

Investigating transactions in cryptocurrencies

Haaroon M. Yousaf

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science
University College London

March 29, 2022

I, Haaroon M. Yousaf, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

This thesis presents techniques to investigate transactions in uncharted cryptocurrencies and services. Cryptocurrencies are used to securely send payments online. Payments via the first cryptocurrency, Bitcoin, use pseudonymous addresses that have limited privacy and anonymity guarantees. Research has shown that this pseudonymity can be broken, allowing users to be tracked using clustering and tagging heuristics. Such tracking allows crimes to be investigated. If a user has coins stolen, investigators can track addresses to identify the destination of the coins. This, combined with an explosion in the popularity of blockchain, has led to a vast increase in new coins and services. These offer new features ranging from coins focused on increased anonymity to scams shrouded as smart contracts. In this study, we investigated the extent to which transaction privacy has improved and whether users can still be tracked in these new ecosystems. We began by analysing the privacy-focused coin Zcash, a Bitcoin-forked cryptocurrency, that is considered to have strong anonymity properties due to its background in cryptographic research. We revealed that the user anonymity set can be considerably reduced using heuristics based on usage patterns. Next, we analysed cross-chain transactions collected from the exchange ShapeShift, revealing that users can be tracked as they move across different ledgers. Finally, we present a measurement study on the smart-contract pyramid scheme Forsage, a scam that cycled \$267 million USD (of Ethereum) within its first year, showing that at least 88% of the participants in the scheme suffered a loss. The significance of this study is the revelation that users can be tracked in newer cryptocurrencies and services by using our new heuristics, which informs those conducting investigations and developing these technologies.

Impact Statement

The work presented in this thesis is intended to inform those conducting cryptocurrency investigations and may facilitate the design and development of projects within distributed ledger ecosystems, particularly with respect to privacy and security.

The work shown in Chapter 4 was responsibly disclosed to the founders of Zcash, who responded with updated privacy recommendations and best practices for users [150, 157]. It was covered by multiple media outlets [1, 21, 38, 50] and is mentioned in the later revisions of the Zcash protocol specification [69].

The work in Chapter 5 was responsibly disclosed to ShapeShift and was covered by the MIT Technology Review[111]. The work introduces new tracing heuristics and highlights criminal activity with case studies. These heuristics can be applied not just to ShapeShift, but to platforms which offer cross-currency trading. Weeks before we published the research, The Wall Street Journal published an investigation into ShapeShift which paralleled our work [132], and which was responded to by ShapeShift's CEO[45]. When the work was originally published, the ShapeShift exchange did not have a Know-Your-Customer (KYC)/Anti-money laundering policy, however this has since been introduced.

The analysis in Chapter 6 will inform those investigating cryptocurrency scams. The work presents a multi-angled analysis of a large pyramid scheme operating on the Ethereum cryptocurrency. Since publication, we have been contacted by a law enforcement investigator requesting advice on how to proceed with active investigations on similar scams.

All the work published in this thesis has been uploaded to open conference

proceedings and open access repositories. As of November 27, 2021 the work in this thesis had over 147 citations. The relevant source code for each project has been shared publicly to Github [[githubsourcecode](#)]. Talks at each of the conference proceedings are freely available to view online. This will facilitate impact for those researching into improving privacy and combatting scams in distributed ledgers.

Acknowledgements

This thesis would not have been possible without support from my primary supervisor, Professor Sarah Meiklejohn. Since our first run-in at Euston Station, there was never a discussion where I was not left with increased motivation and knowledge. I am truly indebted for all the support. A special thanks to my closest collaborator, George Kappos.

I have been privileged to work with many across the globe and would also like to thank all of my co-authors and contributors (in alphabetical order): Sarah Allen, Sarah Azouvi, Ben Steer, Sergi Delgado-Segura, Bernhard Haslhofer, Alex Hicks, Ari Juels, Sanket Kanjalkar, Tyler Kell, Mary Maller, Andrew Miller, Ania Piotrowska, Pierre Reibel, Sofia Rollet and Rainer Stuetz.

This research was funded with a PhD scholarship which I would like to thank the European Commission and The Initiative For Cryptocurrencies & Contracts (IC3) for providing.

Finally, I would like to especially thank my parents, Farzana and Muhammad Yousaf, for keeping my stomach full, my head dry and letting me skip years of chores in the name of science. Without their support, I would not have come this far. It is true when they say that no one is self-made.

Contents

1	Introduction	14
1.1	Scope and Contributions	16
1.2	Included Work	18
1.3	Additional Work	18
1.4	Work Done in Collaboration	19
2	Background	21
2.1	Cryptocurrencies	21
2.1.1	Bitcoin	21
2.1.2	Blockchain components	22
2.1.3	Accounting models	26
2.2	Anonymity	28
2.2.1	Multi-input heuristic	29
2.2.2	Countermeasures	30
2.3	Privacy Coins	31
2.3.1	ZCash	32
2.3.2	Dash	33
3	Literature Review	35
3.1	Cryptocurrencies	35
3.1.1	Early digital cash	35
3.1.2	Bitcoin and the alt-coins	36
3.2	Privacy	37

	<i>Contents</i>	8
3.2.1	Anonymity of Bitcoin	37
3.2.2	Anonymity of privacy coins	40
3.3	Blockchain crime	43
3.3.1	Dark markets	44
3.3.2	Thefts, Ransomware and Sextortion	45
3.3.3	Investment Programs and Money Laundering	49
4	An Empirical Analysis of Anonymity in Zcash	54
4.1	Overview	54
4.2	Background	57
4.3	General Blockchain Statistics	58
4.3.1	Transactions	58
4.3.2	Addresses	59
4.4	T-Address Clustering	60
4.4.1	Clustering addresses	60
4.4.2	Tagging addresses	62
4.4.3	Results	63
4.5	Interactions with the Shielded Pool	65
4.5.1	Founders	70
4.5.2	Miners	73
4.5.3	Other Entities	75
4.6	Interactions within the Shielded Pool	77
4.7	Case Study: The Shadow Brokers	78
4.7.1	Techniques	79
4.7.2	Results	80
4.8	Discussion and Future work	81
4.9	Conclusions	82
5	Tracing Transactions Across Cryptocurrency Ledgers	83
5.1	Overview	83
5.2	Background	85

Contents

5.2.1	Digital asset trading platforms	85
5.3	Data Collection and Statistics	86
5.3.1	Changelly	86
5.3.2	ShapeShift	86
5.3.3	Blockchain data	89
5.4	Identifying Blockchain Transactions	89
5.4.1	Accuracy of our heuristics	91
5.4.2	Alternative Phase 2 identification	93
5.5	Tracking Cross-Currency Activity	94
5.5.1	Pass-through transactions	95
5.5.2	U-turns	96
5.5.3	Round-trip transactions	100
5.6	Clustering Analysis	102
5.6.1	Shared ownership heuristic	102
5.6.2	Common relationship heuristic	103
5.7	Patterns of ShapeShift Usage	105
5.7.1	Starscape Capital	106
5.7.2	Ethereum-based scams	106
5.7.3	Trading bots	107
5.7.4	Usage of anonymity tools	109
5.8	Conclusions	112
6	Forsage: An Anatomy of a Cryptocurrency Pyramid Scheme	113
6.1	Overview	113
6.2	Background	115
6.2.1	Smart contracts	115
6.2.2	Scams	116
6.2.3	Blockchain scams	116
6.3	Forsage Overview	117
6.4	Forsage Contract Deconstruction	119
6.5	Contract Measurement Study	128

	<i>Contents</i>	10
6.5.1	Scheme statistics	128
6.5.2	Account behavior and profitability	130
6.6	Study of Forsage Community	134
6.6.1	Analysis of Forsage YouTube Promotion	136
6.6.2	Forsage user geography	139
6.7	Proposed Solutions	141
6.7.1	Targeted education	141
6.7.2	Law enforcement and regulation	141
6.7.3	Voluntary blocklisting	142
6.8	Future work	142
6.9	Conclusions	143
7	Conclusion	145
7.1	Future Directions	146
7.2	Closing Thoughts	147

List of Figures

2.1	A simple diagram illustrating a transaction	24
2.2	A simple diagram illustrating a block chain.	25
2.3	A diagram illustrating UTXO transactions.	28
2.4	Colouring nodes based upon the multi-input heuristic.	29
2.5	Transactions with and without CoinJoin.	30
2.6	A diagram illustrating the different types of Zcash transactions . . .	32
4.1	Total number of the different types of transactions over time.	58
4.2	Value of Transactions over time.	59
4.3	Total value in the shielded pool over time	60
4.4	Amount of ZEC deposited and withdrawn from the shielded pool . .	66
4.5	Amount of ZEC deposited into the shielded pool by miners, founders, and others	67
4.6	Addresses that put more than 10,000 ZEC into the shielded pool over time	68
4.7	z-to-t transactions associated with miners, founders, and ‘other’, with our heuristics.	69
4.8	Founder deposits and withdrawals into the pool detected using our heuristic	72
4.9	Value of deposits made by known mining pools into the shielded pool	74
4.10	Value linked by Heuristic 5	77
4.11	Count of z-to-z vJoinSplits over time	78

5.1	Total number of transactions per day reported via ShapeShift’s API, broken down by cryptocurrency	88
5.2	Transactional patterns analysed in ShapeShift	94
5.3	Number of transactions identified as being a pass-through	96
5.4	Total number of U-turns over time	98
5.5	Total number of U-turns over time identified by our heuristics	99
5.6	Trading bot clusters categorized by traded currencies	109
5.7	Types of transactions investigated between ShapeShift and Zcash . .	110
6.1	Screenshots of the forsage.io website	118
6.2	A visualization of the state the Forsage contract keeps for each user in the X3 matrix	121
6.3	Histogram of transaction costs on the Ethereum blockchain involv- ing smart contract function calls	124
6.4	Flow chart for the logic of who gets paid when a new user registers in the X3 system	124
6.5	Distribution of users that unlocked a given number of levels in the Forsage contract	126
6.6	Total number of users acting as slot referrer	127
6.7	Number of transactions sent from users to the five various scam contracts	129
6.8	Daily transaction count associated with the six most transacted con- tracts between April 1 and September 30, 2020	130
6.9	Total ether received by Forsage users over time and total number of users according to when their accounts were first used	132
6.10	Profit/loss histogram of Ethereum accounts that interacted with the Forsage smart contract	132
6.11	Profit/loss histogram of Ethereum accounts that interacted with the Forsage smart contract across a linear scale	133
6.12	Forsage social media interaction heat map by country.	139

List of Tables

4.1	Total number of Zcash transaction types	59
4.2	Services and identified clusters after running heuristics	63
4.3	Behaviour of the 14 active Zcash founder addresses	71
4.4	Summary of identified Zcash mining pool activity	73
4.5	Zcash amounts charged for TheShadowBrokers monthly dumps . .	79
4.6	Number of clusters that put the required Shadow Broker amounts into the pool	81
5.1	Eight most traded coins used on ShapeShift	89
5.2	Percentage of ShapeShift transactions found matching on-chain transactions for both the basic and augmented heuristics	92
5.3	Number of U-turns identified for each cryptocurrency	98
5.4	Number of regular round-trip transactions identified for each cryp- tocurrency	101
6.1	Average number of Ethereum instruction operations per transaction .	123
6.2	Summary statistics of the investigated Forsage smart contracts . . .	129
6.3	Five most profitable accounts that interacted with Forsage	133
6.4	Coded claims extracted from the top 10 most viewed Forsage videos	136
6.5	Top five countries with the highest absolute level of Forsage user engagement	137
6.6	Top 10 Forsage videos from the official channel ordered by views .	138

Chapter 1

Introduction

Bitcoin alleviates the problems of centralisation and censorship within a financial system. Anyone has the freedom to create a cryptocurrency wallet and have coins sent to them, simply by providing their public key. No user identification, passport or verification is needed to create such a wallet. This bypasses traditional financial Know Your Customer (KYC) principles, which inform a set of rules used by financial services in order to identify users prior to conducting business, and thereby anticipate and prevent crime. With this freedom, coins can be freely sent to any address without discrimination and free from censorship. These coins cannot be returned or reversed (unless the new owner explicitly does so). By default the system offers pseudonymity, as mentioned in the Bitcoin white paper, transaction privacy is supposedly preserved [106].

Notably, users in oppressed circumstances can purchase and freely send coins without the fear of banks or governments impeding the transaction, and have done so with Bitcoin [13, 123]. There is no risk from hyper-inflation as there is only a fixed number of mintable coins. Donations to charitable organisations can be made without the sender revealing who they are [16, 74]. However, such a system has flaws, and some may argue that those flaws are within its pseudonymity, which allows users to obscure their identity.

For some, this acts as a layer of privacy, preventing their financial transactions from being monitored. For others, this acts as a shroud which allows them to conceal their crimes, such as stealing coins or selling illegitimate goods. Such crimes

may need to be investigated. For example, a victim wants to track coins stolen from them to discover if they were sent to an exchange, in the hopes of identifying the criminal and collecting their funds. However, if KYC has not been adopted by that particular exchange, it is much more difficult to discover the identity of the criminal. Alternatively, cryptocurrency exchanges may want to ensure the coins they accept from users were neither stolen nor gained nefariously. These instances, amongst many others, illustrate the need to be able to track coins.

Scientific research has shown that Bitcoin is not private, and entities can be identified and tracked through the use of clustering and wallet identification techniques [63, 95, 108, 121]. This, combined with the openness of the Bitcoin source code, has inspired others to create so called privacy coins. Privacy coins are alternative cryptocurrencies to Bitcoin which improve the underlying user privacy with new features. For example: Zcash has introduced the notion of a shielded pool which uses zero-knowledge proofs to obscure the properties of transactions [156]; Dash uses Coinjoins (as PrivateSpend) which allows users to perform transactions together, to make it difficult to identify which sender paid which recipient [41, 91]; and Monero uses ring signatures to create mix-ins allowing users to include keys of other users within their own transaction to increase their anonymity set [101, 107].

Parallel to privacy coins, the ecosystem has seen an introduction in companies allowing users to freely trade between different coins. July 2014 saw the announcement of cryptocurrency exchange ShapeShift, a service that allowed users to trade coins across different cryptocurrency ledgers without the need for KYC. The service officially launched in 2015, and for the first three years operated without any identity checks, until forced to do so by regulators [147]. In 2017, the WannaCry ransomware hackers reportedly used ShapeShift to move their illicitly acquired cryptocurrencies [42].

With respect to privacy coins, one should also ask whether the claims by the developers of these hold true. Given the avenue for cross-currency trading and potential for crime, one must ask whether cross-currency trading has any affect on user privacy.

The advancements of cryptocurrency technology has resulted in an increase in new coins and services in the ecosystem. As of June 15, 2021, CoinMarketCap lists over 10,000 different cryptocurrencies [39]. Given these opportunities, criminals are quick to exploit new technologies to scam those who are not as technologically literate. For example, Bitconnect, a ponzi scheme, was introduced in 2016 and sold a blockchain-based coin that claimed to offer a high rate of return [49, 72]. The scheme was ultimately closed after regulators ruled that it was a scam, causing the value of the coin to tumble by 92% [4]. Similarly Wotoken, a ponzi scheme claiming large profits due to advanced trading bots, raised over \$1.1 billion USD before being shutdown by law enforcement [61]. As scams continue to appear, it becomes important to understand the magnitude and dynamics of these schemes.

In this thesis, we empirically analyse transactions in new cryptocurrencies and services. We investigate transaction privacy on the blockchain by empirically measuring the privacy coin Zcash, developing strategies to trace coins on the cross-trading service ShapeShift, and reveal the scale of a modern day smart-contract pyramid scheme, Forsage. We question whether coins offer the privacy they promise and, if not, what techniques can be introduced to defeat them.

1.1 Scope and Contributions

Our study analysed public transactions that occurred on-chain within the public ledger. Off-chain privacy, for example in software such as the lightning network [117] (an off-chain payment protocol), has been implemented but was beyond the scope of this research. At the start of this study there was no evidence of any published techniques that analysed Zcash [79], cross-currency trading [155] or thoroughly investigated pyramid schemes in cryptocurrencies [81].

First, this work offers the reader a foundation in (Chapter 2). We start with presenting a background on cryptocurrencies, in particular Bitcoin and its relevant components. Then we explain the anonymity of Bitcoin and state-of-the-art techniques used to defeat anonymity/privacy via address clustering, along with countermeasures proposed by the community. We end with some of the core concepts used

in privacy coins.

In Chapter 3 we present a literature review of the ecosystem. We start with an overview of the research that preceded cryptocurrencies. We then discuss the literature used to defeat and improve anonymity in both Bitcoin and privacy coins and end with a review of the various aspects of crime in the blockchain ecosystem.

Our first contribution to analysing transaction privacy is presented in Chapter 4. Here we analyse the privacy coin Zcash, finding that address clustering and tagging is a very viable technique, privacy guarantees are severely limited due to a small anonymity set, and users perform unwise transactions with traceable patterns that damage the privacy of themselves and others. We end with a case study of a prominent hacker collective that used the coin to sell security vulnerabilities.

With the rise of alternative cryptocurrencies, exchanges began to provide trading services, allowing users to directly swap coins between different cryptocurrencies. In Chapter 5 we present one of the first academic analyses of cross-currency trading. We show techniques for tracing users moving across chains, heuristics clustering cross-chain user addresses and multiple case studies showcasing criminal use.

Having been around for over a century, pyramid schemes have scammed users out of billions of dollars [64]. In Chapter 6 we contribute to other analyses of this crime in blockchain and present an in-depth empirical study of Forsage, a smart-contract pyramid scheme on Ethereum. We explain how the obfuscated smart contract , using a purpose-built transaction simulator, quantify the gains and losses, and study the promotional videos showing how the promoters leverage the new technology of smart contracts to lure users.

The question that then arises is about who are the 'others' who would implement such work. We define 'others' as the following; Fellow scientists who discover and implement mediation's which would improve user privacy, thus thwarting our attacks. Scientists who produce follow-up work with even more attacks, as well as investigators who use our heuristics to measure risk in their services, or trace transactions to combat crime.

1.2 Included Work

Parts of this thesis have been published in the following papers. All papers are joint work unless otherwise stated.

- George Kappos, Haaroon Yousaf, Mary Maller, and Sarah Meiklejohn. An Empirical Analysis of Anonymity in Zcash. In 27th USENIX Security Symposium (USENIX Security 18), pages 463–477, Baltimore, MD, 8 2018, USENIX Association, <https://www.usenix.org/conference/usenixsecurity18/presentation/kappos>. Source code: <https://github.com/manganese/zcash-empirical-analysis/>. Included in **Chapter 4**.
- Haaroon Yousaf, George Kappos, and Sarah Meiklejohn. Tracing Transactions Across Cryptocurrency Ledgers. In 28th USENIX Security Symposium (USENIX Security 19), pages 837–850, Santa Clara, CA, 8 2019. USENIX Association. Paper: <https://www.usenix.org/conference/usenixsecurity19/presentation/yousaf>. Source code: <https://github.com/manganese/tracingTransactionsAcrossCryptocurrencyLedgers>. Included in **Chapter 5**.

Other parts of the thesis are under submission to conferences, and have been published in pre-print.

- Tyler Kell, Haaroon Yousaf, Sarah Allen, Sarah Meiklejohn and Ari Juels. Forsage: Anatomy of a Smart-Contract Pyramid Scheme. In: *arXiv preprint arXiv:2105.04380*, 2021. Paper: <https://arxiv.org/abs/2105.04380>. Source code: <https://github.com/initc3/forsage>. Included in **Chapter 6**.

1.3 Additional Work

The following papers were published, as part of my research, and are not included in this thesis.

- George Kappos, Haaroon Yousaf, Ania M. Piotrowska, Sanket Kanjalkar, Sergi Delgado-Segura, Andrew Miller, and Sarah Meiklejohn. An Empiri-

cal Analysis of Privacy in the Lightning Network. In International Conference on Financial Cryptography and Data Security. Springer, 2021. Paper: <https://fc21.ifca.ai/papers/130.pdf>.

- Pierre Reibel, Haaroon Yousaf, and Sarah Meiklejohn. Short Paper: An Exploration of Code Diversity in the Cryptocurrency Landscape. In International Conference on Financial Cryptography and Data Security, pages 73–83. Springer, 2019. Paper: <http://fc19.ifca.ai/preproceedings/134-preproceedings.pdf>.
- George Kappos, Haaroon Yousaf, Rainer Stuetz, Sofia Rollet, Bernhard Haslhofer and Sarah Meiklejohn. How to Peel a Million: Validating and Expanding Bitcoin Clusters. arXiv pre-print 2021, TBC.

1.4 Work Done in Collaboration

A large part of the work in this thesis was completed in collaboration with researchers across the globe, all of whom are listed in Section 1.2.

In **Chapter 4**, Sarah Meiklejohn discovered that the Zcash shielded pool was leaking non-trivial information that could damage the privacy of users. I managed and processed blockchain data used for the project, general statistics, tagging analysis and leading the case study on the hacker collective with Mary Maller. Joint work includes the clustering heuristic with Sarah Meiklejohn and tag collection with Mary Maller. George Kappos analysed the interactions with the pool and shielded pool.

In **Chapter 5**, I wrote the scraping tool to collect all data (including all blockchain nodes and exchanges), statistics, cross-currency tracing via pass-through and patterns of ShapeShift Usage (excluding Trading bots). With regards to joint work, all authors contributed to identifying blockchain transactions and I worked on the clustering analysis with Sarah Meiklejohn. George Kappos worked on tracking cross-currency u-turns, round trip and the trading bots.

In **Chapter 6**, I processed the data from the Ethereum nodes and performed the contract measurements study. All authors contributed to the proposed solutions.

Tyler Kell worked on contract deconstruction. Sarah Allen and Tyler Kell worked on the community-dynamics study.

With regards to work not included in this thesis. In the paper, *Why is a Raven-coin Like a TokenDesk? An Exploration of Code Diversity in the Cryptocurrency Landscape* [120], I came up with the study and proposed the breakdown of work. Sarah Meiklejohn led and wrote the paper and worked with Pierre Rebel who performed the scraping and analysis.

In the paper, *An Empirical Analysis of Privacy in the Lightning Network* [80], I ran and managed our Bitcoin and lightning nodes, created and performed the property heuristic and extended the previous version of the balance discovery attack based upon the original from Sergi Delgado-Segura.

In the paper, *How to Peel a Million: Validating and Expanding Bitcoin Clusters* (under re-submission), I worked on identifying the transaction and address features, and jointly worked on the algorithms with George Kappos.

Chapter 2

Background

2.1 Cryptocurrencies

This chapter details the basic components needed to gain an understanding of cryptocurrencies.

2.1.1 Bitcoin

The first decentralised electronic cryptocurrency, Bitcoin, was created by Satoshi Nakamoto in 2008 [106]. A white paper explaining the architecture and design was originally sent as a link to the metzdowd cryptography email list in October 2008, and was shortly followed by the release of Bitcoin open source software in January 2009 [105]. That paper lays the foundation for a decentralised financial system through a digital asset class called cryptocurrencies.

By definition, cryptocurrencies are a form of digital currency coupled with cryptography and a blockchain. A blockchain is a distributed ledger which contains a record of all transactions. In Bitcoin, the complete record is public, viewable and verifiable by all participants and thus reduces the need for a central authority. All participants can choose to have a copy of the ledger, allowing it to be decentralised. Public key cryptography is used in Bitcoin to maintain the integrity and authenticity of each transaction.

Digital coins are used as a medium of exchange and allowed to be freely traded between users without fear of censorship. All coins are stored in wallets each of which has one or many public key(s), or wallet address(es), and has one or many

private key(s). Users can generate as many wallets and keys as needed, without the requirement for any verification of identity.

Coin minting and supply is fixed within the protocol, and awarded to users who maintain security by mining. Transaction finality is protected by this consensus of miners.

This was the first system of its kind that allows users to transfer digital cash without the fear of double spending and need for any intermediaries, effectively a non-custodial, trustless and decentralised global payments network. On 8th November 2021, the value of Bitcoin soared to the highest currently recorded: \$67,566.83 USD per Bitcoin. This innovation has led to the development of new research fields and new cryptocurrencies (e.g., Ethereum, ZCash and Monero). According to CoinMarketCap, as of November 27th 2021 there are over 14,000 cryptocurrencies with a total market capitalisation of 2.4 trillion USD [39].

2.1.2 Blockchain components

In this section we explore the core foundational components used in blockchain technologies. We begin with Keys and Wallets which are responsible for holding, sending and receiving coins. Then we discuss the format of transactions which are used to send and receive coins. Finally, we discuss the structure of blocks and role of miners. Further information about the cryptography can be found in various textbooks [**narayanan2016bitcoin**, **boneh2017graduate**].

Keys and Wallets

Bitcoin's foundation is built with its use of public key cryptography. This allows the user to create two related but different keys: a private key and a (derived) public key.

The private key is the secret component that controls and signs transactions, akin to a bank PIN code which is used to authorise money that is spent. However, unlike a PIN code, the private key also acts as the vault to store coins, and if leaked allows all coins to be compromised. If a user loses their private key, this would also mean a loss of access to the coins belonging to that key.

Public keys are derived from private keys. These are the user's Bitcoin address and digital fingerprint, with workings akin to a bank account number. *19Yq6pRM3mRUZMbWZoBpRWhNehiQHqznGR* is an example of a public key used in Bitcoin. Users send coins to public keys, similar to the way in which cash is transferred to bank account numbers. When sending coins, the user must sign the transaction with a unique signature created from the associated private key. The transaction, signature and public address combination is then verified by users of the network to ensure the user owns the coins being spent. Both keys, public and private, are generated using the Elliptic Curve Digital Signature Algorithm (ECDSA) [75]. Key management is performed via digital wallets. Just like a physical wallet which can hold one or more cards/cash, digital wallets hold private and public keys. The wallet is controlled by the software the user chooses to run, for example in Bitcoin this is done via the *bitcoind* program.

A user can spend funds from multiple private keys in the same transaction, and thus signs the transaction which each of the corresponding keys. The signature is stored within the transaction inside the input field, which allows the network to verify that the coins have been rightly spent. Keys are pseudonymous and are not tied to any physical identity. However, if the identity of a public key is revealed, then blockchain analytics and heuristics can be used to identify all transactions that involved the said key, thus tying and identifying the financial habits of the user.

Transactions

A transaction is the structure used to transfer coins. It has inputs, outputs and features. Inputs are the addresses from which coins are received, and outputs are addresses to which coins are sent. The number of inputs and outputs is determined by the user, but there must be at least one input and one output in a transaction. Features are attributes on transactions which trigger certain conditions. For example, a Bitcoin transaction can set a feature called locktime which means the transaction is only valid once the specific time has passed.

Figure 2.1 illustrates a simple example of a two input and one output transaction. On the input side, two addresses *1Gy9Qp...* and *1By7Qp...* are sending

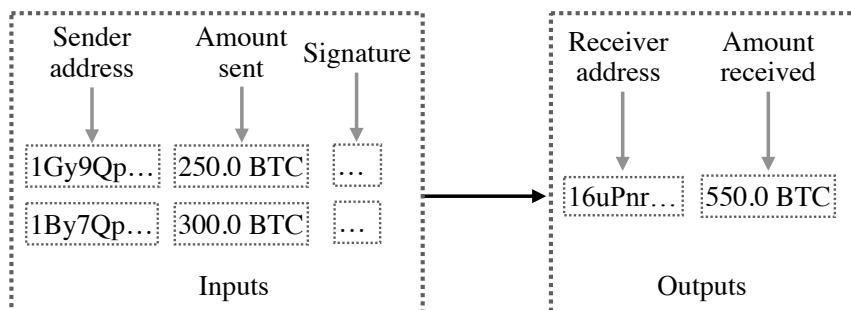


Figure 2.1: A simple diagram illustrating a transaction. The left side shows two inputs sending 250 and 300 BTC respectively. Each is attached with a signature that proves the coins were spent by the senders and can be used by the network to verify the transaction. The right side shows a single output receiving these coins.

250 BTC and 300 BTC respectively to address *16uPnr...*, the output. In this instance a single entity may own both of the input addresses or it may be two users performing a transaction together to send coins to the same recipient. Only public addresses are shown on the blockchain and only public addresses are used to send and receive coins. The sending user(s) sign the transaction with the corresponding private key. This produces signature(s) to be included within the transaction to allow for verification. Once the transaction is confirmed and verified by the network, it is stored on the blockchain within blocks.

In order to maintain security and incentivise miners to validate and publish transactions, users can optionally add a transaction fee. This fee represents a portion of the bitcoins transferred, calculated by deducting the total bitcoins in the output the total bitcoins in the input. The fee is collected by the miner who publishes the transaction onto the chain. As space within a block is limited, miners can choose the transactions they would like to include in their block. Thus, there is an economic incentive to choose transactions with high fees. This creates a market for transaction fees, and thus users must be aware of the current fees being accepted to ensure their transaction is published.

Blocks and Miners

A block is a data structure that contains both a header and block data. The data in a block is made of newly confirmed transactions (can hold zero or more transactions). The header contains block metadata and a cryptographic hash of the previous block.

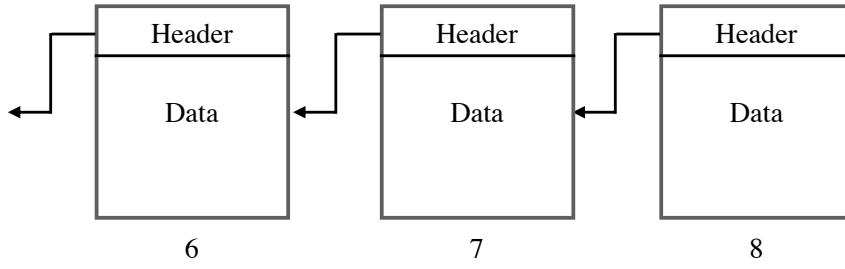


Figure 2.2: A simple diagram illustrating a block chain.

As shown in Figure 2.2, multiple blocks are chained together with their respective hashes forming a block chain. This is akin to a linked list, using hash pointers instead of traditional pointers.

As Bitcoin is an append-only ledger, the cryptographic hash pointers of previous blocks are a vital and necessary core component of maintaining security. Any modification to the data in any part of the chain changes the cryptographic hash of the modified block, subsequently causing all following hashes to change. Thus, data is only appended to the end in the form of new blocks. Each block is stamped with a block number indicating its position in the chain. This, combined with the chain of hashes, makes blocks immutable to change.

All blocks are created (in blockchain terminology *mined*) by miners who follow the Nakamoto consensus [106]. One part is a proof-of-work (PoW) algorithm which uses a set of rules that govern the network and the another is the heavy emphasis on following the longest chain.

In the case of Bitcoin, miners solve a difficult mathematical puzzle in return for the reward of newly minted coins and transaction fees. This ensures that nodes generate proof to show they have spent some computational power/energy. The difficulty of the puzzle prevent miners from spamming the network with fake blocks to quickly obtain new coins/fees as puzzle solutions require some computational power to generate but are very easy to verify. This allows block generation and transaction confirmation to be secure and decentralised within a trustless network, as trust is obtained from verifying a miners puzzle and the process incentivizes nodes to behave honestly in order to obtain the rewards.

In Bitcoin proof-of-work, the puzzles are difficult to solve, easy to verify and have varying levels of difficulty depending on the overall power of the network, with

the answer to the puzzle being a hash within the target difficulty. To generate this hash, miners use the hash of the previous block, data from transactions they choose to validate, a time stamp and a nonce. The nonce is the value that is iterated in order to find the hash. Once found, the miner then enters their public key and pushes the block to the network. Full nodes in the network verify the block contents and the answer to the puzzle. If no issues are found, this new block is added to their chain, the miner earns the reward containing both newly minted coins and transaction fees from pending transactions they included (which are now marked as confirmed) and the entire process repeats. Rewards and fees act as incentives for miners to follow the rules and participate in the network.

On average this routine takes 10 minutes which provides proof that the majority of the power in the network participated. In the case where two miners find a block at the same time, causing a chain split, the miners are all programmed to follow the longest chain which removes the need for a third party to direct the system. The system is protected against double spend attacks, as the attackers would require more than 50% of the network hash power to take control of the chain.

The longest chain rule, states that nodes in the network must follow and build upon the blockchain with the most blocks as this is the most legitimate as it has had the most computational power spent upon it. New users in the network simply follow the longest chain and begin to mine on top of it, allowing users to join and leave the network without having any negative impacts. This also prevents the network from needing to delegate authority, as by default all nodes follow the authority of the longest chain.

2.1.3 Accounting models

In this section we explain the two accounting models used in the blockchain technologies we explored, these being the Unspent Transaction Output model (used in systems such as Bitcoin, Dash and Zcash) and Account-based model (used in smart-contract based cryptocurrencies such as Ethereum).

Unspent Transaction Output

In the Unspent Transaction Output (UTXO) accounting model transactions have one or more inputs and one or more outputs. The outputs determine the receiving amount (the coins) and condition. Each output is a UTXO.

The receiving condition is a script, that when true, allows the coins to be spent. In Bitcoin, one example of this script consists of the address of the recipient who can spend the coins, provided they can sign the follow-on transaction with the associated private key. At a high level, this output is seen simply as the address. For example, a user who mined a block would receive newly minted coins sent to a script only redeemable with their address. To spend this coin, the user must turn this into an input within a new transaction.

The input contains a signature which proves the corresponding output to be true. For example, a user wants to send their newly minted coins to a cryptocurrency exchange. With their private key they create a signature that solves the script that was sent to them, thus allowing them to spend their coins, and also marking the output as spent. With this they repeat the process by creating a new output with the recipient amount and condition. The spent input combined with the new output creates a transaction.

Inputs cannot be split; when spent, the entire input is used. This ensures that each is spent only once. If users want to partially spend their UTXO, they would create an additional change output to themselves. This process of change is usually handled by the wallet software. This process is akin to spending physical cash. For example, to spend a £20 note one must give the note in its whole physical form to the recipient, who then keeps the note and in exchange returns any change. One does not physically tear a note.

Figure 2.3 shows a high level diagram of three UTXO-based transactions. Here, user Alice has received two transactions: Transaction A with newly minted coins sent to address *16uPnr...* and Transaction B with 5 BTC sent to her second address *14uPnr....* She spends both of these in Transaction C by spending both as inputs, sending 7 BTC to Bob *1Fy7Qp...* and the remainder to herself as change.

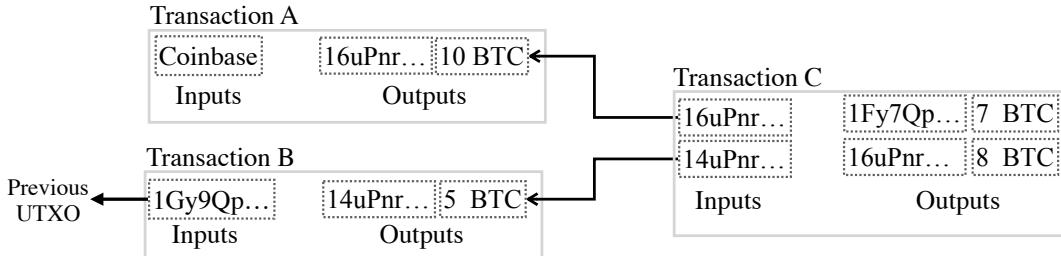


Figure 2.3: A diagram illustrating UTXO transactions.

The UTXO model ensures that no user is able to double spend, as every UTXO can only be spent once and must be completely used. Users can spend UTXOs independently of one another. The network can determine who owns what coins by simply taking a list of all UTXOs. From an analytics perspective, UTXOs make it easier for users to trace coins to their source.

Account-Based

Account-based models are an alternative and used in cryptocurrencies such as Ethereum, a smart-contract-based blockchain. An address acts as an account (with a private and public key) and coins are deposited directly to the address. Accounts have a balance and keep a record of all transactions. In Ethereum, accounts can be controlled by a user or by code within a smart-contract.

When a user wants to send funds, the transaction reduces their (sender) balance and increases the recipient's balance. Each account has a public nonce which acts as a protection against malicious users attempting to replay transactions. This nonce is incremented and attached to each transaction sent from the sending account. Compared to UTXOs, coins in accounts can be split and do not need to be fully spent. This means account-based blockchains do not create change as in Bitcoin.

2.2 Anonymity

As previously mentioned, Bitcoin is designed to operate with pseudonymity on an open ledger that maintains the public history of all transactions. This transparency allows transactions to be monitored. By analysing transaction patterns, researchers have created heuristics able to reduce the effectiveness of pseudonymity by linking transactions and addresses to real world entities.

2.2.1 Multi-input heuristic

A multi-input (also known as co-spend or clustering) heuristic is the foundational process used to cluster Bitcoin addresses, linking together addresses that may be owned by the same entity. This, when combined with address tagging, reveals the real world identity behind the transactions. The effectiveness of this technique has been demonstrated by many researchers [8, 95, 121, 124] and has formed the basis for commercial cryptocurrency analytics and surveillance companies (e.g., Chainalysis and Elliptic).

A user must have access to the private key in order to spend coins. Inputs in a transaction are spent as the owner(s) signs the transaction(s). It can therefore be deduced that addresses spent together in the same inputs are probably owned or at least controlled by the same entity. By cascading this process across all transactions, one is left with clusters of addresses that have been used together.

