

The Unreasonable Effectiveness of Address Clustering

Martin Harrigan^{*1} and Christoph Fretter²

¹Waterford Institute of Technology

²Elliptic Enterprises Limited, London

Abstract

Address clustering tries to construct the one-to-many mapping from entities to addresses in the Bitcoin system. Simple heuristics based on the micro-structure of transactions have proved very effective in practice. In this paper we describe the primary reasons behind this effectiveness: address reuse, avoidable merging, super-clusters with high centrality, and the incremental growth of address clusters. We quantify their impact during Bitcoin’s first seven years of existence.

1 Introduction

Bitcoin is a double-edged sword for financial privacy. It allows anyone to conduct financial transactions with anyone else in the world without having to divulge identifying information to intermediaries. However, it requires those transactions to be broadcast to the world. The contents of the transactions, their relationship with other transactions, and the very act of broadcasting the transactions themselves may unintentionally disclose information about the transactors to interested third parties. In fact, many interested third parties systematically gather and analyze this information for reasons such as market research, competitor analysis, compliance, and law enforcement.

Address clustering is a cornerstone of this analysis. It partitions the set of addresses observed in Bitcoin transactions into maximal subsets of addresses that are likely controlled by the same entity. Each subset in the partition is an address cluster. When combined with address tagging (associating real-world identities with addresses) and graph analysis, it is an effective means of analysing Bitcoin activity at both the micro- and macro-levels, see, e.g., [1, 3, 6, 9, 16, 17, 20, 23]. Experimental analysis has shown that a single heuristic (the multi-input heuristic) can identify more than 69% of the addresses in the wallets stored by lightweight clients.

^{*}Email: martinharrigan@gmail.com

sidered the impact of address clusters on anonymity. This approach can be augmented with change heuristics [1,9,23], temporal behavior [10,17] and transaction fingerprinting [3]. Although the analyses in the present paper are based on the multi-input heuristic only, they can be extended to any combination of heuristics.

Nick [15] analyzed the performance of several clustering heuristics by exploiting a vulnerability in connection Bloom filtering used by lightweight clients. He found that the multi-input heuristic can identify more than 69% of the addresses in the vulnerable wallets.

Ober et al. [16] studied the sizes and lifespans of address clusters and showed that the sizes of the address clusters follow a scale-free distribution. Lischke and Fabian [6] showed that major darknet markets, gambling services, exchanges and mining pools were major hubs in the address cluster graph (similar to Fig. 1 but not limited to the largest address clusters) during Bitcoin’s first four years of existence.

Maxwell described CoinJoin [7], a protocol for trustless centralized Bitcoin mixing. This causes the multi-input heuristic to produce false positives. CoinJoin is a centralized mixing protocol; it requires a third-party or CoinJoin server to operate. Other protocols in this category include Mixcoin [2] and Blindcoin [25]. Decentralized mixing protocols do not require any third-party, trusted or trustless. Protocols in this category include CoinSwap [8], CoinShuffle [21] and CoinParty [26]. Shentu and Yu [22] review several trustless Bitcoin protocols.

Möser et al. [11,12] considered the implications of blockchain analyses, including address clustering, for anti-money laundering. Imwinkelreid [5] discussed its implications for digital forensics.

3 Counting Address Clusters

The following analyses were performed when the block at the tip of the Bitcoin blockchain was at height 396577 and the last eight hexadecimal digits of the block hash were 900a6f4c.

Figure 2 compares the monthly counts of transactions (red line) with the monthly counts of new addresses (blue line). The number of new addresses has grown in line with the number of transactions. The monthly counts of address clusters (green line) with at least two addresses has grown at a much slower rate.

We consider the relationship between these counts in Fig. 3. We plot the number of new addresses observed per transaction (purple line) and the number of newly merged address clusters created per transaction (orange line). To adjust for the rapid growth in transactions in Bitcoin’s recent history, we replace the horizontal time axis with ordinal transaction numbers: this compresses low-activity periods and expands high-activity periods. We observe that both ratios have been relatively stable for the past two years and that the former is an order of magnitude larger than the latter.

