methods improve, so should tracking methods to remain effective and relevant. We identify two possible improvements for both address taint analysis and filtering criteria, as follows:

- *More extensive address profile data.* The filtering process with address profiling can still be expanded to reduce the number of false positive results further. For example, as the results of the previous studies indicate that each mixer service typically does not directly interact with each other for their mixing operations [47, 127, 181], the address profiling can include other PETs unrelated to the targeted mixer service.
- *Incorporating more complex address clustering heuristics.* There is another address clustering heuristic that clusters based on transaction chain behaviour instead of a single transaction [79]. For example, when one address distributes its Bitcoins to multiple other addresses, then those addresses transfer all of the distributed Bitcoins to a single address. We can assume that most of the addresses involved are likely to belong to the same user. Such a clustering technique can also be combined to address taint analysis similar to the one we

implemented in this chapter.

While the use of cryptocurrency mixer services can remove the connection of the mixed Bitcoins from the original deposited Bitcoins and evade taint analysis tracking, the mixer services can still have weaknesses in their mixing algorithm that we can exploit to reveal the removed connections of mixed Bitcoins.

The address taint analysis methods we propose in this chapter have the potential for reconnecting the original deposited Bitcoins to the zero-taint mixed Bitcoins – this has not been possible with earlier taint analysis methods. We also illustrate that address taint analysis can be incorporated into other tracking methods such as address clustering and backward tainting methods for mixer services that utilise an irregular mixing algorithm. While the number of false positive results is still not substantially different between the Baseline and the other methods, by exploiting the transaction pattern of the withdrawn transactions to create filtering criteria, the number of false positive results can be reduced further.

With further improvement, our approach can be used to assist cryptocurrency crime forensics in clearing the mystery of past illegal activities, such as exchange service thefts. Nevertheless, more mixing samples from other mixer services are still required for evaluating and improving the tracking method further, considering that mixer services are constantly evolving as new transaction obscuring techniques are introduced.

Chapter 5

Wasabi CoinJoin Transaction Detection

As mentioned in Chapter 4, there has been only a few research that investigated cryptocurrency PET services. To the best of our knowledge, there is only one study that investigates one of the most well-known PETs in the current Bitcoin market, the Wasabi Wallet’s CoinJoin mixing [199].

We focus our investigation on the Wasabi Wallet in this chapter as it is a widely utilised PET for illegal activities in the present cryptocurrency ecosystem. For example, the report by the European Union Agency for Law Enforcement Cooperation (Europol) [57] shows that as high as 15 million USD worth of Bitcoins from darknet marketplaces were obscured by the Wasabi Wallet’s CoinJoin mixing in three weeks. Furthermore, there is no detection method for Wasabi CoinJoin transactions as far as we know, which is the necessary first step for the development of the demixing process to identify the movement of obscured Bitcoins and provide transaction context (PETs) to the taint analysis process.

While a Europol’s report [57] suggests that Wasabi CoinJoin transactions can be easily identifiable from other transactions, there are still two potential challenges that can significantly affect the accuracy of the detection results. First, it is possible for other transactions to have the same or some of the characteristics that Wasabi CoinJoin transactions possess. Second, the CoinJoin mechanism that Wasabi Wallet employ can be similar to other PET services, which can cause the CoinJoin transactions from those services to share similar transaction behaviours. Therefore, this

work investigates these two challenges and proposes Wasabi CoinJoin transactions detection method described below.

Methodology We follow a three-step methodology to construct and evaluate a detection mechanism for Wasabi CoinJoin transactions. First, we analyse the transaction patterns of published Wasabi CoinJoin transactions from different time periods. Second, we formulate transaction criteria from common characteristics shared by the transactions over the whole dataset and each individual time period. Third, we evaluate the criteria as a detection mechanism by applying individual time period criteria to other periods and all Bitcoin transactions since the start of the Wasabi Wallet. To expand this evaluation, we analyse the trails of Bitcoins originating from nine theft cases that took place during the timeline to see if any stolen Bitcoin is reaching Wasabi CoinJoin transactions.

Plan The remainder of the chapter is structured as follows. We discuss the relevant background and work related to the Wasabi Wallet and its CoinJoin mixing operation in Section 5.1. We describe the transaction data sets in Section 5.2. We identify the patterns of Wasabi CoinJoin transactions and propose our transaction detection method in Section 5.3. We discuss the results and evaluation of the proposed detection method and demonstrate its practical application with illegal transaction tracking in Section 5.4. In Section 5.6, we conclude and discuss potential improvements for future works.

5.1 Wasabi Wallet

Wasabi Wallet is an open-source Bitcoin wallet client that was launched on 2018-10-31 [140]. Wasabi Wallet can function as either a full node client or lightweight node. Wasabi Wallet can be classified as a decentralised mixer service due to its main characteristic, which is the integration of Chaumian CoinJoin privacy feature to its wallet client.

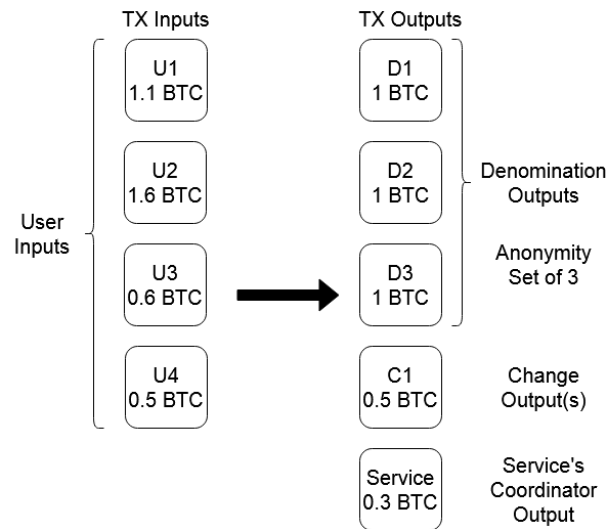


Figure 5.1: Example of Wasabi CoinJoin transaction

In this example, the service fee is not at the same rate as the actual Wasabi CoinJoin fee.

5.1.0.1 Wasabi CoinJoin Mechanism

As shown in Figure 5.1, the Wasabi CoinJoin protocol uses an *anonymity set* system where CoinJoin transactions combine transaction inputs from several users together and generate sets of denomination outputs (mixed outputs) with exactly the same transaction output values (e.g., 50 of 0.1 BTC outputs). The minimum value of denomination outputs is roughly between 0.095 and 0.105 BTC. While the service recommends around 50 to 100 anonymity sets for the minimum privacy level, the minimum number of anonymity sets can start from two outputs¹ with no upper limit.

The Wasabi Wallet's CoinJoin mixing for one user can occur in multiple transactions (rounds) until the target anonymity set level assigned by the users is approximately reached. Using the example in Figure 5.1, only 1 BTC out of U2's 1.6 BTC is mixed by the service in this round, and the 0.5 BTC change output will be anonymised in another transaction set with another anonymity set. The service also specifies that users can only participate in the CoinJoin mixing if the total starting Bitcoin value is no lower than 0.1 BTC.

Similar to other decentralised mixer services, the Wasabi Wallet's CoinJoin mixing also uses coordinator address(es) to receive the mixing fee from the users in the

¹It is possible for the anonymity set to be at one output, whereas Wasabi CoinJoin system returns the single output as a change output [190].

Table 5.1: Published transaction data sets

Period	Time Range	# Day	# Total TX	# Published TX	Avg. Published TX per Day	Source
1*	2018-07-19/2019-02-18	215 days	55,212,658	1,432	13.5	Static service addresses
2	2019-02-19/2019-07-18	150 days	52,053,626	2,562	18.6	Static service addresses
3	2019-07-19/2020-02-03	200 days	63,068,691	4,136	21	Static service addresses
4	2020-02-04/2020-03-15	41 days	13,349,570	1,338	33	Service API [199]
5 [∅]	2020-03-16/2021-09-23	557 days	159,642,256	Unknown	-	Not collected
6	2021-09-23/2021-09-29	7 days	1,608,747	140	19.5	Service API
7 [∅]	2021-09-30/2021-12-18	80 days	21,835,306	Unknown	-	Not collected
8 ^E	2021-12-19/2022-01-06	19 days	4,761,319	429	21.45	Service API
Total	2018-12-19/2022-01-06	1,269 days	371,532,173	10,037	21.1	Various

* The first period includes transactions that occurred during the beta testing before the official release. [∅] indicates the lack of transaction data in the period. ^E indicates that we use transaction data in the period only for evaluation.

CoinJoin transactions. Wasabi Wallet has a static mixing fee rate for each mixing attempt at 0.003% of the base denomination per anonymity set (e.g., mixing fee of 1 BTC denomination output of 50 anonymity set is at 0.15% with 0.15 BTC fee).

5.2 Data Sources and Collection

In this section, we describe the data sources and collection process of the Wasabi’s CoinJoin transactions we utilise for transaction pattern analysis and as ground-truth data to evaluate the detection method for this experiment.

5.2.1 Sources and Collection

In order to observe the transaction patterns of the Wasabi CoinJoin transactions and measure the effectiveness of the detection method, we make use of transaction data that can be verified as CoinJoin transactions performed by the Wasabi Wallet service. For this purpose, we obtained Wasabi CoinJoin transaction data from five different time periods from the start of the service until the present day, as shown in Table 5.1. It is worth noting that we exclude transactions with no anonymity set (i.e., two or more outputs with the exact same value) from the data sets as these transactions are unrelated to the CoinJoin mixing operation. We also obtained Wasabi CoinJoin transaction data from one additional time period for the evaluation of the detection method.

According to the developers of Wasabi Wallet, the service used two specific static coordinator addresses for its CoinJoin transactions during its first two years of operation [195]. The developers stated that they changed the CoinJoin system to use a fresh coordinator address for every CoinJoin transaction to improve privacy

and reduce the detectability of the mixing transactions starting from the date 2020-01-31.

We obtain Wasabi CoinJoin transactions in the first two years by identifying transactions with the two coordinator addresses, which were in use in Wasabi CoinJoin transactions during the 2018-07-19/2020-02-03 period. We divide the transactions we obtained using the two static addresses into three separate periods to observe and identify potential changes in transaction behaviours and mixing mechanisms that can affect detection criteria, as shown in Table 5.1.

Wu et al.’s research [199] on Wasabi Wallet discovered that the Wasabi Wallet provides a public API² that allows anyone to access and retrieve the current unconfirmed CoinJoin transaction hash data, the transaction data is removed from the API when the transactions are confirmed into the blockchain. The researchers retrieved the Wasabi CoinJoin transaction data using a website crawler on the API, continuously every one minute during the 2019-12-26/2019-03-15 dates.

We retrieve the scraped transaction data from Wu et al.’s research [199] to use more recent data sets of Wasabi’s CoinJoin transactions that occur after the discontinuation of reused coordinator address between the 2019-02-04/2020-03-15 periods. We replicated the same API crawling method to obtain more recent Wasabi CoinJoin transaction data. We performed a continuous website crawling every 30 seconds for one week and retrieved transaction data during the 2021-09-23/2021-09-29 period for the creation of criteria and for around three weeks to retrieve transaction data during the 2021-12-19/2022-01-06 period for evaluation purposes. We designate the transactions in the data sets as “published transactions” and the sourceless periods as “5 \emptyset ” and “7 \emptyset ” to signify that we did not obtain the transaction data in these periods.

We are unable to obtain Wasabi CoinJoin transactions that occurred between the 2020-03-16/2021-09-21 period, as there are no other external sources for the information or method that we are aware of to retrieve the data.

As shown in Figure 5.2, the overall average number of transactions per day is between 20 and 30 transactions, which is consistent with the Wasabi developers’ statement that the service aims to perform CoinJoin transactions at least once every

²<https://wasabiwallet.io/api/v4/btc/chaumiancoinjoin/unconfirmed-coinjoins>

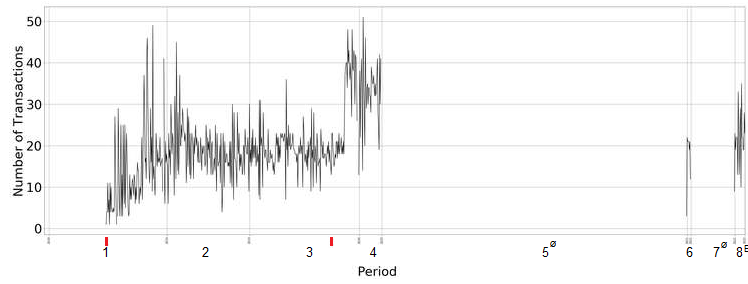


Figure 5.2: Collected Wasabi CoinJoin transactions per day
 Red lines below the horizontal axis denote events related to the Wasabi Wallet service that can affect transaction activity. We did not collect Wasabi CoinJoin in Period 5[∅] and 7[∅] due to the lack of data source. We utilise transactions from Period 8^E only for evaluation purposes.

hour [193]. The number of published transactions per day show a steady increase as time passes, possibly due to the increase in the size of the service’s user base. For example, as indicated by the red lines below the horizontal axis in Figure 5.2, the release of Wasabi version 1.0 [189] in Period 1 (2018-10-31) that occurred two months after the beginning of Period 1 shows a significant increase in transaction activity. There is a noticeable increase in transactions again after the release of Wasabi version 1.1.10 in Period 3 (2019-12-14), which contains a large number of feature updates, such as client performance and user interface improvement, which may entice a large number of new users.

5.2.2 Reliability of Data per Period

While the published transaction data we obtained are from the sources provided by the Wasabi Wallet, there is still the question of the reliability of the sources of information and the possibility that the data sets do not include every Wasabi CoinJoin transaction.

There is a possibility that the two published coordinator addresses are not the only two addresses that the Wasabi Wallet use to perform every CoinJoin transaction in Period 1, 2, and 3. The transaction analysis results from Wu et al.’s research [199] indicate the presence of the two same addresses being the most reused addresses in the Wasabi CoinJoin transactions in Period 1, 2, and 3. Hence, these two addresses are still likely involved in the majority of Wasabi CoinJoin transactions in those periods.

The service API scraping method has potential limitations, namely that the

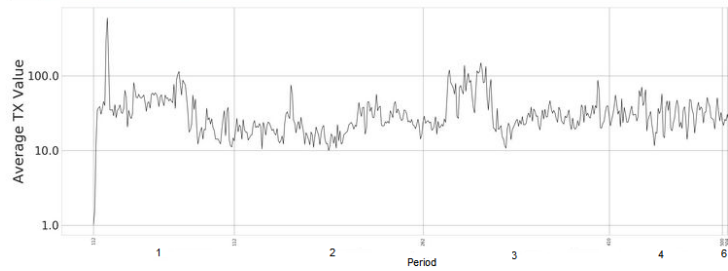


Figure 5.3: Wasabi CoinJoin transaction value

scraping process can miss some transactions if the service API goes down for a significant period of time or transactions occur and are confirmed during the scraping process's waiting time interval. Therefore, there is a possibility that the transaction data in Period 4 and 6 may miss some Wasabi CoinJoin transactions.

5.3 Identification of Criteria

We first examine the transaction patterns of the published transactions (Section 5.3.1) to formulate a set of criteria for Wasabi CoinJoin transaction (Section 5.3.2) that we later use for detection.

5.3.1 Examining Wasabi CoinJoin Transaction Patterns

Transaction Value As shown in Figure 5.3, the published transactions' value in all five periods shows a considerably similar pattern where the majority of the transactions have an average transaction value between 20 and 70 BTC. Although, the transactions at the start of Period 1 and 3 show considerably higher values, compared to the other periods. This difference can be either due to a change in the CoinJoin mixing algorithm or simply the increase in the service's activity, which would increase the transaction value in the CoinJoin transactions.

Transaction Shape As shown in Figure 5.4, the published transactions illustrate a significant increase in the transaction size in the more recent periods. The constant proportion pattern throughout the five periods reduces the possibility that the change in transaction size is due to the change in the CoinJoin mixing mechanism, but rather the activity of its user base at the time. We also discover a common

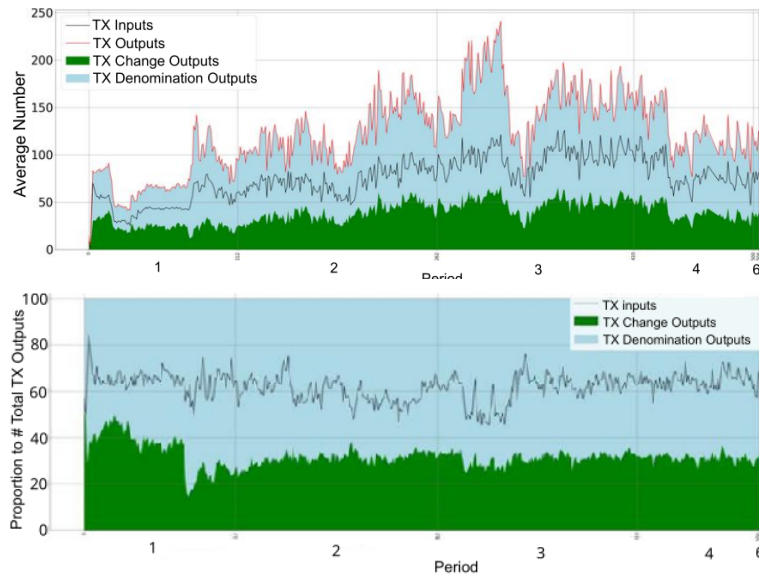


Figure 5.4: Wasabi CoinJoin transaction inputs and outputs

For the above graphs, the bottom green stacked part represents change outputs, the light blue part represents denomination outputs (both of which make up the overall outputs), and the black line represents inputs. The first graph shows average numbers while the second shows proportions of the number of the output.

transaction pattern that the number of transaction inputs in the Wasabi CoinJoin transactions is consistently lower than the number of transaction outputs as the transaction outputs always contain change outputs in addition to the denomination outputs.

Anonymity Set The number and proportion of denomination outputs indicate that the significant portion of the transaction outputs in the Wasabi CoinJoin transactions are typically denomination outputs, as shown in Figure 5.4. The majority of the published transactions show an average denomination output number at around 100 for all five periods, which is consistent with the developers' statement [187, 188]. We discover three common transaction patterns regarding the anonymity set. First, the transaction order of denomination outputs from the same anonymity set is always in continuous order. Second, the transaction order of different anonymity sets is sorted starting from the lowest to the highest output value. For example, if a transaction has two different anonymity sets with the denomination output values of 0.1 and 0.5 BTC, the 0.1 BTC anonymity set will be positioned first in the transaction outputs order. Third, the number of transaction inputs must be equal to or greater than the number of the most frequent denomination outputs.

Change Output The change output pattern seems to correlate with the anonymity set pattern, where the periods with a higher number of denomination outputs also show a higher number of change outputs. Interestingly, there is no Wasabi CoinJoin transaction in the data sets without change outputs, which implies that the payment outputs to coordinator addresses are unlikely to have the same output value as the denomination outputs.

Address Type All of the published Wasabi CoinJoin transactions contain only the Pay-to-Witness-Public-Key-Hash (P2WPKH) address type for both transaction inputs and transaction outputs, which indicate that the Wasabi Wallet makes use of only one address type for its CoinJoin mechanism.

5.3.2 Defining Criteria

There are five common transaction patterns that we can apply to the Wasabi CoinJoin transaction detection as General Criteria regardless of the time period, as shown in Table 5.2. Since published transactions also possess significantly different transaction patterns in each period, we derive the minimum value of the transaction patterns discovered in each period to create a set of Period-specific Criteria for detecting all Wasabi CoinJoin transactions, including outlier transactions.

We combine the General Criteria and each set of Period-specific Criteria to create a detection method. We designate the combination of General Criteria and each Period-specific Criteria as “Period *number* Criteria” (e.g., Period 2 Criteria). In essence, the detection method inspects transactions in the Bitcoin blockchain and identify any transaction that matches the criteria as potential Wasabi CoinJoin transactions. We use the published transaction data to evaluate the detection method’s performance, where any transaction outside the data sets in the same period is potential false positive results.

As discussed in Section 5.2.2, the data sets may not contain every Wasabi CoinJoin transaction. We identify the Wasabi CoinJoin transactions that are outside of the data sets by checking whether the transactions contain transaction inputs or outputs used in the published transactions. If transactions that are classified as Wasabi CoinJoin transactions by the detection method but are not found in the

Table 5.2: General Criteria and Period-specific Criteria

General Criteria			Period				
			1	2	3	4	6
Transaction Shape	The number of transaction inputs is lower than the number of transaction outputs, but equal to or greater than the most frequent denomination outputs.						
Denomination Output Value	The denomination outputs in anonymity sets have a value higher than 0.095 BTC.						
Denomination Output Order	The denomination outputs from the same anonymity set is in continuous order with no change outputs in between.						
Anonymity Set Order	The order of anonymity sets is sorted from the lowest to the highest value.						
Address Type	Every transaction input and output in the transactions is a P2WPKH address type.						
Period-specific Criteria (includes General Criteria)			1	2	3	4	6
Transaction Value	The minimum total Bitcoin value in BTC in transaction outputs.		1	0.9	1.5	3.4	8.18
Transaction Output Number	The minimum number of transaction outputs in the transaction.		4	9	14	12	68
Transaction Input Proportion	The minimum ratio of transaction inputs to transaction outputs, in percentage.		40%	33%	28%	44%	53%
Denomination Output Proportion	The minimum ratio of denomination outputs to transaction outputs, in percentage.		8.1%	1.3%	56%	3.3%	60%
Change Output Proportion	The minimum ratio of change outputs to transaction outputs, in percentage.		8.1%	15%	16%	17%	24%

There are no Period-specific Criteria for Period 5[Ⓞ] and 7[Ⓞ] due to the lack of published transaction data. Period 8^E also does not have criteria as we only use published transaction data in this period for evaluation of the criteria.

Table 5.3: Detection method results

Criteria	Published TX Coverage %						Potential False Positive TX % (Excluding Connected TXs)								# Period 5 ^o and 7 ^o TX Detection				
	1	2	3	4	6	8 ^E	1		2		3		4			6		8 ^E	
General	100	100	100	100	100	100	4.2	(3.7)	1.6	(1)	5.2	(2.4)	13.9	(2.5)	17.1	(15.9)	6.5	(6.5)	21,355
Period 1	100	99.7	99.6	100	100	100	1.6	(1.3)	0.3	(0.2)	3.1	(0.6)	11.5	(0.8)	6.6	(5.3)	3.3	(3.3)	20,362
Period 2	96.1	100	100	100	100	100	0.1	(0.1)	0.03	(0)	2.4	(0.02)	11.1	(0)	3.4	(2)	0.4	(0.4)	19,281
Period 3	94	100	100	99.8	100	100	0.07	(0)	0	(0)	2.3	(0)	10.8	(0)	1.4	(0)	0.2	(0.2)	18,384
Period 4	91.5	98.7	97.7	100	100	100	0.07	(0.19)	0	(0)	2.3	(0)	10.7	(0)	2.1	(0.7)	0.2	(0.2)	18,071
Period 6	33.5	73.3	84.3	87.3	100	85.7	0	(0)	0	(0)	2.1	(0)	11.2	(0)	0	(0)	0	(0)	14,582

The values in parentheses in the potential false positive transaction percentage column is the percentage after excluding those that have a direct connection to the published transactions as described in Section 5.3.2.

published transaction data share the direct connection with the published transactions, there is a high possibility that these transactions are also Wasabi CoinJoin transactions, and thus unlikely to be false positive results.

