EIP Security Analysis: Application Program Standards, Attack Events, and Security Vulnerabilities

Xueyan Tang*
Salus Security, Beijing, China, Corresponding author's e-mail:
mirror.tang@alumni.stanford.edu

Yuying Du
Salus Security, Beijing, China
yuying@salusec.io

Zhengyang Wang
Salus Security, Beijing, China
zhengyang@salusec.io

Shawn Chong
Salus Security, Beijing, China
shawn@salusec.io

ABSTRACT

With the ongoing advancement and widespread adoption of blockchain technology, Ethereum has established itself as the foremost platform for executing smart contracts.As a focal point of the industry, the security and stability of the Ethereum ecosystem have become crucial. Ethereum Improvement Proposals (EIPs) play a vital role in promoting the technological progress of Ethereum and providing direction for innovation in the blockchain field. ERCs, as a type of EIP, define application-level standards and conventions that are crucial for promoting innovation and ensuring the scalability and maintainability of the system. In particular, some ERCs, such as the ERC-20 token standard, have become the foundation of the Ethereum ecosystem. However, if EIPs have security issues, any security vulnerabilities or design flaws can have severe consequences, including financial losses, data leaks, and damage to the reputation of the entire Ethereum network. This paper primarily focuses on the security analysis of application standard proposals (ERC). We conducted research on the current state of EIP security, performed case studies, and provided security recommendations. The goal is to gain a comprehensive understanding of the security features and potential risks of these proposals, and to propose practical solutions to enhance the security of EIPs.

CCS CONCEPTS

• **Security and privacy**; • **Cryptography**;

KEYWORDS

Blockchain, Ethereum, Security, EIP

ACM Reference Format:
Xueyan Tang*, Yuying Du, Zhengyang Wang, and Shawn Chong. 2023. EIP Security Analysis: Application Program Standards, Attack Events, and Security Vulnerabilities. In *2023 7th International Conference on Electronic Information Technology and Computer Engineering (EITCE 2023), October 20–22, 2023, Xiamen, China*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3650400.3650609

EITCE 2023, October 20–22, 2023, Xiamen, China

ACM ISBN 979-8-4007-0830-5/23/10
https://doi.org/10.1145/3650400.3650609

1 INTRODUCTION

With the continuous development and popularization of blockchain technology[1], Ethereum[2], as the world's leading smart contract platform[3], its ecosystem's security and stability have become the focus of attention in the industry. Ethereum Improvement Proposals (EIPs)[4], as a core component of the Ethereum community, are design documents that provide information or describe new features, processes, or environments of Ethereum. The introduction of EIPs not only promotes the technological progress of Ethereum, but also provides direction for innovation in the entire blockchain field.

EIPs can be divided into three main types: Standard Track EIPs, Meta EIPs, and Informational EIPs. Among them, Standard Track EIPs are further divided into Core, Networking, Interface, and ERC (Application-level standards and conventions). In particular, some ERCs have become the cornerstone of the Ethereum ecosystem. ERC not only defines a series of widely accepted application-level standards and conventions, but also promotes interoperability[5] and consistency of decentralized applications[6]. These standards play a key role in promoting innovation and ensuring the scalability and maintainability of the system. This paper mainly focuses on the research of application standards proposals.

By defining shared application-level standards and conventions, these proposals provide developers with guidelines to build more reliable and interoperable decentralized applications. However, if EIPs have security issues or are improperly used by developers, it not only affects the stability and reliability of individual applications, but also directly impacts the reputation and user trust of the entire Ethereum network. Any security vulnerabilities or design flaws can lead to financial losses, data leaks, or other serious consequences, causing long-term negative impacts on the entire ecosystem[7].

In response to the above issues, our research focuses on the security of EIPs, specifically the security analysis and specification standards of application standards proposals. Our goal is to gain an in-depth understanding of the security features and potential risks of these proposals and propose practical solutions to enhance the security of EIPs. Through this research, we call for the enhancement of security and compliance awareness for EIPs, and hope to provide valuable insights and contributions to the Ethereum community, promoting the development of a more secure, healthy, and sustainable blockchain ecosystem.

