# Ethical Challenges in Blockchain Network Measurement Research

Yuzhe Tang
Syracuse University
ytang100@syr.edu

Kai Li[†]
San Diego State University
kli5@sdsu.edu

Yibo Wang
Syracuse University
ywang349@syr.edu

Jiaqi Chen
Syracuse University
jchen217@syr.edu

***Abstract*—Public blockchains are the digital infrastructure that powers the multi-trillion-dollar economy in cryptocurrencies. Understanding the security and performance of deployed blockchain networks is critically important, especially when the open-membership nature of blockchain results in a large attack surface. However, measuring operational blockchain networks raises ethical concerns and could interfere with the businesses running atop the blockchains. This work presents a survey of the recent measurement studies on the Ethereum networks and discusses their ethical issues, practices, and solutions. The paper also identifies several open ethical challenges faced by blockchain researchers.**

## I. INTRODUCTION

Today, cryptocurrency is a multi-trillion-dollar economy. Understanding the security of the blockchain technology that underpins cryptocurrencies is critically important to ensuring crypto-asset safety. This urgent need has driven a line of security-oriented measurement studies on the public blockchains [21], [27], [18], [19], [16], [15], [12], [14], [10]. The measurement studies have lead to significant findings on the security, privacy, performance, and other properties of blockchain networks in the wild.

On the one hand, measuring real-world blockchain networks is necessary for scientific discovery. While alternative measurement subjects do exist (e.g., blockchain simulation [26], local blockchain networks, testnets [8], [7], [3]), measuring deployed blockchain mainnet is preferable as it holds the ground truth of various deployment-specific information. For instance, mainnet nodes may be configured and operated differently to testnet nodes, forming a distinct network topology or deploying unique defensive measures. Understanding these mainnet-specific features entails direct measurement on the blockchain mainnet.

On the other hand, measurement studies on real-world blockchain networks raise ethical concerns. There are real business entities whose daily operations depend on blockchains; these companies may run network infrastructures for blockchains (e.g., RPC services and transaction relays), run mining pools, operate wallet software (e.g., web browser extensions), and others to serve a large quantity of customers including miners and cryptocurrency holders. The measurement study on blockchain networks can cause service interference to these stakeholders.

The key to solving the conflict lies in ethical measurement methodologies and studies on blockchain networks. In this paper, we review several recent blockchain measurement studies [21], [27], [18], [19] to examine the current state of the affair in the field. Specifically, we review DETER [21], [27] and DoERS [17] which measure the Ethereum mainnet to understand the exploitability of design flaws in Ethereum client software. We also review TopoShot [19], a network measurement study uncovering the Ethereum network topology on mainnet. We examine the extensive measures taken in these works to address the ethical concerns, including following ethical research methodology, the technical design of non-intrusive measurement methods, responsible bug reporting, and result releases. The four case studies are presented in Section II.

We further draw insights into the open challenges blockchain researchers face today and call for future works in this domain. The challenges are discussed in detail in Section III.

## II. CASE STUDIES

We choose to survey the measurement works on Ethereum, the second largest blockchain and the largest smart-contract platform. We focus on the following four papers because they either directly study the deployed Ethereum networks (including mainnet and testnets) [21], [27], [18], [19] or indirectly have impacts on those networks [32].

**Case 1**: In DoERS [17], the authors have conducted security analysis of one critical component in the Ethereum blockchain ecosystem, the RPC service. An RPC service receives web clients' queries and processes them against its blockchain states. The authors found one of the standard query API, `eth_call`[1], is exploitable, as it admits generic program execution without charging a querying client. This API, if exposed to the Internet, can be misused to run denial of service attacks exhausting the computing resources on the RPC service. Then, the authors formulate research by measuring the exploitability of the `eth_call` API on several real-world RPC services.

[†] This work is done when Kai Li was a Ph.D. student at Syracuse University.

[1] `eth_call` is standardized in the Ethereum protocol and is widely supported in Geth [2], OpenEthereum [6] and other clients [5], [4].

Workshop on Ethics in Computer Security (EthiCS) 2023
27 February 2023, San Diego, CA, USA
ISBN 1-891562-85-1
https://dx.doi.org/10.14722/ethics.2023.235725
www.ndss-symposium.org

From the perspective of scientific discovery, it is preferable to conduct an exploitability study on real-world services. Specifically, while there are RPC services run on testnets and the mainnet, the mainnet RPC service may treat their business more seriously and deploy more comprehensive defensive measures. Thus, only through studying the mainnet services can one uncover these defensive measures before further discovering flaws. Besides, the findings on the mainnet services will have broader impacts than those on the testnet counterpart.

