

# Cryptocurrency Privacy in Practice

Malte Möser

A DISSERTATION  
PRESENTED TO THE FACULTY  
OF PRINCETON UNIVERSITY  
IN CANDIDACY FOR THE DEGREE  
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE BY  
THE DEPARTMENT OF  
COMPUTER SCIENCE

Adviser: Arvind Narayanan

January 2022

© Copyright by Malte Möser, 2021.

All rights reserved.

# Abstract

Cryptocurrencies like Bitcoin balance privacy and transparency goals. In these systems, all transactions are public by design, allowing nodes in a decentralized network to validate them. However, this reveals many potentially privacy-sensitive transaction details of users, whose privacy is only protected by varying degrees of pseudonymity. Understanding the privacy cryptocurrencies provide in practice is hence important for users transacting in them as well as law enforcement or regulatory agencies concerned about their illicit use.

This thesis explores privacy in cryptocurrencies with differing transparency goals. We find that cryptocurrency privacy is inherently interdependent, as some users' behavior can negatively affect other users' privacy. Furthermore, privacy-sensitive disclosures from a few users can impact the privacy provided by the system at large. Yet, transparency need not come at the expense of privacy, and we show how it could be useful to impede illicit activity.

First, analyzing the early use of the privacy-focused cryptocurrency Monero, we find that some users' decision to forgo optional privacy protections obfuscating which coins they spend actively reduced those privacy provisions for others who opted for increased privacy. Furthermore, the obfuscation mechanisms employed by the Monero wallet did not match users' actual behavior well, negatively affecting the privacy of all transactions.

Next, we turn to address clustering, a blockchain analysis technique essential to understanding how Bitcoin is used in practice. As Bitcoin users can use many addresses, increasing their privacy, address clustering aims at revealing all addresses under a user's control. However, current techniques haven't been rigorously evaluated or optimized. Using ground truth data extracted from the blockchain

based on users' privacy-compromising behavior, we build new models to identify users' addresses with high accuracy and use them to create enhanced clusterings.

Finally, we consider a system intentionally designed to aid law enforcement goals of deterring money laundering and other financial crime. Public blacklists of coins involved in illicit activity can incentivize individual users to reject payments that can be traced back to listed coins, impeding money laundering attempts. We discuss how blacklisting would change the Bitcoin ecosystem and provide a theoretical and empirical evaluation of different tainting policies.

# Acknowledgments

I am grateful to my adviser, Arvind Narayanan, for his advice and guidance during my time in Princeton. His steadfast belief in my abilities encouraged me to use the freedom he provided to pursue the projects that were most interesting to me. Thank you for your support, and for prioritizing my personal well-being over any academic or work-related goals.

My life would look very different today and I probably would have never even dreamed of coming to Princeton if I hadn't met Rainer Böhme during my undergraduate studies in Germany. It was his encouragement, over beers and pizza in a small restaurant in Münster, that made me seriously consider applying for graduate school in the US. Thank you for always being available when I needed help or feedback, and for your continued mentorship and friendship over all these years.

I'm very grateful to Nicolas Christin, Jonathan Mayer and Matt Weinberg for completing my committee. Throughout the various stages of my PhD, I was often inspired by their research and greatly benefited from their insights and feedback.

I'm also thankful to Ed Felten and Nick Feamster for providing me with unique teaching experiences and advice, to Andrew Miller for his help and guidance when we collaborated on my first project here at Princeton, to Matt Salganik for many interesting sessions learning about computational social science, to Emin Gün Sirer and Ittay Eyal for inviting me to Cornell in the fall of 2015, which ultimately convinced me to come to the US, and to Matthias Kirchner for his advice on moving to and living in the US.

One of the most enjoyable aspects of my PhD was working together with and learning from other researchers. I'm grateful to have collaborated with Rainer Böhme, Alishah Chator, Nicolas Christin, Steven Goldfeder, Henry Heffan, Ethan

Heilman, Jason Hennessey, Kyle Hogan, Harry Kalodner, Kevin Lee, Patrick McCorry, Andrew Miller, Martin Plattner, Kyle Soska, and Shashvat Srivastava.

I'm also grateful to the many office-mates and colleagues I've had in Princeton and at the Center for Information Technology Policy (CITP) who made this time truly unique and special. Thank you, Kevin, for many fun afternoons working on our papers together, for our regular get-togethers, in person or virtually, and for showing me around NYC. Thank you, Ryan and Ben, for all the fun we've had in the office and for sharing memorable teaching experiences. Thank you, Harry and Steven, for your help and guidance during my early years in Princeton. Thank you, Danny, for our discussions on my work and for filling our lab with joy by bringing Momo to the office. Thank you to everyone else who resided in 312 Sherrerd Hall during this time, you all made this time truly special.

I'd also like to thank all the people in Princeton who helped and supported me during my time here. I'm especially grateful to Nicki Mahler and Judy Farquer in the Department of Computer Science for helping me navigate the various administrative hurdles and processes, Laura Cummings-Abdo and Jean Butcher at CITP who made this place a home, Katie Sferra and Regina George at the Davis International Center as well as Lily Secora at the Graduate School. Finally, I'm beyond grateful to Lisa Moscatiello and Lienna Wilson for supporting me during the COVID-19 pandemic.

Outside of Princeton, I'm grateful to Seth Connors for providing me with a memorable internship experience at Deloitte as well as for his continuing advice and mentorship.

Not less important than the academic side of my graduate school experience were the non-academic connections I forged. I'm grateful to Angela Quinn for introducing me to the yoga practice and local community, and to Vineet Chander for taking me and eleven other students on an immersive journey through India. And

I'm beyond grateful to all the friends I made, here or before, that all contributed to and shaped this experience in unique ways: Andrew, Carmen, Johanna, Kevin, Lydia, Matheus, Patrick, Svetlana, Tristan, Vanessa, and everyone who was part of the Yoga and Meditation Fellowship program.

Thank you, Cemil, for being a friend and constant I could rely on during this entire time. I'll miss our campus walks and late night conversations.

Thank you, Sinem, for sharing a big part of this experience with me, for listening and being there for me during hard times, and for all the wonderful moments we've enjoyed together. I won't forget you.

Thank you, Jeni, for filling my life with love and joy.

My biggest source of courage and optimism far away from home was knowing that I could always go back. I'm forever grateful to my family for their limitless support. Thank you—Christel, Jürgen and Bodo—for all the joy you've brought into my life. Clara, I love you little sister and hope to hug you again soon. Mom and dad, I lack the words to express how grateful I am for your unconditional support and love. This wouldn't have been possible without you. I love you both.

The work in this thesis was supported by NSF grants CNS-1421689 and CNS-1651938 and a grant from the Ripple University Blockchain Research Initiative.

To my parents.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Acknowledgments</b>	<b>v</b>
<b>List of Tables</b>	<b>xv</b>
<b>List of Figures</b>	<b>xvii</b>
<b>Bibliographic Notes</b>	<b>xxi</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Motivation . . . . .	2
1.2. Takeaways . . . . .	6
1.2.1. Cryptocurrency privacy is interdependent . . . . .	6
1.2.2. Tyranny of the blockchain minority . . . . .	7
1.2.3. Transparency and privacy are not at odds . . . . .	8
1.3. Overview . . . . .	8
1.3.1. Chapter 3: Traceability in Monero . . . . .	9
1.3.2. Chapter 4: Address clustering in Bitcoin . . . . .	9
1.3.3. Chapter 5: Blacklist-based cryptocurrency regulation . . . . .	10
<b>2. Background</b>	<b>12</b>
2.1. Bitcoin . . . . .	13
2.1.1. Transactions and addresses . . . . .	13
2.1.2. Blockchain and consensus . . . . .	15
2.1.3. Protocol and software upgrades . . . . .	16

2.2.	Privacy in Bitcoin . . . . .	17
2.2.1.	Addresses provide pseudonymity . . . . .	17
2.2.2.	Traceability: Transactions link recipients and senders . . . . .	18
2.2.3.	Address reuse and clustering . . . . .	19
2.2.4.	External linkage of identity to cluster . . . . .	20
2.2.5.	Countermeasures . . . . .	20
2.3.	Monero . . . . .	22
2.3.1.	Stealth Addresses . . . . .	22
2.3.2.	Ring signatures . . . . .	22
2.3.3.	Confidential transactions . . . . .	23
2.4.	BlockSci . . . . .	24
2.4.1.	Data format and parser . . . . .	24
2.4.2.	Address deduplication . . . . .	25
2.4.3.	Analysis interface . . . . .	26
2.4.4.	Clustering . . . . .	26
<b>3.</b>	<b>An Empirical Analysis of Traceability in the Monero Blockchain</b>	<b>28</b>
3.1.	Introduction . . . . .	28
3.2.	Background . . . . .	35
3.3.	Deducible Monero Transactions . . . . .	39
3.3.1.	Implementation . . . . .	41
3.3.2.	Results on deducible transactions . . . . .	41
3.4.	Tracing With Temporal Analysis . . . . .	43
3.4.1.	Effective-Untraceability . . . . .	43
3.4.2.	The Guess-Newest heuristic . . . . .	44
3.4.3.	Monte Carlo simulation . . . . .	46

3.5.	Characterizing Monero Usage . . . . .	48
3.5.1.	Quantifying mining activity . . . . .	48
3.5.2.	Usage on online anonymous marketplaces: the AlphaBay case	51
3.6.	Countermeasures . . . . .	54
3.6.1.	Fitted mixin sampling distribution . . . . .	54
3.6.2.	Binned mixin sampling . . . . .	57
3.7.	Discussion and Recommendations . . . . .	62
3.7.1.	Criminal uses of Monero . . . . .	63
3.7.2.	Recommendations . . . . .	64
3.7.3.	Subsequent impact . . . . .	66
<b>4.</b>	<b>Resurrecting Address Clustering in Bitcoin</b>	<b>68</b>
4.1.	Introduction . . . . .	68
4.2.	Building a Ground Truth Data Set . . . . .	74
4.2.1.	Data collection and overview . . . . .	78
4.2.2.	Refining the candidate set of ground truth transactions . . . .	79
4.2.3.	Assessing the final set of ground truth transactions . . . . .	80
4.3.	Evaluating Individual Change Heuristics . . . . .	85
4.3.1.	Background on change address detection . . . . .	85
4.3.2.	Evaluating individual heuristics . . . . .	89
4.4.	Combining Heuristics . . . . .	91
4.4.1.	Threshold vote . . . . .	92
4.4.2.	Random forest classifier . . . . .	92
4.4.3.	Additional model validation . . . . .	96
4.5.	Clustering Change Outputs . . . . .	97
4.5.1.	Naive merging leads to cluster collapse . . . . .	98
4.5.2.	Constraints prevent cluster collapse . . . . .	99

4.6.	Impact on Blockchain Analyses . . . . .	103
4.6.1.	Increased cashout flows from darknet markets to exchanges .	103
4.6.2.	Improved estimate of velocity . . . . .	104
4.6.3.	Comparison to the Meiklejohn et al. heuristic . . . . .	104
4.7.	Discussion . . . . .	106
4.8.	Summary . . . . .	106
<b>5.</b>	<b>Effective Cryptocurrency Regulation Through Blacklisting</b>	<b>108</b>
5.1.	Introduction . . . . .	108
5.2.	Combating Money Laundering in Bitcoin through Blacklisting . . . .	111
5.2.1.	Background: AML regulation in the US . . . . .	111
5.2.2.	Existing AML regulation's focus on accounts is ineffective in cryptocurrencies . . . . .	113
5.2.3.	The effectiveness of regulating exchanges is limited and users are at risk of accepting money they cannot spend . . . . .	114
5.2.4.	Public blacklists of funds derived from illicit activity make AML more effective and protect innocent users . . . . .	116
5.3.	How Blacklisting Would Work . . . . .	118
5.3.1.	Recursive blacklisting . . . . .	120
5.3.2.	Legal grounds . . . . .	122
5.3.3.	Blacklist governance . . . . .	123
5.3.4.	Adding an entry to the blacklist . . . . .	124
5.3.5.	Regulated intermediaries . . . . .	126
5.3.6.	Making and accepting payments . . . . .	127
5.4.	Managing the Risk of Future Blacklisting . . . . .	129
5.4.1.	Time-delayed payments . . . . .	130
5.4.2.	Risk scoring . . . . .	132

5.4.3.	Payment networks . . . . .	133
5.4.4.	Identity and insurance . . . . .	135
5.5.	Blacklisting Policies . . . . .	136
5.5.1.	Local characteristics . . . . .	138
5.5.2.	Global characteristics . . . . .	140
5.5.3.	Technical characteristics . . . . .	142
5.5.4.	Empirical evaluation . . . . .	143
5.5.5.	Discussion . . . . .	146
5.6.	Blacklisting and Privacy . . . . .	147
5.6.1.	Compatibility with privacy techniques . . . . .	147
5.6.2.	Incompatibility with privacy techniques . . . . .	148
5.6.3.	Towards regulation that balances privacy with regulatory and investigatory needs . . . . .	151
5.7.	Discussion of Common Concerns . . . . .	153
5.7.1.	Concern: Blacklisting destroys Bitcoin . . . . .	153
5.7.2.	Concern: Blacklisting destroys privacy . . . . .	154
5.7.3.	Concern: Criminals will use anonymous cryptocurrencies instead . . . . .	155
5.7.4.	Concern: Blacklisting will turn into whitelisting . . . . .	156
5.7.5.	Concern: Blacklisting can be avoided by moving coins across chains . . . . .	157
5.7.6.	Concern: Blacklists can be abused for political censorship . . . . .	158
5.8.	Summary . . . . .	158
<b>6.</b>	<b>Conclusion</b>	<b>160</b>
	<b>Bibliography</b>	<b>162</b>

<b>A. Appendix to Chapter 3</b>	<b>183</b>
A.1. Deducibility Attack as a SAT problem . . . . .	183
A.2. An Analysis of Bytecoin . . . . .	184
<b>B. Appendix to Chapter 4</b>	<b>187</b>
B.1. Protocol characteristics used for fingerprinting . . . . .	187
B.2. Additional plots and tables . . . . .	189
B.3. Further insights and technical details . . . . .	191
B.3.1. Filtering the ground truth data set . . . . .	191
B.3.2. Additional details on classification . . . . .	194
<b>C. Appendix to Chapter 5</b>	<b>197</b>
C.1. Blackmail Scam Addresses . . . . .	197
C.2. Extended Taint Analysis . . . . .	199

# List of Tables

1.1. Difference between the concepts of transparency, pseudonymity, anonymity and privacy. . . . .	3
3.1. Traceability of Monero transaction inputs with 1+ mixins. . . . .	30
3.2. Monero transaction inputs where the real input can be deduced . . .	40
3.3. Percentage of deducible transaction inputs where the real input is the “newest” input. . . . .	45
3.4. Min-untraceability for different bin sizes and mixins . . . . .	61
4.1. Comparison of transaction characteristics between ground truth transactions and transactions with 2 outputs for which change is unknown. . . . .	83
4.2. Change heuristics proposed in the literature and used in this work. .	84
4.3. True and false positive rates of heuristics applied to transactions in the ground truth data set. . . . .	88
4.4. Transaction counts of smaller clusters being merged. . . . .	101
5.1. How blacklisting would change the Bitcoin ecosystem . . . . .	119
5.2. Characteristics of different blacklisting policies . . . . .	143
5.3. Total number of outputs tainted, transactions traversed and percentage of tainted value ending up in transaction fees for different datasets on 06/31/2020 . . . . .	144
A.1. Bytecoin transaction inputs (with 1 or more mixins, at least 1000 TXOs available) where the real input can be deduced. . . . .	185

A.2. Percentage of deductible Bytecoin transaction inputs where the real input is the “newest” input. . . . .	186
B.1. Increase in outgoing transaction volumes of darknet markets to exchanges using the base clustering (before) and our enhanced clustering (after). . . . .	191
B.2. Characteristics of clusterings created with the Meiklejohn heuristic in comparison to our constrained clustering . . . . .	192
B.3. Number of transactions (in million) in our ground truth data set with fresh or reused spend and change outputs. . . . .	195
C.1. Number of outputs tainted by FIFO, and total number of tainted chunks despite merging adjacent chunks, with different datasets on 06/31/2020 . . . . .	199
C.2. Average number of outputs tainted by applying the heuristic to each individual output in the dataset separately and propagating taint throughout the next 2018 blocks, starting from the output’s block height (2 weeks) . . . . .	199



# List of Figures

1.1. Spectrum of different transparency goals in digital currencies . . . . .	5
2.1. Schema of a typical Bitcoin transaction with two inputs and two outputs. . . . .	14
2.2. Schema of the blockchain. . . . .	15
2.3. State of privacy preserving techniques for cryptocurrencies in 2017 .	21
2.4. Input referencing in Monero . . . . .	22
3.1. Transactions and tracing in Bitcoin and Cryptonote. . . . .	29
3.2. Age distributions of Monero mixins . . . . .	29
3.3. Data considered in our experiment. . . . .	37
3.4. 0-mixins effectively reduce the untraceability of other transactions. .	39
3.5. Fraction of transaction inputs that can be deduced and transactions including at least one deducible input . . . . .	40
3.6. Transaction inputs are less likely to be deducible if they have more mixins and if they are found among later blocks in the Monero blockchain. . . . .	43
3.7. Estimated vulnerability to the Guess-Newest heuristic for varying sampling policies in Monero . . . . .	47
3.8. Fraction of blocks created by mining pools for which payment trans- actions are made public and have been collected. . . . .	49
3.9. Comparison of the volume of estimated non-mining transactions to the overall transaction volume. . . . .	51
3.10. Estimated AlphaBay sales volume since inception . . . . .	52
3.11. CDFs of spend-time distributions in Bitcoin and in Monero . . . . .	55

3.12. Projection of the Guess-Newest vulnerability, and the improvement due to our proposed scheme. . . . .	58
3.13. Binned mixin sampling, an input spends an output (red dotted line) and references outputs in two bins. . . . .	60
4.1. Schema of a typical Bitcoin transaction with two inputs and two outputs. . . . .	69
4.3. Our process in this chapter . . . . .	74
4.4. Multi-input clustering reveals the change address . . . . .	75
4.5. Distribution of different types of transactions in the Bitcoin blockchain until June 2021 . . . . .	79
4.6. Share of ground truth transactions of all and standard transactions over time. . . . .	81
4.7. Number of base clusters represented in our ground truth by total address count. . . . .	82
4.8. Number of transactions in ground truth and full blockchain per base cluster. . . . .	82
4.9. Schema of how transactions are created, and how consistency of a transaction's fingerprint allows to identify change. . . . .	85
4.10. Time until both outputs of a transaction are spent. . . . .	87
4.11. Number of votes from heuristics on transactions in the ground truth data set . . . . .	90
4.12. Average number of correct and uncorrect votes per transaction and type of heuristic in the ground truth data set, over time . . . . .	91
4.13. ROC curves for predicting change in the ground truth data set using the threshold vote and the random forest classifier, compared to individual heuristics . . . . .	93

4.14. Comparison of the random forest classifier and the threshold vote on the test set. . . . .	96
4.15. Our constrained clustering prevents the merging of clusters A and B due to conflicting types of payments between them. . . . .	99
4.16. Absolute increase in address and transaction count per affected cluster	102
5.1. The structure of the Bitcoin transaction graph allows to follow coins from one transaction to the next . . . . .	120
5.2. In a blacklisting regime users are at risk of receiving coins that might get blacklisted in the future [113] . . . . .	129
5.3. How different tainting policies propagate taint from inputs to outputs	137
5.4. Limited untraceability does not preclude taintability . . . . .	149
5.5. Applicability of taint policies given levels of anonymity in the cryp- tocurrency . . . . .	150
5.6. Framework for Effective Cryptocurrency Regulation . . . . .	152
A.1. Deducibility as a SAT problem . . . . .	184
B.1. Number of votes from heuristics (with compressed power-of-ten heuristic) . . . . .	189
B.2. Average number of correct and incorrect votes per transaction and type of heuristic, over time (with compressed power-of-ten heuristic)	189
B.3. Probabilities returned by the random forest classifier for standard transactions with unknown change . . . . .	190
B.4. Revised estimate of velocity . . . . .	190
C.1. Analysis of the Blackmail data set . . . . .	200
C.2. Analysis of the OFAC data set . . . . .	201
C.3. Analysis of the Ransomware data set . . . . .	202

C.4. Analysis of the Random Outputs data set . . . . .	203
--	-----

# Bibliographic Notes

Chapters 1 to 2 include material from the following publication:

- Arvind Narayanan and Malte Möser. “Obfuscation in Bitcoin: Techniques and politics”. In: *International Workshop on Obfuscation: Science, Technology, and Theory*. 2017

Chapter 3 was originally published in 2018:

- Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, and Nicolas Christin. “An Empirical Analysis of Traceability in the Monero Blockchain”. In: *Proceedings on Privacy Enhancing Technologies* 2018.3 (2018), pp. 143–163

A shorter version of Chapter 4 has been accepted for publication:

- Malte Möser and Arvind Narayanan. “Resurrecting Address Clustering in Bitcoin”. In: *Financial Cryptography and Data Security*. 2022, forthcoming

An earlier version of Chapter 5 was published as a preprint in 2019:

- Malte Möser and Arvind Narayanan. *Effective cryptocurrency regulation through blacklisting*. Preprint. 2019

The analyses in Chapters 4 to 5 are based on BlockSci, a high-performance blockchain analysis framework that I contributed to during my PhD:

- Harry Kalodner, Malte Möser, Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, and Arvind Narayanan. “Blocksci: Design and applications of a blockchain analysis platform”. In: *29th USENIX Security Symposium*. 2020, pp. 2721–2738

During my time in Princeton I furthermore co-authored the following publications:

- Malte Möser and Rainer Böhme. “The price of anonymity: empirical evidence from a market for Bitcoin anonymization”. In: *Journal of Cybersecurity* 3.2 (2017), pp. 127–135
- Patrick McCorry, Malte Möser, and Syed Taha Ali. “Why Preventing a Cryptocurrency Exchange Heist Isn’t Good Enough”. In: *Security Protocols XXVI: 26th International Workshop, Cambridge, UK, March 19–21, 2018, Revised Selected Papers*. Vol. 11286. Springer. 2018, pp. 225–233

While unrelated to my thesis, I enjoyed participating in the “Fragile Families Challenge” and am a co-author on the following publication:

- Matthew J. Salganik, and 111 others. “Measuring the Predictability of Life Outcomes with a Scientific Mass Collaboration”. In: *Proceedings of the National Academy of Sciences*. Volume 117, Issue 15: pp. 8398–8403.

# 1

## Introduction

A naive look at Bitcoin, the first and to this day most popular cryptocurrency, could lead to two contrary conclusions about the degree of privacy it provides. On the one hand, all transactions are publicly visible and stored indefinitely, a degree of transparency that had so far been unheard of in the financial world. On the other hand, no personal identifiers are tied to those transactions, leading to initial claims of Bitcoin being a completely anonymous payment system. These two contradictory interpretations raise the question of how much financial privacy users actually enjoy, and how useful the high degree of transparency is for law enforcement entities investigating illicit activity. In this thesis, we explore the privacy offered by cryptocurrencies on a spectrum of differing transparency goals. Our analyses of two different cryptocurrencies, Bitcoin and Monero, reveal systemic interdependent privacy risks stemming from users' behavior as well as recurring risks of large-scale privacy compromise due to the privacy-sensitive disclosures of a small set of users. Finally, we design a system aiding law enforcement to curb illicit activity through increased transparency around illegal activity, and explore its implications for the ecosystem.

## 1.1. Motivation

Cryptocurrencies like Bitcoin are payment systems that allow users to conduct financial transactions without relying on a centralized intermediary. Instead, the Bitcoin protocol publicly distributes all transactions, enabling an open, decentralized network of nodes to check their validity. All valid transactions are eventually recorded in an append-only public database, the *blockchain*, of which every node stores a copy. This high level of *transparency* was not a goal in the design of Bitcoin, rather, it was necessary to achieve security in the decentralized protocol [46].

Making users' transactions publicly visible to any outside observer meanwhile exposes many potentially privacy-sensitive transaction details. Whether this disclosure has actual privacy implications for individual users depends to some degree on whether their real identity can be linked to their activity on the blockchain. In a perfectly *anonymous* system, an observer would be unable to link any particular transaction to any particular individual or other transaction, and thus the privacy impact would be low (cf. [129]).

To protect individuals from being trivially linked to their on-chain activity, Bitcoin uses *pseudonymous* account identifiers (commonly called *addresses*). An address is a unique cryptographic identity that allows users to send and receive funds, while hiding their real identity. To assess individuals' privacy risks, we need to consider the degree of anonymity these pseudonyms can provide.

Every Bitcoin transaction inherently links the sender's and recipient's addresses to each other. When users reuse their addresses across multiple transactions, those transaction can be linked to the same sender or recipient. For this reason, users are encouraged to create new pseudonyms for each transaction, but address reuse is still a common phenomenon [79]. Even if users split their activity among different address, spending from multiple addresses in the same transaction links them to a



**Table 1.1.:** Difference between the concepts of transparency, pseudonymity, anonymity and privacy.

	Definition	In Bitcoin
Transparency	Degree to which contextual information is available [47]	All transactions, including values and pseudonymous account identifiers, are publicly available on the blockchain
Pseudonymity	Use of a pseudonym as an identifier [129]	Bitcoin addresses act as pseudonyms
Anonymity	State of being indistinguishable from other users, not linkable to items of interest [129]	No true anonymity possible due to inherent linkability of transactions, and possibility to link pseudonyms of a user
Privacy	An emergent property stemming from the system design and user behavior, including users' expectations and degree of control user has	Apparent balance between users' desired level of privacy and law enforcement needs to understand activity

common owner [103, 135, 137]. Crucially, if only a single address in such an *address cluster* can be linked to a user's real identity, their entire financial activity on the blockchain can potentially be discovered. Such a link can easily be established when someone uses bitcoins to purchase goods or services (for example, a merchant may need a shipping address for delivery of physical goods).

The transparency of Bitcoin in itself hence does not constitute the privacy risk, but it enables linking transactions to people due to Bitcoin's (imperfect) pseudonymous nature. Financial *privacy* in cryptocurrencies thus is an emergent property that depends not only on the system's design but also on users' actions and behavior and the degree to which it leads to their activity becoming linkable (cf. Table 1.1).

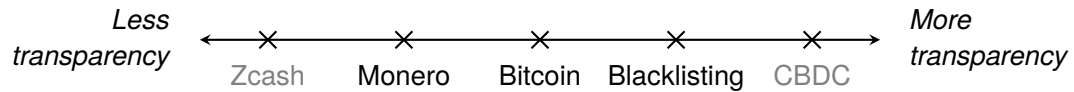
As a result, protecting users privacy can take place on different levels: by making less data available (which is difficult in cryptocurrencies, as protocols are already designed with efficiency in mind), protecting sensitive data using cryptographic

techniques where possible, as well as making interpretation of data harder through obfuscation.

Recognizing the privacy limitations of Bitcoin, a number of privacy-enhancing techniques have subsequently been proposed and developed (e.g., [17, 27, 43, 105, 109, 110, 139, 157]). Many of these techniques try to make interpretation of transactions and subsequent linking harder, but most do not provide provable privacy guarantees. Instead, they rely on obfuscation [28] to reduce the amount of useful information an observer is able to extract from the blockchain.

Adoption of privacy-enhancing techniques in Bitcoin has been limited, both for technical and political reasons [119]. Technically, strong privacy techniques require the use of heavier cryptographic tools, adding significant performance penalties that are undesirable for an already performance-constrained system (e.g., [44]). On the political side, the current degree of transparency has been useful in aiding law enforcement investigations in the system (e.g., [31]). Law enforcement's subpoena power and the use of specialized analysis tools allows them to establish links in select scenarios while retaining privacy for most users [119]. Increasing the baseline privacy of the system could instead lead financial regulators to consider stronger oversight or to impose stricter anti-money laundering regulation, something the cryptocurrency ecosystem has so far successfully lobbied to avoid (cf. [156]).

Strong privacy-enhancing technologies have instead been deployed in a few alternative cryptocurrencies, two prime examples of which are Monero [157] and Zcash [17]. Monero employs a combination of cryptographic techniques to prevent linking known addresses to on-chain activity, using obfuscation to obscure which funds a transaction is spending, as well as encrypting transaction amounts. Zcash hides sensitive transaction information such as payment amounts and relies on zero-knowledge cryptography to completely hide which funds a transaction is spending.



**Figure 1.1.:** This thesis explores a spectrum of different transparency goals in digital currencies.

Importantly, the cryptocurrencies reduce transparency to outside observers but retain the possibility for users to share such information on a voluntary basis (for example, for compliance or audit purposes). This increases users' privacy, but makes it harder for law enforcement to investigate and prosecute illicit activity.

While Bitcoin seemingly provides a degree of transparency helpful to investigate major illicit activity, it does not constitute the upper end on the transparency spectrum. In Ethereum [162], users can create complex programs (so-called smart contracts) that specify how and when funds can be spent. These are transparently stored and executed on the blockchain. Another scenario where higher degrees of built-in transparency may be desired are central bank digital currencies (CBDC). Over the past years, there's been increasing interest on the side of central banks around issuing a digital currency as a new form of central bank issued money [68, 128]. A CBDC could increase efficiency for payments and provide a substitute for the declining use of cash [128]. Today, China is already actively field-testing a digital version of the Yuan [9]. Such systems would provide central banks control over the monetary supply and transaction validation (e.g., [45]) and would likely be designed with existing financial regulation in mind, such as compliance with anti-money laundering (AML) regulation [68]. This highlights the potential for the development of protocols that are more tailored towards regulatory needs.

It is in this broader context that this thesis contributes new insights into the transparency and privacy provisions of cryptocurrencies and the challenge of balancing privacy and law enforcement needs. We explore a spectrum of cryp-

tocurrency designs with varying transparency goals (see Figure 1.1). In Chapter 3, we investigate the early use of Monero, a cryptocurrency specifically designed to increase financial privacy [157]. We find striking examples of user behavior that puts the privacy of other users at risk and evaluate its impact. In Chapter 4, we look at the current state of address clustering in Bitcoin, a common deanonymization technique to group addresses under the control of individual users. Enabled by a novel approach to collect ground truth data, we build new models that can identify addresses that users create for the purpose of receiving change with high accuracy. Our results are important for scientific analyses and law enforcement investigations alike. Finally, recognizing that increased transparency does not necessarily result in reduced privacy, in Chapter 5 we consider a system specifically designed to aid law enforcement efforts. By increasing transparency around illicit activity using public blacklists, regulators can disincentivize financial crime while maintaining Bitcoin’s current level of financial privacy.

## **1.2. Takeaways**

### **1.2.1. Cryptocurrency privacy is interdependent**

A recurring theme throughout our work is that users’ choices and resulting privacy are inherently interdependent (cf. [13, 18]). The actions of some users, whether intentional or accidental, can negatively affect the privacy of other users, even if those took steps specifically to protect their own privacy. For example, some Monero users’ decisions to forgo privacy eroded the privacy of users who opted for privacy protections through an iterative deducibility attack (Chapter 3). And in Bitcoin, a recipient’s choice of wallet and use of protocol features often allows us to identify the sender’s change address with high accuracy (Chapter 4).

This interdependence has important implications for cryptocurrency protocol designers. By definition, individual users' ability to protect against interdependent privacy risks is limited because they depend on the actions of others. Particular care must thus be taken towards standardizing wallet behavior and choosing defaults, especially for privacy-sensitive use cases, as users cannot be expected to understand the implications their choices have on others. And the availability of optional privacy protections may provide users a false sense of security if their privacy provisions are rendered less effective by privacy-compromising behavior of others.

### **1.2.2. Tyranny of the blockchain minority**

Not only may the actions of some users directly affect the privacy of others, but often privacy-sensitive disclosures stemming from a small set of users can affect the privacy that the system can provide at large. For example, to identify Bitcoin change addresses in Chapter 4, we build a machine learning model based on known change outputs from a subset of users that can be applied to the blockchain at large. This phenomenon has previously been observed in the general context of big data: the disclosures of a small set of users (for example, regarding their shopping preferences) can be sufficient to infer privacy-sensitive information about other users through statistical inference [14].

In cryptocurrencies without fully provable privacy guarantees, this calls for caution with regards to privacy claims for a specific system or use case. Claims made purely based on protocol design may overestimate privacy, and the validity of these claims can only be evaluated empirically based on actual user behavior and by taking into account what information an adversary could already possess from studying other users. To identify potential new issues and evaluate previously unanticipated uses cases and behavior, cryptocurrency developers should continually monitor the use

of their systems and incentivize researchers to perform independent measurement studies on their own.

### **1.2.3. Transparency and privacy are not at odds**

In Chapter 5 we discuss a system design with a focus on aiding law enforcement goals through increased transparency. By introducing public blacklists of illicit coins, regulators can create incentives for users to check coins they receive in a transaction. Such a system would increase transparency around illicit activity and can exist completely without changes to the underlying cryptocurrency protocol: the traceability of transactions and the risk of receiving illicit funds from others are sufficient to incentivize users to check the blacklists. At the same time, it retains most existing privacy provisions and remains effective even if certain obfuscation techniques aimed at thwarting address clustering were deployed.

As cryptocurrencies have matured and gained popularity with the general public as well as traditional financial institutions, and as regulators are building out their capacity to investigate financial crime (e.g., [49]), increasing the transparency around illicit activity could help cryptocurrencies achieve further general acceptance in the financial sector. Developers and regulators should explore solutions that do not require extensive financial surveillance (cf. [154]). Blacklisting could be one such tool to maintain balance in the tug-of-war between financial privacy and law enforcement needs.

## **1.3. Overview**

We organize this thesis along the spectrum of transparency depicted in Figure 1.1. Chapter 2 provides the necessary background and we conclude in Chapter 6.

### **1.3.1. Chapter 3: Traceability in Monero**

Monero is a privacy-preserving cryptocurrency based on the Cryptonote protocol [157]. It uses cryptographic tools to prevent address clustering and to obfuscate the coins that are being spent by a user. In our analysis of the early use of the Monero blockchain in Chapter 3, we found two striking examples of interdependent privacy that put users at risk.

Monero provides users the possibility to obfuscate the origin of the coins they are spending by including chaff coins (thereby providing plausible deniability). Initially, users were given the choice to forgo these privacy protections. This had disastrous consequences for other users. We show how it enables an iterative deducibility attack, allowing us to determine the true coins spent in transactions of users who opted for increased privacy, by ruling out coins that were already provably spent.

This attack revealed a second, severe privacy vulnerability: the age distribution of coins chosen by the wallet to obfuscate the true spend severely mismatched users' actual spend time distributions. This allows us to guess the true output spent for other transactions as the most recent one with high (estimated) accuracy, even for transactions unaffected by our initial attack. Crucially, learning privacy-relevant information from only a share of users allows us to generalize to the system at large, putting all users at risk of deanonymization.

### **1.3.2. Chapter 4: Address clustering in Bitcoin**

While address clustering techniques are a cornerstone of Bitcoin blockchain analysis, current techniques haven't been rigorously evaluated or optimized. We use a novel method to extract ground truth data from the Bitcoin blockchain to evaluate existing address clustering heuristics. This enables us to build new models to identify change outputs and their associated addresses for subsequent address clustering.

Change address heuristics are enabled by the fact that wallet and user behavior will differ based on whether the output is a payment or a spend output returning surplus funds. Yet another example of interdependent privacy, we can often identify spend outputs based on other users' behavior. For example, we can eliminate outputs that don't match the senders address type, or those where the spending transaction's characteristics differ from the original transaction.

Equipped with ground truth data based on users' privacy-compromising behavior (namely address reuse), we build a machine learning model that generalizes from our ground truth and allows us to predict the change output in other users' transactions. Crucially, this also impacts users who take care to not reuse addresses (and thus don't show up in our ground truth). Again, privacy-sensitive disclosures stemming from a small share of users affects the privacy of the system as a whole.

### **1.3.3. Chapter 5: Blacklist-based cryptocurrency regulation**

Finally, we consider a system intentionally designed to aid law enforcement goals of deterring money laundering and other financial crime. Currently, knowledge about transactions facilitating financial crimes is mostly unavailable, residing with intermediaries and law enforcement agencies only. This creates a risk for normal users transacting outside of regulated platforms to receive funds they cannot spend.

We consider a system that provides increased transparency around illicit use through the provisioning of public blacklists. If information about transactions involved in illicit activity were publicly available, and policies around how to treat subsequent transactions were standardized, this now quantifiable risk of accepting illicit coins would incentivize individual users to check public blacklists before accepting coins from others.



A major challenge in such a system is to design policies around the interdependent aspects, such as checking blacklists when receiving funds, negotiating transaction arrangements or how to handle funds traceable to illicit activity. We discuss how blacklisting would change the Bitcoin ecosystem and provide both a theoretical and empirical comparison of different taint policies for blacklisting.

# 2

## Background

Since the launch of Bitcoin in 2009, cryptocurrencies have grown from a niche interest of cypherpunks and computer scientists [42] into a global phenomenon. As of October 2021, their total market capitalization exceeds 2 trillion U.S. dollars [39]. There are twenty different cryptocurrencies alone with a market capitalization of more than USD 10 billion, led by Bitcoin with USD 1.15 trillion (at an exchange rate of around USD 60 000) and the smart contract-focused Ethereum with USD 440 billion. And over the past nine years, from 2012 to 2021, investors have poured more than USD 38 billion of venture capital into the ecosystem [90].

At the same time, cryptocurrencies' pseudonymity and the initial unfamiliarity on the side of regulators and law enforcement agencies made them attractive for criminal activity, including their use on darknet markets, for scams or to collect ransomware payments [37, 84, 125, 146]. The company Chainalysis, which provides compliance and blockchain monitoring products to governments, law enforcement and financial intermediaries, reported that in 2019 identified illicit activity accounted for around USD 21.4 billion, or around 2.1 % of all cryptocurrency transfer value [33]. As a result, cryptocurrencies are increasingly becoming the target of regulators (e.g., [63]) and relevant in day-to-day political discussions (e.g., [30]).

In this chapter, we provide relevant background for the remainder of this thesis.

## 2.1. Bitcoin

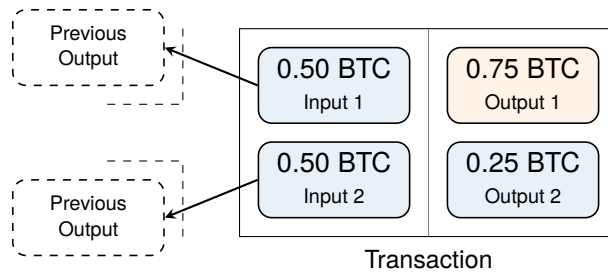
Cryptocurrencies like Bitcoin [118] are decentralized transaction systems that facilitate payments without relying on a central intermediary for coordination and security. Instead, they are based on an open, decentralized peer-to-peer network of nodes that keeps track of a public, append-only data structure called the *blockchain*. The blockchain stores transactions that have been validated for correctness and confirmed by the network. As the network is decentralized, a consensus mechanism ensures that invalid transactions are rejected and that all nodes (eventually) agree on a correct state of the chain.

### 2.1.1. Transactions and addresses

Bitcoin users transact using pseudonymous account identifiers called addresses. A transaction effectively reassigns a quantity of currency from one set of addresses to another set of addresses. Addresses work differently from accounts in that the system does not accumulate bitcoins sent to an address to a total balance, but instead treats them like separate coins with individual denominations.

Every Bitcoin transaction contains a set of inputs and outputs (see Figure 2.1). Each transaction *output* (TXO) assigns value to a specific address. An input refers to a previously created unspent output (UTXO) that is then spent by the transaction. A transaction needs to spend all the value provided in the inputs. For this reason, most transactions have one additional output constituting the change, which is sent back to an address under the payer's control.

Technically, an address is a part of a script that specifies how coins can be redeemed. Each output contains such a script, most of which pose a cryptographic challenge to the recipient in order to redeem it. If a user receives multiple payments



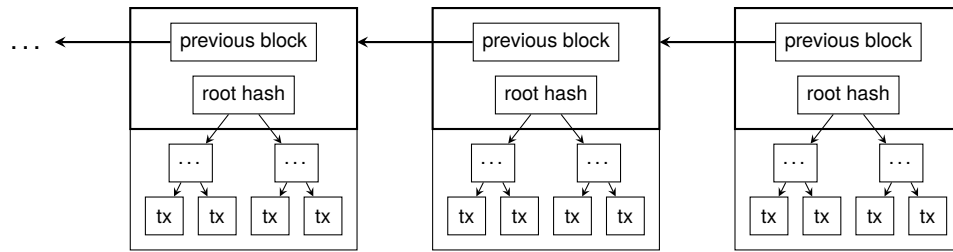
**Figure 2.1.:** Schema of a typical Bitcoin transaction with two inputs and two outputs: the spend output (1) is the intended payment, the change output (2) returns the surplus coins to the sender. Each input and output is associated with an address. Each input refers to the output of a previous transaction that they are spending.

to the same address, these will be distinct UTXOs and a separate input is required to spend each of them.

The most common script used specifies a hash value of a public key (i.e. the address). To spend the output, the recipient has to provide both a public key that corresponds to the hash value as well as a digital signature created with the corresponding private key proving ownership of the key pair. This script type is commonly called “pay to pubkey hash” (P2PKH).

To provide more flexibility around the scripts that can be encoded in an address, a second common script type allows to specify the hash value of an entire script. Then, not only valid signatures but also the full script must be provided by the input. This “pay to script hash” (P2SH) format is commonly used for multisignature transactions, where the valid signatures of more than one public key are required to redeem an output. Another benefit of P2SH is that it allows for backwards-compatible protocol upgrades, as new script types and formats can be encapsulated and only need to be supported by the recipient but not the sender.

The link between a transactions’ inputs and the outputs they are spending creates a directed, acyclic graph, which we call the *transaction graph* (see Figure 2.1). Connecting addresses based on the flow of value between them would instead



**Figure 2.2.:** Every block includes a pointer to the previous block header, thereby creating an authenticated chain of blocks. A Merkle tree is computed over all transactions in the block and the hash of the root node of the tree is included in the block header. This way, all transactions included in the blockchain are also authenticated.

produce an address graph. Finally, by identifying the addresses under the control of a single user we can create a user graph describing interactions between them.

### 2.1.2. Blockchain and consensus

After creating and signing a transaction, a user submits it to the peer-to-peer network for inclusion in the blockchain. Nodes in the network will verify the correctness of the transaction (e.g., that the signature is valid and the funds have not already been spent). Valid transactions then get queued to be included in a block.

Each block contains a Merkle tree computed over all transactions included in the block (Figure 2.2). The root hash of the Merkle tree gets included in the block header. Every block that extends the blockchain points to the block header of the previous block, creating an authenticated data structure.

As Bitcoin is a distributed system, a consensus mechanism is required to ensure that all users eventually share the same view of what is considered the “correct” state of the blockchain. Otherwise, a user could attempt to double-spend an output by creating two conflicting transactions that spend the output in different ways. As there is no central authority to decide which transaction is valid, the network has to vote on the correct transaction.

Bitcoin uses a consensus mechanism known as proof of work (PoW). PoW is designed to make it costly for an adversary to convince the network that a different set of transactions is valid than what it had already agreed on. The mechanism requires validators (so-called miners) to waste computational resources in order to create a valid block. Finding a valid block is a stochastic process and the amount of resources needed is adjusted such that on average every ten minutes a new block is found. In turn, a miner who successfully mines a block is rewarded with newly minted bitcoins (which they can claim in a special coinbase transaction) if the network accepts their block. Over time, as the blockchain grows, blocks anchored deeper in the chain become less likely to get replaced by a competing fork as a large amount of computational power would be necessary [59].

As space in blocks is limited by the protocol, transactions bid for inclusion by paying a transaction fee that is awarded to the miner of the block. The transaction fee is not explicitly recorded in a transaction. Instead, the difference between the sum of all input amounts minus the sum of all output amounts constitutes the fee. Users can freely choose transaction fees, trading off lower fees for an increased time it will take miners to process and include the transaction in a block. This has led to the emergence of a dynamic fee market [93, 111].

### **2.1.3. Protocol and software upgrades**

The decentralization of Bitcoin poses unique challenges to upgrade the network software. Protocol upgrades that involve the consensus rules (which includes, among other things, the functionality of the script opcodes) require a large share of the network nodes to adopt the software before new functionality can be activated. Otherwise, the network would fork into two chains: one supporting the new rules and one supporting the old ones.

