Short Paper: An Exploration of Code Diversity in the Cryptocurrency Landscape

Pierre Reibel[1], Haaroon Yousaf[1], and Sarah Meiklejohn[1]

University College London
{pierre.reibel.16,h.yousaf,s.meiklejohn}@ucl.ac.uk

Abstract. Interest in cryptocurrencies has skyrocketed since their introduction a decade ago, with hundreds of billions of dollars now invested across a landscape of thousands of different cryptocurrencies. While there is significant diversity, there is also a significant number of scams as people seek to exploit the current popularity. In this paper, we seek to identify the extent of innovation in the cryptocurrency landscape using the open-source repositories associated with each one. Among other findings, we observe that while many cryptocurrencies are largely unchanged copies of Bitcoin, the use of Ethereum as a platform has enabled the deployment of cryptocurrencies with more diverse functionalities.

1 Introduction

Since the introduction of Bitcoin in 2008 [23] and its deployment in January 2009, cryptocurrencies have become increasingly popular and subject to increasing amounts of hype and speculation. Initially, the promise behind cryptocurrencies like Bitcoin was the ability to send frictionless global payments: anyone in the world could act as a peer in Bitcoin's peer-to-peer network and broadcast a transaction that — without having to pay exorbitant fees — would send money to anyone else in the world, regardless of their location, citizenship, or what bank they used. This is achieved by the decentralization inherent in the open consensus protocol, known as proof-of-work, that allows any peer to not only broadcast transactions but also act to seal them into the official ledger.

While the realities of Bitcoin have shifted in the ensuing years, the landscape of cryptocurrencies has also shifted considerably. There are now thousands of alternative cryptocurrencies, supporting more exotic functionalities than the simple atomic transfer of money supported by Bitcoin. Ethereum, for example, promises to act as a distributed consensus computer (the Ethereum Virtual Machine, or EVM for short) by enabling arbitrary stateful programs to be executed by transactions, while Monero and Zcash promise to improve on the anonymity achieved by Bitcoin transactions. Others don't promise new functionalities but instead aim to support the same functionality as Bitcoin in more cost-effective ways; e.g., Zilliqa [16, 9, 28, 19, 17] and Cardano [15, 7] incorporate respective ideas from the academic literature about achieving consensus without relying entirely on proof-of-work.

Alongside this rapid expansion in the functionality of cryptocurrencies (or indeed the general applicability of the underlying concept of a blockchain), there has also been a genuine explosion of investment into these technologies. In July 2013, for example, there were 42 cryptocurrencies listed on the popular data tracker CoinMarketCap,[1] and the collective market capitalization was just over 1 billion USD. In July 2018, in contrast, there were 1664 cryptocurrencies, and the collective market capitalization was close to 1 trillion USD. While comprehensive in terms of deployed cryptocurrencies, this list does not even include many of the recent "initial coin offerings" (ICOs) that have similarly attracted millions in investment despite there having been many documented scams.[2][3] Against this backdrop of hype and investment, it is thus crucial to gain some insight into the different types of functionalities offered by these many different cryptocurrencies, to understand which coins offer truly novel features and are backed by genuine development efforts, and which ones are merely hoping to cash in on the hype.

This paper takes a first step in this direction, by examining the entire landscape of cryptocurrencies in terms of the publicly available source code used to support each one. While source code may not be the most accurate representation of a cryptocurrency (as, for example, the actual client may use a different codebase), it does reflect the best practices of the open-source software community, so we believe it to be a reasonable proxy for how a cryptocurrency does (or should) represent itself.

2 Related Work

We treat as related research that measures either general properties of open-source software, or research that measures properties of cryptocurrencies. In terms of the former, there have been numerous papers measuring GitHub repositories. For example, Hu et al. [12] and Thung et al. [29] measured the influence of software projects according to their position of their repositories and developers in the GitHub social graph, and others have taken advantage of the volume of source code available on GitHub to analyze common coding practices [34] or how bugs vary across different programming languages [24].