5.4 Detection Results and Discussion

We test the detection method with the General Criteria only and the General Criteria with each Period Criteria on every transaction in the Bitcoin blockchain for the 2018-07-19/2022-01-06. The algorithm of the detection method we used to perform the experiment is shown in Algorithm 10.

5.4.1 Detection Results

The results of General Criteria and all Period Criteria show a significantly high published transaction coverage for all five periods, as shown in Table 5.3. Both the General Criteria and the first four Period Criteria demonstrate 100% transaction coverage with small decreases in some periods. These Period Criteria show a capability to detect most of the published transactions and generally miss a small number of transactions. Meanwhile, Period 6 Criteria shows much less detection capability for Wasabi CoinJoin transactions with exceedingly lower published transaction coverage in Period 1 at only 33.5% and around 80% for the other periods.

The results show considerable differences in the potential false positive transaction results. The General Criteria show a significantly higher number of potential false positive transactions, compared to all Period Criteria both before and after excluding connected transactions, reaching as high as 3.7% in Period 1, 15.9% in Period 6 and 6.5% in Period 8^E. The General Criteria results indicate that using General Criteria without Period-specific Criteria can produce a large number of false

positive results. Period 1 Criteria also shows a considerably high number of potential false positive transactions for the first three periods and reach as high as 11.5% in Period 4. After excluding the connected transactions, the results of Period 1 Criteria show a significant decrease in potential false positive transactions to around 1% for the first four periods, but remain exceptionally high at 5.3% for Period 6 and no decrease for Period 8^E. The potential false positive transactions connected to the published transactions in the results of Period 1 Criteria are likely to include transactions unrelated to Wasabi CoinJoin due to the less strict parameter.

The remaining four Period Criteria results show a very low number of potential false positive transactions for the first two periods and Period 8^E. Intriguingly, Period 3 and 4 results show similarly high potential false positive transactions at around 2% and 11% respectively, but almost all of the discovered potential false positive transactions show a connection to the published transactions. These results indicate that the published transaction data is likely to miss a considerable number of Wasabi CoinJoin transactions in these periods.

Period 6 results show a considerable increase in the number of potential false positive transactions for Period 1, 2, and 4. Criteria Period 1, 2, and 4 Criteria show potential false positive transactions at 6.6%, 3.4 and 2.1%, respectively, and only around 1% of these transactions are connected to the published transactions. Meanwhile, Period 3 and 5 Criteria results show no false positive transactions. Period 8^E results show a relatively high number of potential false positive transactions for Period 1 Criteria at 3.3% but very low for the other Period Criteria.

The significant reduction of the false positive results after excluding the transactions with connection to the published transactions indicates that there are not many other Bitcoin transactions that share similar transaction patterns as the Wasabi CoinJoin transactions, and presumably that there was no other major PET service that performs Bitcoin mixing in a similar way as the Wasabi Wallet during the early periods.

However, the increase in the number of potential false positive transactions in Period 6 and 8^E for the less strict Period Criteria implies that there is an increasing number of unrelated transactions that share similar patterns with the Wasabi CoinJoin transactions in the later periods. We provide two possible explanations for

this development. First, some of the false positive transactions are Wasabi CoinJoin transactions that our service API crawling process missed and are not connected to the published transactions. Second, these false positive transactions are created by other recently developed PET services that utilise a similar CoinJoin mixing pattern, which is the second challenge we described at the beginning of the chapter.

The second argument presents a crucial issue for the practicality of the detection method since the mixing mechanism that other PET services utilise can be different even if they share similar transaction patterns as Wasabi CoinJoin transactions. The demixing process relies upon the correct identification of PET transactions and their mixing mechanism to determine the movement of obscured Bitcoins correctly. To validate this argument, we examine the other well-known PET services that also utilise an anonymity set based mixing mechanism, namely the ChipMixer service and the Samurai Wallet's Whirlpool CoinJoin.

As described in Section 3.1.2, the ChipMixer service's anonymity set mixing protocol is typically distinguishable from that of Wasabi CoinJoin mixing. The ChipMixer mixing transactions contain only one anonymity set of outputs (chips) with a value between 0.001 BTC to 4.096 BTC. The anonymity set outputs are always numbers with three digits, such as 0.007 BTC. Unlike Wasabi CoinJoin transactions, the ChipMixer's mixing transactions have only one change output, which is the coordinator address output [199]. The algorithm to identify the ChipMixer's mixing transactions is shown in Algorithm 12.

The Samurai Wallet's CoinJoin Whirlpool protocol also perform its CoinJoin transaction in a specific pattern. According to the whirlpool protocol documentation [159], the Samurai's CoinJoin transactions always contain five transaction inputs and five transaction outputs with the same output value of either 0.01, 0.05 or 0.5 BTC. The Samurai's CoinJoin transactions have no change outputs, and consequently, all of the transaction inputs have a Bitcoin value no less than the transaction output value. The algorithm to identify the Samurai Wallet's mixing transactions is shown in Algorithm 11.

We confirm that none of the results classifies transactions that match the patterns of mixing transactions from the ChipMixer and Samurai Wallet. However, there is still the possibility that there may be other PET services with a similar mixing

of the first four Period Criteria with a higher than 90% score for every metric for all periods. Period 1 Criteria show a slightly lower score in the precision and F1-Score metrics, compared to the other Period Criteria.

Overall, the results of Period 2, 3, and 4 Criteria show slightly lower recall and F1-Score performance for the early periods and Period 8^E but still remain above 95%. Meanwhile, the results of Period 6 Criteria show a 100% precision score for all five periods but shows an overall low performance score in the recall and F1-Score metrics due to the very strict criteria parameters.

The evaluation results indicate that for all five time periods, Wasabi CoinJoin transactions are successfully detected with a relatively small number of false positive results by the Period 2, 3, and 4 Criteria. Period 6 Criteria are capable of producing very high precision detection results with minimal false positive results but will likely miss a significant portion of Wasabi CoinJoin transactions. Additionally, the Period Criteria provide a clear benefit over simply utilising only the General Criteria as a detection method.

5.4.3 Analysis of Detection Results for the Sourceless Periods

The Period 5[∅] and 7[∅] transaction detection results are similar to the published transactions coverage results, where the General Criteria and the less strict Period Criteria detect more potential Wasabi CoinJoin transactions than the more strict Period Criteria, as shown in Table 5.3. However, there is a high possibility that a significant number of transactions in Period 5[∅] and 7[∅] are likely to be false positive transactions for the General Criteria and the less strict Period Criteria.

As shown in Figure 5.6, the number of transactions per day reveals further insight into the Period 5[∅] and 7[∅] results. The General Criteria and Period 1, 2, 3, and 4 Criteria reveal relatively similar transaction activity patterns in Period 5[∅], where the transaction activity show a considerable increase in transaction activity to around 50 to 80 transactions per day and reach an abnormally high level with the highest at 140 transactions per day. There is a possibility that some transactions that contribute to the high transaction activity are from false positive results. Meanwhile, the results of Period 6 Criteria exhibit overall more consistent transaction activity throughout

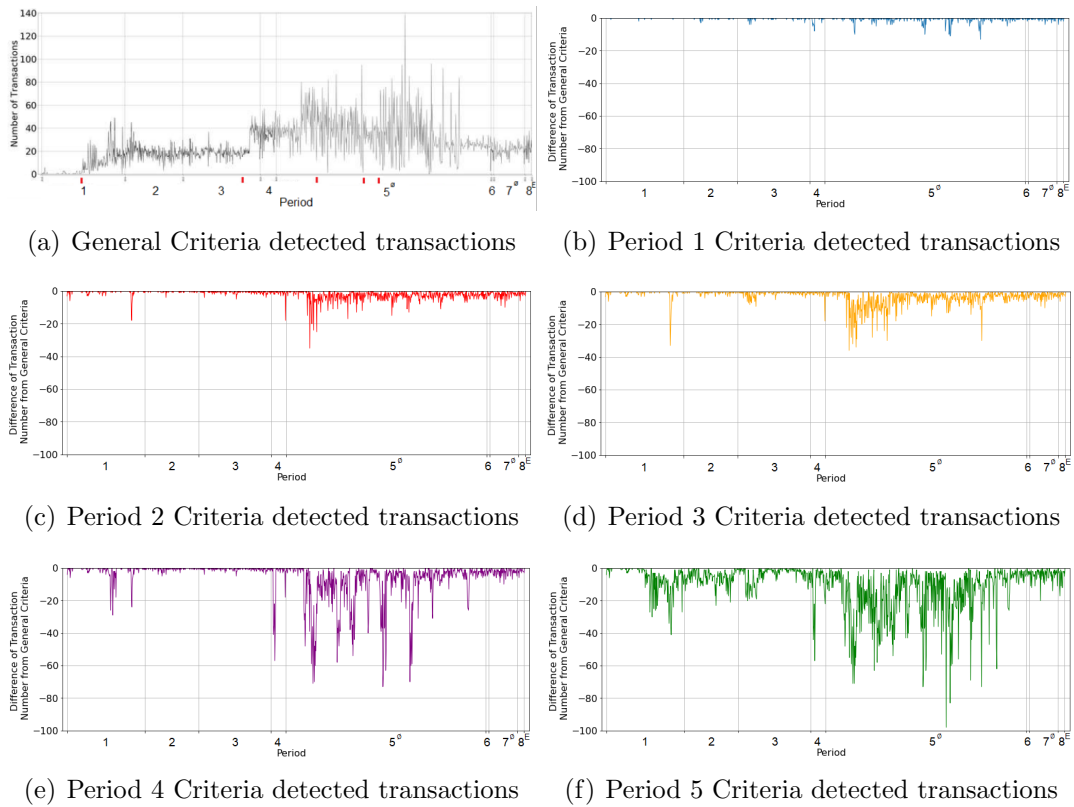


Figure 5.6: Detected Wasabi CoinJoin transactions per day

Red lines below the horizontal axis denote events related to the Wasabi Wallet service that can affect transaction activity. The Period Criteria results figures represent the number of transactions compared to the General Criteria results.

the whole period at no more than 60 transactions per day, but display more instances of days with very few or no transactions, compared to other Period Criteria. The results may indicate that Period 6 Criteria misses a significant portion of Wasabi CoinJoin transactions on those days.

One possible explanation for the increase in transaction activity among the first four Period Criteria at a similar time is due to an increase in the service’s activity influenced by events related to the service, as indicated by the red lines below the horizontal axis in Figure 5.6. These events are the announcement or the software update from the developers of Wasabi Wallet that indicate improvements to the service, some of which improve the anonymisation process. Therefore, these events can attract more users to utilise the service and its CoinJoin feature, which would affect the transaction number.

The events that occurred in Period 5^o, the version 1.1.12 release [191] (2020-08-05), which introduces the *PayJoin* support³, and the announcement of the Wasabi

³Payjoin or pay-to-end-point (P2EP) is a sub-type of CoinJoin that allows two parties to perform

version 2.0 [192] (2020-11-05) do not seem to significantly influence the transaction activity. However, the transaction activity after the status update report of Wasabi version 2.0 [194] (2020-12-12) shows a significant increase in the following months but drops to earlier levels soon after. We hypothesise that this pattern is the result of the news enticing a significant number of new users to try the service, although most choose to wait for the following update release before using the service regularly.

Intriguingly, there are several instances where all of the Period Criteria detect an entire day with a very low or zero number of Wasabi CoinJoin transactions in Period 5^o. The complete absence of the Wasabi CoinJoin transaction for an entire day had not occurred since the official launch of the service. There are two possible explanations for this peculiar pattern; first, the service could not perform CoinJoin transactions due to the lack of users' activity on those days. Second, the service's operation went into major maintenance or suffered downtime. Unfortunately, we are unable to find online evidence that can confirm either of these two hypotheses. Nevertheless, the low number or absence of transaction activity on certain days in Period 5^o further reinforces the hypothesis that Wasabi CoinJoin transactions' patterns are unique in the Bitcoin blockchain.

The number of transactions in all results show a sharp decrease in the middle of Period 5^o and stay relatively stable for the rest of the subsequent periods at around 20 to 30 transactions per day. We are unable to locate the information of an update release that indicates a change in Wasabi CoinJoin transaction frequency. There is a high possibility that this sudden transaction activity change is unlikely to be due to the users' activity but rather is the mixing mechanism change that happened on the service's server-side.

5.4.4 Application of the Detection Method

To demonstrate the practical application of the detection method, we investigate transaction trails of nine publicly known major cryptocurrency theft incidents that occurred close to or during the eight periods that we are investigating in this chapter. We follow the stolen Bitcoins for 20 transaction depths (also commonly referred to as levels or jumps) starting from the theft transactions (i.e., all of the subsequent Bitcoin payment and CoinJoin mixing at the same time [80]).

transactions that use transaction outputs in the theft transactions with the depth level of zero as the inputs are transactions with the depth level of one). We use the results of Period 2 Criteria, which show the highest performance score for all five periods, to identify Wasabi CoinJoin transactions that involve stolen Bitcoins.

We discovered three separate Bitcoin theft incidents where the stolen Bitcoins are directly transferred using few transactions to the detected Wasabi CoinJoin transactions without the presence of unrelated Bitcoins, which provides strong evidence that the criminals obscure the stolen Bitcoins with Wasabi Wallet’s CoinJoin mixing. The proportion of the stolen Bitcoins that directly reach Wasabi CoinJoin transactions in the three cases are 100%, 0.3% and 0.8%, the value of which makes up to the total of around 1.7 million USD⁴. We were able to detect eight transactions in Period 2, accumulating to a total of 424,649 USD, ten transactions in Period 3 with a total of 1,077,140 USD, and five transactions in Period 5[∅] with a total of 200,152 USD.

We also discovered that all of the six other cases show stolen Bitcoins reaching the detected Wasabi CoinJoin transactions but only after being mixed with unrelated Bitcoins in a few prior transactions. The Wasabi transactions are typically discovered at around a transaction depth of two to nine levels. These theft cases require further analysis before we can confirm that the stolen Bitcoins from these cases truly reach the Wasabi CoinJoin transactions.

5.5 Reflection on Alternative Attempts

As presented in Chapter 3, we made use of the two static coordinator addresses to detect Wasabi CoinJoin transactions in earlier experiments. The method can detect only Wasabi CoinJoin transactions from the year 2018 until 2020 as the service changed its mixing mechanism to utilise a fresh coordinator address in every transaction. The method is outdated for the detection of newer CoinJoin transactions.

During earlier experiments, we used only the General Criteria from static transaction characteristics of Wasabi CoinJoin transactions to create the detection method, which produces a considerable number of false positive transactions, as shown in the

⁴We converted the value by using the Bitcoin exchange rate from <https://www.coindesk.com/price/> at the time of Wasabi’s CoinJoin transactions.

experiment results. We introduce Period Criteria to refine the detection method using dynamic transaction characteristics as the minimum threshold to exclude transactions that are unlikely to be Wasabi CoinJoin transactions.

5.6 Conclusion and Future Work

The experiment's results demonstrate that the Wasabi CoinJoin transactions are identifiable and distinguishable from other Bitcoin transactions without requiring the identification of the coordinator addresses due to their unique transaction patterns. Our results are characterised by significantly high precision and minimal potential false positive occurrences in all five periods and demonstrate that the detection method is able to assist the forensic analysis of illegal activities in detecting illegal Bitcoins obscuring by using Wasabi Wallet's CoinJoin mixing. Furthermore, the application of our detection method on several major cryptocurrency theft incidents reveals the stolen Bitcoins with the total value of 1.7 million USD directly reached and obscured by Wasabi CoinJoin transactions.

We conclude that the Period Criteria with less strict parameters are able to discover more transactions but will likely produce a higher number of false positive results. On the other hand, the Period Criteria with more strict parameters are able to detect fewer Wasabi CoinJoin transactions but are less likely to produce false positive results. Overall, Period 2, 3, and 4 Criteria are capable of producing a very high performance for identifying Wasabi CoinJoin transactions from all periods, compared to the General Criteria and other Period Criteria.

The detection method provides essential progress to cryptocurrency forensics by identifying Bitcoin obscuring instances with one of the most well-known mixer services. The detection results can provide the context of transaction purpose for the cryptocurrency forensic, which will allow the tracking process to apply an appropriate demix strategy to accurately follow targeted Bitcoins after they are being mixed by Wasabi CoinJoin transactions.

The Wasabi CoinJoin transaction detection method can still be improved further. The considerable number of false positive transactions suggests the possibility that there are other PET entities that use a similar transaction mixing mechanism as

the Wasabi Wallet. While we have addressed the potential issue of classification overlap with two other well-known PET services, future research should expand and analyse other similar PET services we have yet to explore in order to calibrate the criteria parameter setting and reduce potential false positive results further. The next step of our works would be to reconstruct the transaction activity and devise a demixing heuristic that untangles the movement of obscured Bitcoins to discover their potential destinations.

Chapter 6

Conclusions

We conclude this thesis in this last chapter. We first provide a summary of our contributions (Section 6.1) and then discuss potential future research (Section 6.2). We finish the thesis with concluding remarks (Section 6.3).

6.1 Summary of Contributions

We summarise the key contributions of this thesis to the research objectives we described in Chapter 1 and their potential use cases as follows.

Context-based Tracking (Chapter 3) Our first contribution in Chapter 3 demonstrates that taint analysis can be significantly improved by utilising external information to provide address profile data to the tracking process. The address profiling implementation makes the process capable of avoiding following tainted coins that are no longer in the hands of targeted users and thus produce less number of unessential transactions, which contributes to RO 1 and 3. The implementation of the evaluation metrics reveals that the stolen Bitcoins that pass through service and mixer entities show a significant change in transaction behaviours, and the transaction behaviours in Bitcoin theft cases are considerably different from those of the control groups, which signifies that they can be utilised for detecting changes of coin ownership and thus contributes to RO 1. Additionally, one of the context-based tracking strategies, Dirty-First, shows positive potential for revealing illegal Bitcoins' spending and obscuring strategies, which contributes to RO 3. The

work in this chapter contributes to RO 2 by providing methodologies to obtain address and transaction profile data from both internal (blockchain data) and external sources for the tracking process, thus allowing the tracking methodologies to be able to determine the change of hands of targeted Bitcoins. The work in this chapter also achieves a part of RO 4 by providing detection methods for Bitcoin obscuring with several types of PETs.

The tracking methodologies proposed in this work are designed specifically for tracking Bitcoins with illegal activities background. The parameter values of some methodologies are created based on our assumption of illegal Bitcoin transactions due to the lack of verifiable data. For example, the dirty-first strategy considers any transaction with clean coin mixing as an exiting transaction, and the evaluation metric criteria assume specific common address and transaction behaviours in illegal activities. Cryptocurrency forensics can employ context-based tracking methodologies when tracing known illegal Bitcoins to identify their movement and spending. However, it is possible to apply some of the context-based tracking methodologies with appropriate adjustments to cryptocurrency tracking and analysis in general, such as using address and transaction profiling to categorise cryptocurrency market activities.

Zero-taint Bitcoin Tracking (Chapter 4) In Chapter 4, our second contribution presents address taint analysis tracking that is capable of tracking zero-taint Bitcoins from several centralised mixer services that are untraceable by transaction taint analysis tracking. We propose improvements to the tracking methodology with a set of transaction behaviour criteria and address profiling that further reduce the number of false positive results. We also demonstrate that the address taint analysis tracking in combination with address clustering heuristics are capable of producing results with remarkably fewer false positive results compared to the method proposed in the previous research. The work in this chapter contributes to RO 4 by providing deciphering solutions for several well-known PET services and RO 2 by providing evidence of obscured Bitcoins' movement for forensic processes.

The address taint analysis method and its combination are most practical for tracking Bitcoins known to be obscured by centralised mixer services that utilise

similar mixing mechanisms as the ones examined in this work. There are two limitations to the application of the address taint analysis method as follows; first, the tracking process needs to be able to identify the deposit transactions and the mixer service involved before the method can be applied. Second, the strategy is created based on the assumption that centralised mixer services employ at least one central address group to mix Bitcoins. The method is also less applicable for mixer services that do not use any central address group, such as CoinJoin decentralised mixer services like Wasabi Wallet. Additionally, we set the filtering criteria parameters based on the assumption that the service use one specific withdrawn transaction pattern for mixing operation at the time of the sample transactions. Since the mixer services can update their mixing mechanism, the filtering criteria parameter may require adjustments for mixing transactions in different periods from the ones examined in this work.

Wasabi CoinJoin Transaction Detection (Chapter 5) In Chapter 5, our third contribution provides a detection method for CoinJoin transactions created by the Wasabi Wallet service using discovered transaction patterns. We performed a transaction analysis on various published transactions from different time periods and derived a set of general and period-specific criteria. Our experiment demonstrates that Wasabi CoinJoin transactions are identifiable only using the transaction information available in the blockchain. The results show a significantly high precision with a small number of false positive and practical applications for detecting tracking evasion attempts with Wasabi CoinJoin in Bitcoin theft incidents. The work in this chapter contributes to RO 4 by providing a novel and practical detection method for one of the most utilised PETs and RO 2 by providing evidence of Bitcoin obscuring for forensics. The detection method also serves as the crucial foundation for the future demixing methodologies to decipher Wasabi CoinJoin obscuring.

In practice, the detection method can be used in the tracking process of any Bitcoins and the investigation of the Wasabi service's mixing mechanism or scale of operation. As the method is capable of identifying all Wasabi CoinJoin transactions, it would be possible to calculate the profit and number of Bitcoins mixed by the

service from the beginning of its operation until the present time with high certainty. There is one limitation of the method in that the Wasabi service can adjust its mixing mechanism in the future, which can make the method in its current state impractical. Therefore, investigation for future Wasabi CoinJoin transactions should examine any potential change to the service's mixing mechanism that affects the transaction characteristics, similar to the transaction analysis experiment performed in this work, before applying the detection method.

Cryptocurrency Tracking Tool (Appendix B) Our fourth contribution in Appendix B introduces *TaintedTX*, which is an open-source library that is capable of performing targeted coins tracking with various tracking strategies and adapting to address ownership. The library additionally includes several utility functions that contribute to cryptocurrency transaction and address analysis, such as address clustering, address and transaction profiling, website scraping, and transaction behaviour analysis. The work in this chapter contributes to RO 5.