An important protocol upgrade for Bitcoin was the activation of the “Segregated Witness” (SegWit) rules in 2017. SegWit essentially allows to store transaction signatures outside of a block (anchored in the coinbase transaction), thereby increasing the space available in blocks, allowing for more transactions to be included. The upgrade also introduced new, backwards-compatible address types that allow older clients to process payments that contain no signatures in the inputs.

Upgrades of users’ wallet software functionality is generally uncomplicated. Most major network upgrades to this day have been backwards-compatible, allowing users to continue using old wallets. And wallets can easily change behavior that is not relevant for the consensus mechanism, such as deciding how UTXOs are chosen when making a payment or which protocol features to use.

## **2.2. Privacy in Bitcoin**

Because of Bitcoin’s transparency, users’ transactions are visible to anyone who downloads a copy of the blockchain. Here, we discuss the privacy risks stemming from Bitcoin’s protocol design.

### **2.2.1. Addresses provide pseudonymity**

We start by laying out some common terminology around pseudonymity and anonymity (as put forward by Pfizmann and Köhntopp [129]). Pseudonymity refers to the use of a pseudonym as an identifier. The benefit of using a pseudonym over a personal identifier (such as a name or social security number) is that, at least initially, it cannot be tied to a particular person.

In Bitcoin, addresses constitute pseudonyms that allow users to receive and send payments. They protect them from having their financial activity on the blockchain

immediately linked to their real identity. However, pseudonymity on its own does not provide anonymity.

Anonymity is a state of indistinguishability, or non-identifiability, within a group of subjects that form an *anonymity set*. A critical implication of this definition is that a subject cannot achieve anonymity on their own. Only when they are indistinguishable from at least one other subject we can consider them anonymous.

Anonymity breaks down if an item of interest, such as a transaction, can be linked to a particular subject [129]. In Bitcoin, linkability occurs in a variety of ways.

### **2.2.2. Traceability: Transactions link recipients and senders**

Coins in Bitcoin have value because they are verifiably linkable, back to the transaction in which they were originally mined. Every transaction input needs to specify the origin of funds it spends, which it does in the form of a pointer to a previous transaction output. This links senders and recipients and prevents any true form of unlinkability. In the cryptocurrency community, this specific form of linkability is often referred to as *traceability*.

Traceability in Bitcoin is a double-edged sword. While the ability to look backwards at previous transactions can have privacy implications, it has arguably contributed to Bitcoin's widespread acceptance, including with regulators and law enforcement. Blockchain intelligence services utilize this transparency to identify funds originating from illicit sources, such as the wallet of a darkweb market or an entity sanctioned by the Office of Foreign Asset Control (OFAC) (e.g., [31]). Such services are widely used by intermediaries in the ecosystem and are an important component of their anti-money laundering (AML) compliance programs. In Chapter 5, we explore new ways for regulators to expand upon the usefulness of traceability to prevent illicit activity.



### 2.2.3. Address reuse and clustering

Address clustering techniques are a cornerstone of blockchain analysis. The goal of clustering is to identify addresses under the ownership of an individual by exploiting characteristics of the transaction graph as well as common wallet behavior.

The most common clustering technique is the *multi-input* heuristic [103, 135, 137]. When a transaction combines inputs from different addresses, these are likely under the control of a single user. For example, a user may receive small amounts of bitcoins at different addresses and then combine those in a transaction to transfer a larger amount of bitcoins than they received with any single address. The multi-input heuristic is relatively straightforward to apply, moderately effective in practice [79] and widely used. An important caveat is that the heuristic is vulnerable to false positives from techniques like CoinJoin and PayJoin transactions that intentionally break it by combining inputs from different users (e.g., [51, 99, 102, 110]).

The multi-input heuristic is especially effective if users reuse addresses. If for every payment they receive they used a fresh address, then the multi-input heuristic will only group the addresses found in inputs of a single transaction. If addresses are reused, and coins from one address are spent in different transactions, the transitivity of the heuristic clusters addresses from multiple transactions. Unfortunately, managing this risk can be challenging for users. While they can give out different addresses in different contexts, they cannot prevent multiple payments being made to the same address. In fact, an adversary could actively try to induce a user's wallet to link transactions in this way by making small payments to an address (a so-called "dusting" attack), hoping the wallet will use them jointly with other funds.

A second class of heuristics aims to identify the change output in a transaction, which is under the control of the user who created the transactions (and thus controls the addresses of the inputs). By clustering the change address, we essentially rely

on Bitcoin’s traceability to expand the cluster to the subsequent transaction and all addresses spent in its inputs. Change address clustering was pioneered by Meiklejohn et al. [103], who exploited a specific behavior of the wallet of the Bitcoin reference implementation to generate fresh addresses for change. Since then, there’s been little progress on change address clustering, primarily due to a lack of ground truth data. In Chapter 4, we address this limitation and explore new models to detect and cluster change addresses.

#### **2.2.4. External linkage of identity to cluster**

The final privacy risk for Bitcoin users comes from the possibility to link their real-world identity to an address or a cluster of addresses on the blockchain, thereby revealing their financial activity. Crucially, if only a single address of a user can be linked to them it could break the pseudonymity of all their transactions.

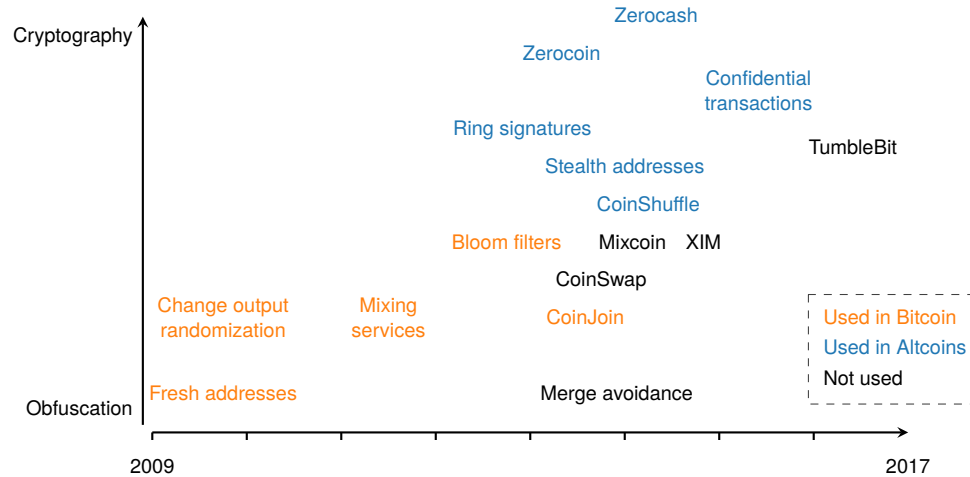
Linking address clusters to a person in this way may be possible whenever a user interacts with an intermediary that requires personally identifying information, such as a mailing address for physical goods or an email address to send a receipt. Furthermore, users may leak their identity when using Bitcoin addresses in online forums or other public venues. Conversely, intermediaries can be identified by interacting with them (e.g., [103]) and there even exist websites<sup>1</sup> solely dedicated to providing such information.

#### **2.2.5. Countermeasures**

As a remedy for the limited privacy Bitcoin provides, privacy-conscious users can employ obfuscation techniques to protect their financial privacy [119]. Obfuscation introduces uncertainty and makes it harder for an analyst to identify users and

---

<sup>1</sup>walletexplorer.com

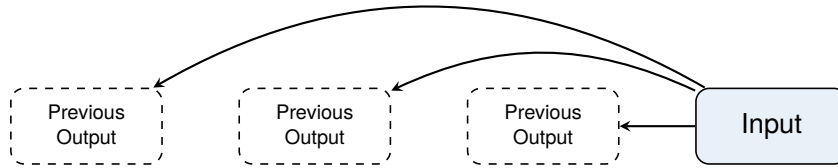


**Figure 2.3.:** State of privacy preserving techniques for cryptocurrencies in 2017

their transactions, but the techniques often do not come with quantifiable privacy guarantees.

Obfuscation techniques in Bitcoin generally fall into one of two categories. Ambiguating obfuscation increases uncertainty around on-chain activity. For example, to make it harder to detect change addresses users can randomize the index of the change output, create fresh addresses, or even create multiple outputs to obscure the transaction further. Cooperative obfuscation, on the other hand, tries to establish an anonymity set through cooperation of multiple users. For example, mixing services allow users to combine their funds, potentially breaking the link between incoming and outgoing payment [112].

When we surveyed the landscape of obfuscation techniques in 2017, few of the available or proposed techniques were used in Bitcoin, and most of them provided only limited privacy benefits (see Figure 2.3). To this day, strong privacy techniques remain available only in alternative, privacy-focused cryptocurrency. Monero is one such example.



**Figure 2.4.:** Inputs in Monero reference not a single output but a set of outputs. One of the outputs is the true output being spent, the others provide plausible deniability.

## 2.3. Monero

Monero is a privacy-focused cryptocurrency based on the CryptoNote protocol [157]. Its design is conceptually similar to that of Bitcoin, but makes use of three cryptographic techniques to increase users' privacy: stealth addresses, ring signatures, and confidential transactions.

### 2.3.1. Stealth Addresses

Stealth addresses [43] prevent an observer from recognizing that multiple payments on the blockchain were made to the same address. Based on the Diffie-Hellman key exchange over elliptic curves, the general idea behind stealth addresses is that every time a payment is made to a stealth address the sender derives a unique public key from the stealth address. They include their own (temporary) public key in the output. Because the recipient doesn't know the unique address the sender derived, they need to scan all outputs on the blockchain and try to access the funds in them by recomputing the shared secret.

### 2.3.2. Ring signatures

Monero uses ring signatures to hide the true output being spent by an input within a set of candidate outputs [122]. A ring signature is a digital signature computed non-interactively over a set of public keys with only one of the private keys used to

create the signature [136]. As it is not feasible to determine to which of the public keys the private key used corresponds to, they provide the signer with a degree of anonymity.

In Monero, an input can refer not only to the single output it redeems, but instead to a set of outputs that includes the actual output being spent (cf. Figure 2.4). The input contains a ring signature that proves the transaction is authorized to spend one of the outputs, but it does not reveal the exact output being spent.

With ring signatures it generally becomes impossible to tell whether an output has actually been spent (our deducibility attack in Chapter 3 highlights an important exception). To prevent double-spending of outputs, every input also contains a key image that is deterministically derived from the corresponding private key. Nodes need to maintain a database of revealed key images and check the key image in every input against it, rejecting inputs with key images that have been seen before. This also highlights the necessity of using the aforementioned one-time address scheme. If address reuse was possible as in Bitcoin, coins would become unspendable as only one output sent to an address can ever be spent due to the same key image.

### **2.3.3. Confidential transactions**

Starting with version 0.10.0, in September 2016, Monero added support for confidential transactions (RingCT) [122]. RingCT introduced a new transaction format that hides the denomination of the outputs. Before, the Monero client split output values into specific denominations in order to create sufficiently large anonymity sets. With RingCT, output values are no longer visible and thus all outputs of RingCT transactions constitute an anonymity set. This furthermore prevents correlation attacks (e.g., [71]), where an adversary would be able to identify a transaction made by a user by knowing that they transferred a specific amount of Monero.

## 2.4. BlockSci

A significant challenge in analysing the privacy of cryptocurrencies like Bitcoin comes from the large amount of data on the blockchain. As of June 2021, the Bitcoin blockchain contains a total of 653 million transactions, or 350 GB of on-disk data. Normal Bitcoin wallets are designed for efficient verification of transactions, but they are not particularly useful to analyze the entire blockchain.

Throughout Chapters 4 to 5 we use the open-source blockchain analysis tool BlockSci [87] that allows for fast analyses of the Bitcoin blockchain. While working on this thesis, we extensively contributed to the development of BlockSci, maintained the code base and built additional tools for testing and quality assurance of its parser and analysis library. For example, we built a testing framework that allows to create small, synthetic blockchains with customized transaction graphs that can be used to test the functionality of any general-purpose blockchain analysis tools [108]. To facilitate the analyses in this thesis, we use BlockSci version 0.7 with some additional enhancements that are not found in the public version.

### 2.4.1. Data format and parser

BlockSci supports blockchains with a transaction graph layout and protocol features similar to Bitcoin (i.e. UTXO-based cryptocurrencies where each input references one output). It stores transaction data in a concise, hybrid format optimized for sequential iterations of the transaction graph:

- The core transaction graph data is stored in a row-based format: inputs and outputs are stored inline with the transaction data, making sequential iterations over the graph extremely fast. This core transaction graph is always loaded into memory.

- Other data, such as additional protocol features or scripts, are stored in a hybrid format and loaded into memory on-demand.
- Indexes, for identifiers such as transaction hashes or addresses, are stored in a RocksDB database.

The BlockSci importer reads block files created by the Bitcoin reference implementation (Bitcoin Core), which stores the blockchain data in raw network format on disk. The parser transforms this data into the BlockSci format and serializes it to disk, enabling memory-mapping for fast analyses and parallelizable access. For our analyses, we modified the parser to record the SegWit serialization format of a transaction in a separate data file that is loaded on-demand.

#### **2.4.2. Address deduplication**

BlockSci deduplicates address data across similar high-level address types. For example, a public key could be used in a raw “pay to pubkey” output, in a P2PKH output, in the SegWit variant P2WPKH, and in a list of keys of a multisignature output script. Since the underlying public key is always the same, it only needs to be stored once. Similar deduplication takes place for P2SH outputs (for the traditional format and the SegWit format). When a P2SH output wraps a normal public key, the P2SH address in BlockSci contains a pointer to the underlying public key.

This degree of deduplication is particularly useful in the context of address clustering: a company might switch to using a new address format (such as from a normal P2PKH to the SegWit variant P2WPKH) reusing an existing key pair. In BlockSci, these two addresses are revealed as referring to the same underlying key and will thus automatically appear in the same cluster.

The duplication creates a small inconvenience when counting addresses, such as when we refer to the size of the cluster. A low-level address may correspond

to multiple, different address types on the blockchain. For easier comparability, whenever we report cluster sizes we count the number of addresses that were used on-chain with their high-level address type on the blockchain.

### **2.4.3. Analysis interface**

BlockSci provides two interfaces: a C++ interface that provides low-level access to the underlying BlockSci data, and a fluent Python interface that allows to write fast queries in a Jupyter notebook environment. For our analyses, we implemented performance-critical operations (e.g., tracing the flow of coins through the transaction graph or detecting change outputs) in the C++ interface and then extend the Python interface to provide convenience methods to access this functionality.

### **2.4.4. Clustering**

BlockSci has a built-in clustering module that performs multi-input clustering, and optionally change address clustering, on all addresses on the blockchain in the order that they appear in transactions. It also, in a first step, clusters the wrapping address and the wrapped address of all scripthash addresses. The clusterer relies on a union-find implementation that operates on consecutive indexes. As BlockSci addresses are stored as a combination of an index and an address type (such as pubkey or scripthash), a pre- and post-processing step is necessary to convert from and to the native format.

To enable the analyses in Chapter 4, we make several modifications to the clusterer:

- To be able to enhance an existing clustering at a later point, we store the intermediate output of the union-find algorithm before it is converted back into BlockSci's native address format.



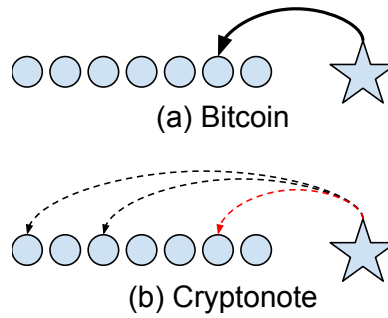
- To iterate over transactions in a cluster, BlockSci would first need to look up individual addresses and then retrieve their transactions. To speed up our analyses of the resulting clusterings, we extract every cluster's transactions after creating a clustering and store this mapping alongside the cluster data, speeding up subsequent analyses.
- We implement a new function to enhance an existing clustering by clustering change outputs. In contrast to the existing functionality to cluster change outputs, we do not provide an individual heuristic to apply but instead an externally computed set of change outputs.

# **An Empirical Analysis of Traceability in the Monero Blockchain**

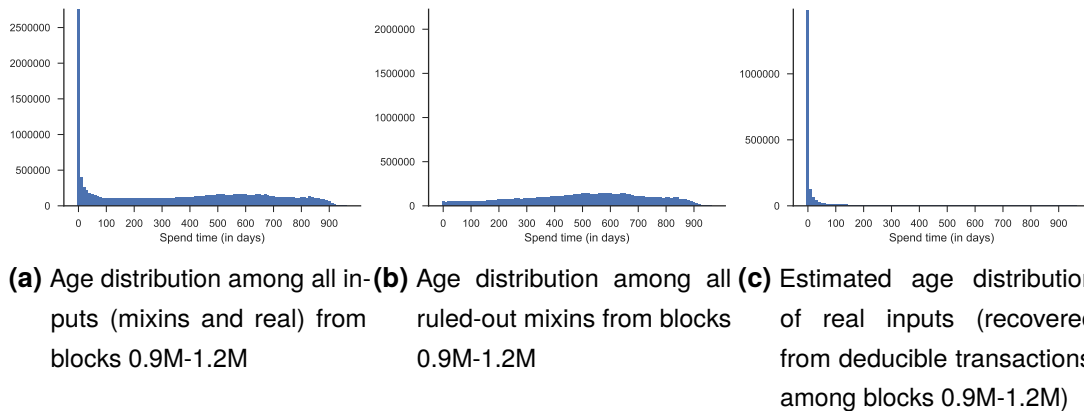
## **3.1. Introduction**

Monero is a leading privacy-centric cryptocurrency based on the Cryptonote protocol. As of November 2017 it was one of the most popular cryptocurrencies at a market capitalization of USD 1.5B. While Bitcoin, the first and currently largest cryptocurrency, explicitly identifies which coin in the transaction graph is being spent, Cryptonote allows users to obscure the transaction graph by including chaff transaction inputs called “mixins” (this is visualized in Figure 3.1).

As a result, Monero has attracted the attention of users requiring privacy guarantees superior to those Bitcoin provides. Some of the most publicized uses are illicit (for instance, the former online anonymous marketplace AlphaBay accepted Monero as a payment instrument), but we estimate that illicit use accounts for at most 25 % of all transactions. For all uses it can be imperative that the advertised privacy guarantees are maintained. Consider an attacker whose goal is to determine whether Alice made a sensitive payment to a merchant. The attacker might have access to Alice’s records at a cryptocurrency exchange (e.g., through hacks or collusion), and/or to the records of the merchant. In this scenario, the payment



**Figure 3.1.:** Transactions and tracing in Bitcoin and Cryptonote. Consider a new transaction (the star) which spends an available coin (the second circle from the right). In Bitcoin (a), each transaction input explicitly identifies the coin being spent, thus forming a linkage graph. In the Cryptonote protocol (b), each transaction input identifies a set of coins, including the real coin along with several chaff coins called “mixins.” However, many mixins can be ruled out by deduction (Section 3.3); furthermore, the real input is usually the “newest” one (Section 3.4).



**Figure 3.2.:** Age distributions of Monero mixins. In each graph, the Y-axis is the number of TXOs, and the X-axis represents the time difference between input and referenced output in days. The left graph (a) shows the distribution of all transaction inputs from blocks 0.9M to 1.2M where at least 1000 possible TXOs are available to choose from. Graph (b) shows the age distribution among mixins that can be ruled out. Graph (c) shows that real TXOs (based on deducible transactions, see Section 3.3) typically conform to a highly-skewed distribution. The disparity between these distributions makes it possible to guess the real input with an estimated 80 % accuracy.

**Table 3.1.:** Traceability of Monero transaction inputs with 1+ mixins (up to block 1288774, excluding RingCT inputs). Deducible inputs can be traced with complete certainty to the transaction output they spend (see Section 3.3). Among deducible transaction inputs, the real input is usually the “newest” one (see Section 3.4). Entries marked [Est.] are estimated by extrapolating from deducible transaction inputs, under the assumption that the spend-time distribution of deducible transactions is representative of the distribution overall.

	Not deducible	Deducible	Total
Real input is not newest	14.82 % [Est.]	302 078 (4.83 %)	19.65 % [Est.]
Real input is newest	22.24 % [Est.]	3 635 253 (58.11 %)	80.35 % [Est.]
Total	2 318 273 (37.06 %)	3 937 331 (62.94 %)	6 255 604 (100 %)

system must prevent the attacker from confirming that money flowed from Alice to the merchant.

In this chapter we conduct an empirical analysis of the Monero blockchain dataset. In particular, we evaluate the impact of two weaknesses in Monero’s mixin sampling strategy, which substantially undermine its privacy guarantees. Even though neither of these weaknesses is entirely new, having been discussed by developers since as early as 2015, we find that users who made privacy-sensitive transactions prior to February 2017 are at significant risk of post hoc deanonymization. We estimate that more than 200 000 transactions from July 2016 to February 2017 may be affected.

**Weakness 1. Most Monero transaction inputs prior to February 2017 contain deducible mixins, and can be traced to prior transactions via analysis.** The Monero software allows users to configure the default number of mixins to include in each transaction. Most Monero transaction inputs (64.04 % of all transaction inputs) do not contain any mixins at all (“0-mixin transactions”), but instead explicitly identify the prior transaction output (TXO) they spend, much like ordinary Bitcoin transactions.

0-mixin transactions not only provide no privacy to the users that created them, but also present a privacy hazard if other users include the provably-spent outputs as mixins in other transactions. When the Monero client chooses mixins, it does not take into account whether the potential mixins have already been spent. We find that among Monero transaction inputs with one or more mixins, 63 % of these are deducible, i.e. we can irrefutably identify the prior TXO that they spend.

**Weakness 2. Mixins are sampled from a distribution that does not resemble real spending behavior, and thus the real inputs can usually be identified.** When the Monero client spends a coin, it samples mixins to include by choosing randomly from a triangular distribution over the ordered set of available TXOs with the same denomination as the coin being spent. However, when users spend coins, the coins they spend are not chosen randomly from the blockchain, but instead appear (based on our empirical observations) as though drawn from a highly skewed distribution.

In Figure 3.2 we show data from the Monero blockchain that confirms these assumptions. Figure 3.2(c) shows the real age of inputs in a representative subset of Monero transactions for which the real input is known (i.e., among deducible transaction inputs as described above). Figure 3.2(b) shows the age distribution of mixins for which we know that they are not real inputs. When looking at the overall distribution of all inputs, shown in Figure 3.2(a), the overall distribution can clearly be seen as a mixture of these two distributions. Among transactions for which we have ground truth (i.e., the deducible transaction shown in (c)), we find that *the real input is usually the “newest” input, 92.33 % of the time*; based on simulation (Section 3.4), we estimate this holds for 80% of all transactions. Our results are summarized in Table 3.1.

**Monero usage.** We turn to the ecosystem and analyze the behavior of intermediaries to better understand the transactions affected by these weaknesses. While the early days of Monero were dominated by mining activity, starting in 2016 we see substantial growth in transaction volume. After accounting for the estimated impact of mining pools, which opt-out of privacy by publishing their transactions on webpages, there remain a substantial number of potentially privacy-sensitive transactions, more than a thousand per day. We estimate that many of these transactions (about 25 % of daily transactions at its peak) relate to the former underground marketplace AlphaBay, which positioned Monero as a more secure alternative to Bitcoin. The seizure of AlphaBay in July 2017 serves as a reminder of the fragility of these marketplaces, whether due to lawful actions, hacks, or exit scams [146], leaving users remain at risk of deanonymization. In Section 3.7 we discuss our results in the context of three high-profile criminal uses of Monero. Of course, Monero and other privacy-preserving technologies have legitimate uses as well, such as providing privacy for activists and supporting free speech within oppressive regimes.

In all of these, we consider an analyst model who has access to not just public blockchain data, but also records from exchanges and merchants, obtained through seizure or subpoena.

**Threat model.** Alice purchases a quantity of Monero currency (XMR) on a popular cryptocurrency exchange, such as Kraken or Poloniex, and withdraws it to a Monero wallet on her home computer, as recommended by Monero best practices guides [4]. Over time, she uses it for a number of innocuous activities such as online shopping and sending money to friends. In other words, she transacts with several parties using Monero under her real identity (the exchange because it is mandatory, the shopping site so as to have goods shipped to her, and so on). Later, Alice uses her

wallet pseudonymously for a sensitive payment, one where she expects privacy. The attacker’s goal is to link the pseudonymous account to a real name.

We consider a powerful attacker who is able to obtain two sets of logs (whether through hacks, seizures, or collusion). The first set of logs is of withdrawal transactions from the exchange (or another transaction where Alice used her real identity). The second set of logs is of deposits at the merchant where Alice made her sensitive payment. Due to the use of one-time addresses in Monero, the withdrawal transaction is the only piece of information on address ownership the attacker possesses. They cannot further infer common ownership of addresses (a deanonymization attack possible in many other cryptocurrencies [103, 135]). Furthermore, the attacker does not possess Alice’s private keys and cannot break any cryptographic primitives.

The success of the attack depends on the attacker’s ability to link the withdrawal transaction to one of the deposits at the merchant by deducing the real spend among Monero’s chaff inputs, thereby confirming Alice as the originator of the payment. We note that in addition to our attack, an attacker who is able to use side channels such as additional timing information (e.g., a time zone) can further reduce the effective anonymity set provided by the mixins.

**Proposed countermeasures.** We propose an improved mixin sampling strategy that can mitigate these weaknesses for future transactions. Our solution is based on sampling mixins according to a model derived from the blockchain data. We provide evidence that the “spend-time” distribution of real transaction inputs is robust (i.e., changes little over time and across different software versions), and can be well approximated with a simple two-parameter model. We extend the improved sampling with a sampling procedure which samples mixins in “bins.” We show

this binned sampling ensures privacy even in spite of a compromised sampling distribution.

**Lessons.** Our work highlights the special difficulties of designing privacy-preserving blockchain systems. As noted by Goldfeder et al. [71], cryptocurrency privacy combines the challenges of both anonymous communication and data anonymization. The data are necessarily public forever and vulnerabilities discovered at any point can be exploited to compromise privacy retroactively, as exemplified by the deducible Monero transactions from mid 2016. Privacy is also weakened by the fact that choices made by some users may affect other users detrimentally, such as the mining pools’ practice of publishing payout transactions.

**Concurrent and related work on Monero traceability.** Two reports from Monero Research Labs (MRL-0001 [123] and MRL-0004 [96]) have previously discussed concerns about Weakness 1, known as the “chain-reaction” attack. However, MRL-0001 considered only an active attacker that must own coins used in previous transactions. Our results show this vulnerability is not hypothetical and does not require an active attack, but in fact leads to the traceability of most existing transactions to prior to February 2017. MRL-0004 [96] discussed a passive attack scenario and provided a simulation analysis predicting that the mandatory 2-mixin minimum (implemented in version 0.9) would “allow the system to recover from a passive attack quite quickly.” Our results (Figure 3.5) show that indeed the fraction of deducible inputs indeed drops steadily after instituting the 2-mixin minimum (from 95 % in March 2016 down to 20 % in January 2017), though a significant fraction remain vulnerable.

Concurrently and independently of our work, Kumar et al. [92] also evaluated the deducibility attack in Monero. Our work differs from (and goes beyond)



their analysis in four main ways. First, we account for correlation between the deducibility attack and temporal analysis in our simulation results (Section 3.4.3). Though deducibility has been addressed in current versions, temporal analysis remains independently effective. Second, our analysis of the Monero ecosystem (Section 3.5) refutes a possible objection that our analysis only applies to irrelevant transactions that do not need privacy (such as mining pool payouts). Third, while we propose a similar countermeasure to temporal analysis, we evaluate ours through simulation. Finally, our binned mixins countermeasure is novel and defends against a strong adversary with prior information.

## 3.2. Background

**Cryptonote: Non-interactive mixing with ring signatures.** The Cryptonote protocol [157] introduces a technique for users to obscure their transaction graph, in principle preventing transaction traceability. Instead of explicitly identifying the TXO being spent, a Cryptonote transaction input identifies a set of possible TXOs, including both the real TXO along with several chaff TXOs, called mixins (as illustrated in Figure 3.1). Instead of an ordinary digital signature, each Cryptonote transaction comes with a ring signature (a form of zero-knowledge proof) that is valid for one of the indicated TXOs, but that does not reveal any information about which one is real. To prevent double-spending, every input must provide a key image that is unique to the output being spent, and the network must check whether this key image has ever been revealed before.

Several cryptocurrencies are based on the Cryptonote protocol, including Monero, Boolberry, Dashcoin, Bytecoin, and more. We focus our empirical analysis on Monero, since it is the largest and most popular as of mid 2017, e.g. it has the 12th largest market cap of all cryptocurrencies, over \$750M. However, our results are

also applicable to other Cryptonote-based protocols (as we show for Bytecoin in Appendix A.2).

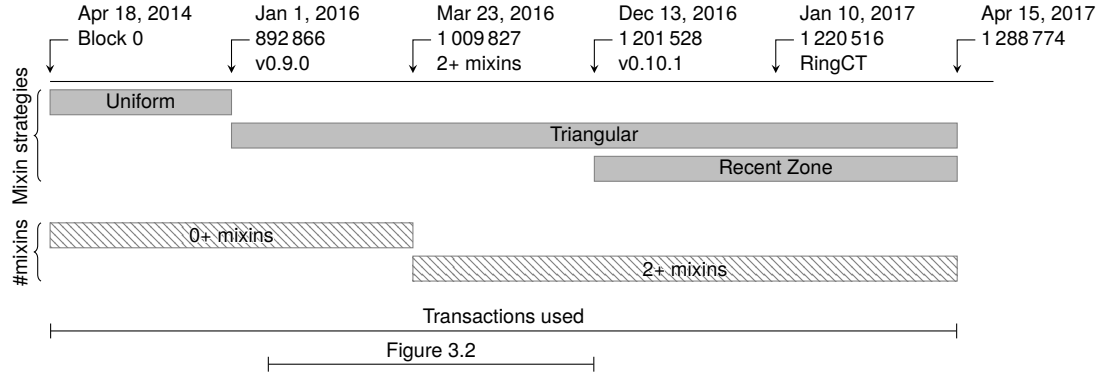
**Choosing mixin values in Cryptonote.** The Cryptonote protocol does not provide an explicit recommendation on how the “mixins” should be chosen. However, the original Cryptonote reference implementation included a “uniform” selection policy, which has been adopted (at least initially) by most implementations, including Monero. Since all the TXOs referenced in a transaction input must have the same denomination (i.e., a 0.01 XMR input can only refer to an 0.01 XMR output), the client software maintains a database of available TXOs, indexed by denomination. Mixins are sampled from this ordered list of available TXOs, disregarding any temporal information except for their relative order in the blockchain.

In principle, it is up to an individual user to decide on a policy for how to choose the mixins that are included in a transaction. Since it is not a “consensus rule,” meaning that miners do not validate that any particular distribution is used, clients can individually tune their policies while using the same blockchain. The Monero command-line interface allows users to specify the number of mixins, with a default of 4 as of September 2017.

Over the past several years, Monero’s mixin selection policy has undergone several changes; we describe the important ones below. A summary of the timeline relevant to our data analysis is shown in Figure 3.3.

**Prior to version 0.9.0 (January 1, 2016):** In the initial Monero implementation, mixins were selected uniformly from the set of all prior TXOs having the same denomination as the coin being spent. As a consequence, earlier outputs were chosen more often than newer ones.

**After version 0.9.0 (January 1, 2016):** Version 0.9.0 introduced a new policy for selecting mixins based on a triangular distribution, favoring newer coins as mixins



**Figure 3.3.:** Data considered in our experiment.

over older ones. This change was motivated by the belief that newer inputs are more likely to be the real input [54]. This version also introduced a mandatory minimum number of 2 mixins per transaction input, as recommended by MRL-0001 [123]. This mandatory minimum was enforced after a “hard fork” flag day, which occurred on March 23, 2016.

**After version 0.10.0 (September 19, 2016):** Version 0.10.0 introduced a new RingCT feature [122], which allows users to conceal the denomination of their coins, avoiding the need to partition the available coins into different denominations and preventing value-based inference attacks. RingCT transactions were not considered valid until after a hard fork on January 10, 2017.

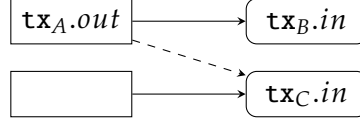
The RingCT feature does not directly address the traceability concern. But as RingCT transactions can only include other RingCT transaction outputs as mixins, and since it was deployed after the 2-mixin minimum took effect (in version 0.9.0), there are no 0-mixin RingCT inputs to cause a hazard.

**After version 0.10.1 (December 13, 2016):** Version 0.10.1 included a change to the mixin selection policy: now, some mixins are chosen from among the “recent” TXOs (i.e., those created within the last 5 days, called the “recent zone”). Roughly, the policy is to ensure 25 % of the inputs in a transaction are sampled from the recent zone.

**After version 0.11.0 (September 07, 2017):** Version 0.11.0 increased the minimum number of mixins per transaction input to 4, which was enforced after a hard fork on September 15, 2017. This version also incorporates temporal analysis countermeasures (increasing the number of mixins chosen from the recent zone, and narrowing the recent zone from 5 days to 3 days) based on recommendations from an early draft of our paper.

**Transaction notation.** We briefly introduce some notation for describing transaction graphs. For a transaction  $\text{tx}$ ,  $\text{tx.in}$  denotes the vector of  $\text{tx}$ 's transaction inputs, and  $\text{tx.out}$  denotes the vector of  $\text{tx}$ 's transaction outputs. We use subscripts to indicate the elements of input/output vectors, e.g.  $\text{tx.in}_1$  denotes the first input of  $\text{tx}$ . Each Cryptonote transaction input contains a reference to one or more prior transaction outputs. We use array notation to denote the individual references of an input. We use a dashed arrow,  $\dashrightarrow$ , to denote this relationship, e.g.  $\text{tx}_A.\text{out}_i \dashrightarrow \text{tx}_B.\text{in}_j[m]$  means that the  $m$ 'th reference of the  $j$ 'th input of transaction  $\text{tx}_B$  is a reference to the  $i$ 'th output of  $\text{tx}_A$ . Although a Cryptonote transaction input may contain more than one reference, only one input is the *real* reference (known only to the sender), indicated by a solid arrow. Thus  $\text{tx}_A \rightarrow \text{tx}_B$  indicates that  $\text{tx}_A$  contains an output that is spent by one of the inputs in  $\text{tx}_B$ .

Transactions included in the blockchain are processed in sequential order; we use  $\text{tx}_A < \text{tx}_B$  to indicate that  $\text{tx}_A$  occurs before  $\text{tx}_B$ . Other properties of a transaction are defined as functions, and introduced as needed. For example,  $\text{time}(\text{tx})$  refers to the timestamp of the block in which  $\text{tx}$  is committed.

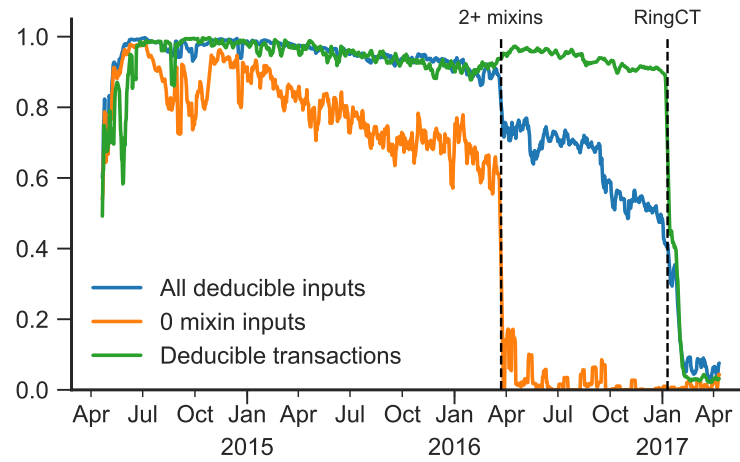


**Figure 3.4.:** 0-mixins effectively reduce the untraceability of other transactions: the dashed reference can be ruled out since  $\text{tx}_A.out$  must have been spent in  $\text{tx}_B.in$ .

### 3.3. Deducible Monero Transactions

A significant number of Monero transactions do not contain any mixins at all, but instead explicitly identify the real TXO being spent. Critically, at the beginning of Monero’s history, users were allowed to create zero-mixin transactions that do not contain any mixins at all. Figure 3.5 shows the fraction of transactions containing zero-mixin inputs over time. As of April 15, 2017 (at block height 1 288 774), a total of 12 158 814 transaction inputs do not contain any mixins, accounting for 64.04 % of all inputs overall.

One might think at first that 0-mixin transactions are benign. Transactions with fewer mixins are smaller, and hence cost less in fees; they thus represent an economical choice for an individual who does not explicitly desire privacy for a particular transaction. However, it turns out that the presence of 0-mixin transactions is a hazard that reduces the untraceability of other transactions, even those that include one or more mixins. For example, as shown in Figure 3.4, suppose a transaction output  $\text{tx}_A.out_i$  is spent by a 0-mixin transaction input  $\text{tx}_B.in_i$  (i.e.  $\text{tx}_A.out_i \dashrightarrow \text{tx}_B.in_j$  where  $|\text{tx}_B.in_j| = 1$ , from which we can conclude  $\text{tx}_A.out_i \rightarrow \text{tx}_B.in_j$ ). Now, suppose  $\text{tx}_A.out_i$  is also included as a mixin in a second transaction with one mixin,  $\text{tx}_A.out_i \dashrightarrow \text{tx}_C.in_k[m]$ , where  $|\text{tx}_C.in_k| = 2$ . Since we know that the given output was actually spent in  $\text{tx}_B$ , we can deduce it is not spent by  $\text{tx}_C$  (i.e.,  $\text{tx}_A.out_i \nrightarrow \text{tx}_C$ ), and hence the remaining input of  $\text{tx}_C.in_j$  is the real one.



**Figure 3.5.:** Fraction of transaction inputs that can be deduced and transactions including at least one deducible input (averaged over intervals of 7 days).

**Table 3.2.:** Monero transaction inputs where the real input can be deduced (1+ mixins,  $\geq 1000$  TXOs available, excluding RingCT).

	Before 2-mixin hardfork			After 2-mixin hardfork			After 0.10.1, prior to Apr 15, 2017		
	Total	Deducible	(%)	Total	Deducible	(%)	Total	Deducible	(%)
1 mixins	683 458	608 087	(88.97)	0	—	—	0	—	—
2 mixins	250 520	206 276	(82.34)	1 882 681	1 209 259	(64.23)	732 251	308 926	(42.19)
3 mixins	634 520	480 500	(75.73)	564 525	376 920	(66.77)	126 795	65 738	(51.85)
4 mixins	217 493	156 767	(72.08)	376 432	192 348	(51.10)	145 687	33 022	(22.67)
5 mixins	87 077	43 214	(49.63)	48 806	26 599	(54.50)	3900	950	(24.36)
6 mixins	115 199	65 546	(56.90)	224 202	119 716	(53.40)	24 817	7890	(31.79)
7 mixins	3671	1680	(45.76)	4499	1770	(39.34)	1711	235	(13.73)
8 mixins	2216	1067	(48.15)	5048	1968	(38.99)	1458	249	(17.08)
9 mixins	1811	838	(46.27)	3264	1069	(32.75)	264	48	(18.18)
10+ mixins	57 363	11 997	(20.91)	46 791	12 970	(27.72)	9145	1682	(18.39)
Total	2 053 328	1 575 972	(76.75)	3 156 248	1 942 619	(61.55)	1 046 028	418 740	(40.03)
Overall					(62.94)				

Notice that in this example, it does not matter if the real spend  $\text{tx}_B$  that renders it deducible occurs after the 1-mixin transaction  $\text{tx}_C$ , i.e., if  $\text{tx}_C < \text{tx}_B$ . Thus at the time  $\text{tx}_C$  is created, it is impossible to know whether a future transaction will render that mixin useless. However, the problem has been exacerbated by the behavior of the Monero client software, which does not keep track of whether a potential mixin had already been clearly spent, and naïvely includes degenerate mixins anyway.

### 3.3.1. Implementation

We extracted relevant information from the Monero blockchain, up to block 1288774 (April 15, 2017) and stored it in a Neo4j graph database (11.5 GB of data in total). We then apply the insight above to build an iterative algorithm, where in each iteration we mark all of the mixin references that cannot be the real spend since we have already deduced that the corresponding output has been spent in a different transaction. With each iteration, we further deduce the real inputs among additional transaction input sets. (An alternative formulation – as a SAT problem – is presented in Appendix A.1.)

### 3.3.2. Results on deducible transactions

Table 3.2 shows the results from applying the approach described above to Monero blockchain data. As it turns out, approximately 63 % of Monero transaction inputs (with 1+ mixins) so far can be traced in this way.

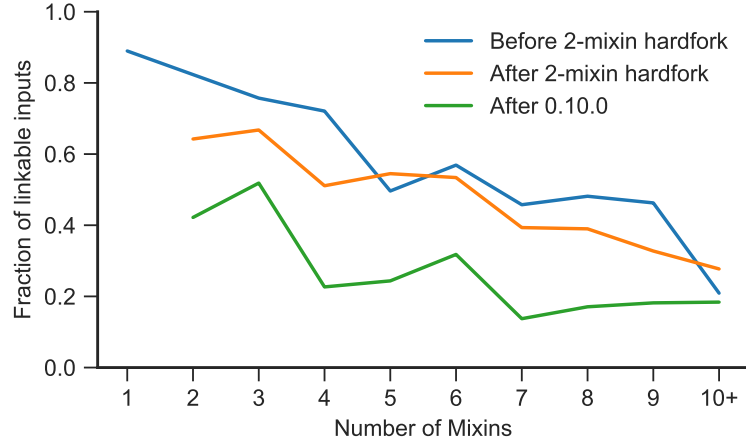
In Figure 3.6, we show how the vulnerability of Monero transactions to deduction analysis varies with the number of mixins chosen, and in Figure 3.5 we show how this has evolved over time. We see that transactions with more mixins are significantly less likely to be deducible, as one would hope. Even among transactions with the same number of mixins, transactions made with later versions of the software are

less vulnerable. This is because at later dates, especially after the network started to enforce a minimum of 2 mixins, the 0-mixin transaction outputs accounted for a smaller number of the available mixins to choose from. However, the share of transactions with at least one deducible input (enough for an attacker to link an account to a user) stays above 80 % even after the 2-mixins minimum. Surprisingly, we found over 100 000 transaction inputs with 10 or more mixins, presumably indicating a high level of desired privacy, that are vulnerable under this analysis.

**Applicability to current and future transactions using RingCT.** The weakness studied in this section is primarily a concern for transactions made in the past, as transactions using the new RingCT transaction option are generally immune. RingCT has been available to users since January 2017, and at the time of writing has already been widely deployed. RingCT transactions are now used by default for the vast majority of new transactions. The reason why these new transactions are immune is not because of the RingCT mechanism itself, but rather because RingCT was only deployed after the mandatory 2-mixin minimum was enforced. Therefore, RingCT transaction outputs cannot be spent by 0-mixin inputs.

**Applicability to other Cryptonote cryptocurrencies.** As the deducibility attack originates from the mixing sampling procedure inherent to the Cryptonote protocol, other cryptocurrencies based on Cryptonote share the same weaknesses. Appendix A.2 contains results of an analysis of Bytecoin, an early implementation of the Cryptonote protocol. We deduce the real input for 29 % of transaction inputs with 1 or more mixins. While this rate is smaller than in Monero, we conjecture it to be a result of lower usage of Bytecoin.





**Figure 3.6.:** Transaction inputs are less likely to be deducible if they have more mixins and if they are found among later blocks in the Monero blockchain.

## 3.4. Tracing With Temporal Analysis

In the previous section, we showed that a majority of Monero transactions inputs can be traced with certainty through logical deduction. In this section, we investigate an entirely unrelated complementary weakness which traces inputs probabilistically.

### 3.4.1. Effective-Untraceability

To quantify the untraceability of a transaction input when some referenced outputs are more likely to be the real spend we first define the effective-untraceability set size or the “effective-untraceability.”

Anonymity set size is a standard privacy metric which typically assumes each element in the anonymity set is equiprobable. This is not the case in Monero as the temporal analysis attacks we will demonstrate show that some referenced outputs are more likely than others to be the real spend. Following the approach of [50, 143] we use entropy to account for these differing probabilities. We measure guessing entropy rather than Shannon entropy [144] because it is intuitive to

our setting, an attacker trying to guess the real spend, and is easily relatable to effective-untraceability.