However, this raises ethical concerns, and the mainnet RPC services being tested are operated by real business entities. The exploitability study needs to be non-intrusive to real-world services, while ensuring effectiveness.

The authors take extensive measures to address the challenges. These include constraining each test's time duration and computation amount, focusing on the trend (how response time increases along with more intensive payload) instead of absolute value, etc. Other than the technical solutions, the authors also send bug reports to the services tested in a timely manner. This initiates a conversation between the authors and the managers in the service companies.

**Case 2**: In DETER [21], [27], authors analyze the security of mempool, a system component in Ethereum blockchain that buffers unconfirmed transactions prior to mining. Unconfirmed transactions may or may not be included in the blockchain. The authors manually discovered a new attack to deny a remote mempool's service by exploiting the mempool's admission control.

The authors have evaluated the attack effectiveness and cost. They have tested Ethereum nodes running in a local network, testnets and the mainnet. In the local network, the evaluation is comprehensive and extensive by running the actual DETER attack payload for as long time as needed. In the testnets, the authors restrict the attack to one selected node in the testnet and limit the load and duration (e.g., 1 minute). They stop the attack as soon as they see the attack starts to show effects in the blocks produced. On the mainnet, they did not launch the actual attack; Instead, they designed lightweight probes to test the cause of the DETER vulnerabilities on selected mainnet nodes.

They report the bugs immediately to the affected developer communities through a bug bounty program. Both bug bounty programs in the Ethereum Foundation (for Geth development) and OpenEthereum are very responsive and cooperative. The bounty programs reproduce the bug and confirm the bug severity quickly. They also generously give bug bounty to the authors to acknowledge their efforts. In addition, through bug reporting, the bounty programs deploy quick fixes in their client software. Other programs, such as Nethermind, are less responsive, possibly due to their declined interest in maintaining the client software.

The authors also report the bug to individual service providers. However, these service providers are much more conservative, and none did respond to the bug reports.

**Case 3**: TopoShot [19] is a proactive measurement method that leverages DETER [21], [27] and transaction replacement to measure the Ethereum network topology. Specifically, TopoShot utilizes Ethereum's price bump feature to enforce the "isolation" property [12] for accurate network measurement.

The price bump is a necessary defensive measure against transaction flooding, specifically flooding via replacement transactions. The authors did discover several clients support transactions without ensuring price bumps. They report the bug to the bug bounty and get the response quickly. The client software quickly fixes the issues by adding a 10% price bump. This allows the authors to be able to measure the Ethereum networks running various client implementations.

Once the Ethereum network topology is measured, the authors did some in-house graph analysis. They found that many nodes have a small degree, with few neighbors. These low-degree nodes are particularly vulnerable to network partitioning attacks, such as eclipse attacks. Thus, directly publishing the raw measurement results would expose these vulnerable nodes to high risk. The authors exercise anonymization and privacy preservation techniques to remove the ID information (e.g., node IP) from the dataset before open-sourcing the dataset for secondary use by researchers.

**Case 4**: In Fluffy [32], the authors build a fuzzer to find consensus bugs in Ethereum clients. To search for consensus bugs, Fluffy mutates and executes a multi-transaction sequence in order to reach the deep states hidden in long paths in Ethereum clients. Fluffy is a differential fuzzer that uses Geth and OpenEthereum clients as cross-checking oracles to detect consensus bugs. The authors found two new consensus bugs in Geth client, shallow copy bug and transfer-after-destruct bug, that are caused by the deviation between client implementation and protocol specification. In the transfer-after-destruct bug, the Geth implementation carries over the balance of a deleted account to a newly created account with the same address. This implementation deviates from EVM (Ethereum Virtual Machine) specification and can be exploited for network split and theft.

The authors reported the bugs to Geth development team who then quickly fixed the bugs in upcoming releases. However, it drags behind for deployed blockchain nodes to install the latest software releases. Four months later, on Nov. 11, 2020, Optimism, an Ethereum scaling project, reportedly on purpose, triggered the bug on the Ethereum mainnet [9] which caused catastrophic damage: It forces the mainnet to partition into two groups of nodes, the nodes running updated Geth and those that don't. In other words, the attack disabled the consensus of nodes in the network. As a result, infrastructure services were down, including Infura, Metamask, MakerDAO, Uniswap, and Compound. The incident raised concerns on the ethics of blockchain security research and the transparency of bug fixing. It is advised that the Geth team should have announced the importance of upgrading to software versions with the fix.

