

AUDITING SMART CONTRACTS*

Wayne Landsman[†] Evgeny Lyandres[‡] Edward Maydew[§]

Daniel Rabetti[¶] Che Zhang^{**}

First draft: February 2023

This draft: October 2025

Abstract

This study assesses the emerging market for voluntary audits of smart contracts used in Decentralized Finance (DeFi) by analyzing thousands of audit reports from hundreds of auditing firms and linking them to thousands of DeFi protocols launched between January 2020 and January 2025. Two types of auditors operate in the DeFi market: centralized auditors, who are hired on a fixed-fee basis, and decentralized auditors—also known as *bounty hunters*—who are compensated based on the vulnerabilities they identify. We address three questions: (1) What factors are associated with a protocol’s decision to engage an auditor, and what type of auditor to hire before and after protocol launch? (2) Do audits effectively mitigate the likelihood of future security breaches? (3) How do developers and auditors respond to security breaches? We find that pre-launch audit adoption is more likely among protocols with risk-exposed designs (e.g., protocols featuring oracles and cross-chain deployments) and that systemic breach events significantly increase demand for top-tier centralized and decentralized audits. Our evidence reveals that, on average, audits do not reduce the likelihood of future security breaches. However, audits conducted by top-tier centralized auditors and decentralized auditors are associated with a lower likelihood of future breaches and losses conditioned on a breach occurring. Following a breach, DeFi developers often replace bottom-tier auditors with top-tier ones and pivot towards decentralized auditors. Affected auditors suffer short-term reputational losses unless they engage in active crisis management. Overall, our findings highlight the potential and limits of voluntary assurance in decentralized finance, with practical implications for: developers evaluating audit strategies, security firms navigating reputation risks, and policymakers considering how to regulate the integrity of decentralized financial systems.

JEL classification: G15, G18, G29, K29, K42, O16.

Keywords: DeFi, auditing, bounty hunters, smart contracts, security breaches, blockchain.

*We are thankful to Nemit Shroff (Editor) and an anonymous reviewer for invaluable comments and suggestions. We are also thankful to Amanda Awyong, Haichen Bai, Daniel Bens, Masaki Bessho, Thomas Bourveau, Mark Bradshaw, Murillo Campello, Qiang Cheng, Jonathan Chiu, Will Cong, Xu Da, Stephanie Dong, Vivian Fang, Cam Harvey, Allen Huang, Serene Huang, Shiyang Huang, Mingyi Hung, Peiyi Jin, Bjorn Jorgensen, Peter Joos, Elsa Juliani, Bin Ke, Stella Kong, Alfred Lehar, Dan Li, Erica Li, Mei Luo, Yuanzhen Lyu, Roni Michaely, Chris Ngoi, Christine Parlour, Yuanyu Qu, Thomas Rivera, Fahad Saleh, Fabian Schar, Joe Schroeder, Rob Smith, Jona Stinner, Tammaro Terracciano, Alexander Wehrli, Michael Wutzke, Xiao Xiao, Yu Yan, Cheng Yin, Bernard Yeung, Shuangchen Yu, Xiaojun Zhang, and Yihong Zhou for comments in early versions of this study. We benefited from helpful discussions at the International Monetary Fund (IMF), ETH Denver Festival, Harvard Business School, Tokenomics Conference, Tsinghua University, Peking University, University of International Business and Economics, Cornell-PBC Summer Institute of Finance, IC3 Blockchain Camp at Cornell Tech, Swiss National Bank and University of Basel via SNB-CIF Conference on Cryptoassets and Financial Innovation, Euroasia Conference, Bank for International Settlements, Bank of Japan, Fintech Center, Payment and Settlement Systems Department, FeAT International Conference on Artificial Intelligence, Singapore Fintech Festival, Financial Markets and Corporate Governance Conference, ICMA Centre at Henley Business School, University of Reading, Capital Market Research in the Era of AI Conference at HKUST, Vietnam Symposium in Banking and Finance, and CBER Symposium in Auditing DeFi Applications. Rabetti thanks the Digital Economy and Financial Technology (DEFT) Lab at Cornell Fintech Initiative, the Research Center for Digital Financial Assets at Tsinghua University, and the Asian Institute of Digital Finance (AIDF) at the National University of Singapore for extended discussions. Lyandres thanks the Cassirer Institute at Tel Aviv University and Fintech Chair at Paris Dauphine University for financial support. Landsman and Maydew thank the Kenan-Flagler Business School for financial support. Zhang acknowledges financial support from the National Natural Science Foundation of China (grant no. 72502128). Ding Chou, Gia Anna George, Vaishnavi Gunasekaran, Xing Hexin, Yiqing Huang, Li Shengzhi, Zhenhui Xi, Huiting Zhou, Xuanhao Zeng, Yilin Zhou, Shimin Zhang, and Li Ziting provided excellent research assistance. A February 2023 version of this paper circulated under the title “*Auditing Decentralized Finance (DeFi) protocols*”. All errors are our own.

[†]University of North Carolina, wayne_landsman@unc.edu.

[‡]Tel Aviv University and Monash University, lyandres@tauex.tau.ac.il.

[§]University of North Carolina, edward_maydew@unc.edu.

[¶]Corresponding Author: National University of Singapore (NUS) Business School, 15 Kent Ridge Drive, Singapore, 119245.

^{||}ABFER Research Fellow. Harvard Business School (visiting), drabetti@hbs.edu.

^{**}Tsinghua University, zhangche@sem.tsinghua.edu.cn.

“On-chain assets are fundamentally financial instruments, and the ecosystem is well past due for the establishment of crypto-specific audit and attestation standards.” —David Sacks, March 7, 2025.¹

I Introduction

This study assesses the emerging market for voluntary audits of smart contracts used in Decentralized Finance (DeFi). Smart contracts are self-executing agreements between parties that operate on a blockchain.² Smart contracts have a wide variety of applications, including exchange of assets, lending/borrowing, investment, derivatives, and insurance (Harvey, Ramachandran, and Santoro (2021), Makarov and Schoar (2022), and Harvey and Rabetti (2024)). Smart contracts enable users to engage in peer-to-peer financial transactions using digital assets such as cryptocurrencies without reliance on centralized financial intermediaries, in principle creating a trustworthy and efficient process for executing transactions.³

The explosive growth in the market for digital assets in recent years—as illustrated by the \$4 trillion market capitalization of cryptocurrencies as of October 2025—has made digital assets lucrative targets for cybersecurity exploits (Cong, Harvey, Rabetti, and Wu (2025)), thereby posing security risks for smart contract protocol developers and users. For instance, in August 2021, a hacker breached PolyNetwork—a large DeFi platform that facilitates the movement of digital assets across blockchains—stealing \$600 million in what became one of the largest DeFi hacks to date.⁴ As a result, there has been a commensurate growth in demand for audits of smart contracts to mitigate these security risks. With the growing size and importance of DeFi, protocols are facing an ever-increasing pressure to assure users that their smart contracts will be executed as intended, particularly without exposure to bugs and security breaches, which could result in loss of funds. To provide such assurance, before opening a new protocol to the public by deploying the smart contracts to the blockchain, which is commonly referred to as “protocol launch,” protocol developers have increasingly relied on hired auditors to find errors in smart contract code and potential sources of security breaches. As with traditional financial statement audit reports (e.g., in the context of Initial Public Offerings), smart contract audit releases may increase investors’ and users’ trust (e.g., Bourveau, Brendel, and Schoenfeld (2024); Bhambhwani and Huang (2024); Knechel, Maex, and Park (2025)).

¹Attributed to David Sacks, the “Crypto Czar,” at the White House Crypto Summit regarding the importance of crypto audits. See <https://tinyurl.com/ye2yvxfjm>.

²In Section 2 we provide a detailed explanation of the institutional setting, including blockchain technology, DeFi, and smart contracts.

³For ease of exposition, in what follows, a smart contract protocol refers to a single online platform running smart contracts to provide decentralized financial services. Each smart contract protocol encompasses as few as one and as many as thousands of smart contracts.

⁴The PolyNetwork hack exploited a vulnerability in the protocol’s cross-chain interoperability mechanism. The attacker was able to manipulate smart contracts to illicitly transfer assets across blockchains, ultimately extracting over \$600 million in cryptocurrencies from blockchains such as Ethereum, Binance Smart Chain, and Polygon. Remarkably, the hacker later returned most of the funds, claiming the attack was meant to expose security flaws.

As described in greater detail in Section 2, there are two types of smart contract auditors. The first type, referred to as *centralized auditors*, resembles traditional financial auditors in that they are hired for a fixed fee to conduct a formal code review, usually in a controlled environment before a protocol is launched. Centralized audits are typically one-time engagements that span a few weeks and produce a detailed audit report designed to increase user confidence that the code will execute securely. The second type of auditor, often called *decentralized auditors* or “bounty hunters,” operates through “bug bounty programs.” These programs invite independent security researchers worldwide to test protocols *after launch*, directly in the live production environment. Unlike centralized auditors, bounty hunters are compensated based on the severity of vulnerabilities they uncover, and the programs run continuously rather than at a single point in time.⁵ Because the assurance value of centralized audits can decay over time as protocols evolve after launch, many projects adopt bounty programs as a complementary and ongoing layer of defense.

Despite these efforts, DeFi protocols remain highly vulnerable to exploits, with security breaches usually occurring within the first six months from protocol launch. When hacks occur, attackers typically exploit flaws in contract logic or infrastructure to withdraw or redirect assets without authorization. The immediate losses are borne primarily by protocol users whose deposits are drained or devalued. Protocol treasuries—assets contributed by protocol developers and investors—may also suffer, either directly through theft or indirectly if protocols must compensate affected users. Because blockchain transactions are irreversible, recovery is rare. The magnitudes of such breaches are economically significant: across our sample, the average breach results in losses exceeding \$20 million. In extreme cases, losses can be considerably more, such as the aforementioned \$600 million PolyNetwork exploit or the \$200 million Euler Finance hack.

This study addresses three fundamental questions about the market for smart contract audits. First, what factors do protocol developers consider when deciding whether to have their smart contracts audited and what types of auditors to engage? Relatedly, are centralized and decentralized audits complements or substitutes? In particular, do centralized audits play a certification and reputational role pre-launch, and do decentralized audits play an ongoing monitoring and risk management role post-launch? Second, is having an audit associated with a lower likelihood of future security breaches? Third, what actions do smart contract developers take after security breaches? In particular, we examine both protocol-specific and broader systemic breaches, and analyze whether protocol-specific breaches affect auditor market share.

To address these questions, we construct a dataset covering nearly 10,000 audit reports issued by more than 100 centralized auditing firms and bounty programs for thousands of DeFi protocols launched between January 2020 and January 2025. We collect audit reports and firm information from a variety of sources, including auditing

⁵See Appendix Table A1 for more details of the differences between centralized and decentralized auditors.

firms’ official websites, blockchain security website aggregators, and public GitHub repositories.⁶ After excluding duplicates, incomplete records, and non-audit documents, we retain verified reports that are matched to individual DeFi protocols. We link each audit report or bounty program to detailed protocol-level data pertaining to financial metrics, usage statistics, security breaches, and governance features. We obtain protocol-level financial data from DeFiLlama, token prices from CoinGecko, wallet concentration data from Ethplorer, and development activity from GitHub. We also gather social media engagement metrics from X (formerly Twitter). We supplement this dataset with protocol characteristics that are defined and discussed in Section 3, such as oracle usage, bridge infrastructure, DAO governance, service type, blockchain deployment, and staking, and listing status, obtained from protocol websites, CoinDesk, and other public aggregators. Finally, we compile a database of over 300 DeFi security breaches based on DeFiLlama’s exploit archive that we cross-validate with post-mortems by security firms, protocol disclosures, and industry news outlets.

To address our first research question, we estimate logistic and multinomial models linking protocol characteristics at launch to centralized auditor choice. In doing so, we distinguish between top-tier and bottom-tier auditors to reflect potential variation in audit quality. Top-tier auditors are an exclusive group of highly visible and reputable blockchain security firms—such as *CertiK*, *PeckShield*, *Hacken*, *Halborn*, *Quantstamp*, and *Slowmist*—that dominate the DeFi audit market in terms of market share. By contrast, bottom-tier auditors are smaller, lesser-known firms with limited market presence. Audit adoption is strongly associated with key protocol features. We find that oracle integration exhibits a strong association with the decision to hire auditors of all types. Oracles are pieces of middleware that enable smart contracts to incorporate external data, such as market prices and interest rates. However, their use exposes DeFi protocols and their smart contract users to heightened security risks arising from external data dependencies. Relatedly, protocols that support multiple blockchains—moving assets across or “bridging” also poses security risks—are also more likely to hire an auditor. In addition, protocols are more likely to choose to be audited when they have a larger customer base, as reflected by initial Total Value Locked (TVL)—a DeFi protocol characteristic that is analogous to total deposits at a bank—and when they rely on “staking” a process whereby assets are temporarily locked to incentivize certain actions by protocol users. Protocols are more likely to engage a top-tier auditor when they have a relatively large public presence (e.g., Twitter followers, exchange listings) and when they are more reliant on external funding.

We next examine how auditor choice responds to high-profile systemic security breaches. Building on the idea that protocols with greater *ex ante* complexity, risk exposure, or scale have stronger incentives to seek reputable third-party assurance (Wallace, 1980; DeAngelo, 1981; Watts and Zimmerman, 1983; Cohen, Dey, and Lys, 2008), we hypothesize that industry-wide hack events operate as external shocks that intensify these incentives. When large-

⁶GitHub provides a centralized location to store and manage files and code, enabling collaboration among software developers.

scale exploits expose systemic vulnerabilities—such as the \$600 million PolyNetwork breach—protocols sharing similar risk profiles (e.g., oracle integration, cross-chain deployment, or lending platforms) face heightened scrutiny from users and investors. In this environment, engaging a top-tier auditor or adopting decentralized bounty programs may serve both as a security enhancement and a credible signal of quality.

Using a stacked difference-in-differences framework around six major DeFi hacks, we find that protocols with higher *ex ante* risk exposure are significantly more likely to hire top-tier auditors prior to launch following systemic attacks. The post-event increase in top-tier audit adoption among oracle-integrated protocols is about 5.1 percentage points relative to unaffected protocols, representing roughly a 22% rise over the baseline adoption rate. Lending protocols and cross-chain deployments also exhibit economically meaningful increases of 1–2 percentage points.

We next examine situations under which centralized and decentralized audits act as complements or substitutes. Because centralized audits typically are static, pre-launch certifications, their assurance value may decay over time, making decentralized bug-bounty programs a natural complement post-launch. Consistent with this observation, we find evidence that over 91% of pre-launch audits are conducted by centralized auditors, whereas 77% of post-launch audits are conducted by decentralized auditors. On the other hand, decentralized audits may serve as substitutes for centralized audits. For example, when protocols face a sudden increase in systemic risk, they may substitute away from lower-tier centralized auditors toward bounty hunters. We test this conjecture by showing that systemic breaches raise the likelihood of post-launch engagement of decentralized auditors by 0.7–1.0 percentage points among high-vulnerability protocols, which is equivalent to a 25% increase over their baseline bounty-usage rate.

As with the traditional audit literature (Wallace, 1980; DeAngelo, 1981; Lennox and Pittman, 2011), our evidence in the DeFi market setting suggests that auditors have emerged as mechanisms to assure users and investors. However, unlike traditional audit markets, decentralized auditing has emerged as a new source of assurance, particularly following security breaches. Thus, our findings regarding decentralized auditors parallel evidence that other market mechanisms (e.g., peer/whistleblower) can play significant roles in the presence of centralized auditors (Dyck, Morse, and Zingales, 2010; DeFond and Zhang, 2014).

Next, we turn to our second research question, whether audits mitigate security risks. We find that most DeFi hacks occur early in a protocol's life cycle—nearly 80% within the first six months of launch—highlighting the acute vulnerability of newly deployed protocols. To examine whether audits are associated with a lower incidence of hacks, we estimate logistic regressions in which the dependent variable is an indicator of whether a protocol is hacked and the key explanatory variable is an indicator of whether the protocol underwent a centralized audit before launch. The baseline regressions yield a counterintuitive result: audited protocols—particularly those audited by top-tier centralized firms—are, on average, more likely to experience a hack than unaudited ones. However, realizing that this finding may reflect selection bias: high-risk or more complex protocols may have a larger likelihood

of being hacked not because audits contribute to security breaches but because audits are more likely to be sought when vulnerabilities are anticipated.

Audit choice is inherently endogenous: protocols that are riskier or more complex may be more likely to seek audits and more likely to be hacked. Following guidance to triangulate when identification is imperfect (e.g., [Armstrong, Kepler, Samuels, and Taylor, 2022](#)), we adopt multiple, complementary designs rather than rely on a single specification. On the *extensive margin* (any audit vs. none), we (i) construct a propensity-score-matched sample to balance observables between audited and unaudited protocols, and (ii) implement an instrumental-variables strategy using the FBI’s public service announcement on DeFi exploits as a plausibly exogenous shock to audit demand. On the *intensive margin* (top- vs. low-tier audit), we address selection into audit quality with three approaches: (i) a Heckman two-step correction, (ii) an IV based on GitHub presence in the Post-ChatGPT period (reflecting an exogenous shift in developer tooling), and (iii) a Bartik (shift-share) IV that interacts *pre-period* ecosystem DAO intensity with each protocol’s launch-time blockchain weights. We also study post-launch decentralized assurance by analyzing the adoption of bounty programs, utilizing similar econometric specifications. Across all these designs—and acknowledging that no observational strategy can eliminate fully endogeneity—our evidence indicates: (i) moving from no audit to “any audit” is not reliably associated with lower breach risk; (ii) *top-tier* centralized audits are associated with *lower* breach likelihood; and (iii) decentralized (bounty) audits are likewise associated with reduced breach incidence and severity in the months following adoption.

We now turn to our third research question: how do smart contract developers and auditors respond to security breaches, and what are the consequences for auditor reputation? We find that protocols are more likely to upgrade their auditor after a security breach, particularly by switching from a bottom-tier or unaudited status to a top-tier centralized auditor. This response highlights the assurance role of high-reputation auditors: following negative security events, developers seek to restore trust by engaging providers perceived to offer stronger assurances ([Cohen et al., 2008](#); [Chi, Huang, Liao, and Xie, 2009](#)). We find limited evidence of auditor downgrading or lateral shifts. Protocols are unlikely to replace top-tier auditors with lower-tier ones, suggesting that switching behavior is primarily a mechanism for upgrading rather than reshuffling assurance providers. This pattern is consistent with traditional audit markets, where negative shocks often trigger reputational repositioning rather than random churn ([Cohen et al., 2008](#); [Cook, Kowaleski, Minnis, Sutherland, and Zehms, 2020](#); [Pan, Shroff, and Zhang, 2023](#)).

Moreover, hacked protocols are more likely to implement a program of hiring bounty hunters following the hack. Notably, the probability of implementing a bounty program within six months of a breach rises by approximately 15–20 percentage points, relative to a pre-breach baseline of around 10%. This shift is not limited to previously unaudited protocols; even those with top-tier auditors increasingly engage bounty hunters post-breach. The pattern is consistent with static, pre-deployment audits being insufficient in the face of evolving threats. The

likelihood of bounty adoption is particularly high for protocols with oracle integration, cross-chain deployments, and strong social visibility, suggesting that infrastructure complexity and public exposure amplify the need for additional layers of assurance.

Security breaches have consequences not just for protocols but also for auditors. In traditional audit markets, auditor reputation constitutes a key component of audit quality (DeAngelo, 1981), and failures erode trust and lead to client attrition (Krishnamurthy, Zhou, and Zhou, 2006; Francis, 2011). We find that auditors lose approximately 4.2 percentage points of market share within three months of an audited protocol breach, representing roughly 20% of average share among top-tier firms. This reputational penalty appears to be transient, as we observe no statistically significant decline at six- or twelve-month horizons. Breach severity also matters: doubling of hack-related financial losses is associated with a 0.2 percentage point reduction in the auditor’s short-term market share, representing roughly 1% of the baseline for top-tier auditors. These results suggest that perceptions of auditor quality are updated not only as a function of the occurrence of breaches but also of their severity (Gipper, Leuz, and Maffett, 2020).

To complement our empirical analysis of post-hack dynamics, we present a case study of the PolyNetwork hack, which illustrates how crisis response strategies influence top-tier auditor reputation management. CertiK’s rapid, detailed forensic reporting enabled a quick rebound, PeckShield’s high-profile social media updates yielded only temporary gains, and Hacken’s low-profile approach aggravated market share decline—highlighting that the intensity and visibility of the response are closely linked to effective reputation management in the DeFi audit market.

Our study contributes to a growing literature on smart contract auditing in decentralized finance (DeFi). Recent work documents that audit disclosures convey value-relevant information and enhance protocol outcomes. Bourveau et al. (2024) show that audit announcements elicit positive token return reactions. Rabetti (2023) and Bhambhwani and Huang (2024) find that audits function as a form of certification, boosting investor confidence and increasing total value locked (TVL) at launch. Knechel et al. (2025) provide complementary evidence that audit adoption is positively associated with subsequent TVL growth. These studies focus on the existence and signaling value of audits; our paper advances this literature in several directions.

First, we examine both the static and dynamic determinants of audit choice, documenting how protocol characteristics (e.g., oracle usage, bridging infrastructure, cross-chain deployment, and governance design) and external shocks (e.g., systemic security breaches) shape audit adoption patterns. Second, we are the first to study the rise of decentralized auditing and its effectiveness in the post-deployment (live) window. Third, we study the post-breach aftermath from both the protocol and auditor perspectives, examining how protocols respond to security incidents—auditor switching and bounty engagement—and whether auditors face short-run and longer-run rep-

utational penalties in the form of market share losses.

Our study also contributes to the literature on voluntary auditing, which shows that firms opting into audits tend to benefit from improved outcomes, such as higher credit ratings and better financing terms (e.g., [Allee and Yohn, 2009](#); [Minnis, 2011](#); [Lennox and Pittman, 2011](#); [Minnis and Shroff, 2017](#); [Lisowsky and Minnis, 2020](#); [Schoenfeld, 2024](#); [Lennox, Schmidt, and Thompson, 2023](#)). Similarly, prior studies in the DeFi literature find that voluntary smart contract audits are associated with increased adoption of platforms and greater platform engagement. Our study extends this literature by examining whether smart contract audits are associated with a lower incidence of security breaches.

Our paper also contributes to the literature on audit quality and auditor choice (e.g., [Simunic, 1980](#); [Watts and Zimmerman, 1983](#); [DeFond and Subramanyam, 1998](#); [Lennox and Pittman, 2010](#); [Reichelt and Wang, 2010](#); [Kaplan and Williams, 2013](#); [DeFond, Erkens, and Zhang, 2017](#); [Lobo, Paugam, Zhang, and Casta, 2017](#); [Duguay, Minnis, and Sutherland, 2020](#); [Fedyk, Hodson, Khimich, and Fedyk, 2022](#)), which generally finds that high-reputation auditors deliver more reliable assurance. In traditional markets, “Big 4” auditors are linked to lower rates of misreporting, restatements, and fraud (e.g., [Magee and Tseng, 1990](#); [Antle and Nalebuff, 1991](#); [Dye, 1991](#); [Mutchler and McKeown, 1997](#); [Blacconiere and DeFond, 1997](#); [Francis, Maydew, and Sparks, 1999](#); [Neal and Riley Jr., 2004](#); [Gul, Fung, and Jaggi, 2009](#); [Kausar, Shroff, and White, 2016](#); [Ashraf, Michas, and Russomanno, 2020](#)). Our findings extend the literature by documenting how audit quality, selection, and switching dynamics operate in the multi-trillion-dollar market that is largely decentralized and unregulated.

The remainder of the paper is organized as follows. Section 2 contains background information on blockchain technology, decentralized finance, smart contracts, and centralized and decentralized audits and auditors. Section 3 discusses our data sources, sample, and summary statistics, and develops hypotheses. Section 4 examines factors associated with the decision by a protocol to have its smart contracts audited and auditor choice. Section 5 examines the effectiveness of smart contract audits in preventing future security breaches. In Section 6, we analyze responses of DeFi protocols to security breaches and the effects of these responses on auditors’ reputation. Section 7 concludes and proposes directions for future research in the DeFi auditing space.

2 Setting and Data

2.1 Blockchain technology

Blockchain technology represents a foundational innovation in the way digital information is recorded, verified, and shared. At its core, a blockchain is a decentralized ledger distributed across a network of computers (nodes). Transactions are grouped into cryptographically linked blocks, which are validated by network participants through

consensus mechanisms (e.g., proof of work or proof of stake). Once added to the chain, a block's contents are effectively immutable and publicly verifiable, ensuring the integrity of the transaction history.

Unlike traditional centralized databases maintained by a single authority, blockchain systems are designed to operate without a central point of control. This decentralization offers several potential advantages: resistance to censorship, improved data transparency, and enhanced security through distributed consensus. These properties make blockchain particularly well-suited to environments where trust is limited or where intermediaries are either inefficient or unnecessary.

Although the technology was originally introduced in 2009 as the infrastructure for Bitcoin, a peer-to-peer digital currency, the scope of blockchain applications has since expanded dramatically. Beyond simple value transfer, modern blockchain platforms support the recording of complex transactional data, ownership records, and multi-party interactions. This has enabled its adoption in a wide range of sectors, including supply chain logistics, digital identity management, and intellectual property.

One of the most transformative applications of blockchain has emerged in the financial sector. Over the past few years, a new ecosystem known as Decentralized Finance (DeFi) has grown rapidly, aiming to improve upon—and in some cases, replace—traditional financial services through decentralized, blockchain-based infrastructure. The following subsection introduces the DeFi ecosystem and examines how it is reshaping the delivery and design of financial services globally.

2.2 Decentralized finance (DeFi)

Decentralized Finance (DeFi) refers to a rapidly evolving ecosystem of financial services built on programmable public blockchain networks, such as Ethereum and Solana, among many others. Unlike traditional financial systems, which rely on centralized institutions such as banks, clearinghouses, and regulators to mediate and enforce transactions, DeFi systems operate through open, distributed networks that allow users to interact directly with financial protocols. These protocols provide a range of financial services—such as lending, borrowing, trading, investing, and insurance—without requiring permission from centralized authorities or intermediaries. By leveraging blockchain's core features of transparency, immutability, and global accessibility, DeFi aims to democratize access to financial tools and infrastructure.

The range of DeFi applications continues to expand. Lending and borrowing platforms allow users to deposit digital assets to earn interest or to borrow against their existing holdings. Decentralized exchanges (DEXs) facilitate peer-to-peer trading of cryptocurrencies, eliminating the need for a centralized exchange or custodian. Many of these platforms use automated market makers (AMMs), which rely on algorithmic pricing mechanisms and liquidity pools rather than traditional order books. Stablecoins—cryptocurrencies pegged to fiat currencies such as the

US dollar—are another foundational component of DeFi, offering a relatively stable medium of exchange within volatile crypto markets and enabling more predictable pricing for services and assets.⁷

Beyond these functions, DeFi also encompasses the creation of synthetic assets and derivatives, which allow users to gain on-chain exposure to the price movements of a wide variety of financial instruments, including equities, commodities, and fiat currencies. Insurance protocols are being developed to offer protection against specific risks such as protocol failures or price manipulation, providing coverage mechanisms that are both capital-efficient and community-governed. Additionally, decentralized asset management tools and yield optimization platforms are emerging to support users in deploying capital across a growing array of investment strategies and products.

In comparison to traditional financial systems, DeFi offers several potential advantages. Most prominently, it is inherently open and inclusive: participation does not depend on geographic location, legal status, or creditworthiness, but instead merely requires access to the internet and a digital wallet. This has the potential to expand financial access, particularly in regions underserved by existing banking infrastructure. Furthermore, DeFi protocols are transparent by design, with transaction histories and system parameters that are publicly accessible and verifiable on-chain. DeFi also benefits from “composability,” which allows various protocols and services to be integrated or “stacked” as modular components in a broader financial ecosystem. This architectural flexibility has accelerated innovation and facilitated the creation of increasingly sophisticated financial products. Finally, by removing the need for intermediaries, DeFi systems may offer lower fees, faster settlement times, and improved efficiency for users.

At the core of the DeFi ecosystem are DeFi protocols, which are autonomous sets of rules governing financial transactions and services deployed on blockchain networks. These protocols typically define the logic and parameters under which assets can be transferred, pooled, lent, or exchanged within the system. Although each protocol may serve a distinct purpose—such as liquidity provision, credit issuance, or asset custody—they share common traits: they are non-custodial, open-source, and often governed by decentralized communities. DeFi protocols are designed to operate continuously and without interruption, executing financial logic directly on-chain and often interacting with other protocols in a composable manner. Their open and programmable nature facilitates experimentation, innovation, and the rapid proliferation of new financial models that diverge from conventional regulatory and operational paradigms. The following subsection discusses the computational mechanisms underlying these decentralized systems, focusing on the role of smart contracts in defining and automating protocol behavior.

⁷See [Hasbrouck, Rivera, and Saleh \(2022\)](#), [Malinova and Park \(2024\)](#), [Milionis, Moallemi, Roughgarden, and Zhang \(2024\)](#), [Capponi and Jia \(2025\)](#), [Lehar and Parlour \(2025\)](#), [Lyandres and Zaidelson \(2025\)](#), and [Tovanich, Kassoul, Weidenholzer, and Prat \(2025\)](#) for examples of theoretical and empirical analyses of various types of DeFi activities.

2.3 Smart contracts

Smart contracts are self-executing agreements whose terms are directly written into code. They automatically perform predefined actions when specific conditions are met. One of the key advantages of smart contracts lies in their ability to automate and streamline a wide range of transactions and agreements, potentially revolutionizing traditional contract mechanisms across various industries. Once a smart contract is deployed on a public blockchain, its code and execution typically cannot be modified, ensuring the integrity of the agreed-upon terms. All transactions and interactions with smart contracts are recorded on the blockchain, allowing for transparent verification of (financial) activities. Smart contracts enable automated transactions that do not rely on compliance by the counterparties, i.e. “trustless” transactions.

Despite the potential of smart contracts to change the landscape of financial arrangements, there are legitimate concerns about potential security breaches and vulnerabilities within the smart contract code. Because of the irreversible and automated nature of smart contract execution, any flaws in smart contract code can have severe consequences, leading to financial losses or exploitation by malicious actors.

Smart contracts have both advantages and disadvantages relative to traditional financial contracts. Because smart contracts do not rely on centralized intermediaries, organizational overhead is lower, reducing transaction costs. Smart contracts deployed on public blockchains are open-source, leading to intense competition and fast development of smart-contract-based financial applications. Competition among platforms is intense because of low entry costs, which result from the open-source nature of smart contracts. DeFi is characterized by an unprecedented degree of interoperability and interconnectedness (Cong, Prasad, and Rabetti (2023)), at least within a given blockchain, allowing sophisticated financial engineering and efficient utilization of liquidity.⁸

One of the largest drawbacks of smart-contract-based DeFi protocols is risks unique to them, referred to as “smart contract risks” hereafter. Because smart contracts can be deployed permissionlessly on a public blockchain, i.e., without control or oversight, sometimes by inexperienced professionals, adventurous enthusiasts, and ambitious amateurs, they may contain logical errors and/or bugs, exposing their users to potential losses of funds as well as funds appropriation by bad actors.⁹ As of the end of 2023, \$7.6 billion in user funds have reportedly been lost to smart contract exploits and hacks.¹⁰

As an illustration of smart contract risk, consider an exploit of Euler Finance, one of the largest lending platforms in DeFi, that allows users to earn interest on deposits and borrow assets by using their crypto holdings as

⁸In the context of DeFi, interoperability refers to the ability of distinct smart contracts to communicate and interact with one another seamlessly. Interconnectedness—the degree to which various protocols are linked or integrated, allowing for transactions spanning several protocols—is an outcome of interoperability.

⁹See Harvey and Rabetti (2024) for a discussion of the advantages and risks of DeFi adoption.

¹⁰See <https://defillama.com/hacks>.

collateral, all governed by smart contracts. The platform’s unique feature is its customizable lending markets, enabling users to create custom collateral types and risk parameters. In March 2023, Euler Finance suffered a \$200 million loss when an attacker exploited vulnerabilities in the protocol’s smart contract, specifically in the liquidation and borrowing mechanisms. By using a flash loan, the attacker manipulated collateral ratios, triggering faulty liquidations and draining the protocol’s funds. This incident highlights significant risks in DeFi protocols, particularly around smart contract vulnerabilities and the use of flash loans for exploiting weaknesses.¹¹

Given the rapid pace of change in smart contract technology and the high degree of exposure of smart contracts to external threats, it is generally impossible to ensure that smart contracts are error-free. In the extreme, smart contract risks could severely restrict the usefulness of transacting in DeFi. Fortunately, the market has developed a mechanism for mitigating smart contract risk: smart contract audits, which involve reviewing smart contract code to identify and rectify bugs and vulnerabilities.

2.4 Smart Contract Audits

As with traditional financial-statement audits, smart contract audits provide third-party certification, and audit fees generally are independent of the report’s outcome. However, whereas traditional audits evaluate whether financial statements are prepared in accordance with applicable accounting standards and whether the financial statements present fairly a firm’s financial position and operations, smart contract audits assess the integrity and security of on-chain code (e.g., logic correctness, privilege boundaries, upgradeability, and interactions with external contracts such as oracles and bridges). Methodologically, financial audits rely on risk-based planning, tests of controls, and substantive procedures over transactions and balances, while smart contract audits emphasize code analysis, property-based testing, differential testing across upgrades, formal verification where feasible, and adversarial review of attack surfaces.