First defined in [98], guessing entropy is commonly used as a measure of password strength [26]. In the context of untraceability guessing entropy is the expected number of guesses before guessing the spent output. A transaction input's guessing entropy is

$$\text{Ge} = \sum_{0 \leq i \leq M} i \cdot p_i$$

where  $p = p_0, p_1, \dots, p_M$  are probabilities, sorted highest to lowest, that a referenced output is the real spend of a transaction input.

We define the effective-untraceability, i.e. effective-anonymity set size, of a transaction input as  $(1 + 2 \cdot \text{Ge})$ . If all referenced outputs of a transaction input are equally likely to be the real spend, the effective-untraceability for that input is  $M + 1$ .

### 3.4.2. The Guess-Newest heuristic

Among all the prior outputs referenced by a Monero transaction input, the real one is usually the newest one.

Figure 3.2(c) shows the spend-time distribution for deducible transaction inputs (i.e., zero-mixin inputs and inputs for which the real TXO can be deduced using the technique from Section 3.3). As can be seen, this distribution is highly right-skewed; in general users spend coins soon after receiving them (e.g., a user might withdraw a coin from an exchange, and then spend it an hour later). In contrast, the distribution from which (most) mixins are sampled (either a uniform distribution or a triangular distribution, for the most part) includes much older coins with much greater probability.

**Table 3.3.:** Percentage of deducible transaction inputs where the real input is the “newest” input.

	Before 2-mixin hardfork			After 2-mixin hardfork			After 0.10.1, prior to Apr 14, 2017		
	Deducible	Newest	(%)	Deducible	Newest	(%)	Deducible	Newest	(%)
1 mixins	608 087	585 424	(96.27)	0	—	—	0	—	—
2 mixins	206 276	191 372	(92.77)	1 209 259	1 126 924	(93.19)	308 926	293 051	(94.86)
3 mixins	480 500	461 154	(95.97)	376 920	353 246	(93.72)	65 738	59 693	(90.80)
4 mixins	156 767	139 626	(89.07)	192 348	149 722	(77.84)	33 022	18 889	(57.20)
5 mixins	43 214	39 854	(92.22)	26 599	24 971	(93.88)	950	473	(49.79)
6 mixins	65 546	51 816	(79.05)	119 716	102 378	(85.52)	7890	6458	(81.85)
7 mixins	1680	1522	(90.60)	1770	989	(55.88)	235	115	(48.94)
8 mixins	1067	964	(90.35)	1968	1310	(66.57)	249	163	(65.46)
9 mixins	838	692	(82.58)	1069	355	(33.21)	48	40	(83.33)
10+ mixins	11 997	10 822	(90.21)	12 970	11 750	(90.59)	1682	1480	(87.99)
Total	1 575 972	1 483 246	(94.12)	1 942 619	1 771 645	(91.20)	418 740	380 362	(90.83)
Overall					(92.33)				

**Cross-validation with deducible transactions.** In order to quantify the effect of these mismatched distributions, we examine the rank order of the transactions with 1 or more mixins for which we have ground truth (i.e. the deducible transactions from Section 3.3). Table 3.3 shows the percentages of deducible transaction inputs for which the real (deduced) reference is also the “newest” reference. It turns out that overall, 92 % of the deducible inputs could also be guessed correctly this way. For  $1 \leq M \leq 10$  such transaction inputs have an effective-untraceability of no more than 1.16–1.80.

We note that transactions with more mixins are only slightly less vulnerable to this analysis: even among transaction inputs with 4 mixins (the required minimum since September 2017) the Guess-Newest heuristic still applies to 81 % of these transactions.

**Validation with ground truth.** We also verify the heuristic using our own ground truth, which we obtain by periodically creating transactions using the default wallet in Monero 0.10.3.1. We set up wallets for three scenarios and send transactions from one wallet to another. The time gap between two transactions follows an exponential

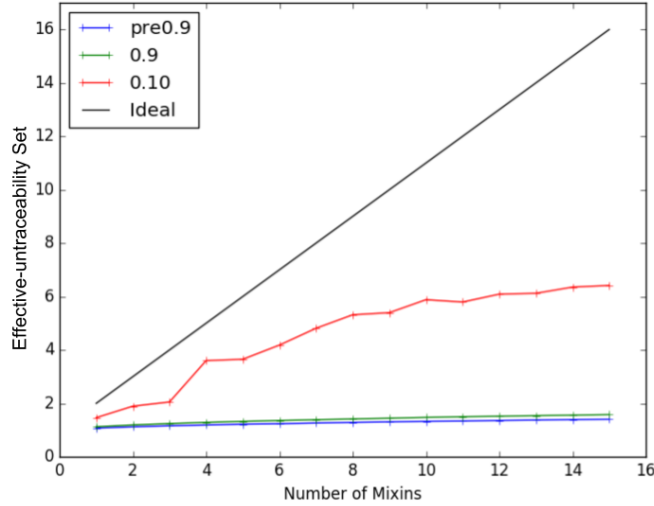
distribution, with the rate parameter set such that the means of the distributions correspond to 30 minutes, 4 hours, and 1 day.

We evaluate whether guessing the most-recently created input identifies the true input. At a 95 %-confidence level, we get success probabilities of  $0.95 \pm 0.02$  for the 30-minute interval ( $n = 120$ ),  $0.90 \pm 0.03$  for 4 hours ( $n = 84$ ), and  $0.42 \pm 0.14$  for 24 hours ( $n = 12$ ).

### 3.4.3. Monte Carlo simulation

The Guess-Newest heuristic and the deducibility attack (Section 3.3) are not entirely independent. The transactions that are deducible tend to be those that include old mixins, which thus also makes them more likely susceptible to Guess-Newest. As a consequence, cross-validation of the heuristic using only deducible transactions overstates the effectiveness of temporal analysis.

To control for this correlation, we also employ an alternative validation strategy based on a Monte Carlo simulation. In each trial, we simulate a transaction, where the real input is sampled from the dataset of deducible transactions, but the mixins are chosen using the sampling algorithms in the Monero software. Two factors determine which mixins are available to choose from: the denomination of the real input, and the blockchain data at the time the transaction is created. To simplify the simulation, we fix the block height at 1 236 196 (Jan 31, 2017). We consider the dataset of deducible and 0-mixin transactions as records of the form (spendtime, denomination), where the spend time is the block timestamp difference between when a transaction output is created and when it is spent, i.e.  $\text{spendtime}(\text{tx}_B.in_j) = \text{time}(\text{tx}_B) - \text{time}(\text{tx}_A)$ , where  $\text{tx}_A \rightarrow \text{tx}_B$ . We sample uniformly from these records to choose the real input. Next we apply the mixin sampling strategy using the available transaction outputs matching the chosen denomination. We note that our sampling procedures



**Figure 3.7.:** Estimated vulnerability to the Guess-Newest heuristic for varying sampling policies in Monero, based on Monte Carlo simulations (100k trials each).

are simplified models, and in particular elide the handling of edge cases such as avoiding “locked” coins that have been recently mined.

In Figure 3.7 we show the results of simulating transactions using the sampling rules from the three main Monero versions (pre 0.9, 0.9, and 0.10.1), as applied to the blockchain at height 1 236 196 (Jan 31, 2017). For versions 0.9.0 and prior, even up to 4 mixins, our simulation suggests that the newest input can be guessed correctly 75 % of the time. Across all versions, including more mixins does reduce the effectiveness of this heuristic, although the benefit of each additional mixin is significantly less than the ideal.

Each subsequent update to Monero’s mixin sampling procedure has improved the resistance of transactions to the Guess-Newest heuristic. We note that from developer discussions [55] it appears that this concern has indeed been the motivation for such changes. In particular, the triangular distribution was adopted in place of the uniform distribution in Monero version 0.9 specifically because it chooses new mixins with higher probability than “old” mixins, and version 0.10.1 explicitly introduces a policy that often includes additional “recent” (within 5 days) mixins.

However, we believe the magnitude of the problem has been underestimated. Under the current default behavior, i.e. 4 mixins, and using the 0.10.1 sampling procedure, we estimate that the correct input reference can be guessed with 45 % probability (rather than the 20 % ideal if all input references were equally likely).

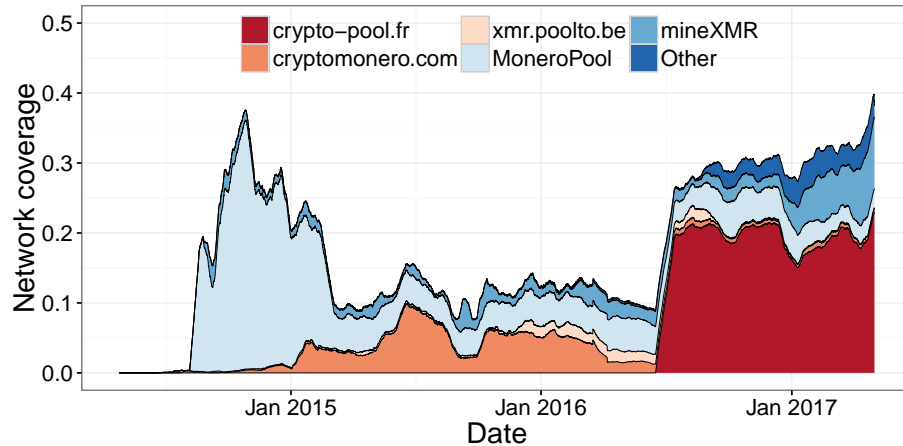
Excluding all of the deducible inputs, the average number of mixins among the remaining inputs is 3.53. We therefore estimate that as a lower bound, based on the “0.10.1” line in Figure 3.7, that at least 60 % of the remaining inputs are correctly traced using Guess-Newest. Extrapolating from this figure, we estimate that in total the Guess-Newest heuristic correctly identifies 80 % of all Monero inputs (Table 3.1).

## 3.5. Characterizing Monero Usage

In the previous section, we showed that a significant number of Monero transactions are vulnerable to tracing analysis. However, not all transactions are equally sensitive to privacy. We consider typical or *normal* transactions in Monero to be privacy-sensitive. On the other hand, *public* transactions, for which privacy is not a major concern to their participants, and where details of the transaction may even be publicly disclosed, form a special case. We next quantify these different usage types for Monero transactions, to assess potential impact of our attacks on privacy-conscious users.

### 3.5.1. Quantifying mining activity

An integral part of a cryptocurrency is mining, which refers to the process of bundling transactions in blocks and minting new currency. Whenever a block is created, the “miner” of that block receives a monetary reward, described by a special “coinbase” transaction. With an initial block interval of 1 minute until March 20,



**Figure 3.8.:** Fraction of blocks created by mining pools for which payment transactions are made public and have been collected.

2016, and a block interval of 2 minutes thereafter, it is conceivable that – at least in the early days – a significant share of daily transactions relate to mining.

Miners often combine their efforts by joining mining pools to reduce the variance of their payouts. Usually, the pool owner receives the full block reward and then distributes the reward according to each miner’s contribution. To provide transparency and accountability, many pools publicly announce the blocks they find as well as the payout transactions in which they distribute the rewards. This, however, reveals sensitive information about the relation between payout and coinbase transactions. If an input in a pool’s payout transaction spends from a coinbase transaction of a block known to belong to the same pool, it is likely the real spend. This deducibility may even weaken the privacy of other transactions, similar to a 0-mixin input. Transactions related to mining activity should thus be considered public.

To account for public transactions related to mining, we crawled the websites of 18 Monero mining pools and extracted information about 73 667 pool payouts. We also collected information about 210 800 non-orphaned blocks that had been won

by these pools. We analyzed the coverage of this mining activity by computing the fraction of all blocks in the network that had been produced by pools in our dataset over a moving window of 20 000 blocks. Figure 3.8 shows the results of this analysis.

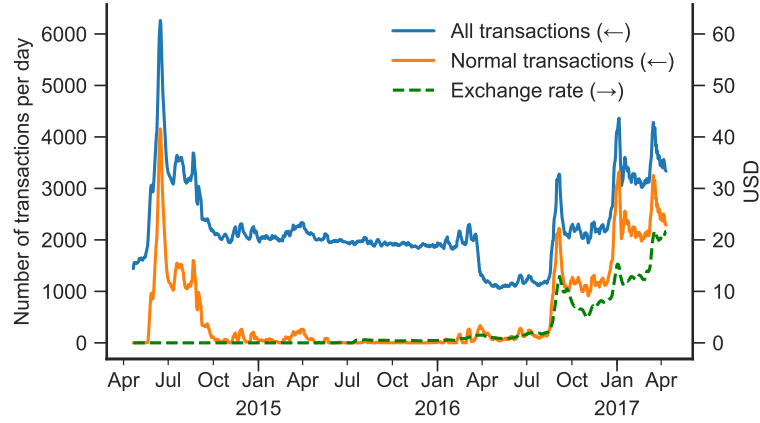
Our data accounts for roughly 30 % of the Monero mining power in late 2014. In early 2015, according to an archive of Monerohash.com [107] retrieved via The Wayback Machine [147], over 70 % of the mining power belonged to a combination of the Dwarf [52] and MinerGate [106] mining pools, neither of which reveals payment transactions, and of unknown mining sources. This centralization of hash power continued until April of 2016 at which point Dwarf and MinerGate were still significant (about 35 %), but there was no longer significant unaccounted for mining power.

Besides the 73 667 transactions for which we have ground truth, we can estimate the number of unlabeled transactions in the network that are used for mining payments. To minimize transaction fees, most pools will pay their miners only a few times per day, batching together payments to many miners into a few transactions with up to hundreds of outputs. Most pools do offer an option for immediate payout upon request, but charge a significant fee for doing so. Additionally, pools will allow users to be paid to an exchange service instead of to their own wallet. Thus, determining precisely the number of mining transactions is challenging.

Working under the assumption that the distribution of payments from the pools we observed tracks that of the pools that we were not able to observe, namely that the distribution of transactions is driven primarily by the needs and demands of the miners, we can estimate the volume of total mining-related transactions.

To do so, we compute the ratio between the total number of transactions observed and the total number of blocks mined by pools in our dataset. We estimate that in





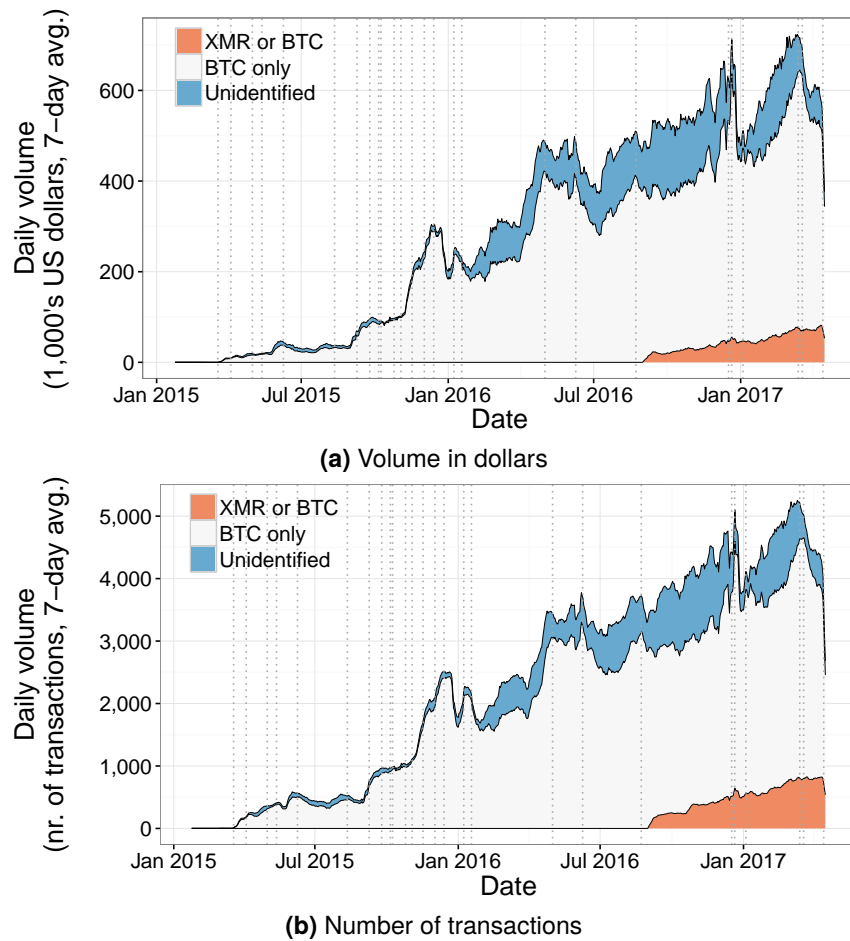
**Figure 3.9.:** Comparison of the volume of estimated non-mining transactions to the overall transaction volume.

addition to the coinbase transaction, approximately 0.438 transactions related to mining payments are created for each new block in Monero.

Figure 3.9 plots the number of overall transactions per day against the number of estimated normal transactions. Non-mining use was generally low during 2015 and until mid-2016, with at most a few hundred transactions per day. Transaction volume increases after mid-2016, rising over 2000 per day in 2017, which cannot be accounted for by mining-related activity. There is also strong correlation between the number of normal transactions and the Monero exchange rate. This suggests that changes in the ecosystem might be responsible for the additional transactions. We next explore a potential source for these unaccounted for normal transactions.

### 3.5.2. Usage on online anonymous marketplaces: the AlphaBay case

Online anonymous marketplaces have traditionally used Bitcoin for monetary transactions [37], which makes them potentially vulnerable to transaction graph analysis [103]. Because Monero advertises stronger anonymity properties than



**Figure 3.10.:** Estimated AlphaBay sales volume since inception

Bitcoin, a couple of online anonymous marketplaces (Oasis, AlphaBay) have started supporting it [74]. AlphaBay, in particular, started its operations in December 2014 and featured tens of thousands of pages until it was shut down by an international law enforcement operation in July 2017, making it one of the longest-lived (and possibly largest) marketplaces to that date.

On August 22, 2016, AlphaBay announced that it would support Monero, and allowed vendors to list items accepting Monero. On September 1, 2016, buyers were then able to use Monero to purchase items on AlphaBay. Inspecting the number of daily transactions in Figure 3.9 reveals a strong correlation between these events and the overall transaction volume. On August 22, the day of the announcement,

the number of transactions in the Monero blockchain increased by more than 80 % compared to the previous day, and it peaked at 4444 transactions on September 3, two days after Monero payments were made available on AlphaBay.

By default, an item listed on AlphaBay only accepts Bitcoin as a payment mechanism. Vendors have to explicitly allow Monero for it to be considered—and can also disable Bitcoin in the process. There are thus three types of items: items that only accept Monero, items that only accept Bitcoin, and items that accept both.

We obtained AlphaBay crawl data from Soska and Christin [146], and multiplied item feedback instances by item prices, to estimate sales volumes. We plot our results in Figure 3.10. Figure 3.10(a) shows the volume in US dollars, accounting for daily changes in Monero exchange rate. Each vertical dashed line corresponds to a scrape of the website. The top curve corresponds to the total daily volume of transactions, averaged over 7-day intervals. While sales volume remained fairly modest until mid-2015, it has steadily climbed to reach approximately USD 600 000 a day in 2017, which is more than the combined volume of the major online anonymous marketplaces in 2013–2015 [146].<sup>1</sup> Of those transactions, we are able to identify that a vast majority (in white) only accept Bitcoin. Starting in September 2016, a modest, but increasing number of items started accepting Monero along Bitcoin. The total amount of sales for these items, represented in orange, gives an upper bound for the dollar amount of Monero transactions on AlphaBay—as of early 2017, approximately USD 60 000/day (or 10 % of all AlphaBay transactions in volume). To get a lower bound on the amount of Monero transactions, we also look at items that *only* accept Monero, but these remain negligible (totaling between 0 and USD 100/day in transactions). In short, up to USD 60 000/day of transactions on the AlphaBay marketplace used Monero.

---

<sup>1</sup>Anecdotal evidence suggests that AlphaBay reached more than \$800 000/day shortly after the end of the measurement interval presented in this paper.

Figure 3.10(b) plots the number of transactions over time (averaged, again, over 7-day rolling windows). We see that, as of early 2017, up to 1000 transactions per day on AlphaBay might be relying on Monero. Comparing with Figure 3.9, we estimate that at most (approximately) a quarter of all Monero transactions could be accounted for by deposits at AlphaBay. Among the remaining 75 % of Monero transactions that we estimate are non-mining, we cannot conclude how many are privacy sensitive. Good candidates for the sources of these transactions include, gambling, exchanges, and traditional direct user to user payments.

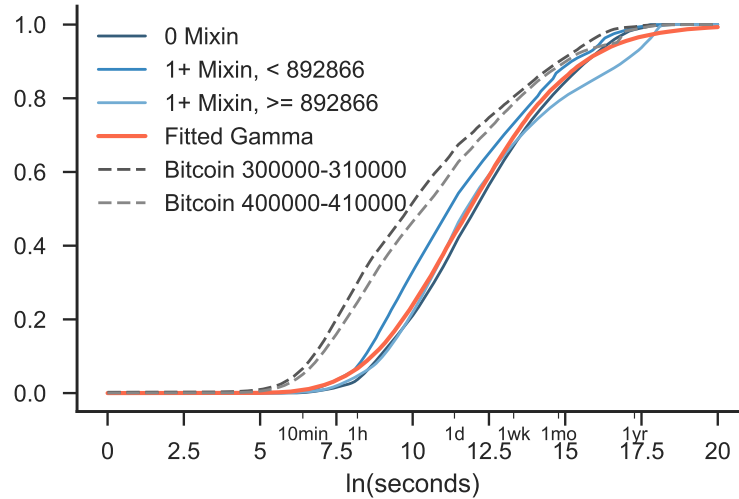
## 3.6. Countermeasures

We now propose two countermeasures to improve Monero’s resistance to temporal analysis. The first is to improve the mixin-sampling procedure to better match the real spend-time of Monero users. The second takes a more pessimist view and aims to preserve some untraceability even in the face of a highly compromised mixin sampling distribution.

### 3.6.1. Fitted mixin sampling distribution

In this section we discuss a way to improve the sampling procedure. At a high level, the idea is to estimate the actual spend-time distribution, and then sample mixins according to this distribution.

**Estimating the spend-time distribution** In Figure 3.11, we show the CDFs of the spend-time distributions from the Monero blockchain as well as the Bitcoin blockchain. For Monero, we split the data by 0-mixin transaction inputs as well as the 1+ mixin inputs for which we can deduce the real input. From this graph, we make the following qualitative observations:



**Figure 3.11.:** CDFs of spend-time distributions in Bitcoin and in Monero (deducible transaction inputs), over multiple time intervals. A gamma distribution (red line) is fitted to the combined Monero data (shape=19.28, rate=1.61).

- Observation 1: The spend-time distribution appears invariant with respect to time.
- Observation 2: The spend-time distribution appears the same for 0-mixin as well as 1+ mixin transactions.
- Observation 3: While the Bitcoin spend-time distribution appears to have a somewhat similar shape, the distributions are quite different (i.e., the Kolmogorov-Smirnov distance is approximately 0.3).

Based on these observations, we set about fitting a parametric model to the combined Monero data. We heuristically determined that the spend time distributions, plotted on a log scale, closely match a gamma distribution. We used R's `fitdistr` function to fit a gamma distribution to the combined Monero data from deducible transaction inputs (in log seconds). The resulting best-fit distribution has shape parameter 19.28, and rate parameter 1.61. By inspection (Figure 3.11), this appears to accurately fit the overall Monero spend-time distribution.

**Sampling mixins using the spend-time distribution.** To make use of the spend-time distribution described above, we need a way of sampling transaction output indices that matches the ideal spend-time distribution as closely as possible. Our proposed method is to first sample a target timestamp directly from the distribution, and then find the nearest block containing at least one RingCT output. From this block, we sample uniformly among the transaction outputs in that block. This procedure is defined Algorithm 1.

To quantify the effectiveness of our new sampling scheme, we turn again to the Monte Carlo simulation described in Section 3.4.3. Figure 3.12 shows the effective untraceability set under several regimes, including the RingCT protocol as of 0.10.1 (the version prior to the initial preprint release of our paper), our proposed mixing sampling routine method (Ppd), and version 0.11.0 (which incorporates a countermeasure based on our preprint). Our proposed countermeasure performs very close to the ideal. At the default setting of 4 mixins, our method nearly doubles the effective untraceability set versus 0.10.1 (approximately  $\approx 4$ , 80% of the ideal, instead of approximately  $\approx 2$ ); for large numbers of mixins, our improvement is nearly four times better. As the simulation is based on the same dataset to which we fit a parametric distribution, this is best understood as a goodness-of-fit test. Under this analysis, we also find that the countermeasure employed in 0.11.0 performs nearly as well as our proposed countermeasure at the default setting, though the gap increases with the ring size.

In the conference version of this paper, we evaluated our countermeasure using the results of several simulations based on an extrapolation that the transaction rate of Monero would remain constant. At the time of the experiment, March 2017, there were an average of 8.29 RingCT transaction outputs per block. We performed this extrapolation because unlike the deducibility weakness, which improves over time

---

**Algorithm 1.** Our proposed mixin sampling scheme.

```
SAMPLEMIXINS(RealOut, NumMixins)
  MixinVector := [];
  while |MixinVector| < BaseReqMixCount do
     $t \leftarrow \text{Exp}(\text{GammaSample}(\text{shape}=19.28, \text{rate}=1.61));$ 
    if  $t > \text{CurrentTime}$  then
      continue;
    Let  $B$  be the block containing at least one RingCT output, with timestamp closest to
      CurrentTime- $t$ ;
     $i \leftarrow$  uniformly sampled output among the RingCT outputs in  $B$  if  $i \notin \text{MixinVector}$  and
       $i \neq \text{RealOut.idx}$  then
      MixinVector.append( $i$ );
  return sorted(MixinVector + [RealOut.idx]);
```

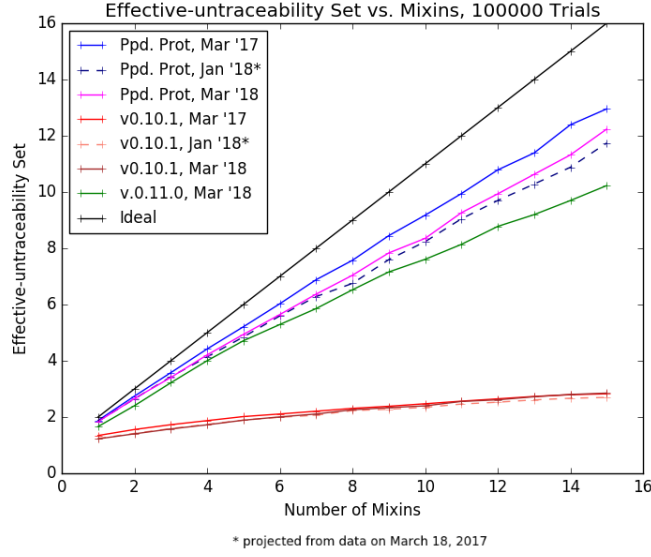
---

(see Figure 3.6), the problem of sampling from the wrong temporal distribution becomes *worse* over time, since the set of “old” mixins to choose from grows larger and larger. Our simulations with extrapolated data performed similarly to our experiments with data from March 2018, although the effective untraceability set at this point is slightly better than projected.

### 3.6.2. Binned mixin sampling

Here we introduce a countermeasure called binned mixin sampling which modifies the current mixin sampling procedure. It is designed to maintain a minimum level of untraceability even in face of compromised mixin sampling distributions or deduction attacks. Binned mixin sampling improves on the current single mixin sampling, which offers no guarantee that temporal analysis will not completely trace a transaction input by reducing effective-untraceability to  $\approx 1$ .

**Compromised mixin sampling distributions.** Binned mixin sampling is designed to hedge against a mixin sampling distribution which poorly matches expected spend-time behavior. This is important because even if the mixin sampling distribution



**Figure 3.12.:** Projection of the Guess-Newest vulnerability, and the improvement due to our proposed scheme.

matches the overall spend-time distribution, the spend-time behavior of some groups may differ greatly. This difference, if known, could be leveraged to perform temporal analysis.

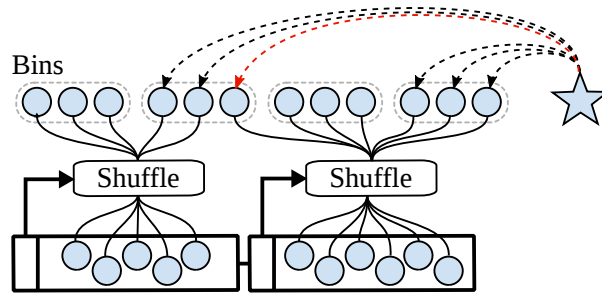
Recall our threat model (Section 3.1) where a forensic attacker Eve has withdrawal records of exchanges as well as deposit records at a merchant site (e.g., AlphaBay). Her goal is to trace Alice’s withdrawal to a deposit at AlphaBay by confirming that it was used as an input of a deposit transaction. Unlike attacks discussed so far, Eve has additional information about Alice, Eve knows Alice’s timezone, i.e. roughly the time of day when Alice receives and sends payments. Eve uses this information to determine the likelihood that each referenced output is the real spend, e.g.,  $G_e = (p_0 = 0.80, p_1 = 0.17, p_2 = 0.02, p_3 = 0.01) = 0.24$  results in Alice’s output having an effective-untraceability of  $2 \times (0.24) + 1 = 1.48$  (defined in Section 3.4.1). However, using binned mixin sampling the same number of mixins would ensure an effective-untraceability  $\geq 2.0$ .



**Binned mixin sampling.** As shown in Figure 3.13 our strategy is to group outputs in the Monero blockchain into sets of some fixed size, called bins, such that each output in a bin is confirmed in the same block or a neighboring block. This ensures each output in a bin has roughly the same age. Any transaction input referencing a transaction output in a bin, either as a mixin or spend, must also reference all other outputs in that bin. Thus, a real spend cannot be distinguished by age from the other mixin outputs in the bin. Additionally, binned mixin sampling ensures that all the outputs in a bin cannot be deduced as spent until the last unspent output in the bin is spent, preventing deduction attacks from reducing the effective-untraceability of an output to less than the bin size. Algorithm 2 specifies our binned mixin sampling procedure.

The simplest way to assign outputs to bins would be to group outputs according to their position within a block; however a malicious miner could, at no cost, choose which outputs are assigned to which bins harming untraceability. Instead we shuffle the assignment of outputs to bins using the block header of the block containing those outputs. Thus, prior to mining a block, a miner cannot know which outputs in that block will be assigned to which bin. To ensure that a spendable output will always be assigned to a bin, any set of outputs that have been confirmed by ten blocks and are not members of a complete bin are merged into the last complete bin. Due to the privacy risks of using coinbase outputs as mixins, our bin assignment scheme could be trivially modified to prevent non-coinbase outputs from being binned with coinbase outputs.

**Choosing the parameters for binned mixin sampling.** If only one bin is referenced, an active attacker could trace a newly created output by broadcasting many attacker controlled outputs. The attack is successful if the targeted output shares a bin with only attacker outputs. Since the number of bins referenced is one when the



**Figure 3.13.:** Binned mixin sampling, an input spends an output (red dotted line) and references outputs in two bins.

targeted output is spent, the spending transaction input can only use mixin outputs from that one bin. Thus all the mixin outputs used by the targeted output will be attacker controlled. To prevent such attacks we require that the number of bins referenced be two or greater.

Another benefit of a larger number of bins is that it provides a larger potential range of referenced output ages, thus obfuscating the exact confirmation time of the spent output. Since we must reference at least two bins and we do not wish to require an onerous number of mixins, we consider bin sizes of four or less; the number of mixins required is  $(\text{number of bins} \times \text{bin size} - 1)$ , thus a bin size of five would require all users to use at least nine mixins. Monero as of the Sept. 15, 2017 hard fork requires at least four mixins.

As this countermeasure is designed to resist worst case attacks we introduce a measure of effective-untraceability, called min-untraceability, bounding untraceability under worst case mixin sampling. By worst case assumptions we mean that we choose inputs that reference outputs with ages that maximize the difference between the spend-time distribution and the mixin sampling distribution, thus minimizing the effective-untraceability (our definition of effective-untraceability is given in Section 3.4.1). Put another way, min-untraceability is the minimum possible effective untraceability given a spend-time and mixin sampling distribution. To

**Table 3.4.:** Min-untraceability for different bin sizes and mixins. Mixins is the total number of mixins referenced and bin size is the number of outputs per bin.  $\varepsilon$  is the maximum percent error

	Bin size	Min-untraceability					
		$\varepsilon =$	0%	25%	50%	75%	100%
5 mixins	1		6.00	5.43	4.33	2.43	1.00
5 mixins	2		6.00	5.18	4.00	2.67	2.00
5 mixins	3		6.00	5.16	4.20	3.35	3.00
7 mixins	1		8.00	7.38	6.09	3.43	1.00
7 mixins	2		8.00	7.02	5.43	3.26	2.00
7 mixins	4		8.00	6.88	5.60	4.47	4.00
8 mixins	1		9.00	8.36	7.00	4.00	1.00
8 mixins	3		9.00	7.76	6.00	4.00	3.00

represent the degree to which the mixin sampling distribution fails to successfully model the spend-time distribution we parameterize our analysis by the maximum percent error,  $\varepsilon$ .

We now provide a formal definition of the maximum percent error  $\varepsilon$  as used by min-untraceability. Denote the spend-time distribution as  $D_S(x)$  and the mixin sampling distribution as  $D_M(x)$  where  $x$  is the the age of an output. Let  $x_{max}$  and  $x_{min}$  be output ages which maximize and minimize the ratios

$$r_{max} = \max_{\forall x} \left( \frac{D_S(x)}{D_M(x)} \right), r_{min} = \min_{\forall x} \left( \frac{D_S(x)}{D_M(x)} \right).$$

The difference between  $r_{max}$  and  $r_{min}$  represents the point of greatest error between the spend-time distribution and the mixin sampling distribution. The maximum percent error  $\varepsilon$  is the error between the spend-time distribution and the mixin sampling distribution defined as

$$r_{min} = (1 - \varepsilon), r_{max} = \frac{1}{1 - \varepsilon}.$$

---

<b>Algorithm</b>	<b>2.</b>	Binned	mixin	sampling	procedure.
------------------	-----------	--------	-------	----------	------------

---

```

SAMPLEBINS(RealOut, NumMixins, BinSize)
  NumBins  $\leftarrow$  (NumMixins + 1)/BinSize;
  RealBin  $\leftarrow$  MapOutToBin(RealOut);
  MixinVector  $\leftarrow$  Copy(RealBin);
  while |MixinVector| < (NumMixins + 1) do
    TxOut  $\leftarrow$  SampleSingleMixin(RealOut, 1);
    if TxOut  $\notin$  MixinVector then
      Bin  $\leftarrow$  MapOutToBin(TxOut);
      MixinVector.appendAll(Bin);
  return sorted(MixinVector);

```

---

Thus, if  $\varepsilon = 0\%$ , a temporal analysis attack can never distinguish between the spend-time and mixin sampling distributions, if  $\varepsilon = 100\%$  a temporal analysis attack can always distinguish the distributions.

In Table 3.4 we compute the min-untraceability for different bin sizes and maximum percent errors using min-untraceability. As shown, increasing the bin size trades off min-untraceability when the maximum percent error is small for increased untraceability when the maximum percent error is large. We argue that a bin size of two is ideal as it maximizes min-untraceability under a small maximum percent error compared to a bin size of three or four, yet still provides an effective-untraceability of 2 against worst-case temporal analysis attacks (attacks which under single mixin sampling would completely trace transactions).

### 3.7. Discussion and Recommendations

We have identified two weaknesses in Monero’s mixin selection policy, and shown that they pose a significant risk of traceability – especially for early Monero transactions. Next, we discuss how these weaknesses can support investigations into present criminal activity, and also offer suggestions for improving Monero’s privacy.

### 3.7.1. Criminal uses of Monero

In 2017 there have been three widely publicized instances of criminal activity involving Monero transactions. Our techniques show that Monero is not necessarily a dead end for investigators.

**AlphaBay:** the most prolific darknet market since the Silk Road (operating between December 2014 and July 2017) began accepting Monero deposits in July 2016, partially leading to the large rise in transaction volume. In July 2017, US law enforcement raided an AlphaBay server and seized 12 000 Monero (worth around \$500 000) [153]. Assuming the AlphaBay server kept logs generated by the default Monero client, the seized logs could include Monero transactions associated with user withdrawals and deposits, including those prior to 2017.

**Shadow Brokers:** From June 2017 onward, the “Shadow Brokers” offered to accept Monero payments for subscription access to zero-day vulnerabilities and exploit tools. They (mistakenly?) advised their hopeful subscribers to publish their email addresses (hexencoded, but publicly visible) in the Monero blockchain, leading to these transactions being identified [161].

**WannaCry:** Ransomware operators received Bitcoin ransomware payments, to a common address. To launder the Bitcoin ransoms, the operators began exchanging them for Monero using the Swiss exchange service ShapeShift. The Swiss exchange subsequently announced their cooperation with US law enforcement, and began blacklisting Bitcoin ransoms. However, \$36 922 have already been exchanged for Monero [66].

In each of these scenarios, an analyst’s goal is to link the criminally-associated transactions to other information, such as accounts at exchanges, which can further their investigation. The analyst starts off having identified several Monero TXOs that belonged to a criminal suspect, and might next ask cooperating exchanges for

information about the relatively small number of related transactions referencing that TXO.

A seller at AlphaBay that received a payment directly into an exchange account could clearly be linked this way. Users following a Monero best practice guide, including “How To Use Monero and Not Get Caught” [4], would have known to avoid this by first passing their coins through a wallet on their own computer; however, for transactions made in mid 2016 to early 2017, they might still be traceable through the “deduction” technique.

In the WannaCry and Shadow Brokers scenarios, since the relevant transaction occurred post-RingCT, deduction is most likely ineffective. However, analysts could still use the temporal analysis and Guess-Newest heuristic to narrow their search at exchanges or to accumulate probabilistic evidence.

### 3.7.2. Recommendations

We make the following three recommendations to the Monero community so that privacy can be improved for legitimate uses in the future.

**The mixing sampling distribution should be modified to closer match the real distribution.** We have provided evidence that the strategy by which Monero mixins are sampled results in a different time distribution than real spends, significantly undermining the untraceability mechanism. To correct this problem, the mixins should ideally be sampled in such a way that the resulting time distributions match.

A report from Monero Research Labs cited the difficulty of frequently tuning parameters based on data collection (especially since the data collection mechanism itself becomes a potential target for an attacker hoping to alter the parameters) [96]. Fortunately, we provide preliminary evidence that the distribution of “spend-times”

changes very little over time. Hence we recommend a sampling procedure based on a model of spending times derived from blockchain data, as discussed in Section 3.6.1.

**Avoid including publicly deanonymized transaction outputs as mixins.** We have empirically shown the harmful effect of publicly deanonymized (i.e. 0-mixin) transactions on the privacy of other users. Since non-privacy-conscious users may make 0-mixin transactions to reduce fees, Monero had instituted a 2-mixin minimum, and recently increased this to 4. However, even 4+mixin transactions may be publicly deanonymized; in particular, as discussed in Section 3.5.1, mining pools have a legitimate interest in forgoing anonymity by publicly announcing their blocks and transactions for the sake of accountability. Thus, we propose that Monero develop a convention for flagging such transactions as “public,” so that other users do not include them as mixins.

**Monero users should be warned that their prior transactions are likely vulnerable to tracing analysis.** A significant fraction (91 %) of non-RingCT Monero transactions with one or more mixins are deducible (i.e., contain at least one deducible mixin), and therefore can be conclusively traced. Furthermore, we estimate that among all transaction inputs so far, the Guess-Newest heuristic can be used to identify the correct mixin with 80 % accuracy. Even after accounting for publicly deanonymized transactions such as pool payouts, we find that at least a few hundred transactions per day in mid 2016 and more than a thousand transactions per day from September 2016 through January 2017 would be vulnerable. Furthermore, we estimate that at most a quarter of these can be attributed to illicit marketplaces like AlphaBay. These users might have incorrectly assumed that Monero provided much higher privacy, especially for transactions taking place in late 2016. Because many transactions on AlphaBay are criminal offenses, with statutes of limitations that will

not expire for many years (if ever), these users remain at risk of deanonymization. We stress that illicit businesses tend to be early adopters of new technology, but there exist many legitimate reasons to use privacy-centric cryptocurrencies (e.g., a journalist protecting her sources). While such scenarios are less visible, their users face the same risk of deanonymization.

Towards fulfilling this recommendation, we released an initial draft of this paper to the Monero community. We believe it has been in the best interest of Monero users that we offered this warning as soon as possible, even before countermeasures have been deployed. One reason for our decision is that the data from the Monero blockchain is public and widely replicated, and thus delaying the release would not mitigate post-hoc analysis, which can be carried out at any future time. Second, countermeasures in future versions of the Monero client will not affect the vulnerability of transactions occurring between the time of our publication and the deployment of such future versions.

Complementing this paper, we launched a block explorer<sup>2</sup>, which displays the linkages between transactions inferred using our techniques. We recommend additionally developing a wallet tool that users can run locally to determine whether their previous transactions are vulnerable.

### **3.7.3. Subsequent impact**

In Monero v0.13.0, released in October of 2018, the developers implemented an improved sampling procedure in the wallet based on the observed spend-time distribution. They also fixed the number of mixins to 10 in order to prevent users from selecting uncommon ring sizes, which could aid in tracing transactions.

---

<sup>2</sup><https://monerolink.com>



Following our work, there was a variety of further research into the privacy provisions of Monero. Hinteregger and Haslhofer [82] analyse the privacy risk stemming from Monero forks: as every input needs to reveal a unique key image, spending the same output on different chains can reveal the true spend simply by taking the intersection of the referenced outputs. For inputs deduced in this way, they report that the effectiveness of the Guess-Newest heuristic has dropped, however, coins cashed out on forks might be skewed towards those held for a longer time. Yu et al. [164] provide a statistical analysis of the deducibility attacks using closed sets. More recently, Monero has also strengthened its defenses against network-level attacks based on the work by Fanti et al. [60].

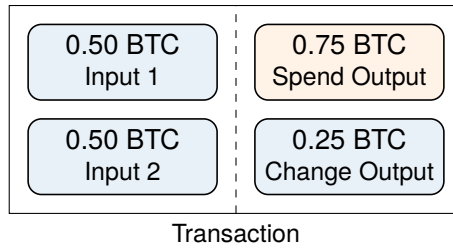
# 4

## Resurrecting Address Clustering in Bitcoin

### 4.1. Introduction

**Motivation.** Blockchain analysis techniques are essential for understanding how cryptocurrencies like Bitcoin are used in practice. A major challenge in analysing blockchains is grouping transactions belonging to the same user. Because users can create an unlimited amount of addresses, each of which can receive and send coins, their activity may be split among a multitude of such addresses. Techniques to group activity of individual users are commonly referred to as *address clustering heuristics*, as they focus on identifying the addresses under an individual user's control using heuristic assumptions about how their transactions are created. As the term *heuristic* suggests, address clustering today is more intuitive than rigorous; our overarching goal in this chapter is to elevate it to a science.

There are at least four applications for which accurate address clustering is important. First, a law enforcement agency may be interested in evaluating the transactions of a specific entity (e.g., a specific exchange, trader or gambling service). They may supplement their own investigation of the entity's behavior with a set of reliable heuristics to identify relevant transactions. Second, and conversely, the ability to accurately determine a user's transactions directly impacts

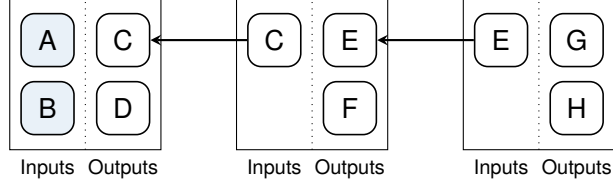


**Figure 4.1.:** Schema of a typical Bitcoin transaction with two inputs and two outputs: the spend output is the intended payment, the change output returns the surplus coins to the sender. Each input and output is associated with an address.

their privacy. This tension between law enforcement needs and everyday users' privacy is inherent to cryptocurrencies due to their transparency and pseudonymity. Advocates from one side push for greater privacy and from the other side for stronger regulation. To better understand this tug-of-war, it is important to quantify how reliable heuristics are in practice. Third, accurate grouping of transaction activity is important for aggregate analyses such as studying economic activity over time. This usually requires a full clustering of all addresses on the blockchain. Finally, the problem of address clustering itself may be interesting for researchers outside of cryptocurrencies. For example, it may pose as an application domain for machine learning models and could be used as a benchmarking application.