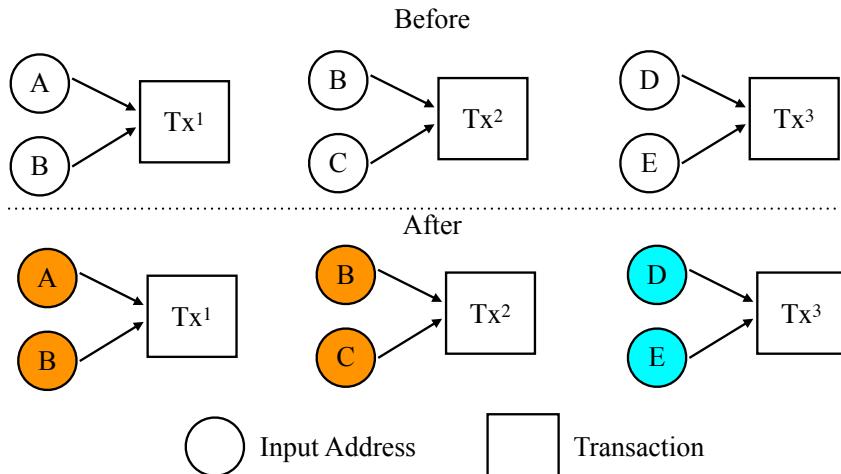


Figure 2.4: Colouring nodes based upon the multi-input heuristic.

Figure 2.4 shows a visual example of this process, with three transactions that each have two input addresses. Before, we can see unclustered transactions, indicated by a white circle. To conduct the heuristic we execute the following steps. We look at the inputs of the first transaction and label these addresses, A and B, with the colour orange. In the next transaction, we look at the inputs and check if they have been coloured (allocated a cluster) before, if so we give them the existing colour, if not we assign a new colour. In this instance the second transaction has input B,

which was labelled previously. Thus these inputs are given the colour orange. We repeat this process until all transactions have been labelled. Finally, we can see that in the final transaction input addresses D and E have not been labelled before and are given a new colour, cyan. This completes the heuristic and the "after" section shows the final result containing two clusters, cluster orange with addresses A, B and C and cluster cyan with addresses D and E.

In practise this algorithm can be modelled in a number of ways. The method mentioned above is akin to the disjoint set (union find) algorithm [124]. Alternatively this can be modelled by extracting clusters from connected components [126] within a graph, where nodes are addresses and are connected by edges to other nodes if they were involved as inputs in a transaction.

2.2.2 Countermeasures

A variety of counter-measures have arisen since the development of the clustering heuristic, namely coin mixing and privacy coins.

Mixing is the process of combining coins with users to attain increased privacy and anonymity. CoinJoin, a form of mixing, was proposed by Maxwell in the Bitcoin forums in 2013 [91]. They present a method for users to mix their coins together in a single transaction. This obfuscates coin movement by severing the link between inputs and outputs. Clusters are then combined, triggering incorrect results when using clustering techniques.

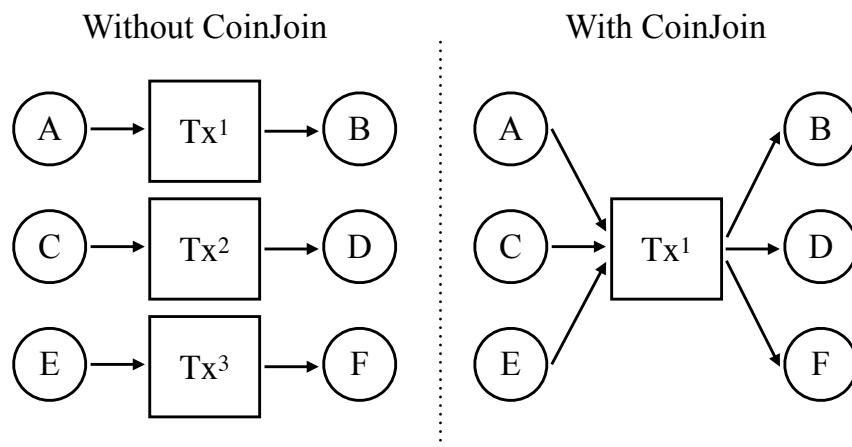


Figure 2.5: Transactions with and without CoinJoin.

For example, let us say that user Alice wants to pay Bob, Charlie wants to pay Dennis and Eve wants to pay Francis. This can either be recorded as three separate transactions on the chain, shown in ‘without CoinJoin’ in Figure 2.5. This approach makes it trivial to identify payees and recipients. Alternatively, users can perform a CoinJoin together and join their coins in a single transaction. This would appear as Alice, Charlie and Eve are sending coins to Bob, Dennis and Francis and hides the intricate details of exactly who paid whom.

Alternatively, let us say that Alice, Bob and Charlie unfortunately had their wallet addresses leaked and tied to their real-world identity, or had stolen Bitcoins and wanted to hide their trail. They could perform multiple CoinJoins one after the other with other users, in an attempt to pool and mix all their tainted coins together with non-tainted coins. This can be performed using a tumbler or mixing service, which given a transaction fee, would automate the entire process and allow the user to select a degree of privacy, e.g. mix and return funds after 10 CoinJoins.

In practice, performing CoinJoins is cumbersome. Users need to find other willing users and perform the advanced task of signing and merging their transactions. Developers have created tools that automate this by automatically finding users and performing the CoinJoin, such as Wasabi Wallet [148] and Samourai Wallet [129]. Tumblers, such as Bitcoin.Fog, allow users to mix their coins, but services like this are often shutdown by law enforcement agencies due to criminal usage and money laundering [88].

2.3 Privacy Coins

Bitcoins pseudonymity was short lived as two years after release the attacks on its protocol came to fruition [95, 124]. Since there has been an increase in privacy-preserving coins (privacy coins) which attempt to solve the privacy flaws of Bitcoin, such as Dash [41, 91], Zcash [15, 156] and Monero [101, 128], each of which provide different guarantees of anonymity.

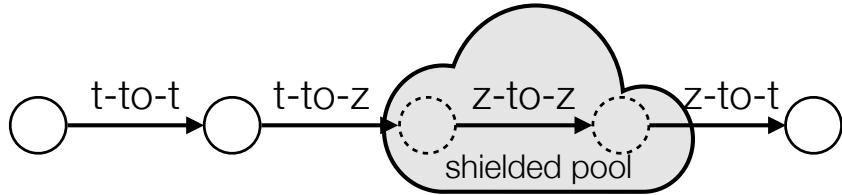


Figure 2.6: A simple diagram illustrating the different types of Zcash transactions. All transaction types are depicted and described with respect to a single input and output, but can be generalised to handle multiple inputs and outputs. In a t-to-t transaction, visible quantities of ZEC move between visible t-addresses ($zIn, zOut \neq \emptyset$). In a t-to-z transaction, a visible amount of ZEC moves from a visible t-address into the shielded pool, at which point it belongs to a hidden z-address ($zOut = \emptyset$). In a z-to-z transaction, a hidden quantity of ZEC moves between hidden z-addresses ($zIn, zOut = \emptyset$). Finally, in a z-to-t transaction, a hidden quantity of ZEC moves from a hidden z-address out of the shielded pool, at which point a visible quantity of it belongs to a visible t-address ($zIn = \emptyset$).

2.3.1 ZCash

Zcash (ZEC) is an alternative cryptocurrency developed as a (code) fork of Bitcoin that aims to break the link between senders and recipients in a transaction. In Bitcoin, recipients receive funds into addresses (referred to as the vOut in a transaction), and when they spend them they do so from these addresses (referred to as the vIn in a transaction). The act of spending bitcoins thus creates a link between the sender and recipient, and these links can be followed as bitcoins continue to change hands. It is thus possible to track any given bitcoin from its creation to its current owner.

Any transaction which interacts with the so-called shielded pool in Zcash does so through the inclusion of a *vJoinSplit*, which specifies where the coins are coming from and where they are going. To receive funds, users can provide either a transparent address (t-address) or a shielded address (z-address). Coins that are held in z-addresses are said to be in the shielded pool.

To specify where the funds are going, a *vJoinSplit* contains (1) a list of output t-addresses with funds assigned to them (called *zOut*), (2) two shielded outputs, and (3) an encrypted memo field. The *zOut* can be empty, in which case the transaction is either *shielded* (t-to-z) or *private* (z-to-z), depending on the inputs. If the *zOut* list contains a quantity of ZEC not assigned to any address, then we still consider

it to be empty (as this is simply the allocation of the miner’s fee). Each shielded output contains an unknown quantity of ZEC as well as a hidden double-spending token. The shielded output can be a dummy output (i.e., it contains zero ZEC) to hide the fact that there is no shielded output. The encrypted memo field can be used to send private messages to the recipients of the shielded outputs.

To specify where the funds are coming from, a vJoinSplit also contains (1) a list of input t-addresses (called zIn), (2) two double-spending tokens, and (3) a zero-knowledge proof. The zIn can be empty, in which case the transaction is either *deshielded* (z-to-t) if $zOut$ is not empty, or private (z-to-z) if it is. Each double-spending token is either a unique token belonging to some previous shielded output, or a dummy value used to hide the fact that there is no shielded input. The double-spending token does not reveal to which shielded output it belongs. The zero-knowledge proof guarantees two things. First, it proves that the double-spending token genuinely belongs to some previous shielded output. Second, it proves that the sum of (1) the values in the addresses in zIn plus (2) the values represented by the double-spending tokens is equal to the sum of (1) the values assigned to the addresses in $zOut$ plus (2) the values in the shielded outputs plus (3) the miner’s fee. A summary of the different types of transactions is in Figure 2.6.

2.3.2 Dash

As in Zcash, the “standard” transaction in Dash is similar to a Bitcoin transaction in terms of the information it reveals. Its main anonymity feature *PrivateSend* transactions are a type of CoinJoin [91].

A CoinJoin is specifically designed to invalidate the multi-input clustering heuristic described in Section 2.2.2, as it allows multiple users to come together and send coins to different sets of recipients in a single transaction. If each sender in a CoinJoin sends the same number of coins to their recipient, then it is difficult to determine which input address corresponds to which output address, thus severing the link between an individual sender and recipient.

In a traditional CoinJoin, users must find each other in some offline manner (e.g., an IRC channel) and form the transaction together over several rounds of

communication. This process is often centralised, as users become aware of one another and may use a service to find each other. Dash aims to simplify this for users by automatically finding other users for them and chaining multiple mixes together. In order to ensure that users cannot accidentally de-anonymize themselves by sending uniquely identifiable values, these PrivateSend transactions are restricted to specific denominations: 0.01, 0.1, 1, and 10 DASH.

Chapter 3

Literature Review

3.1 Cryptocurrencies

3.1.1 Early digital cash

One of the earliest research publications on digital cash is *eCash*, created in 1983 by David Chaum [30]. Banks issue eCash to users, who are able to store it on their local machines. eCash uses a concept called blind signatures, introduced in 1983, which blinds contents of message before being signed, hiding the contents from the signer. In the context of eCash, this allows users to hide their identities from banks, and anonymises the links between spend and withdrawal transactions. However, key issues with the system are that it requires banks to both participate and act as the central authority, and banks have the power to mint digital cash as well as the ability to refuse deposits.

Following this, in 1998 Wei Dai proposed *B-money* an anonymous and distributed cash system [40]. B-money is similar to Bitcoin, whereby, new transactions are broadcast to all users, who also keep a complete record of the ledger. Users are represented by public keys, and mint coins via completing computational puzzles. This solved the previous issues raised by eCash as the new features alleviated the need for an issuing bank. However, B-money raised a number of other issues. One was with the protocol for creating new coins, which required account keepers to decide on the cost of computations. This at the time was not feasible, due to the rapid advancement of technology. Wei did propose an alternative way to mint money

with a four step process, however this seemed complicated. Secondly, B-money was entirely conceptual, having only been written on paper with no code or tests to support its claims.

In 2005, Szabo introduced *bit gold* [115, 138], which used a proof-of-work scheme to mint new coins. In this scheme, users would compete to solve computational puzzles and in doing so would earn bit gold. Each puzzle solution became a part of the next one, and thus would form a chain. However, similarly to B-money, bit gold remained as a concept and was not turned into working code.

3.1.2 Bitcoin and the alt-coins

In November 2008, Satoshi Nakamoto revealed the Bitcoin white paper to the metzdowd's cryptography mailing list [105, 106]. This system solved previous problems by creating a distributed shared public ledger that prevented double spending, did not include any third parties, allowed pseudonymous user accounts and used a modified version of Hashcash's proof-of-work scheme to generate new coins [10]. As of today, Bitcoin is one of the most valuable cryptocurrencies with a market capitalisation of many billions of dollars (USD) [23, 39]. Researchers and companies have examined and evolved the underlying software, forking the source code into many new coins or developing new software based on the presented ideas. These coins are named alternative coins or alt-coins, and examples include Litecoin [85], Dash [41], Zcash [156] and Ethereum [28].

Litecoin, an alt-coin forked from Bitcoin, has some changes such as different proof-of-work algorithm, decreased block generation times and an increased maximum number of coins. Dash offers faster transaction confirmations and is packaged with a coin mixing service. Zcash builds upon Bitcoin by adding privacy preserving transactions that make use of zero-knowledge proofs [15, 69, 97]. Ethereum is a blockchain proposed by Buterin in 2013, is not a fork of Bitcoin but it carries many of the same principles. The main differences include: an introduction of a Turing-complete (limited by execution costs) programming language named Solidity, allowing user created smart contracts to be published on-chain, introduction of custom user created tokens on-chain and an account-based accounting model. As of

July 2021, Ethereum is the second most valuable cryptocurrency, just after Bitcoin.

3.2 Privacy

The openness of the ledger allows the transaction flows of the network to be monitored. A large volume of work is dedicated to analysing the privacy of Bitcoin, including improving its anonymity and analysing the privacy of alt-coins. Here, we outline the most relevant works.

3.2.1 Anonymity of Bitcoin

Reid et al. presented one of the first anonymity analyses on Bitcoin in 2011 [121]. Their work analyses nearly two-years-worth of data across multiple angles. Firstly, they model both transaction and wallets as two separate network graphs, revealing that the network has been increasing over time and that the user network is cyclic with users sending coins to previously-owned addresses rather than treating addresses as one-time use. By focusing on privacy, they also discovered that public-keys can be linked with others via passively analysing the monetary flow between addresses. Using this, they performed one of the first academic analysis of a theft of bitcoins, tracing some of the stolen funds to an online wallet provider. However, some of the limitations of this study are that the analysis was conducted very early in the Bitcoin timeline and therefore was performed on a small graph, with potential users who were early adopters, and thus, this may not have represented a mature graph.

In 2012, Ron and Shamir [124] analysed statistics across the entire blockchain, from genesis until May 2012. Compared to previous analysis, the Bitcoin graph had since grown by a factor of three. Using the union-find algorithm, Ron and Shamir cluster 3M addresses as 1.8M entities, identifying that some of the most active clusters belong to exchange or wallet providers. Furthermore, Ron and Shamir identified that many bitcoins remained unused in sink addresses, and most transactions moved only small quantities of the total bitcoin in circulation, and users created transaction chains in an attempt to hide and weaken the links between their addresses. In their paper, the researchers could have made further efforts to identify

the larger entities, as only three out of a million or so entities are revealed. Their reasoning and usage of the union-find algorithm is largely unclear, as is the method used to identify clusters.

In the same conference as Ron and Shamir (Financial Cryptography and Data Security, 2013), Androulakil et al. presented an evaluation of user privacy [8]. They offered a short evaluation of privacy using two heuristics: a multi-input heuristic and a change heuristic, as well as a longer evaluation on an account-based behaviour simulator. Using the first heuristic they are able classify 1.6M addresses into 1M entities, and this is further reduced to 693k entities when using the second heuristic. With the simulator, they identify that Bitcoin does not do enough to protect the privacy of its users, as despite following the recommended guidelines, behaviour-based clustering can profile 40% of participants. The evaluation of the two heuristics is very short, and the paper largely focuses on the data from the simulation rather than the ground truth data from their defined heuristics.

Meiklejohn et al. [95] performed an in-depth measurement study on Bitcoin and its anonymity. They significantly expanded upon previous clustering efforts and examined the network from creation to April 2013, revealing network statistics, usage and account heuristics, attacks on pseudo-anonymity and the role of entities in the wider ecosystem. They interacted with a wide variety of services, manually obtaining tags for 1,070 addresses. Using both the multi-input, and introducing, a peel-chain heuristic, they analysed the entire network and expanded their tags across clusters covering over 1.8M addresses. Their work demonstrates the effectiveness of such clustering techniques. In addition, they presented multiple real world case studies tracking criminal activities, finding that funds from a Bitcoin ponzi scheme were distributed to a variety of services, and tracking multiple thefts to exchanges. However, Meiklejohn et al. also presented progressive enhancements, as both heuristics had been previously published in some form. The effectiveness of the reidentification attack can be reduced if users use mixing services. Their clusters were not compared against ground truth, however, during this time period in the Bitcoin ecosystem, ground truth data was likely only obtainable by directly

contacting exchanges and services that may not have wanted to share their sensitive data.

Spagnuolo et al. presented BitIodine [136], a modular analytic framework which parses, clusters, classifies and visualises Bitcoin data. In their work they describe the workings of the system which uses a variety of open-source tools, including the C programming language, Neo4J (a graph database) and Gephi (an open source visualiser). They implement both the multi-input and change heuristic, and test these on real-world use cases. Notably, they argue that they find a connection between an address belonging to the owner of a dark market and a separate address which at one point contained over 111k BTC. In addition, they analysed addresses that sent and received coins from the CryptoLocker ransomware. Using their tool they estimate the scammers obtained over 1.1M USD worth of BTC and identify what is claimed to be a potential "test" transaction that occurred days before the first ransom was paid. There are however some limitations with both the work and tool. With regard to the work, the case studies are very short, and stating an address has a "meaningful connection" to another is not indicative of holding any real significance. The tool itself stores all transactions and clusters in memory, which makes the process memory-intensive and costly, given the increasing size of the Bitcoin ledger. It is limited to exporting static graphs and is not bundled with a user interface. Thus the tool requires users to have a high level of technical knowledge to operate.

Kalodner et al. have presented an open-source blockchain analytics tool BlockSci [77]. Their paper describes in detail the design choices, and architectural challenges, and then presents multiple real-world case studies. When compared to standard graph analytic platforms, the BlockSci program is significantly faster and easier to use than BitIodine, as it is bundled with documentation and a python interface. The tool itself features an extensive analytics engine, allowing clustering and tracing across UTXO cryptocurrencies and their corresponding forks. Within their use-cases, the authors demonstrate that anonymity is affected by usage patterns in multi-signature wallets, identifying that 5% of Bitcoin addresses have their privacy

affected as users cash out their coins on corresponding forks, such as Bitcoin Cash. By clustering these transactions across Bitcoin forks, they reveal information about users that can be linked back to their wallets. The paper does not compare the tool against previous blockchain analytics programs such as BitIodone, but instead against graph analytics tools. Since the release of the paper, development of the tool has since discontinued. The tool also requires a machine with very large memory in order to cluster the blockchain.

The research above demonstrates that Bitcoin's anonymity can be breached, since there have been new proposals to resolve these issues and improve privacy without major changes to the protocol. One process is CoinJoin [91], where users create a transaction together, merging their inputs and outputs in order to reduce the linkability within their transaction. This process is one of the foundational features of the cryptocurrency Dash [41] which automates the entire procedure. This CoinJoin mechanism is explained in more detail in Chapter 2.2.2.

Analysis around the anonymity of CoinJoin [9, 90, 94, 96, 102] presents many issues. Atlas affirms that inputs and outputs of a CoinJoin can be linked through brute-forcing all possible summations and more so, if denominations can be uniquely identified [9]. However, the process can quickly become computationally expensive as CoinJoin is a variant of the Knapsack problem [90, 125]. Dash [41] improves the usability of Maxwell's CoinJoin [91] through automating the process of finding participants to mix with using fixed denominations, further explained in Chapter 2.3.2. However this adds a time delay to transactions, as users must wait to find others with whom to mix.

3.2.2 Anonymity of privacy coins

In Bitcoin transactions information is entirely public and, as previously shown, allows any entities to freely track the movement of coins. Protocols have since been developed to integrate anonymity. Many of these have been implemented into alternative coins marketed with a privacy focus.

Zerocash is a protocol, forked from Bitcoin, that adds privacy-preserving transactions [15, 97]. These use zero-knowledge proofs to "shield" coins, adding an

anonymity layer that hides the amount, sender and receiver. Thus, when coins are spent, no information about the transaction is revealed other than a potential fee. Users have the option to use this feature, for anonymous coins they utilise zerocoins, also known as shielded coins/transactions, and for non-anonymous coins they can spend basecoins, also known as transparent coins/transactions. Coins can be converted to either type at the users preference. This concept is further explained in Chapter 2.3.1. Zerocash was subsequently commercialised into the cryptocurrency ZCash. The initial downsides of ZCash were that the original versions required a significant amount of processing power (3GB of RAM and several minutes) to construct the zero-knowledge proof required for a single shielded transaction. This was not suitable for sending quick transaction, nor were users able to perform such computations on their mobile devices. They originally needed to run a full ZCash node, as most wallet software only supported transparent transactions. In doing so, this led to the second issue, whereby running a node and performing shielded transactions required some familiarity with the command-line, as even the software did not come with a user interface. Given that this type of transaction was optional, not enabled by default or the above issues, the anonymity set of the shielded pool was thus confined to users who were technically advanced. Recent advancements have since solved these issue such as reducing the memory usage by 98%, reducing transaction times by 80% [24], and introducing mobile wallets which support all types of transactions such as Zecwallet Lite [161].

Prior to the above advancements, researchers had revealed the short-comings of the shielded pool. In 2018, we published the first peer-reviewed study discussing the limitations and privacy flaws that were then present in Zcash. This is explained in detail in Chapter 4. Whilst we conducted this research, others published studies that had parallels with our work.

In late 2017, Quesnelle [119] also published a short study analysing the privacy of Zcash. They revealed that transaction meta-data can be used to link coins between deshielding and shielded transactions. They identified that the vast majority of coins were not used within the shielded pool, given that only 19.6% of

transactions used some form of a privacy-preserving feature and that 57.7% of these were used to deshield coins. Although Quesnelle is credited with creating round-trip transactions, this is also a concept that we discovered during our research. In this process transparent coins are sent to a shielded address, and then a similar or identical number of shielded coins is returned to a public address within a close time frame. Around 31.5% of shielded coin transactions were conducted within round-trips. Using these, Quesnelle linked coins belonging to miners after their respective mining pool had deshielded newly minted coins. Quesnelle believes the reasons for these short-comings are due to lack of support within third-party wallets and high computational costs. The study has limitations because the work exclusively focuses on brief statistics and round-trips. The background does not clearly explain the workings of Zcash, and the transparent transactions which account for the majority of transactions in Zcash have not been fully discussed.

Biryukov et al. [18] published heuristics analysing the behaviour of miners in both transparent and shielded pools. They extended previous work [79] by presenting heuristics targeting the two payout strategies deployed by mining pools. The first strategy identifies users of mining pools by following transactions that pay users with public addresses, whereas the second strategy identifies mining pools that directly pay users from a shielded address. Combined, their heuristics are able to link 88.4% of all mining rewards to shielded addresses. When combined with the Founder heuristic from previous work, presented in Section 4.5.1, they are able to link over 84% of all de-shielded transactions. In terms of downsides, the work does not focus any analysis on transparent transactions and their heuristics primarily target de-anonymising mining pool transactions.

In 2014, developers created Monero [101], a privacy focused cryptocurrency based on Nicolas van Saberhagen’s whitepaper on CryptoNote [29, 128]. Monero uses ring signatures, which are a digital signature that can be created by a user from a group of users that each have keys. When creating a signature, it is computationally impracticable to identify which member of the group created the signature. When a user wants to spend their coins in Monero, they generate a ”min-in”, which

is a ring signature using their output as a key with other public keys taken from previous outputs in the blockchain. These all act as decoys within the new transaction, and when created the inputs appear to be equally likely to have been originally spent, which masks the origin of the transaction. All aspects of the transaction are obfuscated, hiding the senders, receivers and amounts.

Multiple research projects in 2017 identified issues which can disrupt the anonymity of the system [tramer2020remote, 83, 98]. Kumar et al. presented three heuristics that were used to trace 87% of inputs [83]. By using temporal analytics, they identify that it is not as difficult to predict the correct output in a ring signature. Given that over time it is more likely a UTXO had already been spent and thus the most recent output is very likely to be the real one being spent. Secondly, by using a technique called leveraging output merging, they analyse user behaviour in transactions where two outputs belong to the same entity. Thirdly, they present that users themselves can disrupt the anonymity of others by choosing to do a mix with zero mix-ins, damaging the anonymity set of others who may have referenced their coins. Moser et al. presented similar work, by detailing issues surrounding the coin selection algorithm [98]. Similarly, they found it more likely to select more recently generated outputs.

These works all reveal that, despite best efforts, so called "privacy coins" are vulnerable to attack. Over time, scientific advancements will identify weak points in systems that were previously thought to be secure. Therefore, it is pertinent to develop new techniques in order to strengthen current practices and create better mitigations.

3.3 Blockchain crime

The decentralised, pseudonymous and uncensorable transaction system has attracted a wide range of users, some of whom exploit the system for nefarious reasons. Bitcoin's ability to transfer coins across borders (regardless of location), combined with misleading claims that Bitcoin offers anonymity, have attracted those involved in crime. Such crimes range from operating a dark market, theft, and ran-

somware, to scams and money laundering. This section details the literature that analyses transactions linked to misdeeds.

3.3.1 Dark markets

Underground markets are those which sell goods or services that may or may not be forbidden by law. Historically, there have been many examples with some predating the second world war [43, 143]. With the advancement of technology, these markets have naturally made an online appearance, with the earliest cyber markets selling information about goods and services (e.g., credit card numbers, viruses, botnets) via internet relay chats (IRC) [54, 140]. Following take downs by law enforcement, this activity moved to online forums, marketplaces and registration-only websites with research analysing their appearances in China [163]. The underground economy is now considered to be an integral piece of cybercrime, due to servicing criminal enterprises with illicit goods and services [62].

Silk Road, a notorious and commonly known underground market attributed to Bitcoin launched in 2011 [33]. It operates similarly to Amazon and eBay, where anonymous sellers are able to sell goods and services to anonymous consumers [70]. The market ran as a Tor hidden service using Bitcoin as the medium of exchange [19]. The market sold a large variety of goods, such as unlicensed firearms and drugs (both legitimate and illegitimate), and services such as assassination, botnets, malware and targeted hacking. The initial version, 1.0, operated for over two years and the second version, 2.0, for a year before both being shut down by the FBI [7, 37].

In 2013, Christin [33] published an in-depth measurement study of Silk Road, analysing the types of products sold, the evolution of sellers including their countries of origin, and economic indicators, including the use of Bitcoin and sales volume. With regards to Bitcoin, they estimated that over a 29-day span the marketplace transacted 1.3M BTC (1.22M USD), which they believe corresponded to between 4.5% and 9% of all exchange trades. The Bitcoin analytics in the paper are very short and the authors make no attempts to trace the illicit funds, however this is acknowledged as not being the primary focus. In 2014, Spagnuolo et al. [136]

used their Bitcoin tracing, classification, and verification system to analyse crime and perform an analysis of potential connections between the Silk Road cold wallet and its founder, and measured on-chain crime from the CryptoLocker ransomware. We discuss this paper earlier in Chapter 3.2.1.

Many new anonymous market places have been created since the closure of Silk Road, with recent crime studies reporting an increasing trend in dark market revenues and competition. In 2015, Soska et al. [135] performed a long term measurement analysis of 35 marketplaces across two years, detailing the growth of the underground ecosystems. In their study they scraped and parsed the data from these market places across multiple snapshots to analyse how the system evolved over time. Through analysing sales volumes in Silk Road, they projected the marketplace produced \$100M USD a year, which was in accordance with the amounts projected by the US government. Overall, they estimate that the ecosystem transacts over \$500,000 USD per day. Market place closures and shutdowns resulted in users moving their business elsewhere, and that, interestingly, the ecosystem was resilient to scams and law enforcement take-downs. In terms of limitations, it is unclear whether the researchers have made their datasets available for others to use in their own research. Some of the datasets are censored due to insufficient amounts of data collected. The authors did not appear to directly interact with the services as estimated sales volume is predicted based on user feedback instead of real transactions.

Markets are, however, actively shut down, either voluntarily (Sheep Market place [130]) or forcefully by law enforcement (Utopia [57]). Further analysis of underground markets includes the following works [20, 66, 67, 86, 137], however these do not focus on cryptocurrency transactions but demonstrate techniques used to collect data.

3.3.2 Thefts, Ransomware and Sextortion

Cryptocurrency exchanges and user accounts have been subjected to numerous cases of theft. Opened in July 2010, Mt Gox was one of the worlds largest digital asset exchanges, allowing users to trade via standardised market methods [142].

At its peak the service was responsible for handling over 70% of all bitcoin transactions [55]. In 2011, it was reported that someone had been gradually stealing coins from the exchange-owned wallets, siphoning 750,000 bitcoins (\$330 million USD) which subsequently caused the company to file for bankruptcy [154].

In 2015, the Bitstamp exchange was breached and hackers stole 19,000 bitcoins (\$5 million USD) [65]. Employees from Bitstamp were targeted through phishing campaigns which ran for weeks. The attackers distributed malware via Skype and email channels, compromising internal machines. In 2016, Bitfinex, a Hong Kong based exchange, was also breached with hackers stealing 119,756 BTC (\$72 million USD) causing a 20% drop in the Bitcoin price when announced [22, 31].

Lazarenko et al. [84] list and classify 48 attacks on blockchain projects which had lead to theft of coins. These attacks are grouped into eight categories, some of which include insider attacks, phishing and malware. For each project attacked, they list the number of coins lost and detail the cause. Their analysis shows that the number of attacks on blockchain projects has increased annually, likely due to the increase in available technologies. Exchanges had been subject to the most thefts and attacks on Bitcoin, whereas Ethereum Initial Coin Offerings (when an unregulated entity raises money through distribution of cryptocurrency assets) had the most Ether stolen. Overall the report finds that 59% of blockchain related projects are closed after a cyber attack. The analysis in the paper is very brief, and the paper simply lists out a number of attacks, however this is due to the research being classified as a survey rather than an in-depth study.

Since 2012 a new type of malware has emerged: ransomware. It is malicious software that prevents a user from accessing their computer until they pay a certain amount to the operator. The payloads are spread through a variety of ways, for example through malicious email attachments [2] or via exploited vulnerabilities [100]. In May 2017 the ransomware WannaCry attacked Windows Machines worldwide, encrypting user data and holding it as ransom, requiring users to pay with bitcoin to regain access. Some of the infected machines belonged to national hospitals and telecommunication companies [82]. The ransomware caused an esti-

mated \$4 billion USD worth of damage [17]. After the outbreak, it was discovered that the attackers had used the cross-currency exchange ShapeShift to convert their tainted Bitcoins into Monero, which was reported to then 'disappear' [56].

Conti et al. published an in-depth study of ransomware on Bitcoin, presenting techniques to identify, collect and analyse transaction data [36]. They present an identification framework to identify extorted ransom transactions which consists of three components. First is the identification of addresses from online resources, removal guides and threat reports. These are then clustered using both the multi-input and change heuristics. Next is the extraction of all transactions which contain these addresses, and finally, the classification of the amounts received as ransom if at least one of their payment conditions is satisfied. This methodology is applied to twenty ransomware cases, and for each they detail payment strategies, methods of infection and ransoms extorted. Overall ransomware operators earned an excess of 3M USD, and the operators of WannaCry earned 238 payments averaging a total of 86k USD (47 BTC), compared to the operators of CryptoWall who obtained 2.2M USD (5,351 BTC). The work is very thorough and the appendices list all the addresses used in the initial portions of the investigation, which is useful for future research. Limitations of the work include that the ransom addresses are collected from public sources and such data quality cannot be guaranteed, however they compare their results and find that it is similar to previous works.

Paquet-Clouston et al. [113] present methods identifying illicit Bitcoin transactions by focusing on 35 ransomware families. The work obtains, clusters and filters a set of seed addresses obtained from researchers and online sources. By following the flow of money, they found that multiple ransomware families interacted with the same actor. Across four years they reveal a lower bound of over 22k BTC (\$12M USD) in ransomware payments, with the Locky ransomware receiving over \$7.8M USD and more than 50% of payments. The researchers concluded that a small number of actors, just three families, dominated this genre of crime and were accountable for 86% of the marketshare.

Huang et al. [71] created a framework to track ransomware end-to-end and

applied this to real-world cases across a two-year period. They first obtained a list of addresses from public ransomware infection reports, seed addresses and a form of synthetic address (victims they created and tracked by sending operators micro-payments). These were turned into clusters using co-spend heuristics, and transactions to and from the clusters were filtered and analysed. Overall, they estimated that operators from 10 ransomware attackers obtained \$16M USD from 20k potential victims. Through looking at inflows to clusters, it is found that exchanges account for 40% of payments. They also revealed that BTC-e, a Russian exchange, prior to being seized by US law enforcement, was a key exit point used for criminals in the Locky and CryptoDefense ransomwares, as they saw \$3M USD flow through their exchange.

Some scams feed on the fear, gullibility and technical infancy of the target users. Paquet-Clouston et al. [114] analysed emails sent by criminals to victims of sextortion scams requesting bitcoin as payment. Sextortion is a spam scheme whereby an attacker emails victims, claiming that sensitive and private photos or videos will be leaked to their contacts unless they are paid some bitcoins. The researchers analysed 4.3M emails, bucketing emails into 15 campaigns and extracted all bitcoin addresses. They extracted 245 addresses from the emails, which when clustered and filtered came to 485 payment addresses that had received coins. Their analysis revealed that scammers attempt to extort higher amounts of coins based on the language used in the emails. Emails written in the English language asked for a mean of \$745 USD whereas Spanish language scams were asked for \$249, indicating that attackers adjusted their prices based on the victims' perceived language and location. By following cash flows from the spam clusters, they discovered that entities were moving coins to known exchanges. However, this part of tracing is very limited as tags were obtained from an open source website. By analysing the campaigns they identified that multiple clusters appeared across multiple campaigns, suggesting that the majority of revenue was collected by a single real world entity. Untagged clusters were sent 48 BTC (17% of revenue), and these were mentioned to possibly be potential cash-out services. In conclusion, they estimated that one

entity may be responsible for the majority of crimes, with scammers harnessing a lower bound of over \$1 million USD within 11-months. In terms of limitations, the addresses were obtained from a data set which was caught by a spam filter, and thus may not have reached most recipients, thus the estimations are a lower bound and only cover a small section of this ecosystem. The multi-input heuristic may have clustered in addressess belonging to the same author but perhaps different scams. The source of the tags is extremely limited, and thus it is unclear exactly how many of the coins were sent to and from exchanges.

3.3.3 Investment Programs and Money Laundering

Past research has quantified and described crime and scams running on blockchain ecosystems that use high return investment programs and ponzi schemes as well as mixing services as a form of money laundering.

Vasek et al. [146] presented analysis of four types of scams (ponzi schemes, mining scams, scam wallets and exchanges) that used Bitcoin, identifying 192 scams and tracking their payments. By collecting scams from various data sets online, and after a cleaning process, they extracted all relevant transactions from the Bitcoin blockchain. They modelled 42 scams into four categories and analysed each. One class of scams they analysed was high yield investment programs (HYIPs). These are schemes that promise investors a high rate of return. Previous investors are paid by new investors, and this process repeats until the scheme closes or collapses. By comparing traditional HYIPs to Bitcoin based HYIPs, they found that Bitcoin was not yet widely accepted. However, some act more traditionally using fiat currency and then transitioning into Bitcoin. From these, they identified a user who earned 1.6M USD. Another category they analysed was scam wallets. These are online wallets that claim to host Bitcoin for users, but the operators, in-fact, steal the coins that have been deposited. One scam wallet in particular earned 4,100 BTC (\$1M USD). Overall, they found that scammers earned \$11 million USD from 13,000 victims. The research within this paper is very thorough, covering a wide variety of scams including economic perspectives. The work however only focuses on how many coins were received and they do not track the illicit funds

after they have been received.

In 2018, Bartoletti et al. [11] applied data mining techniques and machine learning algorithms to detect Ponzi schemes in Bitcoin. Ponzi schemes are a form of HYIP where users of the schemes are paid only by other new users that join the scheme. They collected 32 addresses belonging to ponzi schemes and through clustering obtained 1211 addresses. Analyses revealed that these received \$10M USD worth of deposits. Using these clusters they derived and extracted features (e.g., statistics, measures of inequality) from both these and random addresses. By testing a range of supervised binary classifiers, they found that the random forest algorithm was correctly able to identify 96% of addresses as ponzi schemes. This experiment is very thorough, and the authors released addresses and features used and tested across multiple classifiers and ponzi schemes. The limitations are that this approach was only tested against ponzi schemes, so it would be interesting to see whether it is applicable to other types of scams.

