Minimizing Security Risks and Improving System Reliability in Blockchain Applications: a Testing Method Analysis

Iryna Kyrychenko[1], Olha Shyshlo[1] and Nadiia Shanidze[2]

[1] *Kharkiv National University of Radio Eelectronics, Nauky Ave., 14, Kharkiv, 61166, Ukraine*
[2] *National Technical University "KhPI", Kyrpychova str., 2, Kharkiv, 61002, Ukraine*

Abstract

This article presents an analysis of testing methods used in the domain of blockchain applications, with a focus on their impact on system security and reliability. The main objective of the research is to enhance the testing process of blockchain systems and minimize security risks. The study demonstrates that implementing an appropriate approach to testing blockchain applications can considerably reduce the resources required to resolve bugs and increase the level of system reliability. In conclusion, the paper finds that the selection of testing methods during the planning phase has a substantial effect on the security and reliability of the blockchain system

Keywords 1

Blockchain testing, data mining, knowledge management, testing methods, test, testing, bug resolution, planning phase, security risks, smart–contract, system security, system reliability, application domain

1. Introduction

There are many systems that process and save customers' personal information. Every client would like to be sure that their personal data will be transmitted, processed, and stored confidentially, securely, and credibly. This can be achieved using blockchain technology. To solve this problem, many systems use certain encryption algorithms or technologies to ensure the integrity, reliability, and confidentiality of transmitted or received data [1]. Today, the most popular of these is blockchain technology.

A Blockchain is a decentralized, distributed, and digital ledger used to document transactions (blocks) on many computers or, in other words, nodes, so that the record cannot be changed "retroactively" without changing all subsequent blocks and without the consensus of the network [2].

This technology can be used to protect, store, and manage data in a decentralized and cryptographic format. Currently, it fully corresponds to the needs of users regarding the security of their data.

However, can we trust systems that have not been properly tested by experts? The answer is definitely no.

Blockchain was originally developed for digital currencies such as bitcoin, but later its advantages led to its use to record not only financial transactions, but also virtually anything of significant value, from healthcare to banking to retail.

Therefore, one of the most important stages of system development is testing. And systems using blockchain technology are no exception.

Unfortunately, there is a tendency that only systems directly related to payment transactions usually have a high level of test coverage. Although this technology is used in a wide variety of industries and areas.

An important step in the testing process of any system is the development of a testing strategy.

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine
EMAIL: iryna.kyrychenko@nure.ua (I. Kyrychenko); olha.shyshlo@nure.ua (O. Shyshlo); nashanidze@ukr.net (N. Shanidze)
ORCID: 0000-0002-7686-6439 (I. Kyrychenko); 0009-0003-5024-0256 (O. Shyshlo); 0000-0002- 9613-186X (N. Shanidze)

The goal of the research is to determine the impact of the testing methods on security and reliability on the blockchain-based systems.

The research objectives of this study are to consider the key categories of issues that may arise in blockchain-based systems, to examine the testing methodologies for blockchain-based systems, to evaluate the effect of the chosen testing methodologies on the security and dependability of the system, to analyze the results of the tests, and to provide recommendations for the effective use of methodologies by analyzing the test results.

2. Related works

Advantages of blockchain over non-blockchain database. The general advantages and disadvantages of blockchain technology were described in [3].

The advantages include the following characteristics:

- Immutability. Blockchain supports immutability, meaning it is impossible to erase or replace recorded data [4]. Therefore, the blockchain prevents data tampering within the network. maintains immutability, meaning that it is impossible to erase or replace already recorded data. Therefore, blockchain prevents data falsification in the network.
- Transparency. Blockchain is decentralized, which means that any participant in the network can verify their data [5]. Therefore, the public can trust this network.
- Censorship. Blockchain technology is free from censorship because it is not controlled by any side [6]. That is why no authority (including governments) can interrupt the work of the network.

Traceability. Blockchain creates an irreversible auditable trail that allows to easily track changes in the network [7].

