School of Informatics

Informatics Research Review
Oracles in Blockchain: Analysis and Comparation

B160030
January 2024

Abstract

As blockchain technology and Decentralized Finance (De-Fi) continue grow, Oracles have become a key component acting as a bridge between external information and on-chain environment. The paper categorize Oracle systems into five main types: optimistic mechanisms, majority consensus mechanisms, game theoretic mechanisms, trust environment-based mechanisms, and cryptographic proof based mechanisms, and the paper discuss the advantages and potential vulnerabilities of each mechanism. The paper look into the security of Oracles by evaluating their resistence against threats such as Sybil attacks, DoS attacks etc. Additionally, the paper compare the different Oracles on the key properties of Oracls systems, explore potential improvement on reducing costs and enhancing public verifiability for cryptographic proof based Oracles. The paper also stated its limitations on the lacking of coverage of Oracle mechanisms and lack of analysis of undetected attacks.

Date: Thursday 11th January, 2024

Supervisor: Xingran Ruan

1 Introduction

Blockchain and cryptocurrencies is a vast and rapidly evolving field. According to CoinMarketCap, by December 31, 2023, the total market capitalization of cryptocurrencies had reached $1.66 trillion in less than 15 years since Satoshi Nakamoto created the first Bitcoin block in 2009. During this period of swift development in blockchain technology, numerous innovative technologies have emerged, with Decentralized Finance (De-Fi) being one of the most significant advancements. De-Fi allows financial service providers to conduct operations (lending, exchanging, and staking) in a entirely new way. Unlike traditional financial systems that rely on central servers, De-Fi is built on distributed financial protocols. Detailed discussions on the operational mechanisms and principles of De-Fi will be elaborated in the background section of this paper.

Oracles play an indispensable role in De-Fi systems. Since De-Fi applications operate on blockchain, they require Oracles to access external world data, such as currency prices, weather conditions, and sports results, when needed. These applications run automatically and in real-time on the blockchain, offering users transparency, convenience, and speed. However, this also exposes them to threats from hackers. Research indicates that from April 2018 to April 2022, attacks on De-Fi have led to losses exceeding $3.24 billion, with approximately 15% of these attacks targeting Oracles directly [1]. Therefore, asystematic study of the mechanisms and security performances of Oracles holds significant academic and commercial value.

In this paper, I will initially introduce the fundamental background knowledge of blockchain, cryptocurrencies, and Decentralized Finance (De-Fi) to ensure that readers can fully comprehend these concepts and their applications in modern financial technology. Subsequently, in Chapter 3, I will review the mainstream Oracle systems used in the blochchain and categorize their operational principles into five major types, elaborating on the working mechanism of each category. In Chapter 4, I will explore various attack methods targeted at Oracle systems and assess the resilience of the previously mentioned five types of Oracles against these attacks. Chapter 5 will involve a horizontal comparison of these five types of Oracles from multiple dimensions, aiming to provide guidance for De-Fi developers and propose potential improvements and future directions for Oracle systems. The final chapter will summarize the entire research, highlighting its key findings and significance. Additionally, I will discuss the limitations of the study, candidly pointing out potential shortcomings.

2 Background

2.1 Blockchain

In 2008, Satoshi Nakamoto released bitcoin whitepaper: "Bitcoin: A Peer-to-Peer Electronic Cash System" [2]. With the creation of the Bitcoin Genesis block, the blockchain was born. In the following thirteen years, the blockchain technology has flourished. In the design of Bitcoin, the role of the blockchain is simply to implement a distributed wallet account system based on cryptography and proof of work algorithm. Every blockchain block is created by miners and contains transaction information broadcast by users within the network. The process of mining a block is essentially a computationally intensive task, where miners use high-performance computing devices to find a specific number, known as a "nonce," that meets certain requirement. Specifically, miners combine the transaction information contained in the block, the block hash

of previous block along with this nonce number and then perform a hash operation. The goal of mining is to find a nonce that the hash value is lower than the current target value set by the network (this is part of what is commonly known as the "Proof of Work" algorithm). This process involves a significant amount of computational resources and is highly competitive, as the first miner to find a valid hash value is rewarded with newly created cryptocurrency and the transaction fees included in the block. This mechanism ensures both decentralization and a high level of security and immutability. Changing any data in the previous block is practically impossible, as it would require redoing the computationally intensive proof of work for that block and all following blocks, thus maintaining a secure and unchangeable transaction ledger.