Bitcoin is not the only cryptocurrency prone to being targeted by crime; this behaviour also occurs on Ethereum. Chen et al. [32] used data mining and machine learning to detect Ponzi schemes, scams that fraudulently promise high returns by generating income for previous investors by taking money from later investors. They first analyse a known smart contract ponzi scheme by studying the transactions between the contract and participants. With this they developed sets of features that are key to ponzi schemes. The first set of features looks at statistics, including the number of payments into and out, and the proportion of users who sent money to the coin before ever receiving anything in return. The second set analyses the use of opcodes, assembly like commands used by contracts extracted from the Ethereum Virtual Machine. These features were tested with a machine learning algorithm that classified whether the contract resembled a ponzi scheme. The classifier was tested on 54 previously known ponzi schemes and was able to detect 45 schemes. The undetected 9 were manually investigated and found not to be ponzi schemes. Out of 280k contracts on Ethereum, they estimate that 434 ponzi scheme smart contracts exist. In terms of limitations, the reasoning for the machine learning algorithm is

not convincing, it would be clearer to compare this against other algorithms to justify the decision. The results detected 386 algorithms that were not checked, and it would have been helpful to take a random sample to manually check whether they could be confirmed.

Bartoletti et al. [12] performed a survey on ponzi schemes in Ethereum and analysed them from multiple perspectives. Similar to prior works, they extracted features from Ethereum ponzi schemes based on their transaction behaviours and smart contract code, using this to create a machine learning algorithm to classify and detect smart contracts. In addition, they also measured multiple statistics, including the gains and losses incurred by users of 23 contracts, identifying that most users never receive any money after investing with only one or two users who earn high amounts of coin. The study is very thorough and ends with raising a number of recommendations that could be used to combat this crime. However, the study does not analyse where the money goes after being sent to the key players in the scheme.

Ponzi schemes, extortion scams and ransomware are not the only crimes occurring on blockchain. A Pump-and-dump (P&D) is a scheme where users artificially inflate (the “pump”) the price of an asset by recommending that others purchase it at a specific time in a coordinated manner, in the hopes that its value increases. Once the value reaches a certain point, the holders of the scheme sell that asset to profit from the gain in price (the “dump”). This scam was prevalent within the stock market space [25], and is now performed on cryptocurrencies. With cryptocurrencies, organisers use online chat services such as Telegram to coordinate their schemes. Several research articles analysing schemes have been published, identifying how they are run and producing models to detect their activities [60, 78, 87, 153].

Tao et al. [87] published an in-depth study analysing the early P&D ecosystem. They manually collect data from pump and dump groups online and, after filtering and processing, analysed 500 distinct P&D events from 80 groups involving 239 tokens across 3 cryptocurrency exchanges. Their analysis revealed that P&Ds are regularly scheduled at the advertised times, only lasting for several minutes but being able to return an average of 212% for investors. However, an investor’s per-

formance depends on when they receive the message to conduct their trade, with some groups operating with insiders who are aware of the coins beforehand. Jihau et al. [153] extend previous works by revealing the entire anatomy of P&D schemes on cryptocurrencies and by producing a prospective prediction algorithm that able to identify and trade coins that signal movements indicative of P&Ds. Their algorithm strictly uses market data and is able to generate a 60% return over two and a half months.

Previous research has addressed the use of money laundering tools in the ecosystem. These tools can be used to launder dirty Bitcoins, obscuring the trail of coins and making it difficult to trace stolen money. Alternatively, they are sold as anonymisation tools, that protects the users anonymity by merging their coins with others.

Moser et al. [103] explored mixing services designed to anonymise Bitcoin transactions. They sent coins to three services and used graph analytics to reverse-engineer the functionality of service, to determine whether they could back-trace their coins. Two of the three services were revealed to be providing some level of anonymity, as the researchers were unable to trace back their coins. The services from *blockchain.info* and *bitcoin.fog* were found to aggregate small transactions into large bundles before performing any payments. However, using *bitlaundry* they were able to find a connection, but only within a small amount of one of their transactions. This service did not bundle transactions, and instead used user inputs to pay back other users, not delivering on the privacy promised. As researchers were only able to identify a small amount of transactions from one service, this highlights the challenges faced by current anti-money laundering techniques when only using graph analytics. The study is very limited, with the researchers only covering three services with a small number of transactions over a short period of time. It might have been more effective if the researchers had sent more transactions into each of the relevant services over a longer time period.

Wegberg et al. [144] extended the literature by analysing the usability and cash-out effectiveness of mixing services and exchanges. In their experiment, they sent

coins to five mixing services and three of the five mixers stole the coins. By tracing their stolen coins, they identified that two mixers combined their stolen coins in the same transaction after the experiment had concluded, indicating that they might have been collaborating or were the same entity. The two working mixers successfully delivered their coins with no taint, meaning that there was no linkage found between the coins sent and the coins received. They concluded that for smaller denominations the mixers offered a cost effective and user friendly service with a minimised risk of being scammed if the user had read reviews prior to using the service.

Chapter 4

An Empirical Analysis of Anonymity in Zcash

4.1 Overview

Since the introduction of Bitcoin in 2008 [105], cryptocurrencies have become increasingly popular to the point of reaching a near-mania, with thousands of deployed cryptocurrencies now collectively attracting trillions of dollars in investment. While the broader positive potential of “blockchain” (i.e., the public decentralized ledger underlying almost all cryptocurrencies) is still unclear, despite the growing number of legitimate users there are today still many people using these cryptocurrencies for less legitimate purposes. These range from the purchase of drugs or other illicit goods on so-called dark markets such as Dream Market, to the payments from victims in ransomware attacks such as WannaCry, with many other crimes in between. Criminals engaged in these activities may be drawn to Bitcoin due to the relatively low friction of making international payments using only pseudonyms as identifiers, but the public nature of its ledger of transactions raises the question of how much anonymity is actually being achieved.

Indeed, a long line of research [8, 95, 122, 124, 136] has by now demonstrated that the use of pseudonymous addresses in Bitcoin does not provide any meaningful level of anonymity. Beyond academic research, companies now provide analysis of the Bitcoin blockchain as a business [47]. This type of analysis was used in

several arrests associated with the takedown of Silk Road [48], and to identify the attempts of the WannaCry hackers to move their ransom earnings from Bitcoin into Monero [42].

Perhaps in response to this growing awareness that most cryptocurrencies do not have strong anonymity guarantees, a number of alternative cryptocurrencies or other privacy-enhancing techniques have been deployed with the goal of improving on these guarantees. The most notable cryptocurrencies that fall into this former category are Dash [41] (launched in January 2014), Monero [101] (April 2014), and Zcash [156] (October 2016). At the time of this writing all have a market capitalization of over 1 billion USD [39], although this figure is notoriously volatile, so it should be taken with a grain of salt.

Even within this category of privacy-enhanced cryptocurrencies, and despite its relative youth, Zcash stands somewhat on its own. From an academic perspective, Zcash is backed by highly regarded research [15, 97], and thus comes with seemingly strong anonymity guarantees. Indeed, the original papers cryptographically prove the security of the main privacy feature of Zcash (known as the *shielded pool*), in which users can spend shielded coins without revealing which coins they have spent. These strong guarantees have attracted at least some criminal attention to Zcash: the underground marketplace AlphaBay was on the verge of accepting it before their shutdown in July 2017 [5], and the Shadow Brokers hacking group started accepting Zcash in May 2017 (and in fact for their monthly dumps accepted exclusively Zcash in September 2017) [27].

Despite these theoretical privacy guarantees, the deployed version of Zcash does not require all transactions to take place within the shielded pool itself: it also supports so-called *transparent* transactions, which are essentially the same as transactions in Bitcoin in that they reveal the pseudonymous addresses of both the senders and recipients, and the amount being sent. It does require, however, that all newly generated coins pass through the shielded pool before being spent further, thus ensuring that all coins have been shielded at least once. This requirement led the Zcash developers to conclude that the anonymity set for users spending

shielded coins is in fact all generated coins, and thus that “the mixing strategies that other cryptocurrencies use for anonymity provide a rather small [anonymity set] in comparison to Zcash” and that “Zcash has a distinct advantage in terms of transaction privacy” [159].

In this Chapter, we provide the first in-depth empirical analysis of anonymity in Zcash, in order to examine these claims and more generally provide a longitudinal study of how Zcash has evolved and who its main participants are. We begin in Section 4.3 by providing a general examination of the Zcash blockchain, from which we observe that the vast majority of Zcash activity is in the transparent part of the blockchain, meaning it does not engage with the shielded pool at all. In Section 4.4, we explore this aspect of Zcash by adapting the analysis that has already been developed for Bitcoin, and find that exchanges typically dominate this part of the blockchain.

We then move in Section 4.5 to examining interactions with the shielded pool. We find that, unsurprisingly, the main actors doing so are the founders and miners, who are required to put all newly generated coins directly into it. Using newly developed heuristics for attributing transactions to founders and miners, we find that 65.6% of the value withdrawn from the pool can be linked back to deposits made by either founders or miners. We also implement a general heuristic for linking together other types of transactions, and capture an additional 3.5% of the value using this. Our relatively simple heuristics thus reduce the size of the overall anonymity set by 69.1%.

In Section 4.6, we then look at the relatively small percentage of transactions that have taken place within the shielded pool. Here, we find (perhaps unsurprisingly) that relatively little information can be inferred, although we do identify certain patterns that may warrant further investigation. Finally, we perform a small case study of the activities of the Shadow Brokers within Zcash in Section 4.7, and in Section 4.9 we conclude.

All of our results have been disclosed, at the time of the work’s submission, to the creators of Zcash, and discussed extensively with them since. This has resulted

in changes to both their public communication about Zcash’s anonymity as well as the transactional behavior of the founders. Additionally, all the code for our analysis is available as an open-source repository.¹

4.2 Background

In this section we describe four types of participants who interact in the Zcash network.

Founders took part in the initial creation and release of Zcash, and will receive 20% of all newly generated coins (currently 2.5 ZEC out of the 12.5 ZEC block reward). The founder addresses are specified in the Zcash chain parameters [158].

Miners take part in the maintenance of the ledger, and in doing so receive newly generated coins (10 out of the 12.5 ZEC block reward), as well as any fees from the transactions included in the blocks they mine. Many miners choose not to mine on their own, but join a mining pool; a list of mining pools can be found in Table 4.4. One or many miners win each block, and the first transaction in the block is a *coin generation* (coingen) that assigns newly generated coins to their address(es), as well as to the address(es) of the founders.

Services are entities that accept ZEC as some form of payment. These include exchanges like Bitfinex, which allow users to trade fiat currencies and other cryptocurrencies for ZEC (and vice versa), and platforms like ShapeShift [155], which allow users to trade within cryptocurrencies and other digital assets without requiring registration.

Finally, users are participants who hold and transact in ZEC at a more individual level. In addition to regular individuals, this category includes charities and other organizations that may choose to accept donations in Zcash. A notable user is the Shadow Brokers, a hacker group who have published several leaks containing hacking tools from the NSA and accept payment in Zcash. We explore their usage of Zcash in Section 4.7.

¹<https://github.com/manganese/zcash-empirical-analysis>

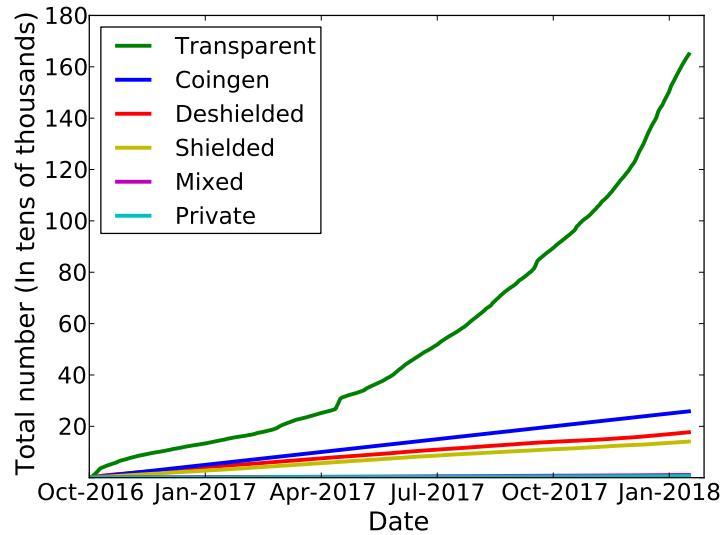


Figure 4.1: The total number of each of the different types of transactions over time.

4.3 General Blockchain Statistics

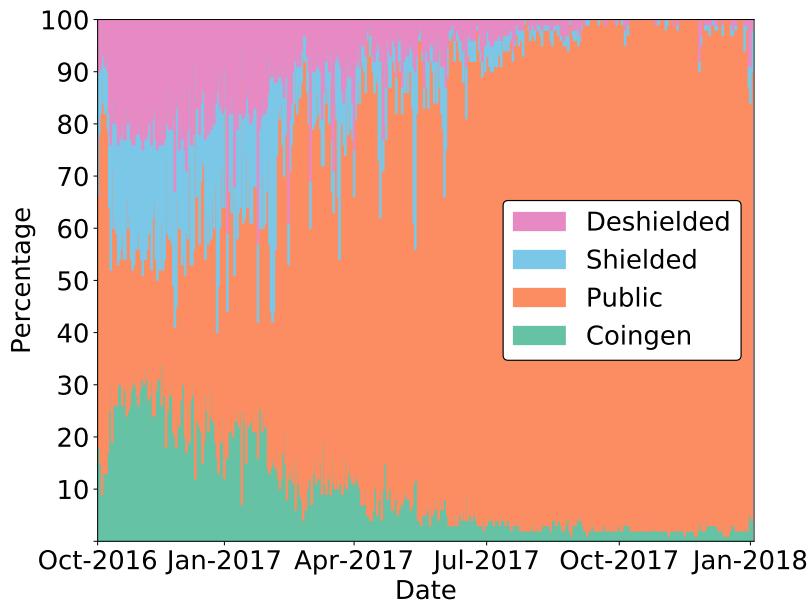
We used the `zcashd` client to download the Zcash blockchain, and loaded a database representation of it into Apache Spark. We then performed our analysis using a custom set of Python scripts equipped with PySpark. We last parsed the block chain on January 21 2018, at which point 258,472 blocks had been mined. Overall, 3,106,643 ZEC had been generated since the genesis block, out of which 2,485,461 ZEC went to the miners and the rest (621,182 ZEC) went to the founders.

4.3.1 Transactions

Across all blocks, there were 2,242,847 transactions. A complete breakdown of the transaction types is in Table 4.1, and graphs depicting the growth of each transaction type over time are in Figures 4.1 and 4.2.² The vast majority of transactions are public (i.e., either transparent or a coin generation). Of the transactions that do interact with the pool (335,630, or 14.96%, in total), only a very small percentage are private transactions; i.e., transactions within the pool. Looking at the types of transactions over time in Figure 4.1, we can see that the number of coingen, shielded, and deshielded transactions all grow in an approximately linear fashion. As we explore in Section 4.5.2, this correlation is due largely to the habits of the miners. Looking at both this figure and Figure 4.2, we can see that while the number

²We use the term ‘mixed’ to mean transactions that have both a `vIn` and a `vOut`, and a `vJoinSplit`.

Type	Number	Percentage
Transparent	1 648 745	73.5
Coingen	258 472	11.5
Deshielded	177 009	7.9
Shielded	140 796	6.3
Mixed	10 891	0.5
Private	6 934	0.3

Table 4.1: The total number of each transaction type.**Figure 4.2:** The fraction of the value in each block representing each different type of transaction over time, averaged daily. Here, ‘public’ captures both transparent transactions and the visible components of mixed transactions.

of transactions interacting with the pool has grown in a relatively linear fashion, the value they carry has over time become a very small percentage of all blocks, as more mainstream (and thus transparent) usage of Zcash has increased.

4.3.2 Addresses

Across all transactions, there have been 1,740,378 distinct t-addresses used. Of these, 8,727 have ever acted as inputs in a t-to-z transaction and 330,780 have ever acted as outputs in a z-to-t transaction. As we explore in Section 4.5.2, much of this asymmetry is due to the behavior of mining pools, which use a small number of addresses to collect the block reward, but a large number of addresses (representing all the individual miners) to pay out of the pool. Given the nature of the shielded

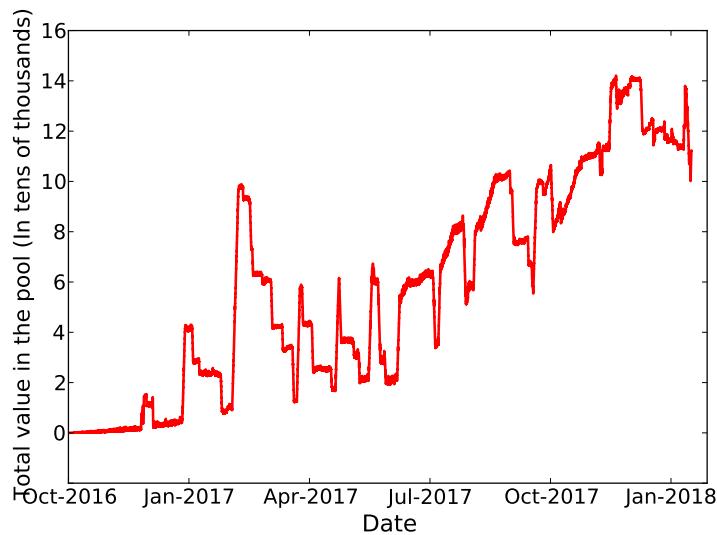


Figure 4.3: The total value in the shielded pool over time, in tens of thousands of ZEC.

pool, it is not possible to know the total number of z-addresses used.

Figure 4.3 shows the total value in the pool over time. Although the overall value is increasing over time, there are certain shielding and de-shielding patterns that create spikes. As we explore in Section 4.5, these spikes are due largely to the habits of the miners and founders. At the time of writing, there are 112,235 ZEC in the pool, or 3.6% of the total monetary supply.

If we rank addresses by their wealth, we first observe that only 25% of all t-addresses have a non-zero balance. Of these, the top 1% hold 78% of all ZEC. The address with the highest balance had 118,257.75 ZEC, which means the richest address has a higher balance than the entire shielded pool.

4.4 T-Address Clustering

As discussed in Section 4.3, a large proportion of the activity on Zcash does not use the shielded pool. This means it is essentially identical to Bitcoin, and thus can be de-anonymized using the same techniques discussed for Bitcoin in Section 2.2.

4.4.1 Clustering addresses

To identify the usage of transparent addresses, we begin by recalling the “multi-input” heuristic for clustering Bitcoin addresses. In this heuristic, addresses that are used as inputs to the same transaction are assigned to the same cluster. In Bitcoin,

this heuristic can be applied to all transactions, as they are all transparent. In Zcash, we perform this clustering as long as there are multiple input t-addresses.

Heuristic 1. If two or more t-addresses are inputs in the same transaction (whether that transaction is transparent, shielded, or mixed), then they are controlled by the same entity.

In terms of false positives, we believe that these are at least as unlikely for Zcash as they are for Bitcoin, as Zcash is a direct fork of Bitcoin and the standard client has the same behavior. In fact, we are not aware of any input-mixing techniques like CoinJoin [91] for Zcash, so could argue that the risk of false positives is even lower than it is for Bitcoin. As this heuristic has already been used extensively in Bitcoin, we thus believe it to be realistic for use in Zcash.

We implemented this heuristic by defining each t-address as a node in a graph, and adding an (undirected) edge in the graph between addresses that had been input to the same transaction. The connected components of the graph then formed the clusters, which represent distinct entities controlling potentially many addresses. The result was a set of 560,319 clusters, of which 97,539 contained more than a single address.

As in Bitcoin, using just this one heuristic is already quite effective but does not capture the common usage of *change addresses*, in which a transaction sends coins to the actual recipient but then also sends any coins left over in the input back to the sender. Meiklejohn et al. [95] use in their analysis a heuristic based on this behavior, but warn that it is somewhat fragile. Indeed, their heuristic seems largely dependent on the specific behavior of several large Bitcoin services, so we chose not to implement it in its full form. Nevertheless, we did use a related Zcash-specific heuristic in our case study of the Shadow Brokers in Section 4.7.

Heuristic 2. If one (or more) address is an input t-address in a vJoinSplit transaction and a second address is an output t-address in the same vJoinSplit transaction, then if the size of $zOut$ is 1 (i.e., this is the only transparent output address), the second address belongs to the same user who controls the input addresses.

To justify this heuristic, we observe that users may not want to deposit all of the coins in their address when putting coins into the pool, in which case they will have to make change. The only risk of a false positive is if users are instead sending money to two separate individuals, one using a z-address and one using a t-address. One notable exception to this rule is users of the zcash4win wallet. Here, the address of the wallet operator is an output t-address if the user decides to pay the developer fee, so it would produce exactly this type of transaction for users putting money into the shielded pool. This address is identifiable, however, so these types of transactions can be omitted from our analysis. Nevertheless, due to concerns about the safety of this heuristic (i.e., its ability to avoid false positives), we chose not to incorporate it into our general analysis below.

4.4.2 Tagging addresses

Having now obtained a set of clusters, we next sought to assign names to them. To accomplish this, we performed a scaled-down version of the techniques used by Meiklejohn et al. [95]. In particular, given that Zcash is still relatively new, there are not many different types of services that accept Zcash. We thus restricted ourselves to interacting with exchanges.

We first identified the top ten Zcash exchanges according to volume traded [39]. We then created an account with each exchange and deposited a small quantity of ZEC into it, tagging as we did the output t-addresses in the resulting transaction as belonging to the exchange. We then withdrew this amount to our own wallet, and again tagged the t-addresses (this time on the sender side) as belonging to the exchange. We occasionally did several deposit transactions if it seemed likely that doing so would tag more addresses. Finally, we also interacted with ShapeShift, which as mentioned in Section 5.2 allows users to move amongst cryptocurrencies without the need to create an account. Here we did a single “shift” into Zcash and a single shift out. A summary of our interactions with all the different exchanges is in Table 4.2.

Finally, we collected the publicized addresses of the founders [158], as well as addresses from known mining pools. For the latter we started by scraping the tags of

Service	Cluster	# deposits	# withdrawals
Binance	7	1	1
Bitfinex	3	4	1
Bithumb	14	2	1
Bittrex	1	1	1
Bit-z	30	2	1
Exmo	4	2	1
HitBTC	18	1	1
Huobi	26	2	1
Kraken	12	1	1
Poloniex	0	1	1
ShapeShift	2	1	1
zcash4win	139	1	2

Table 4.2: The services we interacted with, the identifier of the cluster they were associated with after running Heuristic 1, and the number of deposits and withdrawals we did with them. The first ten are exchanges, ShapeShift is an inter-cryptocurrency exchange, and zcash4win is a Windows-based Zcash client.

these addresses from the Zchain explorer [160]. We then validated them against the blocks advertised on some of the websites of the mining pools themselves (which we also scraped) to ensure that they were the correct tags; i.e., if the recipient of the coingen transaction in a given block was tagged as belonging to a given mining pool, then we checked to see that the block had been advertised on the website of that mining pool. We then augmented these sets of addresses with the addresses tagged as belonging to founders and miners according to the heuristics developed in Section 4.5. We present these heuristics in significantly more detail there, but they resulted in us tagging 123 founder addresses and 110,918 miner addresses (belonging to a variety of different pools).

4.4.3 Results

As mentioned in Section 4.4.1, running Heuristic 1 resulted in 560,319 clusters, of which 97,539 contained more than a single address. We assigned each cluster a unique identifier, ordered by the number of addresses in the cluster, so that the biggest cluster had identifier 0.

4.4.3.1 Exchanges and wallets

As can be seen in Table 4.2, many of the exchanges are associated with some of the biggest clusters, with four out of the top five clusters belonging to popular exchanges. In general, we found that the top five clusters accounted for 11.21% of all transactions. Identifying exchanges is important, as it makes it possible to discover where individual users may have purchased their ZEC. Given existing and emerging regulations, they are also the one type of participant in the Zcash ecosystem that might know the real-world identity of users.

In many of the exchange clusters, we also identified large fractions of addresses that had been tagged as miners. This implies that individual miners use the addresses of their exchange accounts to receive their mining reward, which might be expected if their goal is to cash out directly. We found some, but far fewer, founder addresses at some of the exchanges as well.

Our clustering also reveals that ShapeShift (Cluster 2) is fairly heavily used: it had received over 1.1M ZEC in total and sent roughly the same. Unlike the exchanges, its cluster contained a relatively small number of miner addresses (54), which fits with its usage as a way to shift money, rather than hold it in a wallet.

4.4.3.2 Mining pools and founders

Although mining pools and founders account for a large proportion of the activity in Zcash (as we explore in Section 4.5), many re-use the same small set of addresses frequently, so do not belong to large clusters. For example, Flypool had three single-address clusters while Coinotron, coinmine.pl, Slushpool and Nanopool each had two single-address clusters. (A list of mining pools can be found in Table 4.4 in Section 4.5.2). Of the coins that we saw sent from clusters associated with mining pools, 99.8% of it went into the shielded pool, which further validates both our clustering and tagging techniques.

4.4.3.3 Philanthropists

Via manual inspection, we identified three large organizations that accept Zcash donations: the Internet Archive, torservers.net, and Wikileaks. Of these,

[torservers.net](#) accepts payment only via a z-address, so we cannot identify their transactions (Wikileaks accepts payment via a z-address too, but also via a t-address). Of the 31 donations to the Internet Archive that we were able to identify, which totaled 17.3 ZEC, 9 of them were made anonymously (i.e., as z-to-t transactions). On the other hand, all of the 20 donations to Wikileaks t-address were made as t-to-t transactions. None of these belong to clusters, as they have never sent a transaction.

Most of the donations are small quantities of ZEC. For example, the transparent donations to wikileaks ranged between 0.00065 ZEC and 1.4 ZEC with a median donation of 0.035 ZEC.

4.5 Interactions with the Shielded Pool

What makes Zcash unique is of course not its t-addresses (since these essentially replicate the functionality of Bitcoin), but its shielded pool. To that end, this section explores interactions with the pool at its endpoints, meaning the deposits into (t-to-z) and withdrawals out of the pool (z-to-t). We then explore interactions within the pool (z-to-z transactions) in Section 4.6.

To begin, we consider just the amounts put into and taken out of the pool. Over time, 3,901,124 ZEC have been deposited into the pool,³ and 3,788,889 have been withdrawn. Figure 4.4 plots both deposits and withdrawals over time.

This figure shows a near-perfect reflection of deposits and withdrawals, demonstrating that most users not only withdraw the exact number of ZEC they deposit into the pool, but do so very quickly after the initial deposit. As we see in Sections 4.5.1 and 4.5.2, this phenomenon is accounted for almost fully by the founders and miners. Looking further at the figure, we can see that the symmetry is broken occasionally, and most notably in four “spikes”: two large withdrawals, and two large deposits. Some manual investigation revealed the following:

“The early birds” The first withdrawal spike took place at block height 30,900, which was created in December 2016. The cause of the spike was a single trans-

³This is greater than the total number of generated coins, as all coins must be deposited into the pool at least once, by the miners or founders, but may then go into and out of the pool multiple times.

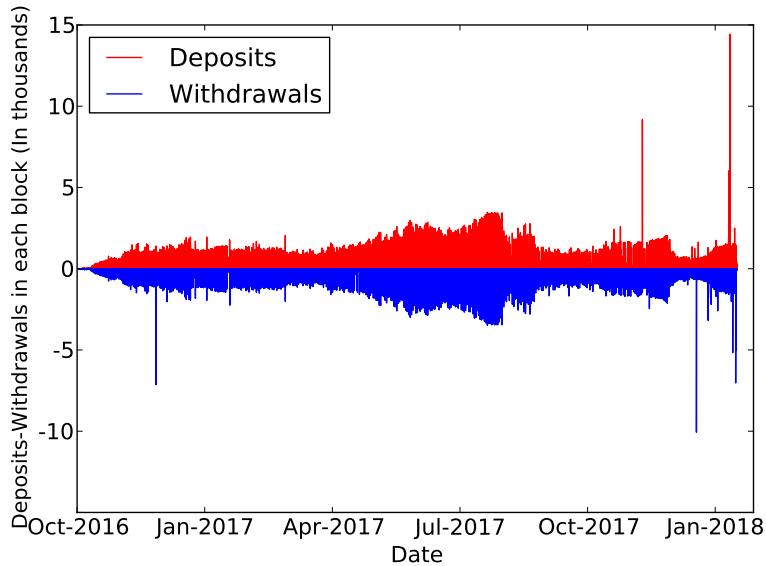


Figure 4.4: Over time, the amount of ZEC put into the shielded pool (in red) and the amount taken out of the pool (in blue).

action in which 7,135 ZEC was taken out of the pool; given the exchange rate at that time of 34 USD per ZEC, this was equivalent to 242,590 USD. The coins were distributed across 15 t-addresses, which initially we had not tagged as belonging to any named user. After running the heuristic described in Section 4.5.1, however, we tagged all of these addresses as belonging to founders. In fact, this was the very first withdrawal that we identified as being associated with founders.

“Secret Santa” The second withdrawal spike took place on December 25 2017, at block height 242,642. In it, 10,000 ZEC was distributed among 10 different t-addresses, each receiving 1,000 ZEC. None of these t-addresses had done a transaction before then, and none have been involved in one since (i.e., the coins received in this transaction have not yet been spent).

“One-man wolf packs” Both of the deposit spikes in the graph correspond to single large deposits from unknown t-addresses that, using our analysis from Section 4.4, we identified as residing in single-address clusters. For the first spike, however, many of the deposited amounts came directly from a founder address identified by our heuristics (Heuristic 3), so given our analysis in Section 4.5.1 we believe this may also be associated with the founders.

While this figure already provides some information about how the pool is

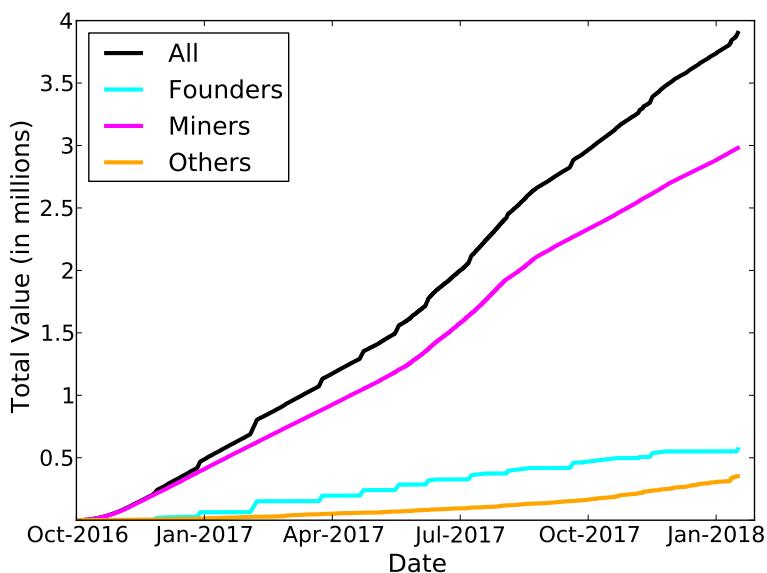


Figure 4.5: Over time, the amount of ZEC deposited into the shielded pool by miners, founders, and others.

used (namely that most of the money put into it is withdrawn almost immediately afterwards), it does not tell us who is actually using the pool. For this, we attempt to associate addresses with the types of participants identified in Section 4.2: founders, miners, and ‘other’ (encompassing both services and individual users).

When considering deposits into the shielded pool, it is easy to associate addresses with founders and miners, as the consensus rules dictate that they must put their block rewards into the shielded pool before spending them further.

As described in Section 4.4.2, we tagged founders according to the Zcash parameters, and tagged as miners all recipients of coingen transactions that were not founders. We then used these tags to identify a founder deposit as any t-to-z transaction using one or more founder addresses as input, and a miner deposit as any t-to-z transaction using one or more miner addresses as input. The results are in Figure 4.5.

Looking at this figure, it is clear that miners are the main participants putting money into the pool. This is not particularly surprising, given that all the coins they receive must be deposited into the pool at least once, so if we divide that number of coins by the total number deposited we would expect at least 63.7% of the deposits to come from miners. (The actual number is 76.7%.) Founders, on the other hand,

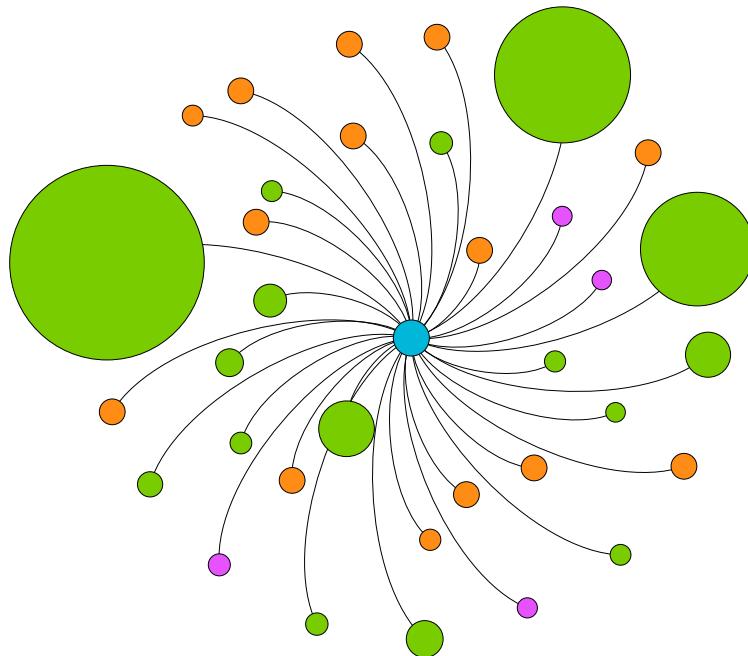


Figure 4.6: The addresses that have put more than 10,000 ZEC into the shielded pool over time, where the size of each node is proportional to the value it has put into the pool. The addresses of miners are green, of founders are orange, and of unknown ‘other’ participants are purple.

don’t put as much money into the pool (since they don’t have as much to begin with), but when they do they put in large amounts that cause visible step-like fluctuations to the overall line.

In terms of the heaviest users, we looked at the individual addresses that had put more than 10,000 ZEC into the pool. The results are in Figure 4.6.

In fact, this figure incorporates the heuristics we develop in Sections 4.5.1 and 4.5.2, although it looked very similar when we ran it before applying our heuristics (which makes sense, since our heuristics mainly act to link z-to-t transactions). Nevertheless, it demonstrates again that most of the heavy users of the pool are miners, with founders also depositing large amounts but spreading them over a wider variety of addresses. Of the four ‘other’ addresses, one of them belonged to ShapeShift, and the others belong to untagged clusters.

While it is interesting to look at t-to-z transactions on their own, the main intention of the shielded pool is to provide an anonymity set, so that when users withdraw their coins it is not clear whose coins they are. In that sense, it is much more interesting to link together t-to-z and z-to-t transactions, which acts to reduce

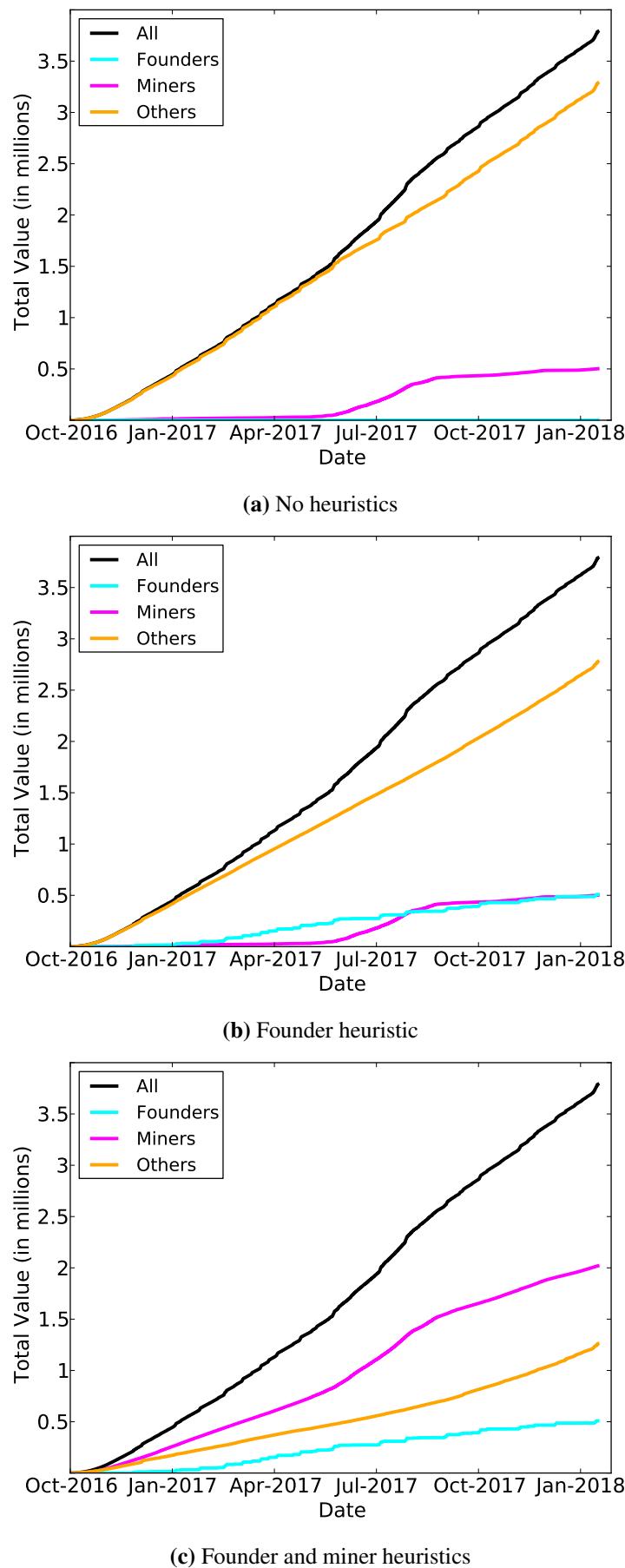


Figure 4.7: The z-to-t transactions we associated with miners, founders, and ‘other’, after running some combination of our heuristics, in millions of transactions.

the anonymity set. More concretely, if a t-to-z transaction can be linked to a z-to-t transaction, then those coins can be “ruled out” of the anonymity set of future users withdrawing coins from the pool. We thus devote our attention to this type of analysis for the rest of the section.