Whereas traditional audits are conducted under established auditing standards (e.g., GAAS/ISA) enforced by regulators and professional bodies, with auditor licensure, continuing education, independence requirements, and well-defined liability regimes, smart contract auditing currently lacks a universally accepted standard-setting framework; there is no formal licensure requirement, and practitioners typically possess expertise in computer science, cryptography, and verification rather than accounting credentials. Mandates and disclosure practice differ as well: financial statement audits often are legally required for reporting entities, whereas smart contract audits—and their public disclosure—are typically voluntary.¹² Finally, the timing and deliverables reflect distinct goals: financial au-

¹¹See <https://tinyurl.com/yccwztuf> for a discussion of the Euler Finance exploit.

¹²Although most audits are disclosed publicly by either the protocol or the auditing firm, some—especially for smaller protocols—may remain private (e.g., [Feng, Hitsch, Qin, Gervais, Wattenhofer, Yao, and Wang \(2023\)](#)). See [Yuyama, Katayama, and Brigner \(2023\)](#) for a proposal for principles of DeFi disclosure.

auditors opine ex post on an annual reporting package, while smart contract audits are frequently undertaken pre-deployment or around major upgrades, produce issue-level findings with severity classifications and remediation notes rather than a standardized audit opinion, and may be re-run iteratively as code evolves.

Smart contract auditors provide independent assurance of the completeness and correctness of smart contract code. The auditing process of smart contracts begins with a comprehensive code review. This step involves examining smart contract code to identify vulnerabilities, potential exploits, and bugs. Auditors also check whether the smart contract code adheres to best coding practices and principles and correctly implements the intended functionality, i.e., it does what it is supposed to do.

After smart contract auditors complete their code review, they typically proceed to “static analysis.” This involves examining the code without actually executing it—a process that can help identify common coding errors and potential vulnerabilities that may have been missed during the code review. Static analysis tools are designed to scrutinize the code, parsing each line and structure. Smart contract auditors often follow up with a “dynamic analysis,” which involves running the code in a controlled environment to observe its behavior. Dynamic analysis helps auditors identify vulnerabilities that might not be detectable through static analysis alone, such as runtime errors or issues with memory management.

Economic analysis forms another crucial part of smart contract auditing, in which an auditor considers economic incentives and mechanisms of users’ interactions with smart contracts. For instance, auditors may analyze how a lending protocol sets interest rates, and how lenders within the protocol contribute to the protocol’s stability. Auditors may scrutinize collateralization mechanisms embedded in smart contracts to assess their effectiveness in mitigating default risks. Overall, economic analysis helps to ensure that smart contract logic does not create opportunities for manipulation or abuse, thereby safeguarding the interests of protocol users.

After protocol developers address the issues identified during the smart contract audit, an auditor conducts re-testing. This step may be repeated several times before the final audit report is released. Overall, the auditing process tends to be time-consuming, often spanning weeks and sometimes months.

The market for smart contract audits has grown substantially in recent years, expanding from just a handful of auditors in 2020 to over a hundred by August 2025. In some respects, the development of decentralized finance (DeFi) mirrors the early evolution of traditional banking, when formal regulation was limited (e.g., [Frishkoff \(1989\)](#); [Bourveau, Breuer, Koenraadt, and Stoumbos \(2025\)](#)). In both contexts, a new form of financial intermediation emerged, creating a need for mechanisms that could instill trust among market participants.¹³

¹³Early solutions included the voluntary provision of financial statements, third-party audits, and state-level supervision. Although these mechanisms initially brought some stability, repeated episodes of bank runs and financial panics eventually led to federal intervention, beginning with the Federal Reserve Act of 1913.

Whether DeFi follows a similar trajectory remains to be seen. The voluntary nature of smart contract audits today is analogous to the low-regulation, early auditing practices in traditional finance during the early 20th century. A key distinction, however, is that traditional audits are largely retrospective, while smart contract audits are inherently forward-looking, aiming to detect and prevent vulnerabilities before protocols go live. Yet, because centralized audits occur before deployment, it remains an open question whether they are sufficient to prevent security breaches, or if post-deployment mechanisms—such as decentralized audits—are also essential.

2.5 Examples of Centralized Smart Contract Auditors

To provide context on the state of the smart contract auditing industry, this subsection describes the largest auditors in our sample, which together were responsible for auditing approximately 17 to 53 percent of all smart contracts, depending on the period within our sample.

Certik. Headquartered in New York, Certik was founded by professors from Columbia and Yale and funded by several leading venture capital funds (see <https://www.certik.com/>). Certik is known for using several auditing processes, including automated auditing and formal verification methods. It claims to be the first auditor of smart contracts that has received SOC II certification—a security standard that offers guidelines to service organizations for the protection of sensitive data from unauthorized access, security incidents, and other vulnerabilities.

Hacken. Headquartered in Estonia, Hacken is an international cybersecurity company with Ukrainian roots (see <https://hacken.io/audits/>). Hacken has over 1,000 global clients, including large and successful blockchains, crypto exchanges, and DeFi protocols such as Binance, Avalanche, Kyber Network, Huobi, Kucoin, Sandbox, and Maker. Besides auditing smart contracts, Hacken provides other services, including tokenomics, penetration tests that simulate cyberattacks to identify vulnerabilities, and a real-time smart contract protection tool. Appendix B provides an example of an audit report conducted by Hacken.

PeckShield. Located in Hangzhou, China, PeckShield was formed by seasoned security professionals and senior researchers from companies such as Microsoft, Intel, Juniper, and Alibaba (see <https://peckshield.com/>). Founded in 2018, PeckShield also offers blockchain forensics services (e.g., transaction mapping and real-time blockchain monitoring). PeckShield has audited several large market players in the DeFi space, including BNB Chain, Polygon, EOS, Maker, Aave, DydX, Bancor, and Rinch.

Halborn. Headquartered in Miami, Florida, Halborn is a blockchain cybersecurity firm founded in 2019 by cybersecurity experts with backgrounds in corporate security and offensive penetration testing (see <https://www.halborn.com/>). Halborn is known for combining manual code reviews with advanced adversarial testing, offering security assessments for smart contracts, wallets, and blockchain infrastructure. Its clients include major DeFi protocols and exchanges such as ApeCoin, Polygon, Avalanche, and Coinbase. Halborn also develops educational

content and public security advisories that help disseminate best practices in smart contract safety.

Quantstamp. Founded in 2017 and based in San Francisco, Quantstamp emerged from the Y Combinator accelerator program and focuses on scalable smart contract verification (see <https://quantstamp.com/>). Quantstamp provides both automated and manual audits, as well as formal verification for high-value smart contracts. In addition to auditing, Quantstamp contributes to blockchain security standards, has conducted audits for major projects such as Ethereum 2.0, Solana, Chainlink, and Binance, and claims to be working closely with regulators worldwide.

SlowMist. Founded in 2018 and headquartered in Xiamen, China, SlowMist is a blockchain security firm that offers auditing, threat intelligence, and incident response services (see <https://www.slowmist.com/>). SlowMist is known for its comprehensive vulnerability detection frameworks and real-time threat monitoring through its MistTrack system, which traces stolen funds across chains. It has audited numerous leading protocols and exchanges, including Huobi, OKX, EOS, and PancakeSwap, and actively contributes to public security reports on major DeFi exploits and phishing schemes.

2.6 Decentralized Smart Contract Audits

The rise of decentralized applications and the growth of the DeFi ecosystem have significantly increased demand for robust smart contract security. In response to the limitations of traditional code auditing models—particularly their high cost, fixed capacity, and opaque methodologies—a parallel market has emerged: decentralized smart contract auditing, often organized around a bounty-driven model. In this model, developers or protocol teams post open calls for review, offering monetary rewards (bounties) to independent security researchers, who are often referred to as “bounty hunters,” and who identify and disclose vulnerabilities in the protocol’s codebase.

Several platforms that facilitate interactions between smart contract developers and bounty hunters have emerged in recent years.¹⁴ These platforms provide a set of incentives and reputational systems to encourage high-quality, responsible disclosures. Unlike traditional audits, which rely on a fixed team of experts delivering a formal report, bounty-based audits engage a broader, more diverse pool of talent, and are often composed of freelance security researchers and software engineers with domain-specific expertise. In some cases, these bounty hunters may identify bugs that are missed by centralized audit firms. Bounty markets offer a scalable, flexible, and incentive-aligned mechanism for identifying vulnerabilities, particularly where continuous security assurance is necessary.¹⁵

Despite their advantages, decentralized auditing systems are not a complete substitute for centralized audits. Centralized auditing firms typically employ standardized review methodologies, perform formal verification, and

¹⁴Two prominent marketplaces for bounty hunters are: <https://code4rena.com/> and <https://immunefi.com/>.

¹⁵See Appendix C for an extended discussion of bounty hunters.

deliver structured reports that serve both technical and regulatory functions. These audits often provide reputational signaling for institutional partners, investors, and users. In contrast, decentralized auditing offers less formal assurance, often lacks rigorous documentation, and may vary significantly in quality depending on the skill of the individual auditors and the incentive structure of the bounty program.

3 Data, Summary Statistics, and Hypotheses

3.1 Data

We assemble a protocol-level panel that links audit activity to technical design, market usage, and security outcomes. We collect audit reports from three sources: (i) official repositories and “clients” pages of commercial auditing firms; (ii) curated blockchain-security aggregators that index published reports and disclosures; and (iii) public codebases (e.g., GitHub) where teams and auditors host artifacts such as PDFs and markdown attestations. We eliminate duplicate reports, remove incomplete or unverifiable documents, and exclude items issued by non-auditing entities. The resulting sample covers the January 2020—January 2025 period and includes several thousand distinct audits authored by more than one hundred firms.

We match each audit report to protocol-level financial, operational, and governance data using multiple sources. We gather protocol financial and usage metrics from DeFiLlama (<https://defillama.com>), token prices from Coin Gecko (<https://www.coingecko.com>), and wallet data and blockchain transactions from Etherscan (<https://etherscan.io>). We extract code development activity from GitHub (<https://github.com>) and social media engagement from X (<https://x.com>). These sources allow us to capture both on-chain fundamentals and off-chain transparency indicators (e.g., Lyandres, Palazzo, and Rabetti (2022); Cong, Landsman, Maydew, and Rabetti (2023); Gefen, Rabetti, Sun, and Zhang (2024); Amiram, Lyandres, and Rabetti (2025)). Additionally, we collect detailed information on key protocol characteristics, including the use of oracles or bridging infrastructure, governance design (e.g., DAO), primary service type (DEX, lending, or yield), blockchain deployment (including cross-chain support), staking, and listing status.¹⁶ We collect this information directly from protocol websites, specialized news providers (e.g., CoinDesk), and data aggregators (e.g., DeFiLlama).

Finally, we assemble a dataset of more than 300 DeFi hack incidents occurring between January 2021 and June 2025. The primary source is DeFiLlama’s exploit archive, which we complement with cross-verified data from security firm post-mortems, official protocol disclosures, and reputable news outlets (e.g., CoinDesk). For each

¹⁶Staking refers to the practice of locking up tokens in a protocol—often to help secure the network, validate transactions, or support governance—in exchange for rewards such as additional tokens or protocol fees. Listing refers to whether a DeFi protocol has tokens listed on a crypto exchange.

exploit, we record the protocol name, event date, attack vector classification, total losses, exploited infrastructure type, and affected blockchains. Additional details and taxonomies are provided in appendices.

3.2 Summary Statistics

Table 1 presents summary statistics of auditor choices, protocol, industry and blockchain characteristics, and hack outcomes.

[Table 1 here]

Approximately 46% of the 4,108 protocols in our sample engaged an auditor (*AUDIT* mean = 0.46): 23% hired a top-tier centralized auditor (*TOP*), 24% hired a bottom-tier auditor (*BOTTOM*).¹⁷ In addition, 4% relied on decentralized bounty-based auditors (*BOUNTY*) before protocol launch. We define a top-tier auditor as one whose market share—measured by the number of audited protocols relative to the total protocols launched in the previous six months—ranks among the top three largest. This approach captures the most prominent auditors in each period, initially dominated by leading firms such as CertiK, PeckShield, and Hacken, and later including established newcomers such as Halborn, Quantstamp, and SlowMist. These trends indicate that while the majority of protocols adopt some form of centralized audit, the uptake of decentralized auditing mechanisms remains limited throughout the sample period.

Approximately 18% of protocols integrate Oracle services (*ORACLE*), while 11% are listed on a cryptocurrency exchange (*LISTED*) before deploying their smart contracts on the mainnet. The average protocol has a log total value locked (*log_TVL*) of 12.14 in one day post-launch, but large protocols reach more than 3.3 million dollars in the same window. DAO-based governance is relatively rare, with only 8% of protocols adopting this structure before launch (*DAO*). Roughly 7% maintain public code repositories on GitHub (*GITHUB*), and, conditional on being open source, development activity (*log_Commits*) averages 8.27, or approximately 17,510.16 commits, indicating intense code activity before launch for open-sourced protocols. Multichain deployment is common, with *log_Chains* averaging 0.89 (about 1.5 chains). Conditional on raising funds before smart contract deployment, the average fundraising (*log_Raised*) is (14.93), or 6.03 million dollars, yet only 5 percent of the protocols succeed in raising funds before launch (*HAS_RAISED* = 0.05).¹⁸ The average log staking measure (*log_Staking*) is 9.71, or more than 16.6 million dollars, implying limited but non-negligible staking participation; but, only 15.5% of the DeFi protocols have staking programs at launch. Finally, social media engagement varies substantially, with protocols averaging 6.18 in log followers (approximately 480 followers), with the top quartile having more than 11,000 followers before launch.

¹⁷We denote binary variables by capitalizing all letters, and denote continuous variables capitalizing only the first letter.

¹⁸See <https://defillama.com/raises>.

Lending protocols (*IND_LENDING*), such as Compound, represent 11% of the sample, decentralized exchanges (*IND_DEXES*), such as Uniswap, comprise 34%, and yield aggregators (*IND_YIELD*), such as Yearn Finance, account for another 11%. The remaining 44% fall into other categories (*IND_OTHERS*). This distribution reflects a balanced mix across major DeFi sectors. These categories are mutually exclusive and we use them as industry fixed effects in our empirical specifications.¹⁹ Additionally, about 10% of protocols are exclusively deployed on Ethereum (*BC_ETHEREUM*), whereas 25% support cross-chains (*BC_CROSSCHAIN*). *log_EthereumTVL*, our proxy for the market activity, averages 24.49, or 43.2 billion dollars, setting it not only as the first but also the largest blockchain for DeFi activity during the period.²⁰ Finally, post-launch vulnerabilities are rare but material when they occur: only 4% of protocols experience a hack within six months of launch (*HACKDUM*), while losses average \$1.07 million (*Hackloss*), with substantial dispersion (standard deviation = 16.48) of 14.3 million dollars.

3.3 Hypothesis Development

We structure our analysis around three overarching research questions that map directly into our hypotheses. First, *what factors are associated with the centralized auditing adoption decision and the type of auditor selected before platform launch?* This includes static determinants—*ex ante* protocol complexity, scale, and risk exposure—and dynamic determinants—how industry-wide systemic breaches shift audit demand (H1a, H1b, H1c). Second, *does smart contract auditing mitigate security breaches?* H2 predicts that undergoing an audit, particularly by a top-tier or decentralized auditor, reduces the likelihood and severity of future hacks. Third, *how do developers and auditors respond to security incidents?* H3a examines whether hacked protocols are more likely to switch auditors or adopt decentralized bounty programs, whereas H3b tests whether breaches erode auditor reputation and market share. Together, these questions link established auditing theory to the decentralized finance setting, enabling us to study both the demand for assurance and the consequences of audit quality in a rapidly evolving market.

H1a (Static determinants of audit choice). *Protocol risks influence the decision to obtain an audit and the type of auditor selected.* Specifically, protocols with greater *ex ante* complexity, risk exposure, or scale are expected to be more likely to engage an external smart contract auditor, particularly a top-tier firm. Classic voluntary auditing theory posits that firms seek audits to mitigate information asymmetry and perceived risk (Wallace, 1980). In a DeFi setting, projects handling substantial user funds or deploying complex code (e.g., lending protocols with liquidation logic) have stronger incentives to seek third-party assurance to certify security and limit vulnerability (Smith and Castonguay, 2020). Engaging a reputable auditor serves as a credible signal of security quality (DeAngelo, 1981; Watts and Zimmerman, 1983), aligning with signaling theory according to which assurance substitutes for missing

¹⁹We use industry definitions as per Defillama.com. See Appendix D for extended explanation.

²⁰We use Ethereum TVL for two reasons: (i) Ethereum is the largest and oldest blockchain for DeFi development; (ii) TVL from other chains often presents measurement issues (Parlour (2023)).

legal enforcement (Cohen et al., 2008). Thus, static features such as reliance on oracles, cross-chain deployment, and protocol size should be positively related to audit adoption and to the selection of top-tier audit services.

H1b (Dynamic determinants of audit choice). *Industry-wide security shocks alter auditing demand among new protocols.* We hypothesize that major hack events in the DeFi ecosystem (e.g., large-scale exploits exposing systemic vulnerabilities) spur a shift toward more frequent or higher-quality audits by subsequently launched protocols, particularly those sharing similar risk characteristics. Prior work shows that negative shocks and scandals in traditional markets increase audit scrutiny and lead to reassessment of risk management practices (Cohen et al., 2008; Francis, 2004). Similarly, when a prominent hack reveals flaws in oracles or cross-chain bridges, new protocols exposed to those weaknesses are expected to adopt top-tier auditing services or additional security layers preemptively.

H1c (Decentralized vs. centralized audits: complements or substitutes). *Usually, centralized and decentralized audits are complements, but after sudden increases in systemic risk, they may be substitutes.* Centralized audits are typically static, pre-launch certifications. However, their assurance value may decay over time, making decentralized bug-bounty programs a natural complement post-launch. Thus, decentralized audits may serve as post-launch complements to centralized audits by providing continuous, live monitoring. On the other hand, decentralized audits may serve as substitutes for centralized audits, particularly when protocols face a sudden increase in systemic risk, such as that following the PolyNetwork breach. In such circumstances, protocols may shift away from lower-tier centralized auditors toward decentralized ones.

H2 (Audit and security breaches). *Undergoing a smart contract audit is expected to mitigate future security breaches.* An independent audit functions as a monitoring mechanism, identifying and remediating vulnerabilities before deployment. Analogous to evidence in financial reporting, where audit quality reduces the likelihood of misstatements or fraud (Francis, 2011; DeFond and Zhang, 2014; Luo, Rabetti, and Yu, 2024), rigorous code audits should lower the probability of hacks and exploits post-launch. Moreover, third-party certification can deter opportunistic attacks by signaling stronger security (Watts and Zimmerman, 1983). We therefore predict that audited protocols experience fewer and less severe hacks relative to unaudited counterparts.

H3a (Post-breach auditor switching). *Protocols that suffer a security breach are expected to be more likely to change their auditor or audit strategy afterward.* Following a hack, projects have incentives to restore credibility by switching from a low-tier to a top-tier auditor or adopting supplemental measures such as bounty programs. This mirrors behavior in traditional markets, where clients replace auditors after failures to signal improved governance (Cohen et al., 2008; Chi et al., 2009). Thus, we hypothesize that security breaches significantly increase the likelihood of an auditor switching or adding decentralized auditing services.

H3b (Auditor reputation after breaches). *Security breaches are expected to impair the reputation of the auditor involved.* Auditor reputation constitutes a key component of audit quality (DeAngelo, 1981). In financial markets, audit failures erode trust and lead to client attrition (Krishnamurthy et al., 2006; Francis, 2011; Cook et al., 2020). Extending this to DeFi, a hack involving a previously audited protocol is expected to damage the auditor’s standing, resulting in reduced future market share. This reputational penalty reflects the heightened sensitivity of decentralized ecosystems to perceived security failures, given the absence of formal enforcement mechanisms.

4 H1: What Factors are Associated with Smart Contract Auditing?

Auditing in decentralized finance is a highly heterogeneous and unregulated activity. It spans two main categories: centralized auditing firms and decentralized auditing mechanisms. Although centralized auditors conduct formal pre-deployment code reviews and issue audit reports to signal security readiness, decentralized audit approaches often emerge after launch in response to heightened security risks, new technology integrations, or vulnerabilities uncovered through internal assessments. Such decisions reflect each protocol’s internal evaluation of security threats, codebase complexity, and the value of reputational signaling.

4.1 H1a: Determinants of Auditor Choice

We address our first research question by analyzing a protocol’s audit choice as a function of its characteristics at launch. These include observable features related to protocol architecture, business classification, deployment scope, development transparency, and prevailing market conditions. To do this, we estimate the following baseline logit model:

$$\text{Auditor}_i = \beta' \text{protocolFeatures}_i + \gamma_j + \lambda_k + \delta_t + \varepsilon_i, \quad (1)$$

Auditor_i denotes a binary indicator equaling one if protocol i hired at least one auditor (*AUDIT*) before launch, and zero otherwise. The vector $\text{protocolFeatures}_i$ includes a rich set of protocol-level characteristics, as described in Section 3.2. γ_j are industry fixed effects (e.g., DEX, Lending, Yield), λ_k are blockchain fixed effects, and δ_t are year fixed effects. All standard errors are clustered by industry \times blockchain.

A limitation of the binary logit approach is that it treats the reference category as a pooled group that includes both protocols with no audit and those that selected an alternative auditor type (e.g., pooling *BOTTOM* auditors and no auditor when examining the determinants of hiring a *TOP* auditor). This conflation may bias coefficient estimates and obscure substitution patterns between top-tier and bottom-tier auditors, as these options are likely

influenced by similar risk factors but represent distinct strategic choices. To account for the substitutability between top- and bottom-tier centralized audits, we also estimate a multinomial logit model:

$$\Pr(\text{AuditorType}_i = c \mid \mathbf{X}_i) = \frac{\exp(\beta'_c \mathbf{protocolFeatures}_i + \gamma_{jc} + \lambda_{kc} + \delta_{ic})}{\sum_{m \in C} \exp(\beta'_m \mathbf{protocolFeatures}_i + \gamma_{jm} + \lambda_{km} + \delta_{im})} \quad \text{for } c \in C, \quad (2)$$

where $C = \{None, Bottom, Top\}$ and the baseline category is *None*. We consider two categories of auditor type: (i) Top-tier centralized auditors (*TOP*)—market leaders with strong reputational capital and industry recognition; and (ii) Bottom-tier centralized auditors (*BOTTOM*)—smaller firms with limited visibility and client bases. The variable *TOP* equals one if a top-tier firm audited the protocol’s contracts before launch.²¹ The indicator *BOTTOM* equals one if such a firm audited the protocol before launch. These classifications enable us to analyze not only how protocol characteristics are associated with the decision to hire an auditor, but also how these characteristics are associated with the decision to hire a top-tier or bottom-tier auditor.

Table 2 reports results from estimating both binary logit and multinomial logit models, i.e., equations (1) and (2). We discuss the main determinants in the order they appear in the table.

[Table 2 here]

Oracle integration emerges as the most significant and consistent determinant of audit adoption across all specifications. In particular, the *ORACLE* coefficient in the logit specifications is significantly positive (coefficient = 0.770). In the multinational specifications, the coefficients of 0.721 and 0.797 indicate that, compared to non-oracle protocols, oracle-integrated protocols are 6.8 percentage points more likely to adopt top-tier audits and 9.0 percentage points more likely to adopt bottom-tier audits, relative to not adopting any audit.²² This pattern reflects the elevated vulnerability that oracles introduce by assessing external data, which increases a protocol’s exposure to single-point-of-failure risks (Cong et al. (2023)).

Protocol size, proxied by the log of first-day total value locked (\log_TVL), is also an important determinant of audit adoption.²³ A significantly positive coefficient on \log_TVL in the *AUDIT* model (coefficient = 0.032) suggests that having greater total value locked is associated with heightened perceived risk of a hack, thereby encouraging at least some level of formal assurance. In addition, the \log_TVL coefficient in the multinomial model

²¹Top-tier auditors are defined as those with consistently high market share and visibility, including CertiK, Hacken, and PeckShield. These firms dominate audit volume and are cited frequently in post-mortems and crisis responses; for example, CertiK offers real-time security monitoring, and Hacken and PeckShield actively engage in incident forensics.

²²The marginal effects are approximated using the formula: Marginal Effect $\approx \beta_j \cdot \Pr(Y = j) \cdot (1 - \Pr(Y = j))$. Assuming baseline probabilities of 25% for top-tier audits and 30% for bottom-tier audits, the coefficients of 0.721 and 0.797 imply marginal effects of approximately 13.5% and 16.7%, respectively. Normalizing across the multinomial structure (with “no audit” as the base category) yields final probability increases of roughly 6.8 and 9.0 percentage points.

²³TVL is widely used as a measure for economic activity (e.g., Cong et al. (2023); Rabetti (2023); Campello, Jin, Rabetti, and Saleh (2023); Bhambhvani and Huang (2024); DeSimone, Jin, and Rabetti (2025); Knechel et al. (2025)).

is significantly positive for *BOTTOM*. This finding, combined with the insignificant coefficient for top-tier audits, suggests that developers of protocols with large total value locked are inclined to hire bottom-tier auditors when they decide to hire an auditor.

The *log_Chains* coefficient is significantly positive in both the logit specification and in the multinomial model (coefficients range from 0.378 to 0.647). These findings indicate that multi-blockchain deployment has a significantly positive influence on the decision to hire an auditor of any type. Staking activity also has a significantly positive influence on the decision to hire any auditor of any type. In particular, all of the (*log_Staking*) coefficients are significantly positive (ranging from 0.026 and 0.041). These findings suggest that protocols that are heavily reliant on staked assets face greater risk in the event of security failures, prompting them to strengthen their assurance by hiring an auditor.

External funding also shows a positive association with audit adoption, as indicated by the significantly positive *log_Raised* in all models (coefficients range from 0.04 to 0.05). Well-capitalized protocols are more likely to hire top-tier auditors. These findings suggest that protocols with external funding use audits to build trust among investors and institutional partners. Social-media presence, measured by Twitter followers (*log_Followers*), consistently predicts higher adoption across all audit type. The *log_Followers* coefficients range from 0.045 to 0.051; public visibility is positively related to the value of securing credible third-party validation. Protocols that are listed on a cryptocurrency exchange are more likely to hire an auditor, but only if the auditor is a top-tier auditor. In particular, the *LISTED* coefficient in the multinomial model is significantly positive only for top-tier audits (coefficient = 0.407).

The coefficient on lending indicator (*IND_LENDING*) is positive and significant for top-tier audit adoption (0.259 in the multinomial model). Lending protocols manage large collateral pools and complex liquidation mechanisms, often relying on oracle-based pricing. These features increase systemic risk, which likely drives stronger demand for premium assurance providers. While effects for bottom-tier audits are less pronounced, the consistently positive sign on the lending indicator reinforces the interpretation that operational complexity is associated with more stringent security practices.

Cross-chain deployment (*BC_CROSSCHAIN*) is significantly positively related to the likelihood of both top-tier and bottom-tier audits. The multinomial logit estimates suggest that cross-chain protocols are about five percentage points more likely to select top-tier auditors, consistent with additional vulnerabilities introduced by bridging infrastructure.

4.2 Hrb: Determinants of Auditor Choice Surrounding High-Profile DeFi Hack Events

We next examine the determinants of auditor choice surrounding high-profile hack events, beginning with a case study of one of the most prominent infrastructure hacks: the PolyNetwork exploit of August 2021. This incident, the first in DeFi history to surpass \$600 million in losses, marked a turning point in industry-wide awareness of smart contract vulnerabilities. The attack exploited weaknesses in PolyNetwork’s cross-chain, oracle-like, bridge infrastructure, allowing attackers to manipulate transaction verification and redirect assets across multiple blockchains. A single point of failure in these systems—such as compromised message validation or access controls—can expose the entire protocol to catastrophic losses, making such protocols prime targets for attackers.

[Figure 1 here]

Figure 1 illustrates the dynamics of top-tier centralized auditor choice around the Poly Network hack event. The figure plots the monthly share of newly launched protocols that selected top-tier centralized auditors, separately for oracle-integrated protocols (red line) and non-oracle protocols (blue line). Following the August 2021 shock, protocols with external data dependencies show a marked increase in their likelihood of engaging top-tier auditors. In contrast, protocols without such dependencies display no comparable shift in auditing behavior. This divergence supports the interpretation that protocols with greater exposure to infrastructure risk—such as those relying on external data inputs—respond more aggressively to the heightened security concerns by prioritizing top-tier code audits.

To address more generally the question of how auditor choice is affected by high-profile hack events, we adopt a stacked difference-in-differences (DiD) framework that leverages multiple quasi-natural experiments. Specifically, we identify six high-profile DeFi hack events as exogenous shocks that plausibly increase perceived risk for newly launched protocols.²⁴ These events were selected based on three criteria: (i) the magnitude of financial loss, (ii) potential for cross-chain contagion, and (iii) intensity of media and community attention. The selected events are: (a) Poly Network (August 2021), (b) BadgerDAO (December 2021), (c) Ronin Network (March 2022), (d) Binance Bridge (October 2022), (e) Euler Finance (March 2023), and (f) Orbit Bridge (December 2023).²⁵ These six systemic hack events collectively resulted in approximately \$2.2 billion USD in losses, accounting for nearly 25% of the total DeFi-related hacking losses during the 2021-2024 period.

For each systemic hack event, we construct a symmetric event window comprising protocols launched within six months before and after the incident. These event-specific windows are then stacked to form a pooled panel, enabling a difference-in-differences (DiD) estimation framework that systematically captures protocol responses

²⁴See Appendix E for extended explanation on security breaches.

²⁵Hack details: [Poly Network Hack](#), [BadgerDAO Hack](#), [Ronin Hack](#), [Binance Bridge Attack](#), [Euler Finance Hack](#), [Orbit Bridge Hack](#).

to distinct security shocks.²⁶ By pooling across multiple quasi-experiments, the stacked DiD design offers several advantages. First, it increases statistical power by aggregating localized treatment contrasts across events. Second, the use of clean and symmetric windows restricts comparisons to protocols exposed to similar temporal environments, thereby mitigating confounding factors from long-run trends or unrelated structural changes. Third, the design accommodates event fixed effects, allowing us to account for heterogeneity in shock characteristics, such as timing and attack vectors. Crucially, because these attacks were both unexpected and widely publicized, they constitute plausibly exogenous shocks in the perceived security environment for newly launched protocols. This quasi-experimental variation enables a robust interpretation of post-hack adjustments effects on auditor choice, particularly among protocols with greater *ex ante* vulnerability.

To test how auditor choice is affected by high-profile hack events, we estimate the following logistic regression:

$$\text{AuditorType}_{it} = \beta_1(\text{BIGHACK}_t \times \text{ProtocolType}_i) + \beta_2\text{ProtocolType}_i + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \theta_g + \varepsilon_{it}, \quad (3)$$

where the dependent variable AuditorType_{it} denotes whether protocol i at time t selected a top-tier (*TOP*) or bottom-tier centralized (*BOTTOM*) auditor. The variable BIGHACK_t is a post-event indicator that equals one for protocols launched within six months after a given systemic hack event, and zero for those launched in the six months preceding it. To identify treatment heterogeneity, we define ProtocolType_i as an indicator for protocol-level exposure to systemic vulnerabilities. Specifically, we focus on three dimensions of heightened security risk: (i) protocols that exposed to external data environments through oracles or bridges (*ORACLE*), (ii) protocols operating in the lending sector (*IND_LENDING*), and (iii) protocols deployed across multiple blockchains (*BC_CROSS_CHAIN*). These characteristics reflect higher risk exposure due to technical complexity or composability with other smart contracts. The coefficient of interest, β_1 , reflects whether protocols with greater *ex ante* exposure to security risk are differentially more likely to engage centralized auditors following systemic hack events. We include controls for a set of protocol characteristics \mathbf{X}'_{it} as described in Section 4.1. We also include fixed effects for industry, blockchain, and launch month, as well as event fixed effects.

[Table 3 here]

Table 3 reports the results of estimating equation (3). Panel (a) focuses on oracle-integrated protocols. The findings in Column (1) reveal that following major systematic hack events, oracle-based protocols are significantly more likely to select top-tier centralized auditors relative to non-oracle protocols (*ORACLE* \times *BIGHACK* coefficient =

²⁶The stacked DiD framework explicitly addresses overlap by treating each event as a separate cohort with its own time-relative window (e.g., Cengiz, Dube, Lindner, and Zipperer (2019)) and allows consistent estimation of average treatment effects even in the presence of overlapping samples. Moreover, the DiD results indicate that the key effects on auditor choice only emerge around the fourth month post-shock, suggesting that protocols take time to internalize security risks and initiate auditor engagement.

0.051). Conversely, Column (2) reveals a significant decline in the likelihood of selecting bottom-tier auditors for oracle-based protocols ($ORACLE \times BIGHACK$ coefficient = -0.050). Together, findings in Columns (1) and (2) suggest that there is an upward shift in audit quality preferences for top-tier auditors among oracle-based protocols in the aftermath of major security breaches.

Panel (b) examines lending protocols. Although the findings in Column (3) reveal there is no significant change in top-tier auditor selection following hack events ($IND_LENDING \times BIGHACK$ coefficient = 0.017), Column (4) reveals that lending protocols become significantly less likely to choose bottom-tier auditors post-shock ($IND_LENDING \times BIGHACK$ coefficient = -0.052). Panel (c) reports estimates for protocols deployed across multiple blockchains. The findings in Columns (5) and (6) reveal no significant shift in either top- or bottom-tier auditor selection in response to hack events ($BC_CROSSCHAIN \times BIGHACK$ coefficients = 0.014 and -0.019).