Also, this technology has its own disadvantages, which include the following:

- Speed and performance. Blockchain is significantly slower than a traditional database because it performs more operations. First, it verifies signatures, which involves cryptographically signing transactions [8]. Blockchain also relies on a consensus mechanism to validate transactions. Some of the consensus mechanisms, such as proof-of-work, have low transaction capacity [9]. There is also redundancy when the network requires every node to play a major role in verifying and storing every transaction.
- High cost of implementation. Blockchain is more expensive compared to a traditional database. In addition, businesses need appropriate planning and execution to integrate blockchain into their business process [10].
- Data modification. Blockchain technology does not allow for easy modification of data after it has been recorded, but requires rewriting the codes in all blocks, which is time-consuming and expensive. The disadvantage of this feature is that it is difficult to correct an error or make any necessary adjustments.

Blockchain security. Blockchain technology and the mechanism of sharing and opening were studied in [11]. They also discussed the management of data based on blockchain technology to ensure that the data is not copied, not stored by a third party and can be safely traded. But their solution is to discuss blockchain technology for data trading without technical implementation content.

It was proposed a solution and framework for a data trading mode based on a smart contract using blockchain and machine learning [12]. This solution can be used to protect the rights and interests of the data owner. In addition, this article covers various mechanisms and algorithms for selling and downloading data, authenticating and authorizing the data owner/data buyer to download data off-chain, resolving disputes during data trading, automatically paying for the completion of trades, and setting fines for dishonest behavior. The data resource will be uploaded via a secure SSL session [13]. It has a high level of confidentiality and can prevent replay attacks and man-in-the-middle (MITM) attacks [14–15]. Therefore, security of this approach can be called reliable.

3. Methods and techniques proposed to resolve the problem in question

First, it is necessary to consider the main bugs that occur in blockchain-based systems.

Categorizing bugs can help to understand the weaknesses of blockchain systems. Therefore, knowing the most vulnerable parts of the system, more efforts can be made to resolve the dominant category of problems.

Most of the bugs are labeled as semantic. Its percentage is 67.23%. Programmers could easily introduce semantic bugs due to the lack of a thorough understanding of the system. Additionally, since semantic bugs are application specific, it is hard to automatically detect semantic bugs [16].

Environment and configuration bugs are the second most frequent by occurrence – 11.42%. This is because blockchain systems are often used by end users in various environments. These environments can run on different hardware and operating systems with different versions of installed libraries. Even if the same versions of libraries are installed, end users may have individual configurations. Errors in libraries, mismatched library versions, and incorrect configurations can lead to environment and configuration errors in blockchain systems.

Security bugs correspond to vulnerabilities that allow attackers to reduce the information security of the system for their own purposes. Even though the average number of bugs related to system security is only 1.90%, the cost of these bugs is extremely high.

As a rule, security bugs are common to all systems using blockchain technology. For example, those related to the buffer overflow vulnerability.

It is also worth noting that fixing security bugs takes the longest average time. It makes sense to implement and use automatic testing tools to detect bugs of this nature in blockchain systems.

In addition, application-specific security bugs, such as synchronization attacks, application-level denial of service, and lack of security checks, are also common in blockchain systems.

The likely cause of such bugs may be that little attention is paid to these vulnerabilities. It could also be that an ineffective approach to system testing was chosen at the planning stage of the testing process.

Traditional testing includes the following types: functional testing, non-functional testing, performance testing, security testing and integration testing.

Functional testing is one of the most basic and fundamental testing methods for any system. Adapting this method to systems based on blockchain technology, the following checks can be identified within this method [17]:

1. Checking the block and chain size. This check should focus on what transaction data should be selected, what encryption methods should be used to connect these blocks, and similar complex scenarios.
2. Checking the ability to transfer data. This test should cover both the successful transmission of data and data loss during transmission between blocks. Data loss should also be tested, as the underlying blockchain architecture is based on transactions and data security.
3. Checking the ability to add blocks. Blocks that are added to the chain should be carefully evaluated, because they cannot be changed after adding them to the chain.
4. Checking the smart contracts. Ensuring that the parties that are involved in transactions adhere to the rules of the smart contract will ensure the smooth functioning of the blockchain application [18].