## III. Open Ethical Challenges

Security researchers face a series of challenges in the research life cycle, from applying for IRB approval to the end of releasing results. Based on our experiences in the first three works [21], [18], [19] and observing the discussion in public forums for the forth one [32], we identify the following ethical challenges.

**Lack of guidelines**. The first-time security researchers lack awareness or guidelines for ethical issues. While Ph.D. students and professors may be well trained on technical issues, they are not on ethical terms. US undergraduate education conforms to ABET, and it does require ethical performance indicators. However, many researchers are international and not educated in US colleges.

Existing available resources are ad-hoc; students rely on prior research publications to seek solutions for ethical issues. Unfortunately, however, technical publications often discard details on ethical issues and sometimes do so intentionally. This unsatisfactory state leaves the researchers unprepared for ethical issues, giving them no option but to trial by fire.

**Dilemma to request approval**. Suppose a security researcher needs to test a real-world service. A dilemma is whether and how to request approvals before the actual test. First, the target service may not have a bug bounty program or does not leave the contact on the Internet, leaving the researcher nowhere to send their requests.

Second, on the one hand, the service provider does not have the incentive to or may not recognize the long-term benefit of exposing their service to a stranger researcher. On the other hand, the researcher may not have a clear mind on what and how to test the service, let alone how to control the level of interference. This makes it hard for the tested service to approve the request.

**Effectiveness of IRB**. The institutions researchers work in may not have an internal review board (IRB) in place. For the institutions that do have IRB, the researchers may not be aware of it. It entails to raise IRB awareness among professors and students on campus.

From the authors' experience, IRB provides a list of standard survey questions to the researchers who answer them. IRB professionals make decisions based on these answers. In this process, IRB lacks technical knowledge and there is a need, commonly unmet, to educate IRB on technical terms.

**Limited support for bug reporting**. Unique in public blockchains, most services (e.g., RPC services, mining pools) are operated by small businesses. Due to reasons such as limited budget, these services have neither extensive protections against Internet hacks nor a bug bounty program. This creates ethical barriers for cyber-security researchers and white-hats to test deployed services and report bugs.

For instance, blockchain services use public forums to host the bug reports. Nethermind relies on the GitHub "issues" feature to report bugs[2]. Each reported bug, before it is fixed, has to be exposed to the public, which also exposes any nodes running the client software to possible exploitation.

TABLE I: Bug report programs in major Ethereum clients

| | Geth | OpenEth | Besu | Nethermind |
|---|---|---|---|---|
| Private bug reporting | Yes | Yes | No | No |
| Responsiveness (days) | 2 | 1 | N/A | N/A |
| Bounty | Yes | Yes | No | No |
| Fix | Yes | No | No | No |
| Deployment in mainnet | 83% | 15% | 1.5% | 0.5% |

We summarize the bug reporting support of different Ethereum client developer communities in Table I. We obtain the data from our personal experience and by private conversation with the authors of papers studied in the above cases. In the table, the private reporting channel refers to whether the client software has a private program to report the discovery of security vulnerabilities, such as a well-maintained email account. This excludes a program that only has a public website or forum to report bugs. Responsiveness refers to how fast a bug report is confirmed. Bounty refers to whether the program has rewarded bounty to the authors of the research described in the previous section. Fix refers to whether the reported bug is fixed six months after the bug report. The deployment in the mainnet refers to the percentage of mainnet nodes running the client (as of 2020 [19]).

As can be seen, large Ethereum clients such as Geth and OpenEthereum have very good support for bug reporting. Other clients with smaller deployment either rely on third-party, public forums to support bug reporting or have private bug reporting programs that are not well maintained.

**Responsible result release**. Research results can be sensitive due to various reasons, such as the potential to damage business reputation, containing information that may expose someone at risk, and others. Releasing the research results to the public may raise concerns in various forms like conference proceedings, open-source datasets/software artifacts, etc. Anonymization, such as removing identifiable information, is a good practice. However, this is often done manually, which is error-prone. It is an open question whether the anonymization practice in cyber-security publications can be reverted? Or how to ensure anonymization in cyber-security publications.

## IV. Related Works

There is a line of blockchain measurement studies uncovering blockchain infrastructures [31], [22] and applications, such as DeFi [25], [24], [33]. This research line takes a passive measurement method, such as collecting transactions from exploration websites (e.g., etherscan.io) or by joining a regular node in the network. We don't consider passive measurement studies in ethical concerns.