Taken together, the estimates in Table 3 suggest that following systemic shocks, protocols with greater exposure to systemic risks—such as those operating in affected environments through shared infrastructure (e.g., oracles) or in industries where customer assets face direct vulnerability to exploitation (e.g., lending protocols)—tend to shift towards higher-quality assurance from top-tier centralized auditors at launch.

4.3 Hic: Decentralized and centralized auditors as complements and substitutes

We next examine whether and under what conditions decentralized audits serve as *complements* to, or *substitutes* for, centralized audits. Consistent with the centralized and decentralized audits serving as complements, we find evidence that over 91% of pre-launch audits are conducted by centralized auditors, whereas 77% of post-launch audits are conducted by decentralized auditors. These statistics suggest that centralized and decentralized auditors play complementary roles in that they are hired by protocol developers at different points in time.

However, the hiring practices of developers in the aftermath of systemic security breaches suggest that decentralized auditors also may serve as substitutes for centralized auditors. Figure 2 plots the time series of the number of decentralized audits and total compensation awards for bounty hunters. For each protocol, the sample begins in the month of its launch (or January 2021, whichever is later) and continues monthly until the end of the observation window. The plots suggest periods of increased demand for decentralized auditing following the occurrence of the six major security incidents described in Section 4.2. This pattern is consistent with decentralized audits serving as substitutes for additional centralized assurance in the aftermath of systemic security breaches.

[Figure 2 here]

To assess more formally whether decentralized audits serve as substitutes following systemic security breaches, we estimate the following logistic regression:

$$\begin{aligned}
BOUNTY_{i,t+1} = & \beta_1(BIGHACK_t \times ProtocolType_i) + \beta_2(BIGHACK_t \times ProtocolType_i \times AuditorType_i) \\
& + \beta_3(BIGHACK_t \times AuditorType_i) + \beta_4(ProtocolType_i \times AuditorType_i) \\
& + \beta_5 ProtocolType_i + \beta_6 AuditorType_i + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_{it},
\end{aligned} \tag{4}$$

where the dependent variable $BOUNTY_{i,t+1}$ is an indicator that equals one if protocol i engages a bounty hunter in month $t + 1$. All other variables are as defined in equation (3). The coefficient β_1 captures whether protocols with higher infrastructure vulnerability—such as oracle or lending protocols—are more likely to select decentralized auditing in the aftermath of major systemic shocks, consistent with a complementary, shock-induced demand for live assurance. The triple-interaction coefficient β_2 further tests whether this response is moderated by the protocol’s initial auditing strategy (e.g., *TOP* for top-tier centralized audit): a negative estimate is consistent with *substitution on the margin*—i.e., protocols with strong pre-launch audits exhibiting a dampened post-shock shift toward decentralized auditing.

Table 4 reports results of estimating equation (4). As in Table 3, we present findings for three categories of high-risk protocols: oracle-based (Panel a), lending (Panel b), and cross-chain (Panel c). The findings in Column (1) reveal that oracle-based protocols are significantly more likely to hire decentralized auditors in the months following a major systemic hack event ($ORACLE \times BIGHACK$ coefficient = 0.007). The findings in Column (2) show that the coefficient on the triple interaction, $BIGHACK \times ORACLE \times TOP$, is significantly negative (−0.020), suggesting that protocols that had engaged top-tier centralized auditors before launch are less likely to hire bounty hunters following high-profile hack events. This pattern indicates that strong pre-launch audits reduce the perceived necessity of reactive bounty engagements.

[Table 4 here]

A similar pattern emerges among lending and cross-chain protocols. In particular, the findings in Columns (3) and (5) reveal that both types of protocols are significantly more likely to hire decentralized auditors in the months following a major systemic hack event ($IND_LENDING \times BIGHACK$ coefficient = 0.010 and $BC_CROSSCHAIN \times BIGHACK$ coefficient = 0.008). In addition, the findings in Columns (4) and (6) show that the corresponding three-way interaction terms are significantly negative (−0.035 and −0.015), providing further evidence of substitution on the margin when strong centralized audits are already in place.

Taken together, our findings suggest that systemic hack events act as salient external triggers for security re-assessment and decentralized audit adoption—especially among protocols with high exposure to operational risk—while protocols that underwent rigorous centralized audits before launch exhibit a dampened post-shock tendency to engage decentralized auditors. Overall, decentralized and centralized audits co-exist as joint inputs to security: decentralized audits *supplement* centralized audits by supplying continuous, live monitoring in normal times and around routine upgrades; however, decentralized audits *substitute* centralized ones on the margin in the immediate aftermath of shocks when rapid assurance is required or when additional centralized reviews exhibit diminishing returns in line with evidence in other settings (Titman and Trueman, 1986; Dyck et al., 2010; Lennox and Pittman, 2011; DeFond and Zhang, 2014; Pan et al., 2023).

5 H2: Does Smart Contract Auditing Mitigate Security Breaches?

This section addresses our second research question: Is having an audit associated with the likelihood of the audited protocol suffering a future security breach, and whether, conditional on having an audit, an audit from a top-tier vs. bottom-tier auditor is associated with the likelihood of a security breach? Before addressing this question, it is helpful to examine the empirical distribution of hack events post-launch. Figure 3 plots the distribution of hack occurrences relative to the launch date of each protocol for all protocols, audited protocols, and unaudited protocols. The figure reveals a large concentration of attacks within the first 90 days post-launch, with nearly 80% of hack events occurring within the first 180 days both for audited and especially for non-audit protocols. This pattern highlights the acute vulnerability of DeFi protocols early in their life cycle, which suggests that attackers target newly launched protocols.

[Figure 3 here]

5.1 Security breaches and *Audit vs. Non-Audit* choice: Extensive margin

We begin our analysis of the effects of pre-launch auditor choices on post-launch security breaches by focusing on the extensive margin of auditor choice, i.e. whether having pre-launch audits is associated with the likelihood of post-launch hacks and with their size. The decision by protocol developers to engage with auditors before launch is clearly endogenously related to protocol characteristics that also reflect decisions made by developers. We adopt two commonly used approaches to addressing non-random selection into auditing. First, we construct a propensity score matched (PSM) sample of audited and unaudited protocols with similar characteristics. Statistics in Appendix

Table A3 reveal that after matching, covariates do not differ significantly across groups. Then, we estimate the following regression equation using both the full sample and PSM sample at the protocol level:

$$\text{HackOutcome}_{it} = \beta_1 \text{AUDIT}_i + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_{it}, \quad (5)$$

where HackOutcome_{it} is either a binary indicator for whether protocol i was hacked in period t (*HACKDUM*) or the log of the dollar losses conditional on a hack (*Hackloss*).²⁷ We estimate the *HACKDUM* specifications using logit and the *Hackloss* specifications using OLS. The indicator Audit_i equals one if the protocol was audited by any centralized auditor before launch. The vector of control variables \mathbf{X}_{it} includes protocol characteristics, such as oracle integration, cross-chain deployment, DAO governance, initial staking activity, external funding, GitHub activity, and social-media presence, as described in Section 4.1. We include fixed effects for industry (γ_j), blockchain (λ_k), and launch year-month (δ_t). Robust standard errors are clustered at the Industry \times Blockchain level.²⁸

[Table 5 here]

Table 5 the regression summary statistics associated with estimation of equation (5). Panel A reports the baseline (naïve) regressions based on the full sample. The findings reveal positive coefficients on *AUDIT* for both outcome variables in Columns (1–2)—which we interpret as selection, since higher-risk or more complex protocols are more likely to be hacked or to have higher hack-related loss.²⁹ Panel B reports results for the matched (PSM) sample. The findings in Columns (3–4) reveal that the coefficients on *AUDIT*, 0.145 and 0.090, are insignificant. These findings suggest that audits have no discernible association with the likelihood of a protocol being hacked post-launch and with the magnitude of hack-related losses.

Our second approach to address the endogenous choice to engage an auditor is to exploit an exogenous demand shock: the issuance of a Public Service Announcement in August 2022 by the U.S. Federal Bureau of Investigation (FBI) of guidelines urging crypto investors to verify whether a DeFi protocol had undergone an audit. This announcement created an upward shift in the demand for audits, which is likely strong for community-governed DeFi protocols with decentralized autonomous organization (DAO) structures. DAO-governed protocols are more

²⁷The *Hackloss* variable parallels measures used in the auditing literature that quantify the economic magnitude of adverse events. For example, Gipper et al. (2020) examines how financial statement users and markets respond to the size (in dollars) of client losses and mis-statements associated with audit deficiencies. In the DeFi setting, *Hackloss* similarly captures the financial materiality of a breach, providing a complementary perspective to the binary hack occurrence measure.

²⁸Because the number of protocols that experience hacks is relatively small (~200 out of 4,100+), some fixed-effect groups (e.g., certain industry types or launch months) exhibit no variation in the hack outcome. We therefore exclude such singleton groups from the logit estimation; see Breuer and DeHaan (2024) for guidance.

²⁹Analogously, neighborhoods with higher crime rates tend to have more police stations, where the correlation is an artifact of the presence of police stations and high crime rates reflecting the same underlying risk, not that police stations cause crime. We follow standard practice (Angrist and Pischke (2009, 2015)) and report naïve (potentially endogenous) OLS regressions alongside PSM/IV estimates to make the source and direction of bias transparent and to demonstrate how designs aimed at controlling for selection potentially mitigate these biases.

likely to face investor-driven pressure to adopt security best practices, including audits. In contrast to traditional centralized governance, DAOs enable direct participation by token holders in decision-making, creating stronger monitoring incentives and collective action on risk mitigation.³⁰ We measure this effect using the interaction of $POSTFBI_t$ and DAO_i , where $POSTFBI_t$ equals one for protocols launched after August 2022 and DAO_i equals one if the protocol has DAO governance in place.³¹

Formally, we estimate the following instrumental-variable (IV) model:

$$\text{First Stage: } AUDIT_{it} = \beta (POSTFBI_t \times DAO_i) + \alpha' \mathbf{X}_{it} + \mu_i + \delta_t + \varepsilon_{it}, \quad (6)$$

$$\text{Second Stage: } HackOutcome_{it+6} = \delta \widehat{AUDIT}_{it} + \theta' \mathbf{X}_{it} + \mu_i + \delta_t + u_{it}, \quad (7)$$

where the first stage instruments audit adoption using the interaction of $POSTFBI$ and DAO . Table 5, Panel C, reports estimates of equations (6) and (7). The significantly positive coefficient on the interaction term, 0.113 ($p = 0.047$, Anderson Rubin test statistic = 4.07, $p = 0.0437$) in the first-stage estimation in Column (1) indicates that the FBI guidance increased audit adoption among DAO-governed protocols; given the baseline audit rate of 42%, this implies an increase of roughly 27% in the probability of conducting an audit for DAO protocols launched after the guidance. The instrument plausibly satisfies the exclusion restriction, as the timing of the FBI announcement and DAO governance status should affect hack outcomes only through audit adoption.

The IV second stage (Column (2)) findings reveal an insignificant coefficient for $HACKDUM$, indicating that, after correcting for endogeneity, whether a protocol is audited has no impact on the probability of a security breach. The findings in Column (3) reveal a similarly insignificant effect for $Hackloss$. These results are consistent with the inferences using matched samples, suggesting that the extensive margin of auditing choice is not associated with (prevention of) future breaches.

5.2 Security breaches and *Top-Tier* vs. *Bottom-Tier* auditor choice: Intensive margin

We next examine whether auditors' credibility—using as a proxy auditors' market share—also is associated with the likelihood of security breaches. Specifically, we distinguish between top- and bottom-tier auditors and estimate

³⁰This mirrors findings in the corporate governance literature on shareholder activism, where dispersed but engaged owners can exert significant influence over governance outcomes (e.g., Gillan and Starks (2000); Gompers, Ishii, and Metrick (2003)). In the DeFi setting, token-holder voting and proposal mechanisms make security-related decisions—including commissioning audits—more salient and responsive to community demand (e.g., Cong, Rabetti, Wang, and Yan (2025)).

³¹The August 2022 issuance of a Public Service Announcement by the FBI warned investors that “cyber criminals are increasingly exploiting vulnerabilities in DeFi protocols” and explicitly recommended that investors ensure a protocol “has conducted one or more code audits performed by independent auditors” (<https://www.ic3.gov/PSA/2022/PSA220829>). This official advisory heightened awareness of the importance of audits, especially among DAO-governed protocols whose active communities demand robust security assurances. The timing and nature of the guidance support both the *relevance* criterion—by prompting DAO-governed protocols to adopt audits post-guidance—and the *exclusion restriction*—because the advisory is unlikely to influence hack outcomes except through its impact on audit adoption.

a version of equation (5) in which we replace the *AUDIT* indicator with indicators for *TOP* and *BOTTOM* auditors:

$$\text{HackOutcome}_{it} = \beta_1 \text{TOP}_i + \beta_2 \text{BOTTOM}_i + \mathbf{X}'_{it} \gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_{it}, \quad (8)$$

where, as before, TOP_i (BOTTOM_i) equals one if the protocol is audited by a top-tier (bottom-tier) auditor before launch.

Similar to the decision whether to hire a centralized auditor pre-launch, the choice of auditor quality (the intensive margin of auditor choice) is also endogenous.³² We employ two strategies to mitigate selection bias. First, we implement a two-step Heckman procedure. In the first stage, we model audit choice (i.e., whether a protocol engages an auditor) as a function of exogenous factors (industry, blockchain, and protocol features), then compute the inverse Mills ratio (IMR) to adjust for the non-random sample of audited protocols (Heckman (1979, 1990)). In the second stage, we re-estimate equation (8), while augmenting the regression with the IMR. The findings reported in Table 6, Panel B, reveal that after the self-selection correction, the coefficient on *TOP* is insignificant in the *HACKDUM* and *Hackloss* specifications. Thus, when restricting the sample to audited protocols, we find no evidence that auditor reputation is systematically associated with vulnerability to security breaches.

[Table 6 here]

Our second strategy to address endogeneity of the auditor quality choice is exploiting an exogenous technological shock—the public release of ChatGPT in November 2022—as a source of variation in the relative cost and feasibility of internal code assurance. ChatGPT and similar AI tools significantly enhanced automated code review, debugging, and vulnerability detection, reducing the marginal benefit of hiring costly top-tier auditors.³³ For example, recent studies (e.g., Xia, Shao, He, Yu, Song, and Zhang (2024); Assaraf (2023)) show that large language models (LLMs) such as ChatGPT can detect software vulnerabilities and coding errors with high accuracy at a fraction of the cost of human experts.³⁴ This effect should be strongest for protocols with in-house technical teams, proxied by presence of a GitHub repository before launch.

We instrument top-tier auditor choice using the interaction $GITHUB_i \times POSTChatGPT_t$, where $GITHUB_i$ indicates whether the protocol maintained a GitHub repository before launch (a proxy for development capacity)

³²For completeness, Table 6, Panel A, include regression summary statistics base on estimation of equation (8) using the full sample.

³³Similarly, recent evidence shows that generative AI may also influence financial analysts’ capabilities (Bertomeu, Lin, Liu, and Ni (2025)).

³⁴In addition, we test the effectiveness of ChatGPT on finding smart contract bugs in Appendix F. Our experimental evidence suggests that generative AI tools, such as OpenAI’s GPT models, can substantially augment in-house smart contract auditing.

and $POSTChatGPT_t$ equals one for protocols launched after November 2022. The first stage is:

$$\text{First Stage: } TOP_{it} = \beta (GITHUB_i \times POSTChatGPT_t) + \alpha' \mathbf{X}_{it} + \mu_i + \delta_t + \varepsilon_{it}, \quad (9)$$

$$\text{Second Stage: } HackOutcome_{it+6} = \delta \widehat{TOP}_{it} + \theta' \mathbf{X}_{it} + \mu_i + \delta_t + u_{it}, \quad (10)$$

and the second stage regresses hack outcomes on the fitted value \widehat{TOP}_{it} .

The instrument satisfies the relevance condition because the release of ChatGPT substantially altered the cost-benefit calculus of engaging a top-tier auditor. Protocols with in-house development capacity—proxied by maintaining a GitHub repository before launch—are best positioned to incorporate generative AI tools into their internal security reviews. Evidence from Appendix G shows that disruptions to ChatGPT service, identified from OpenAI’s system logs, caused a pronounced same-day drop in GitHub commits for open-sourced DeFi projects, highlighting the extent to which AI tools had become embedded in developer workflows. This reliance on AI-assisted coding suggests that, post-release, well-resourced teams could perform more thorough in-house code assurance, thereby reducing the marginal benefit of hiring an expensive top-tier auditor.

Our first-stage regression, estimated based on equation (9), reported in column (1) of Panel C of Table 6, confirms this conjecture. The marginal benefit of hiring an expensive top-tier auditor declined after the technology shock, leading to a significant drop in top-tier auditor adoption (first-stage coefficient -0.170 , $p < 0.01$). The instrument passes weak-instrument diagnostics (Anderson–Rubin test statistic 7.55 , $p = 0.006$). Given a baseline top-tier audit adoption rate of roughly 23%, this coefficient implies a reduction of about 74% in the probability of engaging a top-tier auditor for high-capacity protocols launched after ChatGPT’s release.

The exclusion restriction is also plausible: although the $GITHUB \times POSTChatGPT$ interaction may influence the decision to hire a top-tier auditor, it should not directly affect the likelihood or magnitude of future hacks except through this channel. The logic is that the public release of ChatGPT did not alter the underlying code vulnerabilities or hacker incentives in a way that is systematically correlated with GitHub presence. Moreover, by controlling for protocol characteristics, industry, blockchain, and time fixed effects, we mitigate the possibility that the interaction term captures other contemporaneous trends in security practices or attack patterns.

Columns (2)–(3) of Panel C report the results of the second stage estimation, equation (10). The coefficient on TOP is significantly negative in both the $HACKDUM$ and $Hackloss$ specifications (coefficients = -1.710 and -5.488), indicating that despite AI-assisted tools reducing the likelihood of hiring a top-tier auditor, engaging with a top-tier auditor is associated with lower likelihood and severity of future security breaches.

5.3 Additional Test of Security breaches and *Top-Tier vs. Bottom-Tier* auditor choice: Bartik Shift-Share Instrument

As an additional robustness test to our previous IV analysis, we use a Bartik (shift-share) instrument. Our previous IV interacted GitHub presence with the ChatGPT release. However, GitHub openness is a strategic choice and may correlate with unobserved fundamentals. The Bartik approach interacts a pre-period, ecosystem-level exposure measure with each protocol’s launch-time blockchain weights, so that protocols more concentrated on certain chains are more strongly “exposed” to those chains’ pre-existing environments.³⁵ Hence, the Bartik approach uses plausibly exogenous differences across blockchain ecosystems (measured strictly before our outcomes), and lets each protocol’s pre-set cross-chain footprint translate those aggregate differences into protocol-level variation.³⁶ In our context, blockchains differ in decentralized governance intensity (DAO activity) and peer-monitoring resources. Protocols that are more exposed—by design—to DAO-intensive chains should need less additional assurance from costly top-tier auditors, yielding a strong negative first-stage link from the instrument to top-tier audit choice.³⁷

Appendix H details the analysis and results relating to the two-stage Bartik IV procedure. Table H1 presents the regression summaries for both stages and demonstrates that our Bartik instrument strongly predicts a protocol’s choice of a top-tier auditor in the first stage, while in the second stage, the use of a top-tier auditor correspondingly reduces the protocol’s likelihood of a hack and limits the severity of losses should a breach occur. Table H2 outlines the four broad blockchain categories used to construct the Bartik instrument (Ethereum, other EVM-compatible chains, non-EVM chains, and others) and shows how DeFi protocols are distributed across these different chain environments. Overall, the Bartik-instrumented analysis yields inferences consistent with our earlier instrumental variable approach. Specifically, engaging a top-tier auditor is associated with a lower likelihood of subsequent security breaches and smaller financial losses when breaches occur.

5.4 Security breaches and decentralized audit choice

Finally, we examine the efficacy of decentralized audits. In our analysis of centralized auditors, we examine their influence on the probability of hacks post-launch. Decentralized auditors are typically hired post-launch. Therefore, our analysis of decentralized auditors’ influence on protocol security centers around their hiring in the

³⁵For applications of Bartik instruments in other settings, see [Bartik \(1991\)](#) and [Autor, Dorn, and Hanson \(2013\)](#). See also [Diamond \(2016\)](#) and [Goldsmith-Pinkham, Sorkin, and Swift \(2020\)](#) for identification conditions.

³⁶Specifically, for each blockchain c , we measure a pre-period exposure intensity (e.g., governance/DAO intensity). Each protocol i has a vector of launch-time weights $\{w_{ic}\}$ that sum to one and represent its value-weighted economic footprint across chains (e.g., a protocol listed on three blockchains might have $w_{i,ETH} = 0.60$, $w_{i,EVM} = 0.30$, $w_{i,NonEVM} = 0.10$). Intuitively, these weights reflect the likelihood that protocol i is exposed to a given chain’s environment. The Bartik instrument for protocol i is then the weighted average of pre-period intensities across its chain mix, $B_i = \sum_c w_{ic} s_c^{pre}$.

³⁷As noted in Section 5.1, unlike traditional centralized governance structures, DAOs allow token holders to participate directly in decision-making, thereby strengthening monitoring incentives and facilitating collective action in risk mitigation.

protocol’s post-launch period. For the models involving decentralized auditors, the unit of analysis is protocol-month.

At the protocol-month level, we estimate the following regression equation:

$$\text{HackOutcome}_i = \beta_1 \text{BOUNTY}_{it} + \beta_2 \text{TOP}_i + \beta_3 \text{BOTTOM}_i + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_{it}, \quad (11)$$

where BOUNTY_{it} equals one if protocol i adopts a bounty program in month t . All specifications include industry, blockchain, and year-month fixed effects, with standard errors clustered at the Industry \times Blockchain level.

As with centralized audit decisions, decentralized auditing choices may be endogenous: higher-risk or more sophisticated protocols may be more likely to adopt bounty programs. Table 7, Panel A, reports the regression summary statistics associated with estimation of equation (11) using the full sample. Column (1) shows that BOUNTY has a significantly negative coefficient (-0.004 , $p < 0.01$) on HACKDUM , while Column (2) shows a negative coefficient (-0.112 , $p < 0.01$) on Hackloss . In contrast, the coefficients on TOP and BOTTOM are small and largely insignificant in this post-launch setting. To address the potential effects of endogeneity, we employ the same instrument used in the previous subsection. Specifically, we instrument BOUNTY_{it} with $\text{GITHUB}_i \times \text{POSTChatGPT}_t$ in a two-stage least squares framework:

$$\text{First Stage: } \text{BOUNTY}_{it} = \beta (\text{GITHUB}_i \times \text{POSTChatGPT}_t) + \alpha' \mathbf{X}_{it} + \mu_i + \delta_t + \varepsilon_{it}, \quad (12)$$

$$\text{Second Stage: } \text{HackOutcome}_{it+6} = \delta \widehat{\text{BOUNTY}}_{it} + \theta' \mathbf{X}_{it} + \mu_i + \delta_t + u_{it}, \quad (13)$$

where $\widehat{\text{BOUNTY}}_{it}$ is the predicted adoption of decentralized auditing in the first stage.

[Table 7 here]

Table 7, Panel B, reports the results from estimation of equations (12) and (13). As evident from the first-stage estimation, the instrument predicts decentralized auditing adoption (first-stage coefficient 0.024 , $p < 0.01$) with a partial F -statistic of 62.21 . The second stage shows that bounty programs significantly reduce both the likelihood of security breaches and losses due to hacking events within six months of adoption (BOUNTY coefficients = -0.279 and -1.415). These results suggest that decentralized audits provide a robust additional layer of protection in the post-launch environment.

Altogether, the evidence in this section reveals three main insights: (i) the extensive margin of the auditor choice does not, on average, appear to be associated with the likelihood of security breaches and their severity; (ii) however, conditional on hiring an auditor, hiring a top-tier auditor is associated with a lower likelihood of security breaches;

and (iii) audits conducted by decentralized auditors post-launch are also associated with a lower incidence of future breaches.

6 H3: How DeFi Protocols and Auditors Respond to Security Breaches?

Finally, our last set of empirical tests addresses our third research question, i.e., how do DeFi developers and auditors respond to security breaches and what are the consequences for auditor reputation?

6.1 H3a: Post-breach auditor switching

6.1.1 Switching Centralized Auditors after Security Breaches

We start by examining hacked protocols' responses to security breaches. More formally, we estimate logit models for (i) switching auditors, (ii) moving to a top-tier auditor, and (iii) moving to a bottom-tier auditor, after a hack, using the following equation:

$$SWITCH_{it} = \beta HACKED_{it} + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_i, \quad (14)$$

The dependent variable $SWITCH_{it}$ is a binary indicator for whether protocol i switched auditors or decided to have an audit. Specifically, we consider multiple switching outcomes: $SWITCH$ (any auditor switch, including hiring an auditor for the first time), T to T (remain with top-tier), T to B (switch from top-tier to bottom-tier), B to T (switch from bottom-tier to top-tier), B to B (remain with bottom-tier), N to T (move from no audit to top-tier), and N to B (move from no audit to bottom-tier). The key explanatory variable is $HACKED$, which equals one in periods following a major security breach. All regressions include protocol-level controls \mathbf{X}_{it} as described in Section 3.2, as well as industry, blockchain, and year fixed effects. Standard errors are clustered at the Industry \times Blockchain level.³⁸

[Table 8 here]

Table 8 reports results of estimating equation (14). The findings in Column (1) reveal that the $HACKED$ coefficient, 0.605, is significantly positive, which suggests that there is a substantial increase in the likelihood of protocols switching auditors after a security breach. Columns (2) through (7) provide further insights into the direction of these changes. The insignificant $HACKED$ coefficients in Columns (2) and (3) imply that protocols that hired

³⁸The number of observations used to estimate equation (14) is smaller than the full sample size (4,108) because fixed-effects logit estimation excludes groups with no within-group variation in the dependent variable. For example, if all protocols within a given launch-month cell never switch auditors, all of these observations are excluded, since they provide no information for identifying the coefficients in a conditional logit framework. This mechanical property of fixed-effects logit explains the reduced sample size relative to the full panel.

top-tier auditors before launch are no more likely to switch to any other auditor following security breaches. In contrast, the results in Columns (4) and (5) suggest that protocols that hired bottom-tier auditors before launch are more likely to switch to top-tier and to other bottom-tier auditors following security breaches. However, the likelihood of switching to a top-tier auditor is 2.6 times higher (*HACKED* coefficients = 1.293 vs. 0.496). Finally, the results in Columns (6) and (7) suggest that protocols that did not hire an auditor before launch are more likely to switch to a top-tier auditor following a security breach (*HACKED* coefficient = 1.582), but not to a bottom-tier auditor.

Taken together, the findings in Table 8 indicate that protocols experiencing security breaches are more likely to change auditors, and this adjustment is asymmetric: they overwhelmingly favor upgrading to top-tier auditors rather than downgrading or retaining lower-quality options. This pattern mirrors evidence from traditional audit markets, where high-quality audits are linked to stronger monitoring (DeFond and Zhang, 2014; Knechel and Willenborg, 2016) and firms frequently replace incumbent auditors with higher-reputation alternatives following adverse shocks (Benston and Hartgraves, 2002; Chi et al., 2009).

6.1.2 Hiring Decentralized Auditors after Security Breaches

We proceed by analyzing whether hacked protocols adopt decentralized auditing following security breaches. Specifically, we focus on the following post-hack actions: (i) hiring a decentralized auditor post-launch, (ii) hiring a decentralized auditor post-launch if no centralized auditor was hired pre-launch, and (iii) hiring a decentralized auditor post-launch if either a bottom-tier or top-tier (centralized) auditor was hired pre-launch.³⁹

We modify equation (14) to model the probability that a protocol adopts a bounty program by replacing the dependent variables with indicators equaling to one if protocol i hired a decentralized auditor after a security breach. Specifically, we consider multiple decentralized auditor outcomes: *BOUNTY* (hiring a decentralized auditor), *NON to BOUNTY* (hiring a decentralized auditor post-launch if no centralized auditor was hired pre-launch), *BOTTOM to BOUNTY* (hiring a decentralized auditor post-launch if a bottom-tier auditor was hired pre-launch), and *TOP to BOUNTY* (hiring a decentralized auditor post-launch if a top-tier auditor was hired pre-launch). The key explanatory variable is *HACKED*, which equals one in periods following a major security breach. All regressions include protocol-level controls \mathbf{X}_{it} as described in Section 3.2, as well as industry, blockchain, and year fixed effects. Standard errors are clustered at the Industry \times Blockchain level.

[Table 9 here]

³⁹We restrict the analysis to protocols with total value locked (TVL) above the sample mean (1,179 out of 4,108). Smaller protocols are often too opaque to allow a reliable assessment of data breaches and post-breach crisis management decisions.

The findings in Table 9 reveal a significant increase in decentralized auditing adoption following a hack. In particular, all of the *HACKED* coefficients are significantly positive except for that in the *NON to BOUNTY* specification. Interestingly, the *HACKED* coefficient in the *BOTTOM to BOUNTY* specification, 0.914, is more than double that in the *TOP to BOUNTY* specification, 0.440. These findings are consistent with protocols that had a bottom-tier auditor perceiving greater benefit to hiring a decentralized auditor following a security breach. Overall, the findings in Table 9 parallel insights from the auditing and assurance literature (e.g., Chi et al., 2009; DeFond and Zhang, 2014), which demonstrates that adverse events precipitate changes in assurance intensity and governance strategies.

6.2 H3b: Auditor Reputation and Market Share after Systemic and Protocol-level Hacks

In traditional audit markets, audit failures can lead to client losses, regulatory scrutiny, and diminished market share (e.g., Chi et al., 2009; DeFond and Zhang, 2014; Aobdia and Shroff, 2017). In this section, we examine whether similar dynamics appear to unfold in the smart contract audit markets. In particular, we examine whether the market penalizes auditors associated with high-profile hacks, especially when their crisis response is perceived as inadequate.

6.2.1 Case Study: Crisis Management by Three Large Centralized Auditors

To illustrate these dynamics, we turn to the PolyNetwork hack of August 2021 and examine how three leading centralized audit firms (CertiK, PeckShield, and Hacken) managed the crisis and how their subsequent market shares evolved.

CertiK responded to the PolyNetwork hack by positioning itself as the authoritative investigator. Within 48 hours, it published a widely circulated technical post-mortem, explaining the root cause of the vulnerability and mitigation steps. CertiK’s rigorous communication—via formal blogs rather than through social media—earned endorsements from industry peers and even the PolyNetwork hacker, who credited CertiK for uncovering critical details. This forensic leadership reinforced CertiK’s brand as a security thought leader and plausibly helped the firm capture increased market share in the months following the hack.

PeckShield responded to the PolyNetwork hack by adopting a highly public-facing strategy, providing real-time updates and on-chain transaction tracking through Twitter. Its live commentary kept pressure on the hacker and elevated PeckShield’s visibility among the broader crypto community. Although this response was technically less comprehensive than CertiK’s formal analysis, PeckShield’s role as a rapid-response “watchdog” bolstered its reputation for monitoring and incident alerts, though its actions did not appear to result in a long-term recovery of its pre-hack market share.

Hacken, by contrast, took a quieter approach to the PolyNetwork hack. It focused on client reassurance through private channels and internal vulnerability reviews. Hacken engaged less in public discourse and did not release a prominent incident report immediately. Although some of its existing clients may have appreciated its approach to addressing the hack, Hacken’s muted presence appeared to limit its visibility during the crisis, and may have contributed to its sharp and sustained decline in market share.

[Figure 4 here]

Figure 4 documents these dynamics. The figure suggests that after an initial dip in market share (from roughly 35% to 25%), CertiK rebounded and consolidated its leadership position post the hack. PeckShield also rebounded, but its market share never fully recovered from its initial decline (stabilizing near 5–10%). Hacken had a large and sustained drop in its market share (from 15–20% pre-hack to below 5%), effectively marginalizing its presence among top-tier firms.

6.2.2 Post-Hack Auditor Market Share

We now turn to examine the effect on the market share of the centralized auditor of a hack of one of its audited protocols. In particular, we analyze market share changes following security breaches, including all top-tier auditors in our sample—CertiK, PeckShield, Hacken, Halborn, Quantstamp, and Slowmist. To do this, we estimate the following linear model:

$$MarketShare_{it+h} = \beta HACKED_{it} + \alpha_i + \delta_t + \varepsilon_{it} \quad (15)$$

The dependent variable $MarketShare_{it+h}$ measures the share of audits of auditor i at horizon $t + h$ (where $h \in \{3, 6, 12\}$ months after a hack). We measure market share using equal-weighting and also weighting by total value locked (TVL). $HACKED_{it}$ is an indicator that equals one following a hack of one of their audited protocols. All regressions include auditor fixed effects (α_i) and year-month fixed effects (δ_t). Standard errors are clustered at the auditor level.

[Table 10 here]

Table 10 reports estimates of equation (15). Panel A, which presents findings based on equally weighted market shares, reveals a significant short-term loss in market share for auditors whose audited protocols were hacked. In particular, the significantly negative $HACKED$ coefficient, -0.042 , indicates that within three months of a protocol hack, affected auditors lose approximately 4.2 percentage points of market share. This effect is economically

meaningful: relative to an average market share of 0.20 among top-tier auditors, the loss represents about 21% of market share. However, the insignificant *HACKED* coefficient over the two longer horizons suggests that the effect possibly dissipates over longer horizons. Panel B, which presents findings based on value-weighted market shares, reveals a similar pattern of a significant short-term loss in market share (*HACKED* coefficient = -0.042).

We also examine whether the magnitude of hack losses affects auditor market share. To do this, we replace *HACKED* with *Hackloss* the estimation equation (15). Panel C, which presents the findings from this estimation, reveals that greater hack-related financial losses also are associated with a short-term decline in auditor market share.