The most naïve way to link together these transactions would be to see if the same addresses are used across them; i.e., if a miner uses the same address to withdraw their coins as it did to deposit them. By running this simple form of linking, we see the results in Figure 4.7a. This figure shows that we are not able to identify any withdrawals as being associated with founders, and only a fairly small number as associated with miners: 49,280 transactions in total, which account for 13.3% of the total value in the pool.

Nevertheless, using heuristics that we develop for identifying founders (as detailed in Section 4.5.1) and miners (Section 4.5.2), we are able to positively link most of the z-to-t activity with one of these two categories, as seen in Figures 4.7b and 4.7c. In the end, of the 177,009 z-to-t transactions, we were able to tag 120,629 (or 68%) of them as being associated with miners, capturing 52.1% of the value coming out of the pool, and 2,103 of them as being associated with founders (capturing 13.5% of the value). We then examine the remaining 30-35% of the activity surrounding the shielded pool in Section 4.5.3.

4.5.1 Founders

After comparing the list of founder addresses against the outputs of all coingen transactions, we found that 14 of them had been used. Using these addresses, we were able to identify founder deposits into the pool, as already shown in Figure 4.5. Table 4.3 provides a closer inspection of the usage of each of these addresses.

This table shows some quite obvious patterns in the behavior of the founders. At any given time, only one address is “active,” meaning it receives rewards and deposits them into the pool. Once it reaches the limit of 44,272.5 ZEC, the next address takes its place and it is not used again. This pattern has held from the third address onwards. What’s more, the amount deposited was often the same: exactly 249.9999 ZEC, which is roughly the reward for 100 blocks. This was true of 74.9%

	# Deposits	Total value	# Deposits (249)
1	548	19 600.4	0
2	252	43 944.6	153
3	178	44 272.5	177
4	192	44 272.5	176
5	178	44 272.5	177
6	178	44 272.5	177
7	178	44 272.5	177
8	178	44 272.5	177
9	190	44 272.5	176
10	188	44 272.5	176
11	190	44 272.5	176
12	178	44 272.5	177
13	191	44 272.5	175
14	70	17 500	70
Total	2889	568 042.5	2164

Table 4.3: The behaviour of each of the 14 active founder addresses, in terms of the number of deposits into the pool, the total value deposited (in ZEC), and the number of deposits carrying exactly 249.9999 ZEC in value.

of all founder deposits, and 96.2% of all deposits from the third address onwards. There were only ever five other deposits into the pool carrying value between 249 and 251 ZEC (i.e., carrying a value close but not equal to 249.9999 ZEC).

Thus, while we were initially unable to identify any withdrawals associated with the founders (as seen in Figure 4.7a), these patterns indicated an automated use of the shielded pool that might also carry into the withdrawals. Upon examining the withdrawals from the pool, we did not find any with a value exactly equal to 249.9999 ZEC. We did, however, find 1,953 withdrawals of exactly 250.0001 ZEC (and 1,969 carrying a value between 249 and 251 ZEC, although we excluded the extra ones from our analysis).

The value alone of these withdrawals thus provides some correlation with the deposits, but to further explore it we also looked at the timing of the transactions. When we examined the intervals between consecutive deposits of 249.9999 ZEC, we found that 85% happened within 6-10 blocks of the previous one. Similarly, when examining the intervals between consecutive withdrawals of 250.0001 ZEC, we found that 1,943 of the 1,953 withdrawals also had a proximity of 6-10 blocks. Indeed, both the deposits and the withdrawals proceeded in step-like patterns, in

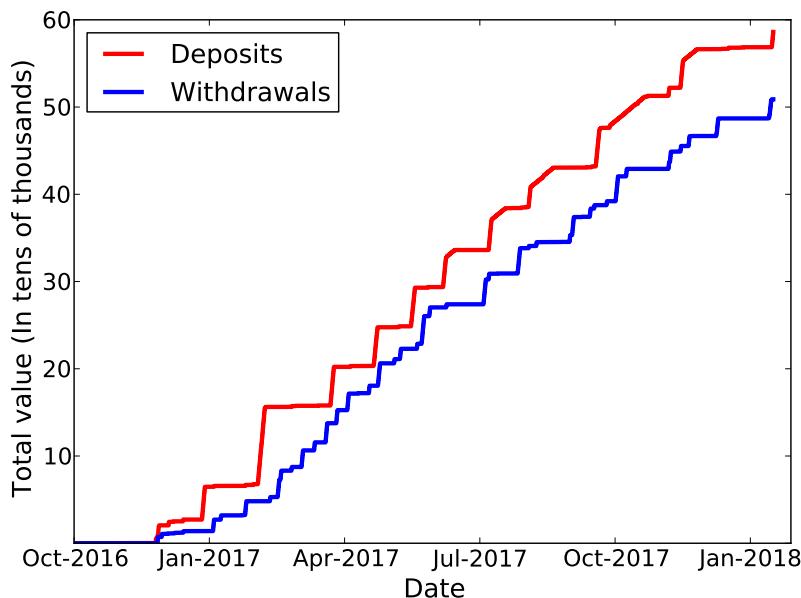


Figure 4.8: Over time, the founder deposits into the pool (in red) and withdrawals from the pool (in blue), after running Heuristic, in tens of thousands of transactions 3.

which many transactions were made within a very small number of blocks (resulting in the step up), at which point there would be a pause while more block rewards were accumulated (the step across). This pattern is visible in Figure 4.8, which shows the deposit and withdrawal transactions associated with the founders. Deposits are typically made in few large steps, whereas withdrawals take many smaller ones.

Heuristic 3. Any z-to-t transaction carrying 250.0001 ZEC in value is done by the founders.

In terms of false positives, we cannot truly know how risky this heuristic is, short of asking the founders. This is in contrast to the t-address clustering heuristics presented in Section 4.4, in which we were not attempting to assign addresses to a specific owner, so could validate the heuristics in other ways. Nevertheless, the high correlation between both the value and timing of the transactions led us to believe in the reliability of this heuristic.

As a result of running this heuristic, we added 75 more addresses to our initial list of 48 founder addresses (of which, again, only 14 had been used). Aside from the correlation showed in Figure 4.8, the difference in terms of our ability to tag founder withdrawals is seen in Figure 4.7b.

Name	Addresses	t-to-z	z-to-t
Flypool	3	65631	3
F2Pool	1	742	720
Nanopool	2	8319	4107
Suprnova	1	13361	0
Coinmine.pl	2	3211	0
Waterhole	1	1439	5
BitClub Pool	1	196	1516
MiningPoolHub	1	2625	0
Dwarfpool	1	2416	1
Slushpool	1	941	0
Coinotron	2	9726	0
Nicehash	1	216	0
MinerGate	1	13	0
Zecmine.pro	1	6	0

Table 4.4: A summary of our identified mining pool activity, in terms of the number of associated addresses used in coingen transactions, and the numbers of each type of transaction interacting with the pool.

4.5.2 Miners

The Zcash protocol specifies that all newly generated coins are required to be put into the shielded pool before they can be spent further. As a result, we expect that a large quantity of the ZEC being deposited into the pool are from addresses associated with miners.

4.5.2.1 Deposits

As discussed earlier and seen in Figure 4.5, it is easy to identify miner deposits into the pool due to the fact that they immediately follow a coin generation. Before going further, we split the category of miners into individual miners, who operate on their own, and mining pools, which represent collectives of potentially many individuals. In total, we gathered 19 t-addresses associated with Zcash mining pools, using the scraping methods described in Section 4.4.2. Table 4.4 lists these mining pools, as well as the number of addresses they control and the number of t-to-z transactions we associated with them. Figure 4.9 plots the value of their deposits into the shielded pool over time.

In this figure, we can clearly see that the two dominant mining pools are Flypool and F2Pool. Flypool consistently deposits the same (or similar) amounts,

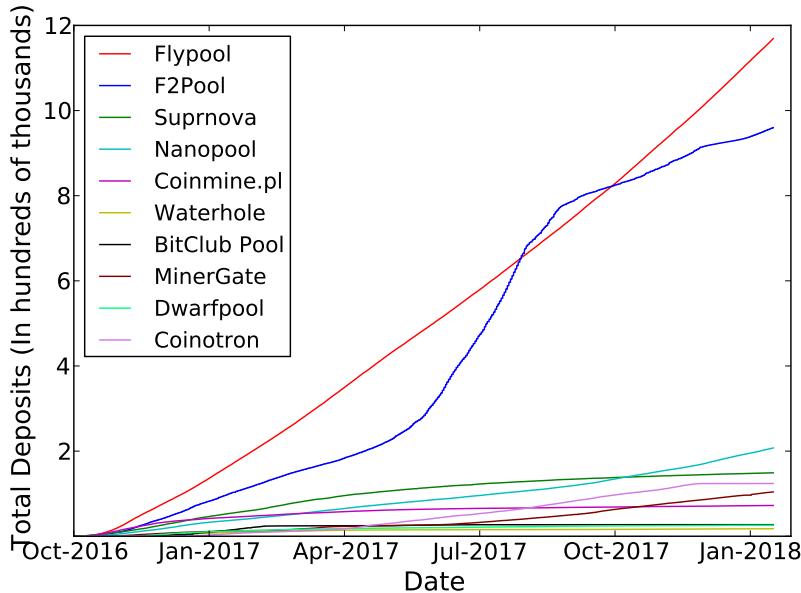


Figure 4.9: Over time, the value of deposits made by known mining pools into the shielded pool, in hundreds of thousands of transactions.

which we can see in their linear representation. F2Pool, on the other hand, has bursts of large deposits mixed with periods during which it is not very active, which we can also see reflected in the graph. Despite their different behaviors, the amount deposited between the two pools is similar.

4.5.2.2 Withdrawals

While the withdrawals from the pool do not solely re-use the small number of mining addresses identified using deposits (as we saw in our naïve attempt to link miner z-to-t transactions in Figure 4.7a), they do typically re-use some of them, so can frequently be identified anyway.

In particular, mining pool payouts in Zcash are similar to how many of them are in Bitcoin [46, 95]. The block reward is often paid into a single address, controlled by the operator of the pool, and the pool operator then deposits some set of aggregated block rewards into the shielded pool. They then pay the individual reward to each of the individual miners as a way of “sharing the pie,” which results in z-to-t transactions with many outputs. (In Bitcoin, some pools opt for this approach while some form a “peeling chain” in which they pay each individual miner in a separate transaction, sending the change back to themselves each time.) In the payouts for some of the mining pools, the list of output t-addresses sometimes includes

one of the t-addresses known to be associated with the mining pool already. We thus tag these types of payouts as belonging to the mining pool, according to the following heuristic:

Heuristic 4. If a z-to-t transaction has over 100 output t-addresses, one of which belongs to a known mining pool, then we label the transaction as a mining withdrawal (associated with that pool), and label all non-pool output t-addresses as belonging to miners.

As with Heuristic 3, short of asking the mining pool operators directly it is impossible to validate this heuristic. Nevertheless, given the known operating structure of Bitcoin mining pools and the way this closely mirrors that structure, we again believe it to be relatively safe.

As a result of running this heuristic, we tagged 110,918 addresses as belonging to miners, and linked a much more significant portion of the z-to-t transactions, as seen in Figure 4.7c. As the last column in Table 4.4 shows, however, this heuristic captured the activity of only a small number of the mining pools, and the large jump in linked activity is mostly due to the high coverage with F2Pool (one of the two richest pools). This implies that further heuristics developed specifically for other pools, such as Flypool, would increase the linkability even more. Furthermore, a more active strategy in which we mined with the pools to receive payouts would reveal their structure, at which point (according to the 1.1M deposited by Flypool shown in Figure 4.9 and the remaining value of 1.2M attributed to the ‘other’ category shown in Figure 4.7c) we would shrink the anonymity set even further.⁴

4.5.3 Other Entities

Once the miners and founders have been identified, we can assume the remaining transactions belong to more general entities. In this section we look into different means of categorizing these entities in order to identify how the shielded pool is being used.

⁴It is possible that we have already captured some of the Flypool activity, as many of the miners receive payouts from multiple pools. We thus are not claiming that all remaining activity could be attributed to Flypool, but potentially some substantial portion.

In particular, we ran the heuristic due to Quesnelle [119], which said that if a unique value (i.e., a value never seen in the blockchain before or since) is deposited into the pool and then, after some short period of time, the exact same value is withdrawn from the pool, the deposit and the withdrawal are linked in what he calls a *round-trip transaction*.

Heuristic 5. [119] For a value v , if there exists exactly one t-to-z transaction carrying value v and one z-to-t transaction carrying value v , where the z-to-t transaction happened after the t-to-z one and within some small number of blocks, then these transactions are linked.

In terms of false positives, the fact that the value is unique in the blockchain means that the only possibility of a false positive is if some of the z-to-z transactions split or aggregated coins in such a way that another deposit (or several other deposits) of a different amount were altered within the pool to yield an amount identical to the initial deposit. While this is possible in theory, we observe that of the 12,841 unique values we identified, 9,487 of them had eight decimal places (the maximum number in Zcash), and 98.9% of them had more than three decimal places. We thus view it as highly unlikely that these exact values were achieved via manipulations in z-to-z transactions.

By running this heuristic, we identified 12,841 unique values, which means we linked 12,841 transactions. The values total 1,094,513.23684 ZEC and represent 28.5% of all coins ever deposited in the pool. Interestingly, most (87%) of the linked coins were in transactions attributed to the founders and miners, so had already been linked by our previous heuristics. We believe this lends further credence to their soundness. In terms of the block interval, we ran Heuristic 5 for every interval between 1 and 100 blocks; the results are in Figure 4.10.

As this figure shows, even if we assume a conservative block interval of 10 (meaning the withdrawal took place 25 minutes after the deposit), we still capture 70% of the total value, or over 700K ZEC. If we require the withdrawal to have taken place within an hour of the deposit, we get 83%.

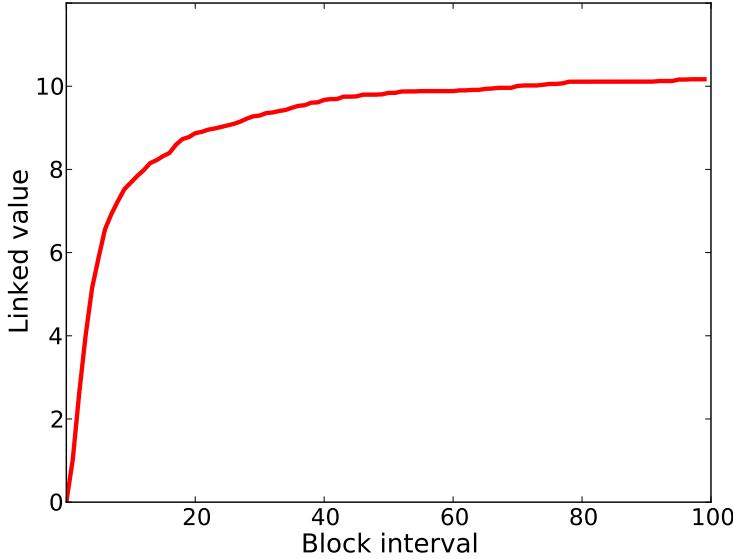


Figure 4.10: The value linked by Heuristic 5, as a function of the block interval required between the deposit and withdrawal transactions.

4.6 Interactions within the Shielded Pool

In this section we consider private transactions; i.e., z-to-z transactions that interact solely with the shielded pool. As seen in Section 4.3.1, these transactions form a small percentage of the overall transactions. However, z-to-z transactions form a crucial part of the anonymity core of Zcash. In particular, they make it difficult to identify the round-trip transactions from Heuristic 5.

Our analysis identified 6,934 z-to-z transactions, with 8,444 vJoinSplits. As discussed in Section 2.3.1, the only information revealed by z-to-z transactions is the miner’s fee, the time of the transaction, and the number of vJoinSplits used as input. Of these, we looked at the time of transactions and the number of vJoinSplits in order to gain some insight as to the use of these operations.

We found that 93% of z-to-z transactions took just one vJoinSplit as input. Since each vJoinSplit can have at most two shielded outputs as its input, the majority of z-to-z transactions thus take no more than two shielded outputs as their input. This increases the difficulty of categorizing z-to-z transactions, because we cannot know if a small number of users are making many transactions, or many users are making one transaction.

In looking at the timing of z-to-z transactions, however, we conclude that it is

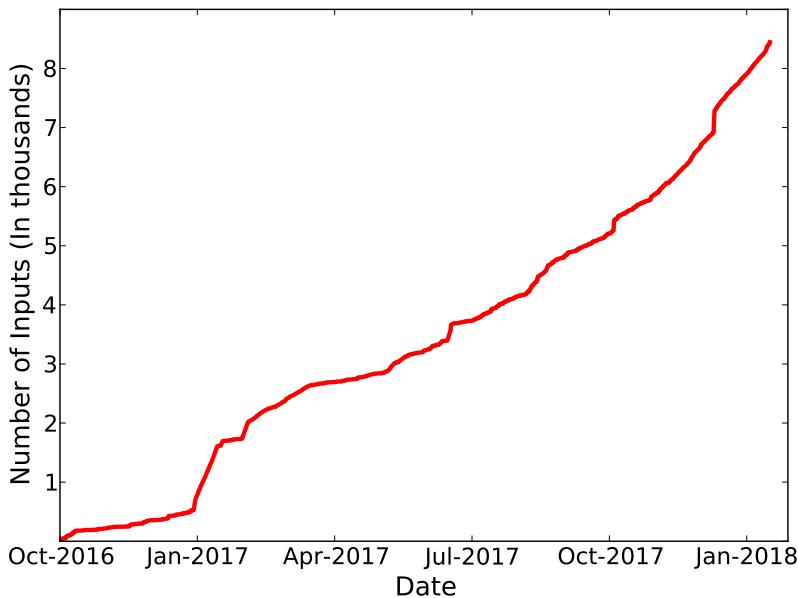


Figure 4.11: The number of z-to-z vJoinSplit transactions over time.

likely that a small number of users were making many transactions. Figure 4.11 plots the cumulative number of vJoinSplits over time. The occurrences of vJoinSplits are somewhat irregular, with 17% of all vJoinSplits occurring in January 2017. There are four other occasions when a sufficient number of vJoinSplits occur within a sufficiently short period of time as to be visibly noticeable. It seems likely that these occurrences belong to the same group of users, or at least by users interacting with the same service.

Finally, looking back at the number of t-to-z and z-to-t transactions identified with mining pools in Table 4.4, it is possible that BitClub Pool is responsible for up to 1,300 of the z-to-z transactions, as it had 196 deposits into the pool and 1,516 withdrawals. This can happen only because either (1) the pool made extra z-to-z transactions, or (2) it sent change from its z-to-t transactions back into the shielded pool. As most of BitClub Pool's z-to-t transactions had over 200 output t-addresses, however, we conclude that the former explanation is more likely.

4.7 Case Study: The Shadow Brokers

The Shadow Brokers (TSB) are a hacker collective that has been active since the summer of 2016, and that leaks tools supposedly created by the NSA. Some of these leaks are released as free samples, but many are sold via auctions and as monthly

May/June	July	August	September	October
100	200	500	100	500
	400		200	
			500	

Table 4.5: Amounts charged for TSB monthly dumps, in ZEC. In July and September TSB offered different prices depending on which exploits were being purchased.

bundles. Initially, TSB accepted payment only using Bitcoin. Later, however, they began to accept Zcash for their monthly dump service. In this section we discuss how we identified t-to-z transactions that could represent payments to TSB. We identified twenty-four clusters (created using our analysis in Section 4.4) matching our criteria for potential TSB customers, one of which could be a regular customer.

4.7.1 Techniques

In order to identify the transactions that are most likely to be associated with TSB, we started by looking at their blog [139]. In May 2017, TSB announced that they would be accepting Zcash for their monthly dump service. Throughout the summer (June through August) they accepted both Zcash and Monero, but in September they announced that they would accept only Zcash. Table 4.5 summarizes the amount they were requesting in each of these months. The last blog post was made in October 2017, when they stated that all subsequent dumps would cost 500 ZEC.

To identify potential TSB transactions, we thus looked at all t-to-z transactions not associated with miners or founders that deposited either 100, 200, 400, or 500 ZEC \pm 5 ZEC. Our assumption was that users paying TSB were not likely to be regular Zcash users, but rather were using it with the main purpose of making the payment. On this basis, addresses making t-to-z transactions of the above values were flagged as a potential TSB customer if the following conditions held:

1. They did not get their funds from the pool; i.e., there were no z-to-t transactions with this address as an output. Again, if this were a user mainly engaging with Zcash as a way to pay TSB, they would need to buy their funds from an exchange, which engage only with t-addresses.
2. They were not a frequent user, in the sense that they had not made or received more than 250 transactions (ever).

3. In the larger cluster in which this address belonged, the total amount deposited by the entire cluster into the pool within one month was within 1 ZEC of the amounts requested by TSB. Here, because the resulting clusters were small enough to treat manually, we applied not only Heuristic 1 but also Heuristic 2 (clustering by change), making sure to weed out false positives. Again, the idea was that suspected TSB customers would not be frequent users of the pool.

As with our previous heuristics, there is no way to quantify the false-positive risks associated with this set of criteria, although we see below that many of the transactions matching it did occur in the time period associated with TSB acceptance of Zcash. Regardless, given this limitation we are not claiming that our results are definitive, but do believe this to be a realistic set of criteria that might be applied in the context of a law enforcement investigation attempting to narrow down potential suspects.

4.7.2 Results

Our results, in terms of the number of transactions matching our requirements above up until 17 January 2018, are summarized in Table 4.6. Before the first TSB blog post in May, we found only a single matching transaction. This is very likely a false positive, but demonstrates that the types of transactions we were seeking were not common before TSB went live with Zcash. After the blog post, we flagged five clusters in May and June for the requested amount of 100 ZEC. There were only two clusters that was flagged for 500 ZEC, one of which was from August. No transactions of any of the required quantities were flagged in September, despite the fact that TSB switched to accepting only Zcash in September. This is possible for a number of reasons: our criteria may have caused us to miss transactions, or maybe there were no takers. From October onwards we flagged between 1-6 transactions per month. It is hard to know if these represent users paying for old data dumps or are simply false positives.

Four out of the 24 transactions in Table 4.6 are highly likely to be false positives. First, there is the deposit of 100 ZEC into the pool in January, before TSB

Month	100	200	400	500
October (2016)	0	0	0	0
November	0	0	0	0
December	0	0	0	0
January (2017)	1	0	0	0
February	0	0	0	0
March	0	0	0	0
April	0	0	0	0
May (before)	0	0	0	0
May (after)	3	1	0	0
June	2	1	1	0
July	1	2	0	0
August	1	0	0	1
September	0	0	0	0
October	2	0	0	0
November	1	0	0	0
December	2	3	0	1
January (2018)	0	1	0	0

Table 4.6: Number of clusters that put the required amounts (± 1 ZEC) into the shielded pool.

announced their first blog post. This cluster put an additional 252 ZEC into the pool in March, so is likely just some user of the pool. Second and third, there are two deposits of 200 ZEC into the pool in June, before TSB announced that one of the July dump prices would cost 200 ZEC. Finally, there is a deposit of 400 ZEC into the pool in June before TSB announced that one of the July dump prices would cost 400 ZEC.

Of the remaining clusters, there is one whose activity is worth discussing. From this cluster, there was one deposit into the pool in June for 100 ZEC, one in July for 200 ZEC, and one in August for 500 ZEC, matching TSB prices exactly. The cluster belonged to a new user, and most of the money in this user's cluster came directly from Bitfinex (Cluster 3).

4.8 Discussion and Future work

Our heuristics would have been significantly less effective if the founders interacting with the pool behaved in a less regular fashion. In particular, by always withdrawing the same amount in the same time interval, it became possible to dis-

tinguish founders withdrawing funds from other users. Given that the founders are both highly invested in the currency and knowledgeable about how to use it in a secure fashion, they are in the best place to ensure the anonymity set is large.

Ultimately, the only way for Zcash to truly ensure the size of its anonymity set is to require all transactions to take place within the shielded pool, or otherwise significantly expand the usage of it. This may soon be computationally feasible given emerging advances in the underlying cryptographic techniques [149], or even if more mainstream wallet providers like Jaxx roll out support for z-addresses. More broadly, we view it as an interesting regulatory question whether or not mainstream exchanges would continue to transact with Zcash if it switched to supporting only z-addresses.

Our study was an initial exploration, and thus left many avenues open for further exploration. For example, it may be possible to classify more z-to-z transactions by analyzing the time intervals between the transactions in more detail, or by examining other metadata such as the miner’s fee or even the size (in bytes) of the transaction. Additionally, the behavior of mining pools could be further identified by a study that actively interacts with them.

4.9 Conclusions

We present the first in-depth exploration of Zcash, with a particular focus on its anonymity guarantees. To achieve this, we applied both well-known clustering heuristics that have been developed for Bitcoin and attribution heuristics we developed ourselves that take into account Zcash’s shielded pool and its unique cast of characters. As with previous empirical analyses of other cryptocurrencies, our study has shown that most users are not taking advantage of the main privacy feature of Zcash at all. Furthermore, the participants who do engage with the shielded pool do so in a way that is identifiable, which has the effect of significantly eroding the anonymity of other users by shrinking the overall anonymity set.

Chapter 5

Tracing Transactions Across Cryptocurrency Ledgers

5.1 Overview

For the past decade, cryptocurrencies such as Bitcoin have been touted for their transformative potential, both as a new form of electronic cash and as a platform to “re-decentralize” aspects of the Internet and computing in general. Traditionally, criminals attempting to cash out illicit funds would have to use exchanges; indeed, most tracking techniques rely on identifying the addresses associated with these exchanges as a way to observe when these deposits happen [95]. Nowadays, however, exchanges typically implement strict Know Your Customer/Anti-Money Laundering (KYC/AML) policies to comply with regulatory requirements, meaning criminals (and indeed all users) risk revealing their real identities when using them. Users also run risks when storing their coins in accounts at custodial exchanges, as exchanges may be hacked or their coins may otherwise become inaccessible [93, 127]. As an alternative, there have emerged in the past few years frictionless trading platforms such as ShapeShift¹ and Changelly,² in which users are able to trade between cryptocurrencies without having to store their coins with the platform provider. Furthermore, while ShapeShift now requires users to have verified accounts [147], this was not the case before October 2018.

¹<https://shapeshift.io>

²<https://changelly.com>

Part of the reason for these trading platforms to exist is the sheer rise in the number of different cryptocurrencies: according to the popular cryptocurrency data tracker CoinMarketCap there were 36 cryptocurrencies in September 2013, only 7 of which had a stated market capitalization of over 1 million USD,³ whereas in January 2019 there were 2117 cryptocurrencies, of which the top 10 had a market capitalization of over 100 million USD. Given this proliferation of new cryptocurrencies and platforms that make it easy to transact across them, it becomes important to consider not just whether or not flows of coins can be tracked within the transaction ledger of a given currency, but also if they can be tracked as coins move across their respective ledgers as well. This is especially important given that there are documented cases of criminals attempting to use these cross-currency trades to obscure the flow of their coins: the WannaCry ransomware operators, for example, were observed using ShapeShift to convert their ransomed bitcoins into Monero [42]. More generally, these services have the potential to offer an insight into the broader cryptocurrency ecosystem and the thousands of currencies it now contains.

In this Chapter, we initiate an exploration of the usage of these cross-currency trading platforms, and the potential they offer in terms of the ability to track flows of coins as they move across different transaction ledgers. Here we rely on three distinct sources of data: the cryptocurrency blockchains, the data collected via our own interactions with these trading platforms, and — as we describe in Section 5.3 — the information offered by the platforms themselves via their public APIs.

We begin in Section 5.4 by identifying the specific on-chain transactions associated with an advertised ShapeShift transaction, which we are able to do with a relatively high degree of success (identifying both the deposit and withdrawal transactions 81.91% of the time, on average). We then describe in Section 5.5 the different transactional patterns that can be traced by identifying the relevant on-chain transactions, focusing specifically on patterns that may be indicative of trading or money laundering, and on the ability to link addresses across different currency

³<https://coinmarketcap.com/historical/20130721/>

ledgers. We then move in Section 5.6 to consider both old and new heuristics for clustering together addresses associated with ShapeShift, with particular attention paid to our new heuristic concerning the common social relationships revealed by the usage of ShapeShift. Finally, we bring all the analysis together by applying it to several case studies in Section 5.7. Again, our particular focus in this last section is on the phenomenon of trading and other profit-driven activity, and the extent to which usage of the ShapeShift platform seems to be motivated by criminal activity or a more general desire for anonymity.

5.2 Background

5.2.1 Digital asset trading platforms

In contrast to a traditional exchange, a digital asset trading platform allows users to move between different cryptocurrencies without needing to set up an account, and thus without needing to follow KYC/AML regulations. Instead, a user approaches the service and selects a supported input currency curln (i.e., the currency from which they would like to move money) and a supported output currency curOut (the currency which they would like to obtain). A user additionally specifies a destination address addr_u in the curOut blockchain, which is the address to which the output currency will be sent. The service then presents the user with an exchange rate rate and an address addr_s in the curln blockchain to which to send money. The user then sends this address the amount amt in curln they wish to convert, and after some delay the service sends the appropriate amount of the output currency to the specified destination address. This means that an interaction with either of these services results in two transactions: one on the curln blockchain sending amt to addr_s , and one on the curOut blockchain sending (roughly) $\text{rate} \cdot \text{amt}$ to addr_u .

This describes an interaction with an abstracted platform; today, the two best-known examples are ShapeShift and Changelly, although Changelly does require account creation. Each platform supports dozens of cryptocurrencies, ranging from better-known ones such as Bitcoin and Ethereum to lesser-known ones such as FirstBlood and Clams. Many of the supported cryptocurrencies actually operate as

ERC20 or BTC tokens, meaning they run as contracts on top of the Ethereum and Bitcoin blockchains, respectively, rather than as their own standalone platforms. In Section 5.3, we describe in more depth the operations of these concrete platforms and our own interactions with them.

5.3 Data Collection and Statistics

In this section, we describe our data sources, as well as some preliminary statistics about the collected data. We begin in Section 5.3.1 by describing our own interactions with Changelly, a trading platform with a limited personal API. We then describe in Section 5.3.2 both our own interactions with ShapeShift, and the data we were able to scrape from their public API, which provided us with significant insight into their overall set of transactions. Finally, we describe in Section 5.3.3 our collection of the data backing eight different cryptocurrencies.

5.3.1 Changelly

Changelly offers a simple API⁴ that allows registered users to carry out transactions with the service. Using this API, we engaged in 22 transactions, using the most popular ShapeShift currencies (Table 5.1) to guide our choices for `curln` and `curOut`.

While doing these transactions, we observed that they would sometimes take up to an hour to complete. This is because Changelly attempts to minimize double-spending risk by requiring users to wait for a set number of confirmations (shown to the user at the time of their transaction) in the `curln` blockchain before executing the transfer on the `curOut` blockchain. We used this observation to guide our choice of parameters in our identification of on-chain transactions in Section 5.4.

5.3.2 ShapeShift

ShapeShift’s API⁵ allows users to execute their own transactions, of which we did 18 in total. As with Changelly, we were able to gain some valuable insights about the operation of the platform via these personal interactions. Whereas ShapeShift did not disclose the number of confirmations they waited for on the `curln`

⁴<https://api-docs.changelly.com/>

⁵<https://info.shapeshift.io/api>

blockchain, we again observed long delays, indicating that they were also waiting for a sufficient number.

Beyond these personal interactions, the API provides information on the operation of the service as a whole. Most notably, it provides three separate pieces of information: (1) the current trading rate between any pair of cryptocurrencies, (2) a list of up to 50 of the most recent transactions that have taken place (across all users), and (3) full details of a specific ShapeShift transaction given the address $addr_s$ in the $curln$ blockchain (i.e., the address to which the user sent their coins).

For the trading rates, ShapeShift provides the following information for all cryptocurrency pairs ($curln, curOut$): the rate, the limit (i.e., the maximum that can be exchanged), the minimum that can be exchanged, and the miner fee (denominated in $curOut$). For the 50 most recent transactions, information is provided in the form: $(curln, curOut, amt, t, id)$, where the first three of these are as discussed in Section 5.2.1, t is a UNIX timestamp, and id is an internal identifier for this transaction. For the transaction information, when provided with a specific $addr_s$ ShapeShift provides the tuple $(status, address, withdraw, inCoin, inType, outCoin, outType, tx, txURL, error)$. The $status$ field is a flag that is either `complete`, to mean the transaction was successful; `error`, to mean an issue occurred with the transaction or the queried address was not a ShapeShift address; or `no_deposits`, to mean a user initiated a transaction but did not send any coins. The $error$ field appears when an error is returned and gives a reason for the error. The $address$ field is the same address $addr_s$ used by ShapeShift, and $withdraw$ is the address $addr_u$ (i.e., the user's recipient address in the $curOut$ blockchain). $inType$ and $outType$ are the respective $curln$ and $curOut$ currencies and $inCoin$ is the amt received. $outCoin$ is the amount sent in the $curOut$ blockchain. Finally, tx is the transaction hash in the $curOut$ blockchain and $txURL$ is a link to this transaction in an online explorer.

Using a simple Web scraper, we downloaded the transactions and rates every five seconds for close to thirteen months: from November 27 2017 until December 23 2018. This resulted in a set of 2,843,238 distinct transactions. Interestingly, we noticed that several earlier test transactions we did with the platform did not show up

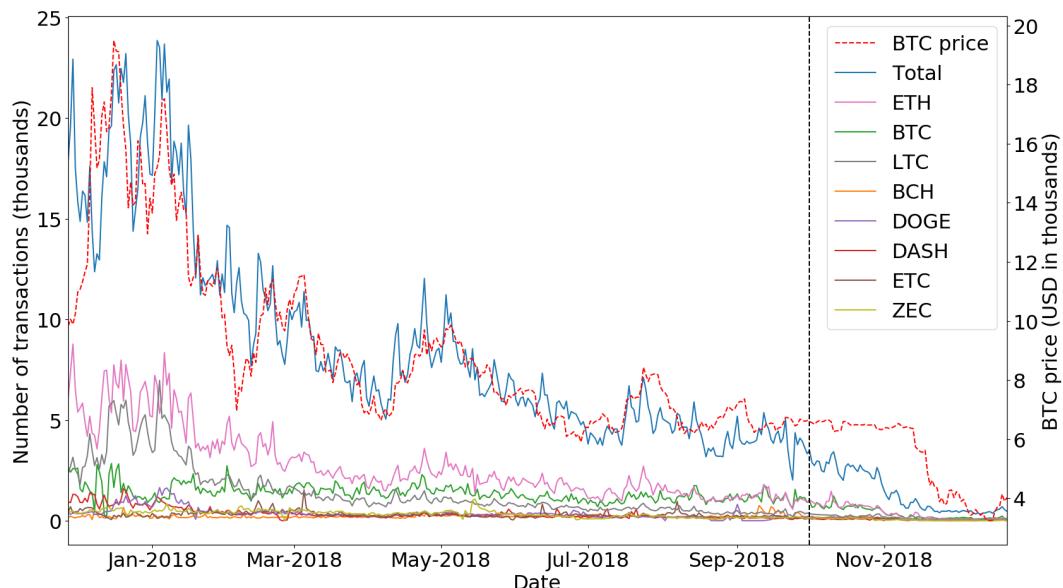


Figure 5.1: The total number of transactions per day reported via ShapeShift’s API, and the numbers broken down by cryptocurrency (where a transaction is attributed to a coin if it is used as either curIn or curOut). The dotted red line indicates the BTC-USD exchange rate, and the horizontal dotted black line indicates when KYC was introduced into ShapeShift.

in their list of recent transactions, which suggests that their published transactions may in fact underestimate their overall activity.

5.3.2.1 ShapeShift currencies

In terms of the different cryptocurrencies used in ShapeShift transactions, their popularity was distributed as seen in Figure 5.1. As this figure depicts, the overall activity of ShapeShift is (perhaps unsurprisingly) correlated with the price of Bitcoin in the same time period. At the same time, there is a decline in the number of transactions after KYC was introduced that is not clearly correlated with the price of Bitcoin (which is largely steady and declines only several months later).