These are the key points that make it possible to confirm the correct functionality of a blockchain application.

Although smart contract testing is part of functional testing, QA should pay special attention to it. Because it is one of the main components of the system that gives the customer transparency, confidence, and trust in the reliability of the blockchain-based system.

The next test method that should be included in the test strategy is penetration testing [19]. Cyberattacks and fraudulent hacking attacks are extremely difficult to perform in blockchain-based systems when smart contracts are working correctly [20]. However, any mistake in the complex mathematical and programmatic rules can be a good opportunity for unauthorized users with malicious intentions.

Blockchain penetration testing allows QA engineers to create and execute test cases that simulate the behavior of cyberattacks to see if the actions of any unauthorized user in the system will be blocked. This method involves testing that simulates real-world scenarios by attempting to log in through network services, wireless networks, social engineering networks, and blockchain applications.

Security testing is also one of the most important testing methods [21]. At this stage, QA should check whether the blockchain application is fully protected from attacks, including viruses and

malware. Any outgoing transaction or data transfer cannot be stopped, so QA testers must identify potential threats through security testing.

Also, at this stage, you should make sure that the blockchain system prevents the processing of faulty or damaged data before it becomes a risk for the user.

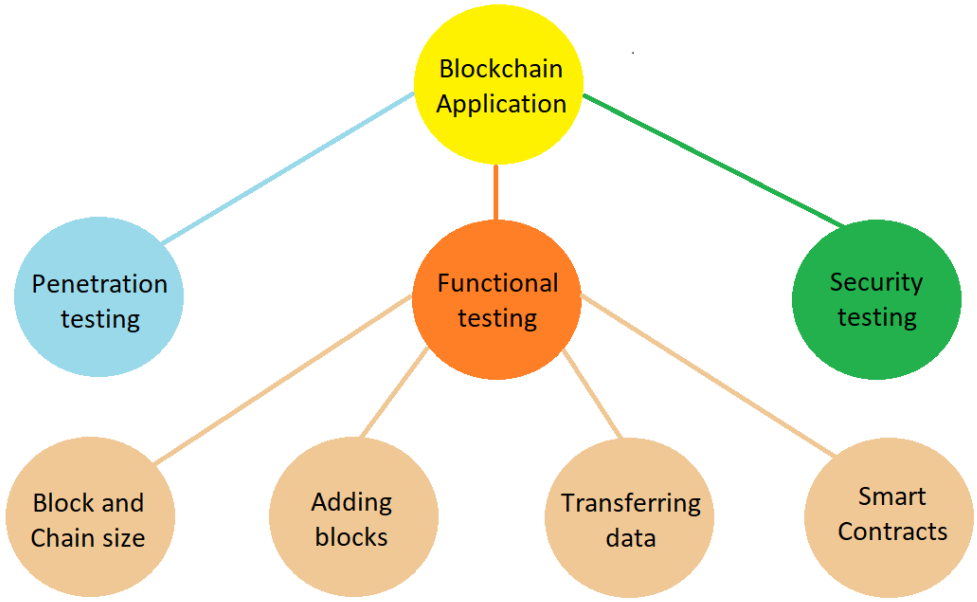

Figure 1: Testing methods that impact the security and reliability of a blockchain-based system

In order to ensure the security and reliability of a blockchain-based system, it is imperative to give particular attention to the following methods during the testing phase:

- functional testing;
- smart contract testing;
- security testing;
- penetration testing.

It is important to determine the extent to which each of these methods contributes to the overall security and reliability of the system.

4. Experiment

Each bug has its own weight, which depends on many factors. This includes:

- Criticality of the bug from a technical point of view.
- Criticality of the bug from the business logic point of view.
- Time spent on the fix.
- Complexity of the fix from a technical point of view.
- Time spent on checking the fix.
- Complexity of checking the fix.
- Potential risks after the fix.
- Potential risks if the bug was not found.