Taken together, the results in Table 10 suggest that DeFi auditors' market share is sensitive to security failures in the short run. This dynamic is parallel to findings with traditional audit markets, where reputational shocks can have adverse effects on client retention (e.g., Cohen et al., 2008; Chi et al., 2009), although the effects appear to be more short-lived in the DeFi market.

7 Conclusion

This study examines the emerging market for voluntary audits of smart contracts. We address three core research questions: (1) What factors are associated with protocols' decision to engage an auditor—and what type of auditor to hire—both before and after protocol launch? (2) Do audits, and the choice of auditor, effectively mitigate the likelihood of future security breaches? (3) How do protocols and auditors respond to a security breach? Using a dataset of thousands of audit reports and hundreds of bounty programs linked to over 4,000 DeFi protocols launched between 2020 to 2025, we examine both centralized and decentralized auditing.

Our findings yield several insights. First, audit adoption is strongly associated with key protocol features. For example, protocols with risk-prone designs—such as oracle integration or cross-chain deployment—are more likely to engage top-tier auditors. These patterns intensify in response to systemic security shocks, such as the PolyNetwork hack, showing that the demand for auditing before launch is sensitive to market-wide vulnerabilities. In addition, we provide evidence that decentralized audits are generally complementary to centralized one, but the two types of audits become substitutes following security breaches. Second, we find that although, on average, audits do not reduce the likelihood of future security breaches, audits conducted by top-tier centralized auditors and decentralized auditors are associated with lower likelihood of future breaches and losses condition on a breach occurring. Following a breach, protocols tend to upgrade to higher-quality auditors or turn to decentralized auditing following a security breach, and affected auditors suffer short-term losses in market share. Reputational recovery for auditors seems to depend on public response strategies and market confidence rebuilding.

Our study contributes to several strands of literature. We contribute to the auditing and assurance literature by empirically examining theories of voluntary audit choice, assurance demand under uncertainty, and audit quality differentiation in the emerging and under-regulated market for smart contract audits. We contribute to the DeFi literature by providing the first systematic evidence on the scope and effectiveness of both centralized and decentralized audits in preventing security breaches. Our findings have practical implications for protocol developers evaluating audit strategies, for security firms managing reputation risks, and for policymakers seeking to strengthen the integrity of code-based financial systems.

Overall, the results of our study highlight a dual evolution in DeFi assurance: centralized audits by top-tier auditors appear useful for pre-launch assurance, but decentralized auditing has emerged as another key source of assurance, particularly following security breaches. Together, our study's findings suggest a future in which both centralized and decentralized auditing play important roles in sustaining trust and resilience in decentralized finance.

Declaration of generative AI and AI-assisted technologies in the writing process: During the preparation of this work, the authors used ChatGPT to improve the readability and language of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

- Allee, K. D. and T. L. Yohn (2009). The demand for financial statements in an unregulated environment: An examination of the production and use of financial statements by privately held small businesses. *The Accounting Review* 84(1), 1–25.
- Amiram, D., E. Lyandres, and D. Rabetti (2025). Trading volume manipulation and competition among centralized crypto exchanges. *Management Science* 71(3), 1369–1387.
- Angrist, J. D. and A. B. Krueger (1995). Split-sample instrumental variables estimates of the return to schooling. *Journal of Business & Economic Statistics* 13(2), 225–235.
- Angrist, J. D. and J.-S. Pischke (2009). *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton, NJ: Princeton University Press.
- Angrist, J. D. and J.-S. Pischke (2015). *Mastering 'Metrics: The Path from Cause to Effect*. Princeton, NJ: Princeton University Press.
- Antle, R. and B. Nalebuff (1991). Conservatism and auditor-client negotiations. *Journal of Accounting Research* 29, 31–54.
- Aobdia, D. and N. Shroff (2017). Regulatory oversight and auditor market share. *Journal of Accounting and Economics* 63(2–3), 262–287.
- Armstrong, C., J. D. Kepler, D. Samuels, and D. Taylor (2022). Causality redux: The evolution of empirical methods in accounting research and the growth of quasi-experiments. *Journal of Accounting and Economics* 74(2), 101521.
- Ashraf, M., P. N. Michas, and D. Russomanno (2020). The impact of audit committee information technology expertise on the reliability and timeliness of financial reporting. *The Accounting Review* 95(5), 23–56.
- Assaraf, A. (2023, July). Is chatgpt helping or hurting your developers' productivity? DevOps.com.
- Atanasov, V. and B. S. Black (2021). The trouble with instruments: Pretreatment balance in shock-based instrumental variable designs. *Management Science* 67(2), 1270–1302.
- Autor, D. H., D. Dorn, and G. H. Hanson (2013). The china syndrome: Local labor market effects of import competition in the united states. *American economic review* 103(6), 2121–2168.
- Bartik, T. J. (1991). *Who Benefits from State and Local Economic Development Policies?* Kalamazoo, MI: W.E. Upjohn Institute for Employment Research.
- Benston, G. J. and A. L. Hartgraves (2002). Enron: What happened and what we can learn from it. *Journal of Accounting and Public Policy* 21(2), 105–127.
- Bertomeu, J., Y. Lin, Y. Liu, and Z. Ni (2025). The impact of generative AI on information processing: Evidence from the ban of ChatGPT in italy. *Journal of Accounting and Economics* 80(1), 101782.
- Bhambhwani, S. and A. H. Huang (2024). Auditing decentralized finance. *British Accounting Review* 56(2), 101270.
- Blaconiere, W. and M. DeFond (1997). An investigation of audit opinions and subsequent auditor litigation of publicly-traded failed savings and loans. *Journal of Accounting and Public Policy* 16, 415–454.
- Bourveau, T., J. Brendel, and J. Schoenfeld (2024). Decentralized finance (DeFi) assurance: Early evidence. *Review of Accounting Studies* 29, 2209–2253.
- Bourveau, T., M. Breuer, J. K. Koenraadt, and R. Stoumbos (2025). Public company auditing around the securities exchange act: historical lessons for esg assurance. *The Accounting Review* 100(3), 107–138.

- Breuer, M. and E. DeHaan (2024). Using and interpreting fixed effects models. *Journal of Accounting Research* 62(5), 1183–1226.
- Campello, M., P. Jin, D. Rabetti, and F. Saleh (2023). The market for crypto zombies: Under-collateralization in DeFi lending. Working Paper. (<https://tinyurl.com/3zfm9a8m>).
- Capponi, A. and R. Jia (2025). Liquidity provision on blockchain-based decentralized exchanges. *Review of Financial Studies* 38(10), 3040–3085.
- Cengiz, D., A. Dube, A. Lindner, and B. Zipperer (2019). The effect of minimum wages on low-wage jobs. *The Quarterly Journal of Economics* 134(3), 1405–1454.
- Chi, W., H. Huang, Y. Liao, and H. Xie (2009). Mandatory audit partner rotation, audit quality, and market perception: Evidence from taiwan. *Contemporary accounting research* 26(2), 359–391.
- Cohen, D. A., A. Dey, and T. Z. Lys (2008). Real and accrual-based earnings management in the pre- and post-sarbanes-oxley periods. *The Accounting Review* 83(3), 757–787.
- Cong, L., C. R. Harvey, D. Rabetti, and Z.-Y. Wu (2025). An anatomy of crypto-enabled cybercrimes. *Management Science* 71(4), 3622–3633.
- Cong, L., D. Rabetti, C. C. Y. Wang, and Y. Yan (2025, March). Centralized governance in decentralized organizations. (<https://ssrn.com/abstract=5168660>).
- Cong, L. W., W. R. Landsman, E. L. Maydew, and D. Rabetti (2023). Tax-loss harvesting with cryptocurrencies. *Journal of Accounting and Economics* 76(2–3), 101607.
- Cong, L. W., E. Prasad, and D. Rabetti (2023). Financial and informational integration through decentralized oracle networks. (<https://dx.doi.org/10.2139/ssrn.4495514>).
- Cook, J., Z. T. Kowaleski, M. Minnis, A. Sutherland, and K. M. Zehms (2020). Auditors are known by the companies they keep. *Journal of Accounting and Economics* 70(1), 101314.
- DeAngelo, L. (1981). Auditor independence, “low-balling” and disclosure regulation. *Journal of Accounting and Economics* 3, 113–127.
- DeFond, M., D. H. Erkens, and J. Zhang (2017). Do client characteristics really drive the big n audit quality effect? new evidence from propensity score matching. *Management Science* 63(11), 3628–3649.
- DeFond, M. and K. Subramanyam (1998). Auditor changes and discretionary accruals. *Journal of Accounting and Economics* 25, 35–67.
- DeFond, M. and J. Zhang (2014). A review of archival auditing research. *Journal of Accounting and Economics* 58(2–3), 275–326.
- Desai, M. A. and D. Dharmapala (2009). Corporate tax avoidance and firm value. *The Review of Economics and Statistics* 91(3), 537–546.
- DeSimone, L., P. Jin, and D. Rabetti (2025). Tax planning, illiquidity, and credit risks: Evidence from DeFi lending. (<http://dx.doi.org/10.13140/RG.2.2.32320.85760>).
- Diamond, R. (2016). The determinants and welfare implications of u.s. workers’ diverging location choices by skill: 1980–2000. *American Economic Review* 106(3), 479–524.
- Dowling, C. and S. A. Leech (2014). A big four firm’s use of information technology to control the audit process: How an audit support system is changing audit practice. *Contemporary Accounting Research* 31(1), 230–252.
- Duguay, R., M. Minnis, and A. Sutherland (2020). Regulatory spillovers in common audit markets. *Management Science* 66, 3389–3411.

- Dyck, A., A. Morse, and L. Zingales (2010). Who blows the whistle on corporate fraud? *The Journal of Finance* 65(6), 2213–2253.
- Dye, R. (1991). Informationally motivated auditor replacement. *Journal of Accounting and Economics* 14(2), 347–374.
- Fedyk, A., J. Hodson, N. Khimich, and T. Fedyk (2022). Is artificial intelligence improving the audit process? *Review of Accounting Studies* 27, 938–985.
- Feng, D., R. Hitsch, K. Qin, A. Gervais, R. Wattenhofer, Y. Yao, and Y. Wang (2023). DeFi auditing: Mechanisms, effectiveness, and user perceptions. *Cryptology ePrint Archive, Paper 2023/1207*.
- Francis, J., E. Maydew, and H. Sparks (1999). The role of big 6 auditors in the credible reporting of accruals. *Auditing: A Journal of Practice & Theory* 18(2), 17–34.
- Francis, J. R. (2004). What do we know about audit quality? *The British Accounting Review* 36(4), 345–368.
- Francis, J. R. (2011). A framework for understanding and researching audit quality. *Auditing: A Journal of Practice & Theory* 30(2), 125–152.
- Frishkoff, P. (1989). Some observations on the extent of bank audits in america: 1800-1863. Article 19. *Accounting Historians Notebook* 12(1), 41–47.
- Gefen, O., D. Rabetti, Y. Sun, and C. Zhang (2024). Code-washing: Evidence from open-source blockchain startups. (<https://ssrn.com/abstract=5068292>).
- Gillan, S. L. and L. T. Starks (2000). Corporate governance proposals and shareholder activism: The role of institutional investors. *Journal of Financial Economics* 57(2), 275–305.
- Gipper, B., C. Leuz, and M. Maffett (2020). Public oversight and reporting credibility: Evidence from the pcaob audit inspection regime. *The Review of Financial Studies* 33(10), 4532–4579.
- Goldsmith-Pinkham, P., I. Sorkin, and H. Swift (2020). Bartik instruments: What, when, why, and how. *American Economic Review* 110(8), 2586–2624.
- Gompers, P. A., J. L. Ishii, and A. Metrick (2003). Corporate governance and equity prices. *Quarterly Journal of Economics* 118(1), 107–156.
- Gul, F., S. Fung, and B. Jaggi (2009). Earnings quality: Some evidence on the role of auditor tenure and auditors' industry expertise. *Journal of Accounting and Economics* 47, 265–287.
- Harvey, C., A. Ramachandran, and J. Santoro (2021). DeFi and the Future of Finance. Hoboken, New Jersey: Wiley.
- Harvey, C. R. and D. Rabetti (2024). International business and decentralized finance. *Journal of International Business Studies* 55, 840–863.
- Hasbrouck, J., T. Rivera, and F. Saleh (2022). The need for fees at a dex: How increases in fees can increase dex trading volume.
- Heckman, J. (1979). Sample selection bias as a specification error. *Econometrica* 47(1), 153–161.
- Heckman, J. (1990). Varieties of selection bias. *The American Economic Review* 80(2), 313–318.
- Kaplan, S. and D. Williams (2013). Do going concern audit reports protect auditors from litigation? A simultaneous equations approach. *The Accounting Review* 88, 199–232.
- Kausar, A., N. Shroff, and H. White (2016). Real effects of the audit choice. *Journal of Accounting and Economics* 62, 157–181.

- Knechel, W. R., S. Maex, and H. J. Park (2025, December). Decentralized finance (DeFi) and cybersecurity assurance. (<https://ssrn.com/abstract=4658750>).
- Knechel, W. R. and M. Willenborg (2016). Economics-based auditing research published in JAR. *Journal of Accounting Research Virtual Issue*.
- Krishnamurthy, S., J. Zhou, and N. Zhou (2006). Auditor reputation, auditor independence, and the stock-market impact of andersen's indictment on its client firms. *Contemporary Accounting Research* 23(2), 465–490.
- Lehar, A. and C. Parlour (2025). Decentralized exchange: The uniswap automated market maker. *Journal of Finance* 80(1), 321–374.
- Lennox, C. and J. Pittman (2010). Big five audits and accounting fraud. *Contemporary Accounting Research* 27(1), 209–247.
- Lennox, C., J. Schmidt, and A. Thompson (2023). Why are expanded audit reports not informative to investors? Evidence from the united kingdom. *Review of Accounting Studies* 28, 497–532.
- Lennox, C. S. and J. A. Pittman (2011). Voluntary audits versus mandatory audits. *The Accounting Review* 86(5), 1655–1678.
- Lisowsky, P. and M. Minnis (2020). The silent majority: Private U.S. firms and financial reporting choices. *Journal of Accounting Research* 58(3), 547–588.
- Lobo, G. J., L. Paugam, D. Zhang, and J.-F. Casta (2017). The effect of joint auditor pair composition on audit quality: Evidence from impairment tests. *Contemporary Accounting Research* 34(1), 118–153.
- Luo, M., D. Rabetti, and S. Yu (2024). Blockchain adoption and audit quality. (<https://ssrn.com/abstract=5074602>).
- Lyandres, E., B. Palazzo, and D. Rabetti (2022). ICO success and post-ICO performance. *Management Science* 68(12), 8658–8679.
- Lyandres, E. and A. Zaidelson (2025). Quantitative investments in decentralized finance.
- Magee, R. and M. Tseng (1990). Audit pricing and independence. *The Accounting Review* 65(2), 315–336.
- Makarov, I. and A. Schoar (2022). Cryptocurrencies and decentralized finance (DeFi). Working Paper. (<http://www.nber.org/papers/w30006>).
- Malinova, K. and A. Park (2024). Learning from defi: Would automated market makers improve equity trading.
- Milionis, J., C. Moallemi, T. Roughgarden, and A. L. Zhang (2024). Automated market making and loss-versus-rebalancing.
- Minnis, M. (2011). The value of financial statement verification in debt financing: Evidence from private U.S. firms. *Journal of Accounting Research* 49(2), 457–506.
- Minnis, M. and N. Shroff (2017). Why regulate private firm disclosure and auditing? *Accounting and Business Research* 47, 473–502.
- Mutchler, J.F., H. W. and J. McKeown (1997). The influence of contrary information and mitigating factors on audit opinion decisions on bankrupt companies. *Journal of Accounting Research* 35(2), 295–310.
- Neal, T. and R. Riley Jr. (2004). Auditor industry specialist research design. *Auditing: A Journal of Practice and Theory* 23(2), 169–177.
- Pan, Y., N. Shroff, and P. Zhang (2023). The dark side of audit market competition. *Journal of Accounting and Economics* 75(1), 101520.

- Parlour, C. A. (2023). How safe is DeFi? Systemic risk and fragility. *Key Note Talk at the 2023 Global AI Finance Research Conference*.
- Rabetti, D. (2023, May). Auditing decentralized finance (DeFi) protocols. (<https://ssrn.com/abstract=4458298>).
- Reichelt, K. and D. Wang (2010). National and office-specific measures of auditor industry expertise and effects on audit quality. *Journal of Accounting Research* 48(3), 647–686.
- Rozario, A. M. and M. A. Vasarhelyi (2018). Auditing with smart contracts. *International Journal of Digital Accounting Research* 18, 1–27.
- Schoenfeld, J. (2024). Cyber risk and voluntary service organization control (SOC) audits. *Review of Accounting Studies* 29, 580–620.
- Simunic, D. (1980). The pricing of audit services: Theory and evidence. *Journal of Accounting Research* 18(1), 161–190.
- Smith, S. S. and J. Castonguay (2020). Blockchain and accounting governance: Emerging issues and considerations for accounting and assurance professionals. *Journal of Emerging Technologies in Accounting* 17(1), 119–131.
- Titman, S. and B. Trueman (1986). Information quality and the valuation of new issues. *Journal of Accounting and Economics* 8(2), 159–172.
- Tovanich, N., M. Kassoul, S. Weidenholzer, and J. Prat (2025). Contagion in decentralized lending protocols: A case study of compound.
- Wallace, W. A. (1980). The economic role of the audit in free and regulated markets. Research monograph, Graduate School of Management, University of Rochester, Rochester, NY.
- Watts, R. and J. Zimmerman (1983). Agency problems, auditing, and the theory of the firm: Some evidence. *Journal of Law and Economics* 26(3), 613–633.
- Wooldridge, J. M. (2010). *Econometric Analysis of Cross Section and Panel Data* (Second ed.). Cambridge, MA: MIT Press.
- Xia, S., S. Shao, M. He, T. Yu, L. Song, and Y. Zhang (2024). Auditgpt: Auditing smart contracts with chatgpt. *arXiv preprint arXiv:2404.04306*.
- Yuyama, T., K. Katayama, and P. Brigner (2023). Proposal of principles of DeFi disclosure and regulation. *Financial Cryptography and Data Security*. 13953.

Figure 1. (Systemic Shock) Probability of Choosing an Auditor (Systemic Hack Event): This figure plots the monthly probability of choosing a top-tier centralized auditor for newly launched protocols, separated by whether the protocol integrates an oracle service (Oracle = 1, red line; Oracle = 0, blue line). The vertical dashed line marks the timing of the Poly Network hack in August 2021, which was the largest cross-chain exploit in DeFi history. The attack specifically targeted the oracle mechanism and led to the unauthorized transfer of over \$600 million in assets across multiple blockchains. The y-axis indicates the mean audit probability in each launch month, and the x-axis denotes the protocol's launch time.

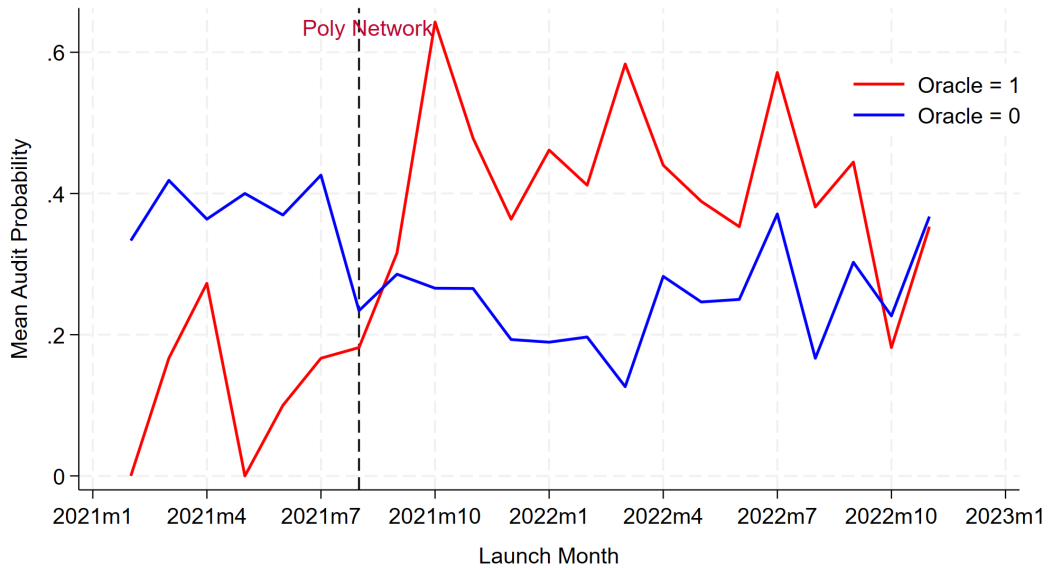


Figure 2. (Systemic Shock) Largest Hack Events and Decentralized Audit Demand. This figure illustrates the evolution of decentralized security activity in the DeFi ecosystem. The blue bars represent the total monthly value of bounty program awards (in USD), while the red line tracks the number of decentralized bounty programs per month. The six vertical dashed lines mark the timing of major systemic hack events: Poly Network, Badger DAO, Ronin Bridge, Binance Bridge, Euler Finance, and Orbit Bridge.

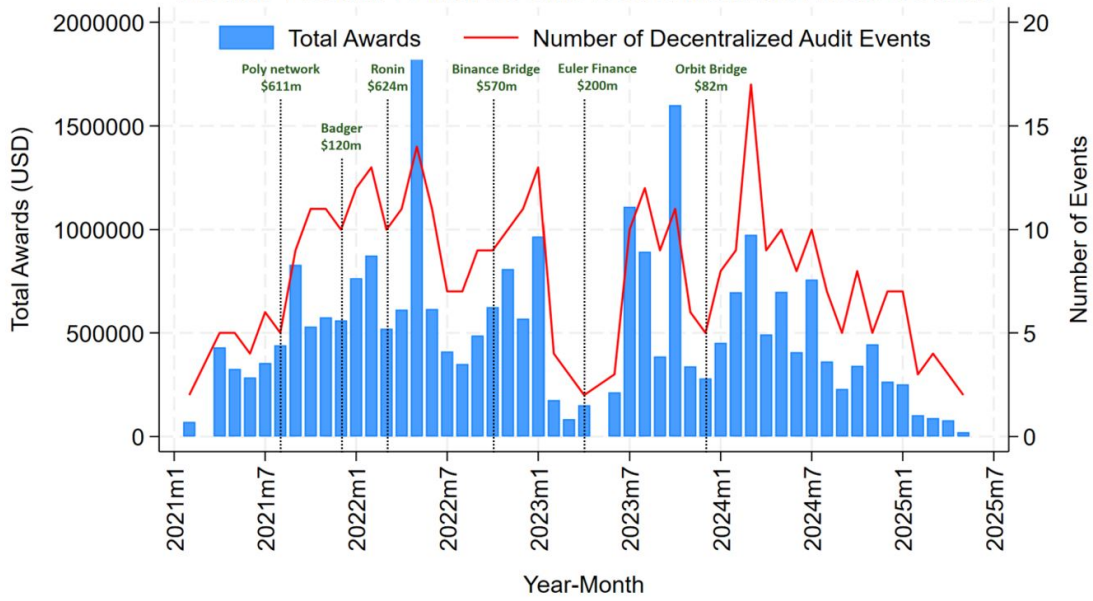


Figure 3. Distribution of Hacking Episodes: This figure visualizes the distribution of the number of days between each protocol’s launch and its first recorded hack event. The top panel shows the pooled distribution across all protocols in the sample. The middle and bottom panels split the sample by audit status at launch: protocols that received a centralized audit (AUDIT = 1) versus those that did not (AUDIT = 0). Histogram bins are set to 90-day intervals. The horizontal axis reflects the number of days from launch to the first hack, while the vertical axis indicates the number of affected protocols within each bin.

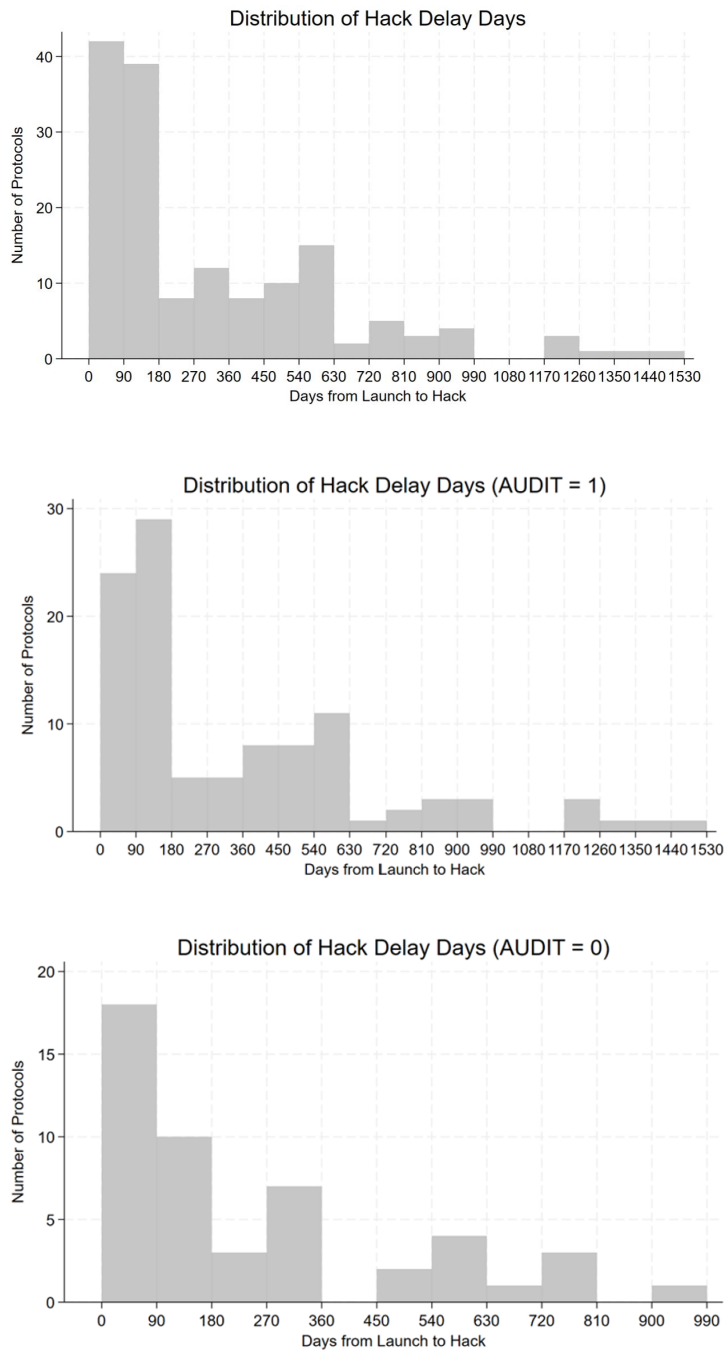


Figure 4. Auditor Reputation (PolyNetwork Case): This figure plots the monthly market share of three prominent centralized auditing firms—Certik (blue), PeckShield (red), and Hacken (green)—that were responsible for auditing Poly Network before its launch. Market share is measured as the share of newly launched protocols in a given month that were audited by each respective firm. The vertical blue bar indicates the timing of the Poly Network hack in August 2021, one of the largest and most publicized exploits in DeFi history.

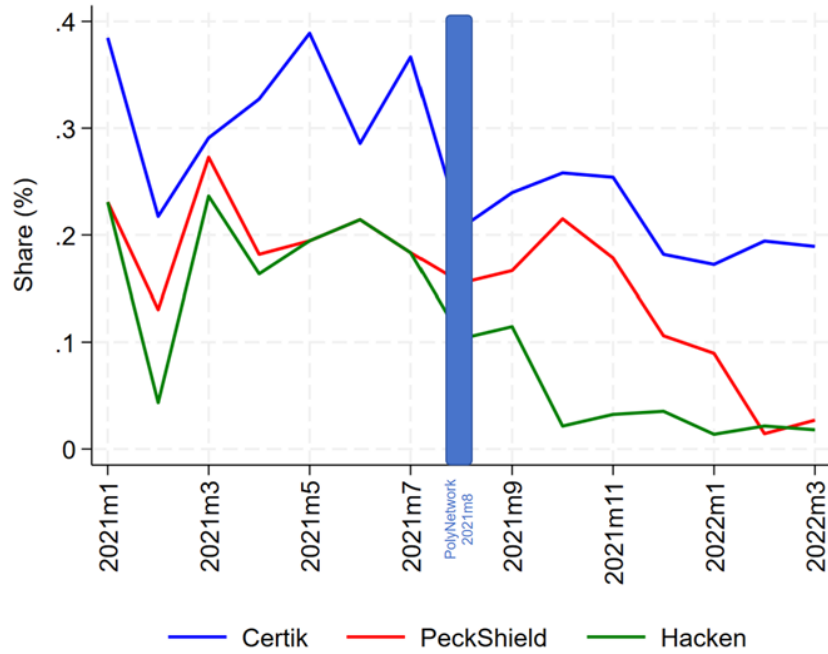


Table 1. Summary Statistics: Auditor and protocol Characteristics and Outcomes. This table presents summary statistics of variables used in the empirical analysis. *Audit choice:* *AUDIT* is an indicator equal to 1 if the protocol engaged any auditor; *TOP* equals 1 for top-tier centralized auditors (e.g., recognized large firms), and *BOTTOM* for bottom-tier centralized auditors. *protocol characteristics:* *ORACLE* equals 1 if the protocol uses external price feeds or off-chain data sources; *LISTED* equals 1 if the protocol is listed on a DeFi aggregator; *log_TVL* is the natural log of total value locked (TVL) in USD at launch; *DAO* equals 1 if the protocol uses decentralized governance; *GITHUB* indicates public code repositories; *log_Commits* measures development activity via GitHub commits (log); *log_Chains* is the log number of blockchains supported; *log_Raised* captures cumulative funding raised (log); *log_Staking* indicates staking functionality (log scale); *log_Followers* measures protocol social media presence (log number of followers). *Industry characteristics:* *IND_LENDING*, *IND_DEXES*, and *IND_YIELD* are sector dummies for lending protocols, decentralized exchanges, and yield aggregators, respectively. *Blockchain characteristics:* *BC_ETHEREUM* equals 1 if deployed on Ethereum; *BC_CROSSCHAIN* equals 1 if deployed on multiple blockchains; *log_EthereumTVL* is the log of aggregate TVL on Ethereum at time *t*. *Hack outcomes:* *HACKDUM* equals one if the protocol experienced a hack post-launch; *Hackloss* measures the natural logarithm of total monetary losses from hacks (in millions of USD). See Appendix Table A2 for variable definitions.

VARIABLES	Obs.	Mean	SD	p25	p50	p75
(a) Auditors						
<i>AUDIT</i>	4,108	0.46	0.50	0	0	1
<i>TOP</i>	4,108	0.23	0.42	0	0	0
<i>BOTTOM</i>	4,108	0.24	0.43	0	0	0
(b) Protocol characteristics						
<i>ORACLE</i>	4,108	0.18	0.39	0	0	0
<i>LISTED</i>	4,108	0.11	0.31	0	0	0
<i>log_TVL</i>	4,108	12.14	4.26	10.01	12.68	15.00
<i>TVL (\$ million)</i>	4,108	22.00	140.98	0.02	0.32	3.26
<i>DAO</i>	4,108	0.08	0.28	0	0	0
<i>GITHUB</i>	4,108	0.07	0.25	0	0	0
<i>log_Commits</i>	4,108	0.56	2.16	0	0	0
<i>Commits (#)</i>	4,108	1180.70	8,215.54	0	0	0
<i>log_Chains</i>	4,108	0.89	0.40	0.69	0.69	0.69
<i>Chains (#)</i>	4,108	1.43	0.49	0.99	0.99	0.99
<i>HAS_RAISED</i>	4,108	0.05	0.22	0	0	0
<i>log_Raised</i>	4,108	0.76	3.28	0	0	0
<i>Raised (\$ million)</i>	4,108	0.31	2.75	0	0	0
<i>log_Staking</i>	4,108	1.51	3.95	0	0	0
<i>Staking (\$ million)</i>	4,108	1.29	24.63	0	0	0
<i>log_Followers</i>	4,108	6.18	3.97	2.64	7.31	9.31
<i>Followers (#)</i>	4,108	482.96	52.09	13.97	1,500.77	11,061.90
(c) Industry characteristics						
<i>IND_LENDING</i>	4,108	0.11	0.31	0	0	0
<i>IND_DEXES</i>	4,108	0.34	0.47	0	0	1
<i>IND_YIELD</i>	4,108	0.11	0.31	0	0	0
(d) Blockchain characteristics						
<i>BC_ETHEREUM</i>	4,108	0.10	0.30	0	0	0
<i>BC_CROSSCHAIN</i>	4,108	0.25	0.43	0	0	0
<i>log_EthereumTVL</i>	4,108	24.49	0.49	24.05	24.51	24.91
(e) Hack outcomes						
<i>HACKDUM</i>	4,108	0.04	0.20	0	0	0
<i>Hackloss (\$million)</i>	4,108	1.07	16.48	0	0	0

Table 2. Determinants of Auditor Choice. This table reports regression estimates of the factors associated with the selection of any auditors, top-tier auditors, and bottom-tier auditors.

$$\text{Auditor}_i = \beta' \text{protocolFeatures}_i + \gamma_j + \lambda_k + \delta_t + \varepsilon_i$$

$$\Pr(\text{AuditorType}_i = c \mid \mathbf{X}_i) = \frac{\exp(\beta'_c \text{protocolFeatures}_i + \gamma_{jc} + \lambda_{kc} + \delta_{tc})}{\sum_{m \in C} \exp(\beta'_m \text{protocolFeatures}_i + \gamma_{jm} + \lambda_{km} + \delta_{tm})} \quad \text{for } c \in C$$

The dependent variable Auditor_i in the binary logit models equals 1 if protocol i hires at least one auditor (e.g., any auditor (*AUDIT*)), and 0 otherwise. For the multinomial logit specification, $\text{AuditorType}_i \in \{\text{Top}, \text{BOTTOM}, \text{None}\}$ denotes the mutually exclusive choice between hiring a centralized top-tier auditor (*TOP*), hiring a centralized bottom-tier auditor (*BOTTOM*), or having no auditor (*None*). The reported coefficients are relative to a baseline category (*None*), and probabilities are computed using the multinomial logit link. The regressor vector $\text{protocolFeatures}_i$ includes protocol, industry, and blockchain characteristics described in Table 1. Industry fixed effects γ_j , blockchain fixed effects λ_k , and year fixed effects δ_t are included in all regressions.