From the users' perspective, users use asymmetric encryption algorithms to ensure transaction security. Under this mechanism, each user possesses a pair of keys: a public key and a private key. The public key is disclosed publicly as the user's wallet address, while the private key is securely kept confidential by the user. When executing a transaction, users sign the transaction information with their private key and then broadcast to the entire cryptocurrency network. Other miners in the network verify the authenticity of the signature. This method of employing asymmetric encryption and digital signatures not only secures the transactions and maintains user anonymity but also enhances the decentralization and immutability of the blockchain ledger. Any attempt to alter transaction information recorded on the blockchain would require to break the secure asymmetric encryption algorithms, which is computationally impracticable.

The method of creating blocks previously mentioned is the Proof of Work (PoW) mechanism. However there are many different block creation algorithms other than PoW. For instance, Ethereum 2.0 implement a Proof of Stake (PoS) mechanism, where the validators of blocks are chosen based on the amount of currency they hold. Although these different algorithms have their unique characteristics and advantages in terms of security, efficiency, and scalability, they do not significantly effect the focus of this paper which is the mechanisms and security analysis of blockchain Oracles. Therefore, this paper will not delve into the specific details and differences of these varying block creation algorithms.

2.2 Smart Contract & De-Fi

The next significant innovation in blockchain technology occurred with the invention of Ethereum. In 2014, Vitalik Buterin released the Ethereum whitepaper [3], marking a significant advancement over Bitcoin. Ethereum's core innovation was the introduction of smart contracts on the blockchain. Smart contracts are codes that can automatically execute on blockchain, bringing a revolutionary extension to blockchain technology. Anyone can write and deploy smart contracts on the Ethereum network using a Turing-complete programming language called Solidity. These contracts require payment in Ether as transaction fees (known as Gas fees) when executed. The compiled and deployed code on the blockchain can be interacted with by any user. This openness and flexibility have made Ethereum a hotbed for blockchain innovation and decentralized application (DApp) development, also leading to the flourishing of the decentralized finance (De-Fi) industry.

Compared to traditional finance, the biggest difference of Decentralized Finance (De-Fi) is its act as a permissionless and decentralized system of smart contracts running on blockchain. In De-Fi, the code of smart contracts is completely transparent, allowing anyone to read, verify its correctness and security. The inherent characteristics of blockchain ensure that once a smart

contract is deployed, it becomes nearly immutable, thereby guaranteeing the transparency and integrity of financial activities. In recent years, the De-Fi has grow rapid, a variety of systems with diverse functionalities and objectives has been released. For instance, stablecoins like DAI [4] and USDT [5] aim to maintain a value binding to the US dollar, platforms like Compound [6] offer decentralized lending services, exchanges like UniSwap [7] enable trading and exchanging of various cryptocurrencies; decentralized insurance services are provided by platforms such as Nexus Mutual [8], and Synthetix [9] allows trading and collateralization of off-chain assets, for instance stocks and commodities. With the rapid development of Decentralized Finance (De-Fi), numerous De-Fi systems have also become prime targets for hackers. Because of the immutability of smart contracts, if hackers discover a method to exploit these smart contracts, counteracting these attacks can be extremely difficult.

2.3 Oracle

Most De-Fi systems rely on Oracles to obtain external data, such as price feeds, which highlights the significance of the security and reliability of Oracles. Consequently, research into the mechanisms and security of Oracles has increasingly become a focus in both academic and industrial area. These studies concern not only the security aspects of the Oracles themselves but also how they impact the robustness and vulnerabilities of the entire De-Fi world.

The primary function of Oracles is to act as a bridge between the external world and the blockchain (on-chain world), providing external data to on-chain De-Fi smart contracts, such as market prices or real-world event information. De-Fi developers have the option to develop their own Oracle or to use existing public Oracle services, such as ChainLink [10]. For the operation of Oracles, it usually requires off-chain machines to collect data from internet then transmit it back to the on-chain Oracle. In most of times, Oracle would introduce multiple data providers from different sources to reduce the impact of malicious actions or errors from individual nodes. Various Oracle models have their unique operational mechanisms and security measures, which will be elaborately discussed in Chapter 3. These models may involve different trust assumptions, data validation methods, and incentive mechanisms to ensure the reliability of the provided data and the effective functioning of on-chain smart contracts.