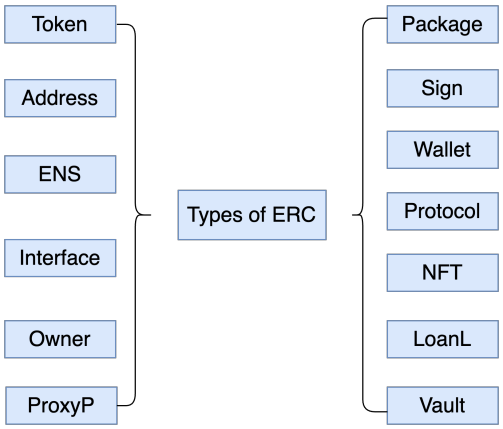

Figure 1: The various types of ERC standards currently present in Ethereum

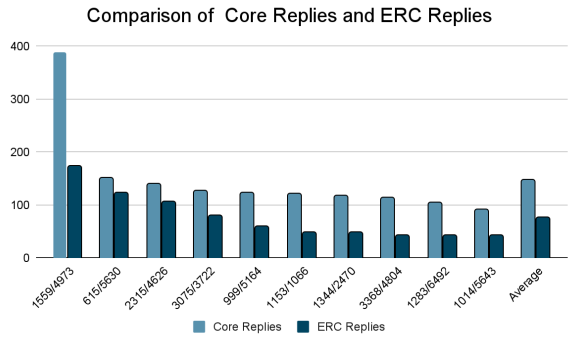

Figure 2: Comparison of the top 10 reply counts for core and ERC proposals on the ethereum magicians forum

2 BACKGROUND

Considering that smart contracts on Ethereum control a significant amount of assets, and some EIPs, such as ERC-20[8] and ERC-721[9], have become fundamental components of cryptocurrencies and decentralized applications, any potential security vulnerability or defect could result in significant financial losses and a collapse of trust in the entire ecosystem. Only through rigorous and continuous ERC security audits can we ensure the health, stability, and long-term prosperity of the Ethereum ecosystem. However, the current state of research and specification standards for EIP security has failed to garner attention.

2.1 The current state of research on the security of ERCs

In the following Figure 1, we can observe the various types of ERC standards currently present in Ethereum. This indicates that ERCs have a wide range of influence and are exposed to multiple security threats.

Among the various classifications of ERCs shown, there should not be a prioritization of security. Any type of vulnerability can potentially result in significant financial loss or other forms of harm to users.

For instance, token contracts that users frequently interact with are susceptible to security risks. Malicious attackers can exploit vulnerabilities to profit at the expense of user interests. If the project's code relies on the control of the owner, then any issues with owner-type ERCs would allow malicious attackers to gain direct control over the smart contract, resulting in substantial losses. Therefore, it is crucial that we prioritize the security concerns of ERCs and take preventative measures. This is not redundant work, but rather a necessary step to avoid potential risks.

As of July 4th, 2023, Ethereum has passed a total of 74 ERCs, while there have been 52 proposals for changes to the core protocol. The official explanation for the core proposals is that they involve improvements that require consensus, as well as changes that may be related to discussions among "core developers" but do not necessarily require consensus. Meanwhile, the level of discussion surrounding ERCs is significantly lower compared to proposals for changes to the core protocol. This results in security issues in application standard proposals being less likely to receive attention. Figure 2 shows a comparison of the top 10 reply counts for core and ERC proposals on the ethereum magicians forum.

From Figure 2, it is evident that there is a significant disparity in the number of replies between the top 10 Core and ERC proposals. Generally, the number of replies for Core proposals exceeds that of ERC proposals. The largest difference in reply count can be observed between the first EIP-1559 and ERC-4973, with the reply count for EIP-1559 being almost twice as high as that of ERC-4973. Similarly, there is a notable difference in reply count between the tenth-ranked EIP-1014 and ERC-5643. The same pattern can be observed in the comparison of average reply counts, indicating that there is much less attention given to ERCs compared to Core EIPs.

In the research on EIP security, we utilized Google Scholar as a medium and searched for results using "EIP" and "ERC" as keywords. We examined the content of the first 40 pages of search results and summarized the research topics, quantity, and corresponding EIPs. The findings are presented in Table 1.