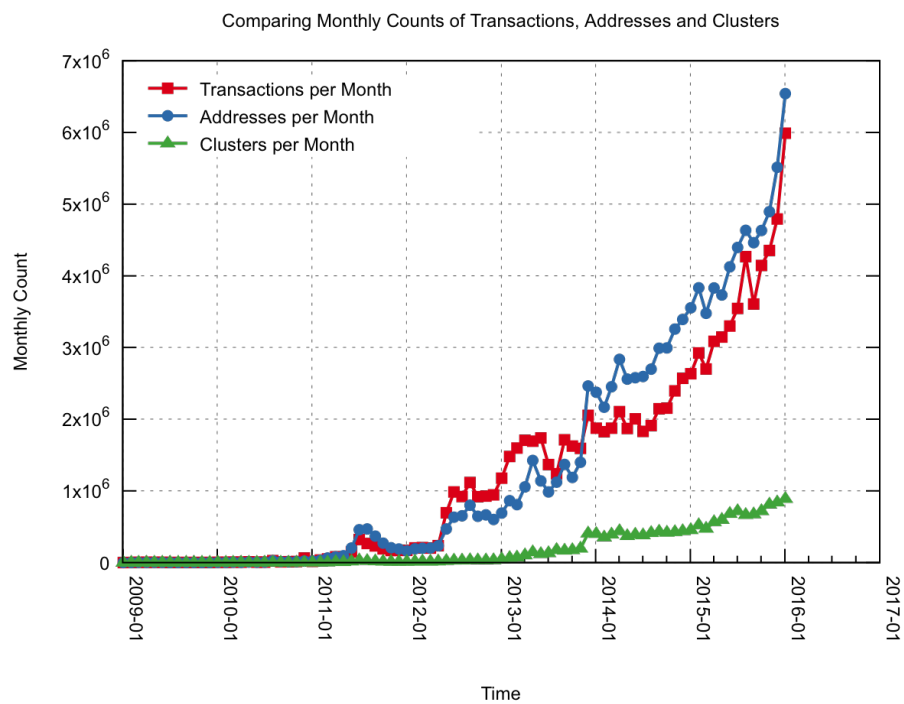


Figure 2: A plot of the monthly counts of transactions (red line), new addresses (blue line) and address clusters (green line) with at least two addresses. For the past two years, the monthly number of new addresses is greater than the monthly number of transactions.

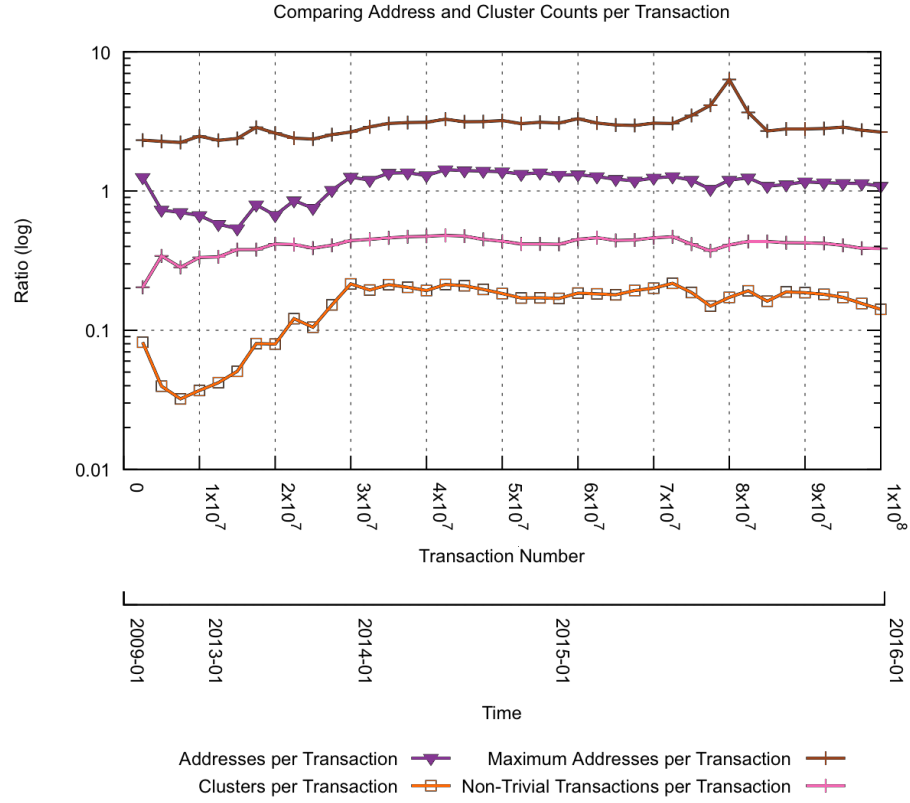


Figure 3: A plot of the ratios of new addresses per transaction (purple line) and newly merged address clusters per transaction (orange line). The maximum addresses per transaction (brown line) and non-trivial transactions per transaction (pink line) are respective upper bounds.