VARIABLES	(1) <i>AUDIT</i> (LOGIT) Mean = 0.46	(2) <i>TOP</i> (MNL) Mean = 0.23	(3) <i>BOTTOM</i> (MNL) Mean = 0.24
<i>ORACLE</i>	0.770 ^{***} (0.135)	0.721 ^{***} (0.131)	0.797 ^{***} (0.160)
<i>DAO</i>	0.198 (0.139)	0.207 (0.142)	0.201 (0.156)
<i>log_TV_L</i>	0.032 ^{***} (0.009)	0.015 (0.011)	0.050 ^{***} (0.010)
<i>log_Chains</i>	0.494 ^{***} (0.110)	0.647 ^{***} (0.161)	0.378 ^{***} (0.117)
<i>log_Staking</i>	0.032 ^{***} (0.011)	0.041 ^{***} (0.012)	0.026 ^{**} (0.013)
<i>log_Raised</i>	0.045 ^{***} (0.012)	0.050 ^{***} (0.012)	0.040 ^{***} (0.014)
<i>GITHUB</i>	-0.120 (0.398)	-0.341 (0.476)	0.077 (0.452)
<i>log_Commits</i>	0.058 (0.048)	0.087 (0.056)	0.029 (0.053)
<i>log_Followers</i>	0.049 ^{***} (0.012)	0.051 ^{***} (0.015)	0.045 ^{***} (0.013)
<i>LISTED</i>	0.297 ^{**} (0.122)	0.407 ^{***} (0.146)	0.193 (0.143)
<i>log_EthereumTVL</i>	-0.066 (0.179)	-0.037 (0.153)	-0.107 (0.244)
<i>IND_LENDING</i>	0.251 [*] (0.146)	0.259 [*] (0.142)	0.231 (0.169)
<i>BC_CROSSCHAIN</i>	0.499 ^{***} (0.130)	0.482 ^{***} (0.138)	0.492 ^{***} (0.138)
Observations	4,108		4,108
Pseudo R-squared	0.128		0.093
Industry FE	YES		YES
Blockchain FE	YES		YES
Year FE	YES		YES
Years	2020–2025		2020–2025

Notes: Robust standard errors clustered at the Industry \times Blockchain level in parentheses. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table 3. Dynamic Probability of Selecting Centralized Auditors Post-Large Security Breach Events. This table presents stacked difference-in-differences (DiD) regression results that estimate changes in the likelihood of selecting top or bottom centralized auditors for protocols exposed to specific risk types (Oracle, Lending, or Cross-chains), around systemic hack events. For each systemic hack event, we construct a symmetric event window comprising protocols launched within six months before and after the incident. These event-specific windows are then stacked to form a pooled panel, enabling a difference-in-differences (DiD) estimation framework that systematically captures protocol responses to distinct security shocks.

$$\text{AuditorType}_{it} = \beta_1(\text{BIGHACK}_t \times \text{ProtocolType}_i) + \beta_2\text{ProtocolType}_i + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \theta_g + \varepsilon_{it}$$

The dependent variable AuditorType_{it} indicates whether protocol i in month t selected a top (*TOP*) or bottom centralized auditor (*BOTTOM*). The main variable of interest is the interaction between BIGHACK_t , a dummy for the post-large hack period—The selected events are: (a) Poly Network (August 2021), (b) BadgerDAO (December 2021), (c) Ronin Network (March 2022), (d) Binance Bridge (October 2022), (e) Euler Finance (March 2023), and (f) Orbit Bridge (December 2023); and ProtocolType_i , which identifies risk-related protocol types: protocols using oracles or bridges to assess external data environments, lending protocols, or cross-chain protocols. Control variables \mathbf{X}_{it} include protocol-level characteristics reported in Table 2 and discussed in Section 4.1. Parallel trends plot on Appendix Figure A1. All specifications include industry, blockchain, year-month, and hack event fixed effects. Robust standard errors are clustered at the Industry \times Blockchain level.

	(1) <i>TOP</i>	(2) <i>BOTTOM</i>	(3) <i>TOP</i>	(4) <i>BOTTOM</i>	(5) <i>TOP</i>	(6) <i>BOTTOM</i>
TREATED	Panel (a): Oracle		Panel (b): Lending		Panel (c): Cross-chains	
<i>ORACLE</i> \times <i>BIGHACK</i>	0.051 ^{***} (0.018)	-0.050 ^{**} (0.020)				
<i>IND_LENDING</i> \times <i>BIGHACK</i>			0.017 (0.017)	-0.052 ^{**} (0.023)		
<i>BC_CROSSCHAIN</i> \times <i>BIGHACK</i>					0.014 (0.013)	-0.019 (0.013)
<i>ORACLE</i>	0.051 [*] (0.026)	0.104 ^{***} (0.027)	0.080 ^{***} (0.023)	0.074 ^{***} (0.028)	0.080 ^{***} (0.023)	0.074 ^{***} (0.028)
<i>IND_LENDING</i>	0.051 ^{**} (0.021)	0.033 [*] (0.019)	0.041 (0.026)	0.063 ^{**} (0.026)	0.052 ^{**} (0.020)	0.032 [*] (0.019)
<i>BC_CROSSCHAIN</i>	0.042 [*] (0.025)	0.075 ^{***} (0.025)	0.043 [*] (0.025)	0.075 ^{***} (0.025)	0.035 (0.028)	0.085 ^{***} (0.027)
Observations	7,204	7,204	7,204	7,204	7,204	7,204
R-squared	0.093	0.065	0.093	0.065	0.093	0.065
Controls	YES	YES	YES	YES	YES	YES
Industry FE	YES	YES	YES	YES	YES	YES
Blockchain FE	YES	YES	YES	YES	YES	YES
Year-mon FE	YES	YES	YES	YES	YES	YES
Event FE	YES	YES	YES	YES	YES	YES
Years	2020–2025	2020–2025	2020–2025	2020–2025	2020–2025	2020–2025

Notes: Robust standard errors clustered at the Industry \times Blockchain level in parentheses. ^{*} $p < 0.1$, ^{**} $p < 0.05$, ^{***} $p < 0.01$.

Table 4. Dynamic Probability of Selecting Decentralized Auditors Post-Large Security Breach Events. This table reports estimations of the likelihood of dynamic decentralized auditor selection after major security breaches. We construct a protocol-month level panel, capturing post-launch dynamics through December 2024. For each protocol, the sample begins in the month of its launch (or January 2021, whichever is later), and continues monthly until the end of the observation window. We estimate the following model:

$$\begin{aligned}
 BOUNTY_{it+1} = & \beta_1(BIGHACK_t \times ProtocolType_i) + \beta_2(BIGHACK_t \times ProtocolType_i \times AuditorType_i) \\
 & + \beta_3(BIGHACK_t \times AuditorType_i) + \beta_4(ProtocolType_i \times AuditorType_i) \\
 & + \beta_5 ProtocolType_i + \beta_6 AuditorType_i + \mathbf{X}'_{it} \gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_{it}
 \end{aligned}$$

The dependent variable $BOUNTY_{it+1}$ is an indicator equal to one if protocol i engages a bounty hunter in month $t+1$. The key regressors are interactions between the systemic hack period ($BIGHACK_t$), protocol-specific risk type (e.g., Oracle, Lending, Cross-chains), and audit quality. Columns (1), (3), and (5) show dynamic treatment effects of hack exposure. Columns (2), (4), and (6) extend the model to include three-way interactions that capture heterogeneous effects by audit quality. Control variables include protocol-level characteristics reported in Table 2 and discussed in Section 4.1. All specifications include industry, blockchain, and year-month fixed effects.

TREATED	Panel (a): Oracle		Panel (b): Lending		Panel (c): Cross-chains	
	(1)	(2)	(3)	(4)	(5)	(6)
$ORACLE \times BIGHACK$	0.007* (0.004)	0.013** (0.006)				
$IND_LENDING \times BIGHACK$			0.010*** (0.003)	0.023*** (0.009)		
$BC_CROSSCHAIN \times BIGHACK$					0.008** (0.003)	0.012** (0.005)
$ORACLE \times BIGHACK \times TOP$		-0.020** (0.010)				
$IND_LENDING \times BIGHACK \times TOP$				-0.035*** (0.012)		
$BC_CROSSCHAIN \times BIGHACK \times TOP$						-0.015* (0.009)
$ORACLE$	0.025** (0.011)	0.028** (0.014)	0.026** (0.011)	0.026** (0.011)	0.026** (0.011)	0.026** (0.011)
$IND_LENDING$	0.038 (0.027)	0.038 (0.027)	0.037 (0.027)	0.051 (0.037)	0.038 (0.027)	0.038 (0.027)
$BC_CROSSCHAIN$	0.050* (0.027)	0.050* (0.027)	0.050* (0.027)	0.050* (0.027)	0.049* (0.027)	0.046* (0.028)
Observations	98,995	98,995	98,995	98,995	98,995	98,995
R-squared	0.063	0.063	0.062	0.063	0.062	0.062
Controls	YES	YES	YES	YES	YES	YES
Industry FE	YES	YES	YES	YES	YES	YES
Blockchain FE	YES	YES	YES	YES	YES	YES
Year-month FE	YES	YES	YES	YES	YES	YES
Years	2020–2025	2020–2025	2020–2025	2020–2025	2020–2025	2020–2025

Notes: Robust standard errors clustered at the Industry \times Blockchain level are in parentheses. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table 5. Audit and Security Breach Mitigation. This table estimates the association between audit status and future hacking outcomes. Panel (a) presents baseline regressions using the full sample. Panel (b) reports results using a 1:1 propensity score matched (PSM) sample to account for non-random audit adoption — see Appendix Table A3 matching results and covariates balancing. Columns (1) and (3) estimate the likelihood of a security breach (*HACKDUM*), while columns (2) and (4) estimate the magnitude of losses (*Hackloss*) conditional on breach occurrence.

$$\text{HackOutcome}_{it} = \beta_1 \text{AUDIT}_i + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_{it}$$

The dependent variable HackOutcome_{it} corresponds to either (i) a hack occurrence in the subsequent period (*HACKDUM*), or (ii) the magnitude of losses in USD conditional on a breach (*Hackloss*). The key explanatory variable AUDIT_i indicates whether the protocol was audited before launch. The vector \mathbf{X}_{it} includes controls for protocol characteristics, with industry, blockchain, and year-month fixed effects. Robust standard errors are clustered at the industry \times blockchain level.

VARIABLES	(a) Baseline		(b) Matched Samples	
	(1) <i>HACKDUM</i>	(2) <i>Hackloss</i>	(3) <i>HACKDUM</i>	(4) <i>Hackloss</i>
<i>AUDIT</i>	0.304* (0.157)	0.151** (0.073)	0.145 (0.173)	0.090 (0.072)
<i>ORACLE</i>	0.046 (0.224)	0.085 (0.174)	-0.191 (0.283)	-0.137 (0.170)
<i>log_TVL</i>	0.074** (0.030)	0.036* (0.019)	0.049 (0.041)	0.028 (0.022)
<i>log_Chains</i>	0.429 (0.372)	0.737 (0.503)	0.423 (0.436)	0.301 (0.316)
<i>log_Staking</i>	0.014 (0.018)	0.015 (0.017)	0.012 (0.029)	0.008 (0.017)
<i>log_Raised</i>	0.028 (0.022)	0.028 (0.022)	0.092*** (0.019)	0.070*** (0.025)
<i>GITHUB</i>	0.582 (0.862)	0.631 (1.072)	1.438 (1.056)	1.180 (1.548)
<i>log_Commits</i>	-0.027 (0.107)	-0.014 (0.124)	-0.099 (0.129)	-0.078 (0.177)
<i>log_Followers</i>	-0.007 (0.023)	-0.008 (0.015)	0.009 (0.029)	-0.001 (0.016)
<i>LISTED</i>	-0.106 (0.247)	-0.102 (0.220)	0.278 (0.350)	0.162 (0.253)
<i>log_EthereumTVL</i>	-1.920* (1.146)	-0.805 (0.653)	-3.618** (1.512)	-1.086* (0.555)
<i>IND_LENDING</i>	1.397*** (0.215)	1.262*** (0.373)	1.627*** (0.288)	1.246*** (0.385)
<i>BC_CROSSCHAIN</i>	0.778** (0.304)	0.022 (0.368)	0.518 (0.371)	0.123 (0.215)
Observations	3,721	4,108	2,356	2,850
(Pseudo) R-squared	0.152	0.072	0.143	0.061
Industry FE	YES	YES	YES	YES
Blockchain FE	YES	YES	YES	YES
Year-mon FE	YES	YES	YES	YES
Years	2020–2025	2020–2025	2020–2025	2020–2025

Notes: Robust standard errors in parentheses. *** $p < 0.01$, ** $p < 0.05$, * $p < 0.1$.

Table 5 — Continuation. Instrumental Variable (IV) Estimates: FBI Intervention. Panel C estimates the effect of audits on preventing future security breaches using an instrumental variable (IV) design. The instrument is the interaction term $POSTFBI \times DAO$, which captures the introduction of FBI guidelines recommending DeFi investors to check if the protocol has been audited. Column (1) reports the first-stage IV-probit estimates for audit adoption. Columns (2) and (3) report second-stage estimates for the probability of a security breach ($HACKDUM$) and loss magnitude ($Hackloss$), respectively.

$$\text{First Stage: } AUDIT_{it} = \beta (POSTFBI_t \times DAO_i) + \alpha' \mathbf{X}_{it} + \mu_i + \delta_t + \varepsilon_{it},$$

$$\text{Second Stage: } HackOutcome_{it+6} = \delta \widehat{AUDIT}_{it} + \theta' \mathbf{X}_{it} + \mu_i + \delta_t + u_{it},$$

The outcome variable $HackOutcome_{it}$ corresponds to either (i) a binary indicator for whether protocol i experienced a security breach in period t ($HACKDUM$), or (ii) the dollar value of exploit losses conditional on a breach ($Hackloss$). The key explanatory variable $Audit_{it}$ captures whether the protocol engaged any audit firm before deployment. To address endogeneity in audit adoption, we use a two-stage approach. The first stage instruments audit adoption using the interaction term $POSTFBI \times DAO$, which captures the introduction of U.S. FBI guidance on auditing DeFi protocols conditional on DAO governance status. The second stage regresses each security outcome on the fitted audit probability from the first stage. All models include controls for protocol characteristics (e.g., service type, code activity, user engagement), as well as industry, blockchain, and year-month fixed effects. Standard errors are clustered at the industry-by-blockchain level.

(c) Instrumental Variable			
VARIABLES	(1) <i>AUDIT</i> First Stage	(2) <i>HACKDUM</i> Second Stage	(3) <i>Hackloss</i> Second Stage
<i>IV = (POSTFBI × DAO)</i>	0.113** (0.047)		
<i>AUDIT</i>		0.382 (1.582)	-0.987 (3.300)
<i>POSTFBI</i>	-0.072 (0.052)	0.024 (0.268)	-0.204 (0.440)
<i>DAO</i>	-0.001 (0.034)	0.270** (0.135)	0.282 (0.627)
Observations	4,108	4,108	4,108
(Pseudo) R-squared	0.160	0.194	0.060
Controls	YES	YES	YES
Industry FE	YES	YES	YES
Blockchain FE	YES	YES	YES
Year FE	YES	YES	YES
Weak-IV Diagnostics (AR Test)	4.07 ($p = 0.0437$)		

Notes: Robust standard errors clustered at the industry × blockchain level in parentheses. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table 6. Audit Type (Top-Tier vs Bottom-Tier) and Security Breach Mitigation. This specification estimates the association between audit type and the likelihood or magnitude of a hacking event, accounting for potential selection into auditing. Panel (a) reports results at the protocol launch level for centralized auditors. Columns (1)–(2) present logit and OLS estimates; Columns (3)–(4) report results with Heckman two-stage correction for selection bias.

$$\text{HackOutcome}_{it} = \beta_1 \text{AuditorType}_i + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_{it}$$

The dependent variable HackOutcome_{it} corresponds to either (i) a hack occurrence in the subsequent period (HACKDUM), or (ii) the magnitude of losses in USD conditional on a breach (Hackloss). The key explanatory variable AuditorType_i captures whether the protocol was audited by a top- or bottom-tier centralized auditor at launch. The Heckman second-stage specifications additionally incorporate the inverse Mills ratio (IMR) to correct for potential selection bias from non-random audit adoption. All models control for protocol-level covariates and include fixed effects for industry, blockchain, and year-month.

VARIABLES	(a) Baseline		(b) Heckman - 2nd Stage	
	(1) <i>HACKDUM</i>	(2) <i>Hackloss</i>	(3) <i>HACKDUM</i>	(4) <i>Hackloss</i>
<i>BOTTOM</i>	0.203 (0.224)	0.095 (0.127)		
<i>TOP</i>	0.371** (0.181)	0.199* (0.117)	0.173 (0.254)	0.120 (0.218)
<i>log_TVL</i>	0.075** (0.030)	0.087 (0.175)	-0.091 (0.324)	0.788* (0.418)
<i>ORACLE</i>	0.051 (0.269)	0.580** (0.269)	0.615*** (0.178)	1.018*** (0.288)
<i>log_Chains</i>	0.421 (0.374)	0.036* (0.019)	0.026 (0.032)	0.058* (0.031)
<i>log_Staking</i>	0.013 (0.018)	0.732 (0.399)	0.365 (0.399)	1.409** (0.613)
<i>log_Raised</i>	0.027 (0.022)	0.014 (0.032)	-0.009 (0.027)	0.018 (0.032)
<i>GITHUB</i>	0.022 (0.864)	0.017 (0.022)	-0.022 (0.024)	0.022 (0.032)
<i>log_Commits</i>	-0.032 (0.107)	0.637 (1.331)	0.165 (1.331)	0.305 (1.331)
<i>LISTED</i>	-0.112 (0.245)	0.124 (0.164)	0.005 (0.164)	0.044 (0.164)
<i>log_Followers</i>	-0.008 (0.023)	0.014 (0.034)	-0.047* (0.028)	0.018 (0.034)
<i>DAO</i>	0.537*** (0.193)	-0.008 (0.269)	-0.299 (0.269)	0.020 (0.269)
<i>log_EthereumTVL</i>	-1.965* (1.172)	-0.221 (0.654)	-0.567** (0.288)	-0.971 (1.297)
<i>IND_LENDING</i>	1.379*** (0.217)	1.262*** (0.373)	0.952*** (0.256)	1.486*** (0.234)
<i>BC_CROSSCHAIN</i>	0.695** (0.308)	0.368 (0.368)	0.240 (0.462)	0.288 (0.632)
<i>IMR (λ)</i>			-1.555 (1.207)	2.053 (1.406)
Observations	3,721	4,108	1,849	1,849
(Pseudo) R-squared	0.151	0.072	0.111	0.057
Industry FE	YES	YES	YES	YES
Blockchain FE	YES	YES	YES	YES
Year-mon FE	YES	YES	YES	YES

Notes: Robust standard errors clustered at the Industry \times Blockchain level in parentheses. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table 6 — Continuation. Instrumental Variable (IV) Estimates: AI-Assisted Code Verification. This table reports two-stage IV-Probit and two-stage least squares (2SLS) estimates evaluating the effect of engaging top-tier centralized auditors on DeFi protocol security. We use an interaction between GitHub activity and the ChatGPT release ($GITHUB \times POSTChatGPT$) as an instrument.

$$\text{First Stage: } TOP_{it} = \beta (GITHUB_i \times POSTChatGPT_t) + \alpha' \mathbf{X}_{it} + \mu_i + \delta_t + \varepsilon_{it},$$

$$\text{Second Stage: } HackOutcome_{it+6} = \delta \widehat{TOP}_{it} + \theta' \mathbf{X}_{it} + \mu_i + \delta_t + u_{it},$$

The outcome variable $HackOutcome_i$ corresponds to either (i) a binary indicator equal to one if protocol i was hacked after deployment ($HACKDUM$), or (ii) the dollar amount of loss conditional on a breach ($Hackloss$). The key explanatory variable TOP_i is a binary indicator equal to one if protocol i engaged a top-tier centralized auditor prior to launch. To address endogeneity in the choice of audit quality, we implement a two-stage instrumental variable strategy. The first stage instruments TOP_i using an interaction between GitHub developer activity and the public release of ChatGPT ($GITHUB \times POSTChatGPT$), which proxies for AI-assisted code verification and audit awareness. The second stage regresses each security outcome on the predicted probability of selecting a top-tier auditor. All regressions control for protocol-level characteristics, as well as industry, blockchain, and year-month fixed effects. Standard errors are clustered at the industry-by-blockchain level.

(c) Instrumental Variable			
VARIABLES	(1) <i>TOP</i> First Stage	(2) <i>HACKDUM</i> Second Stage	(3) <i>Hackloss</i> Second Stage
$IV = (GITHUB \times POSTChatGPT)$	-0.170 ^{***} (0.060)		
<i>TOP</i>		-1.710 ^{***} (0.323)	-5.488 (5.145)
<i>GITHUB</i>		-0.354 (0.760)	0.565 (1.835)
Observations	1,627	1,627	1,627
(Pseudo) R-squared	0.552	0.617	0.413
Controls	YES	YES	YES
Industry FE	YES	YES	YES
Blockchain FE	YES	YES	YES
Year-mon FE	YES	YES	YES
Weak-IV Diagnostics (AR Test)	7.55, $p = 0.006$		

Notes: Robust standard errors clustered at the Industry \times Blockchain level. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table 7. Decentralized Audit (Bounty Programs) and Security Breach Mitigation. This specification estimates the association between decentralized bounty audits and the likelihood or magnitude of a hacking event at the protocol-month level. We also control for whether the protocol was initially audited by a top-tier or bottom-tier centralized auditor. Columns (1)–(2) report OLS estimates for the binary indicator of a hack event (*HACKDUM*); Columns (3)–(4) report OLS estimates for the log of the dollar losses conditional on a breach (*Hackloss*).

$$\text{HackOutcome}_{it} = \beta_1 \text{AuditorType}_i + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_{it}$$

The dependent variable HackOutcome_{it} corresponds to either (i) a hack occurrence in the subsequent period (*HACKDUM*), or (ii) the magnitude of losses in USD conditional on a breach (*Hackloss*). The key explanatory variable AuditorType_i captures the audit structure adopted by protocol i at time t , including bounty-based decentralized audits and whether the protocol was previously audited by a top- or bottom-tier centralized auditor. All specifications include controls for protocol-level characteristics and the fixed effects indicated in the table.

VARIABLES	(1) <i>HACKDUM</i>	(2) <i>HACKDUM</i>	(3) <i>Hackloss</i>	(4) <i>Hackloss</i>
<i>BOUNTY</i>	−0.004 ^{***} (0.001)	−0.017 ^{***} (0.006)	−0.112 ^{***} (0.027)	−0.356 ^{***} (0.085)
<i>TOP</i>	0.001 (0.001)		0.026 [*] (0.013)	
<i>BOTTOM</i>	0.000 (0.001)		0.026 (0.016)	
<i>ORACLE</i>	0.003 [*] (0.002)		0.025 (0.024)	
<i>DAO</i>	0.005 ^{***} (0.001)		0.070 ^{***} (0.021)	
<i>log_TVL</i>	0.000 (0.000)		0.007 ^{***} (0.002)	
<i>log_Chains</i>	0.008 ^{***} (0.003)		0.134 ^{***} (0.041)	
<i>log_Staking</i>	0.000 ^{***} (0.000)		0.004 ^{**} (0.002)	
<i>log_Raised</i>	0.000 ^{***} (0.000)		0.007 ^{***} (0.002)	
<i>GITHUB</i>	0.001 (0.004)		0.055 (0.064)	
<i>log_Commits</i>	0.000 (0.000)		0.004 (0.007)	
<i>log_Followers</i>	0.000 ^{***} (0.000)		0.001 (0.001)	
<i>LISTED</i>	−0.003 ^{***} (0.001)		−0.060 ^{***} (0.015)	
<i>log_EthereumTVL</i>	−0.002 ^{***} (0.001)		−0.045 ^{***} (0.010)	
<i>IND_LENDING</i>	0.018 ^{***} (0.002)		0.242 ^{***} (0.028)	
<i>BC_CROSSCHAIN</i>	0.002 (0.002)		0.003 (0.030)	
Observations	99,818	99,818	99,818	99,818
R-squared	0.014	0.238	0.013	0.251
Industry FE	YES	NO	YES	NO
Blockchain FE	YES	NO	YES	NO
Protocol FE	NO	YES	NO	YES
Year FE	YES	YES	YES	YES
Year	2020–2025	2020–2025	2020–2025	2020–2025

Notes: Robust standard errors clustered at the Industry × Blockchain level in parentheses. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table 7 — Continuation. Instrumental Variable (IV) Estimates: AI-Assisted Code Verification. This table reports two-stage least squares estimates evaluating the effect of engaging bounty programs on DeFi protocol security. The instrument is the interaction between GitHub activity and ChatGPT release.

$$\text{First Stage: } BOUNTY_{it} = \beta \cdot (GITHUB_i \times POSTChatGPT_t) + \alpha' \mathbf{X}_{it} + \mu_i + \delta_t + \varepsilon_{it},$$

$$\text{Second Stage: } HackOutcome_{it+6} = \delta \cdot \widehat{BOUNTY}_{it} + \theta' \mathbf{X}_{it} + \mu_i + \delta_t + u_{it}$$

The outcome variable $HackOutcome_{it+6}$ corresponds to either (i) a binary indicator equal to one if protocol i was hacked within six months of bounty adoption ($HACKDUM$), or (ii) the dollar amount of exploit losses conditional on a breach ($Hackloss$). The key explanatory variable $BOUNTY_{it}$ is a binary indicator equal to one if protocol i initiated a decentralized bounty audit program in month t . To mitigate endogeneity in bounty adoption, we use a two-stage instrumental variable approach. The first stage instruments $BOUNTY_{it}$ using the interaction between GitHub development intensity and the introduction of ChatGPT ($GITHUB \times POSTChatGPT$), which proxies for AI-assisted awareness and bounty implementation. The second stage estimates the effect of predicted bounty adoption on subsequent hacking outcomes. All regressions include protocol and time fixed effects, along with time-varying protocol characteristics. Standard errors are clustered at the industry-by-blockchain level.

VARIABLES	(b) Instrumental Variable		
	(1) <i>BOUNTY</i> First Stage	(2) <i>HACKDUM</i> Second Stage	(3) <i>Hackloss</i> Second Stage
<i>IV = (GITHUB × POSTChatGPT)</i>	0.024 *** (0.003)		
<i>BOUNTY</i>		-0.279 ** (0.132)	-1.415 ** (0.584)
<i>POSTChatGPT</i>	0.003 *** (0.001)	0.001 (0.001)	0.000 (0.001)
Observations	99,818	99,818	99,818
R-squared	0.874	0.238	0.245
Controls	YES	YES	YES
protocol FE	YES	YES	YES
Year FE	YES	YES	YES
Weak-IV Diagnostics (F-statistic)	62.21, $p = 0.000$		

Notes: Robust standard errors clustered at the Industry × Blockchain level. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table 8. Post-Hack Switch to Centralized Auditors. This table reports logit regression estimates of the likelihood that a protocol switches auditors. The dependent variables are binary indicators for the following outcomes: *SWITCH* (Column 1), *T to T* (Column 2), *T to B* (Column 3), *B to T* (Column 4), *B to B* (Column 5), *N to T* (Column 6), and *N to B* (Column 7). These columns distinguish transitions between tiers: e.g., *T to T* indicates staying with a top-tier auditor, *B to T* indicates moving from a bottom-tier to a top-tier auditor, and *N to T* indicates moving from a non-audit to a top-tier auditor.

$$SWITCH_{it} = \beta_1 HACKED_i + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_{it},$$

The dependent variable *SWITCH*_{*it*} is a binary indicator for whether protocol *i* switched auditors, selected a top auditor, or selected a bottom auditor in period *t*. The main explanatory variable *HACKED* equals one in periods following a major security breach. All regressions include the full set of controls from Table 2 and fixed effects for industry, blockchain, and year.

VARIABLES	(1) <i>SWITCH</i>	(2) <i>T to T</i>	(3) <i>T to B</i>	(4) <i>B to T</i>	(5) <i>B to B</i>	(6) <i>N to T</i>	(7) <i>N to B</i>
<i>HACKED</i>	0.605*** (0.208)	0.122 (0.486)	0.294 (0.386)	1.293*** (0.445)	0.496* (0.298)	1.582*** (0.603)	-0.665 (0.629)
<i>ORACLE</i>	0.476** (0.243)	1.059*** (0.384)	0.480 (0.382)	-0.156 (0.653)	0.662* (0.386)	-1.007 (0.832)	-0.080 (0.531)
<i>DAO</i>	0.353 (0.222)	0.311 (0.460)	0.405 (0.390)	0.569 (0.828)	0.047 (0.372)	-0.805 (1.196)	0.711 (0.474)
<i>log_TVL</i>	0.031 (0.038)	0.122 (0.096)	0.028 (0.094)	0.153 (0.104)	-0.050 (0.043)	0.056 (0.095)	0.109 (0.092)
<i>log_Chains</i>	0.557** (0.257)	0.893 (0.552)	0.471 (0.488)	-0.335 (0.686)	0.608* (0.362)	-1.533* (0.850)	1.146* (0.682)
<i>log_Staking</i>	0.031* (0.017)	0.064* (0.034)	0.021 (0.034)	-0.054 (0.051)	0.020 (0.025)	0.083*** (0.032)	0.020 (0.031)
<i>log_Raised</i>	0.022 (0.020)	0.096*** (0.033)	0.007 (0.035)	0.048 (0.043)	-0.058 (0.052)	0.053 (0.073)	0.010 (0.045)
<i>GITHUB</i>	-1.169 (1.081)	-0.295 (1.676)	-0.727 (1.891)	0.039 (1.740)	-5.021 (5.251)		1.364 (1.302)
<i>log_Commits</i>	0.107 (0.119)	-0.017 (0.184)	0.147 (0.202)	0.069 (0.162)	0.472 (0.562)		-0.296*** (0.104)
<i>log_Followers</i>	0.084*** (0.030)	0.067 (0.059)	0.103*** (0.034)	0.051 (0.059)	0.089* (0.052)	0.027 (0.072)	0.048 (0.082)
<i>LISTED</i>	-0.157 (0.213)	0.246 (0.512)	0.118 (0.402)	-1.422** (0.606)	-0.052 (0.400)	-0.110 (0.980)	-0.454 (0.975)
<i>log_EthereumTVL</i>	0.125 (0.249)	0.989* (0.524)	0.522 (0.538)	-0.139 (0.635)	0.073 (0.506)	-0.835 (0.725)	-0.389 (0.458)
<i>IND_LENDING</i>	-0.184 (0.203)	-1.291 (0.856)	0.580 (0.425)	0.055 (0.621)	-0.114 (0.332)	-0.116 (1.065)	-1.258 (1.263)
<i>BC_CROSSCHAIN</i>	-0.270 (0.286)	-0.616 (0.740)	0.483 (0.634)	0.694 (0.796)	-0.042 (0.404)	0.515 (0.880)	-2.008** (0.808)
Observations	1,179	1,179	1,179	1,179	1,179	1,051	1,179
Pseudo R-squared	0.05	0.16	0.09	0.11	0.07	0.18	0.09
Industry FE	YES	YES	YES	YES	YES	YES	YES
Blockchain FE	YES	YES	YES	YES	YES	YES	YES
Year FE	YES	YES	YES	YES	YES	YES	YES

Notes: Standard errors in parentheses are clustered at the Industry × Blockchain level. This sample includes only protocols whose 90-day TVL is larger than the sample median. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Table 9. Post-Hack Adoption of Bounty Programs. This table reports logit regression estimates of the likelihood that a protocol engages a decentralized bounty program. The dependent variables are binary indicators: *BOUNTY* (Column 1), *switch_non_bounty* (Column 2: from no audit to bounty), *switch_low_bounty* (Column 3: from bottom-tier audit to bounty), and *switch_top_bounty* (Column 4: from top-tier audit to bounty). Key regressors include breach exposure, protocol risk characteristics, and control variables. *HACKED* equals one if the protocol experienced a major security breach. Control variables include protocol complexity (e.g., Cross-chain deployment, oracle use), financial resources (funding raised), activity (staking, commits), and governance structure. All models include industry, blockchain, and year fixed effects. Standard errors are clustered at the Industry \times Blockchain level.