In terms of the latter, there are by now many papers that have focused on measuring properties of both the peer-to-peer networks [18, 8, 4, 1] and the blockchain data associated with cryptocurrencies [25, 26, 20, 27, 22, 14, 5, 30, 6, 3], as well as their broader ecosystem of participants [21, 33, 31, 32]. Given the volume of research, we focus only on those papers most related to our own, in that they analyze properties across multiple cryptocurrencies, rather than within a single one like Bitcoin. In terms of comparing Bitcoin and Ethereum, Gencer et al. [10] compared the level of decentralization in their peer-to-peer networks and found, for example, that Ethereum mining was more centralized than it was in Bitcoin, but that Bitcoin nodes formed more geographic clusters. Azouvi et

[1] `https://coinmarketcap.com/historical/20130721/`
[2] `https://deadcoins.com/`
[3] `https://magoo.github.io/Blockchain-Graveyard/`

al. [2] also compared their level of decentralization, in terms of the discussions on and contributions to their GitHub repositories, and found that Ethereum was more centralized in terms of code contribution and both were fairly centralized in terms of the discussions. Gervais et al. [11] introduced a framework for identifying the tradeoff between security and performance in any cryptocurrency based on proof-of-work, and found that the same level of resilience to double-spending attacks was achieved by 37 blocks in Ethereum as by 6 blocks in Bitcoin. Finally, Huang et al. [13] compared the effectiveness of different mining and speculation activities for 18 cryptocurrencies, and found that the profitability of both was affected by when a cryptocurrency was listed on an exchange.

3 Data Collection

In order to collect the source code associated with each cryptocurrency, we started with the list maintained at CoinMarketCap, which is generally regarded as one of the most comprehensive resources for cryptocurrency market data. The site maintains not only market data for each cryptocurrency (its market capitalization, price, circulating supply, etc.), however, but also links to any websites, blockchain explorers, or — crucially for us — source code repositories. We last scraped the site on July 24 2018, at which point there were 1664 cryptocurrencies listed, with a cumulative market capitalization of 293B USD.

3.1 Source code repositories

Of the listed cryptocurrencies, 1123 had a link available on CoinMarketCap to some source code repository. We examined a random sample of 10% of these links (and all the links for the top 20 cryptocurrencies) to ensure that they were legitimate, and in some cases replaced links where the information was inaccurate (for Bitcoin Cash, for example, the provided link was for the repositories backing `bitcoincash.org` rather than the actual software code). Of these links, 1108 (98.7%) pointed to GitHub.

As should be expected, many of the cryptocurrencies had multiple software repositories available; indeed, the links provided on CoinMarketCap were to the lists of repositories for a given GitHub organization, and in total there were 13,694 individual repositories available. The vast majority of these repositories had been created after October 2014, with a notable rise in frequency starting in April 2017. These repositories typically fell into one of three categories: (1) integral to the cryptocurrency itself, such as implementations of the reference client or supporting libraries; (2) irrelevant, such as a different project by the same organization; or (3) unchanged forks or mirrors of popular software projects, such as llvm. Given our goal of differentiating between the various cryptocurrencies, we did not want to clone every available repository but instead sought to isolate the first category of "meaningful" code.

To do this, we assigned a rating to each repository for a given cryptocurrency according to: (1) the gap between its last update and the current date, to capture activity (where this was subtracted from the rating, as a longer gap indicates less activity); (2) its number of forks, to capture popularity and reuse; and (3) information about the name of the repository, to capture relevance. (For example, repositories with names including 'website' were excluded and ones with names including 'core' or 'token' were given a higher rating.) For each cryptocurrency, we then cloned the top 20% of the list of repositories, sorted from high to low by these ratings (or cloned one repository, whichever was larger). We then manually examined the repositories (both selected and unselected) for a random sample of 10% of the cryptocurrencies in order to ensure that we had selected the "right" repositories, although without ground truth data it was of course impossible to guarantee this for all cryptocurrencies. A full list of the 13,694 available repositories, along with our ratings and our decision of whether to clone them or not, is available online.[4] We cloned 2354 repositories in total, which comprised roughly 100 GB of data.