**Goals.** The current state of address clustering techniques available to researchers is sub-optimal in multiple ways. The most common heuristic, *multi-input*, groups addresses that are jointly used in inputs of a transaction (cf. Figure 4.1) [135, 137]. This heuristic is easy to apply, moderately effective in practice, [79] and widely used. However, it misses addresses that are never co-spent with other addresses (cf. Figure 4.2).

Many of these addresses can be identified using *change address* heuristics: as coins in Bitcoin cannot be spent partially, transactions need to return the surplus



**Figure 4.2.:** Clustering with only the multi-input heuristic misses addresses C and E that are not co-spent with other addresses.

value back to the user who created the transaction. Identifying the change output thus allows grouping the associated address with the inputs' addresses. While the effectiveness of change address detection has been demonstrated empirically and through simulation (e.g., [8, 103]), it remains difficult to assess how well it works in practice. As a result, clustering techniques are applied inconsistently across studies: many forgo change address clustering (e.g., [86, 87, 97, 140]), whereas some simply apply a single change heuristic (e.g., [41, 127]) .

A major issue is the lack of ground truth data available to researchers. In the context of change output detection, ground truth consists of a set of transactions for which the change output is known. Such a dataset allows to assess the accuracy of individual heuristics aiming to identify the change output of a transaction. But because the Bitcoin blockchain is used for a variety of different use cases and by a diverse group of users, which may both change significantly over time, ground truth needs to reflect this diversity in order to allow for a reliable assessment. Such data is hard to collect, and, even if available, is unlikely to be made public, e.g., due to privacy concerns. We are only aware of one approach exploiting weaknesses in a specific type of lightweight client [121], which allowed to extract the addresses of 37 585 wallets to assess four different clustering heuristics. Blockchain intelligence companies might have access to manually curated and refined data sets and clusterings, but their techniques and data aren't generally available to researchers (or only shared in limited form, e.g., [78, 160]). As a result, analyses of

clustering heuristics often fall short of quantifying their accuracy and instead resort to analyzing the resulting clusterings (e.g., [36, 165]).

Considering this state of affairs, our goals in this chapter are to address the lack of ground truth data and assessment methods, develop new techniques to apply heuristics to predict change and use them to create improved clusterings.

### **Contributions, methods and findings.**

1. **A new ground truth method and dataset:** We put forward a procedure to select and filter transactions for which the change output has been revealed on the blockchain. Our approach exploits that future transactions of users can reveal change outputs in past transactions. We take specific care evaluating the data set with regards to potential issues, such as violations of our core assumption or existing cluster collapse. We extract a set of 35.26 million transactions with known change (carefully filtered down from 53 million candidate transactions) that can be used as ground truth for validation and prediction. Our method does not rely on interaction with intermediaries, though we use address tag data to assess the quality of our data set and methods. The data set can be continuously updated, improved and shared with other researchers. (Section 4.2)
2. **Evaluating existing heuristics:** We've compiled and evaluate a set of 26 change address heuristics based on previous literature and community resources. Evaluated on our ground truth, most change address heuristics have few false positives at low to medium true positive rates. We find that due to changes in the protocol and usage patterns, heuristics wax and wane in their effectiveness over time, showing the need to use multiple heuristics and combine them in an adaptive way rather than rely on a fixed algorithm. We also report the

heuristics' overall coverage (i.e., how often they return a result) for transactions with unknown change. (Section 4.3)

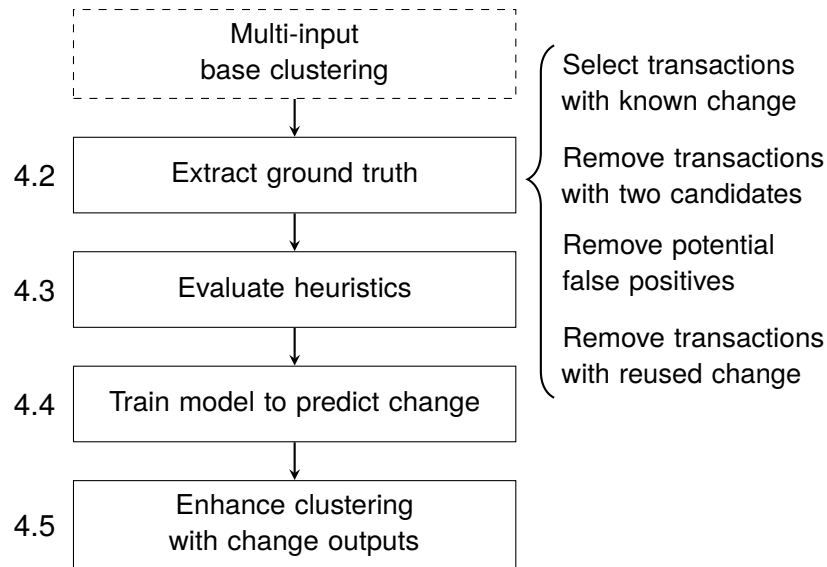
3. **Improved prediction:** We use a random forest classifier to identify change outputs and compare it against a baseline: the majority vote of individual heuristics. While machine learning has been used to classify the type of entity behind a transaction (e.g., [15, 78, 83, 86, 94, 150, 160]), to the best of our knowledge our work is the first to apply it to the problem of change identification. We find that a random forest model outperforms our baseline threshold voting mechanism, especially for low false positive rates (to prevent cluster collapse). For example, targeting a false positive rate below 0.1 % the random forest model correctly detects twice as many change outputs. (Section 4.4)
4. **Preventing cluster collapse:** We analyze the clustering that results from the predicted change outputs with regards to cluster collapse. We find that a naive clustering of predicted change outputs leads to cluster collapse, despite choosing a high threshold to prevent false positives. We then apply constraints to the union-find algorithm underlying our clustering, aiming to prevent cluster collapse that stems from frequent, repeated interaction between entities. This technique prevents large-scale cluster collapse while still enhancing a majority of the involved clusters. (Section 4.5)
5. **Assessing impact:** We assess the impact our enhanced clustering has on two exemplary applications: cash-out flows from darknet markets to exchanges and the velocity of bitcoins. We find that the results of such typical longitudinal analyses are off by at least 11–14 % if they don't fully account for clustering. (Section 4.6)

Our process is summarized in Figure 4.3. We discuss our findings in Section 4.7 and conclude in Section 4.8

**Limitations.** Our results are limited by the availability of “real” (i.e. manually collected and validated) ground truth. As such, our analysis should be treated as a first step towards better understanding the feasibility of change address detection and clustering. However, we do not expect our high-level insights to change significantly in the light of minor corrections to our ground truth data set. We invite the research and blockchain community to evaluate our data set using their own ground truth data or analysis techniques. An interesting avenue for future research would be to build a privacy-preserving tool that allows to crowd-source the validation of both the ground truth data and prediction models, to which individual users could connect their wallet software.

Our extraction mechanism relies on change outputs revealed by the multi-input heuristic. This heuristic is effective in practice [79] and widely used, but vulnerable to false positives from techniques like CoinJoin and PayJoin transactions that are intentionally designed to break the heuristic (e.g., [51, 99, 102, 110]). While we take measures to detect CoinJoin transactions and pre-existing cluster collapse, some errors can remain. Furthermore, entities that more effectively prevent address reuse are less likely to be included in our data set.

We adopt the term *clustering* used in the blockchain community, however, our underlying implementation sequentially processes transactions and uses a union find algorithm that does not retain individual distance scores. In practice it may be desirable to give analysts a choice between different clustering outcomes (e.g., concerning the order in which clusters are merged or the tradeoff between false positive and true positive rates).



**Figure 4.3.:** Our process in this chapter

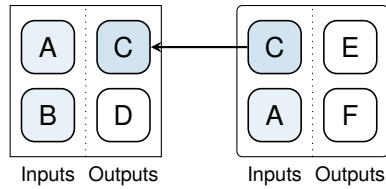
Finally, we work with the Bitcoin blockchain, currently the most popular cryptocurrency by market volume. Our methods are applicable to similar cryptocurrencies, potentially with a different set of heuristics, but may be less effective if transactions are more homogeneous.

**Data.** We plan to make our ground truth data set publicly available and invite other researchers to evaluate our data set using their own private ground truth or analysis techniques. While identifying the change output of a transaction has privacy implications for individual users, we believe that making our data set public does not create additional privacy risk as all the data is already publicly available on the blockchain and our methods are easy to reproduce.

## 4.2. Building a Ground Truth Data Set

**Core assumption.** We focus on the feasibility of detecting the change output in Bitcoin transactions with exactly two spendable outputs, by far the most common





**Figure 4.4.:** Address C is merged into the same cluster as addresses A and B by the multi-input heuristic, thereby revealed as the change address in the first transaction.

type of transaction as of June 2021 (75.8 % of all transactions, see Figure 4.5). Our core assumption is that one of these outputs is a payment, and the other output receives the change. We call this type of transaction a *standard* transaction, as they are created by typical end-user wallet software.<sup>1</sup>

For transactions with only one output there is no good indicator to directly and reliably determine whether the output belongs to the same user. The transaction may correspond to a user sweeping the balance of their wallet, but the destination address may not be under the same user’s control (e.g., it could be managed by a cryptocurrency exchange).

Transactions with more than two outputs are less likely to originate from an ordinary wallet. They may belong to an exchange that batches payouts to multiple users, or correspond to a restructuring of their internal hot and cold wallets. As a result, our assumption that exactly one of the outputs receives change may not hold. Large numbers of outputs could also indicate mixing services or CoinJoin transactions, where the funds of multiple users are mixed. Determining change in CoinJoin transactions requires solving a subset-sum problem (e.g., [71, 102, 110]) and is outside the scope of this work.

<sup>1</sup>There exist a separate notion of a standard transaction, namely those that pass the `isStandard` test of the Bitcoin reference implementation that checks whether a transaction uses one of a handful of default script types.

**Method.** Our approach leverages the phenomenon that change outputs are sometimes revealed by the multi-input heuristic at a later point in time due to address reuse. Figure 4.4 shows an example of how such disclosure may unintentionally happen on the blockchain: a user spends coins at addresses  $A$  and  $B$ , their wallet directs the change to a new address  $C$ . Later, they spend the change at address  $C$  along with other coins at address  $A$ . At this point, the multi-input heuristic reveals that  $A$ ,  $B$  and  $C$  belong to the same user, thereby revealing  $C$  as the change address in the first transaction. By identifying transactions that have their change revealed in this way, we can build a ground truth set of transactions with known change.

**Comparison to interactive collection.** We briefly discuss the advantages and disadvantages of our approach to collecting ground truth interactively. We could download a Bitcoin wallet and send bitcoins to a number of different addresses, thereby creating a corpus of transactions for which both the spend and the change output are known. However, in order to be able to learn and generalize from our ground truth data it should capture the expected heterogeneity of implementations and use cases over Bitcoin’s entire history. An interactive collection would likely yield ground truth inferior in three dimensions: variety, scale, and the collection time frame.

Heterogeneous ground truth requires transactions from a *variety of different use cases and entities*. Compared to prior deanonymization studies that identified address clusters of specific entities by interacting with them (revealing some of the intermediary’s addresses, e.g., [103]), we are interested in the change of transactions made by those intermediaries. Purchasing an item from an online merchant can reveal one of their addresses, but we do not learn about change in any of the merchant’s transactions, only about our own transaction that pays them. The only conceivable way to learn about change in an intermediary’s transactions is to induce

them to make a transaction to an address under our control. This may be possible with exchanges, where we could first deposit and then withdraw funds, but is not applicable to many other intermediaries.<sup>2</sup> Our method, instead, is not limited to a small set of intermediaries of our choosing.

Second, interactive collection would be hard to *scale* beyond a few hundred transactions, as we would need to individually engage with a variety of intermediaries and wallet implementations. This cannot easily be automated. While Nick [121] was able to collect data on a larger scale from exploiting a vulnerability in a specific type of lightweight client, his method is not transferable or generalizable to other types of wallets. Our approach, instead, yields a data set of millions of transactions.

A third issue of collecting data interactively is the *time frame*. Interactive collection, as described above, cannot be done retroactively and is therefore limited to a short, current time window. This limits its utility as the resulting data set wouldn't capture shifting patterns over different epochs of Bitcoin's history. Our non-interactive approach, on the other hand, is applicable to transactions from Bitcoin's entire history.

Our method has a few important limitations. First, because we extract ground truth data non-interactively from the blockchain, we are not able to fully verify its correctness. Second, our core assumption that exactly one of the outputs is a change output may not hold in every scenario. For example, a user sending funds to an address under their control could lead to ambiguous or incorrect labeling of change outputs. This is predominantly an issue for our collection process. We later take care to remove transactions that are more likely to violate the core assumption in this way. Similarly, there could be instances where none of the outputs is a change output because a user made a payment to two different entities using a perfectly matching

---

<sup>2</sup>It could also raise ethical and legal questions, e.g., when interacting with gambling services or intermediaries in other countries.

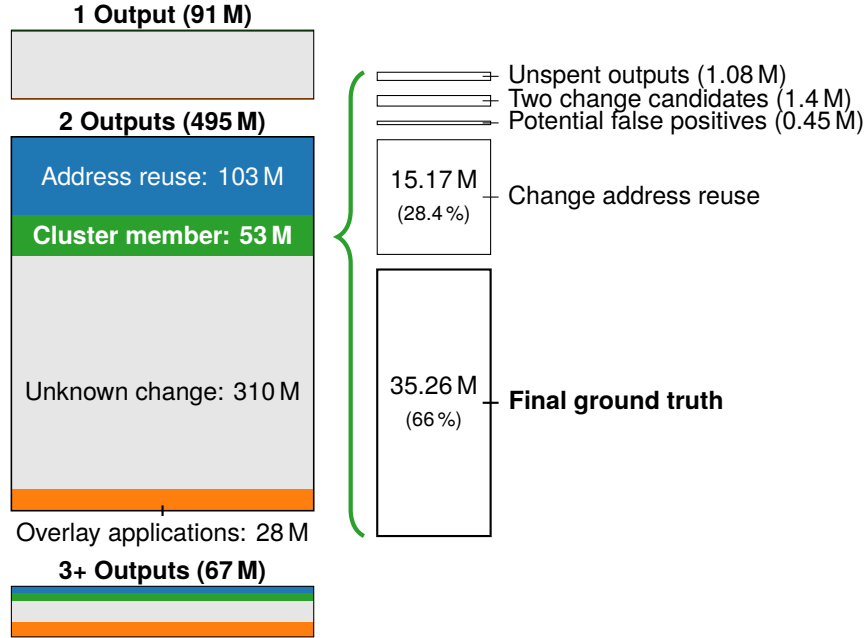
set of inputs that does not require change to be returned. This is an issue during the subsequent prediction of outputs and could lead to cluster collapse. While this becomes more likely the more UTXOs a wallet can choose from, we expect those instances to be rare. Third, as our method relies on address reuse, the resulting transaction corpus could be biased towards entities or wallet implementations that are more prone to reuse and merge addresses. It might contain fewer instances of transactions created with wallets that more effectively discourage address reuse.

#### **4.2.1. Data collection and overview**

We use and build upon BlockSci v0.7 [87], an open-source blockchain analysis framework that provides fast access to blockchain data upon which we can implement custom heuristics and extraction procedures. We parse the Bitcoin blockchain until the end of June 2021 (block height 689 256) and create a *base clustering* using the multi-input heuristic (where we heuristically exclude CoinJoin transactions).

As of June 2021, the blockchain contains 91 million transactions with one output, 495 million with two outputs, and 67 million with three or more outputs (Figure 4.5). We divide the transactions into mutually exclusive categories. Transactions containing unspendable OP\_RETURN outputs often signal the use of an overlay application that stores metadata in the blockchain [16]. Such transactions may have specific rules for how they are constructed, potentially making change detection unreliable.

Both transactions reusing an input address as well as transactions where cluster membership (i.e. the multi-input heuristic) reveals a change output have their change output identified. Direct address reuse however makes change identification trivial and applying further change heuristics is never necessary. We thus only use transactions where the change has been revealed by multi-input clustering as the



**Figure 4.5.:** Distribution of different types of transactions in the Bitcoin blockchain until June 2021. Transactions with two outputs and change revealed through cluster membership form the basis of our ground truth data, which we further refine down to a ground truth data set of 35.26 million transactions.

basis to construct our ground truth data set. For the remaining transactions, i.e. those with yet unknown change, we will later try to predict their change output.

#### 4.2.2. Refining the candidate set of ground truth transactions

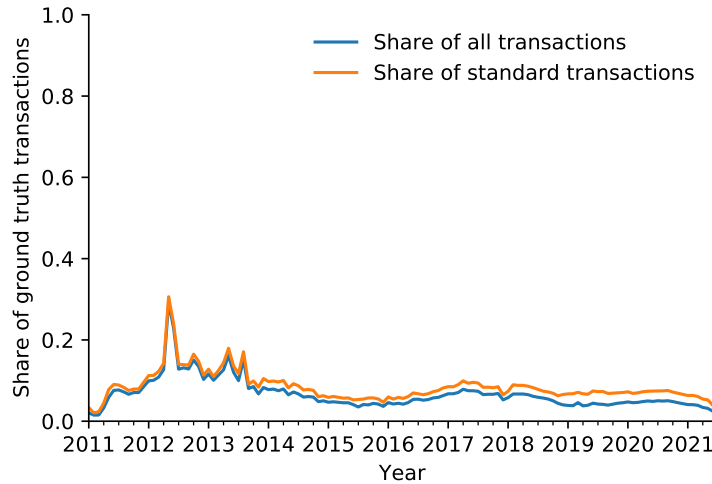
Our candidate set of ground truth transactions consists of transactions with two outputs (ignoring overlay transactions) where no input address is reused for change and where at least one output is in the same base cluster as the inputs. This yields a total of 53.41 million transactions. We further filter the transactions as follows (see Figure 4.5 for a visual breakdown and Section B.3.1 for an extended description):

1. We remove 1.08 million transactions with unspent outputs, as our subsequent analyses rely upon the spending transactions being known.

2. For 0.97 million transactions both outputs are in the same base cluster as the inputs, violating our core assumption. We remove these transactions. We also find that some base clusters are more likely to produce such transactions. We thus exclude transactions from base clusters where more than 10 % of transactions exhibit such behavior. This removes an additional 0.48 million transactions in 9967 base clusters.
3. We check our base clustering for preexisting cluster collapse, which could create false positives. We remove 0.37 million transactions belonging to the Mt.Gox supercluster (cf. [79, 103]) as well as 0.09 million transactions from one possible instance of cluster collapse detected using address tags from the website WalletExplorer.com.
4. We find many instances where the change address did not appear in the inputs, but had been seen before and was known to be the change at the time the transaction was created. For example, there are 5.77 million transactions originating from the gambling service “SatoshiDice” that use only a total of 50 change addresses, and 1.27 million transactions from “LuckyB.it” that use a single change address. For such transactions, applying change address heuristics is never necessary. We remove 15.17 million transactions where the change output was already known at the time the transaction was created.

#### **4.2.3. Assessing the final set of ground truth transactions**

Next, we assess the composition of our ground truth data set and compare it to transactions in the blockchain overall. This is useful to ensure that it contains heterogeneity with regards to scale, time frame and variety (see discussion above),

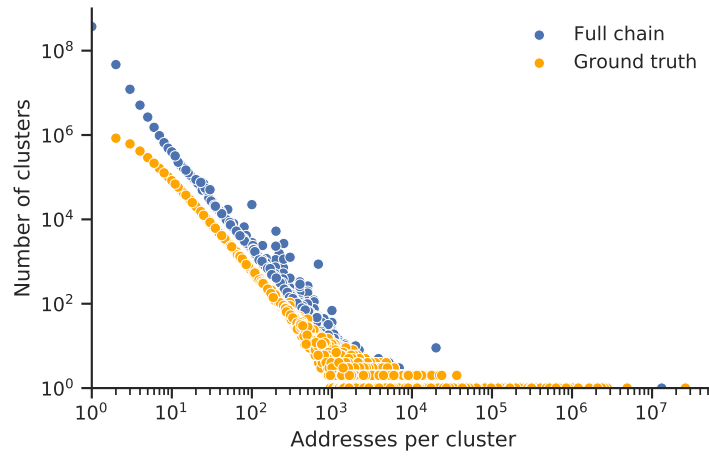


**Figure 4.6.:** Share of ground truth transactions of all and standard transactions over time.

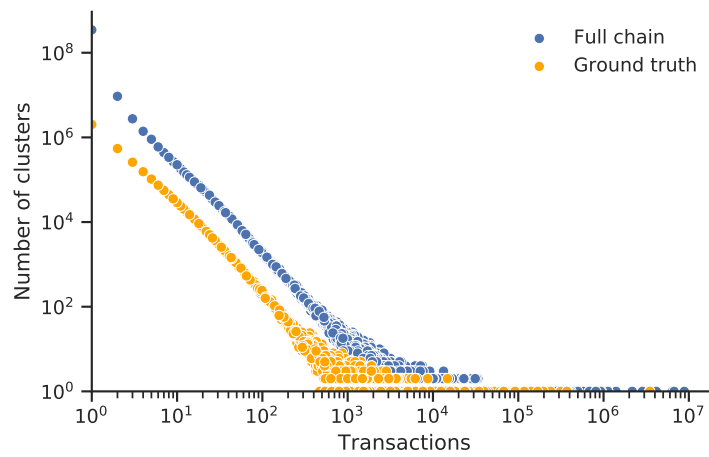
as well as to spot potential biases in the ground truth that could result from our selection process.

**Scale and time frame.** Our ground truth of 35.26 million transactions makes up about 7.6 % of standard transactions and about 5.4 % of all transactions. Figure 4.6 shows that those percentages are relatively stable over time.

**Variety of included clusters.** Our ground truth includes transactions from 3.58 million base clusters. Figure 4.7 shows the distribution of address counts of base clusters that are represented with at least one transaction in our ground truth. Our ground truth contains transactions from base clusters of all sizes, giving us confidence that it can be representative of the blockchain overall. The share of base clusters from which transactions are included is higher towards clusters with larger number of addresses. This is likely a side-effect of our selection process, as base clusters with more addresses may be more likely to combine addresses and thereby reveal change.



**Figure 4.7.:** Number of base clusters represented in our ground truth by total address count.



**Figure 4.8.:** Number of transactions in ground truth and full blockchain per base cluster.



**Table 4.1.:** Comparison of transaction characteristics between ground truth transactions and transactions with 2 outputs for which change is unknown.

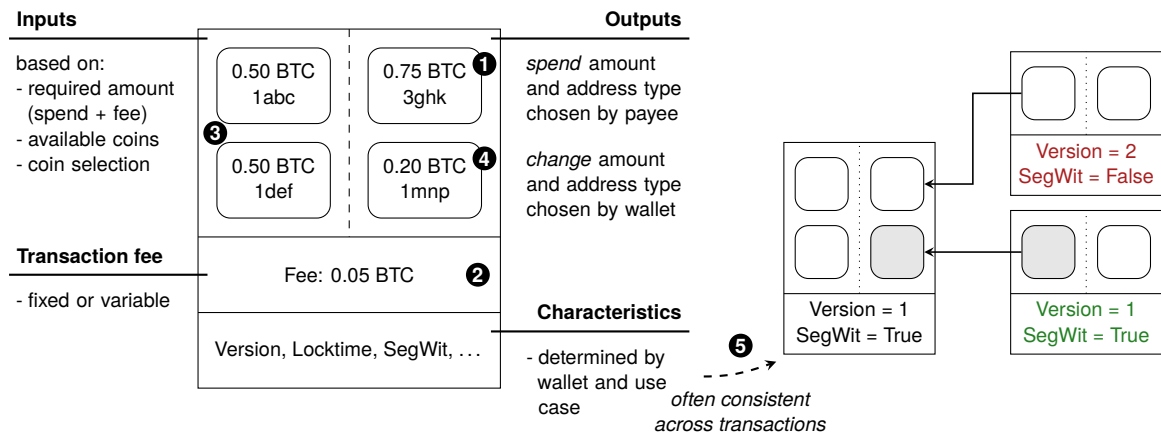
Characteristic	Ground truth (%)	Remaining (%)
1 Input	38.99	78.76
2 Inputs	22.09	13.61
3+ Inputs	38.92	7.63
Version = 1	79.83	80.68
Locktime > 0	25.25	24.60
RBF	3.57	6.22
SegWit	18.30	27.10
$n$ (in million)	35.26	309.65

Figure 4.8 shows the number of transactions per base cluster included in the ground truth compared to the total number of transactions per cluster, showing an overall similar distribution. The largest number of transactions from a single base cluster is 3.49 million, which has 8.85 million transactions in total. We did not find a label for it on WalletExplorer.com. The second highest number of transactions is 383 519, again from an unlabeled cluster.

**Transaction composition and use of protocol features.** Table 4.1 compares characteristics of transactions in our ground truth data to those of standard transactions with yet unknown change, including the number of inputs as well as a number of important protocol features (an overview and description of the protocol characteristics used in this chapter is available in Section B.1). Transactions in the ground truth data set notably tend to have more inputs than those in the set of transactions with unknown change. This is an expected artifact of our selection method, which relies on transactions with more than one input to reveal change outputs. The share of transactions using SegWit serialization or allowing for fee bumping (RBF) is also higher in the set of remaining transactions.

**Table 4.2.:** Change heuristics proposed in the literature and used in this work.

Heuristic	Notes and limitations	Used	Refs.
<b>Optimal change:</b> There should be no unnecessary inputs: if one output is smaller than any of the (2+) inputs, it is likely the change.	Only applies to transactions with 2+ inputs. We use two variants, one ignoring and one accounting for the fee.	✓	[121, 134]
<b>Address type:</b> The change output is likely to have the same address type as the inputs.	Wallets could use different address types to obfuscate the change output.	✓	[87, 134]
<b>Power of ten:</b> As purchase amounts may be rounded, and the change amount also depends on input values and the fee, it is more likely to have fewer trailing zeros.	We use six different variants, which are partially redundant.	✓	[87, 134]
<b>Shadow address:</b> Many clients automatically generate fresh change addresses, whereas spend addresses may be more easily reused.	Modern wallets discourage reuse of receiving addresses. We do not use the heuristic because our ground truth is filtered based on address freshness.	x	[8, 103]
<b>Consistent fingerprint:</b> The transaction spending a change output should share the same characteristics. We use 17 variants based on the following characteristics: <ul style="list-style-type: none"> <li>• input/output counts and order</li> <li>• version</li> <li>• locktime</li> <li>• serialization format (SegWit)</li> <li>• replace-by-fee (RBF)</li> <li>• transaction fee</li> <li>• input coin age (zero-conf)</li> <li>• address and script types</li> </ul>	False positives are possible when a wallet implementation or the protocol change. We only consider characteristics after they are available in the protocol. Section B.1 describes the characteristics we use in more detail.	✓	[23, 134]



**Figure 4.9.:** Schema of how transactions are created, and how consistency of a transaction's fingerprint allows to identify change.

## 4.3. Evaluating Individual Change Heuristics

### 4.3.1. Background on change address detection

The Bitcoin protocol does not explicitly distinguish between change and spend outputs. However, wallets create change outputs automatically to return surplus value when users make payments (cf. Figure 4.9). We briefly describe how this allows to identify change and present the heuristics that have been proposed in the literature (see Table 4.2 for details, limitations and references).

**Spend output.** For standard transactions, the user will typically be given a specific payment amount and an address to which the bitcoins should be sent (1). Payment amounts in Bitcoin are denominated in satoshi, with one bitcoin equal to  $10^8$  satoshi. At current and historic exchange rates, a single satoshi is worth only a fraction of a cent. Merchants may thus round payment values to make it easier for users to enter the correct amounts in their wallet, and transfers initiated by users may use round values as well. Change can hence be distinguished from spends by potentially having fewer trailing zeros (*power of ten heuristic*).

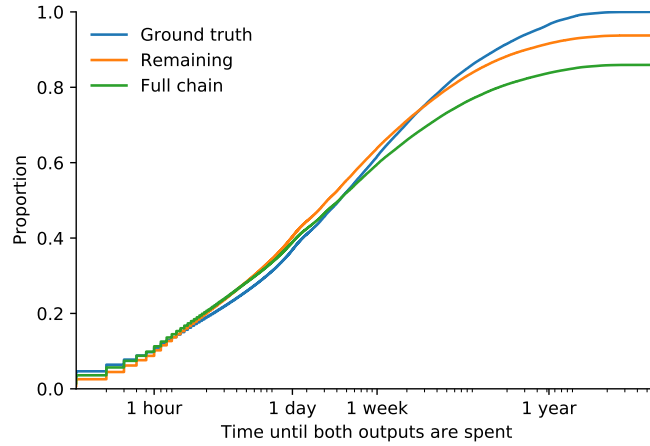
**Input selection.** After the spend amount and a transaction fee have been determined, the wallet chooses a set of coins that covers the sum of both amounts (3). While this selection procedure is not standardized and can differ between wallet implementations (cf. [1, 56]), some behavior is common to many wallets, such as not including unnecessary inputs (*optimal change* heuristic).

**Change output.** The change output is automatically created by the wallet to return the surplus funds. Often, a fresh address is generated (i.e., one that has never received coins before). In the past it was common that payment addresses were reused, allowing to determine change based on this behavior (*shadow* heuristic / *one-time change*). We note that our ground truth data set is filtered based on address reuse (see Section B.3.1). Because this filtering effectively determines the performance of the heuristic, we decide not to use it in our subsequent analyses.

While the address type of the receiving address is determined by the payee, wallets usually use consistent address types. A single output with a different address type than the inputs might indicate a spend output, thus revealing the change (*address type* heuristic).

**Consistency of transaction fingerprints.** The heuristics above are based on expected behavior that should apply to many common wallet implementations. Furthermore, transactions can make use of protocol features that may allow to discriminate between different types of wallets. Such a fingerprint would allow to apply custom heuristics to subsets of transactions.

Transaction fingerprints can also be used for change identification. Assuming that users don't change wallets very often, characteristics should be consistent across multiple transactions. When a wallet spends the change of a previous transaction, those two transactions' fingerprints should be similar (*consistent fingerprint* heuristic,



**Figure 4.10.:** Time until both outputs are spent for transactions in our ground truth data, in the remaining standard transactions and the blockchain overall. Time until spent is set to infinity if not both outputs of a transaction are spent.

see (5) in Figure 4.9). We are not aware of any prior work that has evaluated this across the range of available protocol characteristics. We also note that this heuristics requires the outputs to be spent, whereas the previous heuristics are *universal* as they can be applied to all transactions, including those with unspent outputs. We selected our ground truth data such that it only contains transactions where both outputs are spent, so the consistent fingerprint heuristic can be applied to all of them.

Figure 4.10 shows the distribution of time until both outputs are spent for transactions in the ground truth data set as well as remaining standard transactions with yet unknown change. Overall, the outputs of more than half of all transactions are spent in under a week, making the consistent fingerprint heuristic highly applicable to a majority of transactions, including recent ones.

**Table 4.3.:** True and false positive rates of heuristics applied to transactions in the ground truth data set.

Heuristic	Ground Truth		Remaining
	TPR	FPR	Coverage*
<i>Universal heuristics</i>			
Optimal change	0.306	0.026	0.133
• incl. fee	0.239	0.020	0.096
Address type	0.237	0.031	0.369
Power of ten			
• $n = 2$	0.467	0.012	0.383
• $n = 3$	0.420	0.006	0.311
• $n = 4$	0.375	0.005	0.253
• $n = 5$	0.302	0.006	0.173
• $n = 6$	0.211	0.005	0.104
• $n = 7$	0.107	0.001	0.048
<i>Consistent fingerprint</i>			
Output count	0.283	0.129	0.445
Input/output count	0.263	0.107	0.568
Version	0.245	0.004	0.320
Locktime	0.307	0.003	0.363
RBF	0.075	0.003	0.114
SegWit	0.191	0.021	0.260
SegWit-conform	0.021	0.001	0.028
Ordered ins/outs	0.262	0.053	0.443
Zero-conf	0.100	0.061	0.214
Absolute fee	0.117	0.025	0.305
Relative fee	0.042	0.008	0.204
Multisignature	0.140	0.001	0.154
Address type			
• P2PKH	0.239	0.014	0.312
• P2SH	0.269	0.015	0.334
• P2WPKH	0.181	0.019	0.256
• P2WSH	0.063	0.007	0.082
All address types	0.294	0.023	0.392

\*Coverage denotes share of standard transactions with yet unidentified change where the heuristic returned exactly one output.

### 4.3.2. Evaluating individual heuristics

We start by evaluating the universal and fingerprinting heuristics individually. We explicitly encode our constraint that only one of the outputs can be the change. Thus, applied to a transaction, the heuristic may either return an individual output if it is the only output determined to be change, or no output otherwise. Let  $h$  be a heuristic applied to transaction  $t$  which returns a set of potential change outputs, then our constrained heuristic  $h'$  returns:

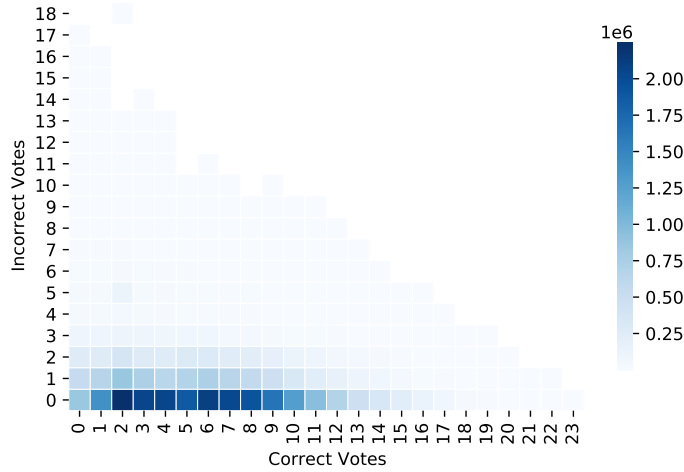
$$h'(t) = \begin{cases} h(t) & |h(t)| = 1 \\ \emptyset & \text{otherwise} \end{cases}$$

This constraint is crucial to prevent cluster collapse: if a heuristic cannot determine a unique change output, we'd rather cluster none of the outputs than both, as clustering both would violate our core assumption and may lead to the merging of two clusters not belonging to the same user.

In Table 4.3 we report the individual heuristics' true and false positive rate (TPR/FPR) for identifying change outputs in our ground truth. We also report the share of all standard transactions with unknown change for which the heuristic returns a unique output (denoted as "coverage").

Most heuristics have a low FPR, with four fingerprinting heuristics being the exception: output count, input/output count, input/output order, as well as zero-confirmation spending. The power of ten heuristic has a notably high TPR compared to many other heuristics (its variants are in many cases redundant).

Comparing the TPR of the heuristics to their coverage, we see moderate positive correlation ( $r = 0.55, p = 0.004$ ), meaning that heuristics that identify more change outputs in our ground truth also tend to be more applicable to the remaining standard transactions. One outlier is the optimal change heuristic, which only



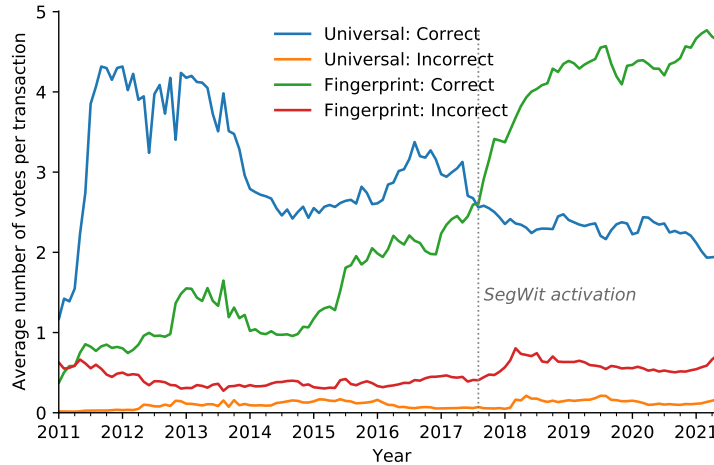
**Figure 4.11.:** Number of votes from heuristics on transactions in the ground truth data set

returns a result in 13.3 % of the remaining transactions (compared to a TPR of 30.6 % in the ground truth). This is due to the difference in the number of inputs for these two sets (see Table 4.1). At the same time, many of the fingerprinting heuristics appear to be more applicable to the remaining transactions, signaling higher heterogeneity among those.

27.49 million transactions have votes from a universal heuristic, and 32.17 million have votes from a fingerprint heuristic. 858 582 transactions don't have any predictions. Among the 34.40 million transactions with at least one vote, 33.34 million (96.91 %) have at least one *correct* vote.

Figure 4.11 shows the number of (correct and incorrect) votes received by each transaction. Figure 4.12 further breaks down the average number of correct and incorrect predictions per transaction over time, grouped by the type of heuristic. We notice three important trends: the universal heuristics drop over time, likely due to some of the power-of-ten variants becoming less useful (as they contain redundant information, we provide additional plots where the heuristics are aggregated in Section B.2). The consistent fingerprint heuristics instead see a steady uptick in the number of correct votes. This highlights how the increasing variety of protocol





**Figure 4.12.:** Average number of correct and uncorrect votes per transaction and type of heuristic in the ground truth data set, over time

features also raises their utility for detecting change outputs. Finally, there’s an uptick in both correct and incorrect fingerprint votes in late 2017 and early 2018, when wallet implementations started to switch to SegWit transaction serialization and address formats (e.g., [21, 142]).

For the following analyses, we exclude the 858 582 transactions that don’t have any predictions in order to reduce the chance of false positives. When we later apply the heuristics to the full blockchain, we will also skip transaction without votes from at least one heuristic.

## 4.4. Combining Heuristics

A clear disadvantage of choosing a single individual heuristic for change output detection is that they apply only to a subset of transactions (cf. Table 4.3). Furthermore, some heuristics may be more applicable during certain epochs of Bitcoin’s history than others. In contrast to prior work (e.g., an evaluation of three change

heuristics [121]), we also have a larger variety of heuristics available that enable new ways of combining them. Here, we consider two approaches.

#### **4.4.1. Threshold vote**

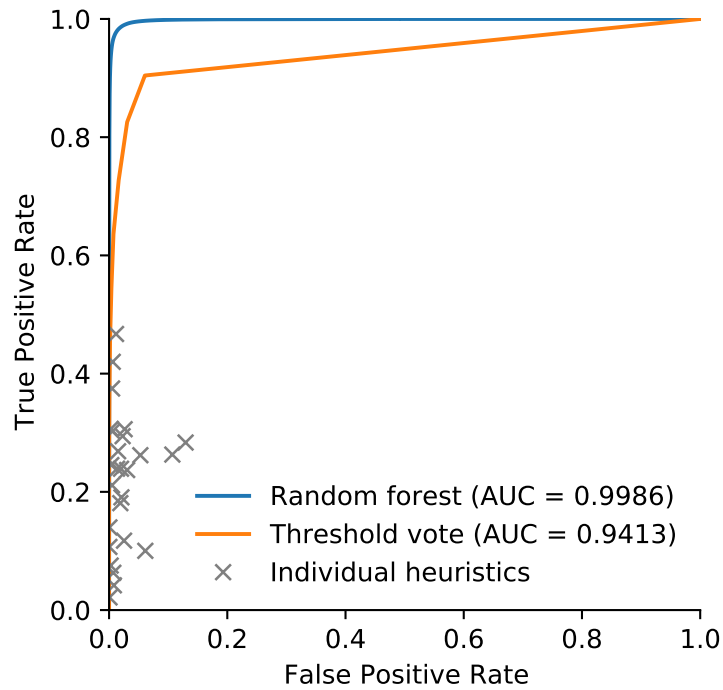
Figures 4.11 and 4.12 suggest that, in general, a majority of cast heuristic votes should produce the correct outcome. However, the number of votes cast varies among transactions, and individual votes could be incorrect. We thus compute a threshold vote: if there are at least  $t$  more votes for output  $a$  than for output  $b$ , then output  $a$  is considered the change. Changing the threshold  $t$  thereby allows the analyst to require higher degrees of confidence.

We use all of the heuristics listed in Table 4.3 to compute the threshold vote on the full ground truth data set (as there is no training involved, we do not split the data set). The resulting ROC curve is shown in Figure 4.13 (for comparison, we also plot the FPR and TPR of the individual heuristics). We can achieve an ROC AUC of 0.94, and, for example, a 37.0 % true positive rate (TPR) below a false positive rate (FPR) of 0.1 % with a threshold of  $t = 7$ .

Using a threshold vote may not be ideal as the individual heuristics have varying true positive and false positive rates, and some might be more or less reliable during different periods of Bitcoin's history. Rather, a specific subset of heuristics may provide better classification accuracy. Instead of manually trying different combinations of heuristics, we opt to use a supervised learning classifier.

#### **4.4.2. Random forest classifier**

We use a random forest classifier to predict the change output of a transaction. A random forest is an ensemble classifier that trains and aggregates the results of individual decision trees. It is inherently well suited for change detection as it is



**Figure 4.13.:** ROC curves for predicting change in the ground truth data set using the threshold vote and the random forest classifier, compared to individual heuristics. The curve of the threshold vote is based on the entire data set, whereas the curve of the random forest is based on the test set. The random forest model includes additional transaction and output characteristics.

capable of dividing the dataset into homogeneous subsets. This allows it to pick up differences in behavior between different types of transactions or time periods. In an initial comparison of common classifiers it also achieved the highest ROC AUC score (Section B.3.2).

We model an output-based binary classification problem, where every output is either a change (1) or spend (0) output. As before, an individual heuristic may produce one of three outcomes: vote for the output, against the output, or not be able to discern between the outputs. Next, we add additional characteristics about each output and corresponding transaction that may help the classifier differentiate between distinct types of transactions, or wallets. Output characteristics include the ratio of an output's value to the total output value of a transaction (the change output is the smaller output in 76.38 % of our ground truth transactions) and its index. Transaction characteristics include its total value in satoshi, the transaction fee paid per byte, its version number, whether it uses SegWit serialization and sets a non-zero locktime, as well as the number of inputs and the time of inclusion (as epochs of 1008 blocks, about one week).

We use the `RandomForestClassifier` implementation in `scikit-learn` 0.24.1. We add regularization by searching a small parameter grid using a successive halving strategy, including three parameters: the number of features considered at each split, the minimum number of samples required before each further split, and the minimum number of samples required at each leaf node. As we consider an analyst that works with a static snapshot of the blockchain, we randomly split our data set into 80 % training and 20 % test set. We use the training set for the hyperparameter search using 4-fold cross-validation, optimizing the area under the curve (AUC) as our scoring metric.<sup>3</sup> To account for the fact that transactions in the same base

---

<sup>3</sup>Additional details are available in Section B.3.2.

cluster may be highly similar, we explicitly ensure that all outputs of a base cluster remain in the same set and fold.

Applying the random forest model (RF-1) to the test set, we achieve an AUC of 0.9986 (Figure 4.13).<sup>4</sup> We see that random forest model is able to detect a higher share of outputs, especially at low false positive rates, compared to the threshold vote.

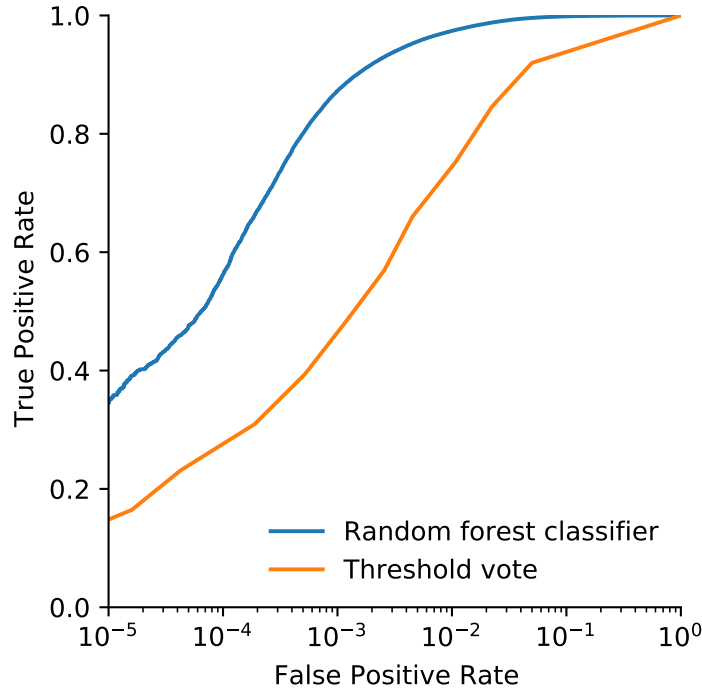
In Figure 4.14 we show the ROC curves of both the threshold vote and the random forest on the same test set, log-transforming the x-axis to highlight the important difference in low false positive rates. The random forest achieves much higher true positive rates at low false positive rates, meaning that it correctly identifies the change output in a larger number of transactions. For example, if we target a false positive rate below 0.1 %, the threshold vote achieves a TPR of around 39 % at a FPR of 0.06 %. For the same FPR, the random forest achieves a TPR of 82 %, more than twice as high.