ShapeShift supports dozens of cryptocurrencies, and in our data we observed the use of 65 different ones. The most commonly used coins are shown in Table 5.1. It is clear that Bitcoin and Ethereum are the most heavily used currencies, which is perhaps not surprising given the relative ease with which they can be exchanged with fiat currencies on more traditional exchanges, and their rank in terms of market capitalization.

Currency	Abbr.	Total	curln	curOut
Ethereum	ETH	1385509	892971	492538
Bitcoin	BTC	1286772	456703	830069
Litecoin	LTC	720047	459042	261005
Bitcoin Cash	BCH	284514	75774	208740
Dogecoin	DOGE	245255	119532	125723
Dash	DASH	187869	113272	74597
Ethereum Classic	ETC	179998	103177	76821
Zcash	ZEC	154142	111041	43101

Table 5.1: The eight most popular coins used on ShapeShift, in terms of the total units traded, and the respective units traded with that coin as curln and curOut.

5.3.3 Blockchain data

For the cryptocurrencies we were interested in exploring further, it was also necessary to download and parse the respective blockchains, in order to identify the on-chain transactional behavior of ShapeShift and Changelly. It was not feasible to do this for all 65 currencies used on ShapeShift (not to mention that given the low volume of transactions for many of them, it would likely not yield additional insights anyway), so we chose to focus instead on just the top 8, as seen in Table 5.1. Together, these account for 95.7% of all ShapeShift transactions if only one of curln and/or curOut is one of the eight, and 60.5% if both are.

For each of these currencies, we ran a full node in order to download the entire blockchain. For the ones supported by the BlockSci tool [77] (Bitcoin, Dash and Zcash), we used it to parse and analyze their blockchains. BlockSci does not, however, support the remaining five currencies. For these we thus parsed the blockchains using Python scripts, stored the data as Apache Spark parquet files, and analyzed them using custom scripts. In total, we ended up working with 654 GB of raw blockchain data and 434 GB of parsed blockchain data.

5.4 Identifying Blockchain Transactions

In order to gain deeper insights about the way these trading platforms are used, it is necessary to identify not just their internal transactions but also the transactions that appear on the blockchains of the traded currencies. This section presents heuristics for identifying these on-chain transactions, and the next section explores the additional insights these transactions can offer.

Recall from Section 5.2.1 that an interaction with ShapeShift results in the deposit of coins from the user to the service on the `curln` blockchain (which we refer to as “Phase 1”), and the withdrawal of coins from the service to the user on the `curOut` blockchain (“Phase 2”). To start with Phase 1, we thus seek to identify the deposit transaction on the input (`curln`) blockchain. Similarly to Portnoff et al. [118], we consider two main requirements for identifying the correct on-chain transaction: (1) that it occurred reasonably close in time to the point at which it was advertised via the API, and (2) that the value it carried was identical to the advertised amount.

For this first requirement, we look for candidate transactions as follows. Given a ShapeShift transaction with timestamp t , we first find the block b (at some height h) on the `curln` blockchain that was mined at the time closest to t . We then look at the transactions in all blocks with height in the range $[h - \delta_b, h + \delta_a]$, where δ_b and δ_a are parameters specific to `curln`. We looked at both earlier and later blocks based on the observation in our own interactions that the timestamp published by ShapeShift would sometimes be earlier and sometimes be later than the on-chain transaction.

For each of our eight currencies, we ran this heuristic for every ShapeShift transaction using `curln` as the currency in question, with every possible combination of δ_b and δ_a ranging from 0 to 30. This resulted in a set of candidate transactions with zero hits (meaning no matching transactions were found), a single hit, or multiple hits. To rule out false positives, we initially considered as successful only ShapeShift transactions with a single candidate on-chain transaction, although we describe below an augmented heuristic that is able to tolerate multiple hits. We then used the values of δ_b and δ_a that maximized the number of single-hit transactions for each currency. As seen in Table 5.2, the optimal choice of these parameters varies significantly across currencies, according to their different block rates; typically we needed to look further before or after for currencies in which blocks were produced more frequently.

In order to validate the results of our heuristic for Phase 1, we use the addi-

tional capability of the ShapeShift API described in Section 5.3.2. In particular, we queried the API on the recipient address of every transaction identified by our heuristic for Phase 1. If the response of the API was affirmative, we flagged the recipient address as belonging to ShapeShift and we identified the transaction in which it received coins as the `curln` transaction. This also provided a way to identify the corresponding Phase 2 transaction on the `curOut` blockchain, as it is just the `tx` field returned by the API. As we proceed only in the case that the API returns a valid result, we gain ground-truth data in both Phase 1 and Phase 2. In other words, this method serves to not only validate our results in Phase 1 but also provides a way to identify Phase 2 transactions.

The heuristic described above is able to handle only single hits; i.e., the case in which there is only a single candidate transaction. Luckily, it is easy to augment this heuristic by again using the API. For example, assume we examine a BTC-ETH ShapeShift transaction and we find three candidate transactions in the Bitcoin blockchain after applying the basic heuristic described above. To identify which of these transactions is the right one, we simply query the API on all three recipient addresses and check that the `status` field is affirmative (meaning ShapeShift recognizes this address) and that the `outType` field is ETH. In the vast majority of cases this uniquely identifies the correct transaction out of the candidate set, meaning we can use the API to both validate our results (i.e., we use it to eliminate potential false positives, as described above) and to augment the heuristic by being able to tolerate multiple candidate transactions. The augmented results for Phase 1 can be found in the last column of Table 5.2 and clearly demonstrate the benefit of this extra usage of the API. In the most dramatic example, we were able to go from identifying the on-chain transactions for ShapeShift transactions involving Bitcoin 65.75% of the time with the basic heuristic to identifying them 76.86% of the time with the augmented heuristic.

5.4.1 Accuracy of our heuristics

False negatives can occur for both of our heuristics when there are either too many or too few matching transactions in the searched block interval. These are more

Currency	Parameters		Basic %	Augmented %
	δ_b	δ_a		
BTC	0	1	65.76	76.86
BCH	9	4	76.96	80.23
DASH	5	5	84.77	88.65
DOGE	1	4	76.94	81.69
ETH	5	0	72.15	81.63
ETC	5	0	76.61	78.67
LTC	1	2	71.61	76.97
ZEC	1	3	86.94	90.54

Table 5.2: For the selected (optimal) parameters and for a given currency used as curln, the percentage of ShapeShift transactions for which we found matching on-chain transactions for both the basic (time- and value-based) and the augmented (API-based) Phase 1 heuristic. The augmented heuristic uses the API and thus also represents our success in identifying Phase 2 transactions.

common for the basic heuristic, as described above and seen in Table 5.2, because it is conservative in identifying an on-chain transaction only when there is one candidate. This rate could be improved by increasing the searched block radius, at the expense of adding more computation and potentially increasing the false positive rate.

False positives can occur for both of our heuristics if someone sends the same amount as the ShapeShift transaction at roughly the same time, but this transaction falls within our searched interval whereas the ShapeShift one doesn't. In theory, this should not be an issue for our augmented heuristic, since the API will make it clear that the candidate transaction is not in fact associated with ShapeShift. In a small number of cases (fewer than 1% of all ShapeShift transactions), however, the API returned details of a transaction with different characteristics than the one we were attempting to identify; e.g., it had a different pair of currencies or a different value being sent. This happened because ShapeShift allows users to re-use an existing deposit address, and the API returns only the latest transaction using a given address.

If we blindly took the results of the API, then this would lead to false positives in our augmented heuristic for both Phase 1 and Phase 2. We thus ensured that the transaction returned by the API had three things in common with the ShapeShift

transaction: (1) the pair of currencies, (2) the amount being sent, and (3) the timing, within the interval specified in Table 5.2. If there was any mismatch, we discarded the transaction. For example, given a ShapeShift transaction indicating an ETH-BTC shift carrying 1 ETH and occurring at time t , we looked for all addresses that received 1 ETH at time t or up to 5 blocks earlier. We then queried the API on these addresses and kept only those transactions which reported shifting 1 ETH to BTC. While our augmented heuristic still might produce false positives in the case that a user quickly makes two different transactions using the same currency pair, value, and deposit address, we view this as unlikely, especially given the relatively long wait times we observed ourselves when using the service (as mentioned in Section 5.3.2).

5.4.2 Alternative Phase 2 identification

Given that our heuristic for Phase 2 involved just querying the API for the corresponding Phase 1 transaction, it is natural to wonder what would be possible without this feature of the API, or indeed if there are any alternative strategies for identifying Phase 2 transactions. Indeed, it is possible to use a similar heuristic for identifying Phase 1 transactions, by first looking for transactions in blocks that were mined close to the advertised transaction time, and then looking for ones in which the amount was close to the expected amount. Here the amount must be estimated according to the advertised amt, rate, and fee. In theory, the amount sent should be $amt \cdot rate - fee$, although in practice the rate can fluctuate so it is important to look for transactions carrying a total value within a reasonable error rate of this amount.

When we implemented and applied this heuristic, we found that our accuracy in identifying Phase 2 transactions decreased significantly, due to the larger set of transactions that carried an amount within a wider range (as opposed to an exact amount, as in Phase 1) and the inability of this type of heuristic to handle multiple candidate transactions. More importantly, this approach provides no ground-truth information at all: by choosing conservative parameters it is possible to limit the number of false positives, but this is at the expense of the false negative rate (as, again, we observed in our own application of this heuristic) and in general it is not

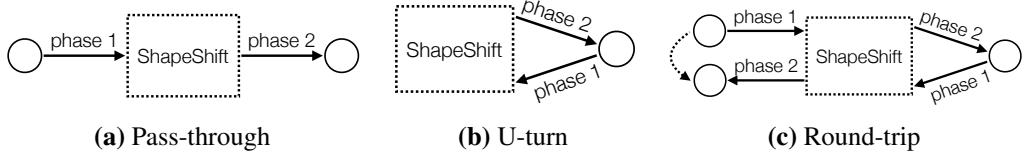


Figure 5.2: The different transactional patterns, according to how they interact with ShapeShift and which phases are required to identify them.

guaranteed that the final set of transactions really are associated with ShapeShift. As this is the exact guarantee we can get by using the API, we continue in the rest of the paper with the results we obtained there, but nevertheless mention this alternative approach in case this feature of the API is discontinued or otherwise made unavailable.

5.5 Tracking Cross-Currency Activity

In the previous section, we saw that it was possible in many cases to identify the on-chain transactions, in both the `curln` and `curOut` blockchains, associated with the transactions advertised by ShapeShift. In this section, we take this a step further and show how linking these transactions can be used to identify more complex patterns of behavior.

As shown in Figure 5.2, we consider these for three main types of transactions. In particular, we look at (1) *pass-through* transactions, which represent the full flow of money as it moves from one currency to the other via the deposit and withdrawal transactions; (2) *U-turns*, in which a user who has shifted into one currency immediately shifts back; and (3) *round-trip* transactions, which are essentially a combination of the first two and follow a user’s flow of money as it moves from one currency to another and then back to the original one. Our interest in these particular patterns of behavior is largely based on the role they play in tracking money as it moves across the ledgers of different cryptocurrencies. In particular, our goal is to test the validity of the implicit assumption made by criminal usage of the platform—such as we examine further in Section 5.7—that ShapeShift provides additional anonymity beyond simply transacting in a given currency.

In more detail, identifying pass-through transactions allows us to create a link between the input address(es) in the deposit on the `curln` blockchain and the output

address(es) in the withdrawal on the curOut blockchain.

Identifying U-turns allows us to see when a user has interacted with ShapeShift not because they are interested in holding units of the curOut cryptocurrency, but because they see other benefits in shifting coins back and forth. There are several possible motivations for this: for example, traders may quickly shift back and forth between two different cryptocurrencies in order to profit from differences in their price. We investigate this possibility in Section 5.7.3. Similarly, people performing money laundering or otherwise holding “dirty” money may engage in such behavior under the belief that once the coins are moved back into the curIn blockchain, they are “clean” after moving through ShapeShift regardless of what happened with the coins in the curOut blockchain.

Finally, identifying round-trip transactions allows us to create a link between the input address(es) in the deposit on the curIn blockchain with the output address(es) in the later withdrawal on the curIn blockchain. Again, there are many reasons why users might engage in such behavior, including the trading and money laundering examples given above. As another example, if a curIn user wanted to make an anonymous payment to another curIn user, they might attempt to do so via a round-trip transaction (using the address of the other user in the second pass-through transaction), under the same assumption that ShapeShift would sever the link between their two addresses.

5.5.1 Pass-through transactions

Given a ShapeShift transaction from curIn to curOut, the methods from Section 5.4 already provide a way to identify pass-through transactions, as depicted in Figure 5.2a. In particular, running the augmented heuristic for Phase 1 transactions identifies not only the deposit transaction in the curIn blockchain but also the Phase 2 transaction (i.e., the withdrawal transaction in the curOut blockchain), as this is exactly what is returned by the API. As discussed above, this has the effect on anonymity of tracing the flow of funds across this ShapeShift transaction and linking its two endpoints; i.e., the input address(es) in the curIn blockchain with the output address(es) in the curOut blockchain. The results, in terms of the percent-

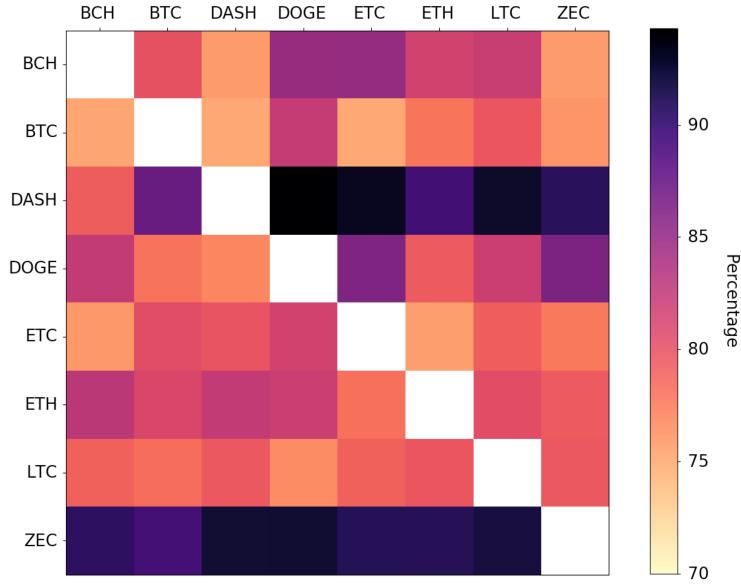


Figure 5.3: For each pair of currencies, the number of transactions we identified as being a pass-through from one to the other, as a percentage of the total number of transactions between those two currencies.

ages of all possible transactions between a pair (`curln`, `curOut`) for which we found the corresponding on-chain transactions, are in Figure 5.3.

The figure demonstrates that our success in identifying these types of transactions varied somewhat, and depended—not unsurprisingly—on our success in identifying transactions in the `curln` blockchain. This means that we were typically least successful with `curln` blockchains with higher transaction volumes, such as Bitcoin, because we frequently ended up with multiple hits (although here we were still able to identify more than 74% of transactions). In contrast, the dark stripes for Dash and Zcash demonstrate our high level of success in identifying pass-through transactions with those currencies as `curln`, due to our high level of success in their Phase 1 analysis in general (89% and 91% respectively). In total, across all eight currencies we were able to identify 1,383,666 pass-through transactions.

5.5.2 U-turns

As depicted in Figure 5.2b, we consider a U-turn to be a pattern in which a user has just sent money from `curln` to `curOut`, only to turn around and go immediately back to `curln`. This means linking two transactions: the Phase 2 transaction used to send money to `curOut` and the Phase 1 transaction used to send money back to `curln`. In terms of timing and amount, we require that the second transaction happens within

30 minutes of the first, and that it carries within 1% of the value that was generated by the first Phase 2 transaction. This value is returned by the ShapeShift API in the `outCoin` field.

While the close timing and amount already give some indication that these two transactions are linked, it is of course possible that this is a coincidence and they were in fact carried out by different users. In order to gain additional confidence that it was the same user, we have two options. In UTXO-based cryptocurrencies (see Section 2.1.3), we could see if the input is the same UTXO that was created in the Phase 2 transaction, and thus see if a user is spending the coin immediately. In cryptocurrencies based instead on accounts, such as Ethereum, we have no choice but to look just at the addresses. Here we thus define a U-turn as seeing if the address that was used as the output in the Phase 2 transaction is used as the input in the later Phase 1 transaction.

Once we identified such candidate pairs of transactions (tx_1, tx_2) , we then ran the augmented heuristic from Section 5.4 to identify the relevant output address in the `curOut` blockchain, according to tx_1 . We then ran the same heuristic to identify the relevant input address in the `curOut` blockchain, this time according to tx_2 .

In fact though, what we really identified in Phase 2 was not just an address but, as described above, a newly created UTXO. If the input used in tx_2 was this same UTXO, then we found a U-turn according to the first heuristic. If instead it corresponded just to the same address, then we found a U-turn according to the second heuristic. The results of both of these heuristics, in addition to the basic identification of U-turns according to the timing and amount, can be found in Table 5.3, and plots showing their cumulative number over time can be found in Figures 5.4 and 5.5. In total, we identified 107,267 U-turns according to our basic heuristic, 10,566 U-turns according to our address-based heuristic, and 1,120 U-turns according to our UTXO-based heuristic.

While the dominance of both Bitcoin and Ethereum should be expected given their overall trading dominance, we also observe that both Dash and Zcash have been used extensively as “mixer coins” in U-turns, and are in fact more popular for

Currency	# (basic)	# (addr)	# (utxo)
BTC	36666	565	314
BCH	2864	196	81
DASH	3234	2091	184
DOGE	546	75	75
ETH	53518	5248	-
ETC	1397	543	-
LTC	8270	1429	244
ZEC	772	419	222

Table 5.3: The number of U-turns identified for each cryptocurrency, according to our basic heuristic concerning timing and value, and both the address-based and UTXO-based heuristics concerning identical ownership. Since Ethereum and Ethereum Classic are account-based, the UTXO heuristic cannot be applied to them.

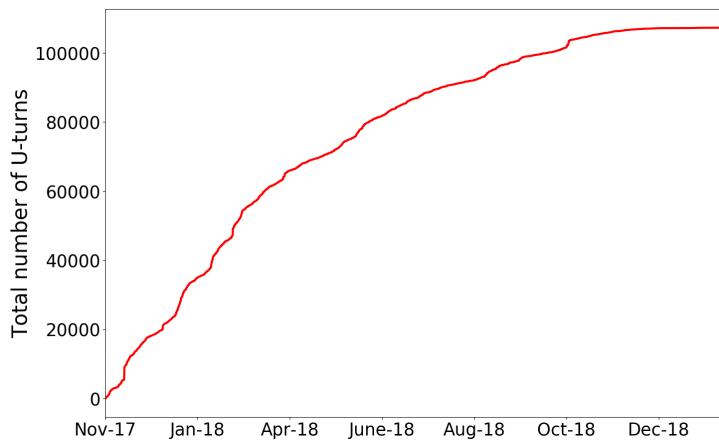


Figure 5.4: The total number of U-turns over time, as identified by our basic heuristic.

this purpose than they are overall. Despite this indication that users may prefer to use privacy coins as the mixing intermediary, Zcash has the highest percentage of identified UTXO-based U-turn transactions. Thus, these users not only do not gain extra anonymity by using it, but in fact are easily identifiable given that they did not change the address used in 419 out of 772 (54.24%) cases, or—even worse—immediately shifted back the exact same coin they received in 222 (28.75%) cases. In the case of Dash, the results suggest something a bit different. Once more, the usage of a privacy coin was not very successful since in 2091 out of the 3234 cases the address that received the fresh coins was the same as the one that shifted it back. It was the exact same coin in only 184 cases, however, which suggests that although the user is the same, there is a local Dash transaction between the two ShapeShift transactions. We defer a further discussion of this asymmetry to Section 5.7.4,

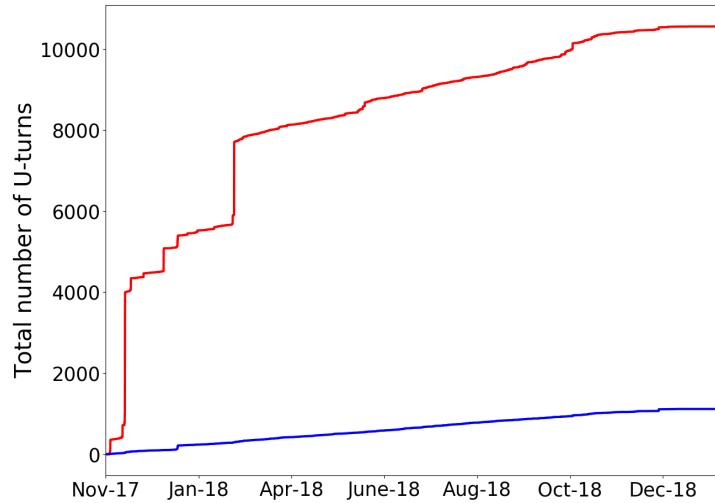


Figure 5.5: The total number of U-turns over time, as identified by our address-based (in red) and UTXO-based (in blue) heuristics.

where we also discuss more generally the use of anonymity features in both Zcash and Dash.

Looking at Figure 5.5, we can see a steep rise in the number of U-turns that used the same address in December 2017, which is not true of the ones that used the same UTXO or in the overall number of U-turns in Figure 5.4. Looking into this further, we observed that the number of U-turns was particularly elevated during this period for four specific pairs of currencies: DASH-ETH, DASH-LTC, ETH-DASH, and LTC-ETH. This thus affected primarily the address-based heuristic due to the fact that (1) Ethereum is account-based so the UTXO-based heuristic does not apply, and (2) Dash has a high percentage of U-turns using the same address, but a much smaller percentage using the same UTXO. The amount of money shifted in these U-turns varied significantly in terms of the units of the input currency, but all carried between 115K and 138K in USD. Although the ShapeShift transactions that were involved in these U-turns had hundreds of different addresses in the curIn blockchain, they used only a small number of addresses in the curOut blockchain: 4 addresses in Ethereum, 13 in Dash, and 9 in Litecoin. As we discuss further in Section 5.6.2, the re-use of addresses and the fact that the total amount of money (in USD) carried by the transactions was roughly the same indicates that perhaps a small group of people was responsible for creating this spike in the graph.

5.5.3 Round-trip transactions

As depicted in Figure 5.2c, a round-trip transaction requires performing two ShapeShift transactions: one out of the initial currency and one back into it. To identify round-trip transactions, we effectively combine the results of the pass-through and U-turn transactions; i.e., we tagged something as a round-trip transaction if the output of a pass-through transaction from X to Y was identified as being involved in a U-turn transaction, which was itself linked to a later pass-through transaction from Y to X (of roughly the same amount). As described at the beginning of the section, this has the powerful effect of creating a link between the sender and recipient within a single currency, despite the fact that money flowed into a different currency in between.

In more detail, we looked for consecutive ShapeShift transactions where for a given pair of cryptocurrencies X and Y: (1) the first transaction was of the form X-Y; (2) the second transaction was of the form Y-X; (3) the second transaction happened relatively soon after the first one; and (4) the value carried by the two transaction was approximately the same. For the third property, we required that the second transaction happened within 30 minutes of the first. For the fourth property, we required that if the first transaction carried x units of curln then the second transaction carried within 0.5% of the value in the (on-chain) Phase 2 transaction, according to the outCoin field provided by the API.

As with U-turns, we considered an additional restriction to capture the case in which the user in the curln blockchain stayed the same, meaning money clearly did not change hands. Unlike with U-turns, however, this restriction is less to provide accuracy for the basic heuristic and more to isolate the behavior of people engaged in day trading or money laundering (as opposed to those meaningfully sending money to other users). For this pattern, we identify the input addresses used in Phase 1 for the first transaction, which represent the user who initiated the round-trip transaction in the curln blockchain. We then identify the output addresses used in Phase 2 for the second transaction, which represent the user who was the final recipient of the funds. If the address was the same, then it is clear that money

Currency	# (regular)	# (same addr)
BTC	35019	437
BCH	1780	84
DASH	3253	2353
DOGE	378	0
ETH	45611	4085
ETC	1122	626
LTC	6912	2733
ZEC	472	172

Table 5.4: The number of regular round-trip transactions identified for each cryptocurrency, and the number that use the same initial and final address.

has not changed hands. Otherwise, the round-trip transaction acts as a heuristic for linking together the input and output addresses.

The results of running this heuristic (with and without the extra restriction) are in Table 5.4. In total, we identified 95,547 round-trip transactions according to our regular heuristic, and identified 10,490 transactions where the input and output addresses were the same. Across different currencies, however, there was a high level of variance in the results. While this could be a result of the different levels of accuracy in Phase 1 for different currencies, the more likely explanation is that users indeed engage in different patterns of behavior with different currencies. For Bitcoin, for example, there was a very small percentage (1.2%) of round-trip transactions that used the same address. This suggests that either users are aware of the general lack of anonymity in the basic Bitcoin protocol and use ShapeShift to make anonymous payments, or that if they do use round-trip transactions as a form of money laundering they are at least careful enough to change their addresses. More simply, it may just be the case that generating new addresses is more of a default in Bitcoin than it is in other currencies.

In other currencies, however, such as Dash, Ethereum Classic, Litecoin, and Zcash, there were relatively high percentages of round-trip transactions that used the same input and output address: 72%, 56%, 40%, and 36% respectively. In Ethereum Classic, this may be explained by the account-based nature of the currency, which means that it is common for one entity to use only one address, although the percentage for Ethereum is much lower (9%). In Dash and Zcash, as

we have already seen in Section 5.5.2 and explore further in Section 5.7.4, it may simply be the case that users assume they achieve anonymity just through the use of a privacy coin, so do not take extra measures to hide their identity.

5.6 Clustering Analysis

5.6.1 Shared ownership heuristic

As described in Sections 5.3.1 and 5.3.2, we engaged in transactions with both ShapeShift and Changelly, which provided us with some ground-truth evidence of addresses that were owned by them. We also collected three sets of tagging data (i.e., tags associated with addresses that describe their real-world owner): for Bitcoin we used the data available from WalletExplorer,⁶ which covers a wide variety of different Bitcoin-based services; for Zcash we used hand-collected data from Kappos et al. [79], which covers only exchanges; and for Ethereum we used the data available from Etherscan,⁷ which covers a variety of services and contracts.

In order to understand the behavior of these trading platforms and the interaction they had with other blockchain-based services such as exchanges, our first instinct was to combine these tags with the now-standard “multi-input” clustering heuristic for cryptocurrencies [95, 122], which states that in a transaction with multiple input addresses, all inputs belong to the same entity. When we applied this clustering heuristic to an earlier version of our dataset [155], however, the results were fairly uneven. For Dogecoin, for example, the three ShapeShift transactions we performed revealed only three addresses, which each had done a very small number of transactions. By clustering the addresses we sent coins to and received from the three Changelly transactions we performed, we identified 24,893 addresses, which in total had received over 67 trillion DOGE. These results suggest that the trading platforms operate a number of different clusters in each cryptocurrency, and perhaps even change their behavior depending on the currency, which in turns makes it clear that we did not capture a comprehensive view of the activity of either.

⁶<https://www.walletexplorer.com/>

⁷<https://etherscan.io/>

More worrying, in one of our Changelly transactions, we received coins from a Ethereum address that had been tagged as belonging to HitBTC, a prominent exchange. This suggests that Changelly may occasionally operate using exchange accounts, which would completely invalidate the results of the clustering heuristic, as their individually operated addresses would end up in the same cluster as all of the ones operated by HitBTC. We thus decided not to use this type of clustering, and to instead focus on a new clustering heuristic geared at identifying common social relationships.

5.6.2 Common relationship heuristic

As it was clear that the multi-input heuristic would not yield meaningful information about shared ownership, we chose to switch our focus away from the interactions ShapeShift had on the blockchain and look instead at the relationships between individual ShapeShift users. In particular, we defined the following heuristic:

Heuristic 6. If two or more addresses send coins to the same address in the `curOut` blockchain, or if two or more addresses receive coins from the same address in the `curln` blockchain, then these addresses have some common social relationship.

The definition of a common social relationship is (intentionally) vague, and the implications of this heuristic are indeed less clear-cut than those of heuristics around shared ownership. Nevertheless, we consider what it means for two different addresses, in potentially two different blockchains, to have sent coins to the same address; we refer to these addresses as belonging in the *input* cluster of the output address (and analogously refer to the *output* cluster for an address sending to multiple other addresses). In the case in which the addresses are most closely linked, it could represent the same user consolidating money held across different currencies into a single one. It could also represent different users interacting with a common service, such as an exchange. Finally, it could simply be two users who do not know each other directly but happen to be sending money to the same individual. What cannot be the case, however, is that the addresses are not related in any way.

To implement this heuristic, we parsed transactions into a graph where we defined a node as an address and a directed edge (u, v) as existing when one address u initiated a ShapeShift transaction sending coins to v , which we identified using the results of our pass-through analysis from Section 5.4. (This means that the inputs in our graph are restricted to those for which we ran Phase 1 to find the address, and thus that our input clusters contain only the top 8 currencies. In the other direction, however, we obtain the address directly from the API, which means output clusters can contain all currencies.) Edges are further weighted by the number of transactions sent from u to v . For each node, the cluster centered on that address was then defined as all nodes adjacent to it (i.e., pointing towards it).

Performing this clustering generated a graph with 2,895,445 nodes (distinct addresses) and 2,244,459 edges. Sorting the clusters by in-degree reveals the entities that received the highest number of ShapeShift transactions (from the top 8 currencies, per our caveat above). The largest cluster had 12,868 addresses—many of them belonging to Ethereum, Litecoin, and Dash—and was centered on a Bitcoin address belonging to CoinPayments.net, a multi-coin payment processing gateway. Of the ten largest clusters, three others (one associated with Ripple and two with Bitcoin addresses) are also connected with CoinPayments, which suggests that ShapeShift is a popular platform amongst its users.

Sorting the individual clusters by out-degree reveals instead the users who initiated the highest number of ShapeShift transactions. Here the largest cluster (consisting of 2314 addresses) was centered on a Litecoin address, and the second largest cluster was centered on an Ethereum address that belonged to Binance (a popular exchange). Of the ten largest clusters, two others were centered on Binance-tagged addresses, and three were centered on other exchanges (Freewallet, Gemini, and Bittrex). While it makes sense that exchanges typically dominate on-chain activity in many cryptocurrencies, it is somewhat surprising to also observe that dominance here, given that these exchanges already allow users to shift between many different cryptocurrencies. Aside from the potential for better rates or the perception of increased anonymity, it is thus unclear why a user wanting to shift from one currency

to another would do so using ShapeShift as opposed to using the same service with which they have already stored their coins.

Beyond these basic statistics, we apply this heuristic to several of the case studies we investigate in the next section. We also revisit here the large spike in the number of U-turns that we observed in Section 5.5.2. Our hypothesis then was that this spike was caused by a small number of parties, due to the similar USD value carried by the transactions and by the re-use of a small number of addresses across Dash, Ethereum, and Litecoin. Here we briefly investigate this further by examining the clusters centered on these addresses.

Of the 13 Dash addresses, all but one of them formed small input and output clusters that were comprised of addresses solely from Litecoin and Ethereum. Of the 9 Litecoin addresses, 6 had input clusters consisting solely of Dash and Ethereum addresses, with two of them consisting solely of Dash addresses. Finally, of the 4 Ethereum addresses, all of them had input clusters consisting solely of Dash and Litecoin addresses. One of them, however, had a diverse set of addresses in its output cluster, belonging to Bitcoin, Bitcoin Cash, and a number of Ethereum-based tokens. These results thus still suggest a small number of parties, due to the tight connection between the three currencies in the clusters, although of course further investigation would be needed to get a more complete picture.

5.7 Patterns of ShapeShift Usage

In this section, we examine potential applications of the analysis developed in previous sections, in terms of identifying specific usages of ShapeShift. As before, our focus is on anonymity, and the potential that such platforms may offer for money laundering or other illicit purposes, as well as for trading. To this end, we begin by looking at two case studies associated with explicitly criminal activity and examine the interactions these criminals had with the ShapeShift platform. We then switch in Section 5.7.3 to look at non-criminal activity, by attempting to identify trading bots that use ShapeShift and the patterns they may create. Finally, in Section 5.7.4 we look at the role that privacy coins (Monero, Zcash, and Dash) play, in order to

identify the extent to which the usage of these coins in ShapeShift is motivated by a desire for anonymity.

5.7.1 Starscape Capital

In January 2018, an investment firm called Starscape Capital raised over 2,000 ETH (worth 2.2M USD at the time) during their Initial Coin Offering, after promising users a 50% return in exchange for investing in their cryptocurrency arbitrage fund. Shortly afterwards, all of their social media accounts disappeared, and it was reported that an amount of ETH worth 517,000 USD was sent from their wallet to ShapeShift, where it was shifted into Monero [132].

We confirmed this for ourselves by observing that the address known to be owned by Starscape Capital participated in 192 Ethereum transactions across a three-day span (January 19-21), during which it received and sent 2,038 ETH; in total it sent money in 133 transactions. We found that 109 of these transactions sent money to ShapeShift, and of these 103 were shifts to Monero conducted on January 21 (the remaining 6 were shifts to Ethereum). The total amount shifted into Monero was 465.61 ETH (1388.39 XMR), and all of the money was shifted into only three different Monero addresses, of which one received 70% of the resulting XMR. Using the clusters defined in Section 5.6.2, we did not find evidence of any other addresses (in any other currencies) interacting with either the ETH or XMR addresses associated with Starscape Capital.

5.7.2 Ethereum-based scams

EtherScamDB⁸ is a website that, based on user reports that are manually investigated by its operators, collects and lists Ethereum addresses that have been involved in scams. As of January 30 2019, they had a total of 6374 scams listed, with 1973 associated addresses. We found that 194 of these addresses (9% of those listed) had been involved in 853 transactions to ShapeShift, of which 688 had a status field of complete. Across these successful transactions, 1797 ETH was shifted to other currencies: 74% to Bitcoin, 19% to Monero, 3% to Bitcoin Cash, and 1% to Zcash.

⁸<https://etherscamdb.info/>

The scams which successfully shifted the highest volumes belonged to so-called trust-trading and MyEtherWallet scams. Trust-trading is a scam based on the premise that users who send coins prove the legitimacy of their addresses, after which the traders “trust” their address and send back higher amounts (whereas in fact most users send money and simply receive nothing in return). This type of scam shifted over 918 ETH, the majority of which was converted to Bitcoin (691 ETH, or 75%). A MyEtherWallet scam is a phishing/typosquatting scam where scammers operate a service with a similar name to the popular online wallet MyEtherWallet,⁹ in order to trick users into giving them their account details. These scammers shifted the majority of the stolen ETH to Bitcoin (207 ETH) and to Monero (151 ETH).

Given that the majority of the overall stolen coins was shifted to Bitcoin, we next investigated whether or not these stolen coins could be tracked further using our analysis. In particular, we looked to see if they performed a U-turn or a round-trip transaction, as discussed in Section 5.5. We identified one address, associated with a trust-trading scam, that participated in 34 distinct round-trip transactions, all coming back to a different address from the original one. All these transactions used Bitcoin as curOut and used the same address in Bitcoin to both receive and send coins; i.e., we identified the U-turns in Bitcoin according to our address-based heuristic. In total, more than 70 ETH were circulated across these round-trip transactions.

5.7.3 Trading bots

ShapeShift, like any other cryptocurrency exchange, can be used by traders who wish to take advantage of the volatility in cryptocurrency prices. The potential advantages of doing this via ShapeShift, as compared with other platforms that focus more on the exchange between cryptocurrencies and fiat currencies, are that (1) ShapeShift transactions can be easily automated via their API, and (2) a single ShapeShift transaction acts to both purchase desired coins and dump unwanted ones. Such trading usually requires large volumes of transactions and high precision on their the timing, due to the constant fluctuation in cryptocurrency prices.

⁹<https://www.myetherwallet.com/>

We thus looked for activity that involved large numbers of similar transactions in a small time period, on the theory that it would be associated primarily with trading bots.

We started by searching for sets of consecutive ShapeShift transactions that carried approximately the same value in `curln` (with an error rate of 1%) and involved the same currencies. When we did this, however, we found thousands of such sets. We thus added the extra conditions that there must be at least 15 transactions in the set that took place in a span of five minutes; i.e., that within a five-minute block of ShapeShift transactions there were at least 15 involving the same currencies and carrying the same approximate USD value. This resulted in 107 such sets.

After obtaining our 107 trading clusters, we removed transactions that we believed were false positives in that they happened to have a similar value but were clearly the odd one out. For example, in a cluster of 20 transactions with 19 ETH-BTC transactions and one LTC-ZEC transaction, we removed the latter. We were thus left with clusters of either a particular pair (e.g., ETH-BTC) or two pairs where the `curOut` or the `curln` was the same (e.g., ETH-BTC and ZEC-BTC), which suggests either the purchase of a rising coin or the dump of a declining one. We sought to further validate these clusters by using our heuristic from Section 5.6.2 to see if the clusters shared common addresses. While we typically did not find this in UTXO-based currencies (as most entities operate using many addresses), in account-based currencies we found that in almost every case there was one particular address that was involved in the trading cluster.