3 Mechanism

Currently, most of the Oracle system designs fall into five main categories.

3.1 Optimistic Approach

The easiest and simplest approach of acquiring off-chain data is through an optimistic method implement by UMA (Universal Market Access) [11], which optimistically trusts the data uploaded to the blockchain by data providers. In this system, on-chain users raise a request of the off-chain data, and data providers notice the requese, upload the data to the UMA system and receive rewards for their contributions. And the users assume the accuracy and authenticity of the provided data. To reduce potential malicious actions by data providers, UMA introduces a unique dispute mechanism. This mechanism allows other data providers to verify the accuracy of the data within a time slot set by the user. Once data is uploaded, the data is visible to everyone, and during the time slot anyone could verify the data. If an error is detected, others can propose a dispute event, followed by a community vote to determine whether the data is

correct. UMA incentivizes community involvement in the data verification process by rewarding those who successfully challenge incorrect data and penalizing providers of false information. Once the time slot ended, users can be more confident that the data they have received is verified, accurate, and authentic. This mechanism not only enhances the reliability of the data but also increases the transparency and community participation within the system.

3.2 Majority Consensus Approach

The majority consensus method is the most popular approach in the field of Oracles, used by ChainLink 1.0 [10] and Witnet [12]. Focusing on ChainLink 1.0, its operating mechanism is relatively straightforward. Multiple data providers maintain a connection with the ChainLink network. When a user requese data from ChainLink's Oracle smart contract, all registered data providers receive a notification and report their result back to ChainLink within a time slot. ChainLink then aggregates these results and adopts the majority's response as the final output. Each data provider is assigned a reputation score based on their historical performance. If their response aligns with the majority (considered correct by ChainLink), their reputation score is increased according to a specific algorithm; if not, their score decreases, and they may face financial penalties from their staking. Additionally, data providers can enhance their reputation scores by staking more ChainLink tokens, responding to requests faster, and completing more data requests. Users also have the option to select providers with higher reputation scores to respond their requests. This approach not only enhances the decentralization of the system but also increases the difficulty of manipulating the Oracle, thereby bringing greater level of secure and transparency to the system.

3.3 Game Theory Based Approach

The third approach operates on game theory principles used by NEST [13], using arbitrageurs to maintain the accuracy of token prices. This model requires the token price data providers to lock a certain quantity of the token and USDT (US dollar token) into the system. It then permits other arbitrageurs to trade tokens or USDT at this specified price. These traders are also required to offer a new price and deposit the amount of tokens and USDT greater than what the previous provider deposited, thereby creating a sequential price chain within the system. This mechanism incentivizes arbitrageurs to adjust the price to its true market value in a short period of time whenever discrepancies arise. As a result, an adversary would need a large amount of financial resources to beating all participants in the network and manipulate the price reported by the Oracle. While this Oracle machine is highly effective in providing accurate and stable price data, its scope is limited to price information and does not extend to other types of data. The design cleverly use economic incentives and competition to ensure data accuracy, making it a robust solution for price data Oracle in the blockchain.

3.4 Trusted Environment Approach

The fourth type of Oracle machine relies on a trusted computing environment, for example Town Crier [14] and the Android proof concept in Provable [15]. These systems enhance data integrity by providing verifiable proof of the correct execution of code within secure environments, such as Intel SGX [16]. This approach allows users to confidently verify that the data provider is sourcing information from authentic data sources. The fundamental strength of this model lies in the robustness of the trusted computing environment. However, recent studies indicates that

these secure environments are vulnerable to targeted attacks [17]. Once the trusted computing environment is successfully attacked, the data requested by the user can be tampered easily by adversary.