	(1) <i>BOUNTY</i>	(2) <i>NON TO BOUNTY</i>	(3) <i>BOTTOM TO BOUNTY</i>	(4) <i>TOP TO BOUNTY</i>
<i>HACKED</i>	1.020 ^{***} (0.180)	0.441 (0.278)	0.914 ^{***} (0.166)	0.440 [*] (0.262)
<i>ORACLE</i>	0.553 ^{***} (0.158)	-0.110 (0.385)	0.629 ^{**} (0.293)	0.389 [*] (0.213)
<i>DAO</i>	-0.135 (0.200)	-0.271 (0.455)	-0.150 (0.293)	0.216 (0.216)
<i>log_TVL</i>	-0.011 (0.027)	-0.016 (0.037)	-0.059 ^{**} (0.020)	0.040 (0.050)
<i>log_Chains</i>	0.590 [*] (0.353)	-0.049 (0.365)	0.243 (0.353)	0.559 [*] (0.293)
<i>log_Staking</i>	0.021 (0.017)	-0.044 [*] (0.024)	0.017 (0.020)	0.040 ^{**} (0.020)
<i>log_Raised</i>	0.039 ^{***} (0.015)	0.026 (0.042)	-0.022 (0.019)	0.064 ^{***} (0.020)
<i>GITHUB</i>	-0.422 (1.127)	-2.254 (1.946)	-1.080 (1.825)	1.008 (1.008)
<i>log_Commits</i>	0.057 (0.124)	0.226 (0.198)	0.079 (0.202)	-0.068 (0.119)
<i>log_Followers</i>	0.124 ^{***} (0.029)	0.068 [*] (0.040)	0.161 ^{***} (0.032)	0.037 (0.061)
<i>LISTED</i>	0.195 (0.175)	0.250 (0.314)	-0.015 (0.259)	0.211 (0.198)
<i>log_EthereumTVL</i>	-0.741 ^{***} (0.216)	-0.651 ^{**} (0.320)	-0.794 ^{***} (0.256)	-0.250 (0.250)
<i>IND_LENDING</i>	-0.223 (0.156)	-0.566 (0.383)	0.329 (0.272)	-0.425 [*] (0.256)
<i>BC_CROSSCHAIN</i>	0.119 (0.313)	-0.040 (0.262)	0.326 (0.309)	0.344 (0.379)
Observations	1,179	1,179	1,179	1,179
Pseudo R-squared	0.12	0.08	0.15	0.13
Industry FE	YES	YES	YES	YES
Blockchain FE	YES	YES	YES	YES
Year FE	YES	YES	YES	YES

Notes: Standard errors in parentheses are clustered at the Industry \times Blockchain level. This sample includes only protocols whose 90-day TVL is larger than the sample median. ^{*} $p < 0.1$, ^{**} $p < 0.05$, ^{***} $p < 0.01$.

Table 10. Impact of Security Breaches on Auditor Reputation. This table reports OLS regression results estimating the effect of security breaches on auditor market share at different horizons. Panel A presents equally-weighted outcomes; Panel B reports value-weighted outcomes using protocol Total Value Locked (TVL); and Panel C examines breach severity using the log of hack-related losses. The dependent variable is the monthly market share of each auditor (ranging from 0 to 1). All regressions include auditor and year-month fixed effects.

$$MarketShare_{it+h} = \beta HACKED_{it} + \alpha_i + \delta_t + \varepsilon_{it}$$

The dependent variable $MarketShare_{it+h}$ measures the share of audits held by auditor i at horizon $t + h$ (where $h \in \{3, 6, 12\}$ months after a breach). $HACKED_{it}$ in Panels A and B is an indicator equal to one in the months following a major platform hack. Panel C replaces the indicator with $Hackloss$, defined as the log of USD-denominated losses from the breach. All regressions include auditor fixed effects (α_i) and year-month fixed effects (δ_t). Standard errors are clustered at the auditor level.

Panel A. Market Share (TVL equal-weighted)			
	(1) $t + 3$	(2) $t + 6$	(3) $t + 12$
<i>HACKED</i>	-0.042** (0.015)	0.016 (0.013)	-0.016 (0.018)
Observations	264	246	210
R-squared	0.452	0.445	0.457
Auditor FE	YES	YES	YES
Year-mon FE	YES	YES	YES
Panel B. Market Share (TVL value-weighted)			
	(1) $t + 3$	(2) $t + 6$	(3) $t + 12$
<i>HACKED</i>	-0.042* (0.016)	0.022 (0.015)	-0.022 (0.019)
Observations	230	225	220
R-squared	0.442	0.433	0.444
Auditor FE	YES	YES	YES
Year-mon FE	YES	YES	YES
Panel C. Market Share by Hack Loss Severity			
	(1) $t + 3$	(2) $t + 6$	(3) $t + 12$
<i>Hackloss</i>	-0.002* (0.001)	0.001 (0.001)	-0.001 (0.001)
Observations	264	246	210
R-squared	0.455	0.426	0.384
Auditor FE	YES	YES	YES
Year-mon FE	YES	YES	YES

Notes: Standard errors are clustered at the auditor level. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$. Top auditors included: Certik, Peckshield, Hacken, Quantstamp, Slowmist, and Halborn.

Appendices for “Auditing Smart Contracts”

Landsman et. al (2025)”

Appendix A — Complementary Tables

This appendix provides background and reference material that complements the main text’s discussion of *centralized pre-launch audits* and *decentralized post-launch bug-bounty programs*. In normal times, bounties typically *complement* audits by extending coverage into the live environment, where integrations, parameter updates, and composability can create vulnerabilities that scoped pre-launch reviews may not observe. Teams frequently open or expand bounties around routine but security-relevant milestones (e.g., major version releases or migrations, chain expansions/bridges, governance parameter changes, internal QA flags, or community-raised issues on forums/GitHub). In the immediate aftermath of large shocks—when re-auditing may be slow or costly—bounties can *substitute at the margin*, providing rapid, flexible hardening until fuller centralized reviews are feasible (see Section 4.3). Bounty programs are commonly run through platforms such as *Immunefi* and *Code4rena*, with performance-based rewards that scale with severity. Appendix Table A1 contrasts centralized audits and decentralized audits (bounties) across several features. In brief, audits are pre-deployment, fixed-fee, and conducted by professional firms that issue structured reports used for credibility signaling (e.g., [Rabetti \(2023\)](#); [Bourveau et al. \(2024\)](#); [Bhambhwani and Huang \(2024\)](#); [Knechel et al. \(2025\)](#)); bounty programs are continuous, pay-per-vulnerability, operate on live systems, and leverage a global pool of independent researchers with publicly verifiable outputs. These differences explain the widespread adoption of hybrid strategies that combine ex-ante assurance and reputation benefits (audits) with ongoing, crowdsourced risk mitigation (bounties).

Appendix Table A1. Smart Contract Audits: Centralized vs Decentralized (Bounty Programs)

Feature	Smart Contract Audits	Bug Bounty Programs
Timing	Pre-deployment	Post-deployment
Duration	2–4 weeks	Continuous
Cost	Fixed fee	Pay-per-vulnerability
Scope	Focused code review	Broad attack surface
Participant Pool	Professional auditors	Global security researchers
Output	Detailed audit report	Individual vulnerability reports
Testing Environment	Controlled test environment	Live production system
Reputation	Unobservable: Word of Mouth	Observable: Public Repositories

Appendix Table A2: Definition of variables.

VARIABLES	Definition
(a) Audit choice	
<i>AUDIT</i>	An indicator variable equal to 1 if the protocol engaged at least one auditor, and 0 otherwise.
<i>TOP</i>	An indicator variable equal to 1 if the protocol engaged a top-tier centralized auditor (e.g., a recognized large audit firm), and 0 otherwise.
<i>BOTTOM</i>	An indicator variable equal to 1 if the protocol engaged a bottom-tier centralized auditor, and 0 otherwise.
<i>BOUNTY</i>	An indicator variable equal to 1 if the protocol engages a decentralized bounty program for security auditing, and 0 otherwise.
<i>SWITCH</i>	An indicator equal to 1 if, in period t , protocol i switched auditors, and 0 otherwise.
<i>MarketShare</i>	The share of audits performed by auditor i at horizon $t + h$, where $h \in \{3, 6, 12\}$.
(b) Protocol characteristics	
<i>ORACLE</i>	An indicator variable equal to 1 if the protocol uses external price feeds or off-chain data sources, and 0 otherwise.
<i>LISTED</i>	An indicator variable equal to 1 if the protocol is listed on a cryptocurrency exchange, and 0 otherwise.
<i>log_TVL</i>	The natural logarithm of the total value locked (TVL) in USD, measured one day post the protocol's launch.
<i>DAO</i>	An indicator variable equal to 1 if the protocol employs decentralized governance, and 0 otherwise.
<i>GITHUB</i>	An indicator variable equal to 1 if the protocol maintains at least one publicly accessible code repository on GitHub at launch, and 0 otherwise.
<i>log_Commits</i>	The natural logarithm of one plus the total number of code commits submitted through GitHub at the protocol's launch
<i>log_Chains</i>	The natural logarithm of one plus the number of distinct blockchains on which the protocol is deployed at launch.
<i>log_Raised</i>	The natural logarithm of one plus the cumulative amount of external funding (in USD) raised by the protocol prior to launch.
<i>log_Staking</i>	The natural logarithm of one plus the total value of assets (in USD) staked within the protocol at launch.
<i>log_Followers</i>	The natural logarithm of one plus the number of Twitter followers of the protocol's official account at launch.
(c) Industry and blockchain characteristics	
<i>IND_LENDING</i>	An indicator variable equal to 1 if the protocol operates primarily as a lending or borrowing platform, and 0 otherwise.
<i>IND_DEXES</i>	An indicator variable equal to 1 if the protocol functions primarily as a decentralized exchange (DEX), and 0 otherwise.
<i>IND_YIELD</i>	An indicator variable equal to 1 if the protocol's main business model is yield aggregation or yield optimization, and 0 otherwise.
<i>BC_ETHEREUM</i>	An indicator variable equal to 1 if the protocol is deployed on the Ethereum blockchain, and 0 otherwise.
<i>BC_CROSSCHAIN</i>	An indicator variable equal to 1 if the protocol is deployed across multiple distinct blockchains (i.e., cross-chain deployment), and 0 otherwise.
<i>log_EthereumTVL</i>	The natural logarithm of the total value locked (TVL) in USD on the Ethereum blockchain.
(d) Hack outcomes	
<i>HACKDUM</i>	An indicator variable equal to 1 if the protocol experienced at least one hack post-launch, and 0 otherwise.
<i>Hackloss</i>	The natural logarithm of one plus the total monetary losses from hacks (in USD millions).

Appendix Table A3: Covariate Balance for Audit vs. Non-Audit Protocols. This table reports means of baseline DeFi protocol characteristics for audited and non-audited DeFi protocols, both before and after 1:1 propensity score matching (PSM). The last column in each panel reports the mean difference between the two groups. Standard errors are in parentheses. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

VARIABLES	Before			After		
	Non-audit	Audit	Diff.	Non-audit	Audit	Diff.
<i>ORACLE</i>	0.115 (0.319)	0.248 (0.432)	0.132*** (0.011)	0.180 (0.385)	0.171 (0.376)	-0.010 (0.014)
<i>DAO</i>	0.066 (0.249)	0.120 (0.325)	0.054*** (0.008)	0.082 (0.275)	0.072 (0.258)	-0.011 (0.010)
<i>log_TVL</i>	11.478 (4.344)	12.916 (4.020)	1.439*** (0.131)	12.448 (3.875)	12.351 (3.960)	-0.097 (0.147)
<i>log_Chains</i>	0.810 (0.312)	0.968 (0.463)	0.158*** (0.012)	0.874 (0.378)	0.863 (0.364)	-0.011 (0.014)
<i>log_Staking</i>	1.641 (4.094)	2.462 (4.930)	0.821*** (0.133)	1.400 (3.782)	1.195 (3.380)	-0.206 (0.134)
<i>log_Raised</i>	0.495 (2.677)	1.124 (3.947)	0.629*** (0.098)	0.621 (2.993)	0.537 (2.764)	-0.084 (0.108)
<i>log_Followers</i>	5.786 (4.111)	6.587 (3.807)	0.801*** (0.118)	6.330 (4.021)	6.251 (3.856)	-0.079 (0.148)
<i>log_Commits</i>	0.368 (1.794)	0.886 (2.674)	0.518*** (0.066)	0.431 (1.922)	0.354 (1.748)	-0.076 (0.069)
<i>LISTED</i>	0.094 (0.293)	0.187 (0.390)	0.092*** (0.010)	0.089 (0.285)	0.088 (0.283)	-0.001 (0.011)
<i>log_EthereumTVL</i>	24.469 (0.491)	24.548 (0.504)	0.079*** (0.015)	24.488 (0.486)	24.502 (0.505)	0.014 (0.019)
<i>IND_LENDING</i>	0.076 (0.265)	0.126 (0.332)	0.050*** (0.009)	0.108 (0.311)	0.105 (0.307)	-0.003 (0.012)
<i>BC_ETHEREUM</i>	0.092 (0.290)	0.130 (0.336)	0.038*** (0.009)	0.119 (0.323)	0.117 (0.322)	-0.001 (0.012)
<i>BC_CROSSCHAIN</i>	0.164 (0.371)	0.337 (0.473)	0.173*** (0.012)	0.245 (0.430)	0.229 (0.421)	-0.015 (0.016)
Observations	2,509	2,060	4,569	1,425	1,425	2,850

Appendix Table A4: Audit Type and Likelihood of Hacks Post-Launch. This equation estimates the effect of auditor type on the probability that a protocol experiences a hack in various post-launch windows. The regressions are repeated across 30-, 90-, 180-, and 365-day intervals after launch.

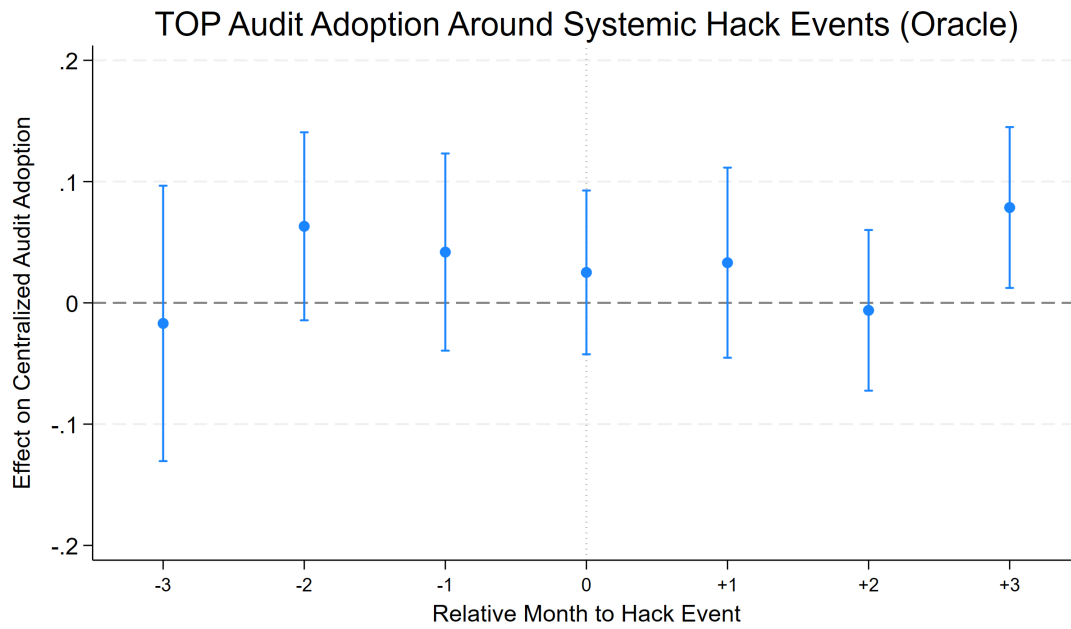
$$HACKDUM_{it+1} = \beta_1 \text{ AuditorType}_{it} + \mathbf{X}'_{it}\gamma + \gamma_j + \lambda_k + \delta_t + \varepsilon_{it}$$

The dependent variable $HACKDUM_{it}$ is a binary indicator equal to 1 if the protocol is hacked within the specified post-launch window. The variable AuditorType_{it} captures whether the protocol was audited by a top-tier, bottom-tier, or unaudited entity. Control variables \mathbf{X}_{it} include protocol-level characteristics described in Table 1. γ_t denotes year-month fixed effects, λ_j industry fixed effects, and δ_k blockchain fixed effects. Standard errors are robust to heteroskedasticity and clustered at the Industry \times Blockchain level.

VARIABLES	(1) 30 Days	(2) 90 Days	(3) 180 Days	(4) 365 Days
<i>BOTTOM</i>	0.002 (0.002)	-0.004 (0.003)	0.004 (0.005)	-0.000 (0.003)
<i>TOP</i>	0.005** (0.003)	0.002 (0.003)	0.003 (0.004)	-0.003 (0.002)
<i>log_TVL</i>	0.008*** (0.002)	0.002* (0.001)	-0.002 (0.002)	0.001 (0.003)
<i>ORACLE</i>	0.001 (0.007)	0.003 (0.005)	0.008 (0.007)	-0.003 (0.003)
<i>log_Chains</i>	-0.008 (0.007)	0.003 (0.006)	0.014** (0.005)	-0.005 (0.008)
<i>log_Staking</i>	-0.000 (0.000)	-0.000*** (0.000)	-0.000 (0.001)	0.000 (0.000)
<i>log_Raised</i>	-0.000 (0.000)	0.000 (0.000)	0.001 (0.001)	-0.000 (0.000)
<i>log_Commits</i>	-0.000 (0.000)	-0.000 (0.000)	0.002 (0.001)	0.000 (0.001)
<i>LISTED</i>	0.002 (0.004)	0.000 (0.003)	0.005 (0.008)	0.001 (0.006)
<i>log_Followers</i>	-0.001** (0.000)	-0.000 (0.000)	-0.000 (0.000)	0.000 (0.000)
<i>DAO</i>	-0.006 (0.004)	0.010 (0.007)	0.013 (0.013)	0.007 (0.006)
<i>log_EthereumTVL</i>	-0.010 (0.016)	-0.015 (0.014)	0.002 (0.020)	0.003 (0.030)
Observations	4,049	3,918	4,049	4,049
R-squared	0.024	0.028	0.032	0.031
Industry FE	YES	YES	YES	YES
Blockchain FE	YES	YES	YES	YES
Year-mon FE	YES	YES	YES	YES

Notes: Robust standard errors clustered at the Industry \times Blockchain level in parentheses. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Figure A1. Dynamic effect of systemic hack events on centralized top-tier audit adoption for Oracle protocols. This figure plots coefficient estimates from the stacked difference-in-differences specification on Table 3, where the dependent variable is an indicator equal to one if a protocol adopts a centralized top-tier auditor in a given month. Each point represents the estimated effect for a given month relative to the hack event (month 0), controlling for protocol characteristics, industry and blockchain fixed effects, and protocol launch-month fixed effects. Vertical bars denote 90% confidence intervals based on standard errors clustered at the protocol-shock level. The dashed horizontal line marks zero effect.



Appendix B — Example of an Audit Report



Binance zk-SNARKs Proof of Solvency Independent Technical Assessment

Feb 14, 2023

Repositories:

<https://github.com/binance/zkmerkle-proof-of-solvency>

Commit:

c1884aae22cd17af023ac4424b4e6623eb0ea9dd

References:

- [Announcement](#)
 - [How to Verify Your Account Balance on Binance](#)
 - [How zk-SNARKs Improve Binance's Proof of Reserves System](#)
 - [Proof of solvency - technical specification](#)
 - [Having a safe CEX: proof of solvency and beyond](#)
-

Authors:

Luciano Ciattaglia (l.ciattaglia@hacken.io)
Bartosz Barwikowski (b.barwikowski@hacken.io)
Yaroslav Bratashchuk (y.bratashchuk@hacken.io)
Sofiane Akermoun (s.akermoun@hacken.io)

Example of Audit Report for Binance PoS tech assessment—Continuation. This figure depicts a sample from a smart contract auditing report for Binance Proof-of-Stake.



2 Project Summary

In the project we identified 1 critical issue which allows to fake the total debt amount in the zero knowledge proof circuit, 1 medium severity issue and 2 other low severity issues. The critical and medium severity issues have been already fixed. However, any proof generated before those issues were fixed cannot be verified to be valid, as the critical one allowed for the total debt amount to be tampered. Although the proofs may appear to be valid, it is not possible to ensure that they were not modified due to the vulnerability. The other low severity issues are very unlikely to be abused and do not need to be addressed immediately.

The project has 1157 dependencies, all of them with checksum verification. There were found 42 vulnerabilities within all dependencies, with 16 of them having public exploits available. 22 with high severity and 20 with medium. None of the vulnerable functions are currently being used in the project.

It uses a [forked version of gnark](#) made on Sep 2022 for the circuits and [Poseidon](#) with BN254 hash function to hash the user information and the Sparse Merkle Tree (SMT) data structure to store the hashes. The SMT is implemented using the [BSMT](#) library, and its maximum depth is set to be 28, which means that this Proof Of Solvency approach may be used for more than 250M users.

The code quality is clean and organized.

The README.md contains instructions on how to run tools one by one, and motivation behind the circuits is also [detailed](#).

The [Panic](#) is used for main function error handling, so all the tools crash with a stack trace in case of an error.

The sample user data (balance sheets) is provided in order to test tools manually. There is a way to fetch (probably production) Postgres configuration from the AWS storage if the `remote_password_config` flag is provided to the tools that use Postgres.

There was a [function to generate fake accounts](#) in the witness service, which was commented out but still left in the code (probably for manual testing purposes). EmptyAccounts are generated [in the witness](#), and they are used in case the last account's batch size is less than 864.

Example of Audit Report for Binance PoS tech assessment—Continuation. This figure depicts a sample from a smart contract auditing report for Binance Proof-of-Stake.



The git history log has been modified several times, and as a result, the git metadata is mixed in some places.

3 Vulnerabilities

3.1 [Critical][Fixed] TotalDebt manipulation vulnerability caused by overflow of BasePrice

The code contains a critical error that enables it to create false user debt, reducing the number of assets needed. This occurs because there is a method to circumvent the assertion that checks if the user's debt exceeds their equity.

There is a bug in the system that allows for bypassing because the BasePrice parameter can be set to an extremely high value. This vulnerability exists because the parameter is not checked for value range, making it easy to manipulate. Although the BasePrice is publicly accessible, it would be simple to identify if it has been changed. However, there is a method to modify the BasePrice in a way that would be undetectable by other users, making it possible to exploit the vulnerability without being detected.

As an optimization, the code splits all the users into batches, each with 864 users. The batches are linked with each other by sharing information about assets and the cryptographic hashes. Each exchange asset is shared using three variables: TotalEquity, TotalDebt and BasePrice. The hash of asset is calculated from one big integer, which is calculated using the following formula:

$$TotalEquity * 2^{128} + TotalDebt * 2^{64} + BasePrice$$

The problem is, that in the code responsible for doing these calculations, only TotalEquity and TotalDebt are checked if they are greater or equal to 0 and lower than 2^{64} . The value of BasePrice is not being checked, which allows to set it to value higher than $2^{64} - 1$ which makes it possible to modify the value of TotalDebt and TotalEquity. Because of that, it is possible to generate the same value for different parameters, for example both TotalDebt = 2, BasePrice = 3 and TotalDebt = 1, BasePrice = $2^{64} + 3$ will have value of $2 * 2^{64} + 3$. The source code responsible for this calculations:

Example of Audit Report for Binance PoS tech assessment—Continuation. This figure depicts a sample from a smart contract auditing report for Binance Proof-of-Stake.

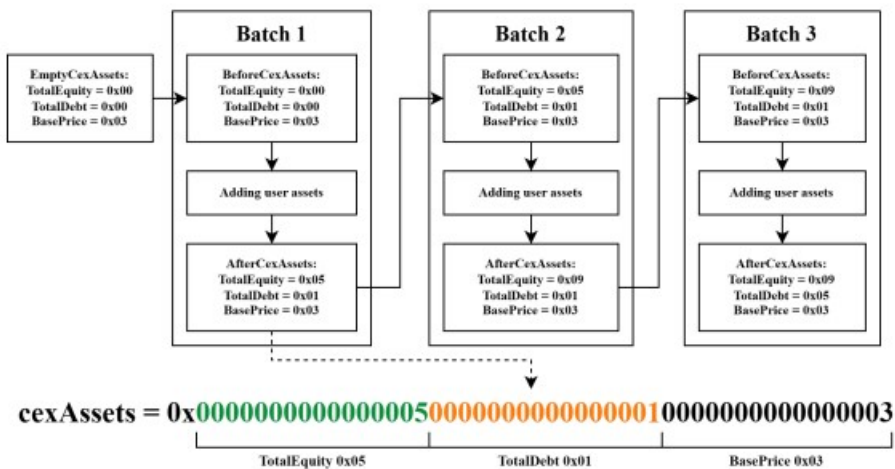


```
// verify whether beforeCexAssetsCommitment is computed correctly
for i := 0; i < len(b.BeforeCexAssets); i++ {
    CheckValueInRange(api, b.BeforeCexAssets[i].TotalEquity)
    CheckValueInRange(api, b.BeforeCexAssets[i].TotalDebt)
    cexAssets[i] = api.Add(api.Mul(b.BeforeCexAssets[i].TotalEquity, utils.Uint64MaxValueFrSquare),
        api.Mul(b.BeforeCexAssets[i].TotalDebt, utils.Uint64MaxValueFr), b.BeforeCexAssets[i].BasePrice)
    afterCexAssets[i] = b.BeforeCexAssets[i]
}
actualCexAssetsCommitment := poseidon.Poseidon(api, cexAssets...)
api.AssertIsEqual(b.BeforeCexAssetsCommitment, actualCexAssetsCommitment)
```

The lack of validation of *BasePrice* allows it to be modified between batches, by lowering the *TotalDebt* by 1, the *BasePrice* can be increased by 2^{64} and vice versa. Because of that, it is possible to generate almost unlimited debt. A user with 1 coin with *BaseValue* greater than 2^{64} (million of dollars) can have almost any debt, the assertion responsible for checking if users have lower debt than equity won't work correctly.

It is possible to generate the debt without anyone noticing it, it is possible by creating a batch of 864 fake users with huge debt but also with a single coin with modified *BaseValue*, which will cover the whole debt. The below diagrams demonstrate how the value of *BasePrice* can be modified in a single batch.

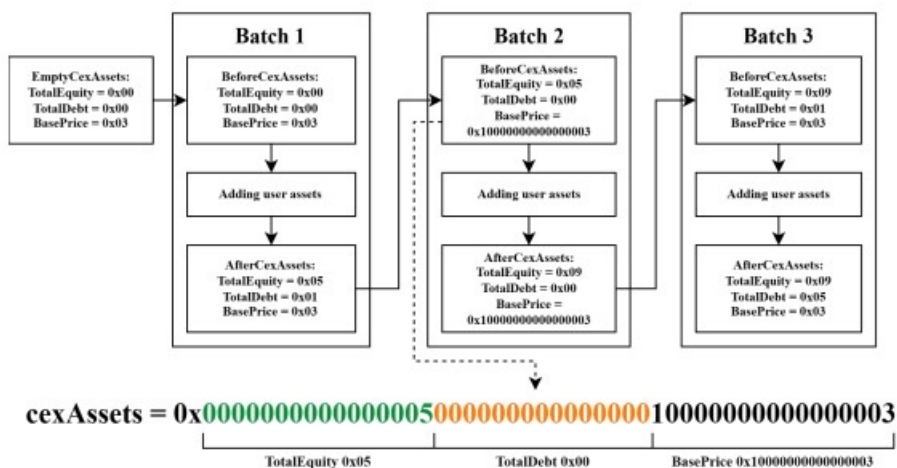
Correct batch calculation



Example of Audit Report for Binance PoS tech assessment—Continuation. This figure depicts a sample from a smart contract auditing report for Binance Proof-of-Stake.



Batch calculation with BasePrice overflow



Hacken team created a dedicated repository to prove the existence of this issue: <https://github.com/hknio/zkmerkle-proof-of-solvency-debt-bug>.

The issue with the fix proposal was reported to Binance Team which confirmed the issue and merged fix proposed by Hacken team: <https://github.com/binance/zkmerkle-proof-of-solvency/pull/5>.

3.2 [Medium][Fixed] TotalDebt value underflow caused by integer overflow

When *TotalEquity* and *TotalDebt* is calculated from user assets, it is possible that it becomes bigger than 2^{64} , an example case is when two users have both 2^{63} debt and equity, then the sum of their debt and equity will be equal to 2^{64} . The code responsible for the calculations:

```

for j := 0; j < len(userAssets); j++ {
    CheckValueInRange(api, userAssets[j].Debt)
    CheckValueInRange(api, userAssets[j].Equity)
    totalUserEquity = api.Add(totalUserEquity, api.Mul(userAssets[j].Equity, b.BeforeCexAssets[j].BasePrice))
    totalUserDebt = api.Add(totalUserDebt, api.Mul(userAssets[j].Debt, b.BeforeCexAssets[j].BasePrice))

    afterCexAssets[j].TotalEquity = api.Add(afterCexAssets[j].TotalEquity, userAssets[j].Equity)
    afterCexAssets[j].TotalDebt = api.Add(afterCexAssets[j].TotalDebt, userAssets[j].Debt)
}

```

Example of Audit Report for Binance PoS tech assessment—Continuation. This figure depicts a sample from a smart contract auditing report for Binance Proof-of-Stake.



When the value of *TotalEquity* or *TotalDebt* will become higher than 2^{64} , then the next part of code, responsible for calculating integer used by hash function (*tempAfterCexAssets*) will work incorrectly because of overflows:

```
for j := 0; j < len(tempAfterCexAssets); j++ {
    tempAfterCexAssets[j] = api.Add(api.Mul(afterCexAssets[j].TotalEquity, utils.Uint64MaxValueFrSquare),
        api.Mul(afterCexAssets[j].TotalDebt, utils.Uint64MaxValueFr), afterCexAssets[j].BasePrice)
}
```

When *TotalEquity* exceeds 2^{64} then the proof in the next batch will be incorrect, however when *TotalDebt* exceeds 2^{64} , then it will overflow into *TotalEquity*. For example, *TotalDebt* equal to exactly 2^{64} would be equivalent to *TotalEquity* equal 1 and *TotalDebt* equal 0. This allows to lower the value of *TotalDebt* in the similar way as it was done in the case of the first issue with *BasePrice*, however it would not be beneficial in any way so this issue is not critical.

We recommend adding additional *CheckValueInRange* for *TotalEquity* and *TotalDebt* when calculating *tempAfterCexAssets*.

The issue was addressed and fixed by Binance Team: <https://github.com/binance/zkmerkle-proof-of-solvency/pull/6>

3.3 [Low] Potential omission of users

The current system of verification lacks a mechanism to confirm the completeness of the provider's inclusion of their users in the Merkle Tree. It is uncertain whether the provider may have excluded some users, who they presume will either not perform a verification of the proof or whose objections, in the event that they do not receive a proof, will not be given due consideration.

In the current implementation, the prover knows which users do the verification process as they need to download the configuration files from their website. Simplifying the process of choosing which users should be included and which ones can be omitted.

While unlikely this would happen in practice, to address this issue, it is necessary a trusted third party, as they become more readily available to support crypto exchanges, must verify that all users were included in the Merkle Tree without any exclusions.

Example of Audit Report for Binance PoS tech assessment—Continuation. This figure depicts a sample from a smart contract auditing report for Binance Proof-of-Stake.



3.4 [Low] Merkle Root hash integrity

When users download the Merkle tree and each user config from the frontend, the Merkle root hash is included in the user_config.json file, but there is no way to check the integrity of this hash across all Binance users in order to be sure that this root hash wasn't tampered depending on the client IP or other parameters of the users.

```
{
  "AccountIndex": 9,
  "AccountIdHash": "0000041cb7323211d0b356c2fe6e79fdaf0c27d74b3bb1a4635942f9ae92145b",
  "Root": "29591ef3a9ed02605edd6ab14f5dd49e7dbe0d03e72a27383f929ef3efb7514f",
  "Assets": [{"Index":7,"Equity":123456000,"Debt":0}],
  "Proof": ["DrPpFsm4/5HntRTf8M3dbgpdrxq3Q81Zk02ngysw2js=", "G1WgD/CvmGApQgmIX0rE0B1Sifkw6IfNwY
  "TotalEquity": 123456000,
  "TotalDebt": 0
}
```

To counteract this, the Merkle root should be signed by a trusted third-party auditor or be published on the blockchain as a public bulletin board, so users can easily verify the transaction's inclusion and the validity of the Merkle root hash they got from their user_config.json. It should be done in a single transaction, which will be easy to detect. It's also possible to address this issues by publishing the hash root in a social media that the proved doesn't control.

3.5 [Informational] Total amount of users inference

The user downloads a proof.csv file in the verification config containing the total amount of batches and their commitments. The current number of batches at the moment of this assessment is 49.789. If we multiply this by the number of users per batch (currently 864), we can infer that the total number of users is around 43.015.104, as one leaf is equal to one user in the current implementation. Still, this amount can also include a number of empty leaves, so it's only an approximation. Randomizing the number of empty leaves in bigger numbers can solve this issue.

Appendix C — Bounty Hunters: Extended

Smart contract **bounty hunters** – often called *white hat* hackers or security researchers – are individuals or teams who seek out vulnerabilities in blockchain applications and smart contracts in exchange for rewards. Instead of exploiting bugs maliciously, these researchers responsibly disclose flaws through structured programs (bug bounties or audit contests) so that projects can fix issues before any exploit occurs. In essence, bounty programs “*invite the entire world to review your code and report vulnerabilities in exchange for a reward, instead of suffering an exploit*”⁴⁰. This open approach incentivizes skilled hackers (including potential black hats) to choose “clean” bounty payouts over illicit gains, aligning their interests with the security of decentralized finance (DeFi) protocols.