3.2 Deployed source code

As evidenced by the 866 (52%) listed cryptocurrencies that were categorized as tokens (and the fact that 74 of these even had 'token' in their name), it is popular to launch new cryptocurrencies not as standalone coins, but as tokens that are supported by existing cryptocurrencies. Of these, by far the most popular type is an ERC20 token, supported by Ethereum. Of these listed tokens, 406 did not have any source code link available. For ERC20 tokens that have been deployed, however, it is often possible to obtain the contract code from another source: the version deployed on the Ethereum blockchain itself is compiled bytecode, but it is common practice to provide the Solidity code and display it on blockchain explorers such as Etherscan.[5]

For these tokens, we thus chose to use Etherscan as a data source (in addition to any provided repositories), in order to aid our Ethereum-based analysis in Section 5. At the time that we scraped Etherscan, there were 612 ERC20 tokens listed, identified by a name and a currency symbol (e.g., OmiseGO and OMG). Of these, we found 438 with a match on CoinMarketCap, where we defined a match as having (1) identical currency symbols, and (2) closely matching names. (We couldn't also require the name to be identical because in some cases the name of the contract was somewhat altered from the name of the cryptocurrency; e.g., SPANK instead of SpankChain.) We scraped the available contract code for each of these tokens, which in all but 9 cases was Solidity code rather than just on-chain bytecode. We thus ended up with 429 deployed ERC20 contracts.

4 Bitcoin Code Reuse

In this section, we attempt to identify the extent to which cryptocurrencies reuse the codebases of others, and in particular of Bitcoin. We do this by looking, very

[4] `https://github.com/manganese/alteramentum-repo-data`
[5] `https://etherscan.io/`

simply, at taking files from other repositories and using them without any modification. To identify this, we computed and stored the hash of every source code file in our cloned repositories; we identified source code file extensions using the CLOC library.[6] We then computed a similarity score S_{hash} between a repository A and another one B by counting the number of files in A with an identical file in B (meaning the hash was the same), and then dividing by the total number of files in A. To elevate this to the level of cryptocurrencies C_1 and C_2, we then computed $S_{\mathsf{hash}}(C_1, C_2)$ as

$$S_{\mathsf{hash}}(C_1, C_2) = \frac{\sum_{A \in C_1} S_{\mathsf{hash}}(A, \cup_{B \in C_2} B)}{\sum_{A \in C_1} \# \text{ files in } A};$$

i.e., for each repository A contributing to C_1 we counted the number of files that were identical to a file in *any* repository contributing to C_2, and then divided this by the total number of files across all repositories contributing to C_1.

We ran this for every pair of cryptocurrencies A and B (for both $S_{\mathsf{hash}}(A, B)$ and $S_{\mathsf{hash}}(B, A)$, since they are not symmetric), and used the results to create a graph in which nodes represent cryptocurrencies and there is a directed edge from A to B if $S_{\mathsf{hash}}(A, B) > 0.7$. This resulted in a graph with 445 nodes and 1854 edges, the largest connected component of which can be seen in Figure 1 (consisting of 302 nodes and 1599 edges).

Most of this component consists of Bitcoin forks. The exception is cluster 9, which consists of one cryptocurrency (Zeepin) that is 100% similar to 16 other cryptocurrencies. The reason is simple: its repository consisted solely of an LGPL-3.0 license, so it matched other repositories with the same version of this license. At the time we scraped CoinMarketCap, Zeepin had a market capitalization of 23 million USD. We can briefly explain clusters 1-8 as follows:

- **1.** The node at the center of this cluster, Akuya Coin, has a directory structure similar (63%) to a version of the Bitcoin codebase from 2013, but many (32%) of its files are empty and thus have the same hash, which makes it appear similar to 76 other Bitcoin forks.
- **2 and 3.** Both of these clusters also have a directory structure similar to older versions of the Bitcoin codebase (the average directory similarity was 89% for cluster 2 and 82% for cluster 3), and are similar to the same cryptocurrency (BumbaCoin). Many also incorporate the Zerocoin code:[7] 84% of the nodes in cluster 2 and 65% of the nodes in cluster 3. This is notable given that this code comes with the emphatic warning "THIS CODE IS UNMAINTAINED AND HAS KNOWN EXPLOITS. DO NOT USE IT." In total it is included in repositories for 97 different cryptocurrencies.
- **4 and 5.** These clusters were the ones most similar to Bitcoin: on average we had $S_{\mathsf{hash}} = 0.51$ and $S_{\mathsf{dir}} = 0.80$ for cluster 4 and $S_{\mathsf{hash}} = 0.37$ and $S_{\mathsf{dir}} = 0.96$ for cluster 5. For cluster 4, the matching versions were also in quite a tight range from September 2013 to September 2014 (our versions 9 to 11), whereas most other clusters ranged more evenly across all 18 versions.