We train a second random forest model (RF-2) only on transactions that contain predictions of the universal heuristics in order to later predict change in transactions that contain unspent outputs. Using a similar evaluation strategy as for the full model, the AUC of this model is 0.9981.

To ensure that the performance of our model is not dependent on our particular split, and to determine the variance of the ROC AUC score, we again randomly split our ground truth data set into 80 % training and 20 % test set 20 times and train a random forest classifier using the previously determined hyperparameters. The average ROC AUC score on the test sets is 0.9974 (SD = 0.0016) for RF-1, and 0.9965 (SD = 0.0036) for RF-2.

---

<sup>4</sup>The AUC on the training set is 0.9994



**Figure 4.14.:** Comparison of the random forest classifier and the threshold vote on the test set. There is a notable difference between the two classifiers for low false positive rates.

We note one caveat: because the base clustering is incomplete, grouping transactions by their base cluster may not fully prevent homogeneous transactions from the same entity to appear in both sets (e.g., an entity’s transactions can be split among multiple clusters, of which some end up in the training and some in the test set). Yet, some of the variability we see comes from unusual clusters that do not appear in the respective training sets. Other researchers with private, more heterogeneous ground truth may be able to evaluate the degree to which this affects the overall performance of the model.

#### 4.4.3. Additional model validation

We use two data sets to assess the performance of the random forest model trained on our entire ground truth data. First, we use the list of 16 764 transactions identified by

Huang et al. [84] as ransom payments related to the Locky and Cerber ransomware. Those payments were identified through clustering, transaction graph analysis and known characteristics of the ransom amounts. The data set contains not only transaction hashes, but also the index (and amount) of the predicted payment output. Out of the 16 764 transactions in the data set, we exclude 3057 (18.2 %) because they don't match our definition of a standard transaction. For 23 transactions, none of the heuristics return any distinct votes. 1636 transactions (9.8 %) directly reuse change, and in 852 transactions (5.1 %) multi-input clustering already correctly revealed the change output. For the remaining 11 196 transactions (66.8 %) we predict the change output using the random forest model and achieve an AUC of 0.996.

Our second data set is constructed using a GraphSense tagpack [72] that contains 382 tags for addresses of 273 distinct entities (such as exchanges or gambling services) extracted from WalletExplorer.com. We identify each associated cluster and then extract transactions between the clusters, assuming that the output belonging to a different cluster is the spend output. In total, we extract 2 236 064 transactions between the clusters. As the data is highly skewed towards a few clusters, we limit the total number of transactions for each combination of interacting clusters to 1000 (sampled randomly), giving us 859 270 transactions. Out of these, 13 338 (1.6 %) don't have any predictions, 117 421 (13.7 %) reuse an address for change and for 459 737 (53.5 %) change outputs have already been identified through cluster membership. For the remaining 268 774 transactions (31.3 %) we predict the change output and achieve an AUC of 0.976.

## 4.5. Clustering Change Outputs

We use our random forest model to enhance the base clustering by clustering change outputs. To this end, we predict the change outputs of the 310 million standard

transactions with yet unknown change. We exclude 10.5 million transactions where no individual heuristic identified a change output and use RF-2 for 19.3 million transactions with unspent outputs.

Informed by the histogram of probabilities<sup>5</sup> (cf. Figure B.3 in the appendix), and in order to keep the likelihood of false positives low, we use a conservative probability threshold of  $p_{change} = 0.99$ .<sup>6</sup> This gives us 155.56 million change outputs (for 50.24 % of transactions). We then enhance the base clustering by merging the base cluster of the inputs with the base cluster of the change address in the order that the transactions appear on the blockchain.

#### 4.5.1. Naive merging leads to cluster collapse

We first assess the sizes of the enhanced clusters to spot possible cluster collapse. A typical measure for the size of the cluster is the number of addresses contained in it. However, this may not always be reliable: if entities reuse addresses, their clusters can appear small despite being responsible for many transactions. We therefore also inspect the number of transactions originating from the cluster, which is independent of address reuse. Without address reuse, these two measures should correlate as a fresh address is created for every outgoing transaction.

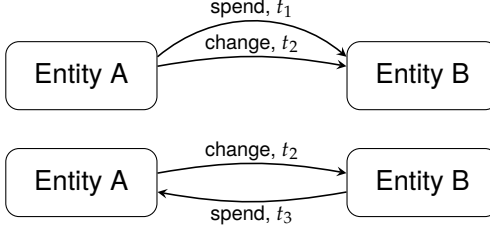
Clustering the identified change outputs reduces 184.3 million affected base clusters into 39.8 million enhanced clusters. However, it leads to severe cluster collapse: there is one large supercluster, of which the prior Mt. Gox supercluster is a part of, that consists of 223.9 million addresses (a 1596 % increase) and 108.2 million transactions (a 2500 % increase). Inspecting the 273 labeled clusters from

---

<sup>5</sup>A probability returned by the random forest is the average of the probabilities returned by the individual decision trees, each of which corresponds to a fraction of samples in the leaf node.

<sup>6</sup>This corresponds to a false positive rate of 0.044 % for RF-1. We use a threshold of 0.997 for RF-2 to match the FPR.





**Figure 4.15.:** Our constrained clustering prevents the merging of clusters A and B due to conflicting types of payments between them.

the Graphsense tag pack, we find that 113 of them have been merged into this supercluster.

#### 4.5.2. Constraints prevent cluster collapse

The majority of cluster merges involve address clusters from which only a single transaction originated. Here, the impact of a single misclassification is low unless a sequence of such merges collapses multiple larger clusters. At the same time, we observe a small number of merges that combine two large clusters. Imagine two large exchanges whose users frequently interact with each other. A single, misidentified change output could collapse their address clusters.

**Approach.** We use this intuition to constrain which clusters we merge. While change outputs predicted by our model should be clustered, we can use predicted spend outputs to prevent cluster merges: the input cluster should not be clustered into the cluster of the spend. Given the probability  $p_i$  returned by the random forest model for output  $i$ , we use two thresholds  $p_{change}$  and  $p_{spend}$  such that if  $p_i > p_{change}$  the clusters should be merged (as before), and if  $p_i < p_{spend}$  then the clusters should not be merged. In many cases, these constraints will prevent the spend and change output of a single transaction to end up in the same cluster (cf.

Figure 4.15). It is therefore a stronger assumption than our core assumption, which only considered the change.

This approach is comparable to that by Ermilov, Panov, and Yanovich [58] to use address tags in combination with a probabilistic model to reduce the number of conflicting tags in the final clustering. However, public sources of address tags likely contain information on a limited number of intermediaries only. Our approach, instead, potentially covers all clusters appearing in the 258 million standard transactions, including those that may be hard to interact with (and identify) manually. Due to the size of our data set we only consider the binary case of preventing any potential conflict, accepting that we may prevent some valid merges in the process.

We implement a constrained union-find algorithm that prevents merging two clusters that are related by a predicted spend output. For every spend from cluster  $c_m$  to cluster  $c_n$ , predicted by the random forest with  $p_i < p_{spend}$ , we add a constraint to cluster  $c_m$  that it must not be merged with cluster  $c_n$ . Before merging two clusters, we check the constraints of both clusters and skip the merge if it would violate them.

**Results.** Using the same  $p_{change}$  as before and setting  $p_{spend} = 0.01$ , the constrained clustering prevents 413 608 merges that would have violated constraints and retains 231 340 more individual clusters than the unconstrained clustering. 10.8 million of the predicted merges are redundant because the clusters had already been merged.

We find that the constrained merging prevents the previously observed severe cluster collapse. For example, the constrained clustering does not produce the large Mt. Gox supercluster: the enhanced cluster contains 4.4 million transactions (a 6 % increase) and 14.5 million addresses (a 10 % increase). Assessing the 273 labeled clusters, there are only seven instances left where two labeled clusters are merged

**Table 4.4.:** Transaction counts of smaller clusters being merged.

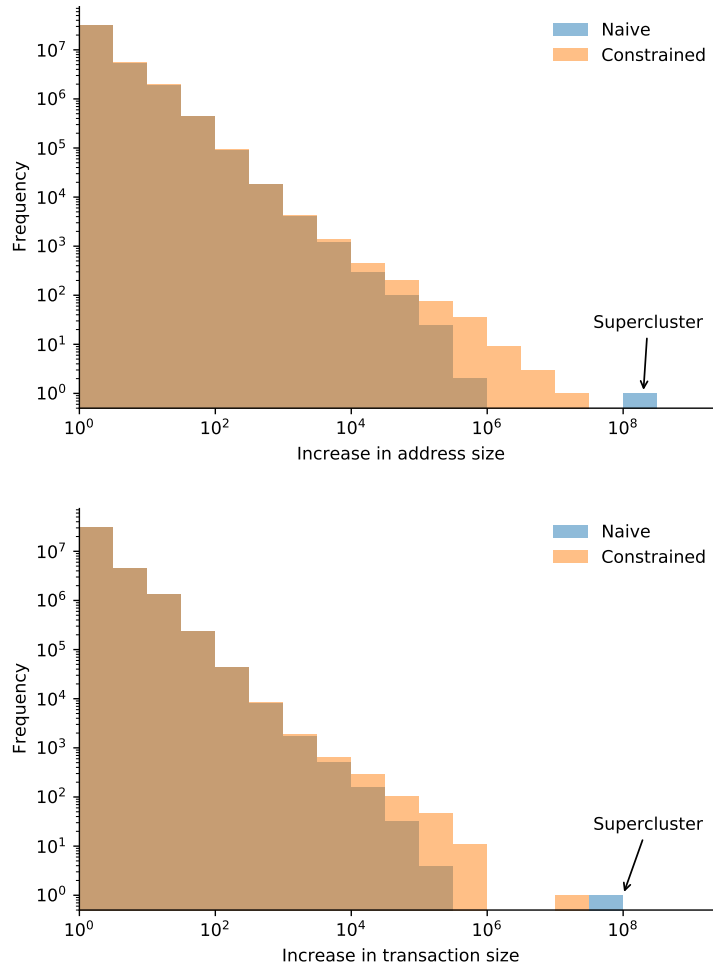
Percentile	Transaction count	
	Naive	Constrained
90	1	1
99	6	6
99.9	25	25
99.99	114	102
99.999	1696	578

together. We suspect that unusual types of payouts from these services might have triggered their collapse.

The largest cluster in the constrained clustering contains 20.4 million transactions and 40.5 million addresses. Inspecting the composition of this cluster, we find that it is the result of merging many small clusters (including 9.4 million single-transaction clusters).

Table 4.4 shows percentiles of the number of transactions associated with each smaller cluster that was merged into a larger cluster. In at least 90 % of merges the smaller cluster created at most one outgoing transaction, highlight the usefulness of change address clustering to merge small clusters that are missed by multi-input clustering (see Figure 4.2). We can also see that up to the 99.99 percentile, transaction counts are very similar between the unconstrained and the constrained clustering, but that the constraints prevent merges of large clusters, preventing cluster collapse.

Figure 4.16 shows histograms of the increase in address count and transaction count for the affected cluster. Increase here is the difference between the resulting size of the enhanced cluster and the largest individual base cluster before clustering. The histograms are plotted on a log-log axis, highlighting the extreme skewness of



**Figure 4.16.:** Absolute increase in address and transaction count per affected cluster, using either the naive or the constrained approach. Note the caveat in footnote 7.

the distribution. Most clusters increase only by a few addresses or transactions.<sup>7</sup> The supercluster can also easily be spotted.

**Varying thresholds.** For our clustering we chose conservative thresholds in order to reduce the possibility of cluster collapse. At the same time, this means that fewer change outputs are being clustered than with larger thresholds. To assess the impact

<sup>7</sup>Due to the log-scale, the histogram showing the increase in transaction size excludes 2.3 million clusters that, using the naive approach, didn't increase in their transaction count, and 2.5 million using the constrained approach, respectively.

of raising the threshold, we create two additional constrained clusterings, one with a threshold corresponding to 0.1 % FPR and one corresponding to a 1 % FPR. At 0.1 %, the number of collapsed clusters in the Graphsense tag pack increases to 12. At 1 %, however, there are 60 instances of cluster collapse, highlighting the importance of choosing conservative thresholds corresponding low false positive rates.

## **4.6. Impact on Blockchain Analyses**

Address clustering is a common preprocessing step before analyzing activity of entities on the blockchain. Using different change heuristics (or none at all) thus affects the outcome of these analyses. In this section we present two exemplary applications to highlight the impact change address clustering has on blockchain analyses. Furthermore, we compare our clustering to one created with a popular change address heuristic based on address reuse.

### **4.6.1. Increased cashout flows from darknet markets to exchanges**

We evaluate the impact of our enhanced clustering on analysing payment flows from darknet markets to exchanges. Such analyses are potentially relevant for cybercrime researchers, economists, regulators or law enforcement, highlighting the importance of address clustering for a variety of use cases. To identify relevant intermediaries, we use address tags in the GraphSense tag pack for 117 exchanges and 15 darknet markets.

We extract the value of all outputs in transactions initiated by a darknet market (most of which were active between 2013 and 2016) that are sending bitcoins to an exchange, comparing the transaction volume calculated using our base clustering to that of our enhanced clustering. The median increase in output value across all 15

darknet markets amounts to 11.5 %. Overall, the total amount of bitcoins flowing from the darknet markets to exchanges increases from BTC 823 839 to BTC 937 330 (a 13.8 % increase). An overview of each individual market’s increase in transaction volumes is presented in Table B.1 in the appendix.

#### **4.6.2. Improved estimate of velocity**

We replicate the analysis of velocity conducted by Kalodner et al. [87], an example for a longitudinal analysis of economic activity occurring on the Bitcoin blockchain. For this analysis, clustering is used to remove self-payments of users (such as change outputs), which would artificially inflate estimates of economic activity. The better and more complete our clustering, the more self-payments are removed and hence the lower the estimate will be.

Our refined clustering reduces their estimate of bitcoins moved per day between January 2017 to June 2021 by about 11.9 % (a longitudinal plot is provided in Figure B.4 in the appendix). This number is quite similar to the impact on darknet market cash-out flows.

In general, much activity is generated by large intermediaries, which are more prone to be clustered by the multi-input heuristic. Our enhanced clustering predominantly merged small clusters, which should give a more realistic estimate for their activity. As we chose relatively conservative thresholds to enhance the clustering, loosening them could lead to larger effect sizes.

#### **4.6.3. Comparison to the Meiklejohn et al. heuristic**

We compare our constrained clustering to one created using the address reuse-based heuristic presented by Meiklejohn et al. [103], which has subsequently been used in other studies (e.g., [41, 127]). While the authors highlight the need for manual

intervention to prevent cluster collapse, this is likely infeasible for analysts without in-depth domain knowledge or the right set of tools. There are two possible ways to implement the heuristic, either such that an output is considered to be change if its address has not appeared in any previous transactions (we call this variant local), or such that over the entire blockchain it appears in only one output (global). The latter thus depends on the particular state of the blockchain.

Applying the local heuristic to the standard transactions with unknown change produces a large supercluster, including 124.8 million transactions and 281.3 million addresses. 166 of the tagged clusters appear in this supercluster. Similarly, the global variant produces a supercluster containing 133.1 million transactions and 298.4 million addresses, with 177 tagged clusters ending up in the supercluster.

To characterize the difference between the clusterings, we look at the probability that two random addresses are clustered in a particular clustering. The probability of two addresses being clustered together increases by a factor of 40 when using the Meiklejohn heuristic compared to our constrained clustering, further highlighting the cluster collapse (cf. Table B.2 in the appendix).

Finally, we take closer look at the individual predictions. First, comparing the predictions of the local Meiklejohn heuristic to those of our constrained clustering, they overlap on 71.2 million transactions and differ for 1.3 million of those. The global heuristic differs on 1.9 million transactions out of an overlapping 81.1 million. For those conflicting predictions, summing up the pairwise differences of output values in each transaction (an upper bound on the economic activity / payment volume that might be misattributed when clustered) amounts to BTC 2.1 million, or USD 16.6 billion (computed with the corresponding daily exchange rate for each transaction) of economic activity for the local heuristic. For the global heuristic, the difference amounts to BTC 4.1 million, or USD 38.7 billion.

## 4.7. Discussion

Our results confirm existing expectations (e.g., [134]) that in many instances change address detection is feasible with high precision. This may motivate users to adopt privacy-enhancing countermeasures. Techniques to avoid address reuse (such as one-time addresses [157]) would reduce the effectiveness of the multi-input heuristic and thus also of our ground truth extraction technique. Widespread use of cooperative obfuscation techniques like CoinJoin [99] and PayJoin [51] would make multi-input clustering unreliable (by causing cluster collapse), and ambiguity-based obfuscation such as the randomization of address types (e.g., [21]) could thwart the fingerprinting heuristics. If users adopted these techniques more widely (cf. [110]), it would also make law enforcement investigations into cryptocurrencies more challenging, increasing the need for complimentary approaches that do not rely on address clustering (cf. Chapter 5).

The transparency blockchains provide, and our work respectively, are highly useful beyond law enforcement purposes. Researchers studying social or economic questions, using cryptocurrencies as a “social science lab” [24], benefit from increased transparency. Blockchains contain a valuable trove of financial data that traditionally hasn’t been available to researchers [22]. In this context, our techniques can help to remove ambiguity about ownership of funds, improving data quality and leading to more realistic estimates.

## 4.8. Summary

Address clustering is an important cornerstone of many blockchain analyses. In this chapter, we’ve taken a first step towards building better models that allow analysts to identify change outputs in transactions, enabled by a new ground truth data set



extracted from the Bitcoin blockchain. Evaluating this data set, we find that for many transactions identifying the change address is feasible with high precision. Importantly, the ability to fingerprint transactions based on Bitcoin protocol features in order to determine change outputs has become highly effective over the past years. This dependence of users' privacy on how others create transactions yet again demonstrates how privacy in cryptocurrencies is inherently interdependent.

Crucially, our work is the first to apply machine learning to the problem of change identification. We find that our random forest model outperforms a baseline voting mechanism, detecting twice as many change outputs when targeting low false positive rates. This model can be applied to other standard transactions on the blockchain, highlighting how the privacy-sensitive disclosures of some users can affect the privacy that Bitcoin provides at large. Turning to the subsequent clustering of change addresses, we've demonstrated that constraints based on our model's predictions can help prevent cluster collapse. Finally, we've explored the impact of our clustering on the outcome of economic analyses. We hope that this work will encourage and enable further research into the privacy implications of change address detection as well as techniques for address clustering.

# Effective Cryptocurrency Regulation Through Blacklisting

## 5.1. Introduction

“Criminals are early-adopters of new technology” — this colloquial statement about the tug of war between criminals and law enforcement readily applies to cryptocurrencies like Bitcoin. Being open financial transaction systems, they allow pseudonymous participation, facilitate global payments, and don’t rely on a central entity. This unique value proposition, however, comes at a price. Since their inception more than ten years ago they have been associated with money laundering [61, 112, 141, 151], the sale of narcotics and illegal goods [37, 146], extortion and ransomware [84, 125, 126, 151], investment fraud and scams [76, 158, 159] or human trafficking [133].

Fighting the criminal use of cryptocurrencies meanwhile faces unique challenges. In most financial systems, know-your-customer (KYC) procedures tie accounts to account holders’ identities, enabling identity-based blacklists and enforcement. In cryptocurrencies, however, identity-based regulation is inherently ineffective (cf. [113]). Bitcoin addresses can be created anonymously, in unlimited quantities, and without central oversight. Criminals can thus easily evade static identity-based measures by generating new addresses to receive funds or launder their coins.

As part of their anti-money laundering (AML) compliance programs, regulated entities such as cryptocurrency exchanges have started using transaction screening services to assess incoming funds (e.g., [35, 120]). As normal users don't have access to these services, their widespread use creates information asymmetries that leave users at risk of receiving funds they cannot spend at exchanges. Detection of illicit activity by intermediaries also does not provide a comprehensive deterrent to combat money laundering that takes place outside of regulated intermediaries.

Instead of relying on identity-based measures and proprietary transaction screening, regulators can make use of the inherent transparency and traceability of the blockchain. As transfers in Bitcoin reference the origin of the funds they are spending, it is possible to follow money derived from illicit activity from one transaction to the next. This allows to blacklist individual transaction outputs (think coins) and recursively enforce such blacklisting [113]. By requiring intermediaries in the Bitcoin ecosystem to check the origin of coins against the blacklist before accepting them, all users are incentivized to adhere to the blacklist, addressing money laundering and other criminal activity more effectively.

Transaction blacklisting in Bitcoin has been discussed both from a technical [2, 5, 6, 25, 75, 113] and legal [5, 25, 75, 81, 138] perspective. However, it hasn't received much attention in policy discussions about regulating cryptocurrencies yet. Regulators so far have followed an optimistic approach of waiting for the ecosystem to mature, and have largely abstained from regulation that could potentially hinder innovation. Cryptocurrency advocacy groups have also supported such a path, often highlighting how Bitcoin's transparency already enables retroactive law enforcement investigations [119].

However, supplementing existing regulation with transaction blacklisting could increase the effectiveness of anti-money laundering in cryptocurrencies today, and

will become even more important as cryptocurrencies get more widely used outside (i.e. without the involvement) of regulated intermediaries. Blacklisting could at some point also start to occur organically through individual regulators' actions or court decisions (e.g., [81]), but a holistic regulatory approach, including common principles and technical standards, is desirable to make it effective in practice.

Like all anti-money laundering regulation, blacklisting has to balance the sometimes competing interests of transaction efficiency, users' privacy, and disincentivizing the criminal use of cryptocurrencies. Compared to an unregulated cryptocurrency, it may raise transaction costs and reduce the fungibility or time-value of coins. At the same time, reducing criminal activity and strengthening the ownership rights of coin holders could benefit the overall adoption of cryptocurrencies. For blacklisting to be a viable option, its impact needs to be well understood and the cryptocurrency remain practical to use.

To this end, we explore the intricacies of blacklisting and how it would impact the cryptocurrency ecosystem. First, we show how blacklisting fills current gaps in AML regulation of cryptocurrencies and protects users from receiving funds derived from illicit activity (Section 5.2). Next, we discuss the practical aspects of setting up and using transaction blacklisting in a cryptocurrency like Bitcoin (Section 5.3). Once blacklisting is in place, users can employ new mitigation strategies against the risk of accepting funds which could get blacklisted in the future (Section 5.4). Next, we systematize important properties of blacklisting policies and review five policies that have been proposed in the literature (Section 5.5). Acknowledging that blacklisting may not be feasible in all cryptocurrencies, we describe the degree to which blacklisting can remain effective in the presence of more privacy-preserving cryptocurrencies (Section 5.6). Finally, we discuss common concerns around

blacklisting, such as its impact on fungibility and privacy (Section 5.7), and conclude with a brief outlook (Section 5.8).

## **5.2. Combating Money Laundering in Bitcoin through Blacklisting**

In this section, we first review the current state of AML regulation as it applies to regulated Bitcoin intermediaries such as exchanges. Then, we discuss the shortcomings of the current regulatory landscape, specifically how the focus on accounts and regulated intermediaries misses money laundering happening outside of exchanges and how it puts users at risk of receiving money they cannot spend. Finally, we sketch a more effective solution: how making information about money laundering public would make combating money laundering more effective and better protects users.

### **5.2.1. Background: AML regulation in the US**

Anti-money laundering (AML) regulation aims to prevent and disincentivize the (re-)introduction of money derived from illegal activity into the legal economy. In the United States, money laundering is prohibited through multiple, complementary laws, the most important ones being the Bank Secrecy Act (BSA) of 1970, the Money Laundering Control Act of 1986, as well as Title III of the USA PATRIOT Act of 2001 (also known as Title III: International Money Laundering Abatement and Financial Anti-Terrorism Act of 2001).

The Bank Secrecy Act of 1970 requires financial institutions to fulfill record keeping obligations and to report large or suspicious transfers. The original intention of the BSA was to make the placement of cash proceeds of drug sales harder by creating

a paper trail that investigators could follow [104]. Yet, it did not make money laundering itself illegal: money laundering was only considered a by-product of the actual crime from which the proceeds were derived, which would be prosecuted. The BSA also did not make (attempted) evasion of the reporting requirements illegal. Criminals were able to structure transactions in a way that would avoid reporting requirements, e.g., by making cash deposits slightly below the reporting threshold.

The Money Laundering Control Act of 1986 made money laundering a federal crime and applies to any US person (not just financial institutions). It also fixed loopholes of the BSA by explicitly making evasion of reporting requirements a crime. 18 U.S. Code § 1956 prohibits the transfer of funds derived from “specified unlawful activity” with the intent of promoting specified unlawful activity. In addition, 18 U.S. Code § 1957 prohibits monetary transactions above \$10,000 derived from specified unlawful activity and conducted by, through or to a financial intermediary, without requiring intent of promoting unlawful activity [29]. Both sections feature a knowledge requirement: a person must know that the money is derived from unlawful activity in order to be guilty of money laundering – just a suspicion that the money might be derived from unlawful activity is not sufficient.

The International Money Laundering Abatement and Financial Anti-Terrorism Act of 2001 (Title III of the USA PATRIOT Act) further strengthened these money laundering regulations. It requires companies to have an AML compliance program, including at a minimum the development of internal AML policies, procedures and controls, the designation of a compliance officer, employee training as well as independent audits of the program (31 U.S. Code §5318 (h)). The act also classified Money Services Businesses (MSB), which are businesses that transmit or convert money, as financial institutions, making it easier to apply existing financial regulations to them (Section 359). The Patriot Act furthermore widened the scope

of anti-money laundering regulation by including the support of foreign terrorist organizations as a money laundering offense.

### **5.2.2. Existing AML regulation's focus on accounts is ineffective in cryptocurrencies**

Banks constitute centralized entry points to the traditional financial sector, making them a convenient target to enforce AML regulation. Cryptocurrencies like Bitcoin however operate fundamentally different from traditional financial systems. Account creation is independent of any financial institution and without the ability to enforce identity verification requirements. Instead of maintaining accounts as entries in a centralized database (allowing to enforce KYC), cryptocurrency “accounts” are represented by cryptographic key pairs. The public key of such a pair is used as the account identifier (i.e. the “address”), and digital signatures created with the private key authorize payments. As anyone can create new key pairs and use them to receive and send coins, the system works completely without a central authority. Because creating keys is fast and cheap, one-time use of addresses is encouraged: every time a user receives money they can create a new address, increasing privacy as their activity is split among many different addresses. Identifying addresses belonging to the same user is possible only heuristically and likely to be incomplete.

With no oversight over account creation, cryptocurrencies lack a natural leverage point to enforce KYC or record keeping requirements. Identities can only be acquired and verified when users cross the boundary to the traditional financial system by interacting with regulated intermediaries, such as exchanges or payment providers. But even if identities can be acquired at these intersections, regulators won't gain a complete picture of an individual's transactions as these could be facilitated using

many different addresses (cf. Chapter 4), only a small subset of which become known to the intermediary. As a result, much activity in the system is not attributable.

This inherent limitation has started to draw regulators' attention. In 2019, the Swiss regulator FINMA issued guidance requiring financial institutions to verify that the receiving wallet of an outgoing transfers belongs to the customer [64]. In December 2020, the Financial Crimes Enforcement Network (FinCEN) proposed rule making that would require financial intermediaries to record and verify additional information about counterparties using "unhosted" wallets [63]. And recent guidance by the Financial Action Task Force (FATF) suggested to include transfers from virtual asset service providers (VASP) to non-VASPs under the travel rule, which requires financial intermediaries to collect identifying information about the recipient [154]. Yet, none of these approaches are effective as they do not affect any transactions after the initial withdrawal to the unhosted wallet, and at the same time require intermediaries to collect large amounts of personal data.

Right now, many cryptocurrency transactions are likely driven by speculation [132]. But in the future, cryptocurrencies could enable large volumes of decentralized commerce that takes place between individuals only, facilitated by decentralized protocols that don't require the involvement of a regulated intermediary. Cryptocurrencies also do not distinguish between national and international payments. National regulation efforts that depend on sensitive private information about users' identities and their spending habits are ineffective in such an environment.

### **5.2.3. The effectiveness of regulating exchanges is limited and users are at risk of accepting money they cannot spend**

Cryptocurrency exchanges currently constitute the major interface between the traditional financial sector and the Bitcoin ecosystem. Exchanges allow users to



deposit bitcoins or fiat currency and convert them from one to the other. They are usually custodial, as they hold users' funds in their own wallets (in the case of cryptocurrency) or bank accounts (for fiat currency). In the United States, custodial exchanges are considered to be MSBs and therefore must comply with the previously discussed AML regulations [62].

To fulfill the abstract requirements of these regulations, including the development of an effective AML compliance program that has "reasonable" measures to prevent money laundering, many exchanges have adopted the use of blockchain intelligence services to screen customers' transactions.

These services are designed to detect transactions that originate from illicit sources, such as dark web markets or sanctioned individuals. Exchanges use them to screen customers' incoming funds before depositing them into the respective accounts. If suspicious transactions are detected, accounts may be frozen and customers asked for additional information about the origin or purpose of funds, or customers might be prevented from further using the service altogether.

As (to the best of our knowledge) no detailed technical information about how these services work exactly is publicly available, the following description is based on blog posts (e.g., [85, 145]) and informal discussions. To detect suspicious transactions, these services rely on large databases mapping Bitcoin addresses to known identities. Their ground truth comes from interaction with platforms and marketplaces in the Bitcoin ecosystem, from the use of their products by intermediaries, as well as from law enforcement and other sources. The identified entities are then grouped into risk classes. To derive a risk score for a specific transaction, it is checked for originating from or being destined for a known entity.

These current practices raise three general concerns. First, the existence and use of such databases is a potential privacy concern for Bitcoin users. While exchanges

do not send customers' identities to the screening services [85, 145], the collected data may still allow to re-identify users when combined with address clustering and external datasets, or specific knowledge of an individual's activity.

Second, transaction screening at intermediaries is not sufficient to combat money laundering in Bitcoin. While currently most exchanges rely on such services to identify money derived from illicit activity in order to comply with the (rather unspecific) requirements of AML regulation, their use does not prevent money laundering occurring outside of regulated intermediaries. And companies in countries with less strict AML regulation may have little incentive to implement such measures in the first place.

Third, individual users who want to make payments without the involvement of an intermediary don't have easy access to transaction screening services to perform their own due diligence before accepting coins. This creates information asymmetries about the quality of coins. As a result, whenever they accept coins from anywhere but regulated entities they are at risk of receiving illicit funds that, unknowingly to them, they won't be able to spend.

#### **5.2.4. Public blacklists of funds derived from illicit activity make AML more effective and protect innocent users**

We identified three areas in which the current regulatory landscape can be improved: reducing the dependence on knowledge of account ownership, improving the effectiveness of AML outside of regulated intermediaries, as well as providing more transparency to protect innocent users. A regulatory approach based on public blacklists for illicit coins [25, 75, 113] would address all three of these issues.

Blacklists are a well-known tool in a regulator's toolbox to prevent interaction of regulated parties with certain outside entities. For example, the Office of Foreign

Asset Control (OFAC), a US regulator responsible for enforcing trade and economic sanctions against foreign countries, maintains a list of identities of foreign nationals that US entities are forbidden from interacting with, in order to prevent those listed from participating in the global economy. And in many areas of computer security, such as spam prevention or defending against denial of service attacks, defense mechanisms rely on blacklists. In the context of Bitcoin, a blacklist would contain specific coins that are known to be derived from illicit activity. By tracing these coins from one transaction to the next, they can be separated from the legal economy [25, 113].

Blacklisting coins works on the basis of transactions and is recursively applied when an illicit coin is spent in a new transaction. The advantage of a transaction-based approach is that it is not necessary to know any identities behind addresses. Anyone can check whether coins of a particular transaction are illicit by checking the blacklist. Whenever illicit coins are spent, coins in the new transaction inherit the illicit status, ensuring that it is impossible to launder coins by moving them through a chain of transactions to an unknown address (in fact, the exact address a coin is associated with becomes irrelevant).

Public blacklists are effective at fighting money laundering outside of regulated intermediaries. Since anyone can check the coins they are about to receive, those holding illicit coins won't be able to spend them. While only regulated intermediaries would be legally required to reject or confiscate illicit coins, *every user* has an incentive to check the blacklist, as they wouldn't be able to spend listed coins themselves. At the same time, once it is publicly known which coins are derived from illicit activity, the ability to check coins before accepting them reduces the risk of receiving coins that the user cannot spend at an exchange.

Public blacklists enable public scrutiny: while there's currently a lack of transparency about exchanges' decisions to accept coins, any incorrect or malevolent blacklisting of coins (e.g., for political censorship) would be detectable, and could then be openly questioned (or challenged in court). Having a standardized process to list funds will furthermore improve regulators', law enforcements' and courts' effectiveness at combating crime in cryptocurrencies. And if blacklists effectively deter crime, they can even reduce current reliance on potentially privacy-invasive practices of mapping out the Bitcoin ecosystem.

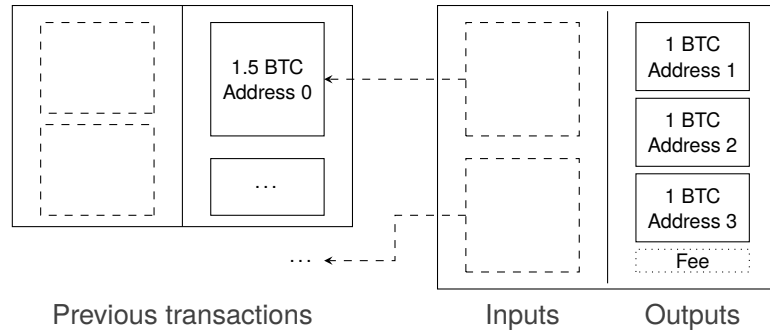
Of course, implementing public blacklists for cryptocurrencies is not a panacea. It requires additional infrastructure and makes sending and receiving cryptocurrency more involved as users will want to check the blacklist before accepting coins. And to make blacklists effective for a global payment system, some degree of international coordination is necessary to prevent criminals from routing their funds through countries that don't enforce the same blacklists. Regulators and law enforcement might be reserved towards a regulatory solution that affects normal users and requires making information about illicit activity public. And the cryptocurrency community itself might object such an approach, since it introduces centralized control that cryptocurrencies were envisioned to evade in the first place. In the rest of this chapter, we describe how such a system can work in practice and address these concerns in more detail.

### **5.3. How Blacklisting Would Work**

Enforcing transaction blacklists impacts the way payments are conducted in a cryptocurrency. In this section we recall how recursive transaction blacklisting works, discuss the role regulators would need to take on and how blacklisting affects

**Table 5.1.:** How blacklisting would change the Bitcoin ecosystem

Stakeholder	Aspect	Change	Description	Section
Ecosystem	Decentralization	Decrease	Centralized entities may be better able to offer more convenient solutions for users to handle blacklisting compared to decentralized alternatives	5.3, 5.4
	Privacy	Mixed	Centralization could reduce privacy, less reliance on privacy-invasive address tagging can increase privacy, payment networks and privacy overlays can increase privacy	5.6
All users	Taint check	Major change	Users are incentivized to check coins against blacklists before accepting payments	5.3.1, 5.3.6
	Risk of future blacklisting	Major change	Exposed by public information, now requires use of mitigation strategies	5.4
Money Services Businesses	Existing AML requirements (KYC, reporting, etc.)	No change	Potentially reduced reliance on privacy-invasive identity-based measures if blacklists are effective	5.2.1
	Taint check	Major change	Required to check incoming funds against blacklists and freeze/reject tainted funds	5.3.5
	Risk of future blacklisting	Objective change	Risk assessment to protect against future blacklisting of high-risk transactions	5.4
	Additional services	New	Exchanges can provide insurance against future blacklisting	5.4.4
Wallets	Taint check	Voluntarily	Integration of automated blacklist checks	5.3.6
	Risk of future blacklisting	Voluntarily	Integration of mitigation strategies	5.4
	Coin selection	Minor change	Depends on taint policy and objectives	5.3.6
Payment Channel Networks	Channel establishment	Minor change	Taint check + risk assessment as for normal payments, preference for trusted counterparties	5.4.3
	Risk of future blacklisting	Minor change	Only immediate channel hop poses risk	5.4.3
Decentralized protocols	Taint check, risk of future blacklisting	Voluntarily	May incorporate signaling or coin negotiation into protocol	5.6.2
Regulators / Law Enforcement	Blacklists	Major change	Issue cryptocurrency-specific blacklists, define taint propagation policies and specify actions cryptocurrency intermediaries and users need to implement, define standardized API for clients to query blacklists	5.3.3
	International cooperation	Major change	International cooperation and coordination required (e.g., through FATF or Interpol)	5.3.3
	Reporting crimes	Major change	Users can submit evidence of crimes to law enforcement to blacklist coins	5.3.4
	Contesting listings	Major change	Legal process to object unreasonable or incorrect blacklisting of coins required	5.3.4
Courts	Forfeiture	Minor change	Able to blacklist known funds that cannot be seized by LE	[81]



**Figure 5.1.:** The structure of the Bitcoin transaction graph allows to follow coins from one transaction to the next

exchanges, merchants and normal users. We provide a summary of the anticipated changes in Table 5.1.

### 5.3.1. Recursive blacklisting

The goal of AML regulation is to prevent money that was acquired through illegal activity from entering the legal economy. A blacklist-based approach can achieve such separation, requiring users to check coins before accepting them and refusing those with illicit origin. In the context of Bitcoin, a blacklist would include specific outputs of transactions that are associated with illegal activity, such as theft, extortion, money laundering or trade of illegal goods [25]. When those outputs are spent, their illicit status (we refer to it as taint) is inherited by the transaction spending it. That means that even when coins originating from an address known to be involved in illicit activity are moved through other addresses multiple times, they retain their taint and can be identified as illicit. By recursively assigning taint to all following transactions laundering illicit coins becomes impossible.

Figure 5.1 shows the structure of a typical Bitcoin transaction. Transactions contain outputs that associate an amount of bitcoins with an address and spend value from previous outputs that are referenced by inputs. By mapping taint from

the inputs to the outputs (discussed in detail in Section 5.5), taint of a former transaction is retained and applied to new outputs [113].

Blacklisting outputs effectively freezes the value contained within the output when regulated entities are forbidden to accept these coins. When coins are transferred to a different address, the link between inputs and outputs allows to follow the coins and recursively apply the blacklisting to the new transactions. Intermediaries would not be allowed to accept coins that stem from these blacklisted outputs, or would need to ignore or seize the part of the funds that are tainted.

Because taint is only propagated when outputs are being *spent*, it is not possible to taint other users' funds simply by sending tainted coins to their addresses. Only when those tainted coins are combined in a transaction with other inputs, their taint can propagate to the user's coins.

**A note on terminology.** Fox [65] discusses the difference between the terms “following” and “tracing” in English law (i.e. common law in England and Wales). “Following” implies that the same asset is being followed, whereas “tracing” means identifying a new substitute for the original asset. In many cryptocurrencies (e.g., Bitcoin), a transaction effectively destroys the value present in its inputs and recreates the value in its outputs. Just as if someone were to melt gold coins to create new coins of a different weight, there is no clear mapping from the value in the inputs to the value in the outputs. Hence, it cannot be followed, only traced.

In the context of cryptocurrencies the term “tracing” is used with a different meaning: the deanonymization of patterns or techniques designed and employed to obfuscate payment flows. Privacy-focused cryptocurrencies like Monero use the term “untraceable” to describe a specific form of unlinkability: between the input in a transaction and the output that is being spent by this input [117]. In a way, there can be two types of tracing: identifying the output (of a previous transaction) that

is spent by an input (taking a backwards look), and identifying which output (in the same transaction) receives value from an input (taking a forward look).

To make the distinction between these two types of tracing clear, we will use the following terminology in this chapter. First, we are primarily interested in the flow of illicit value. We call such value “tainted”. A “taint policy” defines how taint is “traced” from inputs to outputs (within one transaction), but we use the term “mapping” instead. When such mappings are applied recursively, we say that taint is “propagated” through the transaction graph. Second, we use the terms “traceable” and “untraceable” as used in the cryptocurrency privacy literature, i.e. referring to unlinkability between outputs and the inputs in which they are spent.

### 5.3.2. Legal grounds

When property is stolen and resold to an unsuspecting (*bona fide*) purchaser, even multiple times, the original owner generally retains their title in the property. As a result, they can reclaim the property from the *bona fide* purchaser (in English law this is known as the “*nemo dat rule*” [6]). Blacklisting facilitates such recovery of stolen property in cryptocurrencies.

However, exceptions to this principle exist: the *bona fide acquisition rule* excludes certain types of property, such as cash and other legal tender, from falling under this general principle in order to balance competing interests between unrestricted economic exchange and ownership rights [12]: frequent exchange of cash is critical for efficient economic exchange and tracing the flow of cash from one person to the next is impractical.

Whether the rule applies to cryptocurrencies has not yet been established, and in part depends on their legal status (i.e., whether they should be treated like money). Balthazor [12] argues that “if the cost of tracing a transaction evolves to have little



or no impact on the free flow of money, then this erodes the economic justification for the bona fide acquisition rule.” Blacklisting could provide just such a means for efficient tracing of stolen cryptocurrency. Furthermore, cryptocurrencies lack options and remedies similar to those of traditional enforcement. Their decentralized, cross-national nature complicates law enforcement investigations, the identification of perpetrators as well as the seizure or retrieval of funds.

For the purpose of this chapter we assume that the law permits recursive blacklisting of coins. We refer the interested reader to in-depth discussions of these issues by Balthazor [12] for U.S. law, by Fox [65] for English law, and by Grzywotz [75] for German law.

### **5.3.3. Blacklist governance**

A blacklist would be published by a regulatory agency that is tasked with the prevention of financial crime. The regulator would be responsible for adding entries to the blacklist as well as defining the terms of how it should be followed. For example, they would specify how taint propagates from one transaction to the next (cf. Section 5.5) and how regulated intermediaries should treat tainted coins.

While the abstraction of a single blacklist is useful to reason about blacklisting, in practice there will likely be multiple blacklists issued by different regulators. In the US, financial regulation is split between a variety of different agencies. State regulators, such as the New York Department of Financial Services, exist alongside federal regulators such as FinCEN. Different regulatory and law enforcement agencies have oversight over different areas or jurisdictions, for which they would issue individual blacklists, with potentially different rule sets. For example, FinCEN could maintain a general list of bitcoins stemming from money laundering and

terrorist financing, while OFAC maintains a list related to economic sanctions, and the DEA might list coins specifically related to the illegal sale of drugs and narcotics.

Similarly, other countries would maintain their own blacklists, and entities that conduct business in different countries would need to adhere to their respective blacklists. To make blacklisting effective and prevent criminals from cashing out money that was stolen in the US on a European exchange, international coordination between the different countries is necessary. As with traditional financial regulation, and cybercrime more generally, blacklisting inherits the challenges typical in international cooperation. Countries need to synchronize their blacklisting efforts, such that cross-national transactions are not inhibited by a highly fragmented landscape of blacklists and policies.<sup>1</sup> Information sharing can be carried out by existing structures for international cooperation, such as the Egmont Group of Financial Intelligence Units, the Financial Action Task Force on Money Laundering (FATF) or the International Criminal Police Organization (Interpol). In order to limit the complexity for end-users, blacklists should have a standardized API and machine-readable rules such that client software can automatically parse them and adhere to their requirements.