Can we establish upper bounds for the two ratios? Nakamoto [13] suggested that “a new key pair should be used for each transaction to keep them from being linked to a common owner.” This is from the perspective of the payee(s) only; if the payer requires additional transaction outputs, say, for change, they should also use a new address. For transaction outputs that contain Pay-to-PubKey and Pay-to-PubKey-Hash scripts, the number of transaction outputs per transaction is an upper bound for the number of new addresses per transaction. This can be adjusted for transaction outputs that contain `OP_RETURN` scripts, multi-signature scripts and Pay-to-Script-Hash scripts where the redemption script is known (brown line). The gap between the brown and purple lines is a measure of the level of address reuse; the wider the gap the greater the level of address reuse.

Similarly, the fraction of transactions that spend at least two transaction outputs assigned to different addresses (pink line) is an upper bound for the number of newly merged address clusters per transaction. We refer to these transactions as being non-trivial. If every transaction output created a new address then every non-trivial transaction would create a newly merged address cluster. The gap between the pink and orange lines is a measure of the level of cluster merging; the wider the gap the greater the level of cluster merging. Even in the presence of address reuse, this gap could be narrowed through the use of merge avoidance [4, 18].

The existence of both gaps is significant. New key pairs are not being generated for every transaction allowing the multi-input heuristic to link addresses to a common owner. This is one reason that address clustering is unreasonably effective. There is considerable reuse of addresses and merging of address clusters. We will discuss a second reason in the next section.

4 Measuring Cluster Sizes

The address clusters with at least two addresses are binned by size in Fig. 4. Both the horizontal and the vertical axes use logarithmic scales. We observe the presence of “super-clusters”: there are 1955 address clusters with at least 1000 addresses but less than 10 million addresses. They cover 22% of all of the addresses represented in Fig. 4 and 16% of all of the addresses observed at the time of the analysis.

We exclude the single address cluster with greater than 10 million addresses. This address cluster originally belonged to the Mt. Gox exchange that, for a time, allowed users to import private-keys directly from their wallets. This feature causes the multi-input heuristic to produce false positives and requires more advanced heuristics to separate the Mt. Gox addresses. We will discuss this issue in Sect. 5.

The super-clusters are not only large in terms of the number of addresses they contain, they are also hubs in terms of the number of transactions they are involved in. At the time of the analysis, the 107 million transactions created 319 million transaction outputs and redeemed 285 million of those through