3.5 Cryptographic Proof Based Approach

The final category of Oracle machines operates based on cryptographic proofs over the TLS [18] protocol, used by DECO [19] and TLSnotary [20] implemented in Provable. In DECO, there are three distinct roles: the prover, the verifier, and the server. The prover is responsible for communicating with the server and generate the proof for the accuracy of data to the verifier. The verifier verify the proof then generates an additional proof, proving that the data provided by the prover is accurate and error-free. This approach utilizes cryptographic techniques where both the verifier and the prover share parts of the TLS session key. This shared key ensures the message's tamper-proof nature and enables the data provider to authenticate the data's integrity. However, the process of three-party handshaking and proof verification is computational-intensive. As a result, in Chainlink 2.0 [21], although DECO is implemented, it is not used for verifying every data feed but reserved for use in Second-Tier adjudication events. A significant limitation of this design is that only the initial verifier in the protocol can validate the proof from prover. This limitation poses a potential risk, as it means verifiers and prover could collude to present forged data to external parties. Thus, while this approach offers strong data integrity assurance within a specific verifier-provider context, its utility is somewhat constrained by the lack of public verifiability.

4 Security Analysis

This chapter lists several common methods of attacking target or through Oracle. It is worth noting that this paper only discusses the security of the Oracle machine from the perspective of the mechanism of the protocol, and does not involve the security risks caused by vulnerabilities written in Solidity code or blockchain itself.

4.1 Data Source Manipulation Attack

The function of Oracle is to accurately acquire information from the external world, often involving interactions with internet servers. It is obvious that the most direct method to manipulate Oracle data is by tampering the data at source. For instance, if the runner of server on the internet realizes that manipulating data server provided could yield a significant amount of profit in De-Fi, tampering the data by runner of server become possible. Additionally, if there are vulnerabilities in the server of the data source, allowing adversaries to temporarily manipulate the information, similar outcomes can be achieved. Among the five Oracle mechanisms previously described, game theory-based Oracles (like NEST) can effectively resist such attacks, as they do not rely directly on external data servers but rather ensure the accuracy of price information through an internal price chain. Majority consensus mechanisms (like ChainLink) can also have against these attacks to some extent, since data providers might use different sources, and attacking all sources simultaneously is highly improbable. However, if users specify a particular data source, majority consensus mechanisms might fail to detect the tampering. Similarly, Oracles using optimistic mechanisms, such as UMA, may be vulnerable in the face of data source manipulation. In these systems, community members only verify whether the submitted data corresponds with the specified source. Trust environment based (such as Town

Crier) and cryptographic proof based (like DECO) Oracles cannot defend against the manipulation of the data source itself since they only ensuring correct data retrieval from the data source servers.

4.2 Sybil Attack

In decentralized systems, Sybil attacks represent a common and destructive security threat [22]. Within blockchain and decentralized Oracles, such attacks typically involve creating numerous fake identities or controlling majority nodes to manipulate or break consensus based protocols. For instance, in majority consensus based Oracles like ChainLink, if an attacker gains control over the majority of data provider nodes, they could theoretically control the Oracle's output. However, ChainLink lower the feasibility of such attacks by implementing reputation and staking mechanisms.For Oracles with optimistic mechanisms, such as UMA, if attackers can manipulate the majority of the community, they could potentially control the data passed through the Oracle. However, executing such an attack is challenging in practice due to the need for majority manipulation of community members. Meanwhile, trust environment and game theory based Oracles, like Town Crier and NEST are natually resistant to Sybil attacks as they do not rely on a majority node consensus. While cryptographic proof based Oracles (like DECO) is theoretically be safe from Sybil attacks, in practical applications, they often introduce multi-node verification mechanisms. Because as discussed in Chapter 3, cryptographic proof mechanisms based on the three-party handshake protocol involving TLS key splitting, which lack of public verifiability. If attackers control the majority of verifying nodes and can forge a fake cryptographic proof of the verification, they might still execute a Sybil attack.