We summarize our results in Figure 5.6, in terms of the most common pairs of currencies and the total money exchanged by trading clusters using those currencies. It is clear that the most common interactions are performed between the most popular currencies overall, with the exception of Monero (XMR) and SALT. In particular, we found six clusters consisting of 17-20 transactions that exchanged BTC for XMR, and 13 clusters that exchanged BTC for SALT, an Ethereum-based token. The sizes of each trading cluster varied between 16 and 33 transactions and in total comprise 258 transactions, each of which shifted exactly 0.1 BTC. In total

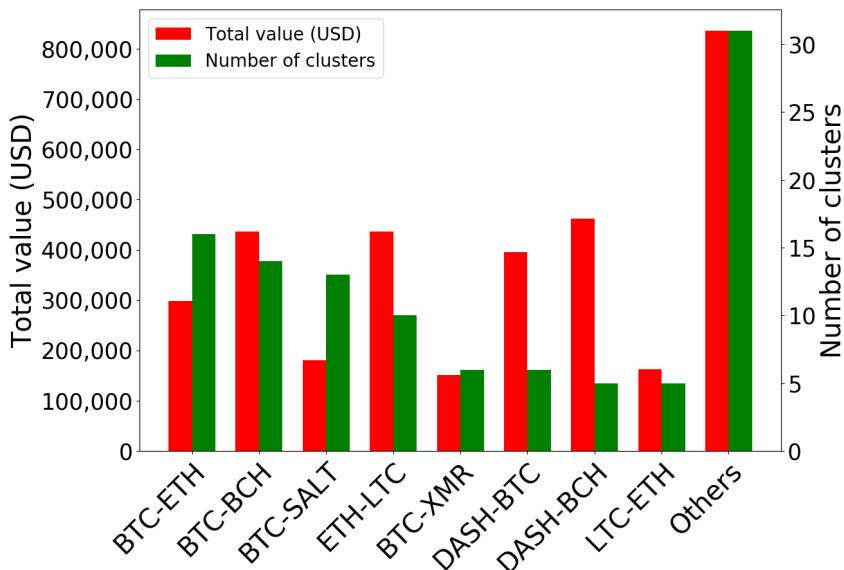


Figure 5.6: Our 107 clusters of likely trading bots, categorized by the pair of currencies they trade between and the total amount transacted by those clusters (in USD).

they originated from 514 different Bitcoin addresses, which may make it appear as though different people carried out these transactions. After applying our pass-through heuristic, however, we found that across all the transactions there were only two distinct SALT addresses used to receive the output. It is thus instead likely that this represents trading activity involving one or two entities.

5.7.4 Usage of anonymity tools

Given the potential usage of ShapeShift for money laundering or other criminal activities, we sought to understand the extent to which its users seemed motivated to hide the source of their funds. While using ShapeShift is already one attempt at doing this, we focus here on the combination of using ShapeShift and so-called “privacy coins” (Dash, Monero, and Zcash) that are designed to offer improved anonymity guarantees.

In terms of the effect of the introduction of KYC into ShapeShift, the number of transactions using Zcash as curIn averaged 164 per day the month before, and averaged 116 per day the month after. We also saw a small decline with Zcash as curOut: 69 per day before and 43 per day after. Monero and Dash, however, saw much higher declines, and in fact saw the largest declines across all eight cryptocurrencies. The daily average the month before was 136 using Monero as curIn,

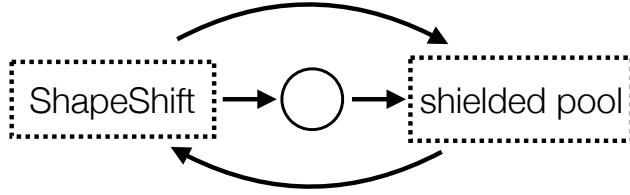


Figure 5.7: The three types of interactions we investigated between ShapeShift and the shielded pool in Zcash.

whereas it was 47 after. Similarly, the daily average using it as `curOut` was 316 before and 62 after. For Dash, the daily average as `curln` was 128 before and 81 after, and the daily average as `curOut` was 103 before and 42 after.

In terms of the blockchain data we had (according to the most popular currencies), our analysis in what follows is restricted to Dash and Zcash, although we leave an exploration of Monero as interesting future work.

5.7.4.1 Zcash

The main anonymity feature in Zcash is known as the *shielded pool*. Briefly, transparent Zcash transactions behave just like Bitcoin transactions in that they reveal in the clear the sender and recipient (according to so-called *t-addresses*), as well as the value being sent. This information is hidden to various degrees, however, when interacting with the pool. In particular, when putting money into the pool the recipient is specified using a so-called *z-address*, which hides the recipient but still reveals the sender, and taking money out of the pool hides the sender (through the use of zero-knowledge proofs [15]) but reveals the recipient. Finally, Zcash is designed to provide privacy mainly in the case in which users transact *within* the shielded pool, which hides the sender, recipient, and the value being sent.

We considered three possible interactions between ShapeShift and the shielded pool, as depicted in Figure 5.7: (1) a user shifts coins directly from ShapeShift into the shielded pool, (2) a user shifts to a t-address but then uses that t-address to put money into the pool, and (3) a user sends money directly from the pool to ShapeShift.

For the first type of interaction, we found 29,003 transactions that used ZEC as `curOut`. Of these, 758 had a z-address as the output address, meaning coins were sent directly to the shielded pool. The total value put into the pool in these

transactions was 6,707.86 ZEC, which is 4.3% of all the ZEC received in pass-through transactions. When attempting to use z-addresses in our own interactions with ShapeShift, however, we consistently encountered errors or were told to contact customer service. It is thus not clear if usage of this feature is supported at the time of writing.

For the second type of interaction, there were 1309 where the next transaction (i.e., the transaction in which this UTXO spent its contents) involved putting money into the pool. The total value put into the pool in these transactions was 12,534 ZEC, which is 8.2% of all the ZEC received in pass-through transactions.

For the third type of interaction, we found 111,041 pass-through transactions that used ZEC as curln. Of these, 3808 came directly from the pool, with a total value of 22,490 ZEC (14% of all the ZEC sent in pass-through transactions).

Thus, while the usage of the anonymity features in Zcash was not necessarily a large fraction of the overall usage of Zcash in ShapeShift, there is clear potential to move large amounts of Zcash (representing over 10 million USD at the time it was transacted) by combining ShapeShift with the shielded pool.

5.7.4.2 Dash

Our parameters for identifying a CoinJoin were thus that (1) the transaction must have at least three inputs, (2) the outputs must consist solely of values from the list of possible denominations (modulo the fees), and (3) and all output values must be the same. In fact, given how Dash operates there is always one output with a non-standard value, so it was further necessary to relax the second and third requirements to allow there to be at most one address that does not carry the specified value.

We first looked to see how often the DASH sent to ShapeShift had originated from a CoinJoin, which meant identifying if the inputs of a Phase 1 transaction were outputs from a CoinJoin. Out of 100,410 candidate transactions, we found 2,068 that came from a CoinJoin, carrying a total of 11,929 DASH in value (6.5% of the total value across transactions with Dash as curln). Next, we looked at whether or not users performed a CoinJoin after receiving coins from ShapeShift, which meant identifying if the outputs of a Phase 2 transaction had been spent in a CoinJoin. Out

of 50,545 candidate transactions, we found only 33 CoinJoin transactions, carrying a total of 187 DASH in value (0.1% of the total value across transactions using Dash as curOut).

If we revisit our results concerning the use of U-turns in Dash from Section 5.5.2, we recall that there was a large asymmetry in terms of the results of our two heuristics: only 5.6% of the U-turns used the same UTXO, but 64.6% of U-turns used the same address. This suggests that some additional on-chain transaction took place between the two ShapeShift transactions, and indeed upon further inspection we identified many cases where this transaction was a CoinJoin. There thus appears to have been a genuine attempt to take advantage of the privacy that Dash offers, but this was completely ineffective due to the use of the same address that both sent and received the mixed coins.

5.8 Conclusions

In this study, we presented a characterization of the usage of the ShapeShift trading platform over a thirteen-month period, focusing on the ability to link together the ledgers of multiple different cryptocurrencies. To accomplish this task, we looked at these trading platforms from several different perspectives, ranging from the correlations between the transactions they produce in the cryptocurrency ledgers to the relationships they reveal between seemingly distinct users. The techniques we develop demonstrate that it is possible to capture complex transactional behaviors and trace their activity even as it moves across ledgers, which has implications for any criminals attempting to use these platforms to obscure their flow of money.

Chapter 6

Forsage: An Anatomy of a Cryptocurrency Pyramid Scheme

6.1 Overview

Cryptocurrencies and smart contracts are new and powerful technologies that promise a range of benefits, including faster monetary transactions, innovative financial instruments, and global financial inclusion for the world’s unbanked. Conversely, though, these same technologies have fuelled new forms of fraud and theft [141, 162] and new ways of perpetrating existing types of crime [76, 116].

Pyramid schemes, for example, are a prevalent type of scam in which top-tier participants in a hierarchical network recruit and profit at the expense of an expanding base of new participants. They have existed for more than a century, but have recently emerged in a new form: as smart contracts on blockchains such as Ethereum.

Smart contracts are in some ways an ideal medium for pyramid schemes and other scams. Because they run in decentralized systems, they cannot easily be dismantled by law enforcement agencies. They can instantaneously ingest payments from victims across the globe. They provide privacy protection for their creators in the form of pseudonymous addresses. Finally, as so-called “trustless” applications—with world-readable (byte)code—they present a veneer of trustworthiness to unsuspecting users.

The flip side of such transparency is that smart contracts offer researchers a degree of visibility into the mechanics of online (and offline) scams that is without historical precedent. Not only is the (byte)code specifying the scam’s mechanics visible on chain, but so is every transaction performed by every participant.

In this chapter, we take advantage of this newfound visibility to conduct an in-depth measurement study of the largest smart contract-based pyramid scheme to date, called *Forsage Smartway* or *Forsage* for short.

Forsage came into existence in late January 2020. It was at one point the second most active contract in Ethereum by daily transaction count, and remains in the top twenty at the time of writing. As we show throughout this paper, it is a classic pyramid scheme, defined by the SEC as “a type of fraud in which participants profit almost exclusively through recruiting other people to participate in the program” [73]. The *Forsage* contract requires players to send currency (Ether) in order to participate. Funds sent by newly recruited users immediately pass through the contract to existing players, with those at the top of the (smart contract-defined) pyramid obtaining the largest returns.

Understanding the success of *Forsage* requires study of not just the contract itself, but also its community of hundreds of thousands of users, many of whom have actively discussed and marketed the scam. Consequently, to paint a detailed picture of how *Forsage* lures and defrauds users, our study combines measurement and analysis of a range of complementary forms of data, including source code, on-chain transaction data, and social media interactions.

Our results come from three basic, mutually illuminating forms of study: smart contract deconstruction (Section 6.4), blockchain analytics (Section 6.5), and analysis of video and social media interactions (Section 6.6).

We believe that our study’s findings are not just relevant to *Forsage*, but provide durable insights into the conception, mechanics, and evolution of smart-contract scams and financial scams more generally. They also point to effective strategies that government authorities and the cryptocurrency community can use to combat pyramid schemes and other scams, as we discuss in Section 6.7.

We emphasize that our results, which reveal a combination of classic and smart contract-specific scam characteristics, offer insights not just into Forsage, but into both blockchain and non-blockchain scams more generally.

6.2 Background

6.2.1 Smart contracts

Forsage is realized as a *smart contract*. Smart contracts are applications that execute on *blockchains*, decentralized systems that indelibly and immutably record transactions in an authoritative sequence and are best known as the platforms that realize cryptocurrencies such as Bitcoin.

The most popular public (permissionless) blockchain for smart contracts today is *Ethereum* [28], whose native currency is known as *Ether* (ETH). Ethereum smart contracts are launched in the form of bytecode that runs in a Turing-complete environment known as the Ethereum Virtual Machine (EVM). Smart contract creators often also publish corresponding source code, typically written in the Solidity programming language, but such publication is optional. *Transactions* sent to smart contracts by users are processed by contract code and are publicly visible on chain.

Transactions may send money to a contract from user accounts or other contracts and must specify payment of execution fees to miners in the form of *gas*, a parallel currency converted into ETH upon transaction execution. This conversion is calculated by multiplying the amount of work performed by a transaction (its “gas consumed”) by the price of gas in ETH set by user when submitting the transaction [151].

Correctness of contract execution is enforced by the consensus mechanism underlying the Ethereum blockchain, so a miner’s execution of contract code in the EVM must be agreed upon by all network participants to be included in a confirmed block.

Other permissionless blockchains with smart contract functionality are growing in popularity, e.g., Tron [53], to which Forsage has also been ported. Ethereum, however, remains the dominant smart contract platform.

6.2.2 Scams

Scams, i.e., fraudulent schemes involving financial deception, have been documented for centuries. Many scams involving large populations of victims assume the form of *pyramid schemes*. The U.S. Securities and Exchange Commission (SEC) defines a pyramid scheme as “a type of fraud in which participants profit almost exclusively through recruiting other people to participate in the program” [73]. Pyramid schemes, which are illegal in most jurisdictions, come in a number of variants. One variant is a *Ponzi scheme*, which specifically involves investment in financial instruments. *Multi-level marketing* (MLM) schemes, which involve the sale of a product or service, are related to pyramid schemes. They are legal in the U.S., but outlawed in some jurisdictions (e.g., China) [99].

6.2.3 Blockchain scams

A multitude of scams have arisen within the blockchain ecosystem. Some scams have solicited investments from victims in new blockchain technologies. Examples include Onecoin, a Ponzi scheme that involved a fake (centralized) blockchain in which victims invested \$19+ billion [89], Bitconnect, a token that promised returns of 1% per day and saw investment of \$3.5 billion from victims, as well as other, related \$1+ billion schemes such as Plustoken and WoToken.pro [14, 112].

Other scams instead use blockchain technology to realize variants of scams, such as pyramid schemes, that were seen well before the advent of blockchains. Prominent examples are Million.Money¹ and Doubleway.io², which are both currently active, as well as the defunct scheme Bullrun.live.³ All three have similarities with Forsage: they use similar promotional materials, have a similar structure for the user dashboard, and use similar language and terminology (e.g., a referrer to the program is called an “upline”). We explore Forsage user interactions with multiple scam contracts in 6.5.2.

¹<https://million.money>

²<https://doubleway.io/>

³<http://bullrun.live>

6.3 Forsage Overview

The creators and promoters of Forsage advertise it as a *matrix* MLM scheme, despite the lack of a service or product. It operates primarily on Ethereum, where its initial Matrix contract has been active since January 31st, 2020. Since then, Forsage creators have also launched a Forsage contract on Tron (TRX) and an additional, followup smart contracts called Forsage xGold on both Tron and Ethereum. At the time of this writing, the Forsage authors have since released contracts ([0x2C...](#), [0x98...](#)) on the Binance Smart Chain (BSC).

The Forsage website: Users interact with Forsage using the [forsage.io](#) website, which shows how much they have paid into and earned from the contract. The website encourages the use of user-friendly cryptocurrency tools. It shows users how to purchase cryptocurrency using Trust Wallet, a user-friendly tool to exchange fiat for cryptocurrency, and how to use MetaMask, a browser extension that allows users to easily transact with cryptocurrency. The combination of these tools makes Forsage accessible to novice users who may not previously have used cryptocurrencies or smart contracts.

Forsage use and structure: A new Forsage user must pay a minimum of 0.05 ETH, which opens up the *slot* at the first *level* in the two matrix systems, called X3 and X4. Each matrix consists of 12 slots. To unlock the ability to use the next slot (at level $i + 1$), a user must pay twice as much ETH as for their currently highest slot (at level i). In both X3 and X4, the first slot costs 0.025 ETH, while the twelfth and final slot costs 51.2 ETH. This means that the total cost to open all slots in either matrix is 102.375 ETH.

Each Forsage user has a *referral code*, created at the time they register. The referral code links a recruited user's account to the account that recruited them, called their *upline*. These referral codes thus organize Forsage users into pyramids, with the oldest accounts at the top. Payments flow upwards within a pyramid as additional users join it. The pyramids of users linked by chains of referral code are referred to as Forsage *teams*. It is possible to join Forsage without entering a referral code; users who do so are assigned the referral code of the contract *owner*.

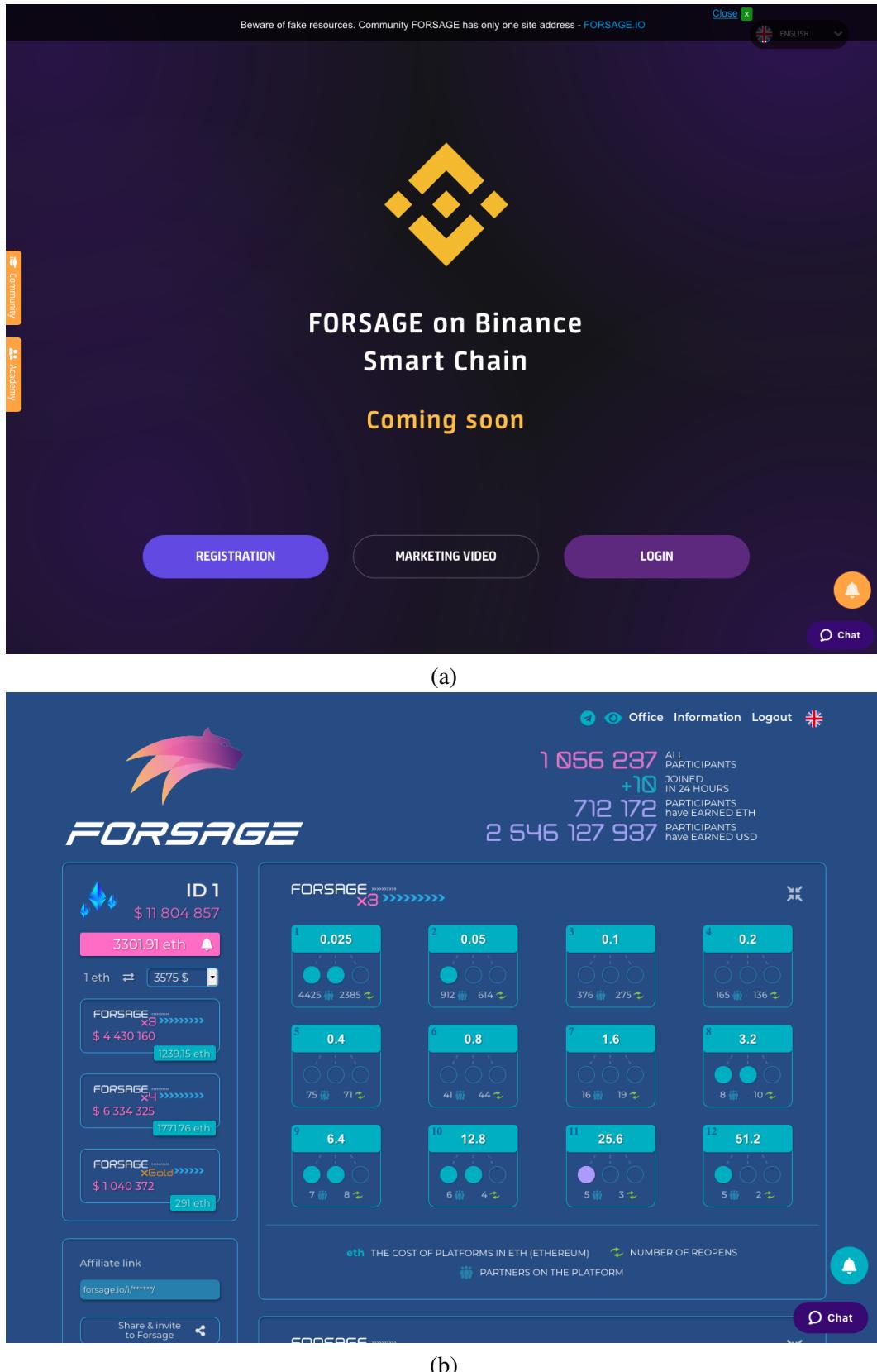


Figure 6.1: Screenshots of the forsage.io website. (a) Homepage of forsage.io as of May 6th, 2021. The website is marketing the arrival of the new scheme that will be used on the Binance Smart Chain. (b) Landing page of the most profitable user showing the progress page of the X3 matrix and other macro statistics.

(the creator).

6.4 contains an explanation of the logic for payment flow of user funds sent through the Forsage contract. Briefly, users earn money in the X3 and X4 matrices as follows.

X3: In X3, users earn income by recruiting others into the system. A user must recruit three additional users to recoup their initial investment within each slot. Any recruits beyond the first three per slot will generate income for the recruiting user and those further up in their pyramid. Each subsequent slot costs more to open, but its resulting payout if filled with recruits will be higher because the expected payout for each three recruits is equal to the initial cost to open the slot for the recruiter. After a user fills a slot (i.e. recruits 3 users into that slot), Forsage *blocks* the filled slot, causing the user to forfeit future earnings from it until it is unblocked. Unblocking means paying to open the slot at the next level up in the system, at which point this lower-level slot cannot become blocked again.

X4: In X4, users can earn both by recruiting other users and by being on an active team whom are opening new slots to ensure minimal blocking. When a user recruits the six additional users necessary to recoup their initial investment in an X4 slot (twice as many as are required in X3), that slot becomes blocked and the user will have received the same amount of money paid to open the slot, with others in their team getting paid as well. X4 also has an element of competition: If a newer user on a team is more active than the user whose referral code they used to join Forsage, that user can switch spots on the team, giving the more active, newer user the profits that would otherwise flow to the older, referring account [109].

6.4 Forsage Contract Deconstruction

Forsage promotional materials imply that the system is trustworthy because its code is open-source, e.g., the promotional materials claim that the contract “guarantees

the purity of conditions.” We took advantage of the availability of the source code to conduct an in-depth analysis of the smart contract’s logic and data structures.

Methodology and data collection: The code for the Matrix smart contract is published on Etherscan.⁴ We first attempted manual source code review, but found the logic too confusing to follow without visualization. We then built a simulator in Python that deployed the contract to a local private test network of Go-Ethereum (Geth) nodes,⁵ and used Web3.py⁶ to send sample transactions. We also wrote a visualizer for the contract’s state machine using GraphViz [44]. The output of that visualizer assisted in creating Figure 6.2, which depicts the data stored in the contract. Although the open source code is pointed to as a source of legitimacy by Forsage promotional materials, our analysis of the contract took weeks of focused effort by a professional research engineer. Our source code for the visualizer and simulator tools will be released as open source software in the near future.

When the Forsage team launched their Tron implementation of the Matrix smart contract, they also released its source code. We found this Tron code to be nearly identical to the Ethereum original, so we did not specifically analyze it. The latest iteration of Forsage launched on both Ethereum and Tron (as of May 2021), the xGold contract, has no publicly available source code.

The Ethereum and Tron blockchains include the data for all transactions performed by Forsage users. We mined this publicly available data to perform further analysis. To obtain Ethereum data we ran the Go-Ethereum (Geth)⁷ and Turbo-Geth⁸ full-node and archive-node software packages, and downloaded the entire blockchain up to January 14, 2021.

We then used the Ethereum-ETL⁹ package to retrieve this data from Geth and store the 345 million transactions included in the Ethereum blockchain between the launch of Forsage (January 31, 2020) and January 14, 2021. We wrote cus-

⁴etherscan.io/address/0x5acc84a3e955Bdd76467d3348077d003f00fFB97

⁵<https://github.com/ethereum/go-ethereum>

⁶<https://github.com/ethereum/web3.py>

⁷<https://github.com/ethereum/go-ethereum>

⁸<https://github.com/ledgerwatch/turbo-geth>

⁹<https://github.com/blockchain-etl/ethereum-etl>

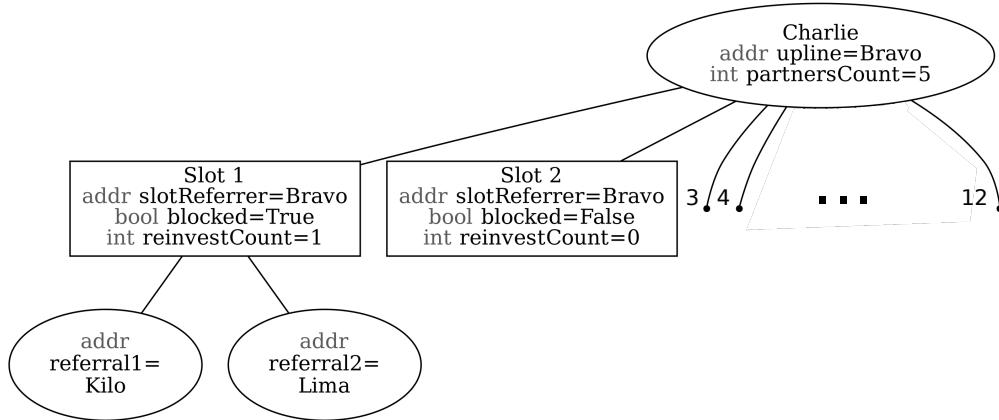


Figure 6.2: A visualization of the state the contract keeps for each user in the X3 matrix, focusing on a user Charlie. The addr variables point to Ethereum addresses, here given NATO-phonetic names. Matrix slots that have not yet been opened are depicted with a numbered dot, instead of a box.

tom Python scripts to analyze this data and found 222,516,680 transactions that involved function calls on smart contracts, of which 3,266,722 were to the Forsage smart contract. To profile user transactions outside Forsage, we used the Chainalysis Reactor tool.¹⁰ Chainalysis Reactor is a web-based investigation platform that connects cryptocurrency transactions to real-world entities, using tags that are either internal to Chainalysis or gathered from public websites and documents.

To collect Tron transaction data we scraped the TronScan API¹¹ and parsed the results directly into CSV form.

Forsage data structures: As discussed in Section 6.3, Forsage consists of two matrix systems, X3 and X4, each consisting of 12 slots. These two matrices differ in the number of users that act to fill each matrix level (three for X3, six for X4) and the logic for how nodes propagate through them over time.

The data for each user is stored in a hashtable (Solidity mapping) on the Ethereum blockchain, with the key being the user's address and the value being a Solidity struct with the data for that user's state tree and arrays of pointers to its children. Figure 6.2 visualizes this mapping for a user's X3 tree, with some minor metadata variables omitted. Each user also has an X4 tree, whose structure is largely similar. As seen in this figure, each user has an *upline*, which is the user that

¹⁰<https://www.chainalysis.com/chainalysis-reactor/>

¹¹<https://tronscan.org/>

referred them to the contract. This is distinct from `slotReferrer`, a variable used per slot as part of the payment logic. The `slotReferrer` variable is initialized to the upline, but changes over time as users refer each other. The `reinvestCount` variable keeps track of the number of times a slot has been filled. In our example, Charlie has filled his first matrix slot once already (and then unblocked it by buying a slot at level 2), meaning he has referred $3 \times \text{reinvestCount} + 2 = 5 = \text{partnersCount}$ users.

External API: The contract exposes 15 functions to read its state, and two state-changing functions, `registrationExt` and `buyNewLevel`. The first registers new users and thus adds them to the contract state. The second changes contract state for an existing user to allow them to continue to gain money from new referrals.

The placement of new users in the contract state depends on the X3 and X4 slots for the user that referred them (their upline). The logic of the contract *scrambles* positions in the upline’s matrices and in the matrices of the upline’s parent when an upline’s slot becomes full, i.e. every time the upline refers a multiple of three users to a given X3 slot (`partnersCount mod 3 = 0`), or a multiple of six users to a given X4 slot. The logic of scrambling leaf nodes in the pyramid depends on the state of the slot `referrer` variable for the affected matrix slot, as well as the `blocked` variable for that slot, and in the X4 system an additional `closedPart` variable for each slot. Scrambling the positions of the existing users in the system helps to make payments through Forsage (falsely!) appear more random. It benefits older users in the pyramid, as users are usually scrambled “up” the pyramid to become children of older users rather than newer ones.

Transaction fees: The fact that Forsage has so much persistent on-chain storage means that its users pay higher gas fees than the average for Ethereum contracts, due to the heavy usage of the (expensive) `SLOAD` and `SSTORE` opcodes. These fees are higher even when comparing Forsage transactions only to other contract function calls in Ethereum (so in particular ignoring simple sends of ETH). In our collected dataset of Ethereum network transactions, we found that the mean transaction fee for all Ethereum transactions that interacted with a contract was 0.00632 ETH with a standard deviation of 0.0618 ETH and a median of 0.00257 ETH. Forsage transac-

Opcode	Avg num per tx (all)	Median (all)	Avg num per tx (Forsage)	Median (Forsage)
SSTORE	4.54 ± 8.10	2	10.76 ± 9.57	6
SLOAD	17.84 ± 51.6	7	36.86 ± 26.21	29

Table 6.1: Average number of instruction operations per transaction, with standard deviation, for both all transactions and only those that interact with Forsage. Due to the intensive computation required to process this data, this table covers only the thousand blocks between block heights 10,600,000 and 10,601,000 (Roughly 13:00-18:00 UTC on August 5th, 2020) rather than our larger dataset including all transactions from 2020. This smaller dataset still contains 188,920 transactions that interact with smart contracts, 5667 of which interact with Forsage.

tions paid a higher average transaction fee of 0.0116 ETH with a standard deviation of 0.0108 ETH and a median of 0.00883 ETH. Forsage users pay more than four times as much on average as other smart contract users.

The most gas-expensive EVM operations/opcodes are those that create a new contract (CREATE, CREATE2); store, change, and access data into persistent on-chain state (SSTORE, SLOAD), and call contract functions or send money to other users in the network (CALL) [151]. Every transaction that interacts with Forsage through its two main functions, `registrationExt` and `buyNewLevel`, uses two of these three most expensive categories, often multiple times: they make use of persistent storage via SSTORE and SLOAD operations, and send money to other users on the network using Solidity operations that compile to the CALL opcode. Forsage uses an average number of CALL operations, but makes heavy use of SSTORE and SLOAD, as shown in Table 6.1.

Figure 6.3 shows a superimposed histogram of Forsage transactions relative to all Ethereum transactions. The higher gas consumption associated with Forsage results in higher transaction fees overall, as demonstrated by the right-shifted peak in the Forsage curve relative to that of all ETH transactions.

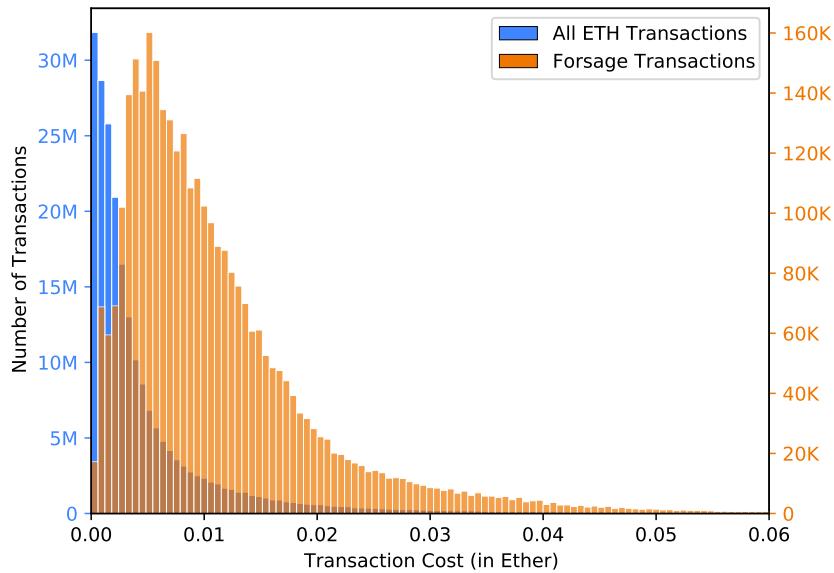


Figure 6.3: Histogram of transaction costs on the Ethereum blockchain—from January 31, 2020 to January 14, 2021—that involve successful smart contract function calls. Blue bars indicate the number of all transactions that paid fees within the given bucket, while orange bars indicate the same data, but only for transactions sent to the Forsage smart contract. The data excludes outlier transactions with fees above 0.06 ETH, which is above the 99th percentile of all transactions from this time period.

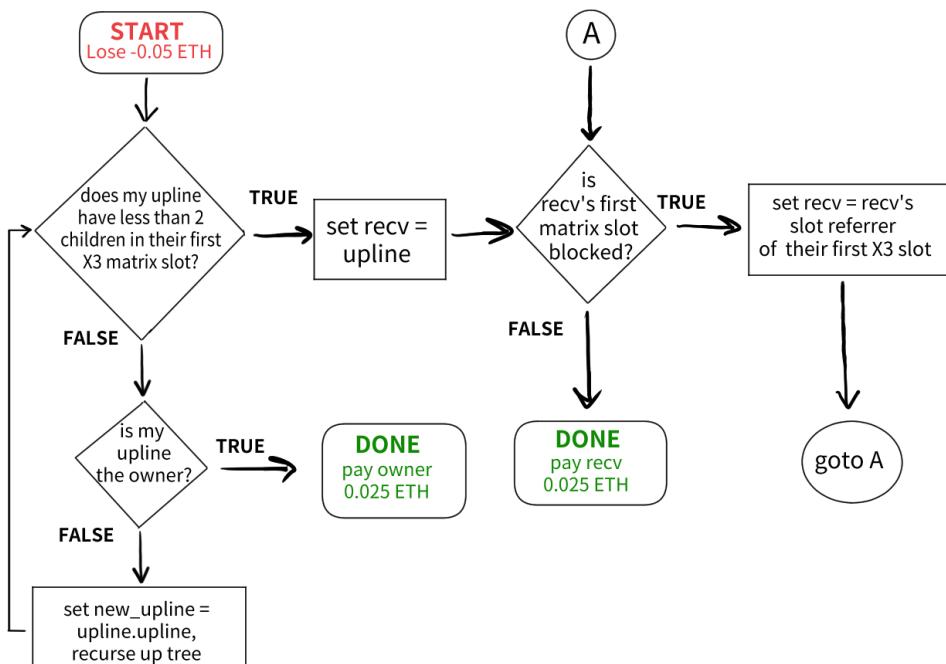


Figure 6.4: Flow chart for the logic of who gets paid when a new user registers, in the X3 system. The BuyNewLevel function follows similar logic, but conditioned on the matrix slot being purchased, rather than the first slot.

Payment logic: There are three ways for a user to get paid in Forsage: (1) by referring new users to the system; (2) when users they have referred in the past buy an additional matrix slot at a level corresponding to one previously purchased by the referrer; and (3) when *spillover* occurs, a condition in the X4 matrix resulting from the slots of another user downstream in the pyramid being blocked. Whenever money is sent to the smart contract by one user, the contract atomically (i.e., in the same transaction) sends those funds to other users based on the logic described below. This allows Forsage promotional materials to claim that the contract “never stores users’ funds.”

When a user buys a new slot, the money they pay typically routes to the first found upline that also has that same slot open. Users are thus incentivized to buy new levels in order to refer users underneath them, which means a user can be generally successful by adding additional matrix slots just before referring additional users, and in general by recruiting as many users as possible.

Figure 6.4 show the logic determining who gets paid when a new user registers with the Forsage contract. The logic for purchases of new slots (`buyNewLevel`) is largely similar but depends on the slot purchased rather than the first one (e.g., if a user buys the third slot then the logic is conditioned on the status of their upline’s third slot).

The flowcharts in these figures show that uplines must keep their slots from becoming blocked, or payments will skip over them. To prevent a slot from becoming blocked, a user must buy the slot at the next level. This will also unblock an existing slot if it already has become blocked, and prevent the slot at $level - 1$ from ever becoming blocked again. Figure 6.5 shows the distribution of levels purchased in aggregate for all users in the Forsage contract, as well as the summed profitability for the group of users that purchased that many slot levels. In general users that purchased more levels were also the most profitable users: The average user of the contract purchased 2.13 levels, with a standard deviation of 2.89 and a median of 1 level purchased.

When a new user joins the system, their payment is split into two equal parts

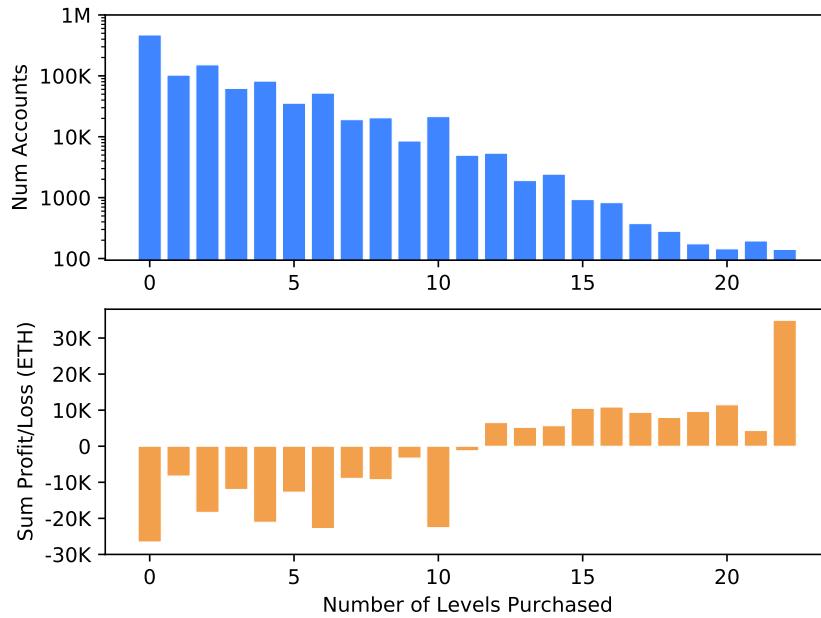


Figure 6.5: The distribution of how many users had unlocked a given number of levels in the contract (on top, and at log scale), and the collective amount of money gained or lost by the users who had unlocked this number of levels (on bottom, and at linear scale). Users that bought the most levels were on average the most profitable.

and the logic in the flowchart is applied to each half, with one half going through the X3 flowchart and one half through the X4 flowchart, to determine which other user(s) should get each half of the payment. If the direct upline of this new user is not blocked, then the upline gets the payment. If the upline has a blocked slot, the contract checks the upline's upline for that matrix slot level to see if it is blocked. This iterates through uplines until the contract finds one that is unblocked, which it then pays. The contract owner (i.e., the user that created the contract) is always unblocked, so the contract always finds a user to pay. This can sometimes result in the same user being payed twice (once by each half), or uncles and aunts being paid by their nephews and nieces in the tree if it has been previously scrambled. This condition is called *spillover*.