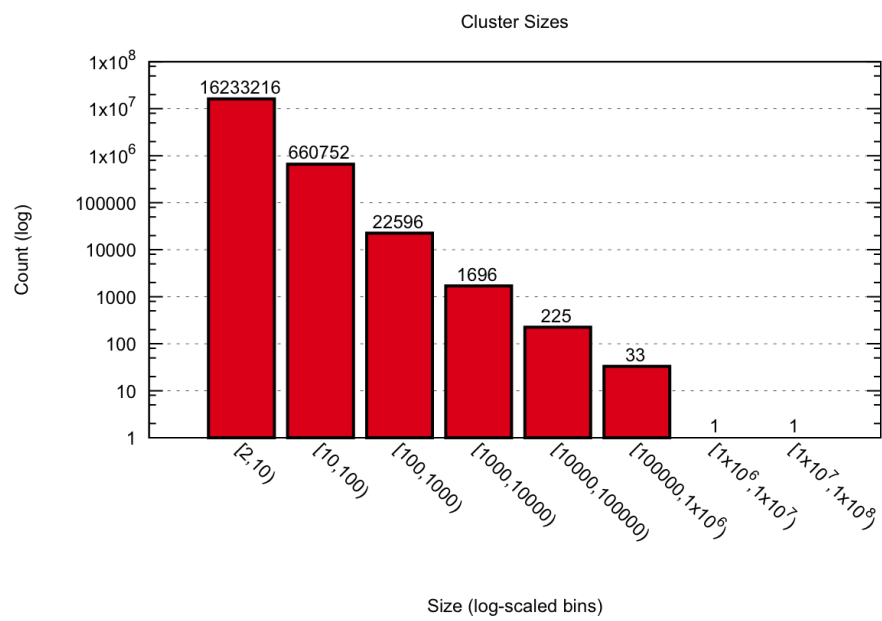


Figure 4: A histogram showing the number of address clusters with at least two addresses in each size range.

transaction inputs. Of those, the super-clusters were responsible for 72 million or 23% of the transaction outputs and 51 million or 18% of the transaction inputs. If we can link identities to the super-clusters then we can identify at least one of the transactors in a considerable number of transactions.

Lischke and Fabian [6] made a related finding—they analyzed the degree centrality of the vertices in a network similar to the one in Fig. 1 but not limited to the largest address clusters, and found that the vertices representing the major darknet markets, gambling services, exchanges and mining pools had the highest degree centralities.

The existence and centrality of super-clusters is another reason that address clustering is unreasonably effective. Many of the major services reuse addresses and generate super-clusters thereby identifying much of their on-chain activity. Furthermore, this identifies much of the activity between the service and their users: deposits and withdrawals can be easily identified. This can be exploited to produce “wallet explorers” such as [WalletExplorer.com](https://walletexplorer.com). It also makes the services vulnerable to re-identification attacks [9].

Major services can avoid creating super-clusters. For example, Coinbase, the Bitcoin exchange and wallet provider, does not create a super-cluster that identifies all activity between the service and their users. They are notably absent from many high-level blockchain analyses. This is not to say that they do not create any large clusters. It simply means that the multi-input heuristic alone is insufficient for identifying all of their on-chain activity.

5 Formation and Structure

The address clustering heuristics listed in Sect. 2 cause address clusters to merge. We are not aware of any published heuristics that cause address clusters to split, e.g., to counter the mixing protocols in Sect. 2 or to partition the Mt. Gox address cluster in Sect. 4. When address clusters merge, we can measure the increases in size of the newly merged cluster. For example, suppose a transaction causes four address clusters of sizes 1, 1, 2 and 10 to merge. This can be represented by increases of 1, 1 and 2. Considering the multi-input heuristic only, the distribution of these increases is heavily concentrated around a median value of one. Figure 5 plots the 99th percentile, 999th permille, 9999th 10 000-quantile and 99 999th 100 000-quantile for every 250 000 transactions. We observe that large increases in address cluster sizes are rare. The multi-input heuristic usually merges at most one large address cluster with one or more small address clusters, but rarely merges two or more large address clusters.

This behaviour can be visualised as follows. Consider a bipartite graph for each address cluster generated using the multi-input heuristic where each vertex represents either an address (an address vertex) or a transaction (a transaction vertex) and each edge between an address vertex and a transaction vertex represents the transaction spending a transaction output that was controlled by the address. Figure 6 is the bipartite graph for a typical address cluster¹. The

¹The address cluster contains the address `1H7RNFmAbtMgVJzK72hNFerGBfKuRekTMU`: it re-