[6] `https://github.com/AlDanial/cloc`
[7] `https://github.com/Zerocoin/libzerocoin`

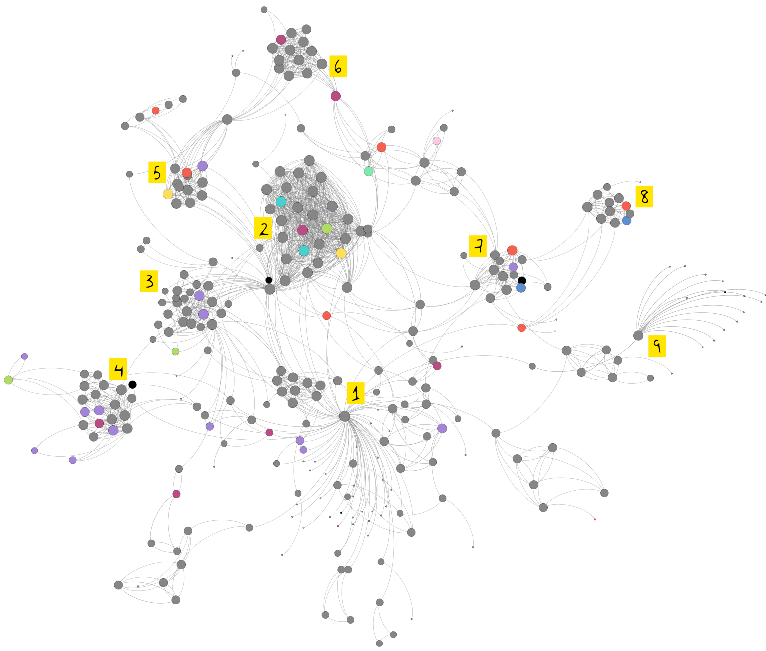


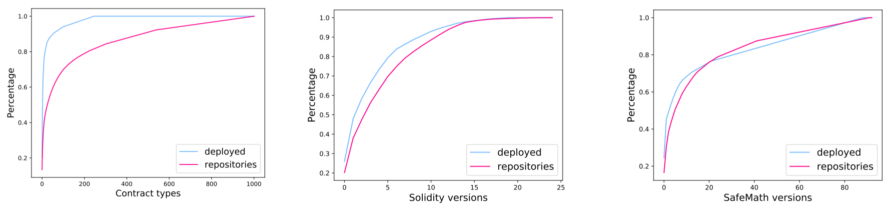
Fig. 1: The largest connected component of the graph formed by creating an edge from A to B if $S_{\mathsf{hash}}(A, B) > 0.7$, along with labels for the most prominent clusters.

– **6 and 7.** These clusters consisted largely of forks from Litecoin: 100% of cluster 6 had the file `scrypt.c`, which is unique to Litecoin. 64% of cluster 7 had files with `scrypt` in the name, although only 21% identified as copyright derivatives of anything other than Bitcoin.
– **8.** The nodes in this cluster were on average newer than the others (with the first repository created in June 2015), and indeed their directory structure is more consistent with newer versions of the Bitcoin codebase.

5 Ethereum as a Platform

As discussed in Section 3.2, it is increasingly popular to deploy cryptocurrencies as tokens on the Ethereum blockchain; indeed, over half of the cryptocurrencies listed on CoinMarketCap fell into this category. This section thus explores this type of cryptocurrency deployment, focusing again on the extent to which ERC20 tokens are similar to or different from each other. As an ERC20 token consists of just a single file, our methods from the previous sections do not apply here so we develop new methods for identifying similarities.