Table 1 shows the academic papers we have collected about ERC/EIP, with a total of 21 articles, out of which only 3 have studied the security of ERC/EIP.

Similarly, when searching on the Ethereum Magicians forum and filtering all search results, we found zero sections discussing and researching security issues related to EIP/ERC. This suggests that there is relatively little research on the security of ERC/EIP at present, indicating a lack of attention to EIP security by people.

2.2 The formatting specifications for EIP proposals

Currently, the formatting specifications for EIP proposals provide detailed requirements for title, description, author, discussion-to, etc. However, the requirements for the security aspect are relatively lacking. Although the specifications include compatibility, test cases, and security considerations, there are no strict details requirements.

Table 1: Google Scholar search results

EIP/ERC	Research Content	Quantity	Reference
EIP-1559	Transaction Fees	6	[10], [11], [12], [13], [14], [15]
	EIP-1559 improved Ethereum's fees.	2	[16], [17]
	Verification Method of EIP-1559 Contracts (Security)	1	[18]
ERC721	NFT Ecosystem Interactions.	1	[19]
	Reversible ERC721	1	[20]
	Shared Mobility Platform for Vehicles	1	[21]
	Creation and Deployment	1	[22]
	The Role of ERC20 Tokens in the Financial Revolution	1	[23]
	Smart Contracts for Universal ERC20 Interaction	1	[24]
	Centralization Risks of ERC-20 Tokens (Security)	1	[25]
	ERC-20 Token Price Volatility	1	[26]
	Deep Learning-Based Security Vulnerability Detection	1	[27]
	Creation and Deployment	1	[28]
ERC	Backdoor Threats in ERC Token (Security)	1	[29]
	Studying Market Fairness	1	[30]

We selected 20 ERCs from those that have been approved, are under review, or in draft status to examine if they have test cases. Out of these randomly selected 20 examples, only 7 ERC proposals have actual test cases, which accounts for only one-third of the total. The lack of test cases makes it challenging to identify potential security issues. Among the sampled ERC proposals, 14 of them have substantial security considerations, accounting for 70% of the total. However, some of these EIPs provide overly brief content in their "Security Considerations" section, which does not offer sufficient assistance to users and developers.

Based on the analysis above, we have noticed a lack of research on the security of ERC, despite its significance. Therefore, it is crucial to conduct research on ERC security. By studying the existing security vulnerabilities in ERC and organizing them, we aim to provide a clearer understanding of ERC's security issues. Additionally, we will propose measures and recommendations to address these security concerns, ensuring users can have a greater sense of trust and security when using ERC. Moving forward, we will employ methods such as case analysis and qualitative analysis to examine relevant data and information. We will categorize EIPs and analyze previous security incidents to identify strong and weak correlations in ERC security. Finally, we will provide suggestions and countermeasures to address these issues.

3 EXPERIMENTAL

During the experiment, we collected audit reports of ERC proposals and their related applications. We classified the vulnerabilities that were identified, tabulated the frequency of occurrence, and identified the causes of high-risk and above vulnerabilities. By organizing the types of vulnerabilities found in the ERC audit reports and analyzing and summarizing 160 attack incidents from the past year, we identified a total of 14 events related to ERC. We analyzed both strongly and weakly related issues with ERC, including the problems that arose and the corresponding solutions.

3.1 ERC Security Issue Types

We conducted a comprehensive data collection from various sources across the internet, obtaining a total of 11 audit reports for 13 ERC proposals and their related applications. By analyzing the content of these reports, we categorized the identified vulnerabilities into four types: Overflow, Logic issue, Access control, and Lack validation. We then compiled the occurrence frequency of each vulnerability type into Table 2

From Table 2, it can be seen that Validation vulnerabilities in the 13 ERC proposals and related applications appear most frequently, followed by Access control.

Table 3 displays the high-risk and above vulnerability information from the audit report of the ERC proposal and related applications, including the categories and descriptions of the vulnerabilities.

Furthermore, a summary based on the issue title and occurrence frequency is presented in Table 4. From this, it is evident that token issues and logic problems remain prevalent in ERC, posing a risk to users' fund security.