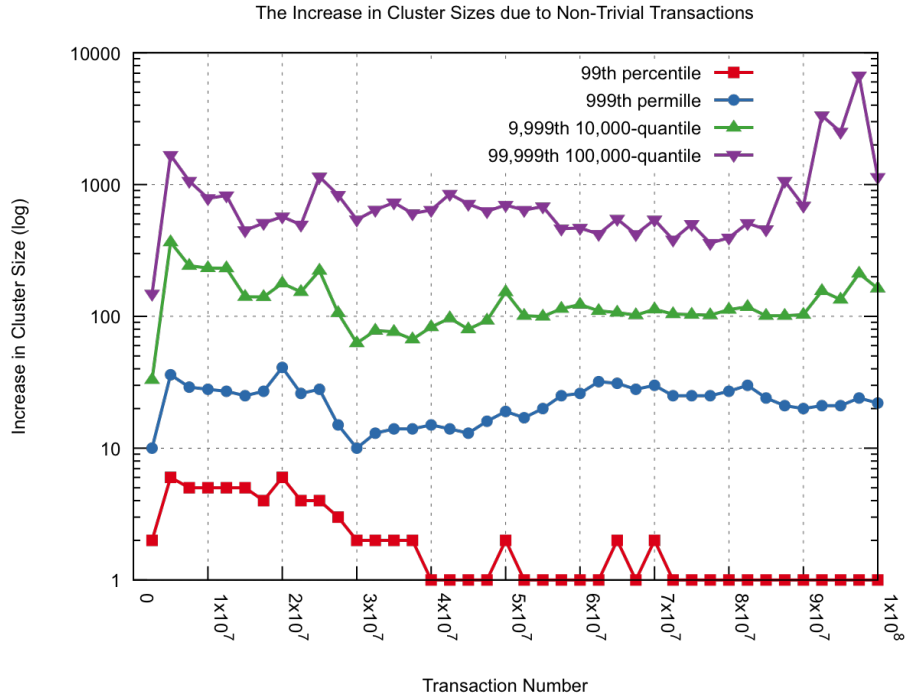


Figure 5: A plot of the $q - 1$ th q -quantiles for $q = 100, 1000, 10000, 100000$ of the distributions of the increases in cluster size due to merging for every 250 000 transactions. The increases are heavily concentrated around median values of one. For the past 30 million transactions, the 99th percentiles are also at one.

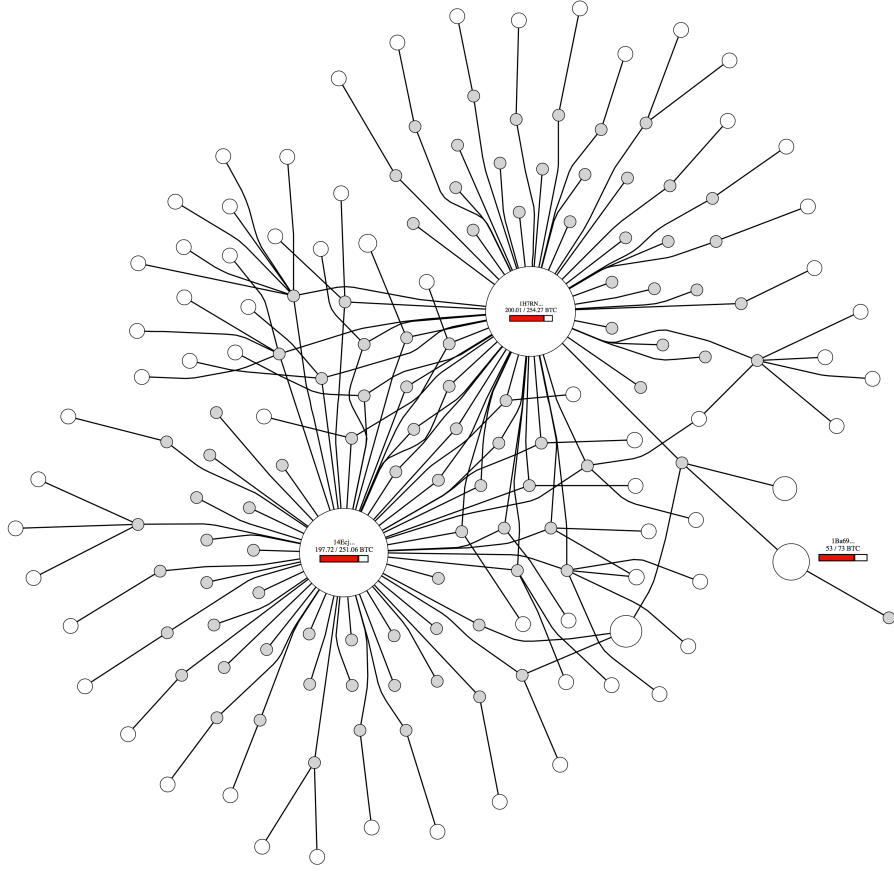


Figure 6: A bipartite graph representing the structure of an address cluster: white vertices are addresses; gray vertices are transactions; edges connect transaction vertices to address vertices when the corresponding transaction spends transaction outputs that were assigned to the corresponding addresses.

white vertices are the address vertices. The address vertices that correspond to addresses with non-zero balances are annotated with their current and all-time maximum balances. The majority of addresses have zero balances. The gray vertices are the transactions—they connect together the addresses to form the address cluster. Address vertices that are connected through multiple independent paths have multiple independent sets of transactions indicating that they are part of the same address cluster.