Blockchain cost efficiency is measured by the amount of cryptocurrencies (e.g., Ether) per computing task. Existing research improves the blockchain cost efficiency by extending the networks with off-chain services (i.e., so-called layer-two solutions [1], [23], [11], [13]) or building on-chain/off-chain middleware [30], [29], [20], [28]. This line of research, while introducing new services to the networks, does not affect other nodes beyond what a newly joined node does; we don't consider them interfering.

## V. Conclusion

This work reviews the recent blockchain measurement studies and discusses their ethical issues, practices, and solutions. The paper also identifies several open ethical challenges faced by blockchain researchers.

[2] https://github.com/NethermindEth/nethermind/issues/3173

## Acknowledgment

The authors are partially supported by the National Science Foundation under Grant CNS1815814, DGE2104532 and CNS2139801.

## References

[1] Ligntning network, scalable, instant bitcoin/blockchain transactions.

[2] Geth: the go client for ethereum. https://www.ethereum.org/cli\#geth, Retrieved May, 2021.

[3] The goerli testnet of ethereum. https://goerli.etherscan.io, Retrieved May, 2021.

[4] Hyperledger besu. https://www.hyperledger.org/use/besu, Retrieved May, 2021.

[5] Nethermind ethereum client. https://nethermind.io/client, Retrieved May, 2021.

[6] Parity ethereum is now openethereum: Fast and feature-rich multi-network ethereum client. https://www.parity.io/ethereum/, Retrieved May, 2021.

[7] The rinkeby testnet of ethereum. https://rinkeby.etherscan.io, Retrieved May, 2021.

[8] The ropsten testnet of ethereum. https://ropsten.etherscan.io, Retrieved May, 2021.

[9] Geth v1.9.17 post mortem. https://gist.github.com/karalabe/e1891c8a99fdc16c4e60d9713c35401f, Retrieved Sep., 2022.

[10] T. Cao, J. Yu, J. Decouchant, X. Luo, and P. Veríssimo. Exploring the monero peer-to-peer network. In J. Bonneau and N. Heninger, editors, Financial Cryptography and Data Security - 24th International Conference, FC 2020, Kota Kinabalu, Malaysia, February 10-14, 2020 Revised Selected Papers, volume 12059 of Lecture Notes in Computer Science, pages 578–594. Springer, 2020.

[11] R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. M. Johnson, A. Juels, A. Miller, and D. Song. Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contract execution. CoRR, abs/1804.05141, 2018.

[12] S. Delgado-Segura, S. Bakshi, C. Pérez-Solà, J. Litton, A. Pachulski, A. Miller, and B. Bhattacharjee. Txprobe: Discovering bitcoin's network topology using orphan transactions. In Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Frigate Bay, St. Kitts and Nevis, February 18-22, 2019, Revised Selected Papers, pages 550–566, 2019.

[13] S. Dziembowski, S. Faust, and K. Hostáková. General State Channel Networks. In CCS 2018, pages 949–966, 2018.

[14] M. Grundmann, T. Neudecker, and H. Hartenstein. Exploiting transaction accumulation and double spends for topology inference in bitcoin. In Financial Cryptography and Data Security - FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers, pages 113–126, 2018.

[15] L. Kiffer, A. Salman, D. Levin, A. Mislove, and C. Nita-Rotaru. Under the hood of the ethereum gossip protocol. In N. Borisov and C. Diaz, editors, Financial Cryptography and Data Security - 25th International Conference, FC 2021, Virtual Event, March 1-5, 2021, Revised Selected Papers, Part II, volume 12675 of Lecture Notes in Computer Science, pages 437–456. Springer, 2021.

[16] S. K. Kim, Z. Ma, S. Murali, J. Mason, A. Miller, and M. Bailey. Measuring ethereum network peers. In Proceedings of IMC 2018, pages 91–104, 2018.

[17] K. Li, J. Chen, X. Liu, Y. Tang, X. Wang, and X. Luo. As strong as its weakest link: How to break blockchain dapps at RPC service. In 28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021. The Internet Society, 2021.

[18] K. Li, J. Chen, X. Liu, Y. R. Tang, X. Wang, and X. Luo. As strong as its weakest link: How to break blockchain dapps at RPC service. In 28th Annual Network and Distributed System Security Symposium, NDSS 2021, virtually, February 21-25, 2021. The Internet Society, 2021.