4.3 Delay Attack & DoS Attack

Delay attacks and Denial of Service (DoS) attacks are similar in their execution methods and objectives, typically involving the disruption of node operations, or delays to respond by node operators for specific goals. These attacks are particularly effective against price Oracles for cryptocurrencies and game result Oracles for sports event gambit Oracles, as they can create huge profit for attackers. For instance, an attacker might initiate a delay or DoS attack during a sharp decline in cryptocurrency prices, then engage in high-leverage shorting in some cryptocurrency leverage De-Fi contracts. Due to the delayed update of information by the Oracle, the plummeting prices in the De-Fi system are not immediately reflected, providing the attacker with significant arbitrage opportunities. In trusted environment-based Oracle systems, such attacks are feasible, as an attacker who is part of the data provider group can simply refuse to respond to data requests when they spot an arbitrage opportunity. However, for Oracles using optimistic, majority consensus, or game theory-based mechanisms, the impact of such attacks is likely to be minimal. Implementing these attacks would require control over all community members or all nodes in the network, which is often impractical. Similarly, in practical applications, cryptographic proof based Oracles are also less effective to these attacks, as they typically involve a variant of the majority consensus mechanism for data verification and provision as mentioned before. This means that even if individual nodes are attacked or refuse service, other nodes can still provide accurate data, thereby maintaining the overall stability and reliability of the system.

4.4 Freeload Attack

Freeload Attack is a form of attack against distributed Oracle systems. Taking ChainLink as an example, this attack involves data providers monitoring and copying the data submitted by other providers on the blockchain as their own. This behavior leads to a significant threat to the integrity of the Oracle system. Firstly, freeloading reduces the diversity of data sources, weakening the Oracle's resistance to Sybil attacks and data source pollution. Secondly, it break the economic incentive mechanism of the Oracle, reduce the system's sustainability and trustworthiness. To counter Freeload Attacks, ChainLink introduce a Commit-Reveal mechanism. Under this scheme, nodes initially submit the hash of their data rather than the plaintext data directly. After a set period or once a sufficient number of nodes have submitted their hash, the nodes then submit the plaintext data. ChainLink's smart contract automatically verifies that the plaintext data matches the previously submitted hash. Due to the irreversible nature of hashing algorithms, this prevents attackers from decrypting the data content from its hash, thereby effectively lower the possibility of freeloading. Trust environment based and game theory based Oracle systems are inherently immune to such attacks due to their unique designs. Similarly, cryptographic proof based Oracles, with mechanisms like TLS key splitting and encrypted data transmission protocols of HTTPS, ensure that other nodes cannot forge a fake cryptographic proof by merely eavesdropping on transmitted information. Thus this types of Oracles are also resistant to Freeload Attacks. For Oracles using optimistic mechanisms, since only one data provider submits information to the blockchain at a time, the opportunity for freeloading is not exist.

4.5 Front-Running Attack

Front-Running attack is a attack targeting De-Fi systems, rather than exploiting directly attacking Oracles. In such attacks, adversary monitor information on the blockchain to early access unconfirmed data soon to be released by Oracles. If this information indicates potential arbitrage opportunities, adversary initiate arbitrage transactions, offering higher transaction fees to ensure it execute before the Oracle data is confirmed. As these attacks use Oracle information to target De-Fi applications and not the Oracles themselves, it is challenging for Oracles to directly counteract them. Though existing some countermeasures like randomization in transaction processing, Commit-Reveal Schemes, or time-delay mechanisms, front-running remains prevalent in the blockchain today.

4.6 Replay Attack

Replay attack involve the use of already transpired Oracle information on the blockchain, copying and resending it at a specific time. Similar to front-running, replay attacks do not directly harm Oracles but exploit transmitted data to impact De-Fi systems that using Oracles. Such attacks can disrupt De-Fi contracts relying on real-time information by replaying outdated data. Present-day Oracles have implemented various effective measures to resist replay attacks, including timestamp verification, digital signatures, and one-time sequence numbers.

4.7 Bribery Attack

Bribery attack result similarly to previously mentioned Sybil and DoS attacks. Under this circumstance, attackers may bribe nodes within the Oracle network to produce specific data

outputs, end up similarly with Sybil attacks. If a Oracle can defend Sybil attacks, it is also prepared to defend against this kind of bribery attack. Similarly, bribery attack can be down with miners, with attackers bribing them to exclude certain Oracle data upload transactions, similar to the objectives of DoS or delay attacks but targeting miners instead. However, given the extensive miner community and high mining computational difficulty, such attacks remain theoretical and impractical in execution.

5 Comparation and Potential Improvement

Considering the pros and cons of different Oracles, as well as the important features of blockchain, De-Fi and the security of the Oracle itself, the following table can be summarized.