We design the *TaintedTX* library specifically for Bitcoin and other similar alternative cryptocurrencies tracking and analysis. The library can be used for cryptocurrency forensic analysis, such as illegal Bitcoin tracking, general Bitcoin spending analysis, and context profiling, as demonstrated throughout the thesis. The applicability of the library for alternative cryptocurrencies depends on the similarity in the transaction and address data structure and mechanism to the Bitcoin. The library should be fully applicable to Bitcoin, such as Bitcoin Cash¹ and Bitcoin Gold², but may require some modification for alternative cryptocurrencies with different mechanisms like Ethereum³. The library is likely to be inapplicable for privacy-oriented cryptocurrencies that obscure their blockchain data, like Zcash⁴, which hides transaction amounts and address identification from third parties.

¹<https://bitcoincash.org>

²<https://bitcoingold.org>

³<https://ethereum.org>

⁴<https://z.cash>

6.2 Future Work

In this section, we discuss potential research topics for future work that can provide improvements for cryptocurrency tracking and analysis methodologies.

6.2.1 Change of Ownership Detection

The address profile data we use in our current work is still incomplete and includes only addresses identified as belonging to cryptocurrency services and centralised mixers. The context-based tracking methodology presented in Chapter 3 would be unable to detect Bitcoin exchanges with unidentified addresses or non-service users. While it is possible to alleviate this limitation by gathering more address profile data from other external sources and using the transaction fingerprinting method described in Fleder et al. [62] research, it is still unlikely for the address profile data to be able to identify every address in the ever-growing blockchain.

As shown in the results of Chapter 3, the evaluation metrics reveal the positive potential for detecting distinct changes in transaction behaviours that can indicate changes in coins' ownership. Therefore, future work could investigate the application of transaction and address patterns as criteria to determine the possibility of tainted coins being possessed by different entities by analysing the characteristics of the addresses that receive tainted coins and subsequent transactions. If there is a significant change in address or transaction behaviours afterwards, there is a high possibility that the tainted coins are already transferred to other users. The address ownership classification metrics can also be applied to the filtering process for address taint analysis tracking to exclude false positive transactions unlikely to be part of the mixing operation.

6.2.2 Tainted Proportion Application

The prospect of applying a scoring or threshold system to identify potential illegal coins has been investigated in several previous studies. For example, both of the studies that develop taint analysis strategies [6, 128] discuss the application of taint analysis for a blacklisting system that indicates which Bitcoins are potentially illegal using taint proportion as a scoring method. However, there has been no study to our

knowledge that investigates the application of taint proportion either in percentage or value as a threshold for change of coin ownership detection method.

Based on our rationale for the Dirty-First strategy that transactions involving only tainted coins are most likely to be performed by targeted illegal users, it may be possible to adapt tainted coin proportion as an evaluation metric criterion for illegal coins tracking where a higher clean coin proportion equates to a higher possibility of PETs' involvement or transactions being performed by other entities and designate a threshold of minimum proportion (e.g., lower than 10%) to classify tainted coins as no longer in the hand of the original illegal users. The tainted coin proportion can also be implemented into the Haircut strategy as a new tracking strategy that tailors the tracking process from tracking coins with tainted coin proportion lower than a specified threshold level. The aim of this new strategy is to solve the Haircut strategy's drawback of producing a considerable number of tainted transactions by establishing a confidence level that the targeted coins are likely to still be in the hands of targeted users and should still be tracked.

6.2.3 Blockchain-wide Transaction Analysis

Our work in this thesis focused mainly on transaction tracking and analysis of illegal activities in cryptocurrency. It is possible to expand our cryptocurrency transaction analysis investigation to analyse general transactions during a specific time period using the evaluation metrics presented in Chapter 3 to observe and classify cryptocurrency users' behaviours for future work. Additionally, as the Dirty-First strategy can create a network of transactions performed by the same users, future work could investigate the potential of adapting the strategy as an address clustering heuristic that classifies addresses in transactions with fully tainted coins into the same entity group. It would be possible to apply the Dirty-First clustering heuristic to cluster all addresses in the blockchain for analysis of cryptocurrency users' behaviours.

6.2.4 Illegal Activities Detection

Another aspect of cryptocurrency forensic analysis that we have yet to explore is the detection methodology for identifying cryptocurrency transactions involving illegal

activities. We have demonstrated in Chapter 3 that cryptocurrency transactions involving illegal coins exhibit distinct behaviours from non-illegal coins transactions, even if they originate from transactions that share similar characteristics. Future work could investigate detection methodology for identifying illegal activities in cryptocurrency using the transaction behaviour data obtainable from the blockchain without requiring external information to detect illegal transactions similar to the PET transaction classifications we presented in this thesis.

6.2.5 PETs Reverse-Engineering Investigation

The scarcity of research into the internal working of obscuring mechanism of cryptocurrency PETs such as centralised mixer services and CoinJoin services is one of the prominent challenges for our work. It will be beneficial for all of our current and future work if we can launch an extensive investigation to study PET services that have never been examined in previous studies. The analysis of PET services would facilitate the creation of PET transaction classification methods that can assist the PET profiling and create opportunities for future work to formulate new demixing/unobscuring methodologies to track obscured illegal coins. The future PET services' reverse-engineering study will also open up an opportunity for investigations into the services' scale of operation and profit, which would assist estimate the services' role in cryptocurrency illegal activities. However, such investigation would require information from services' usage to test and analyse PET services due to service and transaction fees.

6.2.6 External Blockchain Data Investigation

As the cryptocurrency forensic analysis methodologies presented in this thesis make use of only confirmed transaction data that is available within the blockchain of the cryptocurrency we investigate (Bitcoin), there are several other data that we could utilise as evidence for forensic analysis and cryptocurrency transaction analysis. One example is transactions that are never confirmed into the blockchain either due to the block orphaning or rejection by miners. Future work could investigate transaction data in the Memory Pool to observe potential unconfirmed tainted transactions for more illegal Bitcoin activities.

One more cryptocurrency forensic analysis aspect that future work could examine is the investigation of tainted Bitcoins in the blockchain of the alternative Bitcoins that were forked from the main Bitcoin blockchain. As illegal Bitcoin users can claim stolen alternative Bitcoins that were still in their possession when the fork occurred, forensic analysis of tainted Bitcoins in alternative Bitcoin blockchain could potentially reveal more evidence of illegal users' activities that are unavailable in the main Bitcoin blockchain.

6.3 Concluding Remarks

This thesis examines the Bitcoin protocol and its privacy aspect, illegal activities that are facilitated by cryptocurrency, cryptocurrency tracking and deanonymisation methodologies proposed in the previous studies, and privacy techniques and PETs that are developed to counteract the proposed tracking and deanonymisation methods. Our work in this thesis introduces improvements to the cryptocurrency tracking methodologies and develops novel tracking and detection solutions for transaction obscuring with PETs. We also published an open-source library for cryptocurrency tracking and analysis that anyone can easily contribute to its development or adapt to their research work.

Throughout this thesis, we provide improvements and propose novel tracking methodologies that tackle the critical issues presented in the current tracking methodologies, whether due to their shortcomings or PETs. We consider our work to be a part of the ongoing contribution to the cryptocurrency forensic analysis effort to combat cybercrimes, which will have significant implications not only to cybersecurity and law enforcement but to financial regulatory developments of cryptocurrency in the future.

Appendix A

Additional Information for Bitcoin Internal System

A.1 Bitcoin Transactions

Pay-to-Pubkey-Hash Transactions P2PKH Bitcoin transactions contain the following Opcodes¹:

```
OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
```

The resulted transaction outputs contains the signature spending condition:

```
<sig> <pubKey>
```

Pay-to-Script-Hash Transactions P2SH Bitcoin transactions outputs contain the following Opcodes:

```
OP_HASH160 <Hash160(redeemScript)> OP_EQUALVERIFY
```

The P2SH Bitcoin transaction output's signature contain following serialised script:

```
<sig> <serialised script>
```

For a P2SH Bitcoin Transaction with multi-signature addresses, the transaction outputs would contain the following script:

¹Opcodes or operation codes are instruction or command codes that specify the operation to be performed

OP_m $\langle pubKey1 \rangle$... OP_n OP_CHECKMULTISIG

The required input script would have a structure as follows:

0 $\langle sig1 \rangle$... $\langle script \rangle$

Pay-to-Witness-Public-Key-Hash and Pay-to-Witness-Script-Hash Transactions P2WPKH and P2WSH Bitcoin transactions contain the following Op-codes:

0 $\langle witnessScriptHash \rangle$

A.2 Proof-of-work

Proof of work in Bitcoin requires miners to produce a double SHA-256 block hash of block headers and a Nonce (random value) that conforms to the hash value requirement (Target), as shown in the following equation:

$$\text{SHA256}\{\text{SHA256}\{\text{blockheader}|\text{Nonce}\}\} \leq \text{Target}$$

The target difficulty for the next Bitcoin block mining is calculated from the highest level of difficulty possible² that is set when the blockchain is first initiated (Max.Target), as shown in the following equation:

$$\text{Difficulty} = \text{Max.Target} / \text{Target}$$

A.3 Peer-to-Peer Network

Bitcoin relies on a Peer-to-peer overlay network that is connected via the Internet. Bitcoin network enables peer discovery and provides a communication channel to broadcast information between nodes. Every full node in the Bitcoin network is equal in terms of its control over the network, and there are no special or higher privilege nodes.

²The highest difficulty level in Bitcoin is at a difficulty of value 1.

Full nodes must first connect to the Bitcoin network and discover at least one other node when they start. This discovery is typically random (unless specified by users) and unrelated to the geographic location of other nodes. Starting nodes first establish a TCP connection on port 8333, which is the most common port used by Bitcoin network nodes. Once a connection is established, both Bitcoin nodes will exchange a version message containing the version number of Bitcoin protocol, current system time, port number, and blocks data. As shown in Figure A.1, both nodes acknowledge the version message by exchanging a *verack* message to confirm the connection. The starting node will subsequently send an *addr* message containing its IP address to relays connection information and a *getaddr* message for requesting IP address information of the other nodes on the network. The receiving node then forwards the *addr* message that contains the address IP information of other nodes to the starting node, which allows the starting node to connect with the other nodes [55].

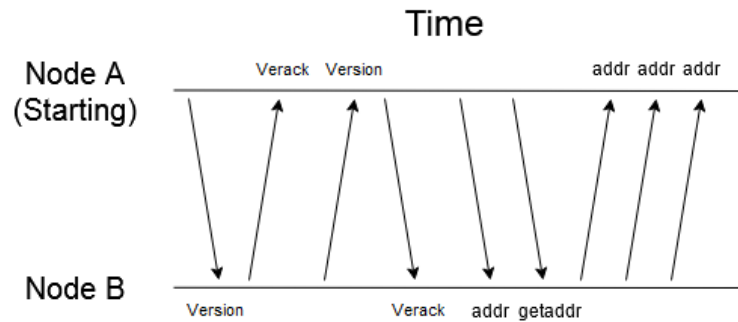


Figure A.1: Bitcoin peer connection and address propagation

Once full nodes are connected to the network, they must attempt to update their blockchain data to the current block height, either starting from the first genesis block for new nodes or updating from the latest block they have in the system. As the version message already contains the block height of the blockchain data of both nodes, the node with a higher block height (longer chain) starts by sending a *getblocks* message and exchanging the latest block hash with each other, as shown in Figure A.2. The higher block height node then sends an *inv* (inventory) message containing the hash of the blocks in the *getblocks* message. The node with lower block height subsequently identifies the blocks it misses using block hashes from the *inv* message. The lower block height node then sends a *getheaders* and *getdata*

message to request for the blockchain data. This process is also utilised when a new block is mined [167].

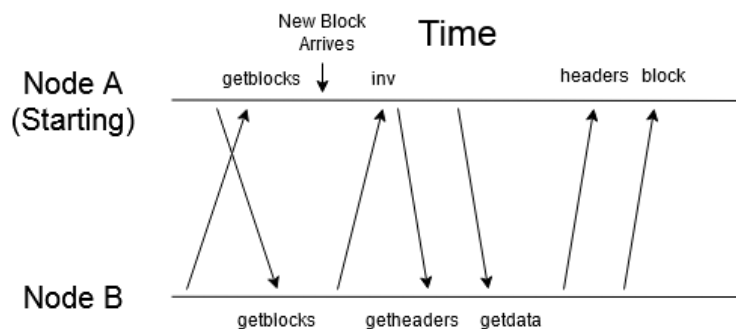


Figure A.2: Bitcoin blockchain propagation for full nodes

As mentioned in Section 2.2.5, lightweight nodes do not maintain blockchain data and only keep the block headers, lightweight nodes are instead required to periodically send a *getheaders* message to their parent full nodes to request for block header data to update their data to the latest block for the Simplified Payment Verification (SPV) process, as shown in Figure A.3.

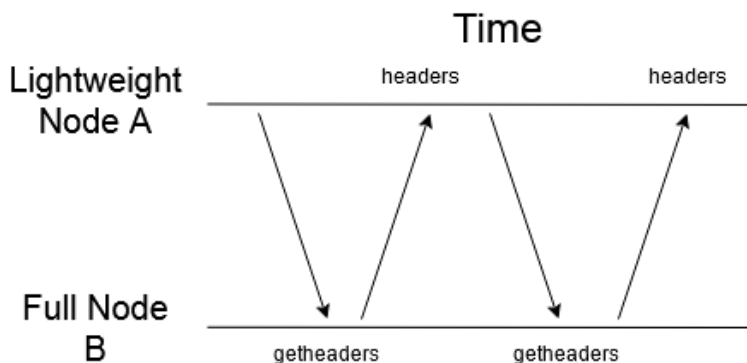


Figure A.3: Bitcoin block headers propagation for lightweight nodes

Appendix B

Cryptocurrency Tracking Tool

Cryptocurrency forensic analysis is a fascinating topic for the purpose of not only law enforcement but also research interests. Despite many interests to track cryptocurrency coins movement, even among its community users, there are few open-source cryptocurrency tracking and analysis libraries or software (discussed in Section 2.4.3) that are easy to use, access, and modify. We discuss the cryptocurrency tracking library that we develop in detail, from the system and data requirements to the library's functions in this chapter.

B.1 TaintedTX Overview

We developed a cryptocurrency tracking library named *TaintedTX* written in Python 3. We published the source code of the *TaintedTX* library on a public repository¹ under MIT license. The functions in the *TaintedTX* library are implementations of the methodologies we presented in the previous chapters. The source code is accompanied by the library's technical documentation, sample data of Bitcoin blockchain, and an online interactive notebooks binder of the repository.

The *TaintedTX* library offers users tracking and analysis functions for Bitcoin and other similar cryptocurrencies that are not available in any other published software as far as we know. The functions in the *TaintedTX* library are based on the methodologies presented in the previous chapters, which are cryptocurrency transaction tracking that is capable of adapting to address ownership and performing

¹<https://github.com/tintiron/taintedtx>

a variety of taint analysis strategies, address taint analysis for tracking (demixing) zero-taint mixed coins, and PET transaction detection. The *TaintedTX* library also includes other functions for cryptocurrency analysis, such as address clustering heuristics, website scraping for address profile data, and transaction analysis. The cryptocurrency tracking and analysis results from the *TaintedTX* library can be exported to external graph databases for visualising or analysis. Furthermore, the Python language’s ease of learning and use will allow future research and community users to support the development or incorporate the *TaintedTX* library to other newly developed tools.

The pseudo-code algorithm written in this chapter is partly based on a Python pseudo-code book guide [104]. We define the objects and functions used in the pseudo-code algorithm in Table B.1.

Table B.1: Pseudo-code objects and functions description

Column	Description
<i>ARRAY</i>	An array type object (e.g., <code>pandas data frame</code>)
<i>empty</i>	An indication for empty array
<i>None</i>	A null value
<i>READ</i>	Reads data file and returns the data
APPEND	Adds the array to the another array
DELETE	Deletes the item inside the array
COUNT	Returns the count of items inside the array
SUM	Returns the sum of items inside the array
HIGHEST	Returns the item with the highest value inside the array
MEDIAN	Returns the median value of the items inside the array
MEAN	Returns the mean value of the items inside the array
UNIQUE	Returns the items with unique value inside the array
ABSOLUTE	Returns non-negative value of the value
NEXT	Returns the next item inside the array
PREVIOUS	Returns the previous item inside the array
IS_MONOTONOUS	Checks whether the value of items in array is in monotonous order

B.1.1 System Requirements

The system requirements for the *TaintedTX* library are varied depending on the scale of the operation. The cryptocurrency tracking process in the *TaintedTX* library relies on the system RAM to store the transaction data required for the tracking and analysis processes. For example, a taint analysis process on the entire Bitcoin blockchain from the first block in 2009 to April 2021 require around 26 Gigabytes of RAM usage. The Python process can be automatically terminated mid-operation if

the memory usage exceeds the memory capacity. For reference, the system we used to perform the experiments in the previous chapters has the following specifications; 2.5 Gigahertz AMD Opteron(TM) Processor 6380 CPU with 512 Gigabytes of RAM. We designed the *TaintedTX* library based on Python version 3.6.8, but the library should be compatible with any Python version 3.

B.1.2 Libraries Requirements

As the *TaintedTX* library does not have an internal blockchain parser function, blockchain data must first be obtained with a blockchain parser software or external scripts. We used *BlockSci* version 0.7 to parse the cryptocurrency (Bitcoin) blockchain data into blockchain data files for the *TaintedTX* library implementation². We include the scripts that we used to export all of the blockchain data required for every function in the *TaintedTX* library from *BlockSci* version 0.7 in the published source code.

We select the Pandas data frame module³ for data management and implementation of the cryptocurrency tracking and analysis for the *TaintedTX* library due to its simplicity and important collection of built-in functions that are suitable for extensive data selection and manipulation operations. Pandas data frame also accepts diverse types of data files with the format that can be converted into a Pandas data frame. Hence, while we used *BlockSci* to obtain the blockchain data for our experiments, the *TaintedTX* library is compatible with the exported blockchain database from any other parser software as long as they have a similar data structure as described in Section B.2. The *TaintedTX* library's functions are written based on the Pandas data frame version 0.22.0.

B.2 Data Schemas

We describe the required data and its schema for the *TaintedTX* library's operation in this section which can be classified into three types; blockchain data (Section B.2.1), classification data (Section B.2.2), and utility data (Section B.2.3).

²See <https://citp.github.io/BlockSci/index.html> for the detailed documentation of the parsing process.

³<https://pandas.pydata.org>

B.2.1 Blockchain Data

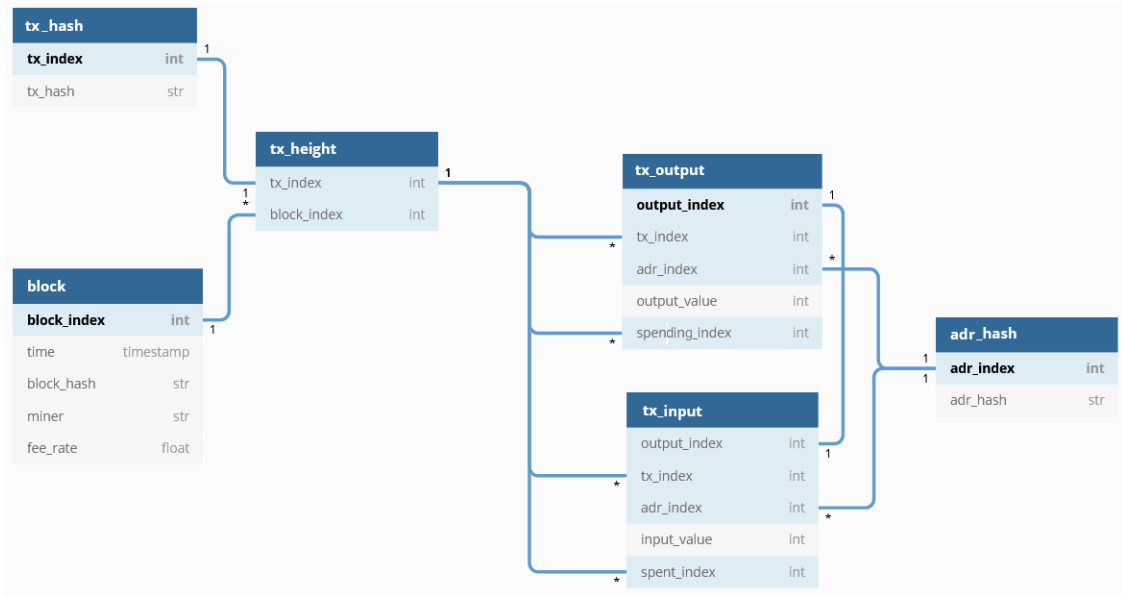


Figure B.1: Blockchain data schemas

The *TaintedTX* library requires a set or subset of blockchain transaction data for most of its functions. The blockchain data can be in the range of any date and time depending on users' specifications. The blockchain data is obtained from cryptocurrency blockchain with a parsing process and is static data that is not modified by the *TaintedTX* library's functions.

As shown in Figure B.1, there are six primary cryptocurrency blockchain data that the *TaintedTX* library requires to perform cryptocurrency tracking and transaction analysis operations as follows, which are block, transaction height, transaction hash, address hash, transaction output and transaction input data.

Table B.2: Block data dictionary

Column	Data Type	Description	Example
<i>block_index</i>	int	The index and height of the block	21,345
<i>time</i>	pandas.Timestamp	The timestamp of the block in Pandas datetime64 format	2011-01-01 01:11:01
<i>block_hash</i>	str	The hash of this block	000000000000somestuff 57b3exampleblock128557 abcd34d7o123if471
<i>miner</i>	str	The name of miner or mining pool that mined the block	Samplepool
<i>fee_rate</i>	int	The average transaction fee per byte value (in the smallest unit per byte) from every transaction in the block	621.083

Block data Block data (*Block*) contains information of blocks in the cryptocurrency blockchain data. The essential values in this data used by functions in the *TaintedTX* library are *block_index* and *time*, which are the main components for the data preparing function (see Section B.2.4). The *miner* value is used for miners' address profiling, and the *fee_rate* value is used for transaction analysis of average transaction fees paid by cryptocurrency users. The *block_hash* value serves as an additional reference for data searching and data integrity checking.

Table B.3: Transaction height data dictionary

Column	Data Type	Description	Example
<i>tx_index</i>	int	The internal index of the transaction	539,779
<i>block_index</i>	int	The internal index of the block the transaction is confirmed in	54,037

Transaction Height Data Transaction height data (*TX_height*) is a database index for linking transactions to their block to obtain the transaction confirmation timestamp in the data preparing function (Section B.2.4). Transaction height data is also required for the transaction behavioural analysis function (Section B.5.2) and control group finding function (Section B.5.3).