This graph was formed by small address clusters (the singleton address vertices along the periphery) merging with the large address cluster, through transaction vertices that connected the singleton address vertices to address vertices with non-zero balances. It is rare for such a graph to form as two large disconnected components, each containing at least one address vertex with a non-zero balance, and then to merge into a single connected component.

Address clusters that form when two large address clusters merge can be flagged as exhibiting unusual merging activity. This can be extended to a heuristic for splitting address clusters that may not be controlled by the same entity. For example, if we identify the 0.01% of transactions that resulted in the largest increases in cluster size during the lifetime of the Mt. Gox exchange (July 2010 to February 2014), then the majority of those transactions spend transaction outputs that were controlled by the Mt. Gox address cluster. This is likely due to their private-key import feature.

The incremental growth of address clusters is beneficial for many high-level blockchain analyses. The address clustering is relatively stable over time. It is a rarity for two large address clusters to merge, thereby drastically changing the results of an earlier analysis. In fact, if two large address clusters do merge, it may indicate that the multi-input heuristic has produced a false positive. Furthermore, the address clustering is suitable for real-time analyses. Small address clusters merge with large address clusters early in their lifetime and the large address clusters are more likely to have identifying information associated with them.

6 Conclusion and Future Work

We have enumerated and analyzed the primary reasons behind the effectiveness of address clustering using Bitcoin’s blockchain. These are the high-levels of address reuse and avoidable merging; the existence of super-clusters with high centrality, and the incremental growth of address clusters.

The results can inform and help blockchain analysts. For example, the super-clusters are primary targets for re-identification attacks. The technique at the end of Sect. 5 can flag address clusters that may include addresses from more than one entity.

The opposing camp, those seeking to hinder blockchain analysis, can also benefit from these results. For example, the adoption and impact of privacy-

ceived bitcoins from a mining pool (DeepBit), sent bitcoins to exchanges (Mt. Gox and `bitcoin.de`) and purchased goods through a Bitcoin payment processor (BitPay).

enhancing techniques such as merge avoidance and Elliptic Curve Diffie-Hellman-Merkle (ECDHM) address schemes, e.g., stealth addresses [24], reusable payment codes (BIP47) [19] and out of band address exchange (BIP75) [14], can be measured indirectly through the gap between the number of non-trivial transactions and the number of address clusters created or merged per transaction (see Sect. 3).

Our future work revolves around the internal structure of address clusters, à la the bipartite graph in Fig. 6. This representation shows the structure of an address cluster beyond a simple set of addresses and may provide further insight into its formation and behavior.

References

- [1] Elli Androulaki, Ghassan Karame, Marc Roeschlin, Tobias Scherer, and Srdjan Capkun. Evaluating User Privacy in Bitcoin. In *Proceedings of the 17th International Conference on Financial Cryptography and Data Security, FC 2013, Okinawa, Japan, April 1–5, 2013*, pages 34–51, 2013.
- [2] Joseph Bonneau, Arvind Narayanan, Andrew Miller, Jeremy Clark, Joshua Kroll, and Edward Felten. Mixcoin: Anonymity for Bitcoin with Accountable Mixes. In *Proceedings of the 18th International Conference on Financial Cryptography and Data Security, FC 2014, Christ Church, Barbados, March 3–7, 2014*, pages 486–504, 2014.
- [3] Michael Fleder, Michael S. Kester, and Sudeep Pillai. Bitcoin Transaction Graph Analysis. *CoRR*, abs/1502.01657, 2015.
- [4] Mike Hearn. Merge Avoidance. <https://medium.com/p/7f95a386692f>. Accessed: 2016-03-01.
- [5] Edward Imwinkelried and Jason Luu. The Challenge of Bitcoin Pseudo-Anonymity to Computer Forensics. 2015.
- [6] Matthias Lischke and Benjamin Fabian. Analyzing the Bitcoin Network: The First Four Years. *Future Internet*, 8(1):7, 2016.
- [7] Gregory Maxwell. CoinJoin: Bitcoin Privacy for the Real World. <https://bitcointalk.org/index.php?topic=279249>. Accessed: 2016-03-01.
- [8] Gregory Maxwell. CoinSwap: Transaction Graph Disjoint Trustless Trading. <https://bitcointalk.org/index.php?topic=321228>. Accessed: 2016-04-01.
- [9] 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 2013 Internet Measurement Conference IMC 2013, Barcelona, Spain, October 23–25, 2013*, pages 127–140, 2013.