To assess each of the coefficients, it is proposed to use a scale from 1 to 5. Therefore, it makes sense to consider the dependence of the impact on system security if a bug is found and fixed at a certain stage of testing.

The testing process for each system includes the following steps:

1. Unit testing.
2. Integration testing.

3. System testing.
4. Acceptance testing.

As an experiment, the authors tested their own blockchain-based system. Below is a more detailed overview of the bugs found as a result of testing this system at each of the testing stages.

Unit testing. The following checks should be performed as part of this testing phase:

- Block size checks as part of the functional testing.
- Checks of the ability to add blocks as part of the functional testing.
- Smart-contracts checks.

Consider the following bugs that were found because of the above checks:

- Max block size is less than 1 MB. This bug was found during block size checks as part of the functional testing.
- Block is added incorrectly. This bug was found during checks of the ability to add blocks as part of the functional testing.
- Buyer data is downloaded off-chain. This bug was found during smart contracts checks.

Table 1 shows each of the parameters that will affect the weight of the bugs described above.

Table 1

Parameters that will affect the weight of the bugs founded during Unit Testing

Parameters/Bug summary	Max block size is less than 1 MB	Block is added incorrectly	Buyer data is downloaded off-chain
Criticality of the bug from a technical point of view	3	3	3
Criticality of the bug from the business logic point of view	3	5	5
Time spent on the fix	2	3	3
Complexity of the fix from a technical point of view	2	3	3
Time spent on checking the fix	1	3	3
Complexity of checking the fix	2	2	2
Potential risks after the fix	2	2	3
Potential risks if the bug was not found	2	2	2

Integration testing. The following checks should be performed as part of this testing phase:

- Chain size checks as part of the functional testing.
- Checks of the ability to transfer data as part of the functional testing.
- Smart contracts checks.

Consider the following bugs that were found because of the above checks:

- The chain size is smaller than specified. Unable to add a new block. This bug was found during chain size checks as part of the functional testing.
- Part of the data is lost during transfer. This bug was found during checks of the ability to transfer data as part of the functional testing.
- Buyer data is downloaded off-chain. This bug was found during smart contracts checks.

Table 2 shows each of the parameters that will affect the weight of the bugs described above.

Table 2

Parameters that will affect the weight of the bugs founded during Integration Testing

Parameters	Bug 1: The chain size is smaller than specified. Unable to add a new block	Bug 2: Part of the data is lost during transfer	Bug 3: Buyer data is downloaded off-chain
Criticality of the bug from a technical point of view	3	4	3

Criticality of the bug from the business logic point of view	4	5	5
Time spent on the fix	3	4	4
Complexity of the fix from a technical point of view	2	4	4
Time spent on checking the fix	2	2	3
Complexity of checking the fix	3	3	3
Potential risks after the fix	4	4	4
Potential risks if the bug was not found	3	3	3

System testing. The following checks should be performed as part of this testing phase:

- Checks of the ability to transfer data as part of the functional testing.
- Smart contracts checks.
- Checks of the system security as part of the security testing.
- Checks of the system security as part of the penetration testing.

Consider the following bugs that were found because of the above checks:

- Part of the data is lost during transfer. This bug was found during checks of the ability to transfer data as part of the functional testing.
- Buyer data is downloaded off-chain. This bug was found during smart contracts checks.
- There is a possibility to decrypt the data. This bug was found during checks of the system security as part of the security testing.
- Unauthorized user in the system is not blocked. This bug was found during checks of the system security as part of the penetration testing.

Table 3 shows each of the parameters that will affect the weight of the bugs described above.