The basic functionality of an ERC20 token — allowing the transfer of tokens from one holder to another — defines a contract type called `Basic` (or `BasicToken`) or — with one slight functional difference — `ERC20`. There are, however, many additional types that ERC20 tokens can have. For example, if they want to allow for the creation of new tokens they can be `Mintable` and if they



(a) Types (b) Solidity version (c) SafeMath version

Fig. 2: When ranked from most to least popular, the cumulative percentage of contracts matching three different features, for both the set of deployed contracts and the ones found in repositories.

---

want to allow for the destruction of existing tokens they can be `Destructible` or `Burnable`. These types are not standardized, and in fact new types can be defined and used within the Solidity code for a contract.

To identify the types of a given token, we identified all lines in its contract of the form `contract X is Y {`, where `X` is the name of the contract and `Y` is its type. Some intermediate types themselves appear as names (e.g., `contract Mintable is Ownable`), which we exclude from our final results but carry over transitively to the higher-level contract names; e.g., if `X` is `Mintable` and `Mintable` is `Ownable` then `X` is both `Mintable` and `Ownable`. This resulted in a map from the higher-level token names to a list of all of their types.

Beyond these types, we also looked at the version the contract used of Solidity and of the SafeMath library, which provides safe arithmetic operations. For the version of Solidity, we looked for lines starting with `pragma solidity` and extracted the version from what followed (typically of the form `0.4.X`). To determine the version of SafeMath, we first used CLOC to strip the comments from the `.sol` file. We then identified the lines of code that defined the SafeMath library (starting with either `contract SafeMath {` or `library SafeMath {` and ending with `}`), and hashed this substring to form a succinct representation.

We extracted this information from all Solidity files, whether deployed on the Ethereum blockchain (and thus scraped from Etherscan, as described in Section 3.2) or contained in a repository.[8] For the types, Solidity and SafeMath versions, we ordered them from most to least popular and plotted this as a CDF, as seen in Figure 2; i.e., we plotted the percentage y of all contracts that had one of the top x attributes.

The relatively long tails in all of the figures indicate a relatively high level of diversity among these features in both deployed contracts and those still under development. For example, the Solidity version most popular among deployed contracts (version 18) was still used in only 23% of them. Whereas Figures 2b and 2c show similar curves for both sets of contracts, Figure 2a shows a much longer tail for contracts contained in repositories, with 246 distinct types in deployed contracts and 1002 in ones in repositories. This indicates — as should

[8] Interestingly, these sets were non-intersecting; i.e., there was no contract in a repository that was identical to a deployed one.

perhaps be expected — that (1) there are just many more possibilities for contract types than for versions, and (2) there is greater experimentation with types in contracts still under development. Even among deployed contracts, 129 out of 429 had a type that did not appear in any other deployed contracts, and 148 of the 246 distinct types appeared in only a single contract.

Finally, we view the points of similarity that did exist as operating primarily in support of the safety of deployed contracts. For example, among the 20 most popular types across both deployed and repository contracts, five of them defined the basic ERC20 functionality, and six of them were related to safety in terms of either including a standard library or in defining an owner who could take action if something went wrong. The same is true of the usage of FirstBlood's `StandardToken`, which was the first safe implementation of this type, or of the `SafeMath` library. We thus view these similarities as a sign of good development practices, rather than the copying of ideas.

6 Conclusions

This paper considered diversity in the cryptocurrency landscape, according to the source code available for each one, in order to identify the extent to which new cryptocurrencies provide meaningful innovation. This was done by examining the source code for over a thousand cryptocurrencies, and — in the case of ERC20 tokens — the deployed code of hundreds more. While more sophisticated static analysis of the source code would likely yield further insights, even our relatively coarse methods clearly indicated the dominance of Bitcoin and Ethereum, as well as the extent to which creating a standalone platform is a significantly greater undertaking (leading to the reuse of much of the Bitcoin codebase) than defining just the transaction semantics of an Ethereum-based token.

Acknowledgements

The authors were supported in part by EPSRC Grant EP/N028104/1 and in part by the EU H2020 TITANIUM project under grant agreement number 740558.