Table B.4: Transaction hash data dictionary

Column	Data Type	Description	Example
<i>tx_index</i>	int	The internal index of the transaction	33,836,486
<i>tx_hash</i>	str	The transaction hash of the transaction	45abcsomethingc4ddae 747sd1z8wpzc4ple821 6ae60809a22211201b823

Transaction Hash Data Transaction hash data (*TX_hash*) contains a transaction hash of each transaction. Transaction hash data is a crucial component serving as a database index to match the users' assigned transaction hash with the internal transaction index in the data preparing function. The transaction indexes can be used to perform tracking operations or search queries.

Address Hash Data Address hash data (*ADR_hash*) contains an address hash of each address. Address hash data is another crucial database index that serves as a reference for address searching and is used in the data preparing function for searching the internal address index of the assigned address hash.

Table B.5: Address hash data dictionary

Column	Data Type	Description	Example
<i>adr_index</i>	int	The internal index of the address, the last digit is the indicator of address format	11,434,885
<i>adr_hash</i>	str	The public key hash of the address	1ABCsomething dEq13412uAXwf 67z2nTe

Table B.6: Transaction output data dictionary

Column	Data Type	Description	Example
<i>output_index</i>	int	The internal index of the output	1,234,560
<i>tx_index</i>	int	The internal index of the output's transaction	211,408
<i>adr_index</i>	int	The internal index of the output's address	666,261
<i>output_value</i>	int	The value of the output in the smallest unit	1,000,000,000
<i>spending_index</i>	int	The internal index of the next transaction that spend the output	356,928

Transaction Output Data Transaction output data (*TX_output*) contains information of transaction outputs in each transaction. Transaction output data is one of the most fundamental data components required for most of the functions in the *TaintedTX* library. The order of transaction output must be in the same order as in the blockchain data for the taint analysis distribution process to operate correctly.

Table B.7: Transaction input data dictionary

Column	Data Type	Description	Example
<i>output_index</i>	int	The internal index of the output used to spend the input	1,234,550
<i>tx_index</i>	int	The internal index of the input's transaction	123,456
<i>adr_index</i>	int	The internal index of the output's address	123,457
<i>input_value</i>	int	The value of the input in the smallest unit	123,456,789
<i>spent_index</i>	int	The internal index of the previous transaction that the input was an output of	1,234,567

Transaction Input Data Transaction input data (*TX_input*) contains information of transaction inputs in each transaction. Transaction input data is an essential data component in the taint analysis distribution, transaction analysis, address

clustering functions and many other utility functions. The internal index value (*output_index*) for each transaction input is the same as its spending transaction output. Similar to the transaction output data, the transaction input order of this data must be in the same order as in the blockchain data for an accurate distribution of taint analysis strategies.

B.2.2 Classification Data

There are other two types of data that can only be obtained from external sources or with the classification functions in the *TaintedTX* library, which are identified data and transaction profiling data. The address and transaction classification data is dynamic auxiliary data that are created by the *TaintedTX* library’s classification functions using blockchain and external data. The classification data can be expanded with the development of new classification methods to include more profiling types. For example, the address profile data can incorporate entity data of identified Bitcoin users, and transaction profile data can include events associated with transactions (e.g., known theft transactions).

Table B.8: Identified address entity data dictionary

Column	Data Type	Description	Example
<i>adr_index</i>	int	The internal index of the address	12,347
<i>entity</i>	str	The name of the entity associates with the address	service.com
<i>type</i>	str	The type of the entity	exchange

Identified Address Profile Data Address profile or entity data can be categorised into two main classifications; service address and mixer service address, as described in Section 3.1.1. Address profile data contains entity information of cryptocurrency and mixer services potentially associated with the identified addresses, which are service name and service type. The detail of the address profile data gathering process that we utilised to obtain this data is described in Section 3.1. The web scraping function that we implement to retrieve address profile data of cryptocurrency services is described in Section B.5.5. The address clustering function that we utilise to perform multi-input address clustering heuristics on the scraped addresses is presented in Section B.5.1.

Table B.9: Identified PET transaction data dictionary

Column	Data Type	Description	Example
<i>tx_index</i>	int	The internal index of the transaction	123,456
<i>tx_hash</i>	str	The hash of the transaction	123abcdefghijklmnop qrs456tuv789xyz10
<i>type</i>	str	The type or service of the PET transaction	PETservice.com

PET Transaction Profile Data PET transaction profile data contain a list of transactions potentially involving a mixer service or the CoinJoin method performed by a CoinJoin service, such as JoinMarket, Wasabi Wallet and Samourai Whirlpool. The mixer transaction classification function is described in Section B.4.2, while the Wasabi Wallet and Samourai Whirlpool CoinJoin transaction classification function is described in Section B.4.1.

B.2.3 Utility Data

In addition to the data described in the previous sections, the transaction analysis function in the current implementation of the *TaintedTX* library (see Section B.5.2) requires additional transaction and address data to perform its operation as follows:

- Reused address data (*reused_ADR*), which contains a list of addresses that have more than one receiving transaction (reused). The transaction analysis function uses this data for the classification of reused addresses.
- Addresses' first transaction data (*fresh_ADR*), which contains a list of the first transaction of every address. The transaction analysis function uses this data to determine whether the addresses were fresh when they received tainted coins.
- Transaction fee data (*fee_TX*), there are three prominent data related to transaction fee used in several functions; transaction fee value, transaction fee rate per size ratio, and the daily average of transaction fee rate per size ratio.

The utility data described above is static auxiliary data generated from transaction data during the cryptocurrency blockchain parsing process and is not modified by the *TaintedTX* library's functions.

In addition to the utility data we described above, future work or updates could introduce new utility data for the implementation of new functions or features. One example is data containing the total number of transactions each address initiate,

which can be used for the classification of potential service addresses based on their transaction traffic.

B.2.4 Data Preparing

The *TaintedTX* library has a function named *prepare_data* that provides an automated data searching and preparing process based on a list of targeted addresses or transactions assigned in the *target_tx* or *target_adr* argument. The primary purpose of this function is to facilitate the data preparing process for other functions that require transaction data (*TX*).

As shown in Figure B.2, the *prepare_data* function performs a search query to find the internal index of the assigned list of transactions (*target_tx*) or addresses (*target_adr*). The function also has a time range limit argument (*time_range*) to search and retrieve data of transactions in a specific time frame, such as one year starting from the earliest assigned transaction or between the years 2010 and 2011. If the function's process can discover transaction data related to the assigned addresses or transactions, the function will return transaction data (*TX*) containing transaction input and output data of transactions within the specified time range as instance variables for further uses in the cryptocurrency tracking and analysis functions. The function also returns *results* data containing a list of transaction outputs related to the assigned transactions or addresses. The algorithm for this function is shown in Algorithm 1.

B.3 Taint Analysis Functions

As shown in Figure B.2, there are three main functions related to cryptocurrency tracking related to taint analysis in the *TaintedTX* library, which are taint analysis (Section B.3.1), taint analysis strategy (Section B.3.2), and address taint analysis (Section B.3.3).

B.3.1 Taint Analysis

The taint analysis or *tx_taint_search* function searches and builds a subset of transaction data (*tx_tainted*) consisting of transaction outputs that have a taint connection

Function *prepare_data(target_tx, target_adr, time_range)*:

Data: *target_tx* contains a list of transaction indexes that serves as a starting point of the returned transaction data, *target_adr* contains a list of address indexes to be used as a starting point, and *time_range* contains a string or list of a time range to limit the transaction data. The function reads and obtains data from the blockchain data (*TX_height*, *TX_hash*, *ADR_hash*, *TX_output*, *TX_input*) mentioned in Section B.2.

Result: *TX* data containing transaction inputs and transaction outputs in the time range limit stored as class variables. *results* data containing discovered transactions related to the *target_tx* or *target_adr*.

results \leftarrow empty ARRAY

if *target_tx* \neq None **then**

 | READ *target* \leftarrow {*tx* \in *TX_hash* | *tx.tx_hash* \in *target_tx*}

 | READ *tx* \leftarrow {*tx* \in *TX_output* | *tx.tx_index* \in *target.tx_index*}

else

 | **if** *target_adr* \neq None **then**

 | READ *target* \leftarrow {*adr* \in *ADR_hash* | *adr.adr_hash* \in *target_adr*}

 | READ *tx* \leftarrow {*tx* \in *TX_output* | *tx.adr_index* \in *target.adr_index*}

 | **end**

end

APPEND *results* \leftarrow *tx*

READ *block_index* \leftarrow *time_range* \in *Block.time*

READ *tx_index* \leftarrow *block_index* \in *TX_height.block_index*

TX \leftarrow {*tx* \in *TX_output* \cup *TX_input* | *tx* \in *tx_index*}

return *TX*, *results*

Algorithm 1: Data preparation

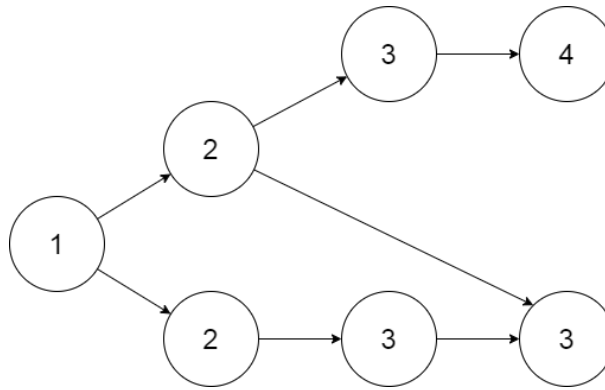


Figure B.3: Transaction depth example

Rectangles represent a transaction and the number inside represents its transaction depth from the starting transaction. Notes that the date and time of the transaction do not affect transaction depth.

with the assigned target transactions ($target_tx$). The purpose of this function is to optimise the distribution process for the taint analysis strategy function described in Section B.3.2 by filtering unrelated transaction data. The function will return $tx_tainted$ data, which has a similar data structure as the transaction output data but with an additional tainted coins value ($taint_value$) for the targeted transaction outputs and a $depth$ value that contains the transaction depth, which indicates the number of transactions that the tainted transactions occurred after the starting targeted transactions as illustrated in Figure B.3.

The tx_taint_search function operates by continuously searching for the subsequent transactions using the $spending_index$ value of the current searching transactions (tx_search) until the process reaches the end of the transaction data or the assigned $depth_limit$ value. The algorithm for this function is shown in Algorithm 2.

The tx_taint_search function also accepts an optional address profile data in the $stop_ADR$ argument to stop the taint analysis process from continuing to taint transaction outputs with an identified address (e.g., service and mixer addresses). The function will check and remove transaction outputs in the searching transactions (tx_search) that contain the addresses in the provided address profile data. This limiting process occurred after adding the discovered transaction outputs (tx_search) to the results ($tx_tainted$) and before beginning another search loop to find the subsequent transactions.

Function $tx_taint_search(TX, target_tx, depth_limit, stop_ADR)$:

Data: TX contains transaction inputs and transaction outputs, $target_tx$ contains a list of targeted transactions' indexes to perform the taint analysis on, $depth_limit$ is an integer value for limiting search loop, and $stop_ADR$ contains identified address entity data.

Result: $tx_tainted$ data containing transactions connected to $target_tx$ by taint analysis process.

```

 $tx\_search \leftarrow \{tx \in TX \mid tx.output \in target\_tx\}$ 
 $tx\_tainted \leftarrow empty\ ARRAY$ 
 $depth \leftarrow 0$ 
APPEND  $tx\_tainted \leftarrow tx\_search$ 
while  $LEN(tx\_search) > 0 \vee depth \neq depth\_limit$  do
  |  $tx\_search \leftarrow \{tx \in TX \mid tx.input \in tx\_search.output\}$ 
  |  $tx\_search \leftarrow \{tx \in tx\_search \mid tx \notin tx\_tainted\}$ 
  |  $tx\_search.depth = depth$ 
  | APPEND  $tx\_tainted \leftarrow tx\_search$ 
  | for  $tx \in tx\_search$  do
  | | if  $tx.output\_adr \in stop\_ADR$  then
  | | | DELETE  $tx\_search \leftarrow tx.output$ 
  | | end
  | end
  |  $depth = depth + 1$ 
end
return  $tx\_tainted$ 

```

Algorithm 2: Taint analysis

B.3.2 Taint Analysis Strategies

The taint analysis strategy functions perform tainted and clean coins distribution on the transactions in the assigned *tx_tainted*. The most notable feature of the taint analysis in the *TaintedTX* library is that it can perform different taint analysis strategies assigned in the *strategy* argument. The current implementation can perform the Poison, Haircut, Dirty-First and a combination of In-Out distribution strategies shown in Table B.10.

Table B.10: In-Out strategy variants

In Variant	Out Variant
First-In (FI)	First-Out (FO)
Last-In (LI)	Last-Out (LO)
Taint-In (TI)	Biggest/Highest-Out (HO or BO)
Clean-In (CI)	Smallest/Lowest-Out (SO)
Biggest/Highest-In (HI or BI)	
Smallest/Lowest-In (SI)	

The In-Out strategy can be in any variant combination of In and Out (e.g., FIHO for First-In-Highest-Out or LOSO Lowest-In-Smallest-Out).

In order to distribute tainted coins with a taint analysis strategy, there is one crucial issue that must first be addressed, which is the mining fee distribution. Similar to the issue mentioned in Section 2.4.1 that Bitcoin and other cryptocurrencies blockchain data do not contain any information that can indicate the exact distribution of coins in transaction inputs to transaction outputs, this issue also applies to the distribution of transaction inputs' coins to transaction fees. The mining fee payment issue can significantly affect the taint distribution result as it concerns how much the tainted coins will be subtracted to miners instead of output addresses.

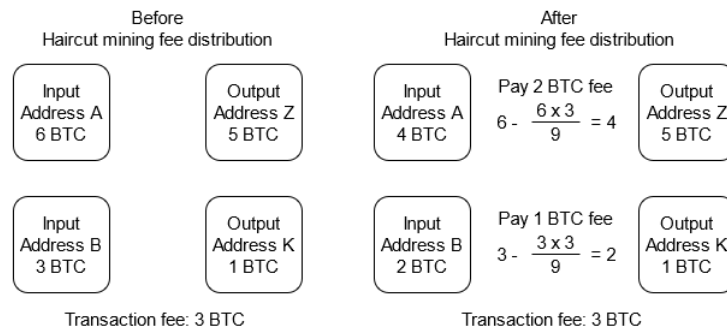


Figure B.4: Haircut transaction fee distribution

As mentioned in Section 2.2.6, the transaction fee is typically calculated based on the transaction data size, which primarily consists of every transaction input and

output in the transaction, it is reasonable to conclude that every transaction input contributes to the transaction fee. Ideally, the closest transaction fee distribution method to how transaction fee is typically calculated should be based on the data size proportion of the transaction inputs, where transaction inputs with a larger data size contribute more to the transaction fee. However, as the *BlockSci* parser does not keep track of the individual input data size but only the whole transaction data size, we lack the necessary data to implement this distribution method. Therefore, we implemented the closest alternative distribution method, which is similar to the Haircut taint analysis strategy where every transaction input contributes to the transaction fee according to their value proportion, as shown in Figure B.4. The algorithm for the Haircut transaction fee distribution is shown in Algorithm 3.

Function *haircut_fee*(*TX*):

```

Data: TX contains transaction inputs and transaction outputs.
Result: TX data with the transaction input value subtracted from the
           transaction fee.
for tx ∈ TX do
  for input ∈ tx do
     $fee = input.input\_value * tx.fee\_value / SUM(tx.input\_value)$ 
     $input.input\_value = input.input\_value - fee$ 
  end
end
return TX

```

Algorithm 3: Haircut transaction fee distribution

The taint analysis strategy functions start by performing a Haircut transaction fee distribution with the script described in Algorithm 3 and replace the transaction input value in the *TX* data with the fee subtracted value. Lastly, the function performs tainted coins distributions on the *tx_tainted* data using the assigned taint analysis strategy in the *strategy* argument.

The current implementation of the taint analysis strategy functions requires the tainted transaction data results (*tx_tainted*) from the *tx_taint_search* function that contains only tainted transactions to operate correctly for some taint analysis strategies, namely the Poison and Haircut strategy. The Dirty-First and In-Out strategies can be performed on the transaction data (*TX*) with an added taint value on targeted transaction outputs, albeit potentially with inferior operation performance due to the larger data size.

It is worth mentioning that the taint analysis implementation in the *TaintedTX* library is slightly different from the other implementations in that the distribution process does not differentiate the proportion of different specific coins. Our taint analysis implementation uses only two proportion classifications – tainted (proportion of the coins from the transactions assigned in the *tx_tainted* argument) or clean (proportion of unrelated coins with no tainted connection to the targeted transactions). Our taint classification also does not look into the past transactions as illustrated in Figure B.5. In the figure’s example, the taint analysis process will start tainting from Transaction 3 as provided in the *tx_tainted* argument and consider coins from Transactions 2 and 4 as clean coins, even if they originate from the same Transaction 1. The backward address taint analysis can be an exception to this scenario (see Section 4.1.2).

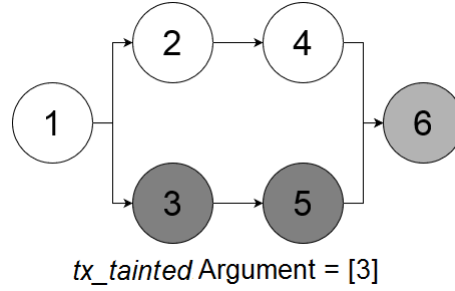


Figure B.5: Example of the *TaintedTX* library’s taint analysis operation
 White circles represent transactions with only clean coins, dark grey circles represent transactions with only tainted coins, and grey circles represent transactions with both tainted and clean coins. Black arrow lines indicate the transaction. The number inside represents the transaction index, and the number list at the *tx_tainted* argument represent transactions in the assigned data.

Poison Strategy The Poison strategy’s algorithm is the simplest of all taint analysis strategies. The process assigns a fully taint value to every tainted transaction output in the *tx_tainted* data, as shown in the Algorithm 4.

```

Function poison_strategy(tx_tainted):
  Data: tx_tainted from the tx_taint_search function.
  Result: tx_tainted data containing the results of tainted coins
              distribution according to the Poison strategy.
  tx_tainted.taint_value ← tx_tainted.output_value
  return tx_tainted
  
```

Algorithm 4: Poison strategy

Haircut Strategy The Haircut strategy’s algorithm performs a proportion distribution process on every transaction in the $tx_tainted$ data from the first to the last transaction according to the order of transaction index. The algorithm for the Haircut process is shown in Algorithm 5.

```

Function haircut_strategy( $tx\_tainted$ ):
  Data:  $tx\_tainted$  from the  $tx\_taint\_search$  function.
  Result:  $tx\_tainted$  data containing the results of tainted coins
            distribution according to the Haircut strategy.
  for  $tx \in tx\_tainted$  do
    if  $depth\_limit > 0$  then
       $input \leftarrow \{input \in tx\}$ 
       $tx.taint\_value = SUM(input.taint\_value)$ 
      for  $output \in tx$  do
         $output.taint\_value =$ 
           $tx.taint\_value * output.output\_value / tx.output\_value$ 
      end
       $tx.clean\_value = tx.output\_value - tx.taint\_value$ 
       $depth\_limit = depth\_limit - 1$ 
    end
  end
  return  $tx\_tainted$ 

```

Algorithm 5: Haircut strategy

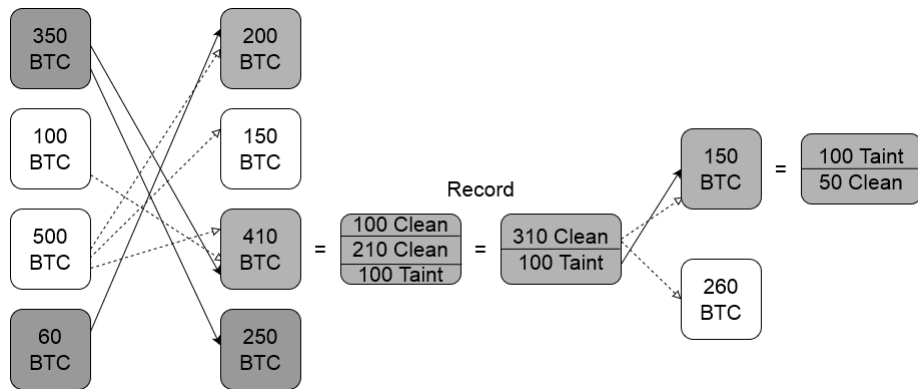


Figure B.6: An example of the LIFO strategy distribution and record. White rectangles represent clean inputs or outputs, dark grey rectangles represent fully tainted ones, and light grey rectangles represent partly tainted ones.

In-Out Strategies The In-Out taint analysis strategies algorithm starts by creating *record* data that contains the records of tainted and clean coin proportion order inside of each tainted transaction, as illustrated in Figure B.6. The distribution

Table B.11: *record* data dictionary

Column	Data Type	Description	Example
<i>tx_index</i>	int	The internal index of the transaction	123,456
<i>adr_index</i>	int	The internal index of the output's address	123,456
<i>spending_index</i>	int	The internal the of the next transaction that spend this output	1,234,567
<i>portion_value</i>	int	The value of the portion in the smallest unit	123,456,789
<i>total_amount</i>	int	The total value of the output in the smallest unit	123,456,789,100
<i>taint</i>	str	The indication whether the portion is either tainted (T) or clean (C)	C

process of the In-Out strategies calculates from both transaction order and distribution proportion order inside transaction inputs documented in the *record* data. The data structure of the *record* data is shown in Table B.11 and the algorithm for this function is shown in Algorithm 6.

The algorithm performs a search loop on transactions with tainted coins in the *tx_tainted* data according to the order of transaction index from the earliest to the latest. The process first retrieves transaction inputs data (*taint_input*) and transaction outputs data (*taint_output*) of the currently searching tainted transactions. Subsequently, the process matches the *taint_input* data with the *record* data to retrieve a proportion order reference of tainted and clean coins. Subsequently, the process sorts the *taint_input* data and the *taint_output* data according to the strategy assigned in the *strategy* argument and distribute the tainted and clean coins in transaction inputs to the transaction outputs accordingly. The algorithm will skip transactions with no tainted coins and transactions outside the *tx_tainted* data. The algorithm for the sorting function of each strategy variant (*variant_sorting*) is shown in Table B.12.