As shown in Table 3 and Table 4, among all the high-risk and above vulnerabilities, logic issues have the highest count. Out of a total of 14 high-risk vulnerabilities, 6 are related to logic issues, indicating that even though logic issues are relatively fewer in number, they can lead to severe consequences. Additionally, it is observed that out of the 74 ERC standards, only 5 have undergone specific audits targeting ERC proposals, revealing a lack of sufficient security measures for ERC standards. Furthermore, some ERC standards do not have standardized code repositories.

Based on the current audit report, the following findings are obtained (as shown in the accompanying Table 5): Among the 13 types of ERC labels, NFT and ENS have the highest number of issues, followed by Token, while the others have comparatively fewer issues. Additionally, out of the 160 attack incidents occurring between July 4, 2022, and July 4, 2023, 15 are related to ERC.

Table 2: frequency of each vulnerability type

Vulnerability type	Overflow	Logic issue	Access control	Validation
Frequency	3	7	8	11

Table 3: vulnerability information from the audit report of the ERC proposal and related applications

Number	Issue Title	Category
137	Memory corruption in Buffer	Overflow
137	SimplePriceOracle.price is susceptible to integer overflow	Overflow
137	ETHRegistrarController.register is vulnerable to front running	Logic Issue
137	SOA record check on the wrong domain	Validation
2098	ECDSA signature malleability	Validation
3525	ERC-721 token receiver contract may not be able to receive ERC-3525 tokens due to an issue in the _checkOnERC721Received() function	Logic Issue
3525	ERC-3525 receiver contract may not be able to receive ERC-3525 values due to an issue in the _checkOnERC3525Received() function	Logic Issue
4337	Deposit manipulation	Validation
4337	Incorrect prefund calculation	Logic Issue
4337	Paymasters can spend locked stake	Access Control
4337	Token transfers may fail silently	Validation
4337	Incorrect gas price	Validation
4337	Duplicate validation gas accounting	Logic issue
4626	Vault deposits can be front-run and user funds stolen	Logic Issue

Table 4: Frequency of vulnerabilities based on Issue Title

Issue Title	Frequency	Category
Lost tokens	3	Logic Issue
Front run	2	Logic issue
Incorrect calculation	2	Validation
Variable Overflow	1	Overflow
Memory Overflow	1	Overflow
Weak crytpo	1	Validation
Deposit manipulation	1	Validation
Privileged function	1	Access Control
Wrong check	1	Validation

3.2 EIP security incidents

After organizing the historical audit reports and incidents of attacks, a total of 8 cases strongly related to ERC (problems arising from auditing ERC protocol code/problems arising from ERC application) were compiled, as well as 5 cases weakly related to ERC (problems arising from insecure coding by developers). Case analysis and code are included at this link: https://github.com/Mirror-Tang/EIP_Security_Analysis_Application_Program_Standards_Attack_Events. Table 6 summarizes the identified issues, scenarios in which they occur, and corresponding solutions.

4 RESULTS AND DISCUSSION

This paper aims to delve into the security of EIP (Ethereum Improvement Proposal), with a specific focus on ERC (Ethereum Request for Comments) standards and conventions at the application level. We have organized ERC into 13 different types, where each type represents unique functionality and purpose, thus adding diversity to the Ethereum ecosystem. This richness provides developers with a wide range of opportunities for application development, spanning various fields such as finance, identity verification, and non-fungible token (NFT) markets. However, through extensive literature research and forum analysis, we have found limited research on the security aspects of EIP, with less emphasis on the security issues of EIP. By analyzing the ERC proposals that have been approved in Ethereum Improvement Proposals, we have also discovered that many proposals have not fully considered security considerations and other security aspects, which is detrimental to the long-term development of EIP.

Furthermore, we have collected audit reports on ERC proposal codes and related applications, categorized vulnerability types, analyzed historical security incidents of EIP, and compiled a list of issues related to ERC (problems encountered through auditing ERC protocol codes/problems encountered in ERC applications). We have found that most vulnerabilities are caused by logical issues in code design. The reasons for these issues can be attributed to the proposers not providing detailed security considerations or test cases for developers to reference, the lack of reliable third-party audit of the proposal code, leading to undiscovered and unpatched

Table 5: Frequency of ERC labels issues.