	Optimistic Oracle Mode	Majority Consensus Oracle Model	Game Theory Based Oracle Model	Trusted Environment Oracle Model	Cryptographic Proof Based Oracle Model
Not reliant on data source accuracy	🟩	🟩	🟩	🟥	🟥
Multiple data type support	🟩	🟩	🟥	🟩	🟩
High timeliness	🟥	🟨	🟨	🟩	🟩
Sybil Attack resistance	🟨	🟨	🟨	🟩	🟩
DoS Attack resistance	🟩	🟩	🟩	🟨	🟨
No need for specific hardware	🟩	🟩	🟩	🟥	🟩
No need for specific communication protocol	🟩	🟩	🟩	🟥	🟥
Low gas cost	🟩	🟩	🟩	🟨	🟥
Integrity can be verified by anyone	🟥	🟥	🟨	🟩	🟥
High attack difficulty	🟥	🟨	🟩	🟨	🟩

Fig. 1: Comparation

In different De-Fi applications, the importance of certain data property is different base on the type of application. For instance, in De-Fi apps that use weather data, accuracy might be more important than the timeliness of the data because weather predictions don't change as quickly as live market data. But for De-Fi apps like gambit contracts on sports games, it's crucial to have the latest results quickly because the fairness and effectiveness of these bets rely on having timely and accurate information about the game outcomes. Similarly, for decentralized exchanges (DEXs), having accurate and up-to-date cryptocurrency prices is important since trading decisions and market depend on the latest price information. Moreover, De-Fi applications involving complex financial derivatives (like as cross-chain aggregators, on-chain quantitative trading robots) might need to consider more aspects of data quality, such as the reliability and diversity of data sources, to ensure transparency and fairness in trading. Therefore, when designing and using De-Fi applications, developers and users should consider different data qualities based on specific scenarios and needs to maximum the performance and security of the system.

Base on the above mentioned graph, theoretically, Oracles based on cryptographic proofs offer a high level of security, but they come with relatively high costs. Future developments in Oracle technology could focus on improving this type of Oracle. These improvements could be approached from two main aspects.

Firstly, reducing the cost of cryptographic prove and verification. Currently, the field of zero-knowledge proofs is rapidly developing, and several methods have been developed to reduce the time and space complexity of this process, such as Groth16 [23], FRI (Fast Reed-Solomon Interactive Oracle Proofs) [24], Recursive SNARKs (Succinct Non-interactive Arguments of Knowledge) [25]. However, applying these advanced zero-knowledge proof techniques to cryptographic proof based Oracles remains an under-explored area. Combining these two could significantly reduce the cost associated with cryptographic proof based Oracles.

Another direction for improvement is enhancing the public verifiability of cryptographic-proof-based Oracles. Current systems (like DECO) lack this feature. Only verifiers who directly interact with the prover can verify data integrity. If these proof processes could be made verifiable by anyone, achieving public verifiability, it would be an ideal solution from a security perspective. Users would be able to rely on reliable cryptographic processes to verify data accuracy, rather than just relying on the honest majority in the network. Although researchers are currently exploring this direction, significant breakthroughs have yet to be achieved.

6 Conclusion and Limitation

This paper delves into the critical role of Oracles in blockchain and Decentralized Finance (De-Fi) systems. It begins by thoroughly analyzing of existing Oracle systems into five types by their operational mechanisms: optimistic mechanisms, majority consensus mechanisms, game theoretic mechanisms, trust environment based mechanisms, and cryptographic proof based mechanisms. While detailing the operating principles of these Oracles, the paper also reveals their strengths and weaknesses. Furthermore, the paper conducts a security analysis of these five types of Oracle on some common attacks they might encounter, offering a comprehensive evaluation of their resilience to various types of threats. Chapter four summarizes the key attributes of Oracle systems and compares the five types, providing guidance for De-Fi developers on choosing the right Oracle solution based on their specific needs. After that, the paper suggests that developments in Oracle technology may focus on improvements in cryptographic proof based mechanisms, particularly in reducing costs and enhancing public verifiability.

Nevertheless, the paper also contain some limitations. Due to the rapid development of blockchain technology, new and unique Oracle mechanisms that may not have been covered could emerge. Additionally, the discussion on security threats may not be exhaustive; the safety issues discussed are well-known and widely studied, whereas future threats remain unknown, just like we will never know what will happen in the future. Therefore, the paper can only discuss based on the currently known security vulnerabilities. Moreover, the paper does not cover code architectural vulnerabilities (such as re-entrancy attacks) for De-Fi software when calling Oracles.