Bounty hunters operate via specialized platforms and communities. Two prominent platforms in DeFi security are **Code4rena** (<https://code4rena.com/>) and **Immunefi** (<https://immunefi.com/>). Code4rena organizes competitive *audit contests* in which dozens of auditors (called “wardens”) simultaneously review a project’s code over a short period for a share of a pre-funded prize pool. Immunefi, on the other hand, pioneered continuous *bug bounty programs* where projects offer rewards for any critical bug reports on an ongoing basis. In both cases, bounty hunters use their expertise to scrutinize smart contract code, develop proof-of-concept exploits, and submit detailed vulnerability reports. Successful reports can earn anywhere from a few hundred dollars for minor issues to seven-figure payouts for critical, high-impact discoveries. The process is mutually beneficial: researchers receive monetary rewards and community recognition, while projects gain security insights from a broad pool of talent, helping to protect user funds and prevent catastrophic hacks.

Bounty-Based Security vs Centralized Auditing

Bounty-driven security programs (such as Code4rena contests or Immunefi bounties) differ significantly from centralized audits by professional firms. Key differences include:

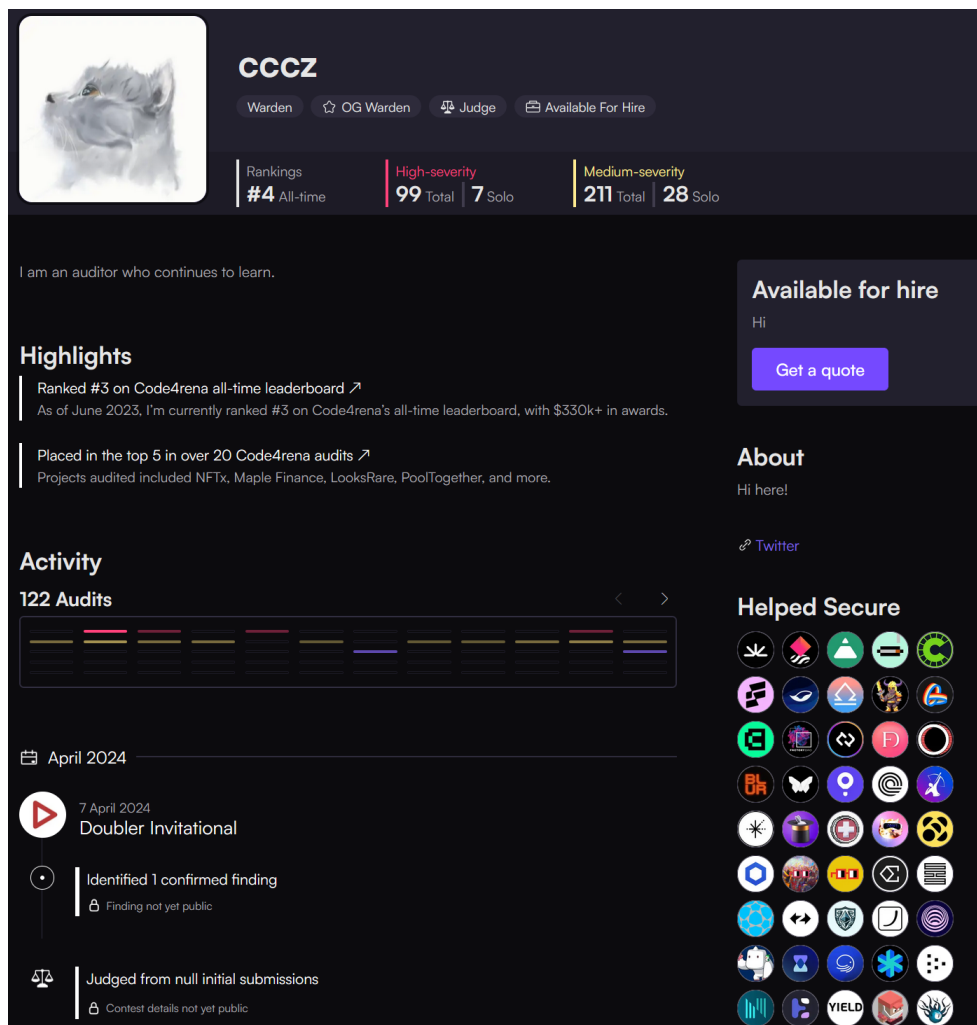
Cost and Payment Model: Traditional audits typically involve a fixed upfront fee paid to an auditing firm, regardless of how many bugs are found. In contrast, bounty-based approaches are results-driven: projects pay out rewards *only if* vulnerabilities are discovered⁴¹. This can make bounties more cost-effective, as unused reward funds remain with the project. For example, Code4rena contests use “conditional” prize pools – if no medium- or high-severity bugs are found, a large portion of the pool is refunded to the sponsor⁴². Immunefi similarly notes that its model is “*20% more cost-effective*” than comparable audit engagements. Essentially, a bounty program functions like an insurance policy: if no bugs are found, the cost stays low, and if critical bugs *are* found, the payout is justified by the averted damage.

⁴⁰<https://immunefi.com/>

⁴¹<https://immunefi.com/>

⁴²<https://code4rena.com/>

Appendix Figure Cx: The figure below shows an example of bounty hunter.



Incentive Structure: Bounty hunters are directly incentivized by the severity of bugs they uncover. The reward scales with impact, which strongly motivates researchers to hunt for critical vulnerabilities. In competitive audit contests, the scoring and payout system “*heavily favors critical vulnerabilities*”, sometimes giving an order of magnitude more reward for a critical vs. a medium issue⁴³. This can lead to focused effort on high-impact bugs (a boon for security) but may also mean less attention to low-risk issues or architectural insights that don’t qualify for bounties. Traditional auditors, being paid a fixed fee, aim for comprehensive coverage (including minor issues and best practices) as part of their mandate. Bounty platforms like Code4rena have introduced special allocations (e.g., a small “QA” reward pool for minor findings and optimizations) to encourage thoroughness beyond just critical bugs. Still, the competitive nature of contests means many participants prioritize the most lucrative findings first. The upside is that critical bugs rarely go unnoticed in a contest of dozens of skilled hackers racing to find the worst flaw.

Coverage and Effectiveness: Bounty-based models can provide broader coverage through “many eyes” on the code. A single

⁴³<https://code4rena.com/>

audit team might consist of 2–4 people, whereas a Code4rena contest can involve “*an average of 70 professional auditors per contest*” reviewing the codebase⁴⁴. This crowd-sourced scrutiny often translates to a higher volume of findings. Indeed, Code4rena’s contest approach has identified over 950 high-severity vulnerabilities across ~280 audits⁴⁵ – a testament to how multiple independent perspectives can uncover edge-case bugs that a small team might miss. Likewise, Immunefi’s open bug bounty community has proven effective at surfacing issues in live platforms. According to one report, “*8x more vulnerabilities are found on Immunefi compared with alternatives*”, with Immunefi’s white hat hackers collectively preventing an estimated millions of dollars in potential hack losses for Web3 projects⁴⁶. In short, an engaged community of independent researchers can often find more bugs in total than a lone audit team, especially over an extended period. However, not all those bug reports are high quality – bounty platforms must filter out false positives and duplicates. Traditional audits yield a curated report (with fewer but more vetted findings), whereas bounty programs generate many raw reports that require triage by experts or the project team.

Speed and Flexibility: Traditional audits need to be scheduled weeks or months in advance and typically take days or weeks to complete. In contrast, bounty approaches can move very fast. Code4rena advertises audit contests that can begin within 48 hours of a project signing up⁴⁷, and contests usually run just 3–7 days. This quick turnaround is valuable for projects on tight deployment timelines. Immunefi bug bounties are even more flexible – the program can be launched at any time and remains open indefinitely, continuously welcoming reports. This means security feedback can be continuous (post-launch) rather than one-and-done. The time-bound nature of contests also creates a sense of urgency that can compress what might be weeks of audit work into an intense few days of review.

Community Involvement and Transparency: Bounty-based security is inherently community-driven. Platforms like Code4rena and Immunefi have cultivated large communities of researchers (Code4rena alone has 10,000+ registered wardens as of 2023, while Immunefi hosts 35,000+ white hats worldwide). This decentralized talent pool means any given project’s code can be examined by specialists with diverse backgrounds – from crypto economics to low-level EVM hacking – increasing the chance to catch different classes of bugs. Community involvement also brings a level of public transparency and knowledge sharing. Code4rena publishes the results of each contest (usually a consolidated findings report), which often becomes a valuable public audit report for the platform. Immunefi encourages post-mortems and even awards “white hat Hall of Fame” NFTs to researchers who save projects from major hacks, thus publicizing success stories.

⁴⁴<https://code4rena.com/>

⁴⁵<https://code4rena.com/>

⁴⁶<https://immunefi.com/>

⁴⁷<https://code4rena.com/>

Code4rena: Competitive Auditing Contests

Code4rena is a leading platform for *competitive smart contract audits*, introducing a novel “warden” model in early 2021. In a Code4rena audit contest, a DeFi project (the sponsor) offers a fixed prize pool (often tens or hundreds of thousands of USDC) for a short-duration review of their smart contracts. During the contest (typically a few days long), any approved Code4rena warden (auditor) can study the code and submit bug reports. After the contest ends, an independent judge panel evaluates all submissions, assigns severities, and awards portions of the prize pool to the wardens who reported valid bugs. This model guarantees that the allocated bounty will be paid out to researchers (assuming any valid issues are found), rather than refunded to the project.

Code4rena’s competitive scene has produced some star wardens. For example, *Christoph Michel* (handle cmichel) became the first warden to surpass \$1,000,000 in total awards from contests⁴⁸. Another top warden, known as cccz, was ranked #3 all-time with \$330k+ in cumulative winnings and over 20 top-five contest finishes.

Immunefi: DeFi Bug Bounty Programs

Immunefi is the largest bug bounty platform in the Web3 ecosystem, focusing on post-deployment smart contract security. A project on Immunefi sets bounty reward levels (maximum payouts for low, medium, high, and critical vulnerabilities) and publishes an in-scope list. Security researchers then submit vulnerability reports at any time. If valid, the project team pays out the bounty and fixes the bug.

Immunefi hosts bounty programs for many of the largest DeFi projects – including MakerDAO, Compound, Optimism, Polygon, Chainlink, and Lido. As of 2024, Immunefi boasts a community of 35,000+ registered security researchers and over \$90 million in bounties paid.

The largest bounty in history was paid via Immunefi: in 2022, a white hat earned \$10,000,000 for identifying a critical Wormhole bridge vulnerability. Other notable bounties include \$6 million to pwning.eth for a bug in Aurora, and several multi-million-dollar rewards in 2022–2023.

Effectiveness

Strengths: Bounty programs harness the global hacker community for ongoing security, finding bugs audits miss. They provide continuous monitoring and incentivize ethical disclosure over exploitation. Immunefi alone has prevented \$25 billion in potential losses by enabling white hats to report bugs before attackers exploit them.

⁴⁸<https://code4rena.com/>

Limitations: Incentive alignment is critical: if bounties are too low, hackers may prefer exploitation. Coverage is not guaranteed – smaller projects may attract few hunters. Quality control is a challenge: open programs invite spam or low-value reports, requiring triage. Finally, vulnerability disclosure carries risks: projects must fix bugs swiftly to avoid leaks or exploits during remediation.

Appendix Table C1: Top 10 Code4rena Bounty Hunters. This table reports the top 10 bounty hunters ranked by total USD earnings from security contests on Code4rena. Columns show the total reward amount, number of findings, and distribution across severity categories: HIGH (solo and shared), MED (solo and shared), and GAS optimizations. The final column provides a link to each hunter’s profile.

Rank	Hunter	USD	Total	High	High Solo	Med	Med Solo	Gas	Link
1	cmichel	\$1,316,375.72	959	183	80	230	116	117	Profile
2	WatchPug	\$801,777.22	1121	166	56	183	65	504	Profile
3	xuwinnie	\$740,188.87	65	33	10	24	6	0	Profile
4	cccZ	\$332,499.43	423	106	7	213	28	4	Profile
5	bin2chen	\$330,978.51	279	97	3	143	10	0	Profile
6	leastwood	\$326,339.31	262	70	26	85	45	15	Profile
7	hyh	\$311,803.71	413	69	6	122	32	63	Profile
8	gperson	\$301,708.98	371	44	12	53	20	60	Profile
9	pauliax	\$294,506.10	938	60	4	86	17	368	Profile
10	Spearbit	\$292,640.04	5	2	0	1	0	1	Profile

Table truncated for brevity—see full list here <https://code4rena.com/leaderboard>.

Appendix D — DeFi Category, Blockchain, and Compendium

Category

The vector *category* is based on the following 11 DeFi categories. Below we define each category and provide one representative protocol as an example:

- **Algo-Stables:** protocols that manage algorithmic stablecoins, maintaining price stability without traditional collateral. These systems often use algorithmic supply adjustments and incentive mechanisms to peg the token to a target (e.g., USD). *Example:* [Ampleforth](#), which dynamically adjusts supply to maintain its price target.
- **CDP (Collateralized Debt Position):** protocols that allow users to lock collateral (such as ETH or other crypto assets) to mint or borrow tokens, often stablecoins. These protocols are core components of decentralized lending markets. *Example:* [MakerDAO](#), which enables users to generate DAI by locking ETH or other assets.
- **DEX (Decentralized Exchange):** Exchanges that facilitate peer-to-peer trading of digital assets without intermediaries or centralized custody. These often rely on automated market makers (AMMs). *Example:* [Uniswap](#), one of the largest AMM-based DEX protocols.
- **Lending:** protocols that enable users to lend or borrow cryptocurrencies, earning interest on supplied assets or paying interest on borrowed funds. *Example:* [Aave](#), a leading decentralized lending and borrowing protocol.
- **Options:** protocols for trading options and other derivatives on-chain, allowing decentralized hedging and speculation. *Example:* [Opyn](#), which provides decentralized options for various crypto assets.
- **Reserve Currency:** protocols that issue decentralized reserve assets intended to act as stable or alternative currencies within the ecosystem. *Example:* [OlympusDAO](#), which pioneered the concept of a decentralized reserve currency backed by treasury assets.
- **Services:** protocols providing auxiliary infrastructure or services to DeFi, such as price oracles, analytics, and transaction relays. *Example:* [Chainlink](#), the most widely used decentralized oracle network.
- **Staking:** protocols that allow users to stake tokens to secure a network, validate transactions, or participate in governance, earning rewards in return. *Example:* [Lido](#), which provides liquid staking solutions for Ethereum and other chains.
- **Yield:** protocols that optimize return strategies on deposited assets, typically through yield farming or liquidity provisioning. *Example:* [Convex Finance](#), which enhances yield on Curve LP tokens.

- **Yield Aggregators:** protocols that automate yield farming by reallocating user funds across multiple DeFi protocols to maximize returns. *Example:* [Yearn Finance](#), one of the first and most prominent yield aggregation protocols.
- **Other:** All other DeFi protocols that do not fall under the categories above, including experimental or hybrid designs.

Blockchain

The vector *blockchain* is based on the following 10 blockchain categories. Below we describe each and provide an example of a major DeFi application built on it:

- **Avalanche:** A high-performance, scalable chain supporting low-latency transactions. Known for sub-second finality and low fees. *Example:* [Trader Joe](#), the leading DEX on Avalanche.
- **Binance Smart Chain (BSC):** A popular chain offering low transaction fees and high throughput, widely used for DeFi and gaming. *Example:* [PancakeSwap](#), a major AMM-based DEX on BSC.
- **Cronos:** A chain designed to bridge DeFi and the Cosmos ecosystem, with strong integrations to Crypto.com. *Example:* [VVS Finance](#), a leading AMM and yield platform on Cronos.
- **Ethereum:** The most widely used blockchain for DeFi and the origin of most major platforms, offering the highest security but relatively high gas costs. *Example:* [Uniswap](#), the largest DEX by volume.
- **Fantom:** A scalable platform focused on speed and low fees, utilizing a DAG-based consensus mechanism. *Example:* [SpookySwap](#), a popular DEX on Fantom.
- **Harmony:** A chain optimized for decentralized apps and cross-chain finance with fast finality. *Example:* [Tranquil Finance](#), a lending platform on Harmony.
- **Polygon:** A Layer-2 scaling solution for Ethereum providing fast, low-cost transactions while maintaining Ethereum compatibility. *Example:* [QuickSwap](#), a DEX on Polygon.
- **Solana:** A high-performance chain with extremely low fees and high throughput, used for both DeFi and NFTs. *Example:* [Raydium](#), a Solana-based AMM and liquidity provider.
- **Terra:** A blockchain historically focused on algorithmic stablecoins and related DeFi applications (notable collapse in 2022). *Example:* [Anchor platform](#), formerly a major lending platform on Terra.
- **Other:** All other blockchains, including emerging Layer-1s (e.g., Aptos, Sui) or app-specific chains in the Cosmos ecosystem.

Compendium

Appendix Table D1: Compendium of Terms in DeFi, Auditing, and Bounties

Term	Brief explanation (non-technical)
Blockchain	A distributed ledger where participants agree on an ordered record of transactions; blocks are cryptographically linked, so past entries are exceptionally costly to alter.
Smart contract	A program on a blockchain that automatically executes agreed-upon rules for a transaction or service.
DeFi protocol	An online platform built from smart contracts that provides a financial service (e.g., trading, lending).
Total Value Locked (TVL)	Dollar value of assets deposited in a protocol; rough gauge of economic scale/user commitment.
Oracle	A mechanism that feeds off-chain data (e.g., prices) to smart contracts; adds useful functionality but creates dependencies/attack surfaces.
Bridge / Cross-chain	Infrastructure that moves assets or messages across blockchains; essential for integration but may introduce fragility.
DEX (Decentralized Exchange)	Exchange that lets users trade directly via smart contracts; many use automated market makers (AMMs) instead of order books.
Stablecoin	Cryptoasset are designed to track a reference asset (usually the U.S. dollar) to reduce price volatility in on-chain activity.
DAO (Decentralized Autonomous Organization)	A new type of decentralized governance in which token holders vote on parameters, upgrades, and budgets.
Layer-1 (L1) / Layer-2 (L2)	Base blockchain vs. scaling networks built on top (e.g., “rollups“) that inherit L1 security.
EVM (Ethereum Virtual Machine)	Common runtime standard that lets the same contract code run on many EVM-compatible chains.
Gas fee	Fee paid to validators for processing a transaction/executing a contract.
Centralized audit (code audit)	Pre-launch, fixed-fee review of smart-contract code by a professional security firm; yields a formal report.
Top-tier / Bottom-tier auditor	Market shorthand for high-reputation vs. smaller audit firms; used as a proxy for audit quality.
Static analysis	Automated scanning of code (without running it) to flag common bugs and unsafe patterns.
Dynamic analysis	Running the code in a controlled environment to observe behavior and detect runtime issues.
Formal verification	Mathematical checks that code satisfies specified properties (when feasible for critical modules).
Penetration testing	Simulated attacks on systems around the contracts (e.g., front ends, keys, infrastructure) to find weaknesses.
Bug bounty program	Post-launch program that pays independent researchers (“white hats“) for responsibly disclosing vulnerabilities.
Bounty platform (Immunefi / Code4rena)	Marketplaces that host continuous bounties or short audit contests and coordinate disclosures/payouts.
Severity tier (critical/high/...)	Classification of bug impact; rewards and urgency of fixes scale with severity.
White hat	Ethical hacker who reports a vulnerability rather than exploiting it; typically compensated through a bounty.
Exploit	A successful attack that drains funds or manipulates protocol behavior through a code or design flaw.
Common attack vectors	Examples include reentrancy, flash-loan/state manipulation, oracle price manipulation, and bridge key/verification failures.
Composability	“Lego-like“ ability for protocols to plug into each other; boosts innovation but transmits risk across systems.
Post-mortem	Public write-up after an incident explaining the cause, impact, remediation, and prevention steps.

Notes: Definitions are adapted for a non-technical audience.

Appendix E — Security Breaches

Decentralized Finance (DeFi) protocols remain highly vulnerable to cybersecurity attacks due to their reliance on immutable smart contracts, permissionless composability, and cross-chain interoperability. Unlike centralized platforms that can freeze assets or reverse fraudulent transactions, most DeFi systems execute autonomously, making exploits difficult to mitigate after the incident. Hacks typically exploit vulnerabilities in smart contract logic, cross-chain bridges, price oracles, or privileged governance keys.⁴⁹ The magnitude of losses has grown with DeFi adoption: cumulative on-chain theft exceeds \$8 billion since 2016, with peak losses during the 2021–2022 boom period. Notably, attacks are concentrated, with a handful of systemic events accounting for a disproportionate share of the total value stolen. These high-profile incidents often involve infrastructure components such as cross-chain bridges, oracles, and lending protocols, which aggregate large amounts of liquidity and thus present attractive targets.

Trends in Attack Vectors. Early exploits (2016–2019) were dominated by coding logic errors, such as re-entrancy vulnerabilities (e.g., The DAO hack in 2016). From 2020 onward, the rise of composable DeFi applications and Layer-2 scaling solutions introduced complex interdependencies, creating new attack surfaces. Flash loan exploits surged in 2020–2021, enabling attackers to manipulate protocol states without upfront capital. More recently, bridge exploits and governance attacks have emerged as systemic risks, exemplified by the multi-hundred-million-dollar breaches discussed below.

Systemic Hack Events. We highlight six major infrastructure-level hacks that collectively account for roughly \$2.2 billion in losses (nearly 25% of all DeFi-related hacks from 2021–2024):

- **Poly Network (August 2021):** Exploiting flaws in cross-chain message verification, attackers siphoned over \$610 million in assets across multiple chains. Remarkably, the attacker later returned most funds, negotiating a \$500,000 “bug bounty” and immunity promise.⁵⁰
- **BadgerDAO (December 2021):** A front-end compromise targeting BadgerDAO’s UI tricked users into approving malicious smart contracts, draining approximately \$120 million in wrapped BTC and ETH collateral.⁵¹

⁴⁹Common vector of attacks are: *Reentrancy attacks* exploit a contract’s external call that allows malicious contracts to repeatedly withdraw funds before the original function completes. *Oracle manipulation* occurs when attackers influence off-chain data feeds (e.g., asset prices) to distort protocol behavior. *Integer overflows/underflows* arise when arithmetic operations exceed the storage limits of a variable, potentially bypassing logic checks. *Faulty access controls* refer to improperly defined permissions that allow unauthorized users to execute sensitive contract functions. *Bridge vulnerabilities* stem from the complexity of transferring assets across blockchains; they often involve weak verification mechanisms or centralized validator designs, as seen in major exploits like the PolyNetwork and Ronin Bridge hacks.

⁵⁰[Poly Network Hack, Reuters](#).

⁵¹[BadgerDAO Hack, CoinDesk](#).

- **Ronin Network (March 2022):** The largest DeFi hack to date, involving a compromise of validator keys in Axie Infinity’s Ronin bridge, resulted in \$615 million stolen in ETH and USDC. The attack exploited centralized control in a nominally decentralized bridge.⁵²
- **Binance Bridge (October 2022):** Attackers forged proof-of-deposit transactions on Binance Smart Chain’s bridge, minting 2 million BNB (worth \$566 million). While most funds were frozen via network coordination, \$110 million remained unrecovered.⁵³
- **Euler Finance (March 2023):** A sophisticated flash-loan attack manipulated Euler’s borrowing and collateral mechanism, draining \$197 million. After on-chain negotiations, the attacker returned the majority of funds, illustrating evolving norms around “white hat” deals.⁵⁴
- **Orbit Bridge (December 2023):** Exploiters targeted multi-sig management flaws and validator key vulnerabilities in Orbit Chain’s cross-chain bridge, stealing approximately \$82 million across multiple assets.⁵⁵

Economic Impact. These incidents highlight systemic vulnerabilities in DeFi: bridges aggregate liquidity and attract exploits; validator key compromises and governance centralization introduce single points of failure; and post-hack recovery often depends on ad hoc negotiations.

Appendix Table Er: DeFi Hack Events and Amount Lost by Year. Data sourced from [DeFiLlama](#).

Year	Amount Lost (USD Millions)	Number of Attacks
2016	60.0	1
2017	157.7	2
2018	235.0	1
2019	40.0	5
2020	183.8	16
2021	2,290.2	66
2022	3,280.8	59
2023	1,396.0	41
2024	420.0	17
2025*	95.0	6
Total	8,157.5	214

Notes: Figures include cross-chain bridge exploits, governance attacks, oracle manipulation, and flash-loan exploits. Data for 2025 reflect preliminary estimates through July 2025.

⁵²Ronin Hack, Reuters.

⁵³Binance Bridge Attack, Chainalysis.

⁵⁴Euler Finance Hack, Bloomberg.

⁵⁵Orbit Bridge Hack, Bloomberg Law.

Appendix F — Generative AI for Smart Contract Auditing

Our experimental evidence suggests that generative AI tools, such as OpenAI’s GPT models, can substantially augment traditional smart contract auditing workflows. While GPT-4 demonstrates notable improvements over GPT-3.5 in detecting vulnerabilities and proposing viable code fixes, these models are not yet reliable substitutes for professional audits. False positives, occasional logic gaps, and limited ability to assess composability risks underscore the need for human oversight. Nonetheless, AI-driven auditing offers two important benefits: (i) reducing the marginal cost of preliminary reviews for resource-constrained protocols, and (ii) accelerating internal quality assurance and vulnerability triage before formal audits. As LLM capabilities evolve, their integration into hybrid assurance models—combining automated code analysis with expert review—may fundamentally reshape the economics of DeFi security assurance, echoing trends in technology-assisted auditing observed in financial reporting contexts (Dowling and Leech, 2014; Rozario and Vasarhelyi, 2018).

A. Objective

This appendix describes our experimental design to evaluate whether large language models (LLMs) can perform tasks traditionally carried out by professional smart contract auditors. Specifically, we examine whether LLM-based tools can identify and explain security vulnerabilities in Solidity code and propose fixes.

B. Experimental Design

Data Source. We curated a benchmark dataset of Solidity smart contracts with known vulnerabilities using publicly available GitHub repositories and security testing suites. These contracts encompass common exploit categories such as:

- **Reentrancy attacks,**
- **Integer overflow and underflow,**
- **Unchecked external calls,**
- **Storage collision and delegatecall misuses.**

Model Variants. Two versions of OpenAI’s language models were tested:

1. *GPT-3.5* (baseline model from 2023),
2. *GPT-4.1* (latest release during the study).

Both were accessed via API with identical prompting strategies to ensure consistency.

Prompting and Output. Each model received the following:

- Complete Solidity code snippet,
- System instruction to act as a security auditor,
- Request to identify vulnerabilities, explain exploitability, and provide code patches.

C. Evaluation Metrics

1. **Detection Rate (%)**: Share of known vulnerabilities correctly flagged.
2. **False Positive Rate (%)**: Share of flagged issues not present in the ground truth.
3. **Diagnostic Quality**: Accuracy of explanations (graded 1–5).
4. **Remediation Quality**: Whether proposed fixes compile and address the issue.

D. Comparative Performance

Table II summarizes the results for *GPT-3.5* versus *GPT-4.1*.

Table II. Performance Comparison: GPT-3.5 vs GPT-4.1 in Smart Contract Auditing.

Metric	GPT-3.5	GPT-4.1
Detection Rate (%)	62.5	84.2
False Positive Rate (%)	18.3	11.9
Diagnostic Quality (1–5)	3.2	4.4
Remediation Success (%)	54.8	77.6

Key Insight. *GPT-4.1* demonstrates material improvements in vulnerability detection and remediation quality compared to *GPT-3.5*, reducing false positives and generating more reliable explanations.

E. Implications

The findings indicate that LLM-based auditing tools can significantly reduce the marginal cost of preliminary audits but are not yet full substitutes for professional security reviews. These models perform best as complementary tools in hybrid workflows, supporting:

- Rapid internal QA prior to formal audits,
- Continuous monitoring and bug hunting,
- Reducing reliance on expensive auditors for low-risk codebases.

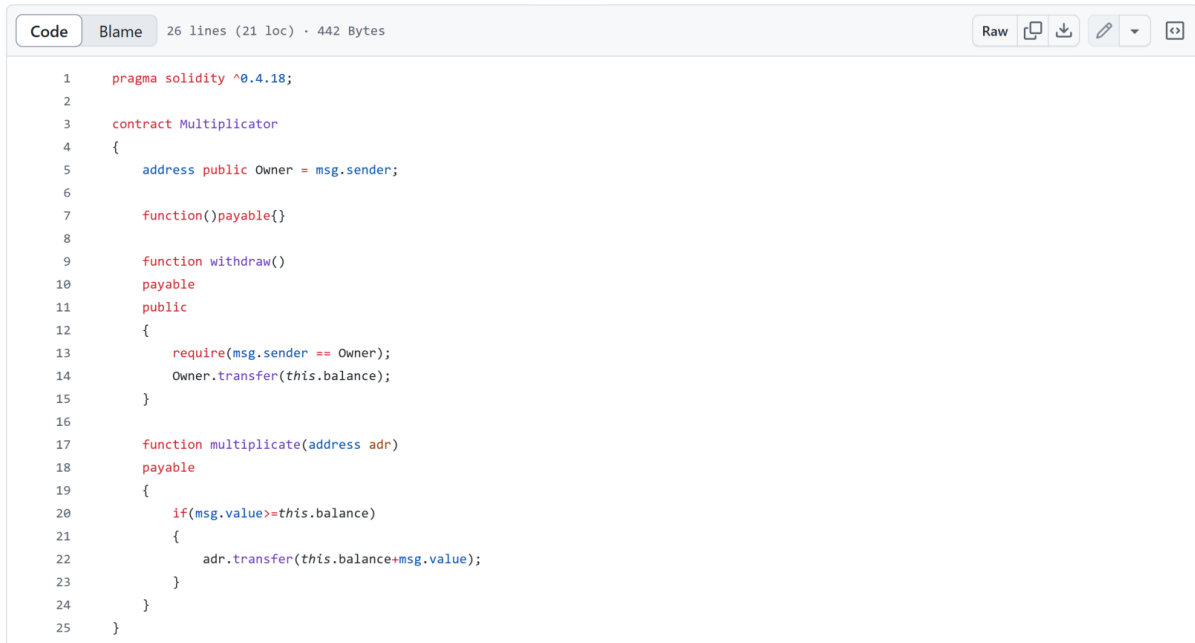
Note: All prompts, evaluation scripts, and anonymized outputs are available in the online supplementary materials.

E. Procedure Visualization

The next set of figures illustrates our experiment./

GenAI for Smart Contract Auditing

[GitHub Repository with Common Smart Contract Vulnerabilities](#)



```
Code Blame 26 lines (21 loc) · 442 Bytes Raw Copy Download Edit View Source
```

```
1  pragma solidity ^0.4.18;
2
3  contract Multiplier
4  {
5      address public Owner = msg.sender;
6
7      function()payable{}
8
9      function withdraw()
10     payable
11     public
12     {
13         require(msg.sender == Owner);
14         Owner.transfer(this.balance);
15     }
16
17     function multiply(address adr)
18     payable
19     {
20         if(msg.value>=this.balance)
21         {
22             adr.transfer(this.balance+msg.value);
23         }
24     }
25 }
```

Prompt

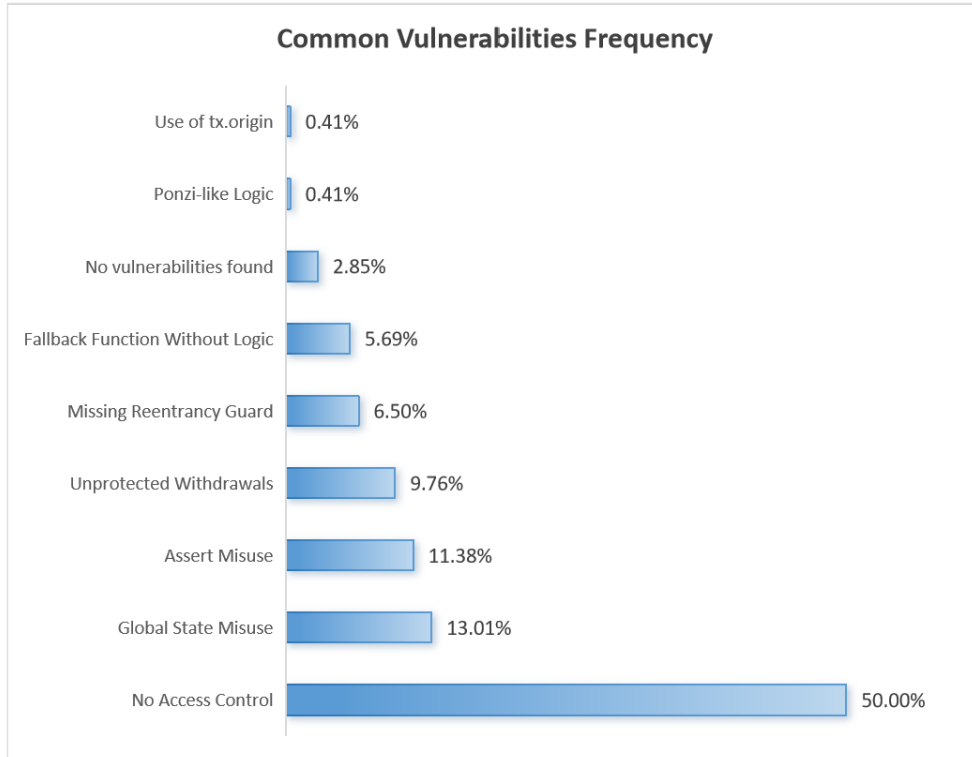
1. Analyze the smart contracts line-by-line,
2. Identify vulnerabilities by category (e.g., reentrancy, integer overflows, access control, etc.),
3. Provide:
 - Descriptions of each issue,
 - Exploit scenarios where applicable,
 - Recommendations or patches to mitigate them.