#### **5.3.4. Adding an entry to the blacklist**

Entries are added to a blacklist as a result of a criminal investigation, for example when investigators were able to locate a criminal's bitcoins but are not able to seize it (cf. [81]), or as a direct reaction to a crime, such as the payment of a ransom. We note that regulators and law enforcement may in specific instances choose not to list assets if they are part of an ongoing investigation, e.g., if they hope to identify a

---

<sup>1</sup>If different jurisdictions select different taint policies, outputs could have different legal status across countries.

specific criminal when they attempt to cash out their proceeds on an exchange.<sup>2</sup> In such a scenario, listing the assets could compromise the integrity of the investigation.

A user whose coins have been stolen, or who was blackmailed and paid a ransom, would need to submit supporting evidence to the operator of the blacklist (e.g., through a law enforcement entity such as the FBI's Internet Crime Complaint Center). To prove ownership of the coins at the time of the incident, the user would create a digital signature with the keys that held the coins. In case of a theft on an exchange (e.g., due to phishing) where the user does not hold the keys themselves, the exchange could submit such a proof on behalf of the user. In addition, the user would submit supporting evidence that their coins were indeed stolen or extorted and a statement of good faith under penalty of perjury. Making incorrect claims about stolen coins would hence be a crime that can be challenged through traditional enforcement channels.

Due to the decentralized nature of Bitcoin and a lack of a central registry, regulators are unable to inform the holder of coins of a listing of their funds. While custodians could be required to inform users when their coins have been listed, users of non-custodial wallets need to regularly check their coins against the blacklist (which could be done automatically by their wallet software).

As the listing of funds effectively corresponds to the freezing of assets, there must be a legal process in place to contest unreasonable listings as well as a way for incorrectly affected persons<sup>3</sup> to complain and take legal action against it [75]. Such a process could be modeled based on existing processes, e.g., OFAC's license application to release blocked funds [124]. Affected users would submit an application to the regulator to remove the listing of coins from the blacklist,

---

<sup>2</sup>Guidance by OFAC strongly encourages companies affected by ransomware to report their incidents to law enforcement in order to facilitate such subsequent investigations [152].

<sup>3</sup>Note that objecting to a listing would only be feasible for the immediate holder of the illicit coins.

along with details about the transaction as supporting evidence that the coins were acquired legally (e.g., bought from a reputable exchange). Depending on the type of crime and evidence presented, some of these steps could be automated.

### **5.3.5. Regulated intermediaries**

Blacklisting is only effective if users and businesses take it into account when accepting coins. However, since Bitcoin is an open system, compelling users, developers or miners to enforce a blacklist is hard due to their geographic dispersion and limited legal options to do so. Intermediaries in the Bitcoin ecosystem, such as exchanges or payment providers, however, do have a clear jurisdiction in which they already follow existing financial regulation. They are responsible for a significant amount of the transaction volume on the Bitcoin blockchain (cf. [91]) and provide onboarding for most users acquiring bitcoins. They could be required to follow a transaction blacklist by amending existing anti-money laundering regulations. If exchanges and payment providers follow the rules in a blacklisting regime, any user who at some point in the future wants to interact with them, either to convert fiat currency into cryptocurrency or back, or to buy goods from a merchant with bitcoins, is thus incentivized to also adhere to the blacklist. Otherwise, they might accept coins that are worthless when being deposited at an exchange.

Depending on the legal implications of blacklisting, regulated intermediaries would either accept blacklisted coins, freeze them and report the incident to the responsible regulator, or be forbidden from accepting them in the first place. If coins get tainted while in control of an exchange, it can potentially sanction the user who deposited the coins, reduce the nominal value of their holdings on the exchange as well as report the user to law enforcement (this may depend on whether the coins were directly listed, or can be traced back to listed coins). In a follow-up

investigation users might be required to reveal from whom they received their tainted coins.

Intermediaries could offer customers a quality guarantee for coins [113], i.e. if purchased coins are retroactively blacklisted the exchange would compensate the user for the difference in quality or exchange the coins for clean ones. They could achieve such higher quality either by holding coins long enough (i.e. using more of their reserves to facilitate payments), or by adding a risk premium to their fees.

Not all intermediaries in the Bitcoin ecosystem are centralized or operate within a identifiable jurisdiction. There is a trend to decentralize services such as exchanges [20] or privacy services [110]. While decentralized platforms cannot be directly targeted by regulators, their users still have an incentive to adhere to blacklisting if they want to spend their coin at a regulated intermediary in the future (or want to transfer it to a user who does). As such, these platforms would lose users if they put them at risk of receiving tainted coins [2].

### **5.3.6. Making and accepting payments**

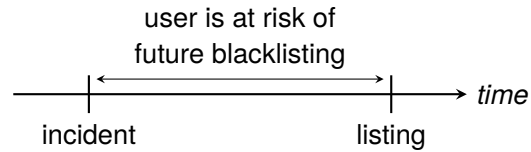
Whenever a user receives funds, they need to verify that the coins they are about to receive are not (significantly) tainted. In order to do so, they need to know which coins they will receive. Here, we sketch three different options.

**Returning coins.** A simple solution would be for a payee to return funds to the payer when the coins received are tainted. However, the address from which the coins were originally sent might not allow the payer to receive them back. For example, when sending money from an exchange, the coins might come from an address under the control of the exchange that is not directly linked to the user's account. Another disadvantage is that a refund transaction incurs additional

transaction fees and adds “unnecessary” transactions to the blockchain. Finally, it is undesirable that the payer has to hand over control of their bitcoins to the payee in order for them to check and accept or reject the payment.

**Refundable transfers.** To address the last issue, the payer could lock funds in a 2-of-2 multisig transaction that contains a time-locked refund. To spend these funds, both payer and payee need to sign a spending transaction. If the payee provides their signature, it indicates acceptance of the coins. Going further, the payer could lock more funds than nominally required and have the payee create a transaction with an amount that they deem equivalent to the risk of future blacklisting. The payer can then choose to agree or reject the transaction by signing or refusing to sign it. If at any point the payee becomes unresponsive, the payer can reclaim their coins after a short timeout period. While this solution provides more flexibility and the coins stay under the payers control until the payment is finalized, it always requires two on-chain transactions. It also may not be desirable when short transaction finalization times are important.

**Extending the Bitcoin Payment Protocol.** The Bitcoin Payment Protocol is a standardized protocol that allows a merchant (or payee) to specify details of a Bitcoin payment, such as the destination address and the amount, and make these programmatically available to the user’s wallet [7]. The user’s wallet receives a link to a payment request (e.g., by opening a URL or scanning a QR-code) and retrieves the payment details from the merchant’s website. This protocol could be extended to include specific details for the handling of blacklisted coins and the risk of future blacklisting. For example, it could specify whether tainted coins are accepted by the merchant and how these would be discounted. It could also specify a set of blacklists (e.g., as API endpoints) that the customer must check their coins against.



**Figure 5.2.:** In a blacklisting regime users are at risk of receiving coins that might get blacklisted in the future [113]

Most of the evaluation could happen on the payer’s side, preserving privacy as the merchant will only see the final set of coins chosen. If the user chooses coins in a way that violate the rules set by the merchant, the merchant can reject and refund the payment or, if legally required to do so, freeze the funds they received.

**Wallet support.** Both the multisig solution as well as the extension of the Bitcoin payment protocol require end-users’ wallets to support these specific protocols. Wallets furthermore need to modify their coin selection algorithms to take the taint of coins into account. Currently, coin selection is often optimized to reduce transaction fees or increase users’ privacy (cf. [1, 56]). In a blacklisting regime, wallets need to take the taint status as an additional constraint and may make further optimization depending on the taint policy (cf. Section 5.5).

## 5.4. Managing the Risk of Future Blacklisting

While it is straightforward to check whether coins are tainted at the time a user accepts them, in a blacklisting regime users also need to account for the risk of receiving coins that get blacklisted in the future [113]. After any illicit activity has taken place, there will be a time delay until those coins appear on the blacklist (cf. Figure 5.2). During this time a user might accept unlisted coins, only to later discover that those originate from illicit activity and have been listed, leaving them with coins they cannot spend. While this risk is similar to the current risk of receiving

coins that an exchange may not accept, the reduced time frame and the availability of public information enable new mitigation strategies. Users can combine these depending on their individual needs.

#### **5.4.1. Time-delayed payments**

The time since a coin was created is a useful indicator of its risk in the context of future blacklisting. Assuming that criminal activity is associated with some transfer of funds (e.g., payment of a ransom or theft of coins), then blacklisting should happen after the transaction has been put on the blockchain. The criminal hence wants to get rid of their illicit coins quickly, before they get listed. A coin that has been sitting in a user's wallet for longer has thus a lower risk of future blacklisting than a coin that has been moved recently. The longer a coin has been in a user's possession, the less likely it should be that it will get blacklisted in the future.

One way to systematically reduce the impact of future blacklisting would be to limit the time frame in which blacklisting can occur. If blacklisting needed to happen within, say, two weeks of an incident, then users could mitigate the risk of blacklisting by preferring coins that have been sitting unspent sufficiently long. (If the coins haven't moved since the crime occurred, they could still be listed after the blacklisting period expired since there is no collateral damage to the rest of the ecosystem). However, regulators might not be willing to limit their blacklist in such a way. Users who do not hold a variety of coins of different ages would also be at a disadvantage.

Using coin age as a proxy for risk is inapplicable when a criminal is able to steal a user's private keys. The coins would appear to have sufficient age, but once spent become subject to blacklisting. Data on private key theft is scarce: while high-profile attacks on exchanges often garner a lot of attention (cf. [32]), the effect on ordinary



users is likely under-reported. A survey conducted by Abramova et al. [3] found that it may pose a risk for specific groups of users, such as traders and investors. The risk of key theft can, however, be reduced by using storage options that are more secure than a regular software wallet. Larger amounts of bitcoins can be kept in multisig arrangements (where multiple keys are needed to authorize a transaction) or hardware wallets, and software solutions that provide built-in fail safes could further help alleviate the issue of private key theft (cf. [19, 100, 114]).

Even without a limited time period in which blacklisting occurs, increasing the time delay between coin transfers is an effective way to reduce the risk of future blacklisting. Users can make use of two features of the Bitcoin protocol to enforce a time-delay until coins can be spent: time-locked transactions, and time-locked outputs. Time-locked transactions are invalid until a future point in time (as specified by the `nLockTime` field) and will be rejected by the network until the time-lock has expired. A user could give such a time-locked transaction to a merchant, who waits until the time-lock expires and then submits it to the Bitcoin network. If coins get blacklisted before the transaction becomes valid, the transaction can simply be disregarded. A slightly different mechanism to create an artificially delay is to make use of the `CheckLockTimeVerify` opcode that allows to specify a time before or after an output can be spent by different public keys. This opcode can be combined with the aforementioned techniques to check the coin status before making a payment, where the payee would effectively receive a specific time window in the future in which they can claim the funds. It has the advantage of producing an on-chain transaction, addressing the issue of stolen public keys mentioned above: the affected user would be alerted by the move of their coins on-chain (similar to Bitcoin vaults [114]) and could blacklist the funds before they are accepted by the merchant.

We note that delaying payments is a common risk mitigation strategy in the financial sector today (e.g., [69]), and is used in the cryptocurrency space as well. Exchanges may delay the conversion or transfer of currency if additional compliance checks are necessary (e.g., [38]), and so they may adopt similar procedures when they deem funds to be at high risk of future blacklisting. Charging a risk premium or using insurance may be viable alternatives in settings where delays are undesirable.

#### **5.4.2. Risk scoring**

While the age of a coin is a strong indicator for the risk of future blacklisting, it is not the only feature that can be used to assess this risk (cf. [113]). A risk score can potentially be constructed based on a variety of indicators, including structural features of the transaction graph or private information about the ownership of addresses. Such a score is similar to traditional fraud detection for credit card payments or the risk assessment that banks are required to perform prior to conducting large payments [75]. However, in order to construct a precise risk model, relevant data must first be made available.

As discussed previously, cryptocurrency exchanges are already using risk scores provided by blockchain intelligence services to screen incoming payments. Chainalysis, for example, offers a “Know Your Transaction (KYT)” API that allows to “identify high risk transactions on a continuous basis” [34]. Currently, these scores are primarily available to enterprise customers such as exchanges. However, they could easily be offered to a larger group of users in order to be useful to supplement risk scoring in a blacklisting regime.

### 5.4.3. Payment networks

The Bitcoin protocol is inherently limited in the number of transactions that can be processed, as all transactions need to propagate through the network and be verified by miners and full-node operators [44]. Payment networks, i.e. a network of payment channels between individual users that can facilitate chains of payments (similar to credit networks), may provide one viable solution [131]. As these payments do not appear on the blockchain, it raises two important questions: can money laundering be facilitated through payment channels, and is blacklisting still effective if they become commonly used?

Payment channels [48, 131] allow users to conduct a potentially unlimited amount of transfers between each other, limited only by the amount of coins locked into the channel, by locally updating the state of the channel (i.e. increasing or decreasing the individual shares). Only when they no longer want to use the channel, the final balance (i.e. the settlement of all intermediary transactions) is committed to the blockchain.

Payment channels can be combined into payment networks that allow to route payments across multiple payment channels [101, 131], similar to transactions in credit networks (but without the risk of a party defaulting). This allows mutually distrusting parties to pay each other without having to open their own, direct channel.

Unsurprisingly, payment networks can be used to facilitate illicit activity. A ransomware operator could ask users to pay a ransom using a payment network instead of a normal Bitcoin transaction. However, payment networks are less attractive for money laundering on a large scale since the bandwidth available (i.e. the amount of funds that can be sent or received) is significantly lower than in a normal cryptocurrency transaction (e.g., the median channel size on the Lightning

Network amounts to only 0.01 BTC, about USD 600, as of October 2021 [70]). For a user to pay another user, there must be a path with sufficient bandwidth available to facilitate the pairwise payments. Normal users might not be willing to make large amounts of bitcoins available through these channels (every channel locks up capital that cannot be used otherwise). High bandwidth may be available from large hubs run by exchanges or other payment providers. These might decide to follow already established KYC procedures before initiating such a connection, enabling follow-up investigations that could reveal the criminal's identity or coins on the Bitcoin blockchain.<sup>4</sup>

Payment channels are not only unattractive for money laundering, they also remedy some of the drawbacks of blacklisting in Bitcoin for small payments. Channels are opened infrequently, hence parties only need to check the taint of the counterparty's funds when a channel is opened. Opening a channel however increases the impact of potential future blacklisting as channels are intended to be kept open for long periods of time. This gives payment networks characteristics similar to credit networks: while there's no direct risk of losing funds (the primary risk in a credit network is a defaulting counterparty), there remains the risk that at the time the channel is closed the funds in the channel have been listed. This incentivizes users to only open payment channels to highly trusted counterparties, such as friends or reputable and regulated intermediaries. Moving high-frequency transactions into off-chain networks also improves the effectiveness of on-chain transaction blacklisting, as coins are less likely to mix with each other and the overhead of checking blacklists and assessing the risk of future blacklisting is reduced.

---

<sup>4</sup>Currently, Lightning Network nodes do not need to register as MSBs as they do not take custody of other users' funds (cf. [156]).

#### **5.4.4. Identity and insurance**

Blacklisting introduces notions of trust into Bitcoin and increases the benefit of knowing the identities of counterparties. While many intermediaries in the Bitcoin ecosystem are already required by existing AML regulation to verify identities, normal users could be inclined to conduct similar identity checks prior to interacting with other users, e.g., to better assess the risk of future blacklisting or to be able to use traditional enforcement channels in case of future blacklisting. This could prevent some users from transacting pseudonymously and raise transaction costs of using Bitcoin. However, when large payments are conducted through Bitcoin they may already be governed by a contractual relationship that is aware of the counterparties' identities. Furthermore, such information is only acquired locally and not shared with outside parties or the government. For small payments, a small risk premium might be a viable alternative.

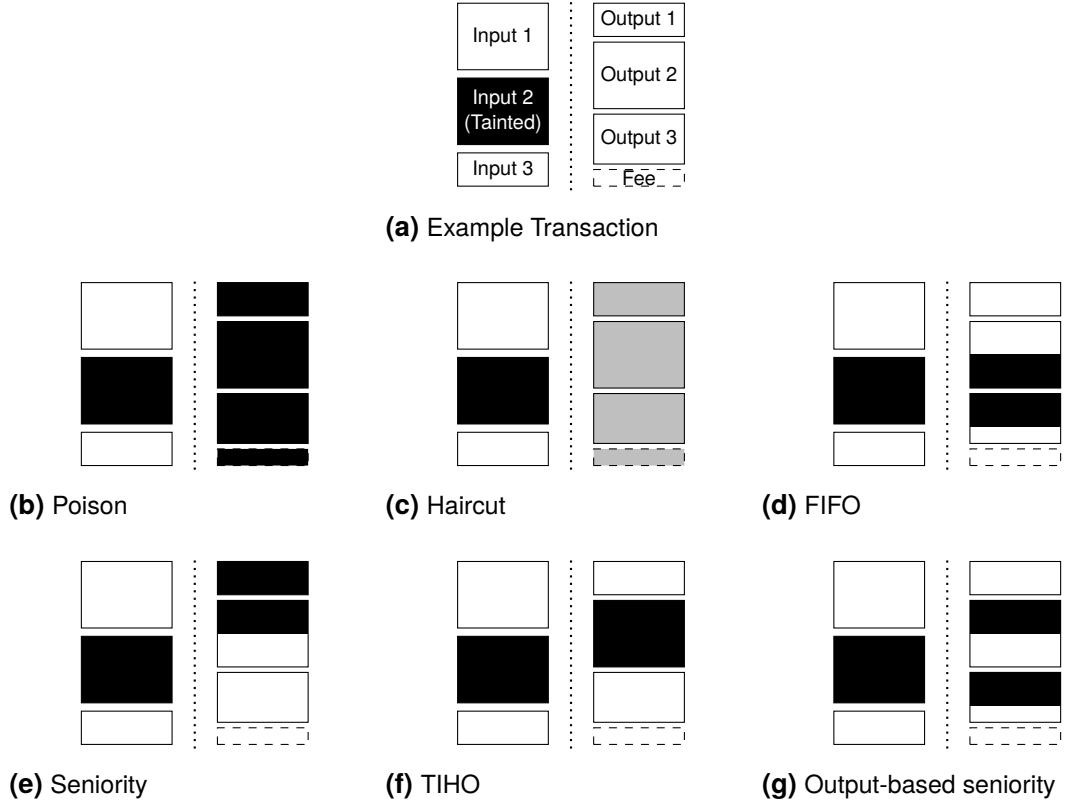
Another potential remedy is optional co-signing of transactions by well-known intermediaries. Today, co-signing is mostly used for enhanced security, but some intermediaries are using it as a mechanism to prevent double-spending: by having the assurance of a trusted third party to not double-spend a co-signed transaction, exchanges can accept payments that haven't yet been confirmed by the network (e.g., [73]). In the context of blacklisting, co-signing would indicate the ability for the regulated party to reveal the user's identity if funds get blacklisted, to signal that the intermediary determined the coins to be of low risk, or even to provide assurance that the third-party provides some form of insurance against such blacklisting. Insurance of this type could dramatically reduce the impact of long chains of transactions that become effectively invalidated due to retroactive blacklisting of funds in the first transaction.

## 5.5. Blacklisting Policies

To make blacklisting effective, taint must propagate through the transaction graph. A blacklisting policy specifies how exactly taint is mapped from incoming coins (inputs) to outgoing coins (outputs).

Several blacklisting policies have been discussed in the literature. Möser, Böhme, and Breuker [113] propose two policies. In the *Poison* policy, any tainted input completely taints all the outputs in a transaction. In the *Haircut* policy, any taint associated with an input is distributed equally among all outputs such that the total amount of tainted value does not change. These policies can quickly affect large parts of the transaction graph, yet may appear desirable from a legal perspective (e.g., due to Poison’s deterrent effect or Haircut’s perceived fairness, cf. [75]). Abramova, Schöttle, and Böhme [2] introduce the *Seniority* policy, in which taint is assigned to outputs in ascending order. The policy assigns taint in the inputs to the first output until all of its value is tainted, followed by the second output, and so on. Anderson, Shumailov, and Ahmed [5] and Anderson et al. [6] discuss the *FIFO* policy, where tainted and untainted value is mapped to outputs in the order that it enters the inputs. They show that there is legal precedent for using FIFO in English law and discuss advantages of using FIFO over the Poison and Haircut policies, such as tainting fewer outputs. Tironsakkul et al. [148] propose a *TIHO* policy that assigns taint towards the outputs containing the highest values, reducing the probability of taint being split by small outputs (e.g., change outputs).

In addition to these existing policies, combining ideas from the FIFO and the Seniority policy, we also analyze an *Output-based Seniority* policy, where taint is assigned to outputs similar to FIFO, but aggregated towards the beginning of each output. This could allow a more granular distribution of taint in transactions with



**Figure 5.3.:** How different tainting policies propagate taint from inputs to outputs

potentially many inputs and outputs. Figure 5.3 provides a visual explanation of these six policies.

We first systematize the design space for blacklisting policies and discuss the advantages and disadvantages of the individual policies. To this end, we put forward a set of characteristics to evaluate them. A *global* characteristic affects other users or impacts the overall system (e.g., by changing incentives), whereas *local* characteristics are relevant to individual users sending or receiving funds in a transaction. A *technical* characteristic is of particular interest to implementations of the policies, or blacklisting more generally. Table 5.2 provides an overview of the characteristics of the different policies.

### 5.5.1. Local characteristics

We start with two basic properties. A policy *preserves value* if the total amount of tainted value is the same before and after propagating taint. The Poison policy does not preserve taint, since any infinitesimal amount of tainted value in the inputs completely taints all value in the transaction's outputs. All other policies preserve taint, though the Haircut policy requires special attention regarding rounding errors.

A policy is *deterministic* when at the time of creating the transaction it is clear how a blacklisted input would affect the distribution of taint to the outputs. All policies discussed are deterministic, as their taint propagation does not depend on any outside circumstances. A counterexample would be a policy where the order in which outputs of a transaction are spent impacts the distribution of taint, which would create unintended incentives for the recipients of coins to spend or hold them.

Blacklisting policies can change the way users construct and agree on payments. Most importantly, the risk of future blacklisting associated with individual inputs can be accounted for by changing the structure of the transaction (e.g., the order of inputs and outputs), as it determines which (and to which degree) outputs are affected.

In a normal setting, the user creating the transaction (i.e. spending their own funds) may have an advantage in the form of additional information about the origin of the coins, and hence about the risk of future blacklisting. For example, they might know that a coin came from a reputable exchange, or from a questionable mixing transaction. Large intermediaries like exchanges or payment providers, on the other hand, may have access to additional private information that the user creating the transaction does not. For example, they might use a proprietary



blockchain analysis software that provides more accurate risk estimates than public sources. Furthermore, a merchant or exchange receiving money usually has a stronger bargaining position to enforce a transaction structure that limits their risk exposure. The construction of a transaction hence depends upon these potential information asymmetries and power imbalances.

A policy may allow users to *distribute risks* of receiving blacklisted coins, e.g., by allocating taint not equally among all outputs, but to specific outputs instead. Both Poison and Haircut do not allow to distribute risk since all outputs are affected equally (either completely tainted with Poison, or tainted relative to their value). FIFO and Output-based Seniority provide limited flexibility: reordering the inputs changes which outputs are affected by them. The Seniority policy provides most flexibility, as it assigns all taint to outputs in order. This enables users to direct the risk by choosing which output to put first. They can potentially even overfund that first output to add some buffer before taint is assigned to subsequent outputs. The TIHO policy assigns taint towards the output with the highest value. While this may allow a similar distribution of risk if output values can be freely chosen, it has several downsides. It requires the payer to have sufficient funds available (e.g., to create a change output larger than the actual payment), could result in larger transaction sizes due to more inputs being used, and might even create security risks if those additional funds need to be moved from cold to hot storage. None of the policies allow for more granular distribution, e.g., among a chosen subset of outputs.

To address the issue that the transaction creator may have private information about the quality of inputs, a policy can be considered *ungameable* only if modifying the order of inputs cannot influence a potential taint distribution in their favor. Any policy that does not allow to distribute risk is ungameable, and so are the Seniority

and TIHO policies where the distribution of taint only depends on the total amount of taint in the inputs.

In a traditional Bitcoin transaction, users may arrange inputs and outputs in any order they choose. This could change in a blacklisting regime, as users may construct transactions such that they achieve some desired outcome based on the specific policy. As a result, more information about the origins and recipients of funds is revealed, reducing users' privacy. Generally, a more even distribution of taint is more *privacy-preserving* than a concentration of taint, as it reduces the concern for which counterparty receives the taint.

Consider a typical Bitcoin transaction that has two outputs: a spend output and a change output. The Seniority policy allows one user to take over a larger share of the risk of receiving blacklisted coins by putting their output first. In a scenario where the buyer may have less bargaining power than the seller, they might be responsible for taking over the risk of receiving blacklisted coins. As a result, it's more likely that the change output is the first output rather than the second, which would be yet another indicator that an analyst could use to identify a users' addresses (cf. Chapter 4). For the TIHO policy, changing the order of the outputs has no effect, but the value of each output could similarly be used (within the limits described above) to distribute the risk. FIFO does not aggregate taint, but the order of inputs may reveal information about the expected risk for each output, which allows similar considerations about identifying the change output to be applied.

### **5.5.2. Global characteristics**

Enforcing blacklisting policies not only changes how users construct transactions, but can also more generally affect the whole cryptocurrency.

Blacklisting has the potential to impact many users once taint gets diffused among coins as they change hands. One strategy to reduce this diffusion is to *aggregate* taint in each transaction: rather than distributing taint equally among all outputs or splitting it up in small chunks, taint entering a transaction is combined. Consider the FIFO policy, where every small chunk of taint entering a transaction in the inputs is mapped identically to the outputs. Instead, the Seniority policy allocates all taint in the transaction towards the first outputs, thereby reducing diffusion. TIHO aggregates taint towards the output with the highest value(s), and output-based Seniority aggregates taint in individual outputs.

Blacklisting policies also need to take transaction fees into account and can thereby affect mining incentives. Tainting fees is important, otherwise any complicit miner could launder stolen coins through a transaction that designates all of its value to the fee. When tainting fees, the taint of individual transaction fees must thus be mapped into the coinbase transaction of the block that includes the transaction, even though no direct reference exist. If the mechanism chosen to taint fees minimizes the risk for miners to receive tainted fees and does not disincentivize them from including transactions, the policy is *miner-friendly*. In practice, one may not be able to prevent transaction fees from ever be tainted or the miners from ever be affected. However, the policy should allow to minimize such interference with a miner's incentive to include transactions based on the fee they pay.

To consider transaction fees in such a way, two design choices must be made: the position of the fee as a “virtual output” in a transaction (cf. Figure 5.1), and the position of the fee as a “virtual input” in the coinbase transaction. The choice of the output position is most important for the Output-based and regular Seniority policy. Setting the fee first would increase the likelihood that a miner receives tainted fees, setting it last makes the policy more miner-friendly. For TIHO, tainting the fee last,

despite of its value compared to the outputs, would maximize its miner-friendliness. The coinbase transaction contains not only transaction fees of all transactions in the block, but also a fixed amount of newly minted (and thus untainted) coins. Positioning the fees after the reward would allow a miner to not claim transaction fees but still include transactions when using the FIFO or Output-based seniority policy (as miners can choose to claim less than the sum of coinbase reward and total amount of fees in a block). Putting the fees first denies miners this flexibility. Poison and Haircut are both unfriendly towards miners, as any taint in the inputs will taint the transaction fee. Hence, likely tainted coins will move slower, or not at all.

### 5.5.3. Technical characteristics

FIFO allows to track the taint of each blacklisted coin individually. Because the policy does not aggregate taint and propagates it exactly as it enters a transaction, tainted chunks do not impact each other. This is in stark contrast to aggregating policies like Seniority, where different tainted inputs can individually lead to the same outcome (e.g., the first output being tainted), but taken together produce a different result (e.g., more than one output being tainted). As a result, FIFO allows to *combine* taint individually propagated from different listed coins into a full taint mapping. This would make it easier to follow different blacklists, as their individual results can be combined easily.

At the same time, since FIFO tracks individual chunks of taint it requires to maintain a lot more information than the other policies, all of which can *compactly store* the outputs' taint status using a single integer value.

Anderson, Shumailov, and Ahmed [5] furthermore highlight that the FIFO policy allows to follow taint backwards (*backtrackability*): starting from a tainted output one can map the taint back to previous transactions. This doesn't have a direct impact

**Table 5.2.:** Characteristics of different blacklisting policies

Characteristic	Poison	Haircut	FIFO	Output-Seniority	Seniority	TIHO
Value-preserving	○	◐	●	●	●	●
Deterministic	●	●	●	●	●	●
Risk is distributable	○	○	◐	◐	●	◐
Ungameable	●	●	○	○	●	●
Privacy-preserving	●	●	◐	◐	○	◐
Aggregating	○	○	○	◐	●	●
Miner-friendly	○	○	◐	◐	◐	◐
Combinable	●	○	●	○	○	○
Compact storage	●	●	○	●	●	●
Backtrackable	○	○	●	○	○	○

Provides property ● fully, ◐ partially, ○ not

on the user, but potentially allows for more efficient calculation of the taint status of outputs in a database system.

#### 5.5.4. Empirical evaluation

We conduct an empirical evaluation of the behavior of the different taint policies. To this end, we implement them on top of the open-source blockchain analysis tool BlockSci [87]. While a backwards-looking analysis is not necessarily indicative of how a blacklisting policy would affect transactions once it is put into place, as users would change their behavior in response, it is useful to get a better understanding of the overhead the policies *could* impose. As such, the analysis could be interpreted as a worst-case analysis, where users pay no attention to the taint of a coin.

We evaluate the policies on four datasets: ransomware payments for the Cerber and Locky ransomware [84], payments to addresses blacklisted in 2018 by OFAC [151], a list of addresses from blackmail (sextortion) scam emails collected by us

**Table 5.3.:** Total number of outputs tainted, transactions traversed and percentage of tainted value ending up in transaction fees for different datasets on 06/31/2020

Dataset/Measure	Poison	Haircut	FIFO	Output-Seniority	Seniority	Juniority	TIHO
<i>Blackmail (n = 273, Apr 2018—Apr 2019)</i>							
# outputs tainted	32,743,834	11,290,000	2,821	11,913	22,215	10,330	304
# txes traversed	223,333,836	183,411,521	317,391	879,722	1,093,422	1,471,504	89,183
% fee taint	—	—	1.06	2.64	1.38	0.69	0.09
<i>OFAC (n = 3305, Sep 2013—Jun 2020)</i>							
# outputs tainted	—	—	270,537	451,581	870,672	456,477	29,963
# txes traversed	—	—	44,365,714	14,566,976	18,145,061	40,740,508	8,557,248
% fee taint	—	—	2.02	4.75	3.63	1.88	0.26
<i>Ransomware (n = 16777, Feb 2016—Aug 2017)</i>							
# outputs tainted	—	—	576,694	840,363	1,591,322	617,263	48,510
# txes traversed	—	—	59,863,788	20,762,137	26,485,103	31,416,342	12,236,429
% fee taint	—	—	1.94	3.82	2.63	1.43	0.33
<i>Random outputs (n = 100, Jan—Dec 2019)</i>							
# outputs tainted	25,986,492	16,800,865	24,813	51,714	91,098	34,096	3,291
# txes traversed	166,087,796	160,081,753	2,872,369	2,147,643	2,589,374	3,024,931	866,940
% fee taint	—	—	1.0	2.37	1.27	0.61	0.23

(listed in Section C.1), as well as a randomly chosen set of 100 outputs created in 2019 (each worth over BTC 0.001). For each dataset we mark the outputs as fully tainted and then propagate the taint through the transaction graph, up to height 637 091 (June 31, 2020) of the Bitcoin blockchain. We are interested in the total number of outputs tainted, the their age and value distributions, as well as their impact on transaction fees.

Both the Poison and Haircut policy affect significantly more outputs than the other policies, confirming our expectation and prior analysis by Anderson et al. [6] (cf. Table 5.3; while in theory Haircut should affect the same total number of outputs, very small taint values and rounding lead to only a subset of those outputs receiving taint). This makes them undesirable for practical use.

Between FIFO, Seniority and Output-based seniority, FIFO consistently tainted fewer outputs than the Seniority policies. This is surprising at first, as Seniority can merge taint of multiple inputs into a single output. Inspecting the tainted outputs, we discovered that structural properties of the current transaction graph are likely responsible for this. For example, funds often ended up in outputs with

very small values (546 satoshi), which are often used to confer additional metadata in transactions by protocols such as Omni Layer (cf. [16]). Other common motifs, such as peeling chains (a pattern where outputs of small amounts are repeatedly created), could similarly play a role in the large number of outputs tainted by Seniority. To assess the effect of transaction graph structure on the Seniority policy, we implemented a Juniority policy (i.e. reversed Seniority), assigning taint to outputs starting from the last to the first. We see that for all datasets this significantly reduces the number of outputs tainted, confirming our suspicion.

The TIHO policy taints the least amount of outputs by a large margin. This is not surprising, as it assigns taint towards the outputs with the highest values, minimizing the need to split taint. At the same time, it suggests that with strategic use aggregating policies, such as Seniority, could achieve far better results (as TIHO and Seniority with outputs ordered by value produce the same outcome).

With FIFO, an output can contain multiple chunks of taint. The difference between the number of outputs and the number of chunks varied considerably between the datasets (cf. Table C.1), and reached up to four times as many chunks as outputs (despite merging adjacent chunks). A larger number of chunks will make reasoning about the selection of inputs for a transaction more complicated in practice. We also find that it impacts transaction fees more than the Juniority and TIHO policy, with between 1–2 % of tainted value ending up in transaction fees.

In Section C.2 we provide value and age distributions for these analyses. Comparing the share of taint within tainted outputs we find that most outputs tend to either be fully or barely tainted. Inspecting the creation time of the final set of tainted outputs, we see that they are skewed towards more recent times, but distributed over the entire time frame.

### 5.5.5. Discussion

Both the Poison and Haircut policies' distribution of taint among all outputs is beneficial for privacy, but—unless users change behavior—too easily impacts large parts of the transaction graph. This motivates the use of more granular policies.

FIFO and Seniority/TIHO present two different approaches towards taint propagation. FIFO aims for an exact tracking of each individual chunk of taint. The primary benefit of this is that changes to the blacklist have straightforward implication and results from different blacklists can easily be combined. However, the policy is not ungameable: it gives the payer an advantage over the payee when negotiating the transaction structure. This complicates users' behavior around negotiating the transaction structure.

Seniority, in contrast, provides ungameability at the cost of reduced privacy. If used strategically, it could also reduce the number of outputs affected (as seen with TIHO). An important avenue for future research will be to model how users could negotiate transaction structures when using Seniority compared to FIFO.

We imagined the Output-based seniority policy as a middle ground between FIFO and Seniority, and we find that it taints fewer outputs than Seniority. However, it inherits FIFO's issue of gameability. Combined with the mix of policy behaviors, it may make it difficult to provide an intuitive experience for users.

We note that all policies discussed here are inherently limited in their flexibility to assign taint. For example, no policy allows the user to distribute taint equally among a subset of outputs. To this end, one might consider allowing transactions to specify more advanced taint policies (e.g., encoded in an `OP_RETURN` output) with a fallback to a default policy.



## 5.6. Blacklisting and Privacy

Blacklisting is enabled by the ability to follow coins from one transaction to the next. At the same time, this transparency has implications for users privacy, and a variety of designs for more private cryptocurrencies have been proposed [17, 105, 157] and are being deployed. This raises questions about the privacy implications of blacklisting, how it interacts with privacy-enhancing techniques, and whether adopting blacklisting in Bitcoin would push criminals to use more private cryptocurrencies where regulation of this form is not possible.

### 5.6.1. Compatibility with privacy techniques

A variety of techniques have been proposed to increase the privacy of Bitcoin transactions. Already, the basic structure and details of a transaction can vary considerably, allowing to tell transactions apart (e.g., the protocol characteristics or script types used can serve as a form of fingerprint [87]). Improving privacy by making transaction appear more uniform is an ongoing process in Bitcoin, and helpful new protocol features are expected to be deployed by the end of 2021 (e.g., Schnorr signatures and Taproot script programs [80]). However, since none of these approaches change the fundamental structure of transactions, nor the link between inputs and previous outputs, they are fully compatible with blacklisting.

Arguably the biggest privacy issue in Bitcoin right now is address clustering and reuse [79, 103]. It allows to link multiple payments to or from the same user together, potentially revealing their transacting patterns and other related addresses that are owned by the same user. Applied on a broad scale, it may allow to associate large parts of activity on the blockchain with specific identities.

Since blacklisting does not depend on addresses, it is compatible with techniques that provide remedies against address reuse, such as the use of one-time addresses

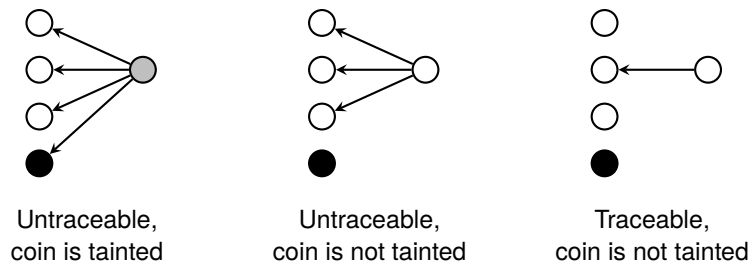
as in Monero [43]. So far, these techniques have seen limited use in Bitcoin due to their complexity and performance overhead [109], but they could be added on top of Bitcoin without affecting blacklisting.

Another possible privacy improvement is to hide the values of transaction outputs. Unique values may allow to infer the purpose of transactions or perform re-identification through other datasets [71]. Hiding them would, however, reduce the choice of a blacklisting policy to the poison policy. All other policies make use of output amounts to map taint to the outputs.

### **5.6.2. Incompatibility with privacy techniques**

A variety of privacy overlays have been proposed for Bitcoin, the most popular being CoinJoin (e.g., [99, 102, 109, 110, 139]). In a CoinJoin transaction, multiple users facilitate a joint transaction by combining multiple sets of inputs and outputs into a single transaction. Such a transaction makes it hard to determine which inputs correspond to which outputs, and it breaks a popular deanonymization heuristic that assumes that all inputs to a transaction belong to the same user. In a blacklisting regime, CoinJoin users face the risk of the coins they receive being tainted by other users' inputs, which would lead to a decline in the use of CoinJoin in the absence of signaling or coin negotiation [2, 110]. The same holds true for other techniques of mixing coins, e.g., using centralized mixers.

Recall that tainting is concerned with mapping taint from inputs to outputs, whereas tracing is concerned with the ability to identify which coins are spent in a transaction (cf. Section 5.3.1). The latter is useful to efficiently verify the validity of a transaction, and so far we've assumed perfect traceability of coins. But while Bitcoin's traceability provides great transparency and enables blacklisting, being able to tell which coins are spent in a transaction is also a potential privacy




**Figure 5.4.:** Limited untraceability does not preclude taintability

issue. Other cryptocurrency designs, such as Zerocash [17] or Cryptonote [117, 157] are designed to obfuscate this connection and increase privacy by making coins untraceable. In these cryptocurrencies there no longer exists a unique link from one transaction to the next, rather, there is an anonymity set of possible links (the size of which varies between designs). In Monero (based on the Cryptonote protocol), transactions select a small number of possible origins for coins spent.

While perfect traceability makes tainting coins straightforward, it is not necessary to give up all privacy in order to achieve effective tainting. Blacklisting aims to separate illicit coins from the legal economy. This requires to identify illicit coins, but not which coin exactly a user is spending. It only matters that they are not spending a tainted coin. In principle, users could still enjoy an anonymity set of potential coins that they are spending from, as long as these don't include any tainted coins (cf. [89]). Figure 5.4 visualizes this distinction: by excluding the tainted coin from the set of possible coins being spent one can prove that their coins don't have illicit origin while still including other, untainted coins in order to retain (now slightly reduced) untraceability.

In cryptocurrencies based on the Zerocash design, the entire set of outputs can make up the anonymity set. So far, users have been slow to adopt these privacy features [88], but widespread use would make the coexistence of untraceability and taintability difficult. The major challenge in enabling blacklisting for untraceable

Privacy technique	Viable taint policies
Transaction uniformity	All
+ address unlinkability	
+ encrypted amounts	Poison only
+ transaction unlinkability	None


  
*higher degree of privacy*

**Figure 5.5.:** Applicability of taint policies given levels of anonymity in the cryptocurrency

cryptocurrencies is to develop an efficient but privacy-preserving mechanism that allows users to dissociate their coins from tainted coins. But because all outputs on the blockchain make up the anonymity set, this needs to be recursively proven for all coins created after the tainted coin appeared on the blockchain. When coins are retroactively blacklisted, or a listing is removed from the blacklist, the proofs would need to be updated as well. This requires a high degree of interactivity and cooperation by users (cf. [89]), or would need to be build into the protocol and client software. Furthermore, such a dynamic system for checking the inclusion of specific coins must not break the anonymity of the system by allowing to enumerate all outputs on the chain, requiring some form of rate limit or authoritative approval. Whether this can be incorporated into the cryptographic zero-knowledge proof systems while preserving the decentralized, non-interactive nature of cryptocurrencies is an interesting open research question.

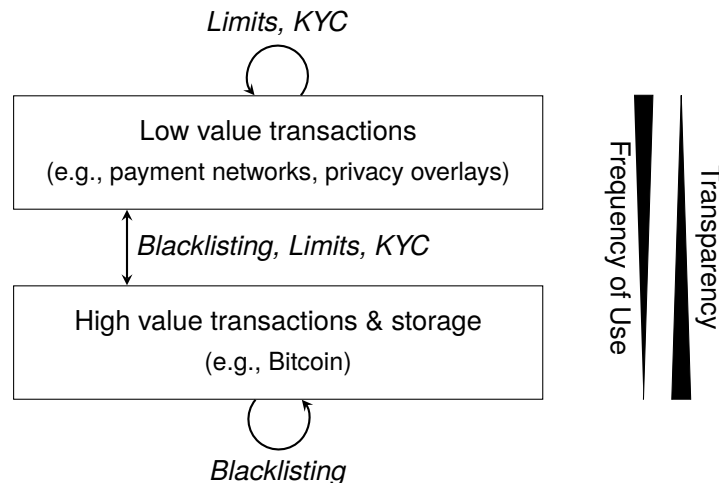
In Figure 5.5 we summarize the effect of the different privacy mechanisms on the choice of a taint policy.

### **5.6.3. Towards regulation that balances privacy with regulatory and investigatory needs**

As the cryptocurrency ecosystem matures, new technical solutions are being developed that increase financial privacy (e.g., [17, 105, 130, 157]). Privacy is important: individual users don't want their everyday purchases to be identifiable on the blockchain, and companies prefer to hide their financial activity from competitors. At the same time, private, decentralized cryptocurrencies raise challenges for regulators and law enforcement to investigate crimes such as money laundering, corruption or tax evasion and to enforce relevant laws. While regulators shouldn't (and likely cannot [155]) hinder the development and distribution of such systems, they may be able to steer their development and adoption towards a cryptocurrency landscape that balances privacy with regulatory needs.

In the following, we attempt to sketch one option for a potential middle ground, combining blacklisting with privacy-preserving payments. Users should be able to transact privately for everyday purchases (e.g., buying medications at a pharmacy), while payment systems would, at the same time, provide transparency and accountability for larger payments (e.g., significant financial donations to political parties) with the ability to blacklist funds from illicit origins.

We divide payment activity in cryptocurrencies along these lines (cf. Figure 5.6). High value transactions and the storage of large amount of coins is done in transparent cryptocurrencies such as Bitcoin. Large cryptocurrency holdings are usually accessed infrequently, and can be securely stored, e.g., using hardware security modules or by implementing additional fail safes [100, 114]. Large transactions are also facilitated in this layer. If any of those coins are stolen or used for illegal purposes, they could be effectively blacklisted due to their low frequency of use.



**Figure 5.6.:** Framework for Effective Cryptocurrency Regulation

More frequent, low value transactions are facilitated on a second layer. These could involve payment networks and other overlays, e.g., focused on increased privacy protections. The value that can be moved through this layer would be limited, either organically (e.g., channel capacity and number of open channels in payment networks) or through technical means. For example, these systems might cryptographically ensure that users can only transact up to a certain limit without regulatory oversight [163], or more rigid KYC requirements could be enforced by centralized overlays. A third approach could be to enforce regular movement of coins out of the private overlay layer into the transparent base layer.