- [10] John Monaco. Identifying Bitcoin Users by Transaction Behavior. 2015.
- [11] Malte Möser, Rainer Böhme, and Dominic Breuker. An Inquiry into Money Laundering Tools in the Bitcoin Ecosystem. In *Proceedings of the APWG eCrime Researchers Summit, ECRIME 2013*, San Francisco, USA, 2013.
- [12] Malte Möser, Rainer Böhme, and Dominic Breuker. Towards Risk Scoring of Bitcoin Transactions. In *Proceedings of the First Workshop on Bitcoin Research in Association with Financial Crypto 2014*, pages 1–16, Barbados, 2014.
- [13] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008.
- [14] Justin Newton, Matt David, Aaron Voisine, and James MacWhyte. Out of Band Address Exchange using Payment Protocol Encryption. <https://github.com/bitcoin/bips/blob/master/bip-0075.mediawiki>. Accessed: 2016-04-01.
- [15] Jonas Nick. Data-Driven De-Anonymization in Bitcoin. Master’s thesis, ETH Zürich, 8 2015.
- [16] Micha Ober, Stefan Katzenbeisser, and Kay Hamacher. Structure and Anonymity of the Bitcoin Transaction Graph. *Future Internet*, 5(2):237–250, 2013.
- [17] Marc Ortega. The Bitcoin Transaction Graph—Anonymity. Master’s thesis, Universitat Oberta de Catalunya, 6 2013.
- [18] Justus Ranvier. Reclaiming Financial Privacy with HD Wallets. <http://bitcoinism.blogspot.ie/2013/07/reclaiming-financial-privacy-with-hd.html>. Accessed: 2016-03-01.
- [19] Justus Ranvier. Reusable Payment Codes for Hierarchical Deterministic Wallets. <https://github.com/bitcoin/bips/blob/master/bip-0047.mediawiki>. Accessed: 2016-04-01.
- [20] Fergal Reid and Martin Harrigan. An Analysis of Anonymity in the Bitcoin System. In Yaniv Altshuler, Yuval Elovici, Armin Cremers, Nadav Aharony, and Alex Pentland, editors, *Security and Privacy in Social Networks*, pages 197–223. Springer New York, 2013.
- [21] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin. In *Proceedings of the 19th European Symposium on Research in Computer Security, Wroclaw, Poland, September 7–11, 2014*, pages 345–364, 2014.
- [22] QingChun ShenTu and Jianping Yu. Research on Anonymization and De-anonymization in the Bitcoin System. *CoRR*, abs/1510.07782, 2015.

- [23] Michele Spagnuolo, Federico Maggi, and Stefano Zanero. BitIodine: Extracting Intelligence from the Bitcoin Network. In *Proceedings of the 18th International Conference on Financial Cryptography and Data Security, FC 2014, Christ Church, Barbados, March 3–7, 2014*, pages 457–468, 2014.
- [24] Peter Todd. Stealth Addresses. <https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2014-January/004020.html>. Accessed: 2016-04-01.
- [25] Luke Valenta and Brendan Rowan. Blindcoin: Blinded, Accountable Mixes for Bitcoin. In *Proceedings of the 19th International Conference on Financial Cryptography and Data Security, FC 2015, San Juan, Puerto Rico, January 30, 2015*, pages 112–126, 2015.
- [26] Jan Henrik Ziegeldorf, Fred Grossmann, Martin Henze, Nicolas Inden, and Klaus Wehrle. CoinParty: Secure Multi-Party Mixing of Bitcoins. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, CODASPY 2015, San Antonio, TX, USA, March 2–4, 2015*, pages 75–86, 2015.