ENS	Token	Address	Interface	Owner	Package	Sign
6	3	0	1	0	0	0
Protocol	NFT	Vault	Proxy	Loan	Wallet	
0	6	1	1	0	0	

Table 6: Information about the identified issues

Issues Identified	Scenarios Observed	General Solutions
Integer Overflow	Issues Caused by Numerical Overflow in Arithmetic Operations.	1.Add integer overflow checks before numerical operations. 2.Use programming language versions with built-in overflow detection.
Unsafe Compiler Version	Vulnerabilities Due to Varied Compilation Versions.	EIP proposes secure compilation versions for different applications.
Memory Overflow	Variable Storage Problems Due to Improper Buffer Memory Allocation.	1.Apply numerical checks for variables after using buffers. 2.Avoid using buffer or memory-allocation-dependent data.
Front-run	Attacks Due to Improper Permission Control Allowing Early Execution of Functions.	Clearly define the calling permissions for each function and restrict callers accordingly.
Reentrancy	Arbitrary Code Execution Due to Issues with the ”fallback” Function.	1.Use secure functions (e.g., transfer). 2.Employ the checks-effects-interactions pattern. 3.Introduce mutex locks.
Access control	Unauthorized Execution of Functions by Non-Specific Users Due to Inadequate Permission Control.	Clearly establish function call relationships and enforce distinct permissions for different functional functions.
Unchecked Return Value	Execution of Subsequent Functions Without Checking the Return Value of Previous Functions.	If the current function calls other functions, ensure thorough checking of return values from all called functions.
Lock token	Inability to Withdraw Funds Due to Lack of Support for Certain ERC Protocols.	Monitor the announcement of new proposals and token types in real-time; update token transfer functions as needed to prevent such issues.
Signature/Replay Vulnerability	Replay Attacks and Other Issues Due to the Use of Vulnerable Signature Functions.	Utilize known and secure encryption methods and functions for cryptographic-related operations.
Usage of delegatecall	Arbitrary Code Execution by Allowing Users to Fill Data for delegatecall Invocation in Functions.	1.Avoid using delegatecall. 2.Restrict usage via whitelists.
DoS Caused by External Calls	DoS Attacks Arising from Exceptions in External Function Calls.	1.Avoid combining calls in a single transaction. 2.Implement exception detection and handling.
Race Condition	Competition Issues Arising from the Order of Function Invocation and Gas Usage.	1.Use signature/hash verification. 2.In subsequent functions dependent on prior ones. 3.Apply mutex locks.
DoS Due to Array Length	DoS Problems Due to Array Length Being Excessively Large for Array Traversal.	For variables of unknown length, impose limits based on the maximum data size.
Ethereum Features	Issues Arising from Neglecting ETH Features, such as Using ETH Balance for Logic Checks but Allowing Forced Transfer.	1.Avoid logic checks using ETH for specific scenarios. 2.EIPs should consider the current features of ETH.

vulnerabilities, and varying levels of expertise among proposers, resulting in problematic ERC codes being proposed.

Although we have identified some security issues in these audit reports, the number of audits specifically focusing on ERC proposal codes is relatively low, suggesting that there may be undiscovered security vulnerabilities pertaining to ERC. Finally, we analyze cases strongly or weakly related to ERC and propose remedial solutions.

Based on the comprehensive research mentioned above, we propose the following suggestions regarding the security issues of ERC:

1. EIP should incorporate specific specifications that require security considerations, standardized test cases, compatibility, and important usage considerations.

- It is advisable to implement permission controls for functions.

- Specify recommendations for safe compilation versions and the use of different compiled version libraries.

- Define the usage scenarios and input restrictions for each function.

- Set limitations on the frequency of function calls, such as preventing reentrancy.

2. When implementing ERC proposal codes, it is recommended to engage third-party security teams for timely auditing and discovery of security issues. During the audit process, meticulous scrutiny of financial mathematical models is crucial.

3.Extra caution should be exercised when dealing with functions or operations related to finance, token handling, etc. Professional testing and comprehensive analysis should be conducted.

4. Encouraging more individuals and organizations to conduct research on the security aspects of EIP/ERC in order to fill the existing gaps in security research in this field.