Table 3

Parameters that will affect the weight of the bugs founded during System Testing

Parameters	Bug 1: Part of the data is lost during transfer	Bug 2: Block is added incorrectly	Bug 3: There is a possibility to decrypt the data	Bug 4: Unauthorized user in the system is not blocked
Criticality of the bug from a technical point of view	4	4	4	4
Criticality of the bug from the business logic point of view	5	5	5	5
Time spent on the fix	4	4	4	4
Complexity of the fix from a technical point of view	4	4	4	4
Time spent on checking the fix	4	4	4	4
Complexity of checking the fix	3	3	4	3
Potential risks after the fix	4	4	4	4
Potential risks if the bug was not found	4	4	4	4

Acceptance testing. The following checks should be performed as part of this testing phase:

- Checks of the ability to transfer data as part of the functional testing.
- Smart contracts checks.
- Checks of the system security as part of the security testing.
- Checks of the system security as part of the penetration testing.

Consider the following bugs that were found because of the above checks:

- Part of the data is lost during transfer. This bug was found during checks of the ability to transfer data as part of the functional testing.
- Buyer data is downloaded off-chain. This bug was found during smart contracts checks.
- There is a possibility to decrypt the data. This bug was found during checks of the system security as part of the security testing.
- Unauthorized user in the system is not blocked. This bug was found during checks of the system security as part of the penetration testing.

Table 4 shows each of the parameters that will affect the weight of the bugs described above.

Table 4
Parameters that will affect the weight of the bugs founded during System Testing

Parameters	Bug 1: Part of the data is lost during transfer	Bug 2: Block is added incorrectly	Bug 3: There is a possibility to decrypt the data	Bug 4: Unauthorized user in the system is not blocked
Criticality of the bug from a technical point of view	5	5	5	5
Criticality of the bug from the business logic point of view	5	5	5	5
Time spent on the fix	4	4	4	4
Complexity of the fix from a technical point of view	4	4	4	4
Time spent on checking the fix	4	4	4	4
Complexity of checking the fix	3	3	4	3
Potential risks after the fix	4	4	4	4
Potential risks if the bug was not found	5	5	5	5

5. Results

After analyzing the results of the experiment described in the previous section, the following conclusions can be drawn.

Correct testing of the system at the unit testing stage can significantly reduce the number and severity of risks, including those to user data security. If a critical error is not found at this stage, the cost of the error will increase depending on which of the subsequent testing stages it is found.

It is also worth noting that errors in blocks or block chains are critical in terms of potential risks. After all, according to blockchain technology, data blocks cannot be changed, and block chains are built based on previous blocks. Therefore, one mistake can lead to significant problems for the entire system, especially when it comes to data security.

As for integration testing, problems with the interaction of system components also have a significant impact on the security and reliability of the system. That is why integration testing plays a significant role in the testing process. The cost of a mistake at this stage will also be much lower than

at one of the subsequent stages. However, the potential risks after a fix are much lower than the potential risks if the bug was not found at this stage.

System testing was performed after integration testing. After analyzing the test results at this stage, we can conclude that the cost of a mistake increases significantly.

As for security testing and penetration testing, they should be conducted during system testing to ensure that the system is stable and meets the specified data security requirements.

One of the last stages is acceptance testing. At this stage, the cost and weight of a bug is the highest. After all, the greatest risks, both after the fix and if a bug that directly affects the security and reliability of the system is not found during acceptance testing.

Table 5 represents the summary data with the average weight and price of a bug found at one of the testing stages, as well as for the case when a bug related to system security was found in production.

Table 5
Average weight and price of a bug found at the testing stages

Testing stage	Bug cost	Bug weight
Unit Testing	2	3
Integration Testing	2	3
System Testing	4	4
Acceptance Testing	4	4
Production	5	5

Summarizing the results at each stage of the experiment, we can conclude that special attention should be paid to unit and integration testing. After all, it is from this point on that the cost of an error begins to grow exponentially. However, system and acceptance testing also play a significant role in the security of a blockchain-based system.

Accordingly, each of the above-mentioned testing methods will significantly affect the reliability of both individual components and the system.

It is necessary to perform functional testing at each stage of testing. Particular attention should be paid to the testing of smart contracts. The number and severity of risks can be greatly decreased by properly evaluating the system during the unit and integration testing stage. Security and penetration testing should also be performed as part of the system testing.