Dirty-First Strategy The Dirty-First strategy (*dirtyfirst_strategy*) function accepts two variants of the Dirty-First strategy; Dirty-First and Pure Dirty-First. Both variants keep tracking fully tainted outputs and stop when they are used in a transaction with any transaction input that contains clean coins. The difference between the two variants is that the Dirty-First variant will keep the transaction clean coins mixing transactions in the *tx_tainted* data, while the Pure Dirty-First will discard them, as illustrated in Figure B.7.

Function *in_out_strategy(tx_tainted, strategy)*:

Data: *tx_tainted* from the *tx_taint_search* function and *strategy* is a string value of a valid in-out strategy name to perform tainted coins distribution.

Result: *tx_tainted* data containing the results of tainted coins distribution according to the selected In-Out strategy variant, *record* data containing records of tainted and clean coin proportion order inside of each tainted transaction.

record $\leftarrow \{tx \in tx_tainted \mid tx.taint_value > 0\}$

for *tx* $\in tx_tainted$ **do**

if *depth_limit* > 0 **then**

taint_input $\leftarrow \{tx \in record\}$

taint_output $\leftarrow \{output \in tx.output\}$

taint_input, taint_output \leftarrow

variant_sorting(taint_input, taint_output, strategy)

\triangleright See Table B.12

for *input* $\in taint_input$ **do**

for *portion* $\in input$ **do**

for *output* $\in taint_output$ **do**

addvalue = *portion.portion_value* –
 ABSOLUTE(*output.output_value* –
 portion.portion_value)

 APPEND *record* $\leftarrow addvalue$

portion.portion_value =

portion.portion_value – *addvalue*

if *portion.portion_value* = 0 **then**

 | break

end

end

end

end

tx.taint_value = SUM(*record.taint_value*)

tx.clean_value = *tx.output_value* – *tx.taint_value*

depth_limit = *depth_limit* – 1

end

end

return *tx_tainted, record*

Algorithm 6: In-Out strategy

Table B.12: In-Out sorting (*variant_sorting*)

Strategy	Input Sorting Algorithm	Output Sorting Algorithm
First	Pass (does not need sorting)	Pass (does not need sorting)
Last	REVERSE(<i>taint_input</i>)	REVERSE(<i>taint_output</i>)
Taint	<i>taint_input.taint_value</i>	Pass (no sorting method)
Clean	REVERSE(<i>taint_input.taint_value</i>)	Pass (no sorting method)
Highest	<i>taint_input.input_value</i>	<i>taint_output.output_value</i>
Lowest	REVERSE(<i>taint_input.input_value</i>)	REVERSE(<i>taint_output.output_value</i>)

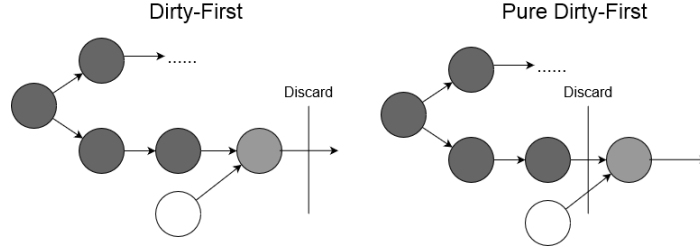


Figure B.7: Difference between Dirty-First and Pure Dirty-First

White circles represent fully clean coins transaction outputs, black circles represent fully tainted coins transaction outputs, and grey circles represent partly tainted coins transaction outputs. Black arrow lines indicate transactions that are being tracked, and three dots represent further tracking.

The clean coin mixing transactions (exit transactions) can be the indication of when the targeted coins change ownership, which can be useful for the analysis of coins spending and obscuring strategies. However, the clean coin mixing transactions can affect the targeted users' transaction behaviour analysis since their transaction characteristics can be significantly different from other non-spending transactions (e.g., distribution transactions).

The Dirty-First strategy process keeps searching the subsequent spending transactions of tainted transactions with conditional scripts that detect the presence of clean coins and remove the subsequent transactions with clean coins from further search. The algorithm for the Dirty-First process is shown in Algorithm 7.

087We also provide a function named *dirtyfirst_convert* that converts any *tx_tainted* data from the other taint analysis strategies except for the Poison strategy results (due to the lack of clean coin value) to either of the Dirty-First or Pure Dirty-First strategy results. The *dirtyfirst_convert* function operates by first separating transactions with clean coins in the provided *tx_tainted* data. The process then performs a continuous search loop on the *tx_tainted* data to remove the remaining transactions that occur after transactions with clean Bitcoins. The algorithm for this function is shown in Algorithm 8.

Function *dirtyfirst_strategy(tx_tainted)*:

Data: *tx_tainted* from the *tx_taint_search* function.

Result: *tx_tainted* data containing the results of tainted coins distribution according to the selected Dirty-First strategy variant.

```

for tx ∈ tx_tainted do
  | if tx.clean_value = 0 ∧ depth_limit > 0 then
  | | for spent_tx ∈ tx do
  | | | spent_tx.taint_value = tx.taint_value
  | | end
  | end
end
if strategy = puredirtyfirst then
  | tx_tainted ← {tx ∈ tx_tainted | tx.clean_value > 0}
end
return tx_tainted

```

Algorithm 7: Dirty-First strategy

Function *dirtyfirst_convert(strategy, tx_tainted)*:

Data: *strategy* is a string value of either *dirtyfirst* or *puredirtyfirst* for selecting the strategy, *tx_tainted* data from the of another taint analysis strategy.

Result: *tx_tainted* data containing the taint analysis results of either Dirty-First or Pure Dirty-First strategy.

```

tx_clean ← {tx ∈ tx_tainted | tx.clean_value > 0}
tx_tainted ← {tx ∈ tx_tainted | tx ∉ tx_clean}
tx_search ← {tx ∈ tx_tainted | tx.input ∉ tx_tainted.output}
tx_search ← tx_search.depth ≠ 0
while LEN(tx_search) > 0 do
  | tx_tainted ← {tx ∈ tx_tainted | tx ∉ tx_search}
  | tx_search ← {tx ∈ tx_tainted | tx.input ∉ tx_tainted.output}
  | tx_search ← tx_search.depth ≠ 0
end
if strategy = dirtyfirst then
  | APPEND tx_tainted ←
  | {tx ∈ tx_clean | tx.input ∈ tx_tainted.output}
end
return tx_tainted

```

Algorithm 8: Dirty-First conversion

B.3.3 Address Taint Analysis

The address taint analysis or *adr_taint_search* function performs address taint analysis tracking described in Chapter 4 on a list of targeted addresses assigned in the *target_adr* argument. As the purpose of address taint analysis is to track the address connection and not the distribution of specific coins, the *adr_taint_search* function will return *tx_tainted* data that does not include the *taint_value* value. In addition to the *tx_tainted* data, the function also returns tainted address data (*adr_tainted*), which contains a list of all tainted addresses that are connected to the targeted addresses.

The *adr_taint_search* function operates by consecutively searching for every transaction (*tx_search*) sent by tainted addresses (*adr_search*) and subsequently finding all addresses that receive coins from the tainted addresses to perform another search loop. The search loop ends when the process discovers every existing address connection of tainted addresses or reaches the assigned *depth_limit* value. The algorithm for this function is shown in Algorithm 9.

The *adr_taint_search* function also includes a *backward* argument to perform either onward or backward address taint analysis. The backward address taint analysis process performs address taint analysis on the assigned targeted addresses in the *target_adr* argument backwards to their receiving transactions instead of the usual sending transactions. Essentially, the process continuously searches for every address that sends coins to tainted addresses to perform another search loop until it discovers every possible connection or reaches the assigned *depth_limit* value. Similar to the *tx_taint_search* function, the *adr_taint_search* function also has an optional *stop_ADR* argument that tailors the searching process to stop tainting on an excluded address.

Withdrawn Transaction Filtering The withdrawn transaction filtering or *filtering* function is the additional process of address taint analysis tracking to reduce the number of potential false positive transaction outputs. The primary function of this function is that it filters out every transaction (*tx*) in the *tx_tainted* data with transaction patterns that do not match the filtering criteria provided in the arguments. The function currently has seven filtering optional arguments, namely,

Function *adr_taint_search*(*TX*, *target_adr*, *depth_limit*, *stop_ADR*, *backward*):

Data: *TX* contains transaction inputs and transaction outputs, *target_adr* contains a list of starting transaction indexes to perform taint analysis process on, *depth_limit* integer value for limiting search loop, and *stop_ADR* contains identified address entity data. The *backward* argument indicates whether the address taint analysis uses an onward or backwards tainting process.

Result: *tx_tainted* data containing transactions of tainted addresses connected to *target_adr* by address taint analysis and *adr_tainted* data containing tainted addresses.

tx_tainted \leftarrow empty ARRAY

adr_tainted \leftarrow empty ARRAY

while LEN(*N*) > 0 \vee *depth_limit* \neq 0 **do**

if *backward* = *False* **then**

 | *tx_search* \leftarrow *target_adr*{*tx* \in *TX* | *tx.output_adr* \in *target_adr*}

else

if *backward* = *True* **then**

 | *tx_search* \leftarrow *target_adr*{*tx* \in *TX* | *tx.input_adr* \in *target_adr*}

end

end

adr_search \leftarrow {*adr* \in *target_adr*}

if *stop_ADR* \neq *None* **then**

for *tx* \in *tx_search* **do**

if *tx.output_adr* \in *stop_ADR* **then**

 | DELETE *tx_search* \leftarrow *tx.output*

end

end

end

 APPEND *tx_tainted* \leftarrow *tx_search*

 APPEND *adr_tainted* \leftarrow *adr_search*

depth_limit = *depth_limit* - 1

end

return *tx_tainted*, *adr_tainted*

Algorithm 9: Address taint analysis

mixing time (mix_time), transaction input number ($filter_input$), transaction output number ($filter_output$), continuous transaction chain ($filter_chain$), reused input address ($filter_reuse$), fixed or percentage mixing fee ($filter_mixfee$) and transaction fee ($filter_txfee$). The filtering algorithm for this function is shown in Table B.13.

Table B.13: Withdrawn transaction filtering

Filtering Criteria	Filtering Algorithm
Mixing Time	$tx.time \geq deposit_tx.time \wedge tx.time \leq deposit_tx.time + mix_time$
Transaction Input Number	$COUNT(tx.input) = filter_input$
Transaction Output Number	$COUNT(tx.output) = filter_output$
Continuous Transaction Chain	$(COUNT(PREVIOUS(tx).input) = filter_input \wedge$ $COUNT(PREVIOUS(tx).output) = filter_output) \vee$ $(COUNT(NEXT(tx).input) = filter_input \wedge$ $COUNT(NEXT(tx).output) = filter_output)$
Reused Input Address	$tx.input_adr \notin filter_reuse$
Fixed Mixing Fee	$\exists v \in tx.output_value, v \leq filter_mixfee$
Percent Mixing Fee	$\exists v \in tx.output_value, v \leq tx.output_value -$ $(tx.output_value * filter_mixfee)$
Transaction Fee	$tx.fee_value = filter_txfee$

B.4 Classification Functions

The *TaintedTX* library includes transaction classification for three types of PETs, which are CoinJoin (Section B.4.1), Mixer service (Section B.4.2), and Lightning Network (Section B.4.3).

B.4.1 CoinJoin Transaction Classification

The *TaintedTX* library offers functions that classify CoinJoin transactions created by two well-known CoinJoin mixing services, Wasabi Wallet and Samurai Wallet Whirlpool. The Wasabi CoinJoin transaction detection algorithm is based on the method described in Section 5.3.2 where we use transaction pattern general criteria and period-specific criteria to classify potential Wasabi CoinJoin transactions. The algorithm is shown in Algorithm 10.

The Wasabi CoinJoin transaction detection algorithm accepts five period-specific criteria in the arguments, which are Transaction Value ($P1$), transaction outputs number ($P2$), transaction inputs number/ratio ($P3$), denomination outputs number/ratio ($P4$), and change outputs number/ratio ($P5$). The algorithm also accepts

Function *wasabi_tx_detect*(*TX*, *heuristic*, *Criteria P1, P2, P3, P4, P5*):

Data: *TX* contains transaction inputs and transaction outputs and *heuristic* is an argument for selecting the heuristic with the string value of either **value** or **ratio** that indicates how the period-specific criteria will be applied. Period-specific criterion (*P1, P2, P3, P4*, and *P5*) is a number value.

Result: *wasabi_tx* data containing transactions classified as a Wasabi CoinJoin transaction.

```

wasabi_tx ← {tx ∈ wasabi_tx | tx.input_adr.type = P2WPKH
  ∧ tx.output_adr.type = P2WPKH }
for tx ∈ wasabi_tx do
  anonymity_set ← {output ∈ tx | COUNT(output.output_value ∈
    tx) ≥ 2}
  for output ∈ anonymity_set do
    if output.output_value < 0.95 ∨ (NEXT(output).output_value ≠
      output.output_value ∧ PREVIOUS(output).output_value ≠
      output.output_value) then
      DELETE output ← anonymity_set
    end
  end
  wasabi_tx ← {tx ∈ TX | COUNT(tx.input) <
    COUNT(tx.output_adr) ∧ COUNT(tx.input) ≥
    HIGHEST(COUNT(tx.denomination_output))}
  if anonymity_set.output_value.IS_MONOTONOUS = False then
    DELETE wasabi_tx ← tx
  end
end
wasabi_tx ← {tx ∈ wasabi_tx | SUM(tx.output_value) ≥ Criteria P1}
wasabi_tx ← {tx ∈ wasabi_tx | COUNT(tx.output) ≥ Criteria P2}
if heuristic = value then
  wasabi_tx ← {tx ∈ wasabi_tx | COUNT(tx.input) ≥ Criteria P3}
  wasabi_tx ← {tx ∈ wasabi_tx | COUNT(tx.denomination_output) ≥
    Criteria P4}
  wasabi_tx ← {tx ∈ wasabi_tx |
    COUNT(UNIQUE(tx.output_value)) ≥ Criteria P5}
else
  if heuristic = ratio then
    wasabi_tx ← {tx ∈ wasabi_tx |
      COUNT(tx.input) * 100 / COUNT(tx.output) ≥ Criteria P3}
    wasabi_tx ← {tx ∈ wasabi_tx |
      COUNT(tx.denomination_output) * 100 / COUNT(tx.output) ≥
      Criteria P4}
    wasabi_tx ← {tx ∈ wasabi_tx |
      COUNT(UNIQUE(tx.output_value)) *
      100 / COUNT(tx.output) ≥ Criteria P5}
  end
end
return wasabi_tx

```

Algorithm 10: Wasabi's CoinJoin transactions pattern-based detection

two types of application for period-specific criteria 2, 4, and 5 in the *heuristic* argument, which are **value** and **ratio** options. The **value** option will apply the assigned criteria value as the minimum fixed value (e.g., Wasabi CoinJoin transactions must have more than five denomination outputs). The **ratio** option will apply the assigned criteria value as the minimum ratio to transaction outputs (e.g., transaction outputs in Wasabi CoinJoin transaction must make up of at least 50% of denomination outputs).

Samourai Whirlpool CoinJoin transaction detection function uses static transaction behaviours as described in Section 3.1.1.2 The algorithm for this function is shown in Algorithm 11.

```

Function samourai_tx_detect(TX):
  Data: TX contains transaction inputs and transaction outputs.
  Result: samourai_tx data containing transactions classified as a
    Samourai Whirlpool CoinJoin transaction.
  samourai_tx  $\leftarrow$  empty ARRAY
  samourai_tx  $\leftarrow$  {tx  $\in$  TX | COUNT(tx.input) =
    5  $\wedge$  COUNT(tx.output) = 5}
  for tx  $\in$  samourai_tx do
    | join_value  $\leftarrow$  tx.output_value[0]
    | if  $\exists v_o \in tx.output\_value, v_o \neq join\_value \vee v_o \notin$ 
    | [1, 000, 000; 5, 000, 000; 50, 000, 000]  $\vee \exists v_i \in tx.input\_value, v_i <$ 
    | join_value then
    | | DELETE samourai_tx  $\leftarrow$  tx
    | end
  end
  return samourai_tx

```

Algorithm 11: Samourai Whirlpool CoinJoin transaction classification

B.4.2 Mixer Transaction Classification

We include the *mixer_find* function that classifies transactions potentially performed by a mixer service. The function operates by matching the transactions in the *TX* data with the transaction behaviour criteria of specific mixer services. The current implementation of this function includes the classification for ChipMixer service described in Section 3.1.1.2 where we use static transaction behaviours to detect the mixer's transactions.

As studies into cryptocurrency mixer services are still scarce, and only a few mixer services have been analysed, the *mixer_find* function serves as a groundwork

Function *mixer_find*(*TX*):

```

Data: TX contains transaction inputs and transaction outputs.
Result: mixer_tx data containing transactions classified as a mixer
           service transaction.
mixer_tx  $\leftarrow$  empty ARRAY
chipmixer_tx  $\leftarrow$  {tx  $\in$  TX | COUNT(tx.output) > 1}
for tx  $\in$  chipmixer_tx do
    join_value  $\leftarrow$  MEDIAN(tx.output_value)
    if COUNT(tx.output_value  $\neq$  join_value) > 1  $\vee$   $\exists t \in$ 
      tx, COUNT(t.output_value < 1,000,000) >
      1  $\vee$  COUNT(t.output_value > 4.096,000,000) >
      1  $\vee$  COUNT(tx.output_value % 1,000,000  $\neq$  0) > 1 then
      | DELETE chipmixer_tx  $\leftarrow$  tx
    end
end
APPEND mixer_tx  $\leftarrow$  chipmixer_tx
return mixer_tx

```

Algorithm 12: Mixer transaction classification

for further development of mixer service transaction detection. The function can become significantly beneficial to cryptocurrency forensic analysis efforts when future studies can reveal transaction behaviours for other mixer services that can be incorporated into the function's classification process.

B.4.3 Lightning Network Transaction Classification

The *TaintedTX* library includes a function named *lightning_find* that detect and classify transactions in the *TX* data as potential Lightning Network transactions using the characteristic of funding (*fund_tx*) and closing (*close_tx*) transactions described in Section 2.5.1. The function also has an optional *svc_ADR* argument for assigning service address data to exclude transactions with cryptocurrency service addresses unrelated to the Lightning Network for both transaction inputs and transaction outputs to reduce the number of potential false positive results.

By default, the process uses the limit of the official Bitcoin Lightning Network protocol with the maximum transaction output value limit of 0.042 BTC (4.2 million satoshis) for transactions that occurred from its alpha release in 2017 to 2020-04-29 and 0.167 BTC (16.7 million satoshis) for transactions that occurred from the date 2020-04-30 forward. The maximum value limit of channel capacity can be specified in the optional *value_limit* argument to bypass the default limit filtering. It is

also worth noting that the algorithm can classify only Lightning Network channel transactions with a closing transaction. The algorithm for this process is shown in Algorithm 13.

Function *lightning_find*(*TX*, *svc_ADR*, *value_limit*):

Data: *TX* contains transaction inputs and transaction outputs, *svc_ADR* contains service address entity data, and *value_limit* is an integer value to limit the transaction output value of *fund_tx*. The function also uses a global variable *lncapup_tx_index* for indicating the first transaction after the network 0.167 BTC capacity update.

Result: *lightning_tx* data containing transactions classified as a Lightning Network transaction.

```

fund_tx ← tx ∈ TX
if value_limit = None then
  | fund_tx ← {tx ∈ fund_tx | tx.output_value ≤ 16,700,000 ∧
  |   (tx.output_value ≤ 4,200,000 ∧ tx.tx_index < lncapup_tx_index)}
else
  | if value_limit ≠ None then
  | | fund_tx ← {tx ∈ fund_tx | tx.output_value ≤ value_limit}
  | end
end
fund_tx ← {∃v ∈ fund_tx.output_adr, v.type = P2WSH }
fund_tx ← {tx ∈ fund_tx | tx.input_adr ∉ svc_ADR ∧ tx.output_adr ∉
  svc_ADR}
close_tx ← {tx ∈ TX | tx.input ∈ fund_tx.output}
close_tx ← {tx ∈ close_tx | COUNT(tx.input) =
  1 ∧ COUNT(tx.output) ≤ 2}
fund_tx ← {tx ∈ fund_tx | tx.output ∈ close_tx.input}
for tx ∈ fund_tx do
  | APPEND lightning_tx ← tx
end
return lightning_tx

```

Algorithm 13: Lightning Network transaction classification

B.5 Utility Functions

The *TaintedTX* library includes several utility functions that can obtain internal and external information to assist in cryptocurrency tracking and analysis, which are address clustering (Section B.5.1), transaction analysis (Section B.5.2), transaction control group finding (Section B.5.3), Neo4J data exporting (Section B.5.4), and website scraping (Section B.5.5).

B.5.1 Address Clustering

While *BlockSci* provides a built-in automatic address clustering function for the multi-input address clustering heuristic, it has a severe limitation in that it requires an enormous disk space size to perform a clustering operation. Inefficient disk space will cause *BlockSci* to produce corrupted results, which is the issue that we encounter on our system. Therefore, we implement an address clustering function (*adr_clustering*) for the *TaintedTX* library that can perform the multi-input and multi-output address clustering heuristics on the provided list of targeted addresses.

The *adr_clustering* function operates on transaction data (*TX*) to search for addresses that share either transaction inputs or outputs with clustered addresses. The function is more suitable for clustering a targeted set of addresses (*target_adr*) rather than every address in the blockchain, such as the service address profile data gathering performed in Chapter 3 and mixed coins tracking demonstrated in Chapter 4.

The function also includes an optional *exclude* argument to provide a list of addresses to exclude from the clustering process and an optional *CoinJoin_TX* argument to exclude addresses in CoinJoin transactions from being clustered. The algorithm for this function is shown in Algorithm 14.

Furthermore, the address clustering function in the *TaintedTX* keeps a record of the level or depth of the clustered addresses. Essentially, the process designates targeted addresses assigned in the *target_adr* argument with a depth of zero and any address that shares transaction inputs with depth zero addresses with a depth of one. Subsequently, any address that share transaction inputs with depth one addresses but not depth zero are designated a depth of two and so forth. The depth level value serves as the indicator of the closeness of the clustered addresses to the targeted addresses. The maximum depth level can be specified in the *depth_limit* argument to limit the clustering loop.

B.5.2 Transaction Analysis

The *TaintedTX* library includes a function named *tx_analysis* that performs transaction analysis on the provided transactions in the *tx_tainted* data and returns the transaction behaviour data that we used as criteria for the evaluation metrics de-

Function *adr_clustering*(*TX*, *target_adr*, *depth_limit*, *CoinJoin_TX*, *heuristic*, *exclude*):

Data: *TX* contains transaction inputs and transaction outputs, *target_adr* contains a list of addresses to perform address clustering, *depth_limit* is an integer value for limiting clustering loop, and *CoinJoin_TX* contains a list of transactions identified as CoinJoin transactions. *heuristic* is an argument for selecting the clustering heuristic with the string value of either *multi-input* or *multi-output*. *exclude* contains a list of addresses to exclude from the clustering.