It is particularly noteworthy that there is currently very limited research on the security of ERC, further highlighting the importance of the ERC security research conducted in this article. The existing research gap needs to be filled, and therefore, this research holds significant value in terms of examining the security of ERC, helping to address the overall lack of research in the ERC field, and providing support for the continuous and healthy development of ERC. Further research can be conducted on this foundation to delve deeper into security analyses. We call on more individuals to focus on EIP security.

REFERENCES

[1] Qi Zhang, Hai Lv, Junwei Ma, Jingye Li, and Jieni Zhang. 2021. Overview of Blockchain Data Privacy Protection. In Proceedings of the 2021 3rd Blockchain and Internet of Things Conference (BIOTC '21). Association for Computing Machinery, New York, NY, USA, 53–58. https://doi.org/10.1145/3475992.3476000.

[2] Ting Chen, Zihao Li, Yuxiao Zhu, Jiachi Chen, Xiapu Luo, John Chi-Shing Lui, Xiaodong Lin, and Xiaosong Zhang. 2020. Understanding Ethereum via Graph Analysis. ACM Trans. Internet Technol. 20, 2, Article 18 (May 2020), 32 pages. https://doi.org/10.1145/3381036.

[3] Morena Barboni, Andrea Morichetta, and Andrea Polini. 2023. Smart contract testing: challenges and opportunities. In Proceedings of the 5th International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB '22). Association for Computing Machinery, New York, NY, USA, 21–24. https://doi.org/10.1145/3528226.3528370.

[4] Retrieved August 17, 2023 from https://ethereum.org/zh/eips/.

[5] Gang Wang, Qin Wang, and Shiping Chen. 2023. Exploring Blockchains Interoperability: A Systematic Survey. ACM Comput. Surv. 55, 13s, Article 290 (December 2023), 38 pages. https://doi.org/10.1145/3582882.

[6] Wei Cai, Zehua Wang, Jason B. Ernst, Zhen Hong, Chen Feng and Victor C. M. Leung 2018. Decentralized applications: The blockchain-empowered software system. IEEE access, 6, 53019-53033.

[7] Wang Z, Jin H, Dai W, *et al.* 2021. Ethereum smart contract security research: survey and future research opportunities. Front. Comput. Sci. 15, 152802. https://doi.org/10.1007/s11704-020-9284-9.

[8] Retrieved May 04, 2023 from https://ethereum.org/zh/developers/docs/standards/tokens/erc-20/.

[9] Retrieved Jan 21, 2023 from https://ethereum.org/zh/developers/docs/standards/tokens/erc-721/.

[10] D. Reijsbergen, S. Sridhar, B. Monnot, S. Leonardos, S. Skoulakis and G. Piliouras. 2021. Transaction Fees on a Honeymoon: Ethereum's EIP-1559 One Month Later. 2021 IEEE International Conference on Blockchain (Blockchain). Melbourne, Australia, 196-204. DOI: 10.1109/Blockchain53845.2021.00034.

[11] Tim Roughgarden. 2020. Retrieved 1 Dec, 2020 from https://arxiv.org/abs/2012.00854.

[12] Stefanos Leonardos, Barnabé Monnot, Daniël Reijsbergen, Efstratios Skoulakis, and Georgios Piliouras. 2021. Dynamical analysis of the EIP-1559 Ethereum fee market. In Proceedings of the 3rd ACM Conference on Advances in Financial Technologies (AFT '21). Association for Computing Machinery, New York, NY, USA, 114–126. https://doi.org/10.1145/3479722.3480993.

[13] Ian C. Moore, Jagdeep Sidhu. 2020. Retrieved 7 May, 2021 from https://arxiv.org/abs/2105.03521.

[14] Sarah Azouvi, Guy Goren, Lioba Heimbach, Alexander Hicks. 2023. Retrieved 8 Aug, 2023 from https://arxiv.org/abs/2304.11478.

[15] H. S. Kallurkar and B. R. Chandavarkar. 2023. Transaction fee forecasting in post EIP-1559 Ethereum using 1-D Convolutional Neural Network. 2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC). Jalandhar, India, 456-462. doi: 10.1109/ICSCCC58608.2023.10176712.