The study's findings can be applied to the planning phases of testing, such as when developing a test strategy and test plan.

6. Discussions

In the found articles, the authors described approaches and mechanisms to ensure the security and reliability of blockchain-based systems, but none of them described an approach to selecting testing methods to ensure the actual reliability and security of such systems.

Accordingly, each of the above-mentioned testing methods will significantly affect the reliability of both individual components and the system.

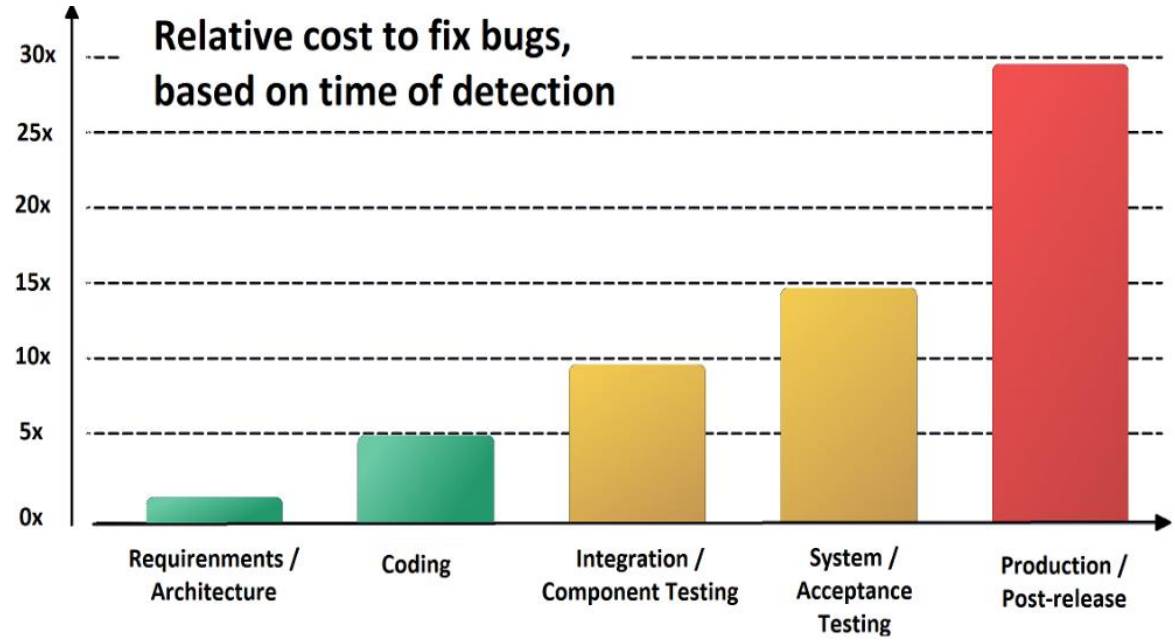

Figure 2: Relative cost to fix bugs, based on time of detection

Kent Beck wrote in his book that most defects end up costing more than it would have cost to prevent them. Defects are expensive when they occur, both the direct costs of fixing the defects and the indirect costs because of damaged relationships, lost business, and lost development time [22].

7. Interpretation of the obtained research results

In the articles reviewed, the authors described approaches and mechanisms for ensuring the security and reliability of blockchain-based systems. The testing process was discussed, but in a rather brief manner. Also, the reviewed articles did not investigate specific methods of testing blockchain-based systems. Accordingly, the impact of the selected methods in the process of testing the system on the actual security and reliability of such systems was not considered either.

In this article, the following testing methods were considered:

1. Functional testing
2. Testing of the smart contract as part of the functional testing
3. Security testing
4. Penetration testing
5. Unit testing
6. Integration testing
7. System testing
8. Acceptance testing

The impact of the quality of testing using the selected testing methods at different levels of the testing process was also considered.

Data security is one of the key requirements for blockchain-based systems. Therefore, our research was aimed at analyzing the impact of the testing process on the system's reliability [23].

Since the choice of testing methods takes place at the planning stage of the system testing process, it is necessary to clearly understand which approach should be chosen to test the system as efficiently as possible and make sure that it is safe and reliable as soon as possible [24].