Spillover means that it is possible to earn money by receiving payments that should have gone to another user who had blocked slots. This passive earning is possible only in the X4 system, and only if a spillover recipient's upline is blocked and cannot currently receive payment. A given user's chance of spillover is unpredictable, because it depends on the actions of other users. In our analysis of

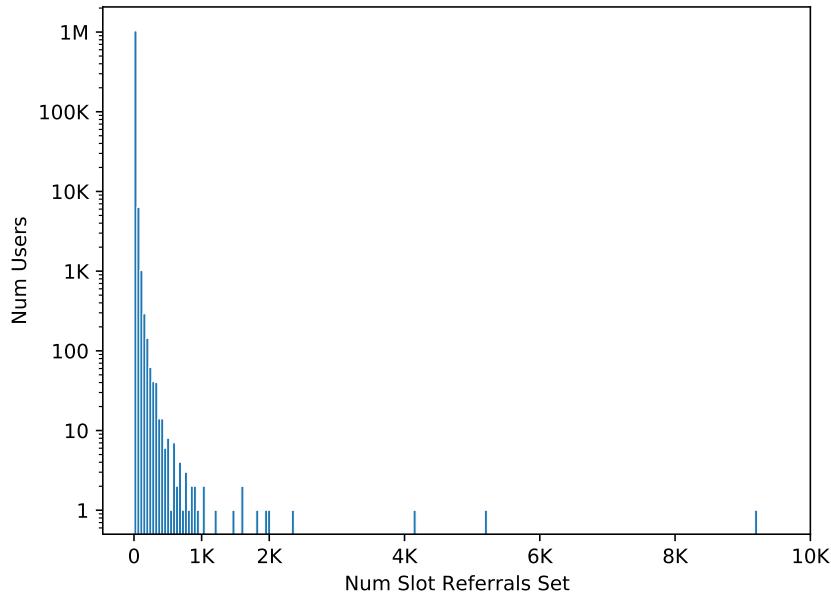


Figure 6.6: On a log scale, the total number of users (on the y-axis) acting as slot referrer for a given number of *other* users (on the x-axis), for both the X3 and X4 matrices. For example, one user (the contract owner, [0x81...](#)) is slot referrer for 9220 other users.

the transactions to Forsage from its conception until January 14th, 2021, we found that 35,251 transactions (only 1.08%) contained spillover payments. Of those transactions 63% were registrations, and the remaining 37% resulted from buying new levels.

Ethereum transaction costs are incurred by each interaction with the Forsage smart contract, and eat into users' profits. Any claims about user profit must thus take gas costs into account.

The privileged role of the owner: The Forsage contract is initialized so that the owner account (i.e., the creator of the contract, [81ca...](#)) has all matrix slots for both X3 and X4 opened for free. Likewise, the owner's slots can never become blocked. This creates ample opportunities for the owner to profit from the contract, which we confirm empirically in Section 6.5.

Beyond the ability to earn money by referring users, the owner also has additional opportunities to earn money passively. If a user sends the contract exactly 0.05 ether for registration without specifically calling the registration function, or calling a function that does not exist, that function call is rerouted to the registration function with the owner set as the user's upline. Likewise, if the upline gets replaced

as the referrer, it is always replaced with a user further up in the pyramid. Thus, as users refer others and have their slots blocked as a result, the upline for all users eventually converges to the owner of the contract. Finally, the logic that prevents the owner’s slots from becoming blocked also means that the owner’s children do not change once set. This means that the owner maintains the oldest users in the pyramid as children, which results in high spillover in the X4 matrix.

We found that the `slotReferrer` variable was set to the contract owner for 9220 slots in the Forsage contract. By comparison, the average Forsage user was set as the referrer for 4.14 other accounts (with a standard deviation of 15.92) and the median account was set as the referrer for one other account. Figure 6.6 shows the full distribution of referrers for all accounts.

6.5 Contract Measurement Study

In this section, we present the results of our measurement study of Forsage contract transactions, which encompasses all monetary transactions in the scheme. A description of our data collection process is in Section 6.4. We first present statistics capturing the degree of user interaction with the various Forsage contracts on Ethereum and Tron (6.5.1). We then present an analysis of the account behaviour and profits over the Forsage user population (6.5.2), in particular analyzing where funds are obtained and how funds flow through the five most profitable accounts.

6.5.1 Scheme statistics

Table 6.2 shows a summary of statistics for the four official Forsage contracts, and one additional contract, TRX Clone, which is a cloned version of the Ethereum Matrix contract operating on Tron. This clone launched before the official TRX Matrix contract, and has a different domain¹² but with graphics and style akin to the official website. The official Forsage website added a warning after the clone’s appearance, asking users to “beware of fake resources” and stating that the “forsage.io” website is the only official domain. In total, the table shows that the official Forsage contracts amassed over 267M USD within the first year of operation. Among all of

¹²forsagetrон.io

Contract	Total TXs	Unique sending addresses	Total coins	Total USD	Launch date	Address
ETH Matrix	3M	1M	721k	225M	Jan 31, 2020	0x5a...
TRX Clone	217k	78k	537M	14M	July 25, 2020	TJRv...
TRX Matrix	1M	342k	1B	31M	Sept 6, 2020	TREb...
TRX xGold	307k	105k	90M	2M	Nov 7, 2020	TA6p...
ETH xGold	37k	17k	8k	9M	Jan 4, 2021	0x48...

Table 6.2: Summary statistics of the four official Forsage smart contracts and one clone. The USD value was calculated by taking a sum of the payments per day and multiplying it by the average of the 24-hour high and low on the respective day.

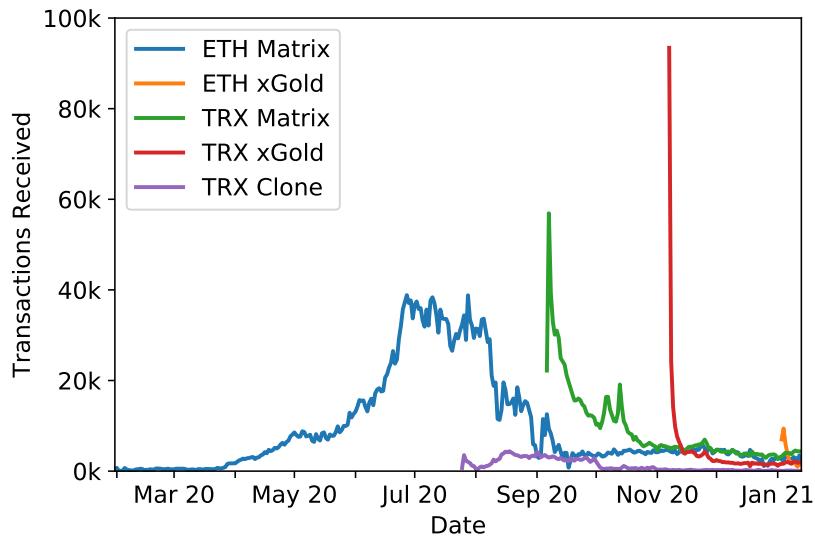


Figure 6.7: Number of transactions sent from users to the four Forsage contracts across Ethereum and Tron and to an unofficial Tron-based clone.

these contracts, the ETH Matrix contract brought in the most money and raised the highest amount on a single day: 3.7 million USD on August 1, 2020. The more recent xGold contracts (deployed on both Ethereum and Tron) were sent a combined 11.53 million USD in ETH and TRX in less than two months.

Figure 6.7 shows the number of transactions received by each contract over time. For each contract introduced after the original ETH Matrix one, we observe a large number of initial transactions followed by a substantial drop. We also see a

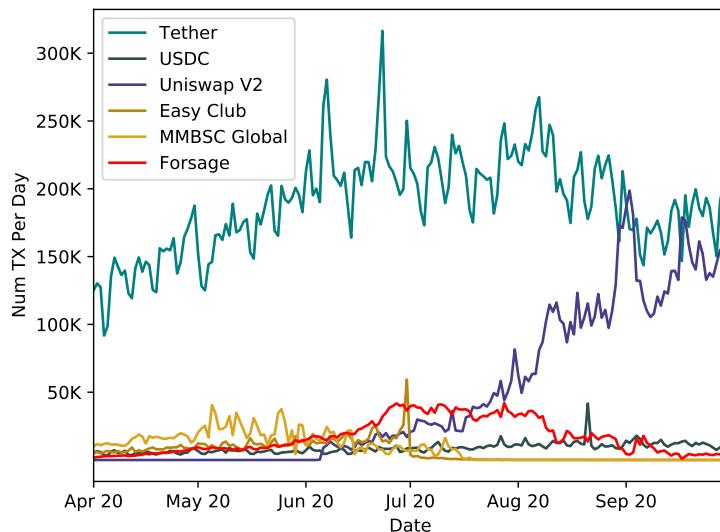


Figure 6.8: The daily transaction count associated with the six most transacted contracts between April 1 and September 30, 2020. Here Forsage refers to the ETH Matrix contract.

decline in the number of transactions sent to the original ETH Matrix contract after the other contracts become available. Given the relatively longevity and popularity of the ETH Matrix contract, we focus primarily on it for the remainder of this section.

To illustrate the popularity of Forsage, Figure 6.8 shows the number of daily transactions associated with the six most popular contracts across a six-month period in 2020. Of these contracts, Tether and USDC are stablecoins; Uniswap is a decentralized exchange; and Easy Club, MMBSC Global, and Forsage are believed to be scams/pyramid schemes. We can see that Tether is consistently the most popular contract and that for most of its peak from June to August, Forsage (as represented by ETH Matrix) had the second highest transaction rate among Ethereum smart contracts. This data is supported by Google Trends results for 2020: From April to August of 2020, Forsage had the highest search traffic globally of any of the smart contracts we studied, including both Tether and Uniswap, the two most heavily used smart contracts on the network as of the time of writing.

6.5.2 Account behavior and profitability

To understand how Forsage users obtained the funds needed to interact with the contract, we looked at the transactions that sent ETH to their accounts, and at when

their accounts first became active. Figure 6.9 shows the ETH received by Forsage users over time and the cumulative count of active Forsage-related accounts (i.e., the first time an account was used that later interacted with the Forsage contract), with a vertical line indicating when Forsage was deployed. It is clear that these accounts became active and began to receive substantially more ether after the deployment of Forsage; in fact, 98.89% of Forsage users had accounts that did not exist (or at least did not transact) before Forsage. We found a similar increase when looking at the number of transactions conducted by these users as well: prior to the deployment of Forsage, 11k accounts were involved in 278k transactions, but after Forsage’s release this increased to 1.04M users engaging in 16M transactions. While the curve in Figure 6.9 looks quite steep given the timescale, it in fact reflects a steady growth in the first appearance of accounts between April and August 2020, which aligns with the peak of Forsage we saw in Figure 6.8. Each of these months saw thousands of new accounts appearing per day, on average: 1659 in April, 3653 in May, 8272 in June, 10,798 in July, and 4987 in August. In contrast there were at most 20 new accounts appearing per day for each month in 2019 (except December, when there were 68).

To identify which types of services were the source of this money, we used tags from Etherscan. Of the ETH sent to Forsage users, over 56% (1.5M) came from untagged sources, and only 15% came from known exchanges, with 5% of this coming from the decentralized exchange Uniswap. As mentioned in Section 6.3, Forsage promotional material recommends that users obtain ETH from TrustWallet. This is a non-custodial service, which means accounts are associated with individual users rather than with the exchange. Thus, if most users followed this advice, we would expect to see that most of the ETH came from untagged sources.

Figures 6.10 and 6.11 show a histogram of all of the accounts that interacted with the ETH Matrix contract organized by the amount of money either gained or lost by each account (including the amount spent on transaction fees) as of January 14, 2021. In total, of the 1.04 million Ethereum addresses that took part in the ETH Matrix scheme, only 11.8% (123,979) earned a profit. These profitable accounts

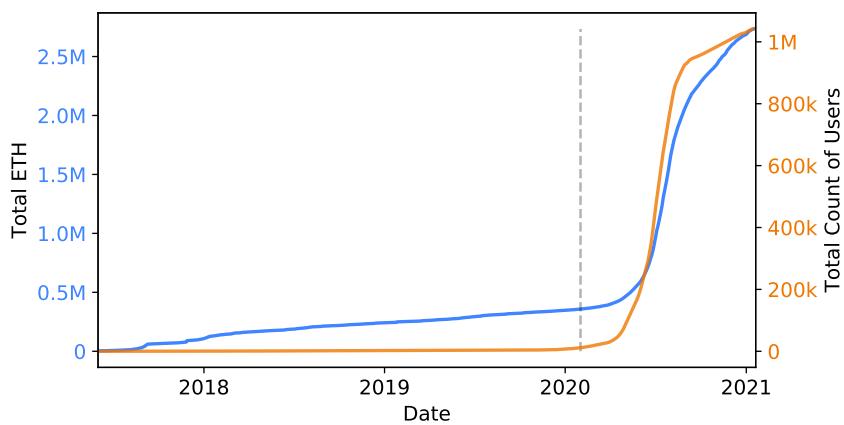


Figure 6.9: Total ether received by Forsage users over time and total number of Forsage users according to when their accounts were first used, with a dashed line indicating the Forsage creation date.

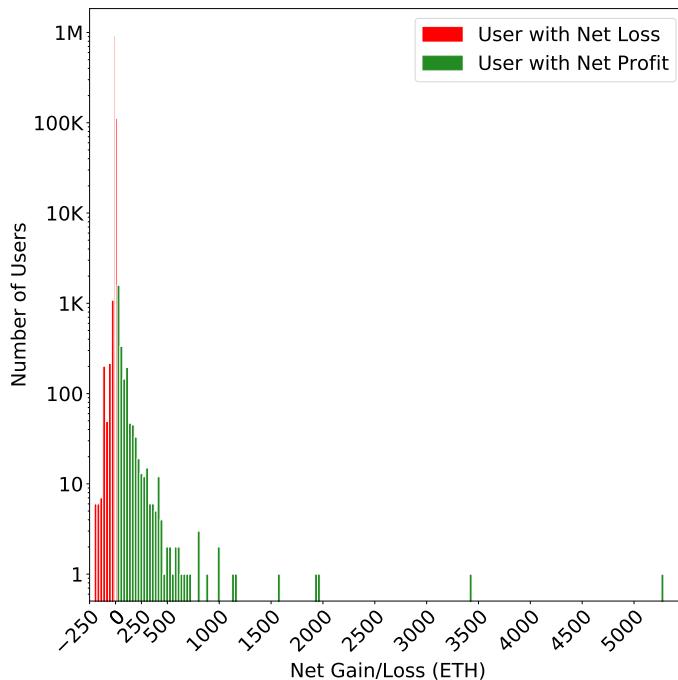


Figure 6.10: Profit/loss histogram of Ethereum accounts that interacted with the Forsage smart contract, on a log scale. This graph shows the number of accounts that made a profit or loss for each range of ETH. The majority of accounts incurred a small net loss, less than 1 ETH.

made 265,618.52 ETH collectively, and the loss-making accounts (919,194 in total) lost 305,785.44 ETH collectively (0.33 ETH on average). We revisit these profit-making accounts below. Users incur additional losses from the high gas fees paid for transacting with the contract, as explained in Section 6.4.

Profit-making accounts: The five addresses with the highest profits in Forsage can be found in Table 6.3. Perhaps unsurprisingly given our discussion in Sec-

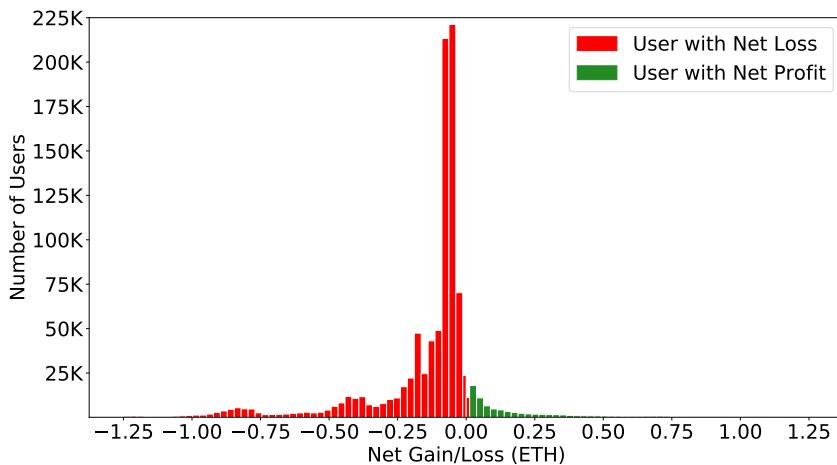


Figure 6.11: Profit/loss histogram of Ethereum accounts that interacted with the Forsage smart contract, centered around 0 and on a linear scale. The vast majority of user accounts that interacted with Forsage lost between 0 and 0.25 ETH, with the peak occurring between 0.038 and 0.063 ETH.

Address	Profit (in ETH)	Notes/First Seen
0x81...	5409.6	Owner of the contract
0x44...	3445.0	March 22, 2020
0xde...	1954.9	March 22, 2020
0x4a...	1943.2	January 31, 2020
0x59...	1573.0	June 4, 2020

Table 6.3: The five most profitable accounts that interacted with Forsage.

tion 6.4, the most profitable Forsage user is the owner of the contract, who earned 5409.6 ETH, or 2.04% of the total profits. Collectively, the five most profitable users made 14,325.7 ETH, or 5.4% of profits, despite representing only 0.0004% of users. The top 1000 users made 50% of the total profits.

Examination of the five most profitable addresses shows that the most profitable address is another Ethereum contract created by the owner of the ETH Matrix contract. Of the money received by this contract, 99% came from ETH Matrix. The fourth highest earner sent 9% of received ETH directly back to Forsage. In fact, if we follow all the addresses to which this user sent money, we see over 1321 ETH sent back to Forsage eventually. Similarly, the fifth highest earner sent 204 ETH directly back to Forsage.

Some of the top addresses interact directly with other known scams, such as Beurax.com and TorqueBot.net, meaning they sent or received coins directly from addresses associated with these scams. The top five profit-making accounts received

6.987 ETH from these scams.

Interestingly, the first transaction sent to the address that deployed Forsage was from [0xb1...](#), which is the Ethereum address that deployed Million.money. This suggests interaction between smart contract-based scam operators.

Finally, we consider the extent to which users who profited by interacting with the Forsage ETH Matrix contract also interacted with other Forsage contracts. The ETH xGold contract has 17,560 users, of which 17,129 (97.5%) also interacted with ETH Matrix. Furthermore, the highest earner in xGold was the third highest earner in Matrix, the fourth highest xGold earner was the seventh highest earner in Matrix, and the eighth highest earner in xGold was the second highest earner in Matrix. These three earners (all of which are within the ten wealthiest Matrix users) hold 21.85% of net profits in xGold. This suggests that at least some prominent users of Matrix did indeed migrate over to xGold.

6.6 Study of Forsage Community

Methodology: We studied the Forsage community by examining the presence of Forsage on social media. The Forsage website promotes official social media presences on Facebook, Instagram, Telegram, Twitter, and YouTube. All of these services have official APIs to collect data, but some of the research we conducted required manual interaction with the various social websites via a web browser, or more sophisticated data collection techniques like web scraping.

We manually watched YouTube videos to understand the claims that Forsage promotional videos make, as discussed in Section 6.6.1, and made requests to the public YouTube API for view count and other popularity-related data.¹³ To get a sense of Forsage’s Facebook and Instagram presence, we manually browsed various Facebook groups and official Instagram accounts and leveraged the Facebook and Instagram Graph APIs.¹⁴ Facebook group data is not available on the Graph API so we wrote a custom Python script leveraging the Selenium WebDriver browser automation tool to collect more in-depth data about Forsage Facebook groups and

¹³<https://developers.google.com/youtube/v3/docs/search/list>

¹⁴<https://developers.facebook.com/docs/graph-api/>

their users.¹⁵ This yielded a dataset of just over 5000 of the most recent members from the largest Facebook group dedicated to Forsage.¹⁶ Using the Twitter API for academic researchers,¹⁷ we were able to scrape all tweets with the word “Forsage” from January 1, 2020 until February 13, 2021. We used the official Telegram API¹⁸ to collect information about telegram groups related to Forsage.

Community size Forsage has a substantial presence on the social network sites that they target. This includes:

- *Facebook*: 131 active Facebook groups with titles or descriptions including “Forsage,” containing 403,029 distinct Facebook members.
- *Instagram*: 24 Instagram accounts with Forsage in the username, disseminating information about Forsage to 24,747 followers of these accounts, with an additional 78,220 posts on the Instagram #forsage hashtag.
- *Telegram*: 285,788 people spread across 49 different channels on Telegram dedicated to Forsage.
- *Twitter*: Our collected Twitter dataset included 85,085 tweets from 21,746 unique accounts, including 513 accounts on Twitter that feature Forsage in the account name.
- *YouTube*: 57,551 video results from 325 different YouTube channels.

The Forsage website also features a “community” subdomain¹⁹ that hosts a tips and tricks section, blog-post style news, a frequently asked questions section, “academy courses” that include video lectures on how to be an effective multi-level-marketer, and a Stack-Overflow-like site where users can ask questions and “Forsage Community Authors” answer.

A substantial amount of the Forsage online social media ecosystem may be driven by bots. We ran the University of Indiana’s Observatory on Social Media

¹⁵<https://www.selenium.dev/>

¹⁶<https://www.facebook.com/groups/forsageinformationgroup>

¹⁷<https://developer.twitter.com/en/docs/twitter-api/tweets/search/introduction>

¹⁸<https://core.telegram.org/>

¹⁹<https://community.forsage.io/>

(OSoMe) Botometer tool [131] on our collected dataset of tweets and found that the tool identified roughly 47% of the Forsage-related tweets we collected as coming from likely bot accounts. For comparison, in March of 2017, Varol et al. [145] used an earlier version of the Botometer tool to perform a measurement study across all of Twitter and found that “between 9 and 15% of active Twitter accounts are bots.”

Type	Claim	Appears	Cumulative Views
Wealth	Forsage users make money forever.	3/10	425,356
	Forsage users make unlimited income.	3/10	449,429
	Forsage users make passive income.	3/10	247,344
	Forsage users can earn hundreds of ETH in the first few weeks or months.	4/10	558,617
Risk	Forsage is risk-free for users.	3/10	393,927
	No one can stop Forsage.	4/10	558,617
	Forsage is safe because the contract does not store funds.	4/10	530,165
	Forsage is scam-proof.	3/10	393,927
Ethereum	The video explains what Ethereum is for new users.	5/10	637,881
Education	The video explains what a smart contract is for new users.	5/10	637,881
How to use Forsage	Successful Forsage users open at least 3 slots per program to start (0.2 ETH).	6/10	745,960
	Users should buy more slots (send Forsage more money) as soon as they earn.	5/10	654,727
	The more slots you open (money you send Forsage), the more you will earn.	4/10	511,858
	If you do not keep opening slots (sending money to Forsage), you will not earn.	5/10	444,539

Table 6.4: We coded repeated claims that appear across the top 10 most viewed, English language videos on YouTube, which mention ”Forsage” in their title to measure user expectations when joining Forsage.

6.6.1 Analysis of Forsage YouTube Promotion

Forsage promotional materials offer a window into users’ expectations for the contract. They also provides insight into how mention of the technical properties of blockchain technology is harnessed to manipulate novice users. We find that the information gap between those who understand blockchain technology and the

Country	Facebook	Twitter	YouTube
Nigeria	84	4878	3
Philippines	272	668	14
India	97	488	88
United States	45	1019	26
Indonesia	17	203	8
TOTAL	771	10200	216

Table 6.5: Top five countries with the highest absolute level of Forsage user engagement.

User engagement here is measured as a country’s total number of Facebook observed users in the most popular Forsage Facebook group, plus its analogous number of Twitter observed users that tweeted about Forsage in 2020, and YouTube data for the number of YouTube channels with geo-tagged locations that produced videos with Forsage in the title of the video.

broader community provides opportunities for scammers.

YouTube is a primary promotional channel for Forsage. Each participant joining Forsage is referred to an official YouTube video explaining the program [109]. We searched YouTube for English language videos with “Forsage” in the title and tracked the claims that repeat across videos to measure user expectations for Forsage. The search for most viewed videos about Forsage also returned promotional videos in Tagalog, Russian, Hindi, Tamil, Bangala, Telugu, Indonesian, and Spanish. Quasi-official (they share the same branding) Telegram chat groups for Forsage news exist in English, Spanish, French, Italian, Russian, Arabic, Portuguese, Hindi, Tamil, German, Azerbaijani, and Turkish.

Recommendation algorithms, like the one used by YouTube for search results, work in terms of popularity measured in views. The most viewed videos on YouTube are the most likely to be seen by users. We selected the top ten videos by views to qualitatively measure what users who search for informational videos about Forsage would see and hear about the program and gain a sense of participant expectations. We did so by coding the claims asserted about Forsage in these videos. We focused on just the top ten videos because coding claims is a labor-intensive, manual process. A researcher watched each video and noted if each video contained any instance of certain claims (see Table 6.4 and Table 6.6). Each video was watched and coded twice to ensure accuracy.

The top ten YouTube videos we coded had between 267,008 views (1st) and

Rank	Title	Views
1	Forsage Overview: Earn Ethereum Daily!	267008
2	Forsage Presentation - How does Forsage work	120425
3	Forsage Smart Contract - \$735 Made Without Referring Anyone	113677
4	FORSAGE: HOW TO EARN WITHOUT RECRUITING ANYONE IN FORSAGE	117931
5	Forsage Smart Contract Review - Is It A SCAM Or Legit Ethereum MLM?	106261
6	FORSAGE.io - BIG SPECIAL EVENT	91973
7	Forsage Smart Contract \$1,778 Made Without Referring a Single Person	91188
8	Forsage Review - Is Forsage a Scam or Legit?	79264
9	Smartway Forsage REVIEW - First Ever SCAM PROOF Program	64923
10	how to make money on forsage without referring anyone	61996

Table 6.6: Top 10 Forsage videos from the official channel ordered by views.

61,996 views (10th). Beyond the videos we coded, the 11th most viewed video had just over 50,000 views²⁰ and the 20th had 33,000 views.²¹

The top 10 “Forsage” YouTube videos by views as of December 14, 2020 (see Table 6.6) fit into three categories: official promotion, user-led recruitment, and user reviews. Two of the videos were official promotion posted to Forsage’s YouTube channel [109, 110]. Table 6.4 shows the repeated claims across the top ten videos.

In recruiting new users, Forsage promoters pointed to users who earned tens of thousands of dollars per day and hundreds of thousands of dollars per month, showing images of successful users’ Forsage dashboards displaying six-figure returns. Forsage official promotion videos highlight the immutable nature of the smart contract and the transparency of Ethereum as proof that Forsage cannot be a scam. They also make claims about the life-changing wealth and unstoppable, passive income that users could unlock from the Forsage contract.

Forsage promotional videos also provide basic explanations of blockchains, Ethereum, smart contracts, and how to use a cryptocurrency wallet to pay the contract, implying that they expect users to be cryptocurrency novices. Only one of the top ten videos identifies Forsage as a scam and warns users against using it.

²⁰<https://youtu.be/aGi5G5mTCUM>

²¹<https://youtu.be/9vI0YRSLaHI>

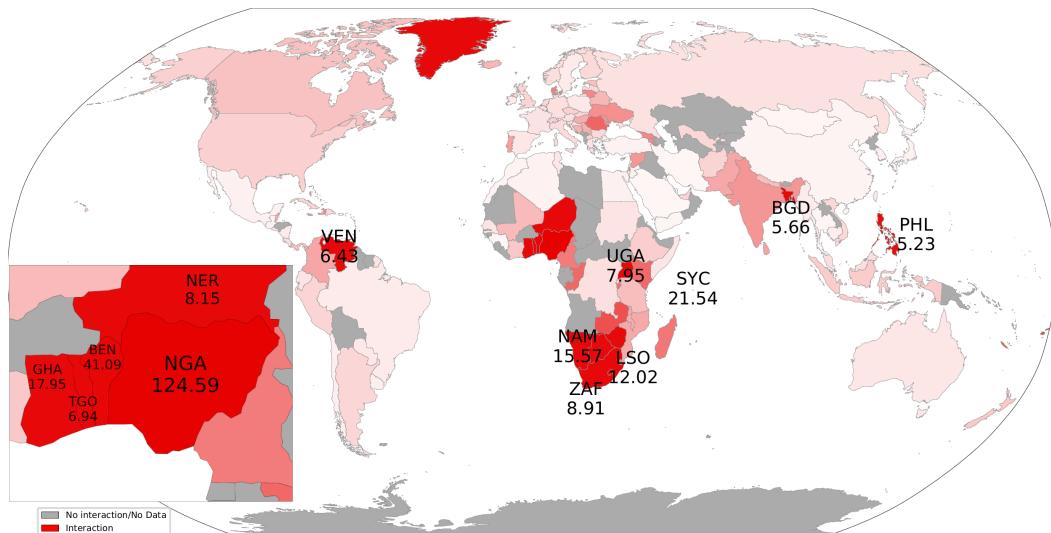


Figure 6.12: Forsage social media interaction heat map by country. Country labels indicate the ISO-alpha-3 name of the country and the number of Forsage users per 100k people in that country. The data reflects the public location of members in a popular Forsage Facebook group and Twitter users that tweeted about Forsage. Countries depicted in gray had no Forsage interaction. The intensity of color from white to red is scaled linearly from the 0th percentile of data to the 90th percentile, and everything above 90% of the data is colored the same shade of dark red. This slightly understates the relative depth of penetration in outlier countries like Nigeria.

Many of the incorrect claims made in the Forsage promotional YouTube videos also appear on the Forsage website and in the questions section of the official Forsage Community website.

6.6.2 Forsage user geography

Since transactions on the Ethereum network do not carry any inherent geographic metadata, we turned to social media analysis in order to gain a sense of the geographic placement of people interested in Forsage. In the data we collected on members of Forsage-related Facebook groups, we found 771 users that publicly listed a country location on their Facebook profile. We also found 10,200 unique Twitter accounts that publicly posted their geographic location. YouTube does not expose information about geographic location of the consumers of YouTube videos, but YouTube channels that produce videos can choose to include country location in their channel profile. We summarize this data for the five countries with the highest number of active users in Table 6.5. Despite having a substantial population and being the nationality of the founders of Forsage, Russia was not a large source of

Twitter or Facebook content, although the country did produce a large number of YouTube videos and content about Forsage.

The high number of Forsage users in the Philippines may explain why the Philippines SEC took action to raise awareness about the malicious intent behind Forsage [51, 134], unlike other countries. Likewise, Nigeria has high penetration rates for both cryptocurrency and Forsage, and has recently banned cryptocurrency payments from its banking sector [3]. While each of these five countries had high Forsage activity in absolute terms, they also have large populations. We thus normalized our Facebook and Twitter data relative to the specific populations on each service for each country (i.e., the number of people per country divided by a public estimate of the number of Facebook and Twitter users in that country) to get a sense of the number of Facebook and Twitter users, per 100,000 users, that interacted on each platform with the Forsage topic. Statistics for the number of Facebook and Twitter users per country came from Miniwatts Marketing Group, WeAreSocial, and Hootsuite [58, 68]. We did not include the YouTube data at this stage as it was too small to be useful. We gave equal weight to the numbers for Facebook and Twitter to produce the heat map in Figure 6.12.

Our normalized data showed that Forsage is most popular in Nigeria and the African continent, the Philippines, and Venezuela. Greenland, the Seychelles, and some Caribbean islands may appear to have heavy Forsage penetration, but may be outliers due to small population sizes. Google Trends traffic and geographic data agree with our conclusions: Google Trends shows the greatest amount of population-adjusted search traffic in Nigeria and surrounding West African countries, and shows a peak in user search interest in July 2020, which is when we observed a similar peak in transactions involved Forsage in Figure 6.7.

Familiarity with cryptocurrency does not appear to have any positive or negative correlation with interest in Forsage: The 2021 Statista Global Consumer Survey [26] lists the top countries globally with the reported highest number of cryptocurrency users. Vietnam (#2) and China (#3) both had relatively high levels of cryptocurrency use, but low levels of interest in Forsage. Similarly, familiarity with

cryptocurrency does not appear to prevent people from falling for the Forsage scam, as in the case of Nigeria and the Philippines (#1 and #3 globally for cryptocurrency usage). Nigeria may be a special case, as Statista found that almost a third of Nigerians said they used cryptocurrency, far beyond most countries. It is also an outlier in the data for interest in Forsage.

6.7 Proposed Solutions

6.7.1 Targeted education

From our analysis of Forsage user locations in Section 6.6.2, the majority of Forsage victims are located in only a few countries. This concentration lends itself well to a targeted education campaign and warnings from local financial leaders about the Forsage scam. For example, a simple user dashboard showing the number of Forsage users who lose money from the contract—more than 88% as of January 15th, 2020—could serve as an effective tool to combat disinformation from Forsage promoters about the wealth users can amass. Such statistics may be more effective than general warnings such as that issued by the Philippines SEC (see below).

6.7.2 Law enforcement and regulation

Past cryptocurrency pyramid schemes, including Plustoken, Wetoken, Onecoin, and Bitconnect, have collapsed as a result of government sanctioning, which included the arrest or warrants for the arrest of the founders and leadership [14, 59, 89, 112]. Similar attempts have been made around the world in regards to Forsage. On June 30, 2020, The Philippines Securities and Exchange Commission (PSEC) issued numerous warnings declaring that Forsage was not a registered entity within their jurisdiction and was operating without a license. On September 30, 2020 the PSEC released a public announcement, mentioning that Forsage was publicly selling securities as investment contracts without a license [51, 52, 133, 134]. The PSEC served a cease-and-desist order. Forsage refused to comply, responding that they “are outside the Commission’s jurisdiction.” On March 22nd 2021, the Commissioner of Securities and Insurance of the U.S. state of Montana ordered Forsage to cease and desist from operating a pyramid scheme in Montana [34, 35].

To date, the authors of this paper are unaware of any public arrests made in relation to the Forsage contract. The contract authors continue to profit and their Ethereum addresses actively submit transactions to the network.

6.7.3 Voluntary blocklisting

Previous research has shown blocklisting can effectively combat scams and illicit activity. Moser et al. found that transaction blocklisting of illicit cryptocurrency funds is an effective additional layer above existing anti-money laundering (AML) and know-your-customer (KYC) requirements for cryptocurrencies [104]. Previous research in illicit online pharmaceutical sales found that the payment processing services are the most fragile part of the scam [92]. These services play a similar role in online pharmaceutical sales to fiat-accepting cryptocurrency exchanges in Forsage, suggesting that access to exchanges, which could be revoked with blocklisting, may be the the most fragile part of the scam. Crypto Defenders Alliance (CDA)²² and CryptoSafe Alliance²³ are two examples of groups that operate a blocklist.

On the other hand, blocklists can be biased and enable forms of censorship, and addresses that are blocked in one region may not be considered suspicious or criminal in other regions. To understand how professionals navigate these tensions, we spoke to an anti-money laundering cryptocurrency investigator at a high profile exchange. This expert expressed a belief that it is the responsibility of law enforcement and regulators to comment on whether or not an address should be blocked, and that it would be unfair and unjust to hold a user's funds without an explicit request from law enforcement or a court of competent jurisdiction. Nevertheless, some exchanges have joined the alliances mentioned above, due to the time and resources required to maintain a dedicated list of blocked addresses themselves.

6.8 Future work

There are a number of areas this work can be extended to. Due to limited time and resources we were unable to further delve into the xGold contract, thus we

²²<https://cryptodefendersalliance.com/>

²³<https://www.cryptosafe.org/>

leave the exploration of this contract for future research. Since the public release of this research, the Forsage group has released two products onto Binance USD blockchain, the matrix product and a brand new platform titled *Forsage xXx*. Future research can look into analysing these two newer products. Our research focused on analysing a single blockchain based pyramid scheme, future work can look into analysing other pyramid schemes and related scams in Ethereum overall, in a way to quantify the potential pyramid schemes on the network.