Result: *results* data containing addresses in the same cluster as the addresses in *target_adr*.

adr_search \leftarrow *target_adr*

results \leftarrow empty ARRAY

depth \leftarrow 0

while COUNT(*adr_search*) > 0 \vee *depth* \neq *depth_limit* **do**

if *heuristic* = *multi-input* **then**

if COUNT(*tx.input*) > 1 **then**

 | *tx_search* \leftarrow {*tx* \in *adr_search* | *tx.input_adr* \in *TX.input*}

end

else

if *heuristic* = *multi-output* **then**

if COUNT(*tx.output*) > 1 **then**

 | *tx_search* \leftarrow {*tx* \in *adr_search* | *tx.output_adr* \in *TX.output*}

end

end

end

tx_search \leftarrow {*tx* \in *tx_search* | *tx.adr* \notin *exclude*}

tx_search \leftarrow {*tx* \in *tx_search* | *tx* \notin *CoinJoin_TX*}

adr_search \leftarrow {*adr* \in *tx_search*}

 APPEND *results* \leftarrow *adr_search*

depth = *depth* + 1

end

return *results*

Algorithm 14: Address clustering

scribed in Section 3.1.4. The function utilises the transaction data (*tx_tainted*) and external utility data mentioned in Section B.2.3 to obtain the following information; transaction frequency (per day), number of reused addresses, number of fresh addresses, transaction fee, number of addresses per transaction, number of identified service addresses, number of identified mixer addresses, identified PET (CoinJoin and Lightning Network) transaction classification, and potential PET transaction classification. The function returns two data; the *analysis* data containing the transaction behaviour for each transaction in the *tx_tainted* data and the *evaluation* data with the average statistic for every transaction in the *tx_tainted* data. The algorithm for this function is shown in Algorithm 15.

The potential PET transaction classification in this function utilises the classification method described in Section 3.1.1.2. The algorithm for the potential PET transaction classification function (*PET_tx_find*) is shown in Algorithm 16.

B.5.3 Control Groups Finding

We provide a function named *control_find* that finds transaction control groups with similar transaction characteristics as the provided transactions in the argument and return data containing a list of control group transactions. The function uses a filtering algorithm that is similar to Table B.13 where it filters the transactions based on a set of assigned criteria. The function includes four arguments used as the control group selection criteria as described in Section 3.2.2. The criteria arguments are as follows; the number of transaction inputs, the number of outputs, transaction value range in percentage (e.g., 10% equates to value range of 90 to 110 BTC for a targeted transaction with 100 BTC), and time range (e.g., transactions that occurred within ten days before and after the targeted transaction).

B.5.4 Export for Neo4J

We provide a script in the *TaintedTX* library for exporting taint analysis results to Neo4J graph database⁴ version 3.4.4 for graph visualisation. The main feature of Neo4j is that it can create and display a network graph that can be interacted with in real-time. However, Neo4j has a limitation in that its performance gradually

⁴<https://neo4j.com>

Function $tx_analysis(tx_tainted, reused_ADR, fresh_ADR, fee_TX, svc_ADR, mixer_ADR, CoinJoin_TX, lightning_TX)$:

Data: $tx_tainted$ contains targeted transactions for analysis, $reused_ADR$ contains a list of reused addresses, $fresh_ADR$ contains a list of addresses' first transaction, svc_ADR contains service address entity data, $mixer_ADR$ contains mixer address entity data, $CoinJoin_TX$ contains a list of transactions identified as CoinJoin transactions, and $lightning_TX$ contains a list of identified Lightning Network transactions.

Result: $analysis$ data containing the transaction behaviour for each transaction and $evaluation$ data containing the average statistic of all targeted transactions.

```

analysis ← tx_tainted.tx
adr_count ← empty ARRAY
frequency ← MEAN(COUNT(tx_tainted.tx)/day)
analysis.reused_adr ← {COUNT(adr) ∈ analysis | adr ∈ reused_ADR}
analysis.fresh_adr ← {COUNT(adr) ∈ analysis | adr ∈
  fresh_ADR ∧ adr.tx ∈ fresh_ADR}
analysis.fee_rate ← {tx.fee_rate ∈ analysis | tx ∈ fee_TX}
for tx ∈ analysis do
  | APPEND adr_count ←
  |   COUNT(tx.input_adr) + COUNT(tx.output_adr)
end
analysis.svc_adr ← {adr ∈ analysis | adr ∈ svc_ADR}
analysis.mixer_adr ← {adr ∈ analysis | adr ∈ mixer_ADR}
analysis.CoinJoin_tx ← {tx ∈ analysis | tx ∈ CoinJoin_TX}
analysis.lightning_tx ← {tx ∈ analysis | tx ∈ lightning_TX}
analysis.pet_tx ←
  PET_tx_find(tx_tainted, svc_ADR, CoinJoin_TX, mixer_ADR, lightning_TX)
  ▷ see Algorithm 16
APPEND evaluation ← MEAN(frequency),
  COUNT(analysis.reused_adr), COUNT(analysis.fresh_adr),
  MEAN(analysis.fee_rate), MEAN(analysis.adr_count),
  COUNT(analysis.svc_adr), COUNT(analysis.mixer_adr),
  COUNT(analysis.CoinJoin_tx), COUNT(analysis.lightning_tx),
  COUNT(analysis.pet_tx)
return analysis, evaluation

```

Algorithm 15: Transaction analysis

Function $PET_tx_find(tx_tainted, svc_ADR, mixer_ADR, CoinJoin_TX, lightning_TX)$:

Data: $tx_tainted$ contains targeted transactions for potential PET transaction classification. The script uses identified address and transaction data from svc_ADR , $CoinJoin_TX$, $mixer_ADR$, $lightning_TX$ to exclude identified service and PET transactions.

Result: pet_tx data containing potential PET transactions.

```

 $pet\_tx \leftarrow \{tx \in tx\_tainted \mid COUNT(tx.input) > 1 \wedge COUNT(tx.output) > 1\}$ 
 $pet\_tx \leftarrow \{tx \in pet\_tx \mid \exists v \in tx.input, v \notin tx\_tainted\}$ 
for  $tx \in pet\_tx$  do
  | if  $\exists v \in tx.adr, v \in svc\_ADR \vee v \in mixer\_ADR \vee \exists t \in tx, t \in$ 
  |    $CoinJoin\_TX \vee t \in lightning\_TX$  then
  |   | DELETE  $pet\_tx \leftarrow tx$ 
  | end
end
return  $pet\_tx$ 

```

Algorithm 16: Potential PET transaction detection

worsens the more nodes are displayed on the browser interface. Hence, we include an optional function argument to alter the complexity and the number of nodes in the exported data structure.

There are three options for the Neo4J export; the *detailed* option, which exports all blocks, transaction inputs, transaction outputs, transactions and addresses as nodes. The *simple* option exports transactions and addresses as nodes and exports transaction inputs (send) and outputs (receive) as relationships. Lastly, the *output-only* option exports only transaction outputs as nodes and include transaction and address information as variables in transaction output nodes.

Additionally, the taint analysis data results can be manually exported to other databases or network visualisation software and modules, such as Python NetworkX⁵, R network analysis software⁶, and Gephi⁷.

B.5.5 Website Scraping

The *TaintedTX* library includes several functions to perform website and API scraping and obtain data related to cryptocurrency addresses and transactions from external sources. The website scraping script for address profile data or *adr_scrape*

⁵<https://networkx.org>

⁶<https://www.r-project.org>

⁷<https://gephi.org>

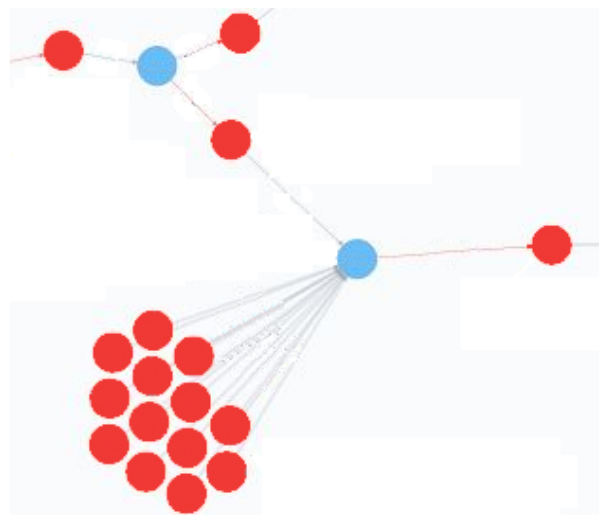


Figure B.8: Example of a Neo4j transaction network graph with the *simple* export option

Red circles represent address nodes and blue circles represent transaction nodes. Arrow lines to a transaction node represent a sending relationship and arrow lines from a transaction node represent a receiving relationship (transaction outputs).

function use the BeautifulSoup Python library⁸ to obtain address entity labels/tags from two Bitcoin address profiling websites namely, Wallet Explorer website⁹ and CheckBitcoinAddress website¹⁰.

The Wallet Explorer is one of the most well-known Bitcoin address labelling database websites used in various cryptocurrency research [115, 132, 175, 197, 205]. Wallet Explorer is a blockchain explorer website service with a feature that provides address entity information using multi-input address clustering heuristics. The service also utilises a deanonymisation method commonly referred to as *Mystery shopper payments*, which involves attackers exchanging their Bitcoins with targeted users or businesses to obtain information of addresses that belong to the targets. The website contains a large number of address labels from both services and users. Although the website's transaction and address clustering data are up-to-date, the website has not updated its address tag data since 2016.

As the Wallet Explorer website contains an enormous number of pages to scrape, the web scraping process will take a considerable amount of time to scrape all of the address data from this website. Hence, it would be more efficient to scrape only a subset of the addresses for each entity and perform multi-input address clustering

⁸<https://www.crummy.com/software/BeautifulSoup>

⁹<https://www.walletexplorer.com>

¹⁰<https://checkbitcoinaddress.com>

heuristics on the subset addresses instead with either the *adr_clustering* function or other software.

The CheckBitcoinAddress is a reporting and labelling website that allows users to report and verify Bitcoin addresses with an identity profile. Users can also send a signed message with an address's signature from the private key or a Bitcoin payment to the website to confirm the address ownership and provide a submitted address tag and website URL link. The website also has an address labelling system link to the users on BitcoinTalk forum¹¹ and Bitcoin-otc off-exchange marketplace¹².

The web scraping function for the CheckBitcoinAddress website can scrape all of the available address submitted link data, BitcoinTalk forum and Bitcoin-otc users within a day with a random and reasonable waiting interval to avoid disrupting the website's traffic. The function also includes an optional argument to scrape only unverified or verified addresses or both. However, the submitted address tags on the CheckBitcoinAddress website can belong to either a cryptocurrency service or a common Bitcoin user. Hence, the scraped address entity data will need to be manually filtered if users intend to utilise only cryptocurrency service address data.

In addition to the address profile scraping function, we implement a *coinjoin_scrape* function that can scrape the current unconfirmed CoinJoin transactions from the Wasabi wallet API¹³, as described in Section 5.2. This Wasabi API scraping function can be beneficial for the tracking of recent Wasabi CoinJoin transactions and building the CoinJoin transaction database.

B.6 Sample Use Cases

The *TaintedTX* library provides several functions that are beneficial for cryptocurrency tracking, transaction behaviour analysis, address and transaction profiling. Although we used the taint analysis function to perform Bitcoin tracking experiments only on illegal Bitcoin activity cases in Chapter 3, it is possible to utilise the *TaintedTX* library for general Bitcoin tracking and analysis purposes outside of crime forensic analysis. We investigate two types of events using the *TaintedTX*

¹¹<https://BitcoinTalk.org>

¹²<https://bitcoin-otc.com>

¹³<https://wasabiwallet.io/api/v4/btc/chaumiancoinjoin/unconfirmed-coinjoins>

library, which are historical Bitcoins payment (Section B.6.1) and miners' spending of newly mined Bitcoins (Section B.6.2).

B.6.1 10,000 BTC For Two Pizzas

```

1 import taintedtx
2 import pandas as pd
3
4 # Create TaintedTX object
5 tt = taintedtx.TaintedTX(path='btcddata/')
6
7 pizza_tx_hash = ["Censored Transaction Hash"]
8
9 # Prepare transaction data starting from the payment transaction for 1800 day
10 tt.prepare_data(tx=pizza_tx_hash, limitoption=np.timedelta64(1800, 'D'))
11
12 # Perform taint analysis for 50 transaction depths
13 tx_tainted = tt.tx_taint_search(tt.result.index.tolist(), case_name="pizza", depth_limit=50)
14
15 tx_tainted

```

output_index	tx_index	adr_index	clean_value	depth	output_value	spent_index	taint_value
624170	62417	612202	0.0	0	1000000000000	62419.0	1.000000e+12
624190	62419	607782	NaN	1	577700000000	62430.0	NaN
624191	62419	612221	NaN	1	422300000000	72953.0	NaN
624300	62430	607672	NaN	1	577700000000	72953.0	NaN
729530	72953	612332	NaN	1	1102200000000	72954.0	NaN
729540	72954	708402	NaN	2	550000000000	77590.0	NaN
729541	72954	708432	NaN	2	552200000000	72956.0	NaN
729560	72956	708452	NaN	3	567700000000	74935.0	NaN
749351	74935	733052	NaN	4	677000000000	104642.0	NaN
749350	74935	685192	NaN	4	500000000000	79684.0	NaN

Figure B.9: A Screenshot of Jupyter notebook running the taint analysis function on the Pizza payment transaction

We performed transaction tracking on the first purchase of Bitcoins to real-world products to illustrate how the *TaintedTX* library can be used for observing Bitcoin exchanges from one user to another. The event involves a Bitcoin user posting a topic in the BitcoinTalk forum [106] with a request to exchange 10,000 BTC Bitcoins (40 USD at the time) for two large pizzas. Another Bitcoin user accepted the request and ordered two pizzas to the assigned address location. The Bitcoin payment transaction occurred on 2010-05-22 at block 57,043, and it became one of the most well-known events in cryptocurrency as the first publicly known Bitcoin exchange with real-world products. The day of the transaction is now commonly referred to as Bitcoin Pizza Day by the Bitcoin user community.

We performed a taint analysis tracking starting from the payment transaction with the Poison strategy for 50 transaction depths. The Poison strategy results indicate that the pizza Bitcoins payment passed through 5,571 addresses in 5,188 transactions. There are 186 of these addresses identified as belonging to over-the-counter trading, payment and donation services, which make up to 3.4% of the

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3
4 start_btc = tx_tainted[tx_tainted['depth'] == 0]['output_value'].sum()
5 labels = ['Other Service', 'Payment', 'Wallet', 'Exchange', 'Darknet Market', 'Gambling', 'Unidentified']
6 colors = ['red', 'purple', 'green', 'yellow', 'orange', 'white']
7 sizes = [0,0,0,0,0,0]
8 service_found = tx_tainted[tx_tainted['adr_index'].isin(service_ADR.index)]
9
10 for index, entity in enumerate(labels):
11     sizes[index] = service_found[service_found['type'] == entity]['output_value'].sum() * 100 / start_btc
12 for index in (6,5,4,3,2,1,0):
13     if sizes[index] == 0: # Remove empty type
14         sizes.pop(index)
15         labels.pop(index)
16         colors.pop(index)
17
18 fig1, ax1 = plt.subplots()
19 ax1.pie(sizes, autopct='%1.1f%%', colors=colors, textprops={'fontsize': 20},
20         pctdistance=1.3, startangle=90, wedgeprops = {'edgecolor': 'black',
21                                                     'linewidth': 2, 'antialiased': True})
22 ax1.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
23
24 plt.show()
25

```

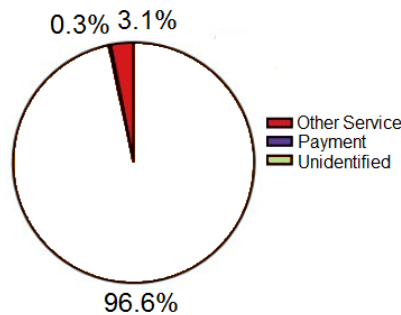


Figure B.10: Type of address entities that received portion of Pizza Bitcoins

addresses that received the Bitcoins, as shown in Figure B.10. At the end of the 50 transaction depth searches, there are 680 addresses that potentially held the pizza payment Bitcoins in their possession.

B.6.2 Miners' Bitcoin Spending Observation

We performed a transaction tracking experiment on Coinbase transactions to observe how newly mined Bitcoins are spent by their miners and enter the Bitcoin economy at large. We select ten Coinbase transactions from one random block in each year from 2011 to 2020 and perform taint analysis on these transactions for 20 transaction depths with the Dirty-First strategies.

As we aim to observe only the miners' spending in this use case experiment, we exclude the transactions of targeted Bitcoins after they are exchanged to a service address from the tracking results. The Dirty-First results show that only four of the sample Coinbase transactions illustrate the newly mined Bitcoins reaching cryptocurrency service addresses, as shown in Figure B.11. The type of cryptocurrency

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3 case_list = ['miner1', 'miner2', 'miner3', 'miner4', 'miner5',
4             'miner6', 'miner7', 'miner8', 'miner9', 'miner10']
5 labels = ['Other Service', 'Payment', 'Wallet', 'Exchange', 'Darknet Market', 'Gambling', 'Unidentified']
6 colors = ['red', 'purple', 'green', 'yellow', 'orange', 'white']
7 y = {key: [] for key in labels}
8
9 for case in case_list:
10 tx_tainted = pd.read_pickle(case + '.pkl')
11 start_btc = found[found['depth'] == 0]['output_value'].sum()
12 sizes = [0,0,0,0,0,0]
13 service_found = tx_tainted.merge(service_ADR, left_on="adr_index", right_on="adr_index", how="left")
14 service_found['type'] = service_found['type'].fillna("Unidentified")
15
16 for entity in labels:
17 y[entity].append(service_found[service_found['type'] == entity]['output_value'].sum() * 100 / start_btc)
18
19 for key in y:
20 plt.bar(x, y[key])
21 plt.xlabel("Bitcoin Spending Percentage")
22 plt.ylabel("Case")
23 plt.legend(labels)
24 plt.show()

```

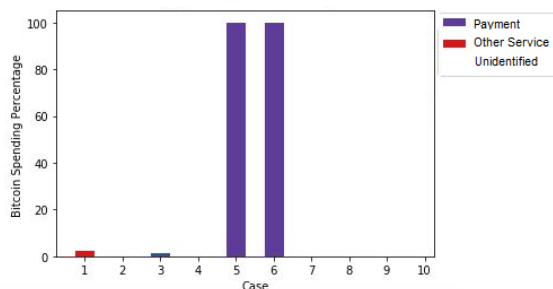


Figure B.11: Miners' Bitcoins spending

services that receive the newly mined Bitcoins is mainly payment services. Interestingly, we observe a small number of the newly mined Bitcoins directly reaching addresses belonging to darknet market service in case 3, which may indicate that darknet markets are one of the methods for miners to sell their Bitcoins.

The lack of spending in most sample cases are because the Dirty-First results typically contain only one or two transactions, which indicate that mining pools typically combine the newly mined Bitcoins with previously mined Bitcoins when sending newly mined Bitcoins to their mining participants, as illustrated in Figure B.12.

B.7 Conclusion and Future Work

The *TaintedTX* library is an open-sourced Python library that provides a collection of functions for cryptocurrency forensic analysis, such as taint analysis, zero-taint Bitcoin demixing, address clustering heuristics, and address and transaction classification. The functions in the *TaintedTX* library were tested and employed to produce the results we presented in previous chapters (Chapters 3, 4, and 5) and

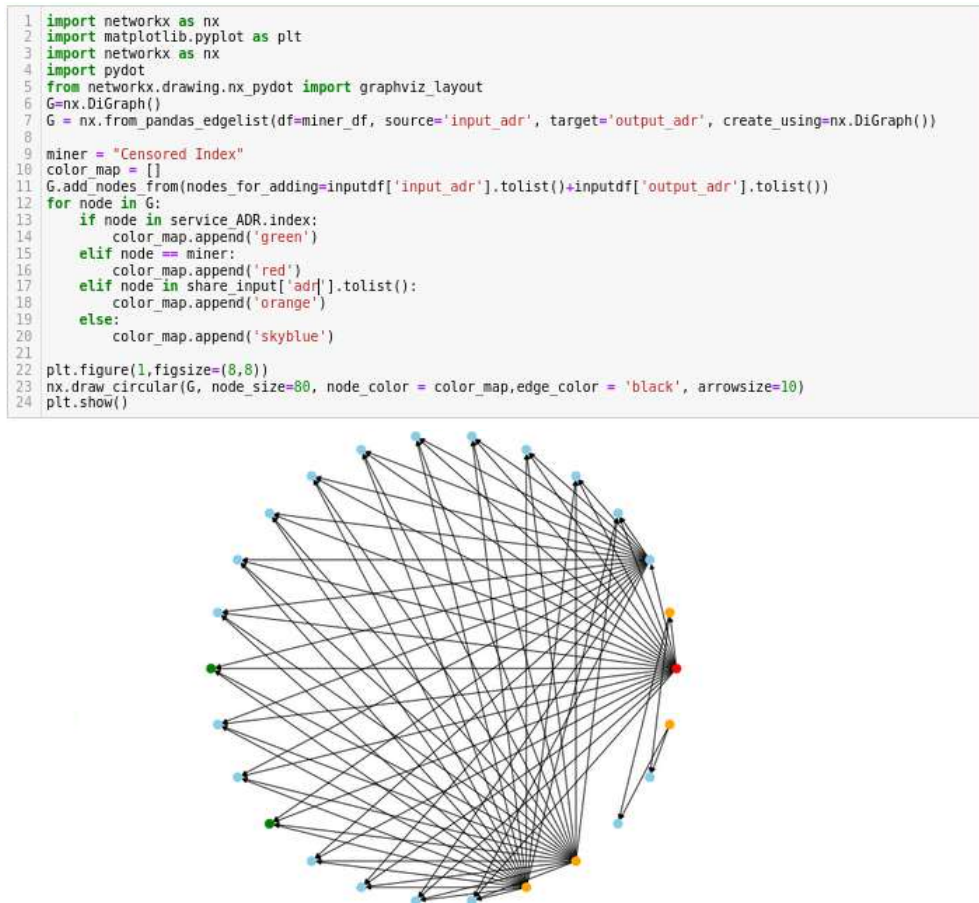


Figure B.12: Network graph of the case 3's Dirty-First results created with Python NetworkX module

Red circles indicate miners' addresses that received the newly mined Bitcoins in the Coinbase transaction. Green circles are cryptocurrency service addresses. Orange circles are input addresses that do not receive the newly mined Bitcoins but share the transaction inputs in the subsequent transactions that distribute the newly mined Bitcoins. Aqua circles are unidentified addresses that receive a portion of the newly mined Bitcoins after the Coinbase transaction.