The solution is designed to help you plan the testing process properly at the planning stage.

After all, the cost of a mistake can be too high, both tangibly (time and money spent on fixing the problem) and intangibly (opportunities for system development and customer confidence).

8. Conclusions

Today, blockchain technology is widely used to secure, store, and manage data due to its advantages, such as:

1. Immutability. Blockchain supports immutability, meaning it is impossible to erase or replace recorded data. Therefore, the blockchain prevents data tampering within the network. maintains immutability, meaning that it is impossible to erase or replace already recorded data. Therefore, blockchain prevents data falsification in the network.
2. Transparency. Blockchain is decentralized, which means that any participant in the network can verify their data. Therefore, the public can trust this network.
3. Censorship. Blockchain technology is free from censorship because it is not controlled by any side. That is why no authority (including governments) can interrupt the work of the network.
4. Traceability. Blockchain creates an irreversible auditable trail that allows to easily track changes in the network.

This technology also has its disadvantages. One of them is the high cost of developing and maintaining the system. The use of this technology requires significant material resources. Accordingly, there is a need to avoid wasting extra resources due to poorly conducted testing. Since the cost of any bug is quite high and tends to increase significantly depending on the stage of the system's life cycle at which it was found.

But in addition to material losses, there are also intangible losses, such as loss of customer confidence, loss of personal information transmitted and stored in the system by fraudsters.

That is why it is extremely important to make sure that the system is safe and reliable. The only way to ensure the reliability of any system is to conduct a quality testing process. That is why this article discusses the testing methods that have the greatest impact on the security and reliability of blockchain-based systems.

The obtained results of the study can be used at the planning stages of the testing process, namely, when creating a test strategy and test plan.

9. References

[1] Sharonova, N., Kyrychenko, I., Tereshchenko, G., “Application of big data methods in E-learning systems”, 2021 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), 2021. – CEUR-WS, 2021, ISSN 16130073. - Volume 2870, PP. 1302-1311.

[2] A. Hayes, Blockchain Facts: What Is It, How It Works, and How It Can Be Used, 2022. URL: https://www.investopedia.com/terms/b/blockchain.asp.

[3] Veera Budhi, Advantages And Disadvantages Of Blockchain Technology, 2022, URL: https://www.forbes.com/sites/forbestechcouncil/2022/10/20/advantages-and-disadvantages-of-blockchain-technology/

[4] Ping Zhong, Qikai Zhong, Haibo Mi, Shigeng Zhang and Yang Xiang. "Privacy-Protected Blockchain System" 2019 20th IEEE International Conference on Mobile Data Management (MDM) 10.07 (2019). doi: 10.1109/MDM.2019.000-2.

[5] Junho Jeong, Donghyo Kim, Yangsun Lee, Jin-Woo Jung and Yunsik Son, A Study of Private Donation System Based on Blockchain for Transparency and Privacy, 2020 International Conference on Electronics, Information, and Communication (ICEIC) 02.04 (2020). doi: 10.1109/ICEIC49074.2020.9051328.

[6] Jae-Seok Kim, Jin-Myeong Shin, Seok-Hwan Choi, Yoon-Ho Choi. "A Study on Prevention and Automatic Recovery of Blockchain Networks Against Persist ent Censorship Attacks" IEEE Access 13.10 (2022): 110770 – 110784.

[7] Xinting Yang, Mengqi Li, Huajing Yu, Mingting Wang, Daming Xu, Chuanheng Sun. "A Trusted Blockchain-Based Traceability System for Fruit and Vegetable Agricultural Products" IEEE Access 01.03 (2021): 36282 – 36293.

[8] Jingang Yu, Yongkang Hou, Shu Li, Zhifeng Wen, A High-Speed Data Retrieval Model on Blockchain, 2022 11th International Conference of Information and Communication Technology (ICTech)) 04.02 (2022). doi: 10.1109/ICTech55460.2022.00029.