Feel free to let me know if you want the output as:

- A summary report,
- Annotated source code,
- A markdown vulnerability table, or
- A formal audit-style document in LaTeX or PDF.

Vulnerabilities List

Subfolder	File	Issue	Description	Severity
Abuse of global semantics	Multiplicator.sol	Unprotected Withdrawals	Unprotected Withdrawals found in Multiplicator.sol	Critical
Abuse of global semantics	Multiplicator.sol	Fallback Function Without Logic	Fallback Function Without Logic found in Multiplicator.sol	Low
Abuse of global semantics	Multiplicator.sol	Ponzi-like Logic	Ponzi-like Logic found in Multiplicator.sol	Critical
Abuse of global semantics	Multiplicator.sol	No Access Control	No Access Control found in Multiplicator.sol	Medium
Abuse of global semantics	Multiplicator.sol	Global State Misuse	Global State Misuse found in Multiplicator.sol	Medium
Arbitrary Jump with Function Type Variable	FunctionTypes.sol	No Access Control	No Access Control found in FunctionTypes.sol	Medium
Asserting EOA from Code Size	mint.sol	No Access Control	No Access Control found in mint.sol	Medium
Assert-Requirement Violation	AssertConstructor.sol	No Access Control	No Access Control found in AssertConstructor.sol	Medium
Assert-Requirement Violation	AssertConstructor.sol	Assert Misuse	Assert Misuse found in AssertConstructor.sol	Medium
Assert-Requirement Violation	AssertMinimal.sol	No Access Control	No Access Control found in AssertMinimal.sol	Medium
Assert-Requirement Violation	AssertMinimal.sol	Assert Misuse	Assert Misuse found in AssertMinimal.sol	Medium
Assert-Requirement Violation	AssertMultiTx2.sol	No Access Control	No Access Control found in AssertMultiTx2.sol	Medium
Assert-Requirement Violation	AssertMultiTx2.sol	Assert Misuse	Assert Misuse found in AssertMultiTx2.sol	Medium
Assert-Requirement Violation	Bar.sol	No Access Control	No Access Control found in Bar.sol	Medium
Assert-Requirement Violation	ConstructorCreate.sol	No Access Control	No Access Control found in ConstructorCreate.sol	Medium
Assert-Requirement Violation	ConstructorCreate.sol	Assert Misuse	Assert Misuse found in ConstructorCreate.sol	Medium
Assert-Requirement Violation	ConstructorCreateArgument.sol	No Access Control	No Access Control found in ConstructorCreateArgument.sol	Medium
Assert-Requirement Violation	ConstructorCreateArgument.sol	Assert Misuse	Assert Misuse found in ConstructorCreateArgument.sol	Medium
Assert-Requirement Violation	ConstructorCreateModifiable.sol	No Access Control	No Access Control found in ConstructorCreateModifiable.sol	Medium
Assert-Requirement Violation	ConstructorCreateModifiable.sol	Assert Misuse	Assert Misuse found in ConstructorCreateModifiable.sol	Medium
Assert-Requirement Violation	GasModel.sol	No Access Control	No Access Control found in GasModel.sol	Medium
Assert-Requirement Violation	GasModel.sol	Assert Misuse	Assert Misuse found in GasModel.sol	Medium



Performance: GPT 3.5 vs GPT 4.1

Challenge	GPT-3.5 Insight	GPT-4 Improvement	Discrepancy	GPT-4 Advantage	DIFFICULT
AssumeOwnershipChallenge	Correctly identifies constructor type vulnerability	Same correct finding, but explains Solidity version context, fix with constructor() keyword	None – both correct.	Stronger recommendation and technical precision (e.g., fix via constructor syntax post-0.4.22).	1
GuessTheNumberChallenge	Correctly notes answer is hardcoded and visible on-chain	Same finding but focuses more on implications of hardcoded constants in public state	None – both correct.	Better summary of Solidity design transparency and attack simplicity.	2
GuessTheSecretNumberChallenge	Eventually gets to brute-force idea, but with multiple iterations and syntax errors	Immediately suggests preimage attack using Web3.solidityKeccak(['uint8'], [i]) and correct number	GPT-3.5 had multiple corrections due to Solidity/Web3 usage	Cleaner, accurate code and direct attack surface explanation.	3
GuessTheNewNumberChallenge	Notes predictability of blockhash and now, but no working exploit initially	Suggests on-chain contract that calculates answer in the same block and provides working exploit	GPT-3.5 missed correct Solidity syntax until final version	Clear attack contract with correct syntax; no debugging iterations.	4
PredictTheBlockHashChallenge	Correctly notes blockhash returns 0 after 256 blocks	Same insight, but gives better description of the block expiration window	Minor wording difference – GPT-3.5 focused more on miners	Sharper clarity on exploit timing logic (e.g., delay settlement to force blockhash = 0x0).	5
TokenBankChallenge	Initially wrong — reentrancy misunderstood due to internal ledger use	Correctly shows attacker needs to first deposit tokens to receive a balance and trigger fallback exploit	GPT-3.5 required multiple failed iterations before correct exploit	Correct from first attempt, with better modeling of reentrancy condition and attacker precondition.	6
TokenSaleChallenge	Correctly flags integer overflow but claims it's impractical to exploit	Provides exact overflow input needed to buy many tokens for little ETH	GPT-3.5 falsely states value is too large to execute	Concrete value + Solidity version context + correct overflow math.	7

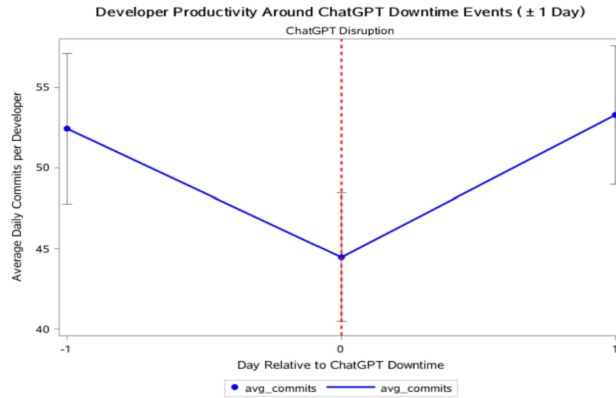
Appendix G — Additional IV Discussion

This appendix provides additional discussion, validation tests, and robustness checks for our instrumental variable (IV) strategy exploiting the interaction between protocol GitHub activity and the public release of ChatGPT. We first elaborate on the economic mechanism underpinning instrument relevance, showing why the AI shock disproportionately benefited open-source DeFi teams with internal development capacity, thereby influencing their demand for top-tier external audits. We then present multiple validity checks, including (i) pre-trend tests to confirm the absence of systematic differences in audit adoption before the shock, (ii) covariate balance tests within a ± 6 month window around the event, and (iii) an exclusion restriction test to rule out direct effects of the shock on security outcomes other than through auditor choice. Finally, we complement this analysis with evidence on the role of AI tools in DeFi development workflows by examining developer productivity during periods of significant ChatGPT service disruption.

Validity of the shock-IV design: By construction, the $GITHUB \times POSTChatGPT$ instrument should capture only the exogenous component of audit choice driven by the AI shock. One might worry about confounding factors—for instance, that projects launched after 2022 could systematically differ in risk or quality. Our ± 6 -month bandwidth around the shock addresses this by comparing contemporaneous launches. Furthermore, we verify in the data that pre-shock trends in audit adoption were similar for would-be treated ($GITHUB=1$) and control ($GITHUB=0$) groups, satisfying the parallel trends intuition for the first stage. Following best practices for shock-IV designs, we also ensure covariate balance: there are no significant differences in observable characteristics (like project size, sector, or complexity) between GitHub and non-GitHub protocols in the narrow window around the shock. These checks imply that aside from the AI tool introduction, GitHub vs. non-GitHub projects were on similar footing, making the interaction-driven variation in auditor choice plausibly exogenous.

ChatGPT Disruption and Impact on DeFi Developer Productivity: We examine whether reliance on AI-assisted coding tools translates into measurable differences in developer productivity for DeFi projects. Specifically, we use GitHub activity metrics—such as the number of commits—for all DeFi protocols in our dataset, as a proxy for coding output, and analyze their variation around exogenous disruptions to OpenAI’s ChatGPT service. Data on outages are obtained directly from OpenAI’s system status logs, and we identify *disruption days* as those in which the service experienced more than three consecutive hours of downtime. We then compare daily coding activity during disruption periods to baseline levels. The figure below shows a pronounced decline in GitHub commits on days with significant ChatGPT outages, suggesting that AI tools

have become integral to DeFi development workflows.⁵⁶



Pre-trends and Covariate Balance: The tables below provide empirical evidence supporting the validity of our IV strategy. Appendix Table G1 reports pre-trend estimates for the probability of hiring a top-tier auditor in the 12 months preceding the ChatGPT release (May 2022–April 2023). The coefficient on the interaction term $GITHUB \times POSTChatGPT$ is statistically significant at the 10% level, while the coefficients on $GITHUB$ and $POSTChatGPT$ alone are also statistically insignificant. This pattern suggests that protocols with GitHub repositories—our proxy for internal development capability—did not exhibit systematically different trends in auditor selection prior to the technological shock. In other words, the first-stage relationship observed post-shock is unlikely to be driven by pre-existing differences between treatment and control groups, satisfying the *parallel trends* intuition emphasized in shock-based identification designs (Desai and Dharmapala, 2009; Atanasov and Black, 2021).

Appendix Table G2 assesses covariate balance within a ± 6 month window around November 2022, the date of the ChatGPT release. Across multiple protocol-level characteristics, including size (TVL), governance features (DAO), and sector indicators, we find no consistent or economically meaningful differences between protocols with GitHub repositories (treatment group) and those without (control group). Most coefficients on $GITHUB$ are statistically insignificant, and even where differences exist (e.g., $\log_Followers$ and $LISTED$), the magnitudes are modest and not systematically correlated with the outcome variable in a way that would bias the IV estimates. These findings align with best practices for shock-based IV designs, which recommend validating *pretreatment balance* to ensure that instrumented variation stems from the exogenous event rather than confounding differences (Angrist and Krueger, 1995; Atanasov and Black, 2021).

Taken together, the absence of significant pre-trends and the strong covariate balance lend credibility to our identification strategy. They suggest that the variation induced by the $GITHUB \times POSTChatGPT$ interaction

⁵⁶This pattern is consistent with the view that large language models (LLMs) enhancing developer productivity, and their absence materially slows down internal code assurance processes.

primarily reflects exposure to the technological shock rather than underlying structural differences. This strengthens the case for instrument relevance and mitigates omitted variable bias, consistent with the conditions for valid IV estimation (discussed in [Angrist and Pischke \(2009\)](#); [Wooldridge \(2010\)](#)).

Appendix Table G1: Pre-Trend for Top-Tier Auditors. This table examines whether protocols with GitHub repositories exhibit differential pre-trends in hiring top-tier auditors in the 12 months leading up to the ChatGPT release (May 2022–April 2023). The dependent variable is an indicator for selecting a top-tier auditor. Robust standard errors are clustered at the industry \times blockchain level.

VARIABLES	<i>TOP</i> (before 2022m5–2023m4)
<i>GITHUB</i>	0.451 (0.400)
<i>POSTChatGPT</i>	-0.134 (0.217)
<i>GITHUB</i> \times <i>POSTChatGPT</i>	-0.301 (0.280)
<i>ORACLE</i>	0.006 (0.061)
<i>DAO</i>	-0.144 (0.097)
<i>log_TVL</i>	-0.007 (0.007)
<i>log_Chains</i>	0.123 (0.109)
<i>log_Staking</i>	-0.001 (0.007)
<i>log_Raised</i>	0.000 (0.007)
<i>log_Commits</i>	-0.018 (0.037)
<i>log_Followers</i>	0.003 (0.006)
<i>LISTED</i>	0.072 (0.098)
<i>log_EthereumTVL</i>	0.021 (0.334)
<i>IND_LENDING</i>	-0.035 (0.088)
<i>IND_DEXES</i>	-0.006 (0.052)
<i>BC_ETHEREUM</i>	0.068 (0.068)
<i>BC_CROSSCHAIN</i>	-0.081 (0.085)
Observations	446
R-squared	0.070
Industry FE	YES
Blockchain FE	YES
Year-month FE	YES

Robust standard errors in parentheses. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Appendix Table G2: Covariate Balance Around the ChatGPT Shock. This table reports differences in baseline characteristics between protocols with GitHub repositories (treatment) and those without (control) in a ± 6 month window around November 2022. Each column reports coefficients from separate regressions of the listed covariate on treatment status and controls. Robust standard errors are clustered at the industry \times blockchain level.

VARIABLES	(1) <i>log_TVL</i>	(2) <i>ORACLE</i>	(3) <i>IND_LENDING</i>	(4) <i>BC_CROSSCHAIN</i>
<i>GITHUB</i>	1.662 (3.082)	0.015 (0.338)	-0.222 (0.224)	0.430 (0.263)
<i>GITHUB</i> \times <i>POSTChatGPT</i>	-0.167 (1.084)	-0.146 (0.169)	0.146 (0.091)	-0.001 (0.088)
<i>ORACLE</i>	-0.292 (0.409)		0.209*** (0.073)	0.042 (0.028)
<i>DAO</i>	-1.465* (0.866)	0.086 (0.073)	-0.102** (0.047)	0.058 (0.036)
<i>log_TVL</i>		-0.004 (0.005)	0.006 (0.004)	-0.006 (0.004)
<i>log_Chains</i>	2.034** (0.912)	-0.097 (0.100)	-0.118 (0.096)	0.813*** (0.058)
<i>log_Staking</i>	0.042 (0.052)	-0.001 (0.006)	-0.002 (0.003)	0.005* (0.003)
<i>log_Raised</i>	-0.052 (0.038)	0.001 (0.007)	-0.003 (0.004)	0.005 (0.004)
<i>log_Commits</i>	-0.034 (0.307)	0.015 (0.028)	0.023 (0.025)	-0.051* (0.026)
<i>log_Followers</i>	0.174*** (0.047)	0.004 (0.006)	0.001 (0.003)	-0.003 (0.003)
<i>LISTED</i>	1.157* (0.672)	0.051 (0.081)	-0.062 (0.065)	-0.045 (0.067)
<i>log_EthereumTVL</i>	-4.679** (1.932)	-0.006 (0.264)	-0.065 (0.167)	0.003 (0.131)
<i>IND_LENDING</i>	0.786** (0.360)	0.365*** (0.050)		0.044 (0.117)
<i>BC_CROSSCHAIN</i>	-1.527* (0.846)	0.123 (0.085)	0.074 (0.197)	
Observations	446	446	446	446
R-squared	0.172	0.246	0.188	0.751
Industry FE	YES	YES	YES	YES
Blockchain FE	YES	YES	YES	YES
Year-month FE	YES	YES	YES	YES

Robust standard errors in parentheses. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Appendix Table G3: Instrumental Variable (IV) Estimates: AI-Assisted Code Verification (6 Months Around Shock). This table reports two-stage IV-Probit and two-stage least squares (2SLS) estimates evaluating the effect of engaging top-tier centralized auditors on DeFi protocol security. We use an interaction between GitHub activity and the ChatGPT release ($GITHUB \times POSTChatGPT$) as an instrument.

$$\text{First Stage: } TOP_{it} = \beta (GITHUB_i \times POSTChatGPT_t) + \alpha' \mathbf{X}_{it} + \mu_i + \delta_t + \varepsilon_{it},$$

$$\text{Second Stage: } HackOutcome_{it+6} = \delta \widehat{TOP}_{it} + \theta' \mathbf{X}_{it} + \mu_i + \delta_t + u_{it},$$

The outcome variable $HackOutcome_i$ corresponds to either (i) a binary indicator equal to one if protocol i was hacked after deployment ($HACKDUM$), or (ii) the dollar amount of loss conditional on a breach ($Hackloss$). The key explanatory variable TOP_i is a binary indicator equal to one if protocol i engaged a top-tier centralized auditor prior to launch. To address endogeneity in the choice of audit quality, we implement a two-stage instrumental variable strategy. The first stage instruments TOP_i using an interaction between GitHub developer activity and the public release of ChatGPT ($GITHUB \times POSTChatGPT$), which proxies for AI-assisted code verification and audit awareness. The second stage regresses each security outcome on the predicted probability of selecting a top-tier auditor. All regressions control for protocol-level characteristics, as well as industry, blockchain, and year-month fixed effects. Standard errors are clustered at the industry-by-blockchain level.

VARIABLES	Instrumental Variable		
	(1) <i>TOP</i> First Stage	(2) <i>HACKDUM</i> Second Stage	(3) <i>Hackloss</i> Second Stage
<i>IV = (GITHUB × POSTChatGPT)</i>	-0.432 ^{***} (0.140)		
<i>TOP</i>		-2.089 ^{***} (0.040)	-9.309 ^{**} (4.559)
<i>GITHUB</i>	0.097 (0.406)	-0.354 (0.760)	3.267 (5.745)
Observations	370	370	370
(Pseudo) R-squared	0.065	0.617	0.093
Controls	YES	YES	YES
Industry FE	YES	YES	YES
Blockchain FE	YES	YES	YES
Year-mon FE	YES	YES	YES
Weak-IV Diagnostics (AR Test)	7.55, $p = 0.006$		

Notes: Robust standard errors clustered at the Industry × Blockchain level. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Appendix Table G4: Instrumental Variable (IV) Estimates: AI-Assisted Code Verification (6 Months Around Shock). This table reports two-stage least squares estimates evaluating the effect of engaging bounty programs on DeFi protocol security. The instrument is the interaction between GitHub activity and the ChatGPT release.

$$\text{First Stage: } BOUNTY_{it} = \beta \cdot (GITHUB_i \times POSTChatGPT_t) + \alpha' \mathbf{X}_{it} + \mu_i + \delta_t + \varepsilon_{it},$$

$$\text{Second Stage: } HackOutcome_{it+6} = \delta \cdot \widehat{BOUNTY}_{it} + \theta' \mathbf{X}_{it} + \mu_i + \delta_t + u_{it}$$

The outcome variable $HackOutcome_{it+6}$ corresponds to either (i) a binary indicator equal to one if protocol i was hacked within six months of bounty adoption ($HACKDUM$), or (ii) the dollar amount of exploit losses conditional on a breach ($Hackloss$). The key explanatory variable $BOUNTY_{it}$ is a binary indicator equal to one if protocol i initiated a decentralized bounty audit program in month t . To mitigate endogeneity in bounty adoption, we use a two-stage instrumental variable approach. The first stage instruments $BOUNTY_{it}$ using the interaction between GitHub development intensity and the introduction of ChatGPT ($GITHUB \times POSTChatGPT$), which proxies for AI-assisted awareness and bounty implementation. The second stage estimates the effect of predicted bounty adoption on subsequent hacking outcomes. All regressions include protocol and time fixed effects, along with time-varying protocol characteristics. Standard errors are clustered at the industry-by-blockchain level.

VARIABLES	Instrumental Variable		
	(1) <i>BOUNTY</i> First Stage	(2) <i>HACKDUM</i> Second Stage	(3) <i>Hackloss</i> Second Stage
<i>IV = (GITHUB × POSTChatGPT)</i>	0.008 ^{***} (0.002)		
<i>BOUNTY</i>		-0.505 ^{***} (0.148)	-2.369 ^{***} (0.277)
<i>POSTChatGPT</i>	0.002 ^{**} (0.001)	-0.002 ^{**} (0.001)	0.008 ^{***} (0.001)
Observations	21,680	21,680	21,674
R-squared	0.961	0.571	0.506
Protocol FE	YES	YES	YES
Year FE	YES	YES	YES
Weak-IV Diagnostics (F-statistic)	11.77, $p = 0.006$		

Notes: Robust standard errors clustered at the Industry × Blockchain level. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Appendix H — Bartik Instruments (IV)

In this section, we develop a Bartik-type (shift-share) instrument (Bartik IV) as an additional robustness test to our analysis in Section 5.2. The baseline IV leverages the release of OpenAI’s ChatGPT (November 2022) to proxy an exogenous improvement in AI-assisted coding capacity, interacting a protocol’s ex ante GitHub presence with a post-event indicator. While informative, that approach may inherit endogeneity from the *GITHUB* indicator, which is itself a pre-launch strategic choice potentially correlated with unobserved fundamentals (e.g., financial constraints, scale, or governance quality) that also relate to audit choice. To mitigate this, we construct an instrument in the spirit of Bartik (1991), which has been widely applied in economics and finance (Diamond, 2016; Goldsmith-Pinkham et al., 2020). The key idea of the Bartik instrument is to exploit plausibly exogenous aggregate trends that differentially affect units according to their cross-sectional exposure.

In our setting, DeFi protocols typically deploy across multiple blockchains, which differ systematically in their technical complexity, developer ecosystem, and governance structure. For example, *Ethereum* is the most established and technologically mature ecosystem, characterized by high developer activity and extensive infrastructure support. Other EVM-compatible chains (e.g., Binance Smart Chain, Polygon, Avalanche) replicate Ethereum’s virtual machine architecture but typically compete through lower fees and faster transaction throughput, attracting protocols seeking scalability and cost efficiency. Non-EVM chains (such as Solana, Tron, or Tezos) operate with alternative architectures and consensus mechanisms, creating distinct development environments and governance frameworks. These structural differences imply heterogeneous access to peer review, code transparency, and decentralized governance across chain categories. Building on this variation, we construct an instrument that shifts audit incentives using *pre-ChatGPT* cross-chain DAO intensity measured at the ecosystem level, measured by the prevalence of protocols with DAO governance structures in each chain category. The intuition is that protocols with higher economic exposure to DAO-intensive chains benefit from stronger peer monitoring and community-based governance resources, thereby reducing the marginal benefit of engaging costly top-tier external auditors. We classify blockchains into four broad categories: (i) Ethereum, (ii) EVM-compatible chains (excluding Ethereum), (iii) non-EVM chains, and (iv) others. Appendix Table H1 provides the complete list of the top 150 blockchains sorted by DeFi market share.

Construction of Bartik IV. Let i index protocols and let $C = \{\text{Ethereum, EVM, NonEVM, Others}\}$ be the four blockchain categories. The instrument is built in three steps:

1. **Launch-time chain weights.** For each protocol i (launching post-ChatGPT), compute its Total Value Locked (TVL) shares across chain categories, $w_{ic} \in [0, 1]$ with $\sum_{c \in C} w_{ic} = 1$. These weights capture where

protocol i economically resides at launch.

2. **Chain-level DAO intensity.** For each chain category c , measure the fraction of *pre-ChatGPT* launching protocols that operated on c and adopted a DAO structure:

$$s_c^{\text{pre}} = \frac{\sum_{j \in \mathcal{P}^{\text{pre}}} \mathbf{1}\{j \text{ operates on } c\} \cdot \text{DAO}_j}{\sum_{j \in \mathcal{P}^{\text{pre}}} \mathbf{1}\{j \text{ operates on } c\}},$$

where \mathcal{P}^{pre} is the set of protocols launched strictly before November 30, 2022 and DAO_j is an indicator for whether protocol j used a DAO. Because our estimation sample consists of *post-ChatGPT* launches only.

3. **Protocol-level exposure.** Aggregate the pre-period chain intensities using the protocol’s launch-time weights:

$$\text{BartikIV}_i = \sum_{c \in C} w_{ic} s_c^{\text{pre}}, \quad C = \{\text{Ethereum, EVM, NonEVM, Others}\}.$$

Intuitively, BartikIV_i is higher for protocols that place more economic weight on chain categories whose ecosystems were *ex ante* more DAO-intensive. A more DAO-oriented environment plausibly expands peer review and community governance resources, reducing the need for top-tier centralized audits and thus predicting audit choice in the first stage. By construction, this measure is plausibly exogenous to individual protocols’ own unobserved characteristics, while still generating cross-sectional variation relevant for predicting audit choice.

Relevance and Exclusion. *Relevance* follows from the idea that DAO-intensive ecosystems provide stronger peer review bandwidth and governance support, decreasing the marginal value of hiring a top centralized auditor; consequently, BartikIV_i should negatively predict the propensity to choose a top auditor. *Exclusion* requires that BartikIV_i affect future hacking probability only through audit choice. This condition is credible for three reasons. First, s_c^{pre} is computed entirely from *pre-ChatGPT* deployments and therefore temporally precedes—and cannot be mechanically influenced by—*post-ChatGPT* audit decisions or security outcomes. Second, BartikIV_i varies only through launch-time portfolio weights w_{ic} interacting with the pre-period ecosystem characteristics s_c^{pre} ; in the structural equation we control for rich protocol covariates and include launch month fixed effects and chain-category controls, mitigating the possibility that BartikIV_i proxies for contemporaneous chain shocks. Third, we show robustness to alternative pre-period windows, value- versus count-based definitions of s_c^{pre} , limiting the possibility that a single category drives identification.

Empirical specifications. We next employ the Bartik-type instrument to estimate the effect of top-tier auditor choice on subsequent hacking outcomes. Here we restrict the estimation sample to protocols launched

within six months after the release of OpenAI's ChatGPT (November 2022). The instrument exploits cross-chain exposure to pre-ChatGPT DAO intensity and is therefore time-invariant within the post-ChatGPT sample.

Formally, the specifications are shown below:

$$\text{First Stage: } TOP_i = \beta BARTIKIV_i + \alpha' \mathbf{X}_i + \mu_{c(i)} + \delta_{t(i)} + \varepsilon_i,$$

$$\text{Second Stage: } HackOutcome_i = \delta \widehat{TOP}_i + \theta' \mathbf{X}_i + \mu_{c(i)} + \delta_{t(i)} + u_i,$$

where the outcome variable $HackOutcome_i$ corresponds to either (i) a binary indicator equal to one if protocol i was hacked after deployment ($HACKDUM$), or (ii) the dollar amount of loss conditional on a breach ($Hackloss$), TOP_i is an indicator for hiring a top-tier external auditor, \mathbf{X}_i is a vector of protocol controls, $\mu_{c(i)}$ are blockchain and industry fixed effects, and $\delta_{t(i)}$ are year-month launch fixed effects. The equations are estimated using either an IV-probit model when the outcome is a binary hack dummy ($HACKDUM$) or 2SLS when the outcome is the continuous hack loss ($Hackloss$).

Results. Table H1 reports the results. The first stage (column 1) shows a strongly negative and statistically significant coefficient of -5.078 ($p < 0.01$) on the *Bartik IV*, indicating that higher exposure to DAO-intensive chains reduces the likelihood of hiring a top-tier auditor. The Anderson-Rubin weak-instrument test yields a statistic of 4.68 ($p = 0.003$), rejecting the null of instrument irrelevance. The second stage results (columns 2 and 3) confirm that engaging a top-tier auditor substantially reduces the risk of security breaches. Specifically, the IV-probit estimates imply that top-tier auditors lower the probability of being hacked ($HACKDUM$) by a coefficient of -1.541 ($p < 0.01$). The 2SLS results show that top-tier auditors also reduce the severity of losses conditional on being hacked ($Hackloss$) with a coefficient of -2.703 ($p < 0.05$). Taken together, the results show that protocols more exposed to DAO-intensive ecosystems are less likely to engage costly top-tier auditors. Nevertheless, hiring a top-tier auditor provides substantial protection against both the likelihood and the severity of subsequent hacks. These findings support the interpretation that the Bartik IV is both relevant and plausibly exogenous, offering credible identification of the effect of audit choice on security outcomes.

Appendix Table H1: Bartik IV: Audit Type (Top-Tier vs Bottom-Tier) and Security Breach Mitigation. This table reports two-stage IV-Probit and two-stage least squares (2SLS) estimates evaluating the effect of engaging top-tier centralized auditors on DeFi protocol security, using a Bartik IV.

$$\text{First Stage: } TOP_i = \beta BARTIKIV_i + \alpha' \mathbf{X}_i + \mu_{c(i)} + \delta_{t(i)} + \varepsilon_i,$$

$$\text{Second Stage: } HackOutcome_i = \delta \widehat{TOP}_i + \theta' \mathbf{X}_i + \mu_{c(i)} + \delta_{t(i)} + u_i,$$

The outcome variable $HackOutcome_i$ corresponds to either (i) a binary indicator equal to one if protocol i was hacked after deployment ($HACKDUM$), or (ii) the dollar amount of loss conditional on a breach ($Hackloss$). The key explanatory variable TOP_i is a binary indicator equal to one if protocol i engaged a top-tier centralized auditor prior to launch. To address endogeneity in the choice of audit quality, we implement a two-stage Bartik instrumental variable strategy. $BartikIV$ is constructed as described in Section 7. All regressions control for protocol-level characteristics, as well as industry, blockchain, and year-month fixed effects. Standard errors are clustered at the industry-by-blockchain level.

VARIABLES	Instrumental Variable (Bartik)		
	(1) TOP First Stage	(2) $HACKDUM$ Second Stage	(3) $Hackloss$ Second Stage
<i>BartikIV</i>	-5.078*** (1.369)		
<i>TOP</i>		-1.541*** (0.307)	-2.703** (1.266)
Observations	192	192	192
(Pseudo) R-squared	0.172	0.178	0.144
Controls	YES	YES	YES
Industry FE	YES	YES	YES
Blockchain FE	YES	YES	YES
Year-mon FE	YES	YES	YES
Weak-IV Diagnostics (AR Test)	4.68, $p = 0.0030$		

Notes: Robust standard errors clustered at the Industry \times Blockchain level. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Appendix Table H2: List of Blockchains sorted by Number of DeFi protocols.

Top 150 Chains by Market Share (# of protocols deployed)								
Chain	Count	Share (%)	Chain	Count	Share (%)	Chain	Count	Share (%)
Ethereum	1174	14.01	X Layer	24	0.29	ThunderCore	9	0.11
Binance	784	9.36	Kucoin	24	0.29	Secret	9	0.11
Arbitrum	699	8.34	Injective	24	0.29	Degen	9	0.11
Polygon	559	6.67	RSK	23	0.27	Wanchain	9	0.11
Avalanche	394	4.70	smartBCH	23	0.27	Stellar	9	0.11
Base	368	4.39	Evmos	23	0.27	Hedera	9	0.11
Fantom	320	3.82	Canto	23	0.27	BEVM	8	0.10
Optimism	252	3.01	Filecoin	22	0.26	Arbitrum Nova	8	0.10
Solana	165	1.97	ZetaChain	22	0.26	Mixin	8	0.10
Kava	135	1.61	Elrond	21	0.25	Wax	8	0.10
Blast	134	1.60	Milkomeda	21	0.25	IOTA EVM	8	0.10
ZKsync Era	118	1.41	EOS	20	0.24	Acala	8	0.10
Linea	110	1.31	Tezos	20	0.24	GodwokenV1	7	0.08
Cronos	107	1.28	EthereumPoW	18	0.21	Kardia	7	0.08
Scroll	100	1.19	BOB	18	0.21	TomoChain	7	0.08
Mantle	97	1.16	zkLink	18	0.21	ShimmerEVM	7	0.08
xDai	69	0.82	Bitlayer	18	0.21	Shiden	7	0.08
Polygon zkEVM	66	0.79	Merlin	17	0.20	Map	7	0.08
Mode	63	0.75	Osmosis	17	0.20	Zilliqa	7	0.08
Manta	60	0.72	Flare	16	0.19	Kroma	7	0.08
Klaytn	58	0.69	Op_Bnb	16	0.19	Songbird	7	0.08
Harmony	58	0.69	BSquared	16	0.19	EOS EVM	7	0.08
Moonbeam	57	0.68	Bittorrent	16	0.19	ApeChain	7	0.08
Aurora	53	0.63	Shibarium	16	0.19	Archway	6	0.07
Metis	53	0.63	Fraxtal	15	0.18	Obyte	6	0.07
Celo	52	0.62	Velas	15	0.18	REI	6	0.07
Moonriver	52	0.62	Conflux	14	0.17	Etherlink	6	0.07
Heco	49	0.58	Kujira	14	0.17	CLV	6	0.07
Aptos	48	0.57	Terraz	14	0.17	NEO	6	0.07
Dogechain	45	0.54	Meter	13	0.16	Horizen EON	6	0.07
Pulse	44	0.53	Stacks	12	0.14	Kadena	6	0.07
CORE	44	0.53	BounceBit	12	0.14	Zircuit	6	0.07
Sui	43	0.51	XDC	11	0.13	Chiliz	6	0.07
Astar	41	0.49	EthereumClassic	11	0.13	Echelon	5	0.06
Telos	40	0.48	Flow	11	0.13	Findora	5	0.06
TON	38	0.45	Neutron	11	0.13	Comdex	5	0.06
Cardano	38	0.45	WEMIX	11	0.13	Theta	5	0.06
Bitcoin	36	0.43	Astar zkEVM	11	0.13	Ripple	5	0.06
OKExChain	35	0.42	Bitgert	11	0.13	CSC	5	0.06
Algorand	31	0.37	Neon	11	0.13	inEVM	5	0.06
Taiko	31	0.37	Radix	11	0.13	Onus	5	0.06
Sei	30	0.36	Zkfair	11	0.13	Heiko	5	0.06
IoTeX	30	0.36	Ergo	10	0.12	Bahamut	5	0.06
Tron	30	0.36	ICP	10	0.12	Parallel	5	0.06
Starknet	30	0.36	re.al	10	0.12	Rollux	5	0.06
Terra	28	0.33	Juno	10	0.12	HAQQ	5	0.06
Near	27	0.32	Cronos zkEVM	9	0.11	Cosmos	5	0.06
Boba	25	0.30	Waves	9	0.11	Icon	5	0.06
Fuse	25	0.30	Oasis	9	0.11	Karura	5	0.06