Section B.6, which illustrate their real-world use cases. We identify limitations in the current implementation of the *TaintedTX* library that can be addressed in future work to improve the library's functionality and performance.

There is a potential scalability issue that the *TaintedTX* library may face in the future, similar to other blockchain analysis software. As cryptocurrency blockchain data grows over time, the storage size requires to save the database will also increase (358.76 Gigabytes as of 2021-08-11 for Bitcoin blockchain). The scalability issue can make storing and reading database files more challenging on personal computer systems. We have alleviated this issue partially by designing the database reading scripts to read data files on a yearly scale instead of the whole blockchain at once, which helps to improve the database reading speed and memory usage performance.

Nevertheless, the performance of the database reading process can be improved further by changing the database reading process to be capable of reading data files on a smaller scale.

While the current implementation of taint analysis in the *TaintedTX* library is sufficient to track Bitcoins up to a monthly scale on the system we described in Section B.1.1, a large scale tracking for several months or years requires an enormous amount of RAM to store transaction data for the taint analysis process and can take a considerable amount of time to complete. The memory issue can be improved further by either introducing a function to read data in smaller parts during the tracking process or changing the database system to the Dask library¹⁴, which is an implementation of Pandas data frame that specialises in parallel data frame operations. Both solutions will also improve the performance of all Pandas functions used in the *TaintedTX* library and allow a large scale or whole blockchain scale tracking to be feasible on an average computer system within a reasonable amount of time.

The overall performance of the functions in the *TaintedTX* library can also be improved by integrating a C language extension for Python likes Cython compiler¹⁵ to make use of the performance advantage of the C language while still keeping the *TaintedTX* library compatible with other Python libraries.

Several of the functions in the *TaintedTX* library can be expanded to include more features. One of the experimental features for the taint analysis function that we have in the future plan is *dynamic taint analysis* feature. As the current implementation of the taint analysis strategy function accepts only one taint analysis strategy per operation on *tx_tainteddata*, the dynamic taint analysis feature will allow the taint analysis process to adaptively switch taint analysis strategies either with automatic transaction behaviour detection or manual control by users. We also consider expanding the In-Out strategies to include transaction behaviours or characteristics as strategy variants. For example, a taint analysis strategy that prioritises coin distribution based on the starting tainted coins value, address format type, or address entity type.

It is also worth noting that the current implementation of the *TaintedTX* li-

¹⁴<https://docs.dask.org/en/latest>

¹⁵<https://cython.org>

brary's taint analysis does not include an option to taint the portion of coins that are paid as transaction fees to miners. Such feature can be implemented in the *tx_taint_search* and taint analysis strategy functions where the tracking algorithm can identify the Coinbase transaction output in the tainted transactions' block and distribute tainted coins to the Coinbase transaction output accordingly. This feature can be beneficial for tracking illegal coins that involve miners as accomplices.

Bibliography

- [1] M. Ahmed, I. Shumailov, and R. Anderson. Tendrils of crime: Visualizing the diffusion of stolen bitcoins. In *International Workshop on Graphical Models for Security*, pages 1–12, Oxford, United Kingdom, 2018. Springer.
- [2] F. Aiolli, M. Conti, A. Gangwal, and M. Polato. Mind your wallet’s privacy: identifying bitcoin wallet apps and user’s actions through network traffic analysis. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 1484–1491, 2019.
- [3] A. Al-Imam and B. A. AbdulMajeed. The NPS phenomenon and the deep web: Trends analyses and internet snapshots. *Global Journal of Health Science*, 9(11):71–85, 2017.
- [4] allinvain. I just got hacked - any help is welcome! (25,000 BTC stolen). <https://bitcointalk.org/index.php?topic=16457.0>, 2011. [Accessed: 2021-03-08].
- [5] K. Allman. The dark side of bitcoin. *LSJ: Law Society of NSW Journal*, (42):28–29, 2018.
- [6] R. Anderson, I. Shumailov, and M. Ahmed. Making bitcoin legal. In V. Matyáš, P. Švenda, F. Stajano, B. Christianson, and J. Anderson, editors, *Security Protocols XXVI*, pages 243–253, Cham, 2018. Springer International Publishing.
- [7] R. Anderson, I. Shumailov, M. Ahmed, and A. Rietmann. Bitcoin redux. In *Workshop on the Economics of Information Security*, pages 1–33, 2019.

- [8] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun. Evaluating user privacy in bitcoin. In A.-R. Sadeghi, editor, *International conference on financial cryptography and data security*, pages 34–51. Springer, 2013.
- [9] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Ocateau, and P. McDaniel. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. *ACM Sigplan Notices*, 49(6):259–269, 2014.
- [10] J. Atik and G. Gerro. Hard forks on the bitcoin blockchain: reversible exit, continuing voice. *Stanford Journal of Blockchain Law & Policy*, 1:1–24, 2018.
- [11] K. Atlas. BIP: 69. <https://github.com/bitcoin/bips/blob/master/bip-0069.mediawiki>, 2015. [Accessed: 2020-06-21].
- [12] N. Atzei, M. Bartoletti, S. Lande, and R. Zunino. A formal model of bitcoin transactions. In *International Conference on Financial Cryptography and Data Security*, pages 541–560. Springer, 2018.
- [13] S. Barber, X. Boyen, E. Shi, and E. Uzun. Bitter to better — how to make bitcoin a better currency. In *Financial Cryptography and Data Security*, pages 399–414, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [14] M. Bartoletti and L. Pompianu. An analysis of bitcoin OP_RETURN metadata. In *International Conference on Financial Cryptography and Data Security*, pages 218–230. Springer, 2017.
- [15] S. Ben Mariem, P. Casas, and B. Donnet. Vivisecting blockchain P2P networks: Unveiling the bitcoin ip network. In *ACM CoNEXT student workshop*, pages 1–3. ACM, 2018.
- [16] K. Bergman and S. Rajput. Revealing and concealing bitcoin identities: A survey of techniques. In *Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure*, pages 13–24, 2021.

- [17] N. Bhaskar and D. Chuen. Bitcoin mining technology. In *Handbook of Digital Currency: Bitcoin, Innovation, Financial Instruments, and Big Data*, pages 45–65. Academic Press, San Diego, 12 2015.
- [18] A. Biryukov and D. Feher. Privacy and linkability of mining in zcash. In *2019 IEEE Conference on Communications and Network Security (CNS)*, pages 118–123. IEEE, 2019.
- [19] A. Biryukov and S. Tikhomirov. Security and privacy of mobile wallet users in bitcoin, dash, monero, and zcash. *Pervasive and Mobile Computing*, 59:101030, 2019.
- [20] S. Bistarelli, I. Mercanti, and F. Santini. An analysis of non-standard transactions. *Frontiers in Blockchain*, 2:7, 2019.
- [21] S. Bistarelli, M. Parrocchini, and F. Santini. Visualizing bitcoin flows of ransomware: Wannacry one week later. In *Proceedings of the Second Italian Conference on Cyber Security*, volume 2058 of *CEUR Workshop Proceedings*, page 13. CEUR-WS, 2018.
- [22] Bitcoin Developers. BIP: 0016. https://en.bitcoin.it/wiki/BIP_0016, 2019. [Accessed: 2021-06-07].
- [23] Bitcoin Developers. BIP: 173. <https://github.com/bitcoin/bips/blob/master/bip-0173.mediawiki>, 2021. [Accessed: 2021-02-12].
- [24] Bitcoin Developers. Technical background of version 1 bitcoin addresses. https://en.bitcoin.it/wiki/Technical_background_of_version_1_Bitcoin_addresses, 2021. [Accessed: 2021-05-13].
- [25] Bitfinex. Bitfinex’s lightning network capacity is now even better. <https://blog.bitfinex.com/announcements/bitfinexs-lightning-network-capacity-is-now-even-better/>, 2020. [Accessed: 2020-02-16].
- [26] Bitrefill. The first 1 BTC lightning channel is live on mainnet! <https://blog.bitrefill.com/the-first-1-btc-lightning-channel-is-live-on-mainnet-a7b30690d8a8>, 2019. [Accessed: 2021-04-06].

- [27] Y. Boshmaf, H. Al Jawaheri, and M. Al Sabah. BlockTag: design and applications of a tagging system for blockchain analysis. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 299–313. Springer, 2019.
- [28] D. Boxler and K. R. Walcott. Static taint analysis tools to detect information flows. In *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*, pages 46–52. The Steering Committee of The World Congress in Computer Science, 2018.
- [29] M. S. Brown and B. Douglass. An event study of the effects of cryptocurrency thefts on cryptocurrency prices. In *2020 Spring Simulation Conference (SpringSim)*, pages 1–12. IEEE, 2020.
- [30] S. D. Brown. Cryptocurrency and criminality: The bitcoin opportunity. *The Police Journal*, 89(4):327–339, 2016.
- [31] D. Bryans. Bitcoin and money laundering: Mining for an effective solution. *Indiana Law Journal*, 89(13):442–472, 2014.
- [32] M. Campbell-Verduyn. Bitcoin, crypto-coins, and global anti-money laundering governance. *Crime, Law and Social Change*, 69(2):283–305, 2018.
- [33] T. Carr, J. Zhuang, D. Sablan, E. LaRue, Y. Wu, M. A. Hasan, and G. Mohler. Into the reverie: Exploration of the dream market. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1432–1441. IEEE, 2019.
- [34] Chainalysis Team. Covid is causing shipping issues, but natural competitive forces are causing darknet market consolidation. <https://blog.chainalysis.com/reports/darknet-markets-cryptocurrency-2020>, 2020. [Accessed: 2021-07-14].
- [35] Chainalysis Team. Crypto mixer usage reaches all-time highs in 2022, with nation state actors and cybercriminals contributing significant volume. <https://blog.chainalysis.com/reports/cryptocurrency-mixers/>, 2022. [Accessed: 2022-08-04].

- [36] T.-H. Chang and D. Svetinovic. Improving bitcoin ownership identification using transaction patterns analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(1):9–20, 2018.
- [37] D. Chaum. Blind signatures for untraceable payments. In *Advances in cryptography*, pages 199–203. Springer, 1983.
- [38] Q. Chen and R. A. Bridges. Automated behavioral analysis of malware: A case study of wannacry ransomware. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 454–460. IEEE, 2017.
- [39] Z. Cheng, X. Hou, R. Li, Y. Zhou, X. Luo, J. Li, and K. Ren. Towards a first step to understand the cryptocurrency stealing attack on ethereum. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*, pages 47–60, Chaoyang District, Beijing, Sept. 2019. USENIX Association.
- [40] Z. Cheng, X. Hou, R. Li, Y. Zhou, X. Luo, J. Li, and K. Ren. Towards a first step to understand the cryptocurrency stealing attack on ethereum. In *22nd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2019)*, pages 47–60, Beijing, China, 2019. USENIX Association.
- [41] U. W. Chohan. Are cryptocurrencies truly trustless? In *Cryptofinance and Mechanisms of Exchange*, pages 77–89. Springer, 2019.
- [42] CoinMarketCap. Today’s cryptocurrency prices by market cap. <https://coinmarketcap.com/>, 2021. [Accessed: 2021-04-20].
- [43] M. Conti, E. Sandeep Kumar, C. Lal, and S. Ruj. A survey on security and privacy issues of bitcoin. *IEEE Commun. Surv. Tutorials*, 20(4):3416–3452, 2018.
- [44] Crystal Analytics Team. Darknet use and bitcoin — a crypto activity report by crystal blockchain. <https://crystalblockchain.com/articles/darknet-use-and-bitcoin-a-crypto-activity-report-by-crystal-blockchain>, 2020. [Accessed: 2021-07-13].

- [45] D. J. Cumming, S. Johan, and A. Pant. Regulation of the crypto-economy: Managing risks, challenges, and regulatory uncertainty. *Journal of Risk and Financial Management*, 12(3):126, 2019.
- [46] W. Dai. B-money. <http://www.weidai.com/bmoney.txt>, 1998. [Accessed: 2021-03-03].
- [47] T. de Balthasar and J. Hernandez-Castro. An analysis of bitcoin laundry services. In *Nordic Conference on Secure IT Systems (NordSec)*, pages 297–312, Tartu, Estonia, 2017. Springer International Publishing.
- [48] C. Decker and R. Wattenhofer. Information propagation in the bitcoin network. In *IEEE P2P 2013 Proceedings*, pages 1–10. IEEE, 2013.
- [49] S. Dhumwad, M. Sukhadeve, C. Naik, M. K.N., and S. Prabhu. A peer to peer money transfer using SHA256 and merkle tree. In *2017 23RD Annual International Conference in Advanced Computing and Communications (ADCOM)*, pages 40–43. IEEE, 2017.
- [50] G. Di Battista, V. Di Donato, M. Patrignani, M. Pizzonia, V. Roselli, and R. Tamassia. Bitconeview: visualization of flows in the bitcoin transaction graph. In *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–8, 2015.
- [51] D. Drainville. An analysis of the bitcoin electronic cash system. <https://cryptochainuni.com/wp-content/uploads/An-Analysis-of-the-Bitcoin-Electronic-Cash-System.pdf>, 2012. [Accessed: 2021-05-23].
- [52] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In *Annual international cryptology conference*, pages 139–147. Springer, 1992.
- [53] D. Easley, M. O’Hara, and S. Basu. From mining to markets: The evolution of bitcoin transaction fees. *Journal of Financial Economics*, 134(1):91–109, 2019.
- [54] D. Ermilov, M. Panov, and Y. Yanovich. Automatic bitcoin address clustering. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 461–466, Cancun, Mexico, 2017. IEEE.

- [55] M. Essaid, H. W. Kim, W. Guil Park, K. Y. Lee, S. Jin Park, and H. T. Ju. Network usage of bitcoin full node. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1286–1291. IEEE, 2018.
- [56] Europol. Dark web hitman identified through crypto-analysis. <https://www.europol.europa.eu/newsroom/news/dark-web-hitman-identified-through-crypto-analysis>, 2021. [Accessed: 2021-06-14].
- [57] Europol. Europol wasabi wallet report. <https://www.tbstat.com/wp/uploads/2020/06/Europol-Wasabi-Wallet-Report.pdf>, 2021. [Accessed: 2021-09-29].
- [58] Europol. Cryptocurrencies: tracing the evolution of criminal finances. <https://www.europol.europa.eu/publications-events/publications/cryptocurrencies-tracing-evolution-of-criminal-finances>, 2022. [Accessed: 2022-02-01].
- [59] G. Fanti and P. Viswanath. Deanonymization in the bitcoin P2P network. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1364–1373, 2017.
- [60] Y. J. Fanusie and T. Robinson. Bitcoin laundering: An analysis of illicit flows into digital currency services. memorandum, Center on Sanctions & Illicit Finance CSIF, 2018.
- [61] D. Ferrin. A preliminary field guide for bitcoin transaction patterns. In *Proceeding of the Texas Bitcoin Conference*, pages 1–10, 2015.
- [62] M. Fleder, M. S. Kester, and S. Pillai. Bitcoin transaction graph analysis. *arXiv:1502.01657 [cs]*, Feb. 2015.
- [63] S. Foley, J. R. Karlsen, and T. J. Putniņš. Sex, Drugs, and Bitcoin: How Much Illegal Activity Is Financed through Cryptocurrencies? *The Review of Financial Studies*, 32(5):1798–1853, 2019.

- [64] D. Fullmer and A. S. Morse. Analysis of difficulty control in bitcoin and proof-of-work blockchains. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 5988–5992. IEEE, 2018.
- [65] A. Gaihre, Y. Luo, and H. Liu. Do bitcoin users really care about anonymity? an analysis of the bitcoin transaction graph. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1198–1207, 2018.
- [66] N. Gandal, J. Hamrick, T. Moore, and T. Oberman. Price manipulation in the bitcoin ecosystem. *Journal of Monetary Economics*, 95:86–96, 2018.
- [67] J. Gao, H. Liu, C. Liu, Q. Li, Z. Guan, and Z. Chen. Easyflow: Keep ethereum away from overflow. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, pages 23–26. IEEE, 2019.
- [68] L. Ge and T. Jiang. A privacy protection method of lightweight nodes in blockchain. *Security and Communication Networks*, 2021, 2021.
- [69] A. Gervais, G. O. Karame, V. Capkun, and S. Capkun. Is bitcoin a decentralized currency? *IEEE security & privacy*, 12(3):54–60, 2014.
- [70] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 3–16, 2016.
- [71] S. Ghesmati, W. Fdhila, and E. Weippl. Studying bitcoin privacy attacks and their impact on bitcoin-based identity methods. In *International Conference on Business Process Management*, pages 85–101. Springer, 2021.
- [72] S. Ghesmati, A. Kern, A. Judmayer, N. Stifter, and E. Weippl. Unnecessary input heuristics and PayJoin transactions. In *International Conference on Human-Computer Interaction*, pages 416–424. Springer, 2021.
- [73] E. Ghysels and G. Nguyen. Price discovery of a speculative asset: Evidence from a bitcoin exchange. *Journal of Risk and Financial Management*, 12(4):164, 2019.

- [74] D. Gilbert. CryptoLocker gang earns millions in just 100 days. <https://www.ibtimes.co.uk/cryptoLocker-criminals-earn-30-million-100-days-1429607>, 2013. [Accessed: 2021-05-21].
- [75] S. Goldfeder, H. A. Kalodner, D. Reisman, and A. Narayanan. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *Proceedings on Privacy Enhancing Technologies*, 2018(4):179–199, 2018.
- [76] G. Grant and R. Hogan. Bitcoin: Risks and controls. *Journal of Corporate Accounting & Finance*, 26(5):29–35, 2015.
- [77] Y. Guo, J. Tong, and C. Feng. A measurement study of bitcoin lightning network. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 202–211, 2019.
- [78] S. Haber and W. S. Stornetta. How to time-stamp a digital document. In *Conference on the Theory and Application of Cryptography*, pages 437–455. Springer, 1990.
- [79] M. Harrigan and C. Fretter. The unreasonable effectiveness of address clustering. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*, pages 368–373, Toulouse, France, 2016. IEEE.
- [80] R. Havar. Bustapay : a practical coinjoin protocol. <https://github.com/bitcoin/bips/blob/master/bip-0079.mediawiki>, 2021. [Accessed: 2021-08-26].
- [81] P. He, G. Yu, and Y. Zhang. Prospective review of blockchain technology and application. *Computer Science*, 44:1–7, 2017.
- [82] H. Hellani, A. E. Samhat, M. Chamoun, H. E. Ghor, and A. Serhrouchni. On blockchain technology: Overview of bitcoin and future insights. In *2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, pages 1–8, 2018.

- [83] J. R. Hendrickson, T. L. Hogan, and W. J. Luther. The political economy of bitcoin. *Economic Inquiry*, 54(2):925–939, 2016.
- [84] J. Herrera-Joancomartí. Research and challenges on bitcoin anonymity. In *Data Privacy Management (DPM)*, pages 3–16, Cham, 2015. Springer International Publishing.
- [85] J. Herrera-Joancomartí, G. Navarro-Arribas, A. Ranchal-Pedrosa, C. Pérez-Solà, and J. Garcia-Alfaro. On the difficulty of hiding the balance of lightning network channels. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, page 602–612, New York, NY, USA, 2019. Association for Computing Machinery.
- [86] J. Herrera-Joancomartí, G. Navarro-Arribas, A. Ranchal-Pedrosa, C. Pérez-Solà, and J. Garcia-Alfaro. On the difficulty of hiding the balance of lightning network channels. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 602–612, 2019.
- [87] J. Herrera-Joancomartí and C. Pérez-Solà. Privacy in bitcoin transactions: new challenges from blockchain scalability solutions. In *International Conference on Modeling Decisions for Artificial Intelligence*, pages 26–44. Springer, 2016.
- [88] G. Hileman and M. Rauchs. Global cryptocurrency benchmarking study. *Cambridge Centre for Alternative Finance*, 33:33–113, 2017.
- [89] Y. Hong, H. Kwon, J. Lee, and J. Hur. A practical de-mixing algorithm for bitcoin mixing services. In *ACM Blockchains, Cryptocurrencies, and Contracts (BCC)*, pages 15–20, New York, NY, USA, 2018. Association for Computing Machinery.
- [90] Hotbit. Hotbit’s announcement on emergency maintenance. <https://hotbit.zendesk.com/hc/en-us/articles/1500008915521->, 2021. [Accessed: 2021-06-24].
- [91] N. Houy. The economics of bitcoin transaction fees. *Groupe d’Analyse et de Théorie Economique (GATE) WP*, 1407:1–10, 2014.

- [92] D. Y. Huang, M. M. Aliapoulios, V. G. Li, L. Invernizzi, E. Bursztein, K. McRoberts, J. Levin, K. Levchenko, A. C. Snoeren, and D. McCoy. Tracking ransomware end-to-end. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 618–631. IEEE, 2018.
- [93] A. Hui and W. Zhao. Over \$280m drained in kucoin crypto exchange hack. <https://www.coindesk.com/markets/2020/09/26/over-280m-drained-in-kucoin-crypto-exchange-hack/>, 2021. [Accessed: 2021-07-25].
- [94] S. Jiang and J. Wu. Bitcoin mining with transaction fees: a game on the block size. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 107–115. IEEE, 2019.
- [95] M. Jourdan, S. Blandin, L. Wynter, and P. Deshpande. Characterizing entities in the bitcoin blockchain. In H. Tong, Z. Jessie Li, F. Zhu, and J. Yu, editors, *International Conference on Data Mining Workshops, ICDM Workshops, Singapore, Singapore, November 17-20, 2018*, pages 55–62. IEEE, 2018.
- [96] M. Jourdan, S. Blandin, L. Wynter, and P. Deshpande. A probabilistic model of the bitcoin blockchain. In *Computer Vision and Pattern Recognition Workshop (CVPRW), 2019*, pages 2784–2792. IEEE, 2019.
- [97] H. Kalodner, M. Möser, K. Lee, S. Goldfeder, M. Plattner, A. Chator, and A. Narayanan. BlockSci: Design and applications of a blockchain analysis platform. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2721–2738. USENIX Association, 2020.
- [98] P. K. Kaushal, A. Bagga, and R. Sobti. Evolution of bitcoin and security risk in bitcoin wallets. In *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, pages 172–177. IEEE, 2017.
- [99] S. Kethineni and Y. Cao. The rise in popularity of cryptocurrency and associated criminal activity. *International Criminal Justice Review*, 30(3):325–344, 2020.
- [100] S. Kethineni, Y. Cao, and C. Dodge. Use of bitcoin in darknet markets:

- Examining facilitative factors on bitcoin-related crimes. *American Journal of Criminal Justice*, 43(2):141–157, 2018.
- [101] J. A. Kroll, I. C. Davey, and E. W. Felten. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, volume 2013, page 11, 2013.
- [102] N. Kshetri and J. Voas. Do crypto-currencies fuel ransomware? *IT professional*, 19(5):11–15, 2017.
- [103] H. Kuzuno and C. Karam. Blockchain explorer: An analytical process and investigation environment for bitcoin. In *2017 APWG Symposium on Electronic Crime Research (eCrime)*, pages 9–16, 2017.
- [104] N. Lacey. *Teach yourself Programming with pseudocode and Python for AQA GCSE Computer Science (8520) Student Workbook*. Droylsden Academy, 2018.
- [105] E. Lam, J. Lee, and J. Robertson. Cryptocurrencies lose 42 billion USD after south korean bourse hack. <https://www.bloomberg.com/news/articles/2018-06-10/bitcoin-tumbles-most-in-two-weeks-amid-south-korea-exchange-hack>, 2018. [Accessed: 2020-04-16].
- [106] laszlo. Pizza for bitcoins? <https://bitcointalk.org/index.php?topic=137.0>, 2010. [Accessed: 2021-06-14].
- [107] Y. Lewenberg, Y. Bachrach, Y. Sompolinsky, A. Zohar, and J. S. Rosenschein. Bitcoin mining pools: A cooperative game theoretic analysis. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 919–927, 2015.
- [108] X. Li, Z. Yang, L. Wei, and C. Zhang. Protecting access privacy for bitcoin lightweight client using trusted hardware. In *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 706–711. IEEE, 2019.
- [109] Y. Li, Z. Liu, and Z. Zheng. Quantitative analysis of bitcoin transferred in bitcoin exchange. In *International Conference on Blockchain and Trustworthy Systems*, pages 549–562. Springer, 2019.