6.9 Conclusions

We presented an in-depth measurement study of Forsage, a smart-contract pyramid scheme. Forsage is currently active and was at one time the second most actively used contract in Ethereum.

We found that community claims regarding the open and verifiable nature of Forsage are belied by the contract’s considerable complexity. Our study consequently required a number of different data gathering approaches. It also required the creation of new tools—of potential independent interest and to be open-sourced—to analyze the state of the Forsage contract. Thanks to these tools, our study provides detailed insights into the mechanism design, transaction costs, and other features of Forsage.

Among our key findings were that the vast majority of Forsage accounts—over 88%—incurred losses, for a combined total loss of 305,785 ETH. The contract owner, in contrast, earned over 5000 ETH (well over 1M USD), while a small number of other accounts at the top of the pyramid earned similarly large sums.

Our analysis of Forsage promotional materials reveals that scammers in the Forsage community have taken advantage of misconceptions and misinformation about blockchain technology, using properties like open-source code and transaction transparency as a source of legitimacy with users who lack the skills necessary to understand the contract’s behavior. Our analysis of Forsage on social media shows geographically distinct communities of scammers and victims, with the scammers based primarily in Russia and victims apparently located mainly in Nige-

ria, southern Africa, the Philippines, Venezuela, Indonesia, and India.

Public warnings about Forsage by entities such as the Philippines SEC have had little apparent effect. We show that Forsage creators have launched new and currently lucrative Forsage variants, some now on blockchains other than Ethereum. We hope that our findings can help stem this spread. In addition to providing insights that may serve to educate potential victims, our study demonstrates highly concentrated earnings among top-earning accounts, suggesting that targeted block-listing could be an effective step to slow the growth of Forsage and contracts like it.

Chapter 7

Conclusion

This thesis has presented techniques to empirically analyse privacy and crime in blockchain technologies: analysing the privacy and usage of Zcash; tracking transactions moving across chains; and studying a high profile and ongoing smart-contract pyramid scheme, Forsage.

In Chapter 4 we presented an analysis of privacy and measurements on Zcash - a privacy-focused cryptocurrency forked from Bitcoin. Our analysis demonstrated that a user's privacy when interacting with the shielded pool is greatly determined by the actions of the surrounding users. User addresses can be clustered using known and novel heuristics. The hacker collective, TheShadowBrokers, used Zcash as one resource to sell vulnerabilities and tools. This chapter offers one of the first academic insights into the working of this privacy coin, demonstrating that privacy coins do not alleviate the anonymity risks demonstrated in non-privacy cryptocurrencies.

We then addressed the issue of cross-currency tracking in Chapter 5. We exhibited working examples of using our novel heuristics to follow coin ownership across different cryptocurrencies and examined and clustered entities within the ecosystem. This concluded with case studies, revealing that multiple scammers used the system to move their funds, and presenting measurements to showcase how privacy coins are used in the system.

Finally, in Chapter 6 we presented an in-depth measurement study of a smart-contract pyramid ecosystem that processed \$267 million USD worth of Ethereum.

The study was conducted using a multi-faceted approach. We revealed the inner workings of one of the obfuscated smart-contracts, measured the profits and losses, revealed the broadness of the global marketing scheme, identified the failed efforts by law enforcement to shutdown the enterprise, and proposed potential countermeasures.

7.1 Future Directions

In this section, we list some interesting areas of research that could be explored to continue research efforts in this field.

Anonymity of Zcash The research in Chapter 4 was conducted between 2017 and 2018. As previously mentioned, the developers have since made multiple changes to Zcash, such as improvements to the underlying cryptography which includes additions of new shielded pools. In addition, they have given grants to external developers to create user friendly mobile wallets which allow transactions with the shielded pool. Future research might look at potentially repeating our work to identify whether the ecosystem has changed and whether the analytics upon the shielded pool still work.

Researching Scams In Chapter 6 we focused our analysis on deconstructing the Forsage matrix contract. The platform has since expanded their offerings with additional schemes called *xGold* and into the Binance USD platform with *xXx*. Future research on pyramid schemes could extend our work and analyse the inner workings of these new schemes.

Privacy with Taproot Taproot [152] is an upgrade in Bitcoin that is expected to launch in November 2021. As one of the first upgrades to be approved by miners in over four years, its aim is to achieve better transaction privacy and efficiency, and to improve the potential for Bitcoin-based smart contracts. A new discussion [6] reveals that this update may not alter the current effectiveness of clustering, however, it will create new opportunities for new clustering heuristics to be applied to Taproot-specific scripts.

7.2 Closing Thoughts

We have concluded that heuristics-based approaches can be applied to analysing privacy and crimes in blockchain-based systems. We demonstrated that privacy-focused cryptocurrencies do not alleviate previous privacy risks and that address tracking and clustering is still possible in these ecosystems. Cross-chain payment systems do not act as a shield against preventing tracking of payments. Those involved in illicit activities in cryptocurrencies can, to some extent, have their transactions analysed and tracked. The methodologies described in this thesis can be used by agencies in order to detect crime, and by developers to test and improve the privacy of payment systems.

Bibliography

- [1] Adrian Colyer. *An empirical analysis of anonymity in Zcash*. <https://blog.acolyer.org/2018/09/14/an-empirical-analysis-of-anonymity-in-zcash/>. 2018.
- [2] John Kevin Adriano. *Fake invoice email with Html attachment spreads Locky ransomware*. <https://www.trendmicro.com/vinfo/us/threat-encyclopedia/spam/3621/fake-invoice-email-with-html-attachment-spreads-locky-ransomware>. Sept. 2017.
- [3] Hannah Akuiyibo. “Nigeria’s Crypto Ban Fuels Mistrust in Government”. In: *CoinDesk* (Feb. 22, 2021). <https://www.coindesk.com/nigerias-crypto-ban-fuels-mistrust-in-government>.
- [4] Tom Alford. *Bitconnect Scam: The \$2.6 BN Ponzi Scheme - 2019 Update*. <https://totalcrypto.io/bitconnect-scam/>.
- [5] *Alphabay will accept Zcash starting July 1st, 2017*. DarkNetMarkets Reddit post. https://www.reddit.com/r/DarkNetMarkets/comments/6d7q81/alphabay_will_accept_zcash_starting_july_1st_2017/. 2017.
- [6] Alexi Anania and Ken Hodler. *The Impact of Taproot and Schnorr on Address Clustering Analysis of Bitcoin Transactions*. http://blockchain.cs.ucl.ac.uk/wp-content/uploads/2020/04/UCL_CBT_DiscussionPaper_Q12020_Anania_2020.pdf. Mar. 2020.
- [7] Nate Anderson and Cyrus Farvar. *How the feds took down the Dread Pirate Roberts*. <https://arstechnica.com/tech-policy/2013/10/how-the-feds-took-down-the-dread-pirate-roberts/>. 2013.

- [8] Elli Androulaki, Ghassan O. Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. “Evaluating User Privacy in Bitcoin”. In: *Financial Cryptography and Data Security*. Ed. by Ahmad-Reza Sadeghi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 34–51.
- [9] Kristov Atlas. *CoinJoin Sudoku*. <http://www.coinjoinsudoku.com>.
- [10] Adam Back. *A partial hash collision based postage scheme*. <http://www.hashcash.org/papers/announce.txt>. Feb. 1997.
- [11] M. Bartoletti, B. Pes, and S. Serusi. “Data Mining for Detecting Bitcoin Ponzi Schemes”. In: *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. 2018, pp. 75–84.
- [12] Massimo Bartoletti, Salvatore Carta, Tiziana Cimoli, and Roberto Saia. “Dissecting Ponzi schemes on Ethereum: Identification, analysis, and impact”. In: *Future Generation Computer Systems* 102 (2020), pp. 259–277.
- [13] Anna Baydakova and Danny Nelson. *How Bitcoin Is Becoming a Lifeline for Cubans*. <https://www.coindesk.com/podcasts/borderless/development-of-cryptocurrency-in-cuba>. Mar. 2021.
- [14] Nick Bel. *The Most Famous Financial Pyramids in the Crypto World*. <https://cointelegraph.com/news/the-most-famous-financial-pyramids-in-the-crypto-world>. July 2020.
- [15] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. “Zerocash: Decentralized anonymous payments from bitcoin”. In: *Proceedings - IEEE Symposium on Security and Privacy*. 2014.
- [16] Karolina Bergman and Saeed Rajput. “Revealing and Concealing Bitcoin Identities: A Survey of Techniques”. In: *Proceedings of the 3rd ACM International Symposium on Blockchain and Secure Critical Infrastructure*. 2021, pp. 13–24.

- [17] Jonathan Berr. '*WannaCry*' ransomware attack losses could reach \$4 billion. <https://www.cbsnews.com/news/wannacry-ransomware-attacks-wannacry-virus-losses/>. May 2017.
- [18] Alex Biryukov and Daniel Feher. "Privacy and linkability of mining in zcash". In: *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE. 2019, pp. 118–123.
- [19] Alex Biryukov, Ivan Pustogarov, Fabrice Thill, and Ralf-Philipp Weinmann. "Content and popularity analysis of Tor hidden services". In: *2014 IEEE 34th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE. 2014, pp. 188–193.
- [20] Alex Biryukov, Ivan Pustogarov, and Ralf Philipp Weinmann. "Trawling for tor hidden services: Detection, measurement, deanonymization". In: *Proceedings - IEEE Symposium on Security and Privacy*. 2013, pp. 80–94.
- [21] Bitcoin Exchange Guide News Team. *Zcash Crypto Transactions on Bitfinex to Shadow Brokers For NSA Stolen Code & Hacking Tools May be Untraceable*. <https://bitcoinexchangeguide.com/zcash-crypto-transactions-on-bitfinex-to-shadow-brokers-for-nsa-stolen-code-hacking-tools-may-be-untraceable/>. 2018.
- [22] Olivier Boireau. "Securing the blockchain against hackers". In: *Network Security* 2018.1 (2018), pp. 8–11.
- [23] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies". In: *Proceedings - IEEE Symposium on Security and Privacy*. Vol. 2015-July. 2015, pp. 104–121.
- [24] Sean Bowe. *Cultivating Sapling: Faster zk-SNARKs*. <https://electriccoin.co/blog/cultivating-sapling-faster-zksnarks/>. July 2019.
- [25] Mark T Bradshaw, Scott A Richardson, and Richard G Sloan. "Pump and dump: An empirical analysis of the relation between corporate financing

- activities and sell-side analyst research”. In: *Available at SSRN 410521* (2003).
- [26] Katharina Buchholz. *How Common is Crypto?* <https://www.statista.com/chart/18345/crypto-currency-adoption/>. Online; accessed 21 March 2021. 2021.
- [27] JP Buntinx. *The Shadow Brokers Only Accept ZCash Payments for Their Monthly Dump Service.* <https://themerkle.com/the-shadow-brokers-only-accept-zcash-payments-for-their-monthly-dump-service/>. May 2017.
- [28] Vitalik Buterin and Vitalik Buterin. “A next-generation smart contract and decentralized application platform. Ethereum white paper”. In: (2014).
- [29] Bytecoin. <https://bytecoin.org>.
- [30] David Chaum. “Blind signatures for untraceable payments”. In: *Advances in cryptology*. Springer. 1983, pp. 199–203.
- [31] Binjie Chen, Fushan Wei, and Chunxiang Gu. “Bitcoin Theft Detection Based on Supervised Machine Learning Algorithms”. In: *Security and Communication Networks* 2021 (2021).
- [32] Weili Chen, Zibin Zheng, Jiahui Cui, Edith Ngai, Peilin Zheng, and Yuren Zhou. “Detecting Ponzi Schemes on Ethereum”. In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW 18* (2018).
- [33] Nicolas Christin. “Traveling the Silk Road: A Measurement Analysis of a Large Anonymous Online Marketplace”. In: *Proceedings of the 22nd international conference on World Wide Web* July 2012 (2013), pp. 213–224.
- [34] Office of the Commissioner of Securities and Insurance State of Montana. *Commissioner Downing Orders For sage to Cease and Desist.* <https://csimt.gov/news/commissioner-downing-orders-forsage-to-cease-and-desist/>. Mar. 22, 2021.

- [35] Office of the Commissioner of Securities and Insurance State of Montana. *US State Issues Cease and Desist Order against DApp Forsage*. <https://csimt.gov/news/us-state-issues-cease-and-desist-order-against-dapp-forsage/>. Apr. 7, 2021.
- [36] 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.
- [37] James Cook. *The FBI Just Started A Second Wave Of Silk Road Arrests*. <http://uk.businessinsider.com/fbi-silk-road-seized-arrests-2014-11>. 2014.
- [38] Joseph Cox. *Cryptocurrency Transactions May Uncover Sales of Shadow Broker Hacking Tools*. <https://www.vice.com/en/article/j5k7zp/zcash-shadow-brokers-uncover-hacking-tool-sales>. 2018.
- [39] *Cryptocurrency Market Capitalizations*. <https://coinmarketcap.com/>.
- [40] Wei Dai. *B-Money*. <http://www.weidai.com/bmoney.txt>. 1998.
- [41] *Dash*. <https://www.dash.org/>.
- [42] Jesse Dunietz. *The Imperfect Crime: How the WannaCry Hackers Could Get Nabbed*. <https://www.scientificamerican.com/article/the-imperfect-crime-how-the-wannacry-hackers-could-get-nabbed/>. Aug. 2017.
- [43] History.com Editors. *History of Drug Trafficking*. <https://www.history.com/topics/crime/history-of-drug-trafficking>. May 2017.
- [44] John Ellson, Emden Gansner, Lefteris Koutsofios, Stephen North, Gordon Woodhull, Short Description, and Lucent Technologies. “Graphviz — open source graph drawing tools”. In: *Lecture Notes in Computer Science*. Springer-Verlag, 2001, pp. 483–484.

- [45] Voorhees Erik. *Shining Light on WSJ's Attack on ShapeShift and Crypto*. <https://shapeshift.com/newsroom/shining-light-on-wsjs-attack-on-shapeshift-and-crypto>. 2018.
- [46] Ittay Eyal. "The miner's dilemma". In: *Proceedings - IEEE Symposium on Security and Privacy*. 2015.
- [47] Yaya J. Fanusie and Tom Robinson. *Bitcoin Laundering: An Analysis of Illicit Flows into Digital Currency Services*. A memorandum by the Center on Sanctions and Illicit Finance and Elliptic. Jan. 2018.
- [48] Cyrus Farivar and Joe Mullin. *Stealing bitcoins with badges: How Silk Road's dirty cops got caught*. <https://arstechnica.com/tech-policy/2016/08/stealing-bitcoins-with-badges-how-silk-roads-dirty-cops-got-caught/>. Aug. 2016.
- [49] Finder.com. *What is BitConnect (BCC) and how does it work?* <https://www.finder.com/uk/bitconnect>.
- [50] David Floyd. *Zcash Privacy Weakened by Certain Behaviors, Researchers Say*. <https://www.coindesk.com/zcash-privacy-weakened-by-certain-behaviors-researchers-say>. 2018.
- [51] FORSAGE. <https://www.sec.gov.ph/advisories-2020/forsage/>. June 2020.
- [52] FORSAGE AND FORSAGE PHILIPPINES. <https://www.sec.gov.ph/cdo-2020/forsage-and-forsage-philippines/>. Sept. 2020.
- [53] Tron Foundation. "Tron: Advanced Decentralized Blockchain Platform. Whitepaper Version: 2.0." In: (2018). https://tron.network/static/doc/white_paper_v_2_0.pdf.
- [54] Jason Franklin, Vern Paxson, Adrian Perrig, and Stefan Savage. "An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants". In: *Proceedings of the 14th ACM conference on Computer and communications security* (2007), pp. 375–388.

- [55] Marius-Cristian Frunza. *Solving modern crime in financial markets: Analytics and case studies*. Academic Press, 2015.
- [56] Sean Gallagher. *Researchers say WannaCry operator moved bitcoins to “untraceable” Monero*. <https://arstechnica.com/gadgets/2017/08/researchers-say-wannacry-operator-moved-bitcoins-to-untraceable-monero/>.
- [57] Andy Greenberg. *Five Men Arrested In Dutch Crackdown On Silk Road Copycat*. <https://www.forbes.com/sites/andygreenberg/2014/02/12/five-men-arrested-in-dutch-crackdown-on-silk-road-copycat/>. 2014.
- [58] Miniwatts Marketing Group. *Internet World Stats: Usage and Population Statistics*. <https://www.internetworldstats.com/>. Mar. 29, 2021.
- [59] Samuel Haig. *PlusToken Scammer Implicated in China’s Second Ten-Figure Crypto Ponzi*. <https://cointelegraph.com/news/plustoken-scammer-implicated-in-chinas-second-ten-figure-crypto-ponzi>. May 2020.
- [60] JT Hamrick, Farhang Rouhi, Arghya Mukherjee, Amir Feder, Neil Gandal, Tyler Moore, and Marie Vasek. “The Economics of Cryptocurrency Pump and Dump Schemes”. In: *SSRN Electronic Journal* (Jan. 2018).
- [61] Kevin Helms. *\$1.1 Billion Crypto Ponzi: Masterminds of Wotoken Head to Prison in China – News Bitcoin News*. Nov. 2020.
- [62] Cormac Herley and Dinei Florêncio. “Nobody Sells Gold for the Price of Silver: Dishonesty, Uncertainty and the UndergroundEconomy”. In: *Economics of Information Security and Privacy*. Ed. by Tyler Moore, David Pym, and Christos Ioannidis. Boston, MA: Springer US, 2010, pp. 33–53.
- [63] Jordi Herrera-Joancomart. “Research and challenges on bitcoin anonymity”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8872 (2015), pp. 3–16.
- [64] Taofik Hidajat. “Predator and prey: Ponzi and pyramid investors”. In: *Facing Global Digital Revolution*. Routledge, 2020, pp. 147–151.

- [65] Stan Higgins. “Details of 5 million bitstamp hack revealed”. In: *CoinDesk*, July 1 (2015). <https://www.coindesk.com/unconfirmed-report-5-million-bitstamp-bitcoin-exchange>.
- [66] Thomas J. Holt. “Identifying gaps in the research literature on illicit markets on-line”. In: *Global Crime* 18 (2017), pp. 1–10.
- [67] Thorsten Holz, Markus Engelberth, and Felix Freiling. “Learning more about the underground economy: A case-study of keyloggers and drop-zones”. In: *Computer Security—ESORICS 2009* (2009), pp. 1–18.
- [68] Hootsuite and WeAreSocial. *Digital in 2021: National Reports*. <https://datareportal.com/library>. 2021.
- [69] Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. “Zcash Protocol Specification”. In: (2017), pp. 1–53.
- [70] Marie Claire Van Hout and Tim Bingham. “‘Silk Road’, the virtual drug marketplace: A single case study of user experiences”. In: *International Journal of Drug Policy* 24.5 (2013), pp. 385–391.
- [71] 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: *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2018, pp. 618–631.
- [72] *Investing in BitConnect Lending - Bitconnect*. <https://web.archive.org/web/20171119183834/https://bitconnect.co/bitcoin-information/19/investing-in-bitconnect-lending>. Feb. 2016.
- [73] *Investor Alerts and Bulletins: Beware of Pyramid Schemes Posing as Multi-Level Marketing Programs*. https://www.sec.gov/oiea/investor-alerts-bulletins/investor-alerts-ia_pyramidhtm.html. Oct. 2013.

- [74] Danushka Jayasinghe, Sheila Cobourne, Konstantinos Markantonakis, Raja Naeem Akram, and Keith Mayes. “Philanthropy on the Blockchain”. In: *IFIP International Conference on Information Security Theory and Practice*. Springer. 2017, pp. 25–38.
- [75] Don Johnson, Alfred Menezes, and Scott Vanstone. “The elliptic curve digital signature algorithm (ECDSA)”. In: *International journal of information security* 1.1 (2001), pp. 36–63.
- [76] US Department of Justice Office of Public Affairs. *Three North Korean Military Hackers Indicted in Wide-Ranging Scheme to Commit Cyberattacks and Financial Crimes Across the Globe*. <https://www.justice.gov/opa/pr/three-north-korean-military-hackers-indicted-wide-ranging-scheme-commit-cyberattacks-and>. <https://www.justice.gov/opa/pr/three-north-korean-military-hackers-indicted-wide-ranging-scheme-commit-cyberattacks-and>. Feb. 17, 2021.
- [77] 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: *Proceedings of the 29th USENIX Security Symposium*. 2020.
- [78] Josh Kamps and Bennett Kleinberg. “To the moon: defining and detecting cryptocurrency pump-and-dumps”. In: *Crime Science* 7 (Nov. 2018).
- [79] George Kappos, Haaroon Yousaf, Mary Maller, and Sarah Meiklejohn. “An Empirical Analysis of Anonymity in Zcash”. In: *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: {USENIX} Association, Aug. 2018, pp. 463–477.
- [80] George Kappos, Haaroon Yousaf, Ania M. Piotrowska, Sanket Kanjalkar, Sergi Delgado-Segura, Andrew Miller, and Sarah Meiklejohn. “An Empirical Analysis of Privacy in the Lightning Network”. In: *International Conference on Financial Cryptography and Data Security*. Springer, 2021, pp. 1–2.

- [81] Tyler Kell, Haaron Yousaf, Sarah Allen, Sarah Meiklejohn, and Ari Juels. “Forsage: Anatomy of a Smart-Contract Pyramid Scheme”. In: *arXiv preprint arXiv:2105.04380* (2021).
- [82] Jakub Kroustek. *Avast reports on WannaCryptOr 2.0 ransomware that infected NHS and Telefonica*. <https://blog.avast.com/ransomware-that-infected-telefonica-and-nhs-hospitals-is-spreading-aggressively-with-over-50000-attacks-so-far-today>. May 2017.
- [83] Amrit Kumar, Clément Fischer, Shruti Tople, and Prateek Saxena. “A Traceability Analysis of Monero’s Blockchain”. In: 2017, pp. 153–173.
- [84] Aleksandr Lazarenko and Sergey Avdoshin. “Financial risks of the blockchain industry: A survey of cyberattacks”. In: *Proceedings of the Future Technologies Conference*. Springer. 2018, pp. 368–384.
- [85] Charlie Lee. *Litecoin*. <https://litecoin.com/en/>. 2011.
- [86] Kirill Levchenko, Andreas Pitsillidis, Neha Chachra, Brandon Enright, Mark Felegyhazi, Chris Grier, Tristan Halvorson, Chris Kanich, Christian Kreibich, He Liu, Damon McCoy, Nicholas Weaver, Vern Paxson, Geoffrey M. Voelker, and Stefan Savage. “Click trajectories: End-to-end analysis of the spam value chain”. In: *Proceedings - IEEE Symposium on Security and Privacy*. 2011, pp. 431–446.
- [87] Tao Li, Donghwa Shin, and Baolian Wang. “Cryptocurrency pump-and-dump schemes”. In: *Available at SSRN 3267041* (2020).
- [88] Kim Lyons. ”*Feds arrest founder of bitcoin mixer they say laundered 335 million over ten years*”. <https://www.theverge.com/2021/4/29/22409501/feds-arrest-founder-bitcoin-mixer-laundered-cryptocurrency>. Apr. 2021.
- [89] António Madeira. *OneCoin: A deep dive into crypto’s most notorious Ponzi scheme*. <https://cointelegraph.com/news/onecoin-a-deep-dive-into-crypto-s-most-notorious-ponzi-scheme>. Sept. 2020.

- [90] Felix Konstantin Maurer, Till Neudecker, and Martin Florian. “Anonymous CoinJoin transactions with arbitrary values”. In: *2017 IEEE Trustcom/BigDataSE/ICESS*. IEEE. 2017, pp. 522–529.
- [91] Gregory Maxwell. *CoinJoin: Bitcoin privacy for the real world*. <https://bitcointalk.org/index.php?topic=279249>. Aug. 2013.
- [92] Damon McCoy, Andreas Pitsillidis, Jordan Grant, Nicholas Weaver, Christian Kreibich, Brian Krebs, Geoffrey Voelker, Stefan Savage, and Kirill Levchenko. “PharmaLeaks: Understanding the Business of Online Pharmaceutical Affiliate Programs”. In: *21st USENIX Security Symposium (USENIX Security 12)*. Bellevue, WA: USENIX Association, Aug. 2012, pp. 1–16.
- [93] Robert McMillan. *The Inside Story of Mt. Gox, Bitcoin’s \$460 Million Disaster*. <https://www.wired.com/2014/03/bitcoin-exchange/>. Mar. 2014.
- [94] Sarah Meiklejohn and Claudio Orlandi. “Privacy-Enhancing Overlays in Bitcoin”. In: 2015, pp. 127–141.
- [95] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. “A fistful of Bitcoins: Characterizing payments among men with no names”. In: *Proceedings of the Internet Measurement Conference - IMC ’13* 59.6 (2013), pp. 127–140.
- [96] Michael_S. *Why CoinJoin, as Used in DarkCoin, does NOT bring Full Anonymity*. <https://www.scribd.com/document/227369807/Bitcoin-Coinjoin-Not-Anonymous-v01>.
- [97] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. “Zerocoin: Anonymous Distributed E-Cash from Bitcoin”. In: 2013, pp. 397–411.
- [98] Andrew Miller, Malte Möser, Kevin Lee, and Arvind Narayanan. “An Empirical Analysis of Linkability in the Monero Blockchain”. In: *Proceedings on Privacy Enhancing Technologies* (2018).

- [99] *MLM law of China: 'Prohibition of Chuanxiao'*. http://www.gov.cn/zwgk/2005-09/03/content_28808.htm. 2005.
- [100] Savita Mohurle and Manisha Patil. “A brief study of wannacry threat: Ransomware attack 2017”. In: *International Journal of Advanced Research in Computer Science* 8.5 (2017), pp. 1938–1940.
- [101] *Monero*. <https://getmonero.org>.
- [102] Malte Möser and Rainer Böhme. “Anonymous Alone? Measuring Bitcoin’s Second-Generation Anonymization Techniques”. In: *IEEE Security & Privacy on the Blockchain (IEEE S&B)*. 2017.
- [103] Malte Möser, Rainer Böhme, and Dominic Breuker. “An inquiry into money laundering tools in the Bitcoin ecosystem”. In: *2013 APWG eCrime researchers summit*. Ieee. 2013, pp. 1–14.
- [104] Malte Möser and Arvind Narayanan. “Effective cryptocurrency regulation through blacklisting”. In: *Preprint* (2019).
- [105] Satoshi Nakamoto. *Bitcoin P2P e-cash paper - The Mail Archive*. <https://www.mail-archive.com/cryptography@metzdowd.com/msg09959.html>.
- [106] Satoshi Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”. In: (2008). <https://bitcoin.org/bitcoin.pdf>, p. 9.
- [107] Shen Noether. “Ring SSignature Confidential Transactions for Monero.” In: *IACR Cryptol. ePrint Arch.* 2015 (2015), p. 1098.
- [108] Micha Ober, Stefan Katzenbeisser, and Kay Hamacher. “Structure and anonymity of the bitcoin transaction graph”. In: *Future internet* 5.2 (2013), pp. 237–250.
- [109] FORSAGE Official. *Forsage Overview: Earn Ethereum Daily!* <https://www.youtube.com/watch?v=m0NzYwFfGH4>. Youtube. May 2020.
URL: <https://www.youtube.com/watch?v=m0NzYwFfGH4>.

- [110] FORSAGE Official. *FORSAGE.io - BIG SPECIAL EVENT*. <https://www.youtube.com/watch?v=NMfcDSCXLK8>. Youtube. Aug. 2020. URL: <https://www.youtube.com/watch?v=NMfcDSCXLK8>.
- [111] Mike Orcutt. *Some crypto-criminals think jumping across blockchains covers their tracks. Big mistake*. <https://www.technologyreview.com/2019/08/22/133272/some-crypto-criminals-think-jumping-across-blockchains-covers-their-tracks-big-mistake/>. 2019.
- [112] Daniel Palmer. *Chinese Authorities Have Seized a Massive \$4B in Crypto From PlusToken Scam*. <https://www.coindesk.com/chinese-authorities-have-seized-a-massive-4-billion-in-crypto-from-plustoken-scam>. Nov. 2020.
- [113] Masarah Paquet-Clouston, Bernhard Haslhofer, and Benoit Dupont. “Ransomware payments in the bitcoin ecosystem”. In: *Journal of Cybersecurity* 5.1 (2019), tyz003.
- [114] Masarah Paquet-Clouston, Matteo Romiti, Bernhard Haslhofer, and Thomas Charvat. “Spams Meet Cryptocurrencies: Sextortion in the Bitcoin Ecosystem”. In: *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*. AFT ’19. Zurich, Switzerland: Association for Computing Machinery, 2019, pp. 76–88.
- [115] Morgen E. Peck. “The cryptoanarchists’ answer to cash”. In: *IEEE Spectrum* 49.6 (2012), pp. 51–56.
- [116] Ross Phillips and Heidi Wilder. “Tracing Cryptocurrency Scams: Clustering Replicated Advance-Fee and Phishing Websites”. In: *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (2020).
- [117] Joseph Poon and Thaddeus Dryja. *The bitcoin lightning network: Scalable off-chain instant payments*. 2016.
- [118] Rebecca S Portnoff, Danny Yuxing Huang, Periwinkle Doerfler, Sadia Afroz, and Damon McCoy. “Backpage and {Bitcoin}: uncovering human traffickers”. In: *Proceedings of the ACM SIGKDD Conference*. 2017.

- [119] Jeffrey Quesnelle. *On the linkability of Zcash transactions*. arXiv:1712.01210. <https://arxiv.org/pdf/1712.01210.pdf>. 2017.
- [120] Pierre Reibel, Haaroon Yousaf, and Sarah Meiklejohn. “Short paper: An exploration of code diversity in the cryptocurrency landscape”. In: *International Conference on Financial Cryptography and Data Security*. Springer. 2019, pp. 73–83.
- [121] Fergal Reid and Martin Harrigan. “An Analysis of Anonymity in the Bitcoin System”. In: *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. 2011, pp. 1318–1326.
- [122] 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.
- [123] Tom Robinson. *How Iran Uses Bitcoin Mining to Evade Sanctions and "Export" Millions of Barrels of Oil*. <https://www.elliptic.co/blog/how-iran-uses-bitcoin-mining-to-evade-sanctions>. May 2021.
- [124] Dorit Ron and Adi Shamir. “Quantitative Analysis of the Full Bitcoin Transaction Graph”. In: *Financial Cryptography and Data Security*. Ed. by Ahmad-Reza Sadeghi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 6–24.
- [125] Keith W Ross and Danny HK Tsang. “The stochastic knapsack problem”. In: *IEEE Transactions on communications* 37.7 (1989), pp. 740–747.
- [126] Todd Rowland and Eric W. Weisstein. *Connected Component*.
- [127] Dominic Rushe. *Cryptocurrency investors locked out of \$190m after exchange founder dies*. <https://www.theguardian.com/technology/2019/feb/04/quadrigacx-canada-cryptocurrency-exchange-locked-gerald-cotten>. Feb. 2019.
- [128] Nicolas van Saberhagen. *CryptoNote v 2.0*. <https://bytecoin.org/old/whitepaper.pdf>. 2013.

- [129] *Samourai Wallet - Home.* <https://samouraiwallet.com/>.
- [130] Aaron Sankin. *Sheep marketplace scam reveals everything that's wrong with the deep web.* <https://newstaging.dailydot.com/crime/sheep-marketplace-scam-shut-down/>. 2013.
- [131] Mohsen Sayyadiharikandeh, Onur Varol, Kai-Cheng Yang, Alessandro Flammini, and Filippo Menczer. “Detection of Novel Social Bots by Ensembles of Specialized Classifiers”. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (Oct. 2020).
- [132] Justin Scheck and Shane Shifflett. *How Dirty Money Disappears Into the Black Hole of Cryptocurrency.* <https://www.wsj.com/articles/how-dirty-money-disappears-into-the-black-hole-of-cryptocurrency-1538149743>. 2018.
- [133] SEC ISSUES CEASE AND DESIST ORDER AGAINST FORSAGE. <https://www.sec.gov/pr-2020/sec-issues-cease-and-desist-order-against-forsage/>. Sept. 2020.
- [134] SEC WARNS AGAINST FORSAGE, OTHER SCHEMES. <https://www.sec.gov/pr-2020/sec-warns-against-forsage-other-schemes/>. July 2020.
- [135] Kyle Soska and Nicolas Christin. “Measuring the Longitudinal Evolution of the Online Anonymous Marketplace Ecosystem”. In: *24th USENIX Security Symposium (USENIX Security 15)* (2015), pp. 33–48.
- [136] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. “BitIodine: Extracting Intelligence from the Bitcoin Network”. In: 2014, pp. 457–468.
- [137] Brett Stone-Gross, Ryan Abman, Richard A Kemmerer, Christopher Kruegel, Douglas G Steigerwald, and Giovanni Vigna. “The underground economy of fake antivirus software”. In: *Economics of Information Security and Privacy III*. Springer, 2013, pp. 55–78.

- [138] Nick Szabo. *Bit gold*. <https://unenumerated.blogspot.com/2005/12/bit-gold.html>. 2005.
- [139] *The Shadow Brokers*. <https://steemit.com/@theshadowbrokers>.
- [140] Rob Thomas and Jerry Martin. *The Underground Economy: Priceless*. 2006.
- [141] Christof Ferreira Torres, Mathis Steichen, and Radu State. “The Art of The Scam: Demystifying Honeypots in Ethereum Smart Contracts”. In: *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 1591–1607.
- [142] Lawrence J Trautman. “Virtual currencies; Bitcoin & what now after Liberty Reserve, Silk Road, and Mt. Gox?” In: *Richmond Journal of Law and Technology* 20.4 (2014).
- [143] C N Trueman. *The Black Market*. <http://www.historylearningsite.co.uk/world-war-two/world-war-two-in-western-europe/britains-home-front-in-world-war-two/the-black-market/>. 2015.
- [144] Rolf Van Wegberg, Jan-Jaap Oerlemans, and Oskar van Deventer. “Bitcoin money laundering: mixed results? An explorative study on money laundering of cybercrime proceeds using bitcoin”. In: *Journal of Financial Crime* (2018).
- [145] Onur Varol, Emilio Ferrara, Clayton Davis, Filippo Menczer, and Alessandro Flammini. “Online human-bot interactions: Detection, estimation, and characterization”. In: *Proceedings of the international AAAI conference on web and social media*. Vol. 11. 1. 2017.
- [146] Marie Vasek and T. Moore. “There’s No Free Lunch, Even Using Bitcoin: Tracking the Popularity and Profits of Virtual Currency Scams”. In: *Financial Cryptography*. 2015.
- [147] Erik Voorhees. *Announcing ShapeShift Membership*. <https://shapeshift.io/blog/2018/09/04/introducing-shapeshift-membership/>. Sept. 2018.

- [148] *Wasabi Wallet - Bitcoin privacy wallet with built-in CoinJoin.* <https://wasabiwallet.io/>.
- [149] *What is Jubjub?* <https://z.cash/technology/jubjub.html>.
- [150] Zooko Wilcox. *Maintaining Privacy.* <https://electriccoin.coindesk.com/zcash-privacy-weakened-by-certain-behaviors-researchers-say/>. 2018.
- [151] Gavin Wood. “Ethereum: A Secure Decentralized Generalized Transaction Ledger”. In: (Dec. 2020). <https://ethereum.github.io/yellowpaper/paper.pdf>.
- [152] Pieter Wuille, Jonas Nick, and Anthony Towns. *Taproot.* <https://github.com/bitcoin/bips/blob/master/bip-0341.mediawiki>. Jan. 2020.
- [153] Jiahua Xu and Benjamin Livshits. “The Anatomy of a Cryptocurrency Pump-and-Dump Scheme”. In: *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 1609–1625.
- [154] Sophie Knight Yoshifumi Takemoto. *Mt. Gox files for bankruptcy, hit with lawsuit.* <https://www.reuters.com/article/us-bitcoin-mtgox-bankruptcy-mt-gox-files-for-bankruptcy-hit-with-lawsuit-idUSBREA1R0FX20140228>.
- [155] Haaroon Yousaf, George Kappos, and Sarah Meiklejohn. “Tracing Transactions Across Cryptocurrency Ledgers”. In: *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 837–850.
- [156] *Zcash.* <https://z.cash>.
- [157] *Zcash. Privacy Recommendations and Best Practices.* <https://z.cash/support/security/privacy-security-recommendations/>. 2018.
- [158] *Zcash chain parameters.* <https://github.com/zcash/zcash/blob/v1.0.0/src/chainparams.cpp#L135-L192>.

- [159] *Zcash FAQs*. <https://z.cash/support/faq.html>.
- [160] *Zchain explorer*. <http://explorer.zcha.in/>.
- [161] *Zecwallet Lite*. <https://www.zecwallet.co/>.
- [162] Xiangfu Zhao, Zhongyu Chen, Xin Chen, Yanxia Wang, and Changbing Tang. “The DAO attack paradoxes in propositional logic”. In: Nov. 2017, pp. 1743–1746.
- [163] Jianwei Zhuge, Thorsten Holz, Chengyu Song, Jinpeng Guo, Xinhui Han, and Wei Zou. “Studying malicious websites and the underground economy on the Chinese web”. In: *Managing Information Risk and the Economics of Security*. Springer, 2009, pp. 225–244.