[16] R. Nourmohammadi and K. Zhang. 2022. Modeling the Fork Probability of Blockchains: Did EIP-1559 Improve Ethereum? 2022 Fourth International Conference on Blockchain Computing and Applications (BCCA). San Antonio, TX, USA, 33-40. DOI: 10.1109/BCCA55292.2022.9922213.

[17] Gontara S, Boufaied A, Korbaa O. 2023. Impact of EIP-1559 on Transactions in the Ethereum Blockchain and Its Rollups. International Conference on Risks and Security of Internet and Systems. Cham: Springer Nature Switzerland, 114-126. https://doi.org/10.1007/978-3-031-31108-6_9.

[18] R. B. Fekih, M. Lahami, M. Jmaiel, A. Ben Ali and P. Genestier. 2022. Towards Model checking approach for Smart contract validation in the EIP-1559 Ethereum. 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC). Los Alamitos, CA, 83-88, doi: 10.1109/COMPSAC54236.2022.00020.

[19] S. Casale-Brunet, P. Ribeca, P. Doyle and M. Mattavelli. 2021. Networks of Ethereum Non-Fungible Tokens: A graph-based analysis of the ERC-721 ecosystem. 2021 IEEE International Conference on Blockchain (Blockchain). Melbourne, Australia, 188-195. doi: 10.1109/Blockchain53845.2021.00033.

[20] Kaili Wang, Qinchen Wang, Dan Boneh. 2022. Retrieved 10 Oct, 2022 from https://arxiv.org/abs/2208.00543

[21] D. Pirker, T. Fischer, H. Witschnig and C. Steger. 2021. A Blockchain-based Shared Mobility Platform for Private and Commercial Vehicles utilizing ERC-721 Tokens. 2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP). Zhuhai, China, 62-67. DOI: 10.1109/CSP51677.2021.9357605.

[22] Bauer, D.P. 2022. ERC-721 Nonfungible Tokens. In: Getting Started with Ethereum . Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-8045-4_5.

[23] Cuffe, P. 2018. The role of the erc-20 token standard in a financial revolution: the case of initial coin offerings. IEC-IEEE-KATS Academic Challenge, Busan, Korea (October 2018), 22-23.

[24] P. Christodoulou and K. Christodoulou. 2020. A Decentralized Voting Mechanism: Engaging ERC-20 token holders in decision-making. 2020 Seventh International Conference on Software Defined Systems (SDS), Paris, France, 160-164. DOI: 10.1109/SDS49854.2020.9143877.

[25] Nikolay Ivanov, Qiben Yan. 2022. Retrieved 17 Sep, 2022 from https://arxiv.org/abs/2209.08370.

[26] Heinonen H T, Semenov A, Boginski V. 2020. Collective Behavior of Price Changes of ERC-20 Tokens. Computational Data and Social Networks: 9th International Conference Dallas. TX, USA, 487-498. https://doi.org/10.1007/978-3-030-66046-8_40.

[27] Goswami, Subhasish, *et al.* 2021. TokenCheck: Towards Deep Learning Based Security Vulnerability Detection In ERC-20 Tokens. 2021 IEEE Region 10 Symposium (TENSYMP), Jeju, Korea, 1-8, doi: 10.1109/TENSYMP52854.2021.9550913.

[28] Bauer, D.P. 2022. ERC-20: Fungible Tokens. In: Getting Started with Ethereum . Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-8045-4_3.

[29] Fuchen Ma, Meng Ren, Lerong Ouyang, Yuanliang Chen, Juan Zhu, Ting Chen, Yingli Zheng, Xiao Dai, Yu Jiang, and Jiaguang Sun. 2023. Pied-Piper: Revealing the Backdoor Threats in Ethereum ERC Token Contracts. ACM Trans. Softw. Eng. Methodol. 32, 3, Article 61 (May 2023), 24 pages. https://doi.org/10.1145/3560264.

[30] Sako, K., Matsuo, S., Meier, S. 2021. Fairness in ERC Token Markets: A Case Study of CryptoKitties. In: Bernhard, M., *et al.* Financial Cryptography and Data Security. FC 2021 International Workshops. FC 2021. Lecture Notes in Computer Science(), vol 12676. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-63958-0_42.