[19] K. Li, Y. Tang, J. Chen, Y. Wang, and X. Liu. Toposhot: uncovering ethereum's network topology leveraging replacement transactions. In D. Levin, A. Mislove, J. Amann, and M. Luckie, editors, IMC '21: ACM Internet Measurement Conference, Virtual Event, USA, November 2-4, 2021, pages 302–319. ACM, 2021.

[20] K. Li, Y. R. Tang, J. Chen, Z. Yuan, C. Xu, and J. Xu. Cost-effective data feeds to blockchains via workload-adaptive data replication. In D. D. Silva and R. Kapitza, editors, Middleware '20: 21st International Middleware Conference, Delft, The Netherlands, December 7-11, 2020, pages 371–385. ACM, 2020.

[21] K. Li, Y. Wang, and Y. Tang. DETER: denial of ethereum txpool services. In Y. Kim, J. Kim, G. Vigna, and E. Shi, editors, CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021, pages 1645–1667. ACM, 2021.

[22] J. Messias, M. Alzayat, B. Chandrasekaran, K. P. Gummadi, P. Loiseau, and A. Mislove. Selfish & opaque transaction ordering in the bitcoin blockchain: the case for chain neutrality. In D. Levin, A. Mislove, J. Amann, and M. Luckie, editors, IMC '21: ACM Internet Measurement Conference, Virtual Event, USA, November 2-4, 2021, pages 320–335. ACM, 2021.

[23] A. Miller, I. Bentov, R. Kumaresan, and P. McCorry. Sprites: Payment channels that go faster than lightning. CoRR, abs/1702.05812, 2017.

[24] K. Qin, L. Zhou, P. Gamito, P. Jovanovic, and A. Gervais. An empirical study of defi liquidations: incentives, risks, and instabilities. In D. Levin, A. Mislove, J. Amann, and M. Luckie, editors, IMC '21: ACM Internet Measurement Conference, Virtual Event, USA, November 2-4, 2021, pages 336–350. ACM, 2021.

[25] K. Qin, L. Zhou, and A. Gervais. Quantifying blockchain extractable value: How dark is the forest? In 43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022, pages 198–214. IEEE, 2022.

[26] M. Tran, I. Choi, G. J. Moon, A. V. Vu, and M. S. Kang. A Stealthier Partitioning Attack against Bitcoin Peer-to-Peer Network. In To appear in Proceedings of IEEE Symposium on Security and Privacy (IEEE S&P), 2020.

[27] Y. Wang, K. Li, and Y. Tang. Towards the comprehensive understanding of ethereum mempool dos security.

[28] Y. Wang, K. Li, Y. Tang, J. Chen, Q. Zhang, X. Luo, and T. Chen. Towards saving blockchain fees via secure and cost-effective batching of smart-contract invocations. IEEE Transactions on Software Engineering, (01):1–20, jan 5555.

[29] Y. Wang and Y. Tang. Poster: Enabling cost-effective blockchain applications via workload-adaptive transaction execution. In H. Yin, A. Stavrou, C. Cremers, and E. Shi, editors, Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022, pages 3483–3485. ACM, 2022.

[30] Y. Wang, Q. Zhang, K. Li, Y. Tang, J. Chen, X. Luo, and T. Chen. ibatch: saving ethereum fees via secure and cost-effective batching of smart-contract invocations. In D. Spinellis, G. Gousios, M. Chechik, and M. D. Penta, editors, ESEC/FSE '21: 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering, Athens, Greece, August 23-28, 2021, pages 566–577. ACM, 2021.

[31] B. Weintraub, C. F. Torres, C. Nita-Rotaru, and R. State. A flash(bot) in the pan: measuring maximal extractable value in private pools. In C. Barakat, C. Pelsser, T. A. Benson, and D. Choffnes, editors, Proceedings of the 22nd ACM Internet Measurement Conference, IMC 2022, Nice, France, October 25-27, 2022, pages 458–471. ACM, 2022.

[32] Y. Yang, T. Kim, and B. Chun. Finding consensus bugs in ethereum via multi-transaction differential fuzzing. In A. D. Brown and J. R. Lorch, editors, 15th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2021, July 14-16, 2021, pages 349–365. USENIX Association, 2021.

[33] L. Zhou, K. Qin, A. Cully, B. Livshits, and A. Gervais. On the just-in-time discovery of profit-generating transactions in defi protocols. In 42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021, pages 919–936. IEEE, 2021.