In conclusion, this paper provides valuable insights and analyses for understanding and applying Oracles, disregarding its limitations. It offers a solid knowledge base for participants in the De-Fi field and propose a thoughtful suggestion for the future direction of Oracle development.

References

[1] Liyi Zhou, Xihan Xiong, Jens Ernstberger, Stefanos Chaliasos, Zhipeng Wang, Ye Wang, Kaihua Qin, Roger Wattenhofer, Dawn Song, and Arthur Gervais. Sok: Decentralized finance (defi) attacks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2444–2461. IEEE, 2023.

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[3] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 3(37):2–1, 2014.

[4] Maker Team. The dai stablecoin system. *white paper*, 2017.

[5] Tether. Tether: Fiat currencies on the bitcoin blockchain. *white paper*, 2014.

[6] Robert Leshner and Geoffrey Hayes. Compound: The money market protocol. *White Paper*, 2019.

[7] Yi Zhang, Xiaohong Chen, and Daejun Park. Formal specification of constant product (xy= k) market maker model and implementation. *White paper*, 2018.

[8] Hugh Karp and Reinis Melbardis. Nexus mutual: A peer-to-peer discretionary mutual on the ethereum blockchain. *Whitepaper. https://nexusmutual. io/assets/docs/nmx_white_paperv2_3. pdf*, 2017.

[9] Norbert Bodziony. Synthetix whitepaper. *White Paper*, 2020.

[10] Steve Ellis, Ari Juels, and Sergey Nazarov. Chainlink: A decentralized oracle network. *Retrieved March*, 11:2018, 2017.

[11] Universal Market Access. *UMA Protocol: How does UMA's Oracle work?*, 2022.

[12] Adán Sánchez de Pedro, Daniele Levi, and Luis Iván Cuende. Witnet: A decentralized oracle network protocol. *arXiv preprint arXiv:1711.09756*, 2017.

[13] nestprotocol.org. Nest decentralized probabilistic virtual machine model. 2022.

[14] Fan Zhang, Ethan Cecchetti, Kyle Croman, Ari Juels, and Elaine Shi. Town crier: An authenticated data feed for smart contracts. In *Proceedings of the 2016 aCM sIGSAC conference on computer and communications security*, pages 270–282, 2016.

[15] provable.xyz. Android proof: Authenticated data gathering using android hardware attestation and safetynet. 2019.

[16] Victor Costan and Srinivas Devadas. Intel sgx explained. *Cryptology ePrint Archive*, 2016.

[17] Jo Van Bulck, Marina Minkin, Ofir Weisse, Daniel Genkin, Baris Kasikci, Frank Piessens, Mark Silberstein, Thomas F Wenisch, Yuval Yarom, and Raoul Strackx. Foreshadow: Extracting the keys to the intel {SGX} kingdom with transient {Out-of-Order} execution. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 991–1008, 2018.

[18] Tim Dierks and Eric Rescorla. The transport layer security (tls) protocol version 1.2. Technical report, 2008.

[19] Fan Zhang, Deepak Maram, Harjasleen Malvai, Steven Goldfeder, and Ari Juels. Deco: Liberating web data using decentralized oracles for tls. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1919–1938, 2020.

[20] Pawel Szalachowski. Blockchain-based tls notary service. *arXiv preprint arXiv:1804.00875*, 2018.

[21] Lorenz Breidenbach, Christian Cachin, Benedict Chan, Alex Coventry, Steve Ellis, Ari Juels, Farinaz Koushanfar, Andrew Miller, Brendan Magauran, Daniel Moroz, et al. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. *Chainlink Labs*, 2021.

[22] John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

[23] Jens Groth. On the size of pairing-based non-interactive arguments. In *Advances in Cryptology–EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II 35*, pages 305–326. Springer, 2016.

[24] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In *45th international colloquium on automata, languages, and programming (icalp 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[25] Sean Bowe, Jack Grigg, and Daira Hopwood. Recursive proof composition without a trusted setup. *Cryptology ePrint Archive*, 2019.