Blacklisting or technical limits at the intersection between the two layers help to prevent structuring, i.e. splitting up large amounts into smaller ones to launder them in the second layer. With payment channel networks, blacklisting creates a disincentive for money mules to open channels with illicit funds, since they would be stuck with tainted coins. The same applies to more centralized solutions, where coins need to be checked when moved in and out of the system.

## 5.7. Discussion of Common Concerns

In this section we address some concerns around blacklisting that have surfaced in discussions about the topic.

### 5.7.1. Concern: Blacklisting destroys Bitcoin

A common concern is that blacklisting could potentially make Bitcoin unusable as a whole. If tainted coins mix too quickly with other coins, especially on exchanges that process large amounts of value, then a large number of coins could quickly become tainted. Also, users could become too afraid of receiving illicit coins and turn to alternative payment methods due instead.

A related concern is reduced fungibility. Fungibility describes the ability to exchange one coin for any other coin of the same denomination. It is reduced if users are able to discriminate between coins of different quality. Bitcoins are by design not fungible. The ability to discriminate between coins based on their history is a core aspect of Bitcoin's design: the ability to uniquely identify and keep track of unspent coins enables the efficient verification of transactions.

There may be instances of successful money laundering where retroactive blacklisting could affect a large number of (innocent) users. This might deter regulators from blacklisting those funds. While the threat of blacklisting even in those situations is (to a certain degree) required to make users check blacklists in the first place, in practice regulators will likely seek a balance between these two competing interest. As discussed in Section 5.5, more granular blacklisting policies than Poison or Haircut can help to reduce the impact of such listings.

Importantly, not implementing blacklisting does not mean that coins are fungible. Funds that blockchain intelligence services suspect to originate from illicit activity may be rejected or frozen by exchanges. Blacklisting makes the loss of fungibility

transparent, allowing users to manage the risk proactively. If taint tracking is included in common wallet software (and dealt with automatically by exchanges), then the practical impact on users can be lowered.

Given today's importance of centralized exchanges, their ability to provide more user-friendly solutions for dealing with blacklisting (cf. Sections 5.3.5, 5.4.2 and 5.4.4) might initially drive more users towards centralized solutions. This could temporarily reduce the utility of unhosted wallets until sufficient tools are available for normal users to manage tainted coins effectively.

### **5.7.2. Concern: Blacklisting destroys privacy**

The goal of tainting illicit coins is to prevent them from being mixed with legitimate coins. Tainting an output does not require information about the owner of the coin and also does not reveal any additional identifying information (unless the regulator chooses to make information about the circumstances that lead to the listing public). In the long term, blacklist-based regulation could reduce the reliance on address-based deanonymization for money-laundering detection, yielding an overall benefit for users' privacy as a substitute to inefficient KYC procedures.

When users decide whether to accept coins, there are two potential impacts on privacy. While the blacklist allows users to check coins for potential taint, they might ask for the counterparty's identity in order to assess the risk of future blacklisting, or to enforce claims through the legal system should the coins get blacklisted in the future. However, in many commercial settings such information is known already and stays private with the two parties involved only.

The choice of the policy also affects privacy (cf. Section 5.5). For example, with the Seniority policy some outputs can be more likely to be change outputs than



others. This should be a consideration when choosing a policy, with FIFO and Output-based seniority providing potentially more privacy.

### **5.7.3. Concern: Criminals will use anonymous cryptocurrencies instead**

Another concern is that blacklisting can be evaded and money laundering continued to be facilitated by switching to more anonymous cryptocurrencies. While cryptocurrencies like Zcash and Monero offer more privacy than Bitcoin and blacklisting coins isn't feasible on their blockchains (cf. Section 5.6.2), they at the same time create additional hurdles for criminals (and their victims) to use. Ease of use and the ability to quickly acquire and sell cryptocurrency are desirable not only for ordinary users, but also for criminals. Compared to Bitcoin, privacy-focused cryptocurrencies enjoy lower acceptance at intermediaries, are available for purchase on fewer exchanges (cf. [40]) and are harder to use (e.g., there are fewer wallet implementations available). And due to their anonymous nature, these cryptocurrencies pose a higher risk for exchanges and as such their use (and customers) may be more closely monitored, which could deter criminals or make subsequent investigations feasible.

All these factors could shift if blacklisting was widely deployed and effective. We believe that the degree to which criminals would then increase their use of privacy-focused cryptocurrencies depends to some degree on regulators' reaction to towards them. That Bitcoin's transparency enables certain types of criminal investigations might be one of the reasons why regulators haven't regulated cryptocurrencies more aggressively yet [119]. If the illicit use of privacy-focused cryptocurrencies increases, the regulatory landscape might change in response (cf. [154]). We believe that finding a balance between transparency and privacy is possible (as discussed in Section 5.6.3).

#### **5.7.4. Concern: Blacklisting will turn into whitelisting**

To manage the risk of future blacklisting, intermediaries might feel pressured to switch the system around, tracking and accepting only coins with perfect quality and known history, effectively creating a system in which only “whitelisted” coins have value.

Regulated intermediaries could indeed exchange information about coins they hold to make risk scoring easier. Coins held by exchanges with strong AML programs would have a lower risk than those with unknown histories. Yet, a purely whitelist-based system would have significant overhead over a blacklist-based system. Tracking and constantly verifying the set of all whitelisted coins would require to exchange a lot of privacy-sensitive information and have much worse performance than tracking blacklisted coins—after all, illicit activity only constitutes a small share of activity in cryptocurrencies [32]. And due to the risk of retroactive blacklisting it would be impossible to avoid that certain coins initially considered to be of good quality will eventually be listed, affecting the quality of other coins and reducing the set of whitelisted coins. Guidance on how to manage both blacklisted coins and coins at high risk of future blacklisting thus needs to be specific to remove legal uncertainty while providing intermediaries with sufficient flexibility to prevent such a scenario from materializing.

Note that even in the absence of blacklisting intermediaries still face an AML risk and could decide to implement a whitelisting-system to reduce it, effectively creating a walled garden rather than embracing the transparency and open nature of cryptocurrencies.

### 5.7.5. Concern: Blacklisting can be avoided by moving coins across chains

If criminals can easily convert their coins into another cryptocurrency, then blacklisting would potentially have to occur across different chain, making it harder to apply and manage in practice.

However, blacklisting across chains is only necessary when the original coins do *not* retain their value on the original chain. In most cases, users will sell their coins to a counterparty and receive new coins in a different cryptocurrency in return. Similar to normal transactions, the counterparty is incentivized to check the taint and risk of future blacklisting before accepting coins, hence there is no need to track taint across currencies. This not only applies to centralized exchanges, but also to most atomic swap protocols that allow users to exchange their coins without counterparty risk. As discussed in Section 5.6.2, the behavioral change that blacklisting induces may effectively “dry out” some high-risk applications in the absence of signaling or effective negotiation protocols (cf. [110]).

In a few scenarios coins may actually (temporarily) lose their value on the original chain. With a sidechain, coins are locked on the original chain, effectively transferring their value to the sidechain [11]. Users accepting coins on the sidechain would need to check the taint of the coins that were locked up on the original chain. Other mechanisms could include “burning” coins on one chain (i.e. making them unspendable) in order to receive coins of equal value on another chain. In such cases, the taint of the original coins would need to be applied to the newly minted coins on the other chain (similar to how transaction fee taint is propagated into the coinbase transaction). This could be automated if redeeming coins on the new chains involves a (transparent) proof of the coins being destroyed/locked up on the original chain.

### **5.7.6. Concern: Blacklists can be abused for political censorship**

Another concern is that governments can use blacklists to freeze funds outside of money laundering, for example, for the purpose of political censorship. However, blacklisting should not significantly increase the potential for such actions as the transparency of the system also increases accountability. A public listing can be discussed and objected much more easily than an account closure at a bank for undisclosed reasons. Thus, blacklists do not enable censorship beyond what could already be attempted today, e.g., through exerting political pressure on regulated intermediaries to freeze or reject certain coins.

## **5.8. Summary**

Transaction blacklisting can be an effective regulation approach that works on top of existing cryptocurrencies, improves AML outside of regulated intermediaries, and protects users from inadvertently accepting illicit funds. It complements existing AML regulation and reduces reliance on opaque transaction screening services widely used today.

Improving AML through blacklisting may introduce some of the inefficiencies from traditional financial systems, such as delayed payouts to protect against future blacklisting. At the same time, it disincentivizes criminal use and retains cryptocurrencies' openness, decentralized infrastructure and transparency. Regulators shouldn't be afraid to trade off some of cryptocurrencies' widely touted advantages in order to reinforce AML efforts through new regulation strategies.

We hope this chapter provides a starting point for discussion around the feasibility of blacklist-based regulation of cryptocurrencies. If cryptocurrencies are to become

more widely adopted, effective means of addressing financial criminal activity will be desperately needed.

# 6

## Conclusion

Our work in this thesis has highlighted the interdependent nature of cryptocurrency privacy. All too often, the behavior of some users affected the privacy of *others*, as we've seen in the deducibility attack on Monero transactions. And an even greater risk comes from privacy-sensitive disclosures of a small group of users that can affect the system as a whole, such as our ability to build a machine learning model to predict change outputs with high accuracy. Developers should consider these risks when building protocols or standardizing behavior, and both consider and analyze the use of their protocols in privacy-sensitive scenarios. Based on conversations with developers, companies in the Bitcoin ecosystem, law enforcement agencies, and regulators, we believe our work in this thesis has been highly insightful for all parties involved, as they work towards a shared goal of protecting privacy while reducing illicit activity.

Looking ahead, our work raises questions about the feasibility of future research into cryptocurrency privacy. Despite the openness of cryptocurrencies, researchers have increasingly limited insight into the nature of transactions as transaction volumes grow, use cases and the number of intermediaries expand, and information about how and why transactions are created is often proprietary. In both analyses of Monero and Bitcoin we utilized privacy-compromising behavior of users to learn about their activity. This is not a sustainable approach in the long term, as cryptocurrency protocols mature and privacy weaknesses get resolved.

Importantly though, in both analyses of Monero and Bitcoin, our collected data allowed us to identify further issues and generalize beyond the initially affected population of users. Fixing the initial weakness does not fix the subsequent issues we discovered, and without insights into the system it will be hard to determine the degree of privacy it actually provides in practice. For example, preventing address reuse in Bitcoin will make multi-input clustering less effective, preventing us from constructing a ground truth data set of known change outputs. However, it would not change the feasibility of change address clustering itself. Similarly, fixing the deducibility issue in Monero did not fix the mismatched sampling distribution.

Cryptocurrency protocol designers and security researchers may need to consider new methods to study privacy in these systems. On the one hand, they could consider cooperating with companies that have access to private information due to their respective business activities. On the other, they could try to incorporate privacy-preserving data collection mechanisms into client software itself. For example, techniques based on differential privacy [53] can be used to collect telemetry data from users in a privacy-preserving way (e.g., [57]).

In Chapter 5 we described how regulators can use the transparency of Bitcoin to address illicit activity. An important takeaway here is that blacklisting induces behavioral change. As such, it would be interesting to study blacklisting experimentally or more formally through the lens of mechanism design. While regulators may not want to rush adopting technology-specific regulation, increasing interest and guidance from the side of regulators such as OFAC could soon make it necessary to think about these issues more concretely.

# Bibliography

- [1] Svetlana Abramova and Rainer Böhme. “Your Money or Your Privacy: A Systematic Approach to Coin Selection.” In: *Cryptoeconomic Systems '20*. 2020.
- [2] Svetlana Abramova, Pascal Schöttle, and Rainer Böhme. “Mixing Coins of Different Quality: A Game-Theoretic Approach”. In: *Financial Cryptography and Data Security*. Ed. by Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y.A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, and Markus Jakobsson. Cham: Springer International Publishing, 2017, pp. 280–297.
- [3] Svetlana Abramova, Artemij Voskoboynikov, Konstantin Beznosov, and Rainer Böhme. “Bits Under the Mattress: Understanding Different Risk Perceptions and Security Behaviors of Crypto-Asset Users”. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–19.
- [4] C. Aliens. *Reddit User Explains How To Use Monero and “Not End Up in Jail”*. 2016. URL: <https://www.deepdotweb.com/2016/09/03/reddit-user-explains-use-monero-not-end-jail/>.
- [5] Ross Anderson, Ilia Shumailov, and Mansoor Ahmed. “Making Bitcoin Legal”. In: *Security Protocols Workshop*. 2018.
- [6] Ross Anderson, Ilia Shumailov, Mansoor Ahmed, and Alessandro Rietmann. “Bitcoin redux”. In: *17th Annual Workshop on the Economics of Information Security (WEIS)*. 2018.
- [7] Gavin Andresen and Mike Hearn. *BIP 70: Payment Protocol*. URL: <https://github.com/bitcoin/bips/blob/master/bip-0070.mediawiki> (visited on 01/13/2019).



- [8] Elli Androulaki, Ghassan O Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. "Evaluating user privacy in Bitcoin". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2013, pp. 34–51.
- [9] James T. Areddy. *China Creates Its Own Digital Currency, a First for Major Economy*. URL: <https://www.wsj.com/articles/china-creates-its-own-digital-currency-a-first-for-major-economy-11617634118> (visited on 10/04/2021).
- [10] Kristov Atlas. *Lexicographical Indexing of Transaction Inputs and Outputs*. URL: <https://github.com/bitcoin/bips/blob/master/bip-0069.mediawiki> (visited on 01/20/2020).
- [11] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. "Enabling blockchain innovations with pegged sidechains". In: (2014).
- [12] Andrew Balthazor. *The Bona Fide Acquisition Rule Applied to Cryptocurrency*. 3 Geo. L. Tech. Rev. 402. 2019.
- [13] Solon Barocas and Karen Levy. "Privacy Dependencies". In: *Washington Law Review* 95 (2020), p. 555.
- [14] Solon Barocas and Helen Nissenbaum. "Big Data's End Run around Anonymity and Consent". In: *Privacy, big data, and the public good: Frameworks for engagement* 1 (2014), pp. 44–75.
- [15] Massimo Bartoletti, Barbara Pes, and Sergio Serusi. "Data mining for detecting Bitcoin Ponzi schemes". In: *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE. 2018, pp. 75–84.

- [16] Massimo Bartoletti and Livio Pompianu. “An analysis of Bitcoin OP\_RETURN metadata”. In: *Financial Cryptography and Data Security: FC 2017 International Workshops, WAHC, BITCOIN, VOTING, WTSC, and TA, Sliema, Malta, April 7, 2017, Revised Selected Papers*. Springer. 2017, pp. 218–230.
- [17] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. “Zerocash: Decentralized Anonymous Payments from Bitcoin”. In: *Proceedings of the 2014 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2014, pp. 459–474.
- [18] Gergely Biczók and Pern Hui Chia. “Interdependent privacy: Let me share your data”. In: *International conference on Financial Cryptography and Data Security*. Springer. 2013, pp. 338–353.
- [19] Bryan Bishop. *Bitcoin vaults with anti-theft recovery/clawback mechanisms*. 2019. URL: <https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2019-August/017229.html> (visited on 09/25/2019).
- [20] *Bisq - The decentralized Bitcoin exchange*. URL: <https://bisq.network/> (visited on 02/13/2019).
- [21] *Bitcoin Core 0.16.0*. URL: <https://bitcoincore.org/en/releases/0.16.0/> (visited on 02/25/2021).
- [22] Desamparados Blazquez and Josep Domenech. “Big Data sources and methods for social and economic analyses”. In: *Technological Forecasting and Social Change* 130 (2018), pp. 99–113.
- [23] *Blockchair.com API v.2.0.76 Documentation: Privacy-o-meter*. URL: [https://blockchair.com/api/docs#link\\_M6](https://blockchair.com/api/docs#link_M6) (visited on 02/22/2021).

- [24] Rainer Böhme, Nicolas Christin, Benjamin Edelman, and Tyler Moore. “Bitcoin: Economics, technology, and governance”. In: *Journal of Economic Perspectives* 29.2 (2015), pp. 213–38.
- [25] Rainer Böhme, Johanna Grzywotz, Paulina Pesch, Christian Rückert, and Christoph Safferling. *Bitcoin and Alt-Coin Crime Prevention: A Recommendation for the Regulation of Virtual Cryptocurrencies*. 2017.
- [26] Joseph Bonneau, Mike Just, and Greg Matthews. “What’s in a Name?” In: *International Conference on Financial Cryptography and Data Security*. Springer. 2010, pp. 98–113.
- [27] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua A. Kroll, and Edward W. Felten. “Mixcoin: Anonymity for Bitcoin with Accountable Mixes”. In: *Financial Cryptography and Data Security*. Ed. by Nicolas Christin and Reihaneh Safavi-Naini. Vol. 8437. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2014, pp. 486–504.
- [28] Finn Brunton and Helen Nissenbaum. *Obfuscation: A user’s guide for privacy and protest*. Mit Press, 2015.
- [29] Danton Bryans. “Bitcoin and money laundering: mining for an effective solution”. In: *Indiana Law Journal* 89 (2014), p. 441.
- [30] Coin Center. *Proposals for Clarifying Laws Around Cryptocurrency and Blockchain Technologies in Response to Requests for Feedback from Senator Pat Toomey*. 2021. URL: <https://www.coincenter.org/proposals-for-clarifying-laws-around-cryptocurrency-and-blockchain-technologies-in-response-to-requests-for-feedback-from-senator-pat-toomey/> (visited on 10/17/2021).

- [31] Chainalysis. *Chainalysis in Action: OFAC Sanctions Russian Cryptocurrency OTC Suex that Received Over \$160 million from Ransomware Attackers, Scammers, and Darknet Markets*. URL: <https://blog.chainalysis.com/reports/ofac-sanction-suex-september-2021> (visited on 10/06/2021).
- [32] Chainalysis. *The 2020 State of Crypto Crime*. 2020.
- [33] Chainalysis. *The 2021 Crypto Crime Report*. 2021. URL: <https://go.chainalysis.com/2021-Crypto-Crime-Report.html>.
- [34] Chainalysis – Blockchain analysis. 2019. URL: <https://www.chainalysis.com/> (visited on 02/19/2019).
- [35] Chainalysis Brings Privacy-Safe Compliance Solution to Cryptocurrency Exchange Bitfinex. 2019. URL: <https://www.prnewswire.com/news-releases/chainalysis-brings-privacy-safe-compliance-solution-to-cryptocurrency-exchange-bitfinex-300973941.html> (visited on 02/18/2020).
- [36] Tao-Hung Chang and Davor Svetinovic. “Improving bitcoin ownership identification using transaction patterns analysis”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50.1 (2018), pp. 9–20.
- [37] Nicolas Christin. “Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace”. In: *Proceedings of the 22nd World Wide Web Conference (WWW’13)*. Rio de Janeiro, Brazil, May 2013, pp. 213–224.
- [38] Coinbase. *Why is my cryptocurrency send delayed?* URL: <https://help.coinbase.com/en/coinbase/trading-and-funding/sending-or-receiving-cryptocurrency/why-is-my-cryptocurrency-withdrawal-delayed> (visited on 10/29/2021).
- [39] CoinMarketCap. *Total Cryptocurrency Market Cap*. URL: <https://coinmarketcap.com/charts/> (visited on 10/17/2021).

- [40] Coinranking. *Top privacy coins*. URL: <https://coinranking.com/coins/tag/privacy> (visited on 10/21/2021).
- [41] Mauro Conti, Ankit Gangwal, and Sushmita Ruj. "On the economic significance of ransomware campaigns: A Bitcoin transactions perspective". In: *Computers & Security* 79 (2018), pp. 162–189.
- [42] Wikipedia contributors. *Bitcoin — Wikipedia, The Free Encyclopedia*. 2021. URL: <https://en.wikipedia.org/w/index.php?title=Bitcoin&oldid=1050392879> (visited on 10/17/2021).
- [43] Nicolas T Courtois and Rebekah Mercer. "Stealth Address and Key Management Techniques in Blockchain Systems." In: *ICISSP*. 2017, pp. 559–566.
- [44] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, Dawn Song, and Roger Wattenhofer. "On scaling decentralized blockchains". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2016, pp. 106–125.
- [45] George Danezis and Sarah Meiklejohn. "Centrally banked cryptocurrencies". In: *Network and Distributed System Security Symposium*. 2016.
- [46] Primavera De Filippi. "The interplay between decentralization and privacy: the case of blockchain technologies". In: *Journal of Peer Production, Issue 7* (2016).
- [47] Primavera De Filippi and Aaron Wright. *Blockchain and the Law*. Harvard University Press, 2018.

- [48] Christian Decker and Roger Wattenhofer. “A fast and scalable payment network with Bitcoin duplex micropayment channels”. In: *Symposium on Self-Stabilizing Systems*. Springer. 2015, pp. 3–18.
- [49] Office of Public Affairs Department of Justice. *Deputy Attorney General Lisa O. Monaco Announces National Cryptocurrency Enforcement Team*. URL: <https://www.justice.gov/opa/pr/deputy-attorney-general-lisa-o-monaco-announces-national-cryptocurrency-enforcement-team> (visited on 10/07/2021).
- [50] Claudia Diaz, Stefaan Seys, Joris Claessens, and Bart Preneel. “Towards Measuring Anonymity”. In: *Privacy Enhancing Technologies: Second International Workshop, PET 2002 San Francisco, CA, USA, April 14–15, 2002 Revised Papers*. Ed. by Roger Dingledine and Paul Syverson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 54–68.
- [51] Nicolas Dorier. *A Simple Payjoin Proposal*. URL: <https://github.com/bitcoin/bips/blob/master/bip-0078.mediawiki> (visited on 01/20/2020).
- [52] DwarfPool. *DwarfPool XMR*. <http://dwarfpool.com/xmr>. 2017.
- [53] Cynthia Dwork. “Differential privacy”. In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2006, pp. 1–12.
- [54] EhVedadoOAnonimato. September 2015. URL: <https://web.archive.org/web/20150913010433/https://forum.getmonero.org/20/general-discussion/2361/question-on-mixin-selection>.
- [55] EhVedadoOAnonimato. September 2015. URL: <https://forum.getmonero.org/6/ideas/2372/using-time-neighbors-in-mixin-selection-in-order-to-solve-temporal-associations>.
- [56] Mark Erhardt. *An Evaluation of Coin Selection Strategies*. Master Thesis. 2016.

- [57] Úlfar Erlingsson, Vasył Pihur, and Aleksandra Korolova. “Rappor: Randomized aggregatable privacy-preserving ordinal response”. In: *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. 2014, pp. 1054–1067.
- [58] Dmitry Ermilov, Maxim Panov, and Yury Yanovich. “Automatic Bitcoin address clustering”. In: *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2017, pp. 461–466.
- [59] Ittay Eyal and Emin Gün Sirer. “Majority is not enough: Bitcoin mining is vulnerable”. In: *International conference on financial cryptography and data security*. Springer. 2014, pp. 436–454.
- [60] Giulia Fanti, Shaileshh Bojja Venkatakrishnan, Surya Bakshi, Bradley Denby, Shruti Bhargava, Andrew Miller, and Pramod Viswanath. “Dandelion++: lightweight cryptocurrency networking with formal anonymity guarantees”. In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 2.2 (2018), pp. 1–35.
- [61] Yaya J. Fanusie and Tom Robinson. “Bitcoin laundering: an analysis of illicit flows into digital currency services”. In: *Center on Sanctions & Illicit Finance memorandum, January* (2018).
- [62] Department of the Treasury Financial Crimes Enforcement Network. *Application of FinCEN’s Regulations to Persons Administering, Exchanging, or Using Virtual Currencies*. FIN-2013-G001. 2013.
- [63] Department of the Treasury Financial Crimes Enforcement Network. *Requirements for Certain Transactions Involving Convertible Virtual Currency or Digital Assets*. 85 FR 83840. 2020. URL: <https://www.federalregister.gov/documents/2020/12/23/2020-28437/requirements-for-certain->

transactions-involving-convertible-virtual-currency-or-digital-assets.

- [64] Eidgenössische Finanzmarktaufsicht FINMA. *FINMA Guidance 02/2019: Payments on the blockchain*. URL: <https://www.finma.ch/en/~media/finma/dokumente/dokumentencenter/myfinma/4dokumentation/finma-aufsichtsmitteilungen/20190826-finma-aufsichtsmitteilung-02-2019.pdf> (visited on 09/30/2021).
- [65] David Fox. *Cryptocurrencies in the Common Law of Property*. 2018.
- [66] Thomas Fox-Brewster. *WannaCry Hackers Are Using This Swiss Company To Launder \$142,000 Bitcoin Ransoms*. August 2017. URL: <https://www.forbes.com/sites/thomasbrewster/2017/08/03/wannacry-hackers-use-shapeshift-to-launder-bitcoin>.
- [67] Mark Friedenbach, BtcDrak, Nicolas Dorier, and kinoshitajona. *Relative lock-time using consensus-enforced sequence numbers*. URL: <https://github.com/bitcoin/bips/blob/master/bip-0068.mediawiki> (visited on 01/20/2020).
- [68] Ben SC Fung and Hanna Halaburda. *Central bank digital currencies: a framework for assessing why and how*. Bank of Canada Staff Discussion Paper, No. 2016-22, Bank of Canada, Ottawa. 2016.
- [69] Stefanie Garber. *Scam victims to be reimbursed under new code: has your bank signed up to protect you?* 2019. URL: <https://www.which.co.uk/news/2019/05/scam-victims-to-be-reimbursed-under-new-code-has-your-bank-signed-up-to-protect-you/> (visited on 10/29/2021).
- [70] Glassnode Studio. *Bitcoin: Lightning Network Channel Size (Median)*. URL: <https://studio.glassnode.com/metrics?a=BTC&category=Lightning&>



chartStyle=column&contractExpiration=1640908800&ema=0&m=lightning.  
ChannelSizeMedian&mAvg=0&mMedian=0&resolution=24h&zoom=all (vis-  
ited on 10/29/2021).

- [71] Steven Goldfeder, Harry Kalodner, Dillon Reisman, and Arvind Narayanan. “When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies”. In: *Proceedings on Privacy Enhancing Technologies* 2018.4 (2018), pp. 179–199.
- [72] *GraphSense Public TagPacks*. URL: <https://github.com/graphsense/graphsense-tagpacks> (visited on 04/01/2021).
- [73] *GreenAddress Bitcoin Wallet*. URL: <https://greenaddress.it/en/> (visited on 02/20/2019).
- [74] Andy Greenberg. *Monero, the Drug Dealer’s Cryptocurrency of Choice, Is on Fire*. Wired. 2017. URL: <https://www.wired.com/2017/01/monero-drug-dealers-cryptocurrency-choice-fire/>.
- [75] Johanna Grzywotz. *Virtuelle Kryptowährungen und Geldwäsche*. Vol. 15. Internetrecht und Digitale Gesellschaft. Duncker & Humblot, 2019.
- [76] JT Hamrick, Farhang Rouhi, Arghya Mukherjee, Amir Feder, Neil Gandal, Tyler Moore, and Marie Vasek. *The Economics of Cryptocurrency Pump and Dump Schemes*. CEPR Discussion Paper No. DP13404. 2018.
- [77] David A. Harding and Peter Todd. *Opt-in Full Replace-by-Fee Signaling*. URL: <https://github.com/bitcoin/bips/blob/master/bip-0125.mediawiki> (visited on 01/20/2020).
- [78] Mikkel Alexander Harlev, Haohua Sun Yin, Klaus Christian Langenheldt, Raghava Mukkamala, and Ravi Vatrapu. “Breaking bad: De-anonymising entity types on the Bitcoin blockchain using supervised machine learning”.

In: *Proceedings of the 51st Hawaii International Conference on System Sciences*. 2018.

- [79] Martin Harrigan and Christoph 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 (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld)*. IEEE. 2016, pp. 368–373.
- [80] Alyssa Hertig. *Bitcoin’s Taproot Privacy Tech Is Ready – But There’s a Catch*. 2018. URL: <https://www.coindesk.com/bitcoins-taproot-privacy-tech-is-ready-but-one-things-standing-in-the-way> (visited on 06/03/2019).
- [81] Andrew Hinkes. “Throw Away the Key, or the Key Holder? Coercive Contempt for Lost or Forgotten Cryptoasset Private Keys, or Obstinate Holders”. In: *Northwestern Journal of Technology and Intellectual Property (2019 Forthcoming)* (2019).
- [82] Abraham Hinteregger and Bernhard Haslhofer. “Short paper: An empirical analysis of Monero cross-chain traceability”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2019, pp. 150–157.
- [83] Yining Hu, Suranga Seneviratne, Kanchana Thilakarathna, Kensuke Fukuda, and Aruna Seneviratne. “Characterizing and Detecting Money Laundering Activities on the Bitcoin Network”. In: *arXiv preprint arXiv:1912.12060* (2019).
- [84] Danny Yuxing Huang, Maxwell Matthaios Aliapoulios, Vector Guo Li, Luca Invernizzi, Elie Bursztein, Kylie McRoberts, Jonathan Levin, Kirill Levchenko, Alex C Snoeren, and Damon McCoy. “Tracking ransomware end-to-end”. In: *IEEE Symposium on Security and Privacy*. IEEE. 2018, pp. 618–631.

- [85] *It's Not Personal: How Chainalysis Collects and Uses Service-Level Data*. 2019.  
URL: <https://blog.chainalysis.com/reports/service-level-data>.
- [86] Marc Jourdan, Sebastien Blandin, Laura Wynter, and Pralhad Deshpande. "Characterizing entities in the Bitcoin blockchain". In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2018, pp. 55–62.
- [87] Harry Kalodner, Malte Möser, Kevin Lee, Steven Goldfeder, Martin Plattner, Alishah Chator, and Arvind Narayanan. "Blocksci: Design and applications of a blockchain analysis platform". In: *29th USENIX Security Symposium*. 2020, pp. 2721–2738.
- [88] George Kappos, Haaron Yousaf, Mary Maller, and Sarah Meiklejohn. "An Empirical Analysis of Anonymity in Zcash". In: *USENIX Security Symposium*. 2018.
- [89] Patrick Keller, Martin Florian, and Rainer Böhme. "Collaborative Deanonimization". In: *Workshop on Coordination of Decentralized Finance (CoDecFin)*. 2021.
- [90] Brandon Kochkodin. 2021. URL: <https://www.bloomberg.com/news/articles/2021-06-18/venture-capital-makes-a-record-17-billion-bet-on-crypto-world> (visited on 10/17/2021).
- [91] Sergej Kotliar. *Another view. This is what a Coinbase outage looks like on the Bitcoin network*. URL: <https://twitter.com/ziggamon/status/951700118830432257> (visited on 04/17/2019).
- [92] Amrit Kumar, Clément Fischer, Shruti Tople, and Prateek Saxena. "A Traceability Analysis of Monero's Blockchain". In: *Computer Security – ESORICS 2017: 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II*. Ed. by Simon N. Foley, Dieter

- Gollmann, and Einar Snekkenes. Springer International Publishing, 2017, pp. 153–173.
- [93] Ron Lavi, Or Sattath, and Aviv Zohar. “Redesigning Bitcoin’s fee market”. In: *The world wide web conference*. 2019, pp. 2950–2956.
  - [94] Yu-Jing Lin, Po-Wei Wu, Cheng-Han Hsu, I-Ping Tu, and Shih-wei Liao. “An evaluation of Bitcoin address classification based on transaction history summarization”. In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. IEEE. 2019, pp. 302–310.
  - [95] Eric Lombrozo, Johnson Lau, and Pieter Wuille. *Segregated Witness (Consensus layer)*. URL: <https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki> (visited on 01/20/2020).
  - [96] Adam Mackenzie, Surae Noether, and Monero Core Team. *MRL-0004: Improving Obfuscation in the CryptoNote Protocol*. January 2015. URL: <https://lab.getmonero.org/pubs/MRL-0004.pdf>.
  - [97] Damiano Di Francesco Maesa, Andrea Marino, and Laura Ricci. “Data-driven analysis of bitcoin properties: exploiting the users graph”. In: *International Journal of Data Science and Analytics* 6.1 (2018), pp. 63–80.
  - [98] James L Massey. “Guessing and Entropy”. In: *Proceedings of 1994 IEEE International Symposium on Information Theory*. IEEE. 1994, p. 204.
  - [99] Gregory Maxwell. *CoinJoin: Bitcoin Privacy for the Real World*. 2013. URL: <https://bitcointalk.org/index.php?topic=279249.0> (visited on 06/30/2019).
  - [100] Patrick McCorry, Malte Möser, and Syed Taha Ali. “Why Preventing a Cryptocurrency Exchange Heist Isn’t Good Enough”. In: *Security Protocols XXVI: 26th International Workshop, Cambridge, UK, March 19–21, 2018, Revised Selected Papers*. Vol. 11286. Springer. 2018, pp. 225–233.

- [101] Patrick McCorry, Malte Möser, Siamak F. Shahandasti, and Feng Hao. “Towards Bitcoin Payment Networks”. In: *Australasian Conference on Information Security and Privacy*. Springer. 2016, pp. 57–76.
- [102] Sarah Meiklejohn and Claudio Orlandi. “Privacy-enhancing overlays in Bitcoin”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2015, pp. 127–141.
- [103] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M Voelker, and Stefan Savage. “A fistful of bitcoins: characterizing payments among men with no names”. In: *Internet Measurement Conference*. ACM. 2013, pp. 127–140.
- [104] Peter E. Meltzer. “Keeping Drug Money from Reaching the Wash Cycle: A Guide to the Bank Secrecy Act”. In: *Banking Law Journal* 108 (1991), p. 230.
- [105] Ian Miers, Christina Garman, Matthew Green, and Aviel D Rubin. “Zerocoin: Anonymous distributed e-cash from Bitcoin”. In: *IEEE Symposium on Security and Privacy*. IEEE. 2013, pp. 397–411.
- [106] MinerGate. *MinerGate*. <https://minergate.com/>. 2017.
- [107] MoneroHash. *MoneroHash*. <https://monerohash.com/#network>. 2017.
- [108] Malte Möser. *Testchain Generator*. 2019. URL: <https://github.com/citp/testchain-generator> (visited on 09/28/2021).
- [109] Malte Möser and Rainer Böhme. “Anonymous alone? Measuring Bitcoin’s second-generation anonymization techniques”. In: *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE. 2017, pp. 32–41.

- [110] Malte Möser and Rainer Böhme. “The price of anonymity: empirical evidence from a market for Bitcoin anonymization”. In: *Journal of Cybersecurity* 3.2 (2017), pp. 127–135.
- [111] Malte Möser and Rainer Böhme. “Trends, tips, tolls: A longitudinal study of Bitcoin transaction fees”. In: *Financial Cryptography and Data Security, 2nd Workshop on BITCOIN Research*. Springer. 2015, pp. 19–33.
- [112] Malte Möser, Rainer Böhme, and Dominic Breuker. “An inquiry into money laundering tools in the Bitcoin ecosystem”. In: *eCrime Researchers Summit (eCRS), 2013*. IEEE. 2013, pp. 1–14.
- [113] Malte Möser, Rainer Böhme, and Dominic Breuker. “Towards Risk Scoring of Bitcoin Transactions”. In: *Financial Cryptography and Data Security, 1st Workshop on BITCOIN Research*. Ed. by Rainer Böhme, Michael Brenner, Tyler Moore, and Matthew Smith. Vol. 8438. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2014, pp. 16–32.
- [114] Malte Möser, Ittay Eyal, and Emin Gün Sirer. “Bitcoin covenants”. In: *Financial Cryptography and Data Security, 3rd Workshop on BITCOIN Research*. Springer. 2016, pp. 126–141.
- [115] Malte Möser and Arvind Narayanan. *Effective cryptocurrency regulation through blacklisting*. Preprint. 2019.
- [116] Malte Möser and Arvind Narayanan. “Resurrecting Address Clustering in Bitcoin”. In: *Financial Cryptography and Data Security*. 2022, forthcoming.
- [117] Malte Möser, Kyle Soska, Ethan Heilman, Kevin Lee, Henry Heffan, Shashvat Srivastava, Kyle Hogan, Jason Hennessey, Andrew Miller, Arvind Narayanan, and Nicolas Christin. “An Empirical Analysis of Traceability in the Monero

- Blockchain". In: *Proceedings on Privacy Enhancing Technologies* 2018.3 (2018), pp. 143–163.
- [118] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. 2008. URL: <https://bitcoin.org/bitcoin.pdf> (visited on 02/10/2017).
  - [119] Arvind Narayanan and Malte Möser. "Obfuscation in Bitcoin: Techniques and politics". In: *International Workshop on Obfuscation: Science, Technology, and Theory*. 2017.
  - [120] Danny Nelson. *Bittrex Adopts Chainalysis KYT Software to Flag Suspicious Activity*. 2019. URL: <https://www.coindesk.com/bittrex-chainalysis-kyt> (visited on 02/18/2020).
  - [121] Jonas David Nick. "Data-driven de-anonymization in Bitcoin". MA thesis. ETH-Zürich, 2015.
  - [122] Shen Noether, Adam Mackenzie, and the Monero Research Lab. "Ring Confidential Transactions". In: *Ledger* 1.0 (2016), pp. 1–18.
  - [123] Surae Noether, Sarang Noether, and Adam Mackenzie. *MRL-0001: A Note on Chain Reactions in Traceability in CryptoNote 2.0*. September 2014. URL: <https://lab.getmonero.org/pubs/MRL-0001.pdf>.
  - [124] OFAC License Application Page. URL: <https://www.treasury.gov/resource-center/sanctions/Pages/licensing.aspx> (visited on 09/17/2019).
  - [125] Masarah Paquet-Clouston, Bernhard Haslhofer, and Benoît Dupont. "Ransomware payments in the Bitcoin ecosystem". In: *Journal of Cybersecurity* 5.1 (2019).
  - [126] Masarah Paquet-Clouston, Matteo Romiti, Bernhard Haslhofer, and Thomas Charvat. "Spams meet Cryptocurrencies: Sextortion in the Bitcoin Ecosystem". In: *ACM conference on advances in financial technologies (AFT'19)*. 2019.

- [127] Francesco Parino, Mariano G Beiró, and Laetitia Gauvin. “Analysis of the Bitcoin blockchain: socio-economic factors behind the adoption”. In: *EPJ Data Science* 7.1 (2018), p. 38.
- [128] Committee on Payments and Bank for International Settlements Market Infrastructures Markets Committee. *Central bank digital currencies*. 2018.
- [129] Andreas Pfitzmann and Marit Köhntopp. “Anonymity, Unobservability, and Pseudonymity – a Proposal for Terminology”. In: *Designing Privacy Enhancing Technologies*. Ed. by Hannes Federrath. Vol. 2009. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2001, pp. 1–9.
- [130] Andrew Poelstra. *Mimblewimble*. 2016. URL: <https://download.wpsoftware.net/bitcoin/wizardry/mimblewimble.pdf> (visited on 02/28/2020).
- [131] Joseph Poon and Thaddeus Dryja. *The Bitcoin lightning network: Scalable off-chain instant payments*. 2016. URL: <https://lightning.network/lightning-network-paper.pdf> (visited on 01/14/2019).
- [132] Nathaniel Popper. *After the Bust, Are Bitcoins More Like Tulip Mania or the Internet?* URL: <https://www.nytimes.com/2019/04/23/technology/bitcoin-tulip-mania-internet.html> (visited on 05/02/2019).
- [133] Rebecca S. Portnoff, Danny Yuxing Huang, Periwinkle Doerfler, Sadia Afroz, and Damon McCoy. “Backpage and Bitcoin: Uncovering human traffickers”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2017, pp. 1595–1604.
- [134] *Privacy - Bitcoin Wiki*. URL: <https://en.bitcoin.it/Privacy> (visited on 12/15/2020).



- [135] Fergal Reid and Martin Harrigan. “An Analysis of Anonymity in the Bitcoin System”. In: *Security and Privacy in Social Networks*. Springer, 2013, pp. 197–223.
- [136] Ronald L Rivest, Adi Shamir, and Yael Tauman. “How to leak a secret”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2001, pp. 552–565.
- [137] Dorit Ron and Adi Shamir. “Quantitative analysis of the full Bitcoin transaction graph”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2013, pp. 6–24.
- [138] Christian Rückert. “Cryptocurrencies and fundamental rights”. In: *Journal of Cybersecurity* 5.1 (2019).
- [139] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. “CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin”. In: *ESORICS’14. Proceedings of the 19th European Symposium on Research in Computer Security*. Vol. 8713. Lecture Notes in Computer Science. Berlin Heidelberg: Springer, 2014, pp. 345–364.
- [140] Jürgen E Schatzmann and Bernhard Haslhofer. “Bitcoin Trading is Irrational! An Analysis of the Disposition Effect in Bitcoin”. In: *arXiv preprint arXiv:2010.12415* (2020).
- [141] Tom Schoenberg and Matt Robinson. *Bitcoin ATMs May Be Used to Launder Money*. URL: <https://www.bloomberg.com/features/2018-bitcoin-atm-money-laundering/> (visited on 02/08/2019).
- [142] *SegWit FAQ*. URL: <https://help.coinbase.com/en/pro/getting-started/general-crypto-education/segwit-faq> (visited on 04/27/2021).

- [143] Andrei Serjantov and George Danezis. “Towards an Information Theoretic Metric for Anonymity”. In: *Privacy Enhancing Technologies: Second International Workshop, PET 2002 San Francisco, CA, USA, April 14–15, 2002 Revised Papers*. Ed. by Roger Dingledine and Paul Syverson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 41–53.
- [144] Claude E Shannon. “A Mathematical Theory of Communication”. In: *Bell System Technical Journal* 27 (4 1948), pp. 623–666.
- [145] James Smith. *Elliptic and Financial Privacy*. 2019. URL: <https://www.elliptic.co/our-thinking/elliptic-financial-privacy> (visited on 04/11/2019).
- [146] Kyle Soska and Nicolas Christin. “Measuring the Longitudinal Evolution of the Online Anonymous Marketplace Ecosystem”. In: *Proceedings of the 24th USENIX Security Symposium (USENIX Security’15)*. Washington, DC, August 2015, pp. 33–48.
- [147] The Internet Archive. *Wayback machine*. <https://archive.org/web/>. 1999.
- [148] Tin Tironsakkul, Manuel Maarek, Andrea Eross, and Mike Just. “Probing the mystery of cryptocurrency theft: An investigation into methods for cryptocurrency tainting analysis”. In: *Available at SSRN* 3403656 (2019).
- [149] Peter Todd. *Discourage fee sniping with nLockTime #2340*. 2014. URL: <https://github.com/bitcoin/bitcoin/pull/2340> (visited on 08/11/2015).
- [150] Kentaroh Toyoda, Tomoaki Ohtsuki, and P Takis Mathiopoulos. “Multi-class Bitcoin-enabled service identification based on transaction history summarization”. 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)*. IEEE. 2018, pp. 1153–1160.

- [151] U.S. Department of the Treasury. *Treasury Designates Iran-Based Financial Facilitators of Malicious Cyber Activity and for the First Time Identifies Associated Digital Currency Addresses*. 2018. URL: <https://home.treasury.gov/news/press-releases/sm556> (visited on 01/11/2019).
- [152] U.S. Department of the Treasury. *Treasury Takes Robust Actions to Counter Ransomware*. 2021. URL: <https://home.treasury.gov/news/press-releases/jy0364> (visited on 09/30/2021).
- [153] *United States of America vs. Alexandre Cazes. Verified complaint for forfeiture In Rem*. July 2017. URL: <https://www.justice.gov/opa/press-release/file/982821/download>.
- [154] Peter Van Valkenburgh. *Comments to the Financial Action Task Force on the March 2021 Draft updated Guidance for a risk-based approach to virtual assets and VASPs*. URL: <https://www.coincenter.org/comments-to-the-financial-action-task-force-on-the-march-2021-draft-updated-guidance-for-a-risk-based-approach-to-virtual-assets-and-vasps/> (visited on 09/30/2021).
- [155] Peter Van Valkenburgh. *Electronic Cash, Decentralized Exchange, and the Constitution*. Coin Center Report. 2019.
- [156] Peter Van Valkenburgh. *FinCEN's new cryptocurrency guidance matches Coin Center recommendations*. URL: <https://www.coincenter.org/fincens-new-cryptocurrency-guidance-matches-coin-center-recommendations/> (visited on 10/29/2021).
- [157] Nicolas Van Saberhagen. *CryptoNote v2.0*. 2013. URL: <https://cryptonote.org/whitepaper.pdf> (visited on 01/14/2018).