References

1. M. Apostolaki, A. Zohar, and L. Vanbever. Hijacking bitcoin: Routing attacks on cryptocurrencies. In *2017 IEEE Symposium on Security and Privacy*, pages 375–392, San Jose, CA, USA, May 22–26, 2017. IEEE Computer Society Press.
2. S. Azouvi, M. Maller, and S. Meiklejohn. Egalitarian society or benevolent dictatorship: The state of cryptocurrency governance. In *Proceedings of the 5th Workshop on Bitcoin and Blockchain Research*, 2018.
3. M. Bartoletti, S. Carta, T. Cimoli, and R. Saia. Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact, 2017.
4. A. Biryukov, D. Khovratovich, and I. Pustogarov. Deanonymisation of clients in bitcoin P2P network. In G.-J. Ahn, M. Yung, and N. Li, editors, *ACM CCS 14*, pages 15–29, Scottsdale, AZ, USA, Nov. 3–7, 2014. ACM Press.

5. J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow. Elliptic curve cryptography in practice. In N. Christin and R. Safavi-Naini, editors, *FC 2014*, volume 8437 of *LNCS*, pages 157–175, Christ Church, Barbados, Mar. 3–7, 2014. Springer, Heidelberg, Germany.
6. W. Chen, Z. Zheng, J. Cui, E. Ngai, P. Zheng, and Y. Zhou. Detecting Ponzi schemes on Ethereum: Towards healthier blockchain technology. In *Proceedings of the 2018 World Wide Web Conference (WWW)*, 2018.
7. B. David, P. Gazi, A. Kiayias, and A. Russell. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain. In J. B. Nielsen and V. Rijmen, editors, *EUROCRYPT 2018, Part II*, volume 10821 of *LNCS*, pages 66–98, Tel Aviv, Israel, Apr. 29 – May 3, 2018. Springer, Heidelberg, Germany.
8. J. A. D. Donet, C. Pérez-Solà, and J. Herrera-Joancomartí. The bitcoin P2P network. In R. Böhme, M. Brenner, T. Moore, and M. Smith, editors, *FC 2014 Workshops*, volume 8438 of *LNCS*, pages 87–102, Christ Church, Barbados, Mar. 7, 2014. Springer, Heidelberg, Germany.
9. I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse. Bitcoin-ng: A scalable blockchain protocol. In *Proceedings of NSDI 2016*, 2016.
10. A. E. Gencer, S. Basu, I. Eyal, R. V. Renesse, and E. G. Sirer. Decentralization in Bitcoin and Ethereum networks. In *Proceedings of the 22nd International Conference on Financial Cryptography and Data Security (FC)*, 2018.
11. A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun. On the security and performance of proof of work blockchains. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *ACM CCS 16*, pages 3–16, Vienna, Austria, Oct. 24–28, 2016. ACM Press.
12. Y. Hu, J. Zhang, X. Bai, S. Yu, and Z. Yang. Influence analysis of GitHub repositories. *SpringerPlus*, 5(1), 2016.
13. D. Y. Huang, K. Levchenko, and A. C. Snoeren. Measuring profitability of alternative crypto-currencies. In *Proceedings of the 22nd International Conference on Financial Cryptography and Data Security (FC)*, 2018.
14. G. Kappos, H. Yousaf, M. Maller, and S. Meiklejohn. An empirical analysis of anonymity in Zcash. In *Proceedings of the USENIX Security Symposium*, 2018.
15. A. Kiayias, A. Russell, B. David, and R. Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. In J. Katz and H. Shacham, editors, *CRYPTO 2017, Part I*, volume 10401 of *LNCS*, pages 357–388, Santa Barbara, CA, USA, Aug. 20–24, 2017. Springer, Heidelberg, Germany.
16. E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford. Enhancing Bitcoin security and performance with strong consistency via collective signing. In *Proceedings of USENIX Security 2016*, 2016.
17. E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, and B. Ford. Omniledger: A secure, scale-out, decentralized ledger. In *Proceedings of the 39th IEEE Symposium on Security & Privacy*, 2018.
18. P. Koshy, D. Koshy, and P. McDaniel. An analysis of anonymity in bitcoin using P2P network traffic. In N. Christin and R. Safavi-Naini, editors, *FC 2014*, volume 8437 of *LNCS*, pages 469–485, Christ Church, Barbados, Mar. 3–7, 2014. Springer, Heidelberg, Germany.
19. L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena. A secure sharding protocol for open blockchains. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, *ACM CCS 16*, pages 17–30, Vienna, Austria, Oct. 24–28, 2016. ACM Press.