- [110] K. Liao, Z. Zhao, A. Doupé, and G.-J. Ahn. Behind closed doors: measurement and analysis of cryptolocker ransoms in bitcoin. In *2016 APWG symposium on electronic crime research (eCrime)*, pages 1–13. IEEE, 2016.
- [111] M. Lischke and B. Fabian. Analyzing the bitcoin network: The first four years. *Future Internet*, 8(7):1–40, 2016.
- [112] Y. Liu, R. Li, X. Liu, J. Wang, L. Zhang, C. Tang, and H. Kang. An efficient method to enhance bitcoin wallet security. In *2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, pages 26–29. IEEE, 2017.
- [113] A. A. Maksutov, M. S. Alexeev, N. O. Fedorova, and D. A. Andreev. Detection of blockchain transactions used in blockchain mixer of coin join type. In *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 274–277. IEEE, 2019.
- [114] S. Matetic, K. Wüst, M. Schneider, K. Kostianen, G. Karame, and S. Capkun. BITE: Bitcoin lightweight client privacy using trusted execution. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 783–800, 2019.
- [115] H. Matsumoto, S. Igaki, and H. Kikuchi. Address usage estimation based on bitcoin traffic behavior. In *International Conference on Network-Based Information Systems*, pages 188–199. Springer, 2020.
- [116] R. Matzutt, J. Hiller, M. Henze, J. Ziegeldorf, D. Müllmann, O. Hohlfeld, and K. Wehrle. A quantitative analysis of the impact of arbitrary blockchain content on bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 420–438. Springer, 2018.
- [117] F. K. Maurer, T. Neudecker, and M. Florian. Anonymous coinjoin transactions with arbitrary values. In *Trustcom/BigDataSE/ICISS, Sydney, Australia, August 1-4, 2017*, pages 522–529. IEEE Computer Society, 2017.
- [118] G. Maxwell. CoinJoin: Bitcoin privacy for the real world. <https://bitcointalk.org/?topic=279249>, 2013. [Online forum comment], [Accessed: 2020-10-26].

- [119] P. McCorry, E. Heilman, and A. Miller. Atomically trading with roger: Gambling on the success of a hardfork. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 334–353. Springer, 2017.
- [120] S. Meiklejohn and C. Orlandi. Privacy-enhancing overlays in bitcoin. In *International Conference on Financial Cryptography and Data Security*, pages 127–141. Springer, 2015.
- [121] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. A fistful of bitcoins: Characterizing payments among men with no names. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, page 127–140, New York, NY, USA, 2013. Association for Computing Machinery.
- [122] I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *2013 IEEE Symposium on Security and Privacy*, pages 397–411, Berkeley, CA, USA, 2013. IEEE.
- [123] J. Ming, D. Wu, G. Xiao, J. Wang, and P. Liu. TaintPipe: Pipelined symbolic taint analysis. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 65–80, 2015.
- [124] P. Monamo, V. Marivate, and B. Twala. Unsupervised learning for robust bitcoin fraud detection. In *2016 Information Security for South Africa (ISSA)*, pages 129–134. IEEE, 2016.
- [125] T. Moore and N. Christin. Beware the middleman: Empirical analysis of bitcoin-exchange risk. In A.-R. Sadeghi, editor, *Financial Cryptography and Data Security*, pages 25–33, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [126] D. T. Morton. The future of cryptocurrency: An unregulated instrument in an increasingly regulated global economy. *Loy. U. Chi. Int’l L. Rev.*, 16:129, 2020.
- [127] M. Möser, R. Böhme, and D. Breuker. An inquiry into money laundering tools

- in the Bitcoin ecosystem. In *2013 APWG eCrime Researchers Summit*, pages 1–14, San Francisco, CA, USA, 2013. IEEE.
- [128] M. Möser, R. Böhme, and D. Breuker. Towards risk scoring of bitcoin transactions. In *Financial Cryptography and Data Security*, Lecture Notes in Computer Science, pages 16–32, Christ Church, Barbados, 2014. Springer Berlin Heidelberg.
- [129] M. Mosk. Underground website used for black market drug sales bigger than the original, report says. <https://abcnews.go.com/Blotter/silk-road-underground-website-black-market-drug-sales/story?id=23528712>, 2014. [Accessed: 2021-05-13].
- [130] J. Moubarak, E. Filiol, and M. Chamoun. Comparative analysis of blockchain technologies and tor network: Two faces of the same reality? In *2017 1st Cyber Security in Networking Conference (CSNet)*, pages 1–9. IEEE, 2017.
- [131] Mt.Gox. Clarification of MT. GOX compromised accounts and major bitcoin sell-off. https://web.archive.org/web/20110919162635/https://mtgox.com/press_release_20110630.html, 2011. [Accessed: 2021-05-05].
- [132] H. Mun, S. Kim, and Y. Lee. A RDBMS-based bitcoin analysis method. In *International Conference on Information Security and Cryptology*, pages 235–253. Springer, 2020.
- [133] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System White Paper. <https://bitcoin.org/bitcoin.pdf>, 2009. [Accessed: 2021-02-15].
- [134] D. Neilson, S. Hara, and I. Mitchell. Bitcoin forensics: A tutorial. In *International Conference on Global Security, Safety, and Sustainability*, pages 12–26. Springer, 2017.
- [135] J. Neisius and R. Clayton. Orchestrated crime: The high yield investment fraud ecosystem. In *2014 APWG Symposium on Electronic Crime Research (eCrime)*, pages 48–58. IEEE, 2014.

- [136] Networked and Embedded Systems Research Group. Coin demixer for best-mixer.io. <https://github.com/nesfit/jane-DeMixer>, 2020. [Accessed: 2021-05-12].
- [137] V. N. Nezamaikin and E. P. Zbirovskaya. Contemporary challenges of OTC trading in digital assets. In *2nd International Conference on Economy, Management and Entrepreneurship (ICOEME 2019)*, volume 10. Atlantis Press, 2019.
- [138] S. Noether. Ring SIGNature confidential transactions for monero. *IACR Cryptol. ePrint Arch.*, 2015:1098, 2015.
- [139] M. Nofer, P. Gomber, O. Hinz, and D. Schiereck. Blockchain. *Business & Information Systems Engineering*, 59(3):183–187, 2017.
- [140] nopara73. Wasabi wallet 1.0 is released. <https://nopara73.medium.com/wasabi-1-stable-f8bc5e48289f>, 2018. [Accessed: 2021-06-02].
- [141] M. Nowostawski and J. Tøn. Evaluating methods for the identification of off-chain transactions in the lightning network. *Applied Sciences*, 9, 06 2019.
- [142] J. Oakley, C. Worley, L. Yu, R. Brooks, and A. Skjellum. Unmasking criminal enterprises: an analysis of bitcoin transactions. In *2018 13th International Conference on Malicious and Unwanted Software (MALWARE)*, pages 161–166. IEEE, 2018.
- [143] K. Oosthoek and C. Doerr. From hodl to heist: Analysis of cyber security threats to bitcoin exchanges. In *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 1–9. IEEE, 2020.
- [144] M. Óskarsdóttir, J. Mallett, A. L. Arnarson, and A. S. Stefánsson. Analysis of tainted transactions in the bitcoin blockchain transaction network. In *International Conference on Complex Networks and Their Applications*, pages 571–581. Springer, 2020.
- [145] M. Paquet-Clouston, B. Haslhofer, and B. Dupont. Ransomware payments in the bitcoin ecosystem. *Journal of Cybersecurity*, 5(1):tyz003, 2019.

- [146] S. Phetsouvanh, A. Datta, and F. Oggier. Analysis of multi-input multi-output transactions in the bitcoin network. *Concurrency and Computation: Practice and Experience*, 33(1):e5629, 2021.
- [147] B. B. F. Pontiveros, R. Norvill, and R. State. Monitoring the transaction selection policy of bitcoin mining pools. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6. IEEE, 2018.
- [148] J. Poon and T. Dryja. The bitcoin lightning network: Scalable off-chain instant payments (draft version 0.5.9.1). <https://lightning.network/lightning-network-paper.pdf>, 2015. [Accessed: 2021-04-05].
- [149] E. Reddy and A. Minnaar. Cryptocurrency: A tool and target for cybercrime. *Acta Criminologica: African Journal of Criminology & Victimology*, 31(3):71–92, 2018.
- [150] F. Reid and M. Harrigan. An analysis of anonymity in the bitcoin system. In *Security and privacy in social networks*, pages 197–223. Springer, 2013.
- [151] R. Richardson and M. M. North. Ransomware: Evolution, mitigation and prevention. *International Management Review*, 13(1):10, 2017.
- [152] T. Robinson. Elliptic follows the \$7 billion in bitcoin stolen from bitfinex in 2016. <https://www.elliptic.co/blog/elliptic-analysis-bitcoin-bitfinex-theft>, 2021. [Accessed: 2022-07-28].
- [153] P. Rodwald. An analysis of data hidden in bitcoin addresses. In *International Conference on Dependability and Complex Systems*, pages 369–379. Springer, 2021.
- [154] D. Ron and A. Shamir. Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*, pages 6–24. Springer, 2013.
- [155] D. Ron and A. Shamir. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography and Data Security*, volume 7859, pages 6–24. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

- [156] D. Ron and A. Shamir. How did dread pirate roberts acquire and protect his bitcoin wealth? In *International Conference on Financial Cryptography and Data Security*, pages 3–15. Springer, 2014.
- [157] R. Russell. #bitcoin-lightning FAQ: Why the 0.042 bitcoin limit? <https://rusty-lightning.medium.com/bitcoin-lightning-faq-why-the-0-042-bitcoin-limit-2eb48b703f3>, 2017. [Accessed: 2021-04-06].
- [158] R. Russell, P.-M. Padiou, L. Mutch, O. Osuntokun, C. Decker, practicalswift, C. Jämthagen, S. Appelcline, J. Posen, F. Drouin, N. Ueno, and S. Dobson. BOLT #2: Peer protocol for channel management. <https://github.com/lightningnetwork/lightning-rfc/blob/master/02-peer-protocol.md>, 2020. [Accessed: 2021-04-06].
- [159] Samurai Wallet Developers. A holistic approach to coinjoin. https://blog.samurairwallet.com/post/186458671552/a-holistic-approach-to-coinjoin?is_related_post=1, 2021. [Accessed: 2021-07-09].
- [160] Z. Samsudeen, D. Perera, and M. Fernando. Behavioral analysis of bitcoin users on illegal transactions. *Adv. Sci. Technol. Eng. Syst. J*, 4(2), 2019.
- [161] Z. Samsudeen, D. Perera, and M. Fernando. Behavioral Analysis of Bitcoin Users on Illegal Transactions. *Advances in Science, Technology and Engineering Systems Journal*, 4(2), 2019.
- [162] L. C. Schaupp and M. Festa. Cryptocurrency adoption and the road to regulation. In *Proceedings of the 19th Annual International Conference on Digital Government Research: Governance in the Data Age*, pages 1–9, 2018.
- [163] Y. Shahsavari, K. Zhang, and C. Talhi. A theoretical model for fork analysis in the bitcoin network. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 237–244. IEEE, 2019.
- [164] M. Spagnuolo, F. Maggi, and S. Zanero. Bitiodine: Extracting intelligence from the bitcoin network. In *International conference on financial cryptography and data security*, pages 457–468, Christ Church, Barbados, 2014. Springer.

- [165] C. Stokel-Walker. The murky world of the bitcoin scam. *New Scientist*, 237(3160):12, 2018.
- [166] N. Szabo. Bit gold. <http://unenumerated.blogspot.com/2005/12/bit-gold.html>, 2005. [Accessed: 2021-03-02].
- [167] J. Tapsell, R. Naeem Akram, and K. Markantonakis. An evaluation of the security of the bitcoin peer-to-peer network. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCoM) and IEEE Smart Data (SmartData)*, pages 1057–1062. IEEE, 2018.
- [168] P. Tasatanattakool and C. Techapanupreeda. Blockchain: Challenges and applications. In *2018 International Conference on Information Networking (ICOIN)*, pages 473–475. IEEE, 2018.
- [169] P. Tasca, A. Hayes, and S. Liu. The evolution of the bitcoin economy: Extracting and analyzing the network of payment relationships. *The Journal of Risk Finance*, 19(2):94–126, 2018.
- [170] S. Tikhomirov, P. Moreno-Sanchez, and M. Maffei. A quantitative analysis of security, anonymity and scalability for the lightning network. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, pages 387–396, 2020.
- [171] T. Tironsakkul, M. Maarek, A. Eross, and M. Just. Probing the mystery of cryptocurrency theft: An investigation into methods for taint analysis. *arXiv preprint arXiv:1906.05754*, 2020. Prior version presented at Cryptocurrency Research Conference (CRC) 2019.
- [172] T. Tironsakkul, M. Maarek, A. Eross, and M. Just. Tracking mixed bitcoins. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 447–457, Cham, 2020. Springer International Publishing.
- [173] T. Tironsakkul, M. Maarek, A. Eross, and M. Just. Context matters: Methods for bitcoin tracking, 2022. Under Review.

- [174] T. Tironsakkul, M. Maarek, A. Eross, and M. Just. The unique dressing of transactions: Wasabi coinjoin transaction detection. In *Proceedings of European Interdisciplinary Cybersecurity Conference (EICC'22)*, volume 4, page 21–28, New York, NY, USA, 2022. ACM. Best Paper Award.
- [175] K. Toyoda, P. T. Mathiopoulos, and T. Ohtsuki. A novel methodology for hyip operators' bitcoin addresses identification. *IEEE Access*, 7:74835–74848, 2019.
- [176] F. Tschorsch and B. Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, 18(3):2084–2123, 2016.
- [177] A. Turner and A. S. M. Irwin. Bitcoin transactions: a digital discovery of illicit activity on the blockchain. *Journal of Financial Crime*, 25(1):109–130, 2017.
- [178] V. Vallois and F. A. Guenane. Bitcoin transaction: From the creation to validation, a protocol overview. In *2017 1st Cyber Security in Networking Conference (CSNet)*, pages 1–7. IEEE, 2017.
- [179] G. van Dam, R. A. Kadir, P. N. E. Nohuddin, and H. B. Zaman. Improvements of the balance discovery attack on lightning network payment channels. In M. Hölbl, K. Rannenber, and T. Welzer, editors, *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 313–323. Springer, 2020.
- [180] N. van Saberhagen. CryptoNote v 2.0. <https://cryptonote.org/whitepaper.pdf>, 2013. [Accessed: 2021-08-13].
- [181] R. van Wegberg, J.-J. Oerlemans, and O. van Deventer. Bitcoin money laundering: mixed results? An explorative study on money laundering of cyber-crime proceeds using bitcoin. *Journal of Financial Crime*, 25:419–435, 2018.
- [182] M. Vasek and T. Moore. There's no free lunch, even using bitcoin: Tracking the popularity and profits of virtual currency scams. In *International conference on financial cryptography and data security*, pages 44–61. Springer, 2015.

- [183] M. Vasek and T. Moore. Analyzing the bitcoin ponzi scheme ecosystem. In *International Conference on Financial Cryptography and Data Security*, pages 101–112. Springer, 2018.
- [184] D. Vujičić, D. Jagodić, and S. Randić. Blockchain technology, bitcoin, and ethereum: A brief overview. In *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–6. IEEE, 2018.
- [185] D. Wang, J. Zhao, and Y. Wang. A survey on privacy protection of blockchain: The technology and application. *IEEE Access*, 8:108766–108781, 2020.
- [186] K. Wang, J. Pang, D. Chen, Y. Zhao, D. Huang, C. Chen, and W. Han. A large-scale empirical analysis of ransomware activities in bitcoin. *ACM Transactions on the Web (TWEB)*, 16(2):1–29, 2021.
- [187] Wasabi Wallet Developers. How long does it take to mix my coins? <https://docs.wasabiwallet.io/FAQ/FAQ-UseWasabi.html#how-long-does-it-take-to-mix-my-coins>, 2018. [Accessed: 2021-06-06].
- [188] Wasabi Wallet Developers. Wasabi: Privacy focused bitcoin wallet for desktop. <https://nopara73.medium.com/wasabi-privacy-focused-bitcoin-wallet-for-desktop-3962d567045a>, 2018. [Accessed: 2021-06-06].
- [189] Wasabi Wallet Developers. Wasabi v1.1.10: The unspendables. <https://github.com/zkSNACKs/WalletWasabi/releases/tag/v1.1.10>, 2019. [Accessed: 2021-07-05].
- [190] Wasabi Wallet Developers. Change coins. <https://docs.wasabiwallet.io/using-wasabi/ChangeCoins.html>, 2020. [Accessed: 2021-06-12].
- [191] Wasabi Wallet Developers. Wasabi v1.1.12 - to the moon. <https://github.com/zkSNACKs/WalletWasabi/releases/tag/v1.1.12>, 2020. [Accessed: 2021-07-05].
- [192] Wasabi Wallet Developers. Announcing wasabi wallet 2.0. <https://blog.wasabiwallet.io/wasabi-wallet-2/>, 2021. [Accessed: 2021-09-29].

- [193] Wasabi Wallet Developers. How long does it take to mix my coins? <https://docs.wasabiwallet.io/FAQ/FAQ-UseWasabi.html#how-long-does-it-take-to-mix-my-coins>, 2021. [Accessed: 2021-07-09].
- [194] Wasabi Wallet Developers. Wasabi wallet 2.0 status update. <https://blog.wasabiwallet.io/wasabi-wallet-2-update/>, 2021. [Accessed: 2021-09-29].
- [195] Wasabi Wallet Developers. What is the coordinator address? <https://docs.wasabiwallet.io/FAQ/FAQ-UseWasabi.html#what-is-the-coordinator-address>, 2021. [Accessed: 2021-07-09].
- [196] N. Webb. A fork in the blockchain: Income tax and the bitcoin/bitcoin cash hard fork. *North Carolina Journal of Law & Technology*, 19(4):283, 2018.
- [197] J. Wu, J. Liu, W. Chen, H. Huang, Z. Zheng, and Y. Zhang. Detecting mixing services via mining bitcoin transaction network with hybrid motifs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021.
- [198] J. Wu, J. Liu, Y. Zhao, and Z. Zheng. Analysis of cryptocurrency transactions from a network perspective: An overview. *Journal of Network and Computer Applications*, 190:103139, 2021.
- [199] L. Wu, Y. Hu, Y. Zhou, H. Wang, X. Luo, Z. Wang, F. Zhang, and K. Ren. Towards understanding and demystifying bitcoin mixing services. In *Proceedings of the Web Conference 2021*, pages 33–44, New York, NY, USA, 2021. Association for Computing Machinery.
- [200] P. Xia, H. Wang, B. Zhang, R. Ji, B. Gao, L. Wu, X. Luo, and G. Xu. Characterizing cryptocurrency exchange scams. *Computers & Security*, 98:101993, 2020.
- [201] Y. Xie, C. Zhang, L. Wei, Y. Niu, and F. Wang. Private transaction retrieval for lightweight bitcoin client. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pages 440–446. IEEE, 2019.
- [202] M. Xu, X. Chen, and G. Kou. A systematic review of blockchain. *Financial Innovation*, 5(1):1–14, 2019.

- [203] W. Yao, K. Xu, and Q. Li. Exploring the influence of news articles on bitcoin price with machine learning. In *2019 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE, 2019.
- [204] H. Yousaf, G. Kappos, and S. Meiklejohn. Tracing transactions across cryptocurrency ledgers. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 837–850, Santa Clara, CA, 2019. USENIX Association.
- [205] X. Yue, X. Shu, X. Zhu, X. Du, Z. Yu, D. Papadopoulos, and S. Liu. Bitextract: Interactive visualization for extracting bitcoin exchange intelligence. *IEEE transactions on visualization and computer graphics*, 25(1):162–171, 2018.
- [206] P. Zabka, K.-T. Förster, S. Schmid, and C. Decker. Node classification and geographical analysis of the lightning cryptocurrency network. In *International Conference on Distributed Computing and Networking 2021, ICDCN '21*, page 126–135, New York, NY, USA, 2021. Association for Computing Machinery.
- [207] L. Zhang, Z. Zhang, W. Wang, R. Waqas, C. Zhao, S. Kim, and H. Chen. A covert communication method using special bitcoin addresses generated by vanitygen. *Comput., Mater. Continua*, 65(1):597–616, 2020.
- [208] Y. Zhao, B. Niu, P. Li, and X. Fan. A novel enhanced lightweight node for blockchain. In *International Conference on Blockchain and Trustworthy Systems*, pages 137–149. Springer, 2019.
- [209] B. Zheng, L. Zhu, M. Shen, X. Du, and M. Guizani. Identifying the vulnerabilities of bitcoin anonymous mechanism based on address clustering. *Science China Information Sciences*, 63(3):1–15, 2020.
- [210] J. Zhu, P. Liu, and L. He. Mining information on bitcoin network data. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 999–1003. IEEE, 2017.