- [158] Marie Vasek and Tyler Moore. “Analyzing the Bitcoin Ponzi scheme ecosystem”. In: *The 5th Workshop on Bitcoin and Blockchain Research*. 2018.
- [159] Marie Vasek and Tyler 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*. Springer. 2015, pp. 44–61.
- [160] Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. “Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics”. In: *arXiv preprint arXiv:1908.02591* (2019).
- [161] wh1sks. *The Shadow Brokers may have received up to 1500 Monero (\$66,000) from their June “Monthly Dump Service”*. July 2017. URL: <https://steemit.com/shadowbrokers/@wh1sks/theshadowbrokers-may-have-received-up-to-1500-monero-usd66-000-from-their-june-monthly-dump-service>.
- [162] Gavin Wood. *Ethereum: A secure decentralised generalised transaction ledger*. Ethereum project yellow paper. 2014.
- [163] Karl Wüst, Kari Kostiaainen, Vedran Capkun, and Srdjan Capkun. “PRCash: Fast, Private and Regulated Transactions for Digital Currencies”. In: *Financial Cryptography and Data Security*. 2019.
- [164] Zuoxia Yu, Man Ho Au, Jiangshan Yu, Rupeng Yang, Qiuliang Xu, and Wang Fat Lau. “New empirical traceability analysis of CryptoNote-style blockchains”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2019, pp. 133–149.
- [165] Yuhang Zhang, Jun Wang, and Jie Luo. “Heuristic-Based Address Clustering in Bitcoin”. In: *IEEE Access* 8 (2020), pp. 210582–210591.



## Appendix to Chapter 3

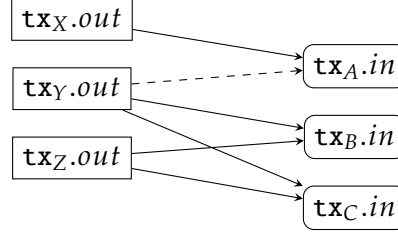
### A.1. Deducibility Attack as a SAT problem

A limitation of the query-based algorithm is that it can only deduce a spend when previous iterations ruled out all other mixins. However, more complex “puzzles” are conceivable where we cannot rule out mixins because they have been provably been spent, but because they must have been spent by another input (cf. Figure A.1). We formalize this combinatorial puzzle as a SAT problem and solve it using the Sat4j SAT solver.

A SAT problem  $p$  is a boolean formula containing  $n$  propositional variables that yield a mapping  $\delta$  between inputs and outputs. Each variable  $r_{xy} \in \delta$  denotes a single reference from an input  $x$  to an output  $y$ . Let  $\delta(\hat{x})$  give us all existing  $y$  values for  $r_{\hat{x}y}$ , and vice versa. We can then specify restrictions on the possible relationships between inputs and outputs.

- Every input can spend any of the referenced outputs:

$$\bigwedge_{x=1}^{|X|} \bigvee_{y \in \delta(x)} r_{xy}$$



**Figure A.1.:** Since  $\text{tx}_Y.out$  must have been spent by either  $\text{tx}_B.in$  or  $\text{tx}_C.in$ ,  $\text{tx}_X.out$  must have been spent by  $\text{tx}_A.in$ .

- An input can only spend a single output:

$$\bigwedge_{x=1}^{|X|} \bigwedge_{i=1}^{|\delta(x)|-1} \bigwedge_{j=i+1}^{|\delta(x)|} (\neg r_{x\delta(x)_i} \vee \neg r_{x\delta(x)_j})$$

- Similarly, every output can only be spent by a single input:

$$\bigwedge_{y=1}^{|Y|} \bigwedge_{i=1}^{|\delta(y)|-1} \bigwedge_{j=i+1}^{|\delta(y)|} (\neg r_{\delta(y)_i y} \vee \neg r_{\delta(y)_j y})$$

Since inputs can only spend outputs of the same value, the SAT problem can be solved individually for each denomination. Assignments of propositional variables that are true in all solutions of  $p$  correspond to true spends, as it reflects the only possible way to spend an output (e.g.,  $\text{tx}_A.in$  always spends  $\text{tx}_X.out$  in Figure A.1).

Solving the SAT problem using Sat4j yields an additional 5149 deanonymized spends across 1157 different denominations.

## A.2. An Analysis of Bytecoin

The cryptocurrency Bytecoin was an early implementation of the Cryptonote protocol. As Monero is based upon Bytecoin's codebase, Bytecoin's mixin sampling procedure shares the same weaknesses. We run the mixin sudoku algorithm on

transaction data extracted from the Bytecoin blockchain and show the results in Table A.1. Overall, we are able to deduce the real spent for 29% of all inputs with more than one mixin. Of those that include only 1 mixin, we can deduce 56% of inputs. We attribute the lower total success rate to a discrepancy between the number inputs to outputs in Bytecoin, suggesting that there exist a lot of unspent outputs from which significantly fewer inputs can choose.

In Table A.2 we show the percentage of inputs where guessing the most recent output yields the true spend. With an accuracy of 97.54% the Guess-Newest heuristic proves to be very effective.

**Table A.1.:** Bytecoin transaction inputs (with 1 or more mixins, at least 1000 TXOs available) where the real input can be deduced.

	Total	Deducible	(%)
1 mixins	4 192 272	2 338 746	(55.79)
2 mixins	813 375	264 610	(32.53)
3 mixins	1 243 428	207 540	(16.69)
4 mixins	2 450 891	249 618	(10.18)
5 mixins	405 140	25 666	(6.34)
6 mixins	1 250 627	51 755	(4.14)
7 mixins	158 753	4198	(2.64)
8 mixins	76 530	539	(0.70)
9 mixins	62 714	226	(0.36)
10+ mixins	204 725	197	(0.10)
Total	10 858 455	3 143 095	(28.95)

**Table A.2.:** Percentage of deducible Bytecoin transaction inputs where the real input is the “newest” input.

	Deducible	Newest	(%)
1 mixins	2 338 746	2 311 078	(98.82)
2 mixins	264 610	249 000	(94.10)
3 mixins	207 540	193 453	(93.21)
4 mixins	249 618	245 236	(98.24)
5 mixins	25 666	19 027	(74.13)
6 mixins	51 755	44 243	(85.49)
7 mixins	4198	3011	(71.72)
8 mixins	539	378	(70.13)
9 mixins	226	149	(65.93)
10+ mixins	197	128	(64.97)
Total	3 143 095	3 065 703	(97.54)



# B

## Appendix to Chapter 4

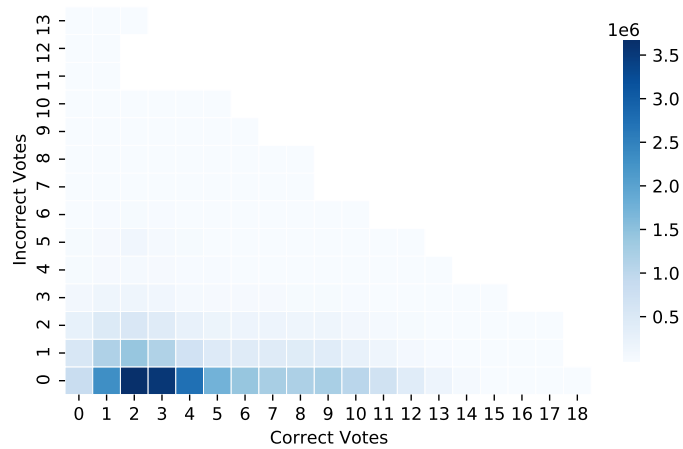
### B.1. Protocol characteristics used for fingerprinting

- Input/output count: the number of inputs and/or outputs may indicate a wallet software's behavior of creating transactions. While the number of inputs depends on the UTXOs available to the user, some commonly occurring patterns such as peeling chains ([103, 112]) have consistent input and output counts.
- Version: BIP 68 [67] introduced relative timelocks for transactions, which requires transaction to set the transaction version to 2.
- Locktime: Transactions can set a timelock such that they are valid only after the tip of the chain has passed a specific block height or timestamp. Some clients (e.g., Bitcoin Core) produce timelocked transactions by default to prevent fee-sniping [149].
- Replace-by-fee (RBF): Transactions opting into the replace-by-fee policy can be replaced by a similar transaction paying a higher fee [77].
- SegWit: Segregated witness [95] is a protocol update that enabled storing the inputs' signatures outside of the transaction, thereby increasing available space in blocks. As the upgrade is backwards-comptable, not all wallets

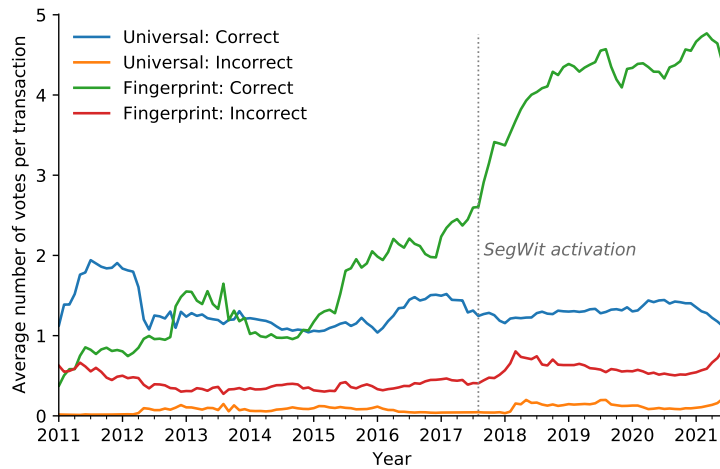
produce SegWit transactions. A wallet might also be able to produce SegWit transaction, but may be required to use non-SegWit serialization if none of the inputs use SegWit. We call this behavior SegWit-conform.

- Ordered inputs/outputs: BIP 69 [10] defines non-binding rules (i.e. not enforced by the consensus mechanism) for lexicographically sorting inputs and outputs in a transaction. (A limitation of our implementation is that it does not compare the raw `scriptPubKey` in case the output values are equal, as they are not available in BlockSci).
- Zero-conf: Bitcoin user's are encouraged to wait for up to six confirmations (about an hour) before accepting a payment, as there is a risk that funds might be double-spent. A transaction spending inputs without any confirmations indicates willingness to accept the double-spending risk, which could be specific to certain intermediaries.
- Transaction fee: Bitcoin users pay transaction fees for their transactions to be included into the blockchain by miners. Some clients may pay the same exact fee (either absolute, or relative to the transaction's size) for every transaction.
- Multisignature: Multisignature scripts allow to specify a list of public keys and a threshold  $m$  such that the redeemer must provide valid signatures for  $m$  out of  $n$  of these keys. They aren't typically used by normal end-user wallets.
- Address types: Bitcoin Core defines a number of standardized output scripts types including Pay-to-Pubkey-Hash (P2PKH), Pay-to-Script-Hash (P2SH) as well as their respective SegWit variants (P2WPKH and P2WSH). Often, a wallet consistently uses a specific address type. (Compared to the normal address type heuristic, the fingerprint checks for overlap with the address types of all inputs of the spending transaction).

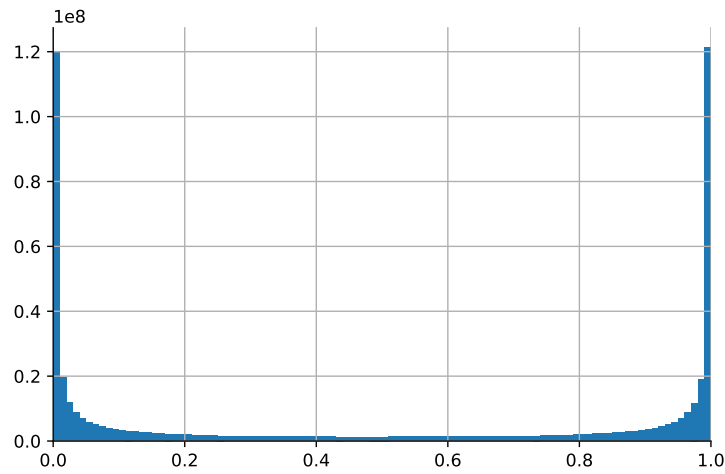
## B.2. Additional plots and tables



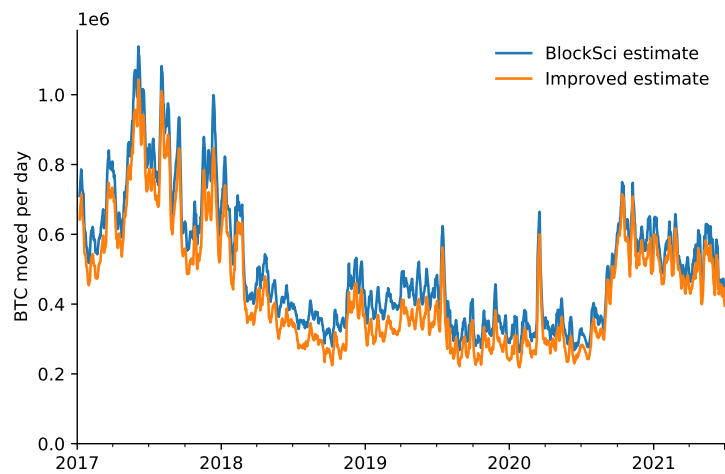
**Figure B.1.:** Number of votes from heuristics (with compressed power-of-ten heuristic)



**Figure B.2.:** Average number of correct and incorrect votes per transaction and type of heuristic, over time (with compressed power-of-ten heuristic)



**Figure B.3.:** Probabilities returned by the random forest classifier for standard transactions with unknown change



**Figure B.4.:** Revised estimate of velocity

**Table B.1.:** Increase in outgoing transaction volumes of darknet markets to exchanges using the base clustering (before) and our enhanced clustering (after).

Tag	# Outputs			Volume (BTC)		
	Before	After	(%)	Before	After	(%)
abraxasmarket	9202	11 373	23.59	21 925	23 368	6.58
agoramarket	82 218	108 026	31.39	158 360	170 970	7.96
alphabaymarket	25 646	33 052	28.88	35 496	41 573	17.12
babylonmarket	422	616	45.97	222	283	27.13
blackbankmarket	3052	3987	30.64	8292	9245	11.49
blueskymarket	6581	9919	50.72	2520	3333	32.3
cannabisroadmarket	48	76	58.33	6	7	25.15
doctordmarket	414	621	50.0	224	277	23.92
evolutionmarket	20 572	53 335	159.26	49 891	84 637	69.64
middleearthmarket	5078	5410	6.54	11 793	12 021	1.93
nucleusmarket	23 840	26 760	12.25	45 265	47 006	3.85
pandoraopenmarket	8246	10 399	26.11	8708	9461	8.64
sheepmarket	4487	5235	16.67	12 104	13 309	9.96
silkroad2market	32 693	38 251	17.0	47 292	49 559	4.79
silkroadmarket	15 152	20 369	34.43	421 741	472 282	11.98
<i>Total</i>	237 651	327 429	37.78	823 839	937 330	13.78

## B.3. Further insights and technical details

### B.3.1. Filtering the ground truth data set

Selecting transactions with two outputs, no OP\_RETURN outputs, where no input address has been directly reused in the outputs and where at least one output is in the same base cluster as the inputs yields a total of 53.41 million transactions. We first exclude 1.08 million transactions with unspent outputs, as our subsequent analyses rely upon the spending transactions being known.

**Transactions with two change candidates.** Out of the 52.33 million transactions with at least one change candidate, for 0.97 million transactions *both* outputs are in the same base cluster. This can happen when a user transfers funds to an address in their own wallet, an online service restructures their funds, or cluster collapse leads

**Table B.2.:** Characteristics of clusterings created with the Meiklejohn heuristic in comparison to our constrained clustering

Characteristic	Meiklejohn heuristics	
	Local	Global
Coverage	47.7 %	54.4 %
Largest cluster		
• # addresses	281.3 M	298.4 M
• # transactions	124.8 M	133.1 M
Address pairs clustered		
• Neither	90.80 %	87.03 %
• Ours only	0.06 %	0.05 %
• Meiklejohn only	8.97 %	12.66 %
• Both	0.17 %	0.26 %
Predictions		
• Total	147.6 M	168.5 M
• Overlapping	71.2 M	81.1 M
Conflicting predictions		
• Count	1.3 M	1.9 M
• Difference in BTC	2.1 M	4.1 M
• Difference in USD	16.6 B	38.7 B

to merging of both outputs' addresses. In a first step, we exclude all transactions with two change candidates.

However, it is possible that there are yet unidentified transactions where both outputs do belong to the same entity. This should occur only in rare cases, but there may be specific intermediaries that create such transactions more frequently. We therefore exclude *all* transactions from base clusters where more than 10 % of transactions exhibit such behavior. This removes an additional 480 845 transactions in 9967 base clusters from our ground truth.

**Potential false positives.** A risk of using the base clustering to extract ground truth is that the multi-input heuristic could already have produced false positives. For example, if a user Alice makes a payment to merchant Bob and their wallet addresses are incorrectly clustered together, her spend output would appear to be the change.

To this end, we first remove 366 926 transactions belonging to the Mt. Gox supercluster (cf. [79]). Next, we spot-check our base clustering against the website WalletExplorer.<sup>1</sup> For the 100 largest base clusters in our ground truth we select 25 addresses at random and collect the tag (which is either explicitly named or pseudo-random) that WalletExplorer assigns to the address. In five instances, the addresses yield multiple tags. Four of these return only additional pseudo-random tags, which upon manual inspection we believe to be the result of a heuristic to not link addresses in transactions with large numbers of inputs. Only one base cluster contains addresses with two different named tags: "LocalBitcoins.com-old" and "AnxPro.com". This could be a result of cluster collapse, or an instance of mislabeling on the side of WalletExplorer. We remove the 87 947 transactions from

---

<sup>1</sup><https://walletexplorer.com>

this base cluster from our ground truth. Overall, this check gives us some confidence that our base clustering does not already include wide-spread cluster collapse.

**Change address reuse.** Our initial selection removed transactions where the change address appeared in an input of the transaction. Yet, we find many instances where the change address did not appear in the inputs but had been seen before. For example, a base cluster labeled by WalletExplorer as the gambling service “SatoshiDice”, contains 5.77 million transactions that use only 50 different change addresses. Similarly, there are 1.27 million transactions from a base cluster tagged as “LuckyB.it” that all use a single change address.<sup>2</sup> In many of these cases, the change address could have already been revealed (before the transaction took place) through the multi-input heuristic.

If the change is known at the time the transaction is created, applying change heuristics is unnecessary. In contrast, whenever a transaction uses a fresh address for change, it cannot possibly be revealed as the change at the time the transaction was created. With this intuition, we remove transactions with change addresses that were not freshly generated if, at the time they were included in the blockchain, the change had already been revealed by the multi-input heuristic. This removes a total of 15.17 million transactions (90.80 % of transactions with reused change addresses). Table B.3 provides an overview of whether the change and spend addresses are fresh in our ground truth data.

### B.3.2. Additional details on classification

**Encoding.** Due to the large size of the data set, we forgo one-hot encoding and instead use the following ordinal encoding for the heuristics:

---

<sup>2</sup>1NxBCFQwejSZbQfWcYNwgqML5wWoE3rK4



**Table B.3.:** Number of transactions (in million) in our ground truth data set with fresh or reused spend and change outputs.

<i>Change</i>	<i>Spend</i>		Total
	Reused	Fresh	
Reused	0.73	0.81	1.54
Fresh	19.38	14.34	33.71
Total	20.11	15.15	35.26

**1** the heuristic votes for the output

**0** the heuristic votes neither for nor against the output

**-1** the heuristic votes against the output

**AUC scores for different classifiers** Initial runs were done for a baseline comparison without hyperparameter tuning. We note that our encoding may not be ideal for some classifiers, specifically for attributes that allow to subdivide behavior between different clients and epochs. This is a major limitation of linear models and one the primary reasons we choose a random forest model, as it is able to split the data set along those attributes.

- Logistic regression (l2 penalty): 0.9933
- Support Vector Machine (linear kernel): 0.9931
- Adaboost: 0.9926
- Random forest: 0.9982

**Hyperparameter tuning for the random forest classifier.** Our hyperparameter grid search returns the following parameters:

- All heuristics / full model

- max\_features: 7
  - min\_samples\_leaf: 10
  - min\_samples\_split: 20
- Universal heuristics only
  - max\_features: 6
  - min\_samples\_leaf: 10
  - min\_samples\_split: 20



## Appendix to Chapter 5

### C.1. Blackmail Scam Addresses

The following Bitcoin addresses were extracted from blackmail scam emails (often referred to as “sextortion”). The outputs sent to these addresses make up the “Blackmail” dataset used in Section 5.5.4.

- 16oE4aRiJQvwMvbdfkx6kX8Wg9GLA1uFnz
- 1Agq5LZhY2UgcHoknVU7rZeE1x4gMuJTvf
- 13bpFXeCW6inWQSSgkVKj9rmemtkNvNoD3
- 1GPjEBZvfFRAGwCorR98upF5ByDu2Cq9Ha
- 18921mhD7bedjrgQuDmNh3oFuMQiUry1X7
- 1B3SBdx6ZqhBjUuYTzoMZq4a6Kvk4psKfM
- 19qL8vdRtk5xJcGNV3WruuSyitVfSAy7f
- 1KzMDhZLokkNd1kcxs2mgwXm97pVvnfRBC
- 1P7bLeCJywaaDRQpT7iwb4qzUHa4CpRFyP
- 1971pHPgLaTmuYtoH4BSGSfFMZaA jotium

- 17EuB8AmyBm81FgCovdr6huCCoSzv2S7nP
- 1WLEChY6S7S97m5voZZtbQcwiEYeSNsja
- 13phdoBirraAtFXKWJQ9HgTYX9b7C2MqXPE
- 1G1qFoadiDxa7zTvppSMJhJi63tNUL3cy7
- 18Pt4B7Rz7Wf491FGQHPsfDeKRqnkyrMo6
- 3ER3byGWbnqgxN4C5amtCXHEXPgGaUWsBd
- 19UW5P6PGDA3SWehh8UMGQQC3ezBzN1mDE
- 1H1K8MfLEJgjCCfDEkTJmv9GJjD3XzEFGR
- 39eaJ2Fxbm4KWVu26BzaEH465aK4yrbuzH
- 38KxdSNjge7hdy7zWZuRRC4hN4krQrrA5b
- 12s4cfoNTzT68gSdxLjmSRT3qdvaqwDWNz
- 1GoWy5yMzh3XXBiYxLU9tKCBMgibpznGio
- 15mWFjVymAdqimVim2f1UgX6oSD4TYeGLE
- 1ELgYTbMLmw9vaHADfZmMcKVMWCNmRH8S2
- 1Gm6q6M2TfL4yrL3TnvsUZZ1C5k6wrvBbX
- 34vhBcVUYwfNYjupWHrgef1zogaRZaSFU6
- 3CEX39owi6EeUAoM4ENyYdGhLZj7nAwuYH
- 1DjuN5PM9VLXCeqYrb9nxzpQ8rb2hXZiEt

**Table C.1.:** Number of outputs tainted by FIFO, and total number of tainted chunks despite merging adjacent chunks, with different datasets on 06/31/2020

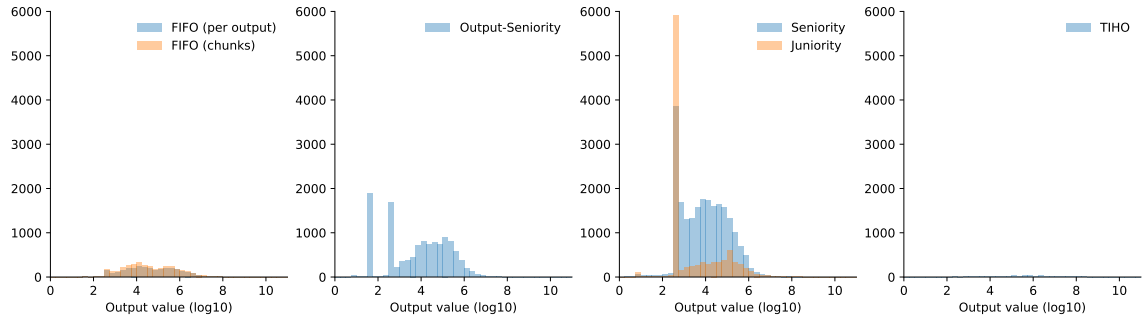
Dataset	# tainted outputs	# tainted chunks
Blackmail	2821	3428
OFAC	270 537	764 601
Ransomware	576 694	2 221 476
Random outputs	24 813	44 472

**Table C.2.:** Average number of outputs tainted by applying the heuristic to each individual output in the dataset separately and propagating taint throughout the next 2018 blocks, starting from the output's block height (2 weeks)

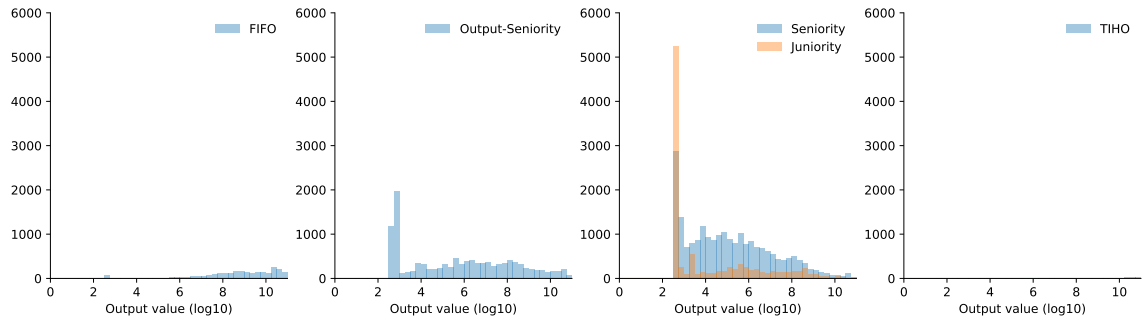
Dataset/Measure	FIFO	Output-Seniority	Seniority	Juniority	TIHO
<i>Blackmail (n = 273)</i>					
median	1.0	1.0	1.0	1.0	1.0
mean	2.4	5.2	6.6	3.66	1.25
<i>OFAC (n = 3305)</i>					
median	5.0	10.0	14.0	32.0	1.0
mean	27.68	45.43	64.45	238.14	9.39
<i>Ransomware (n = 16777)</i>					
median	2.0	10.0	14.0	4.0	2.0
mean	9.4	26.99	40.84	35.69	2.45
<i>Random outputs (n = 100)</i>					
median	4.0	8.5	12.5	6.5	2.0
mean	116.85	154.05	197.93	111.24	51.85

## C.2. Extended Taint Analysis

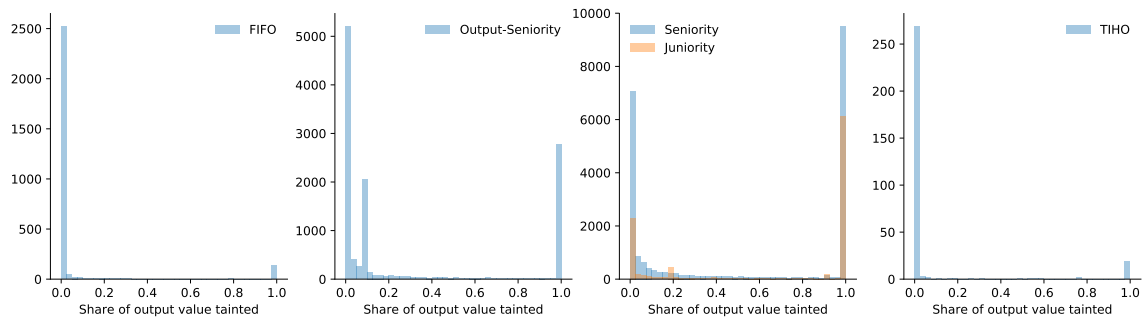
The following pages contain the results of our empirical analysis (see Section 5.5.4) of applying different taint policies to three different exemplary data sets.



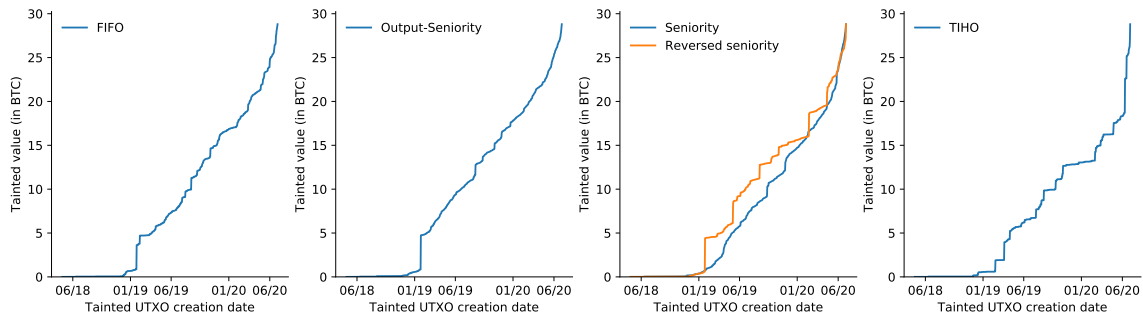
**(a)** Distribution of tainted value in all tainted outputs



**(b)** Distribution of (full) output value of all tainted outputs

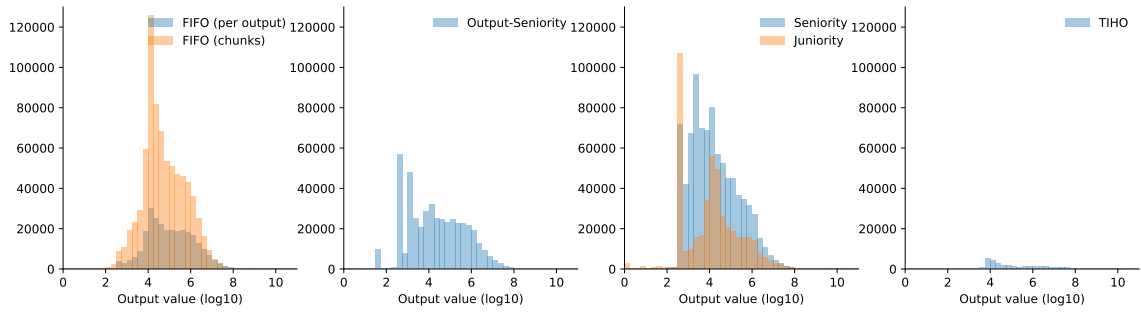


**(c)** Share of output value tainted

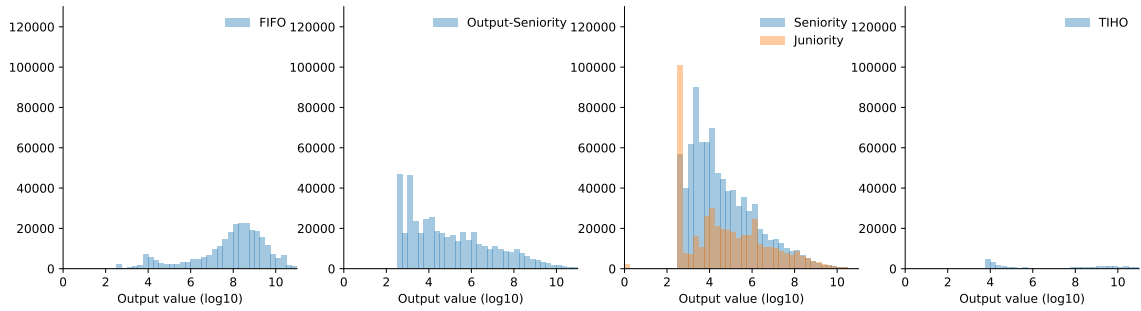


**(d)** Creation time of tainted unspent transaction outputs

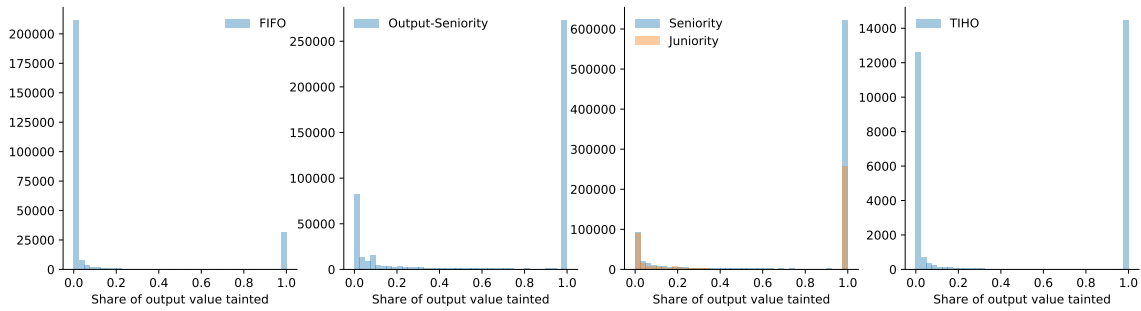
**Figure C.1.:** Analysis of the Blackmail data set



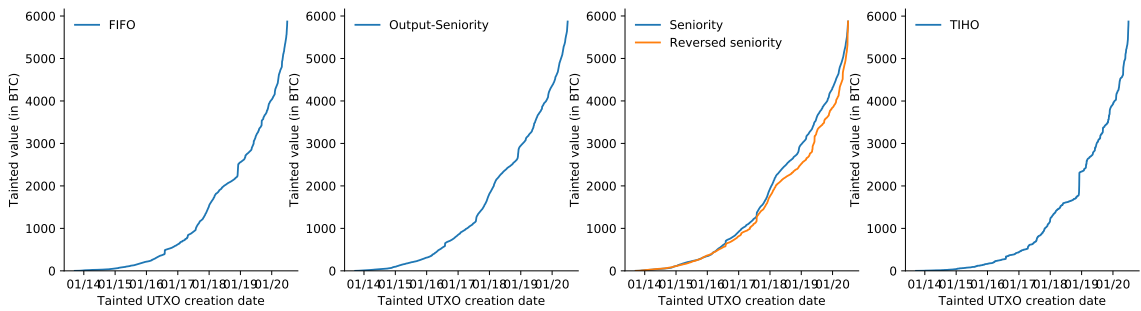
**(a)** Distribution of tainted value in all tainted outputs



**(b)** Distribution of (full) output value of all tainted outputs

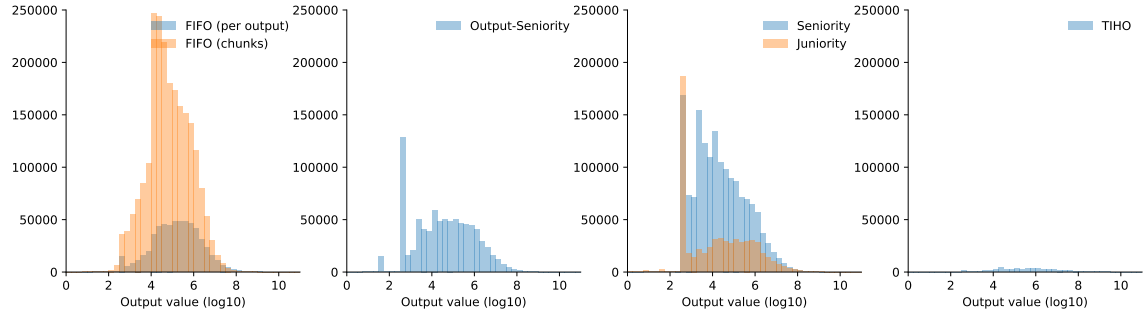


**(c)** Share of output value tainted

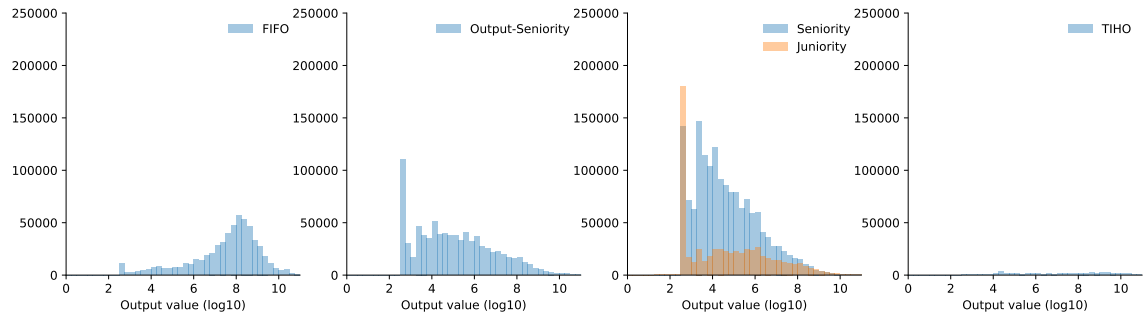


**(d)** Creation time of tainted unspent transaction outputs

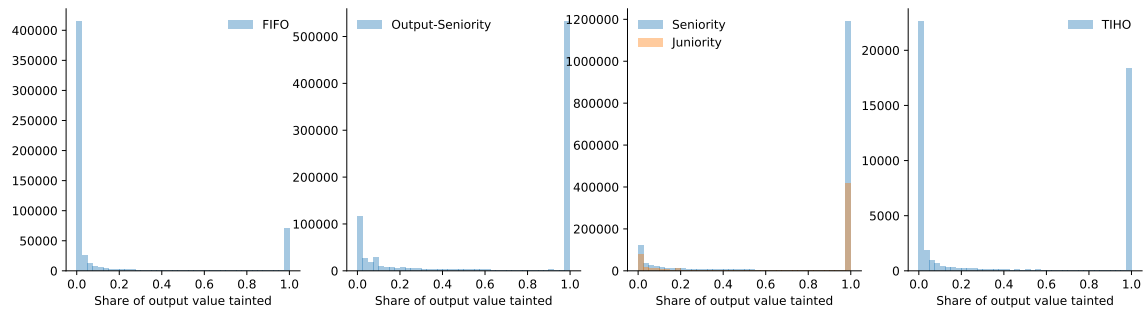
**Figure C.2.:** Analysis of the OFAC data set



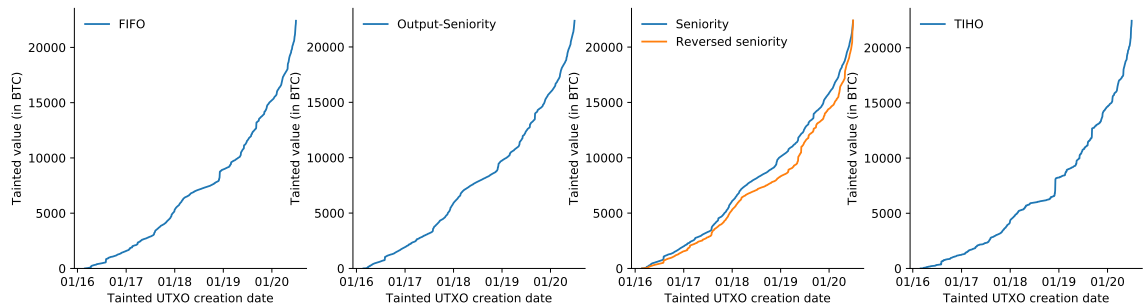
**(a)** Distribution of tainted value in all tainted outputs



**(b)** Distribution of (full) output value of all tainted outputs



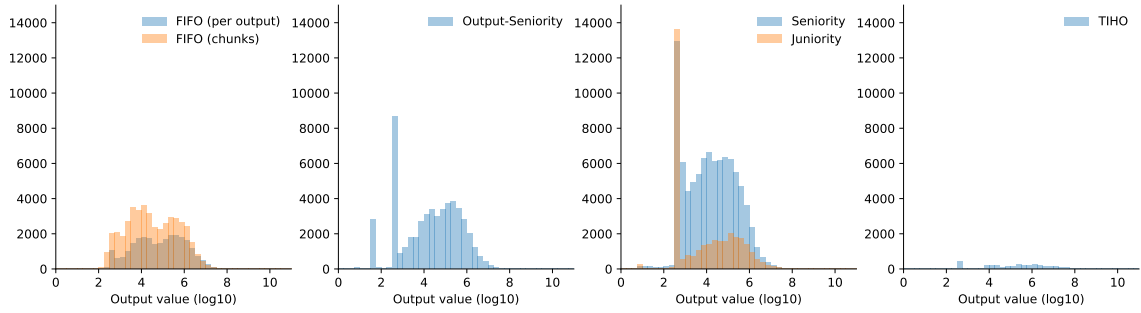
**(c)** Share of output value tainted



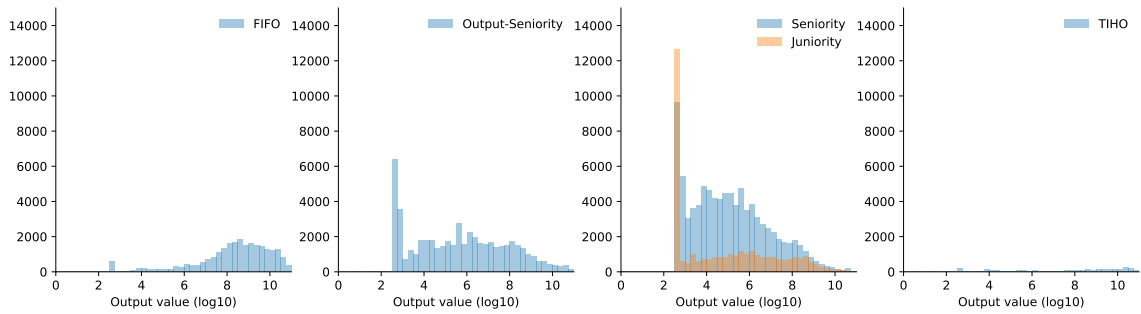
**(d)** Creation time of tainted unspent transaction outputs

**Figure C.3.:** Analysis of the Ransomware data set

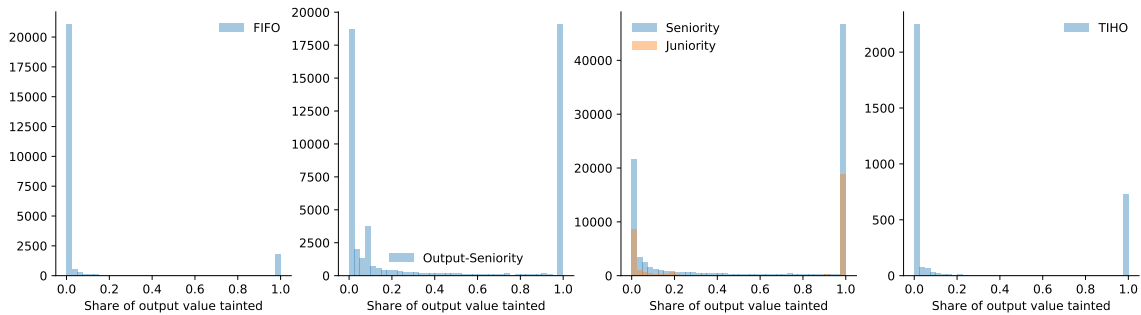




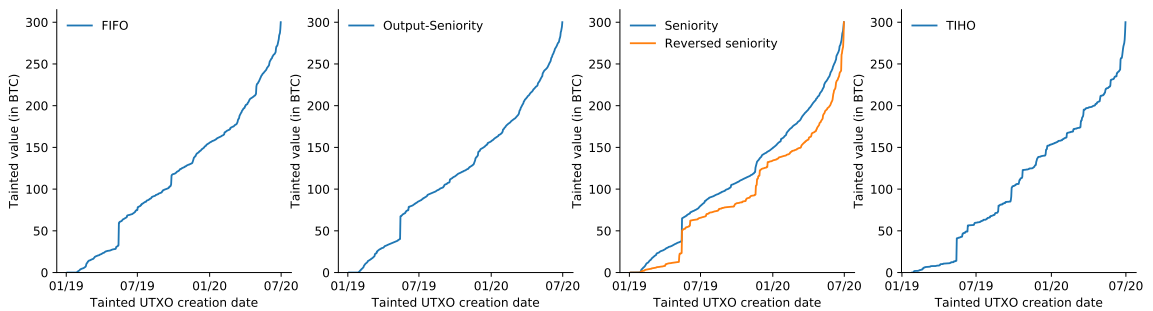
**(a)** Distribution of tainted value in all tainted outputs



**(b)** Distribution of (full) output value of all tainted outputs



**(c)** Share of output value tainted



**(d)** Creation time of tainted unspent transaction outputs

**Figure C.4.:** Analysis of the Random Outputs data set