20. S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage. A fistful of bitcoins: characterizing payments among men with no names. In *Proceedings of the 2013 Internet Measurement Conference (IMC)*, pages 127–140, 2013.
21. T. Moore and N. Christin. Beware the middleman: Empirical analysis of Bitcoin-exchange risk. In A.-R. Sadeghi, editor, *FC 2013*, volume 7859 of *LNCS*, pages 25–33, Okinawa, Japan, Apr. 1–5, 2013. Springer, Heidelberg, Germany.
22. M. Möser, K. Soska, E. Heilman, K. Lee, H. Heffan, S. Srivastava, K. Hogan, J. Hennessey, A. Miller, A. Narayanan, and N. Christin. An empirical analysis of linkability in the Monero blockchain. *Proceedings on Privacy Enhancing Technologies*, 2016(3):143–163, 2018.
23. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. `bitcoin.org/bitcoin.pdf`.
24. B. Ray, D. Posnett, V. Filkov, and P. Devanbu. A large scale study of programming languages and code quality in github. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE)*, pages 155–165, 2014.
25. F. Reid and M. Harrigan. An analysis of anonymity in the Bitcoin system. In *Security and privacy in social networks*, pages 197–223. Springer, 2013.
26. D. Ron and A. Shamir. Quantitative analysis of the full Bitcoin transaction graph. In A.-R. Sadeghi, editor, *FC 2013*, volume 7859 of *LNCS*, pages 6–24, Okinawa, Japan, Apr. 1–5, 2013. Springer, Heidelberg, Germany.
27. M. Spagnuolo, F. Maggi, and S. Zanero. BitIodine: Extracting intelligence from the bitcoin network. In N. Christin and R. Safavi-Naini, editors, *FC 2014*, volume 8437 of *LNCS*, pages 457–468, Christ Church, Barbados, Mar. 3–7, 2014. Springer, Heidelberg, Germany.
28. E. Syta, I. Tamas, D. Visher, D. I. Wolinsky, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, and B. Ford. Keeping authorities“honest or bust” with decentralized witness cosigning. In *2016 IEEE Symposium on Security and Privacy*, pages 526–545, San Jose, CA, USA, May 22–26, 2016. IEEE Computer Society Press.
29. F. Thung, T. F. Bissyandé, D. Lo, and L. Jiang. Network structure of social coding in GitHub. In *Proceedings of the 17th European Conference on Software Maintenance and Reengineering*, 2013.
30. M. Vasek, J. Bonneau, R. Castellucci, C. Keith, and T. Moore. The bitcoin brain drain: Examining the use and abuse of bitcoin brain wallets. In J. Grossklags and B. Preneel, editors, *FC 2016*, volume 9603 of *LNCS*, pages 609–618, Christ Church, Barbados, Feb. 22–26, 2016. Springer, Heidelberg, Germany.
31. M. Vasek and T. Moore. There's no free lunch, even using bitcoin: Tracking the popularity and profits of virtual currency scams. In R. Böhme and T. Okamoto, editors, *FC 2015*, volume 8975 of *LNCS*, pages 44–61, San Juan, Puerto Rico, Jan. 26–30, 2015. Springer, Heidelberg, Germany.
32. M. Vasek and T. Moore. Analyzing the Bitcoin Ponzi scheme ecosystem. In *Proceedings of the 5th Workshop on Bitcoin and Blockchain Research*, 2018.
33. M. Vasek, M. Thornton, and T. Moore. Empirical analysis of denial-of-service attacks in the bitcoin ecosystem. In R. Böhme, M. Brenner, T. Moore, and M. Smith, editors, *FC 2014 Workshops*, volume 8438 of *LNCS*, pages 57–71, Christ Church, Barbados, Mar. 7, 2014. Springer, Heidelberg, Germany.
34. J. Zhu, M. Zhou, and A. Mockus. Patterns of folder use and project popularity: A case study of Github repositories. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 2014.