[9] Shi Yan, Analysis on Blockchain Consensus Mechanism Based on Proof of Work and Proof of Stake, 2022 International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI) 15.08 (2022). doi: 10.1109/ICDACAI57211.2022.00098.
[10] Lina Wang, Yanhong Zhang, Ying Ren, Application of Blockchain in Life Cycle Cost Management of Weapon Equipment, 2020 2nd International Conference on Applied Machine Learning (ICAML) 17.10 (2020). doi: 10.1109/ICAML51583.2020.00079.
[11] Y. Z. Wang, J. Hou and Y. Zhang, "Data management based on block chain technology", *Electron. Des. Eng*., vol. 27, pp. 87-90, 2019.
[12] Wei Xiong, Li Xiong. "Smart Contract Based Data Trading Mode Using Blockchain and Machine Learning" IEEE Access 12.07 (2019): 102331 – 102344.
[13] Mainduddin Ahmad Jonas, Md. Shohrab Hossain, Risul Islam; Husnu S. Narman, Mohammed Atiquzzaman, An Intelligent System for Preventing SSL Stripping-based Session Hijacking Attacks,MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM) 12.11 (2019). doi: 10.1109/MILCOM47813.2019.9021026.
[14] Yunpeng Zhang, Daniel Egwede, Guohui Zhang, Xuqing Wu, Pseudonym Tracking Certificateless Aggregate Signature Scheme for Preventing Replay Attacks in a Platoon of Vehicles, 2022 International Conference on Intelligent Data Science Technologies and Applications (IDSTA) 05.09 (2022). doi: 10.1109/IDSTA55301.2022.9923034.
[15] Ming Ren, Yanhui Tian, Siqi Kong, Dali Zhou, Danping Li, An detection algorithm for ARP man-in-the-middle attack based on data packet forwarding behavior characteristics, 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC) 12.06 (2020). doi: 10.1109/ITOEC49072.2020.9141555.
[16] Zhiyuan Wan, Xin Xia, David Lo, Bug Characteristics in Blockchain Systems: A Large-Scale Empirical Study, 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR) (2017). doi: 10.1109/MSR.2017.59.
[17] Margarita Simonova, Testing Your Blockchain Website: A Step-By-Step Guide To Ensuring Security And Functionality, 2023, URL: https://www.forbes.com/sites/forbestechcouncil/2023/03/07/testing-your-blockchain-website-a-step-by-step-guide-to-ensuring-security-and-functionality/
[18] Nicolás Sánchez-Gómez, Jesus Torres-Valderrama, J. A. García-García, Javier J. Gutiérrez, M. J. Escalona. "Model-Based Software Design and Testing in Blockchain Smart Contracts: A Systematic Literature Review" IEEE Access 03.09 (2020): 164556 – 164569.
[19] Ahmad Salah Al-Ahmad, Hasan Kahtan, Fadhl Hujainah, Hamid A. Jalab. "Systematic Literature Review on Penetration Testing for Mobile Cloud Computing Applications" IEEE Access 29.11 (2019): 173524 – 173540.
[20] Nedas Matulevicius; Lucas C. Cordeiro, Systematic Literature Review on Penetration Testing for Mobile Cloud Computing Applications, 2021 XI Brazilian Symposium on Computing Systems Engineering (SBESC) 22.11 (2021). doi: 10.1109/SBESC53686.2021.9628229.
[21] William Crowe, Tae Tom Oh, Distributed Unit Security for 5G Base-Stations using Blockchain, 2020 International Conference on Software Security and Assurance (ICSSA) 28.10 (2020). doi: 10.1109/ICSSA51305.2020.00010.
[22] Kent Beck, Extreme Programming Explained: Embrace Change, 2nd ed., Addison-Wesley, volume 39, 2004.
[23] K. Smelyakov, A. Chupryna, D. Sandrkin and M. Kolisnyk, "Search by Image Engine for Big Data Warehouse," 2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 2020, pp. 1–4, doi: 10.1109/eStream50540.2020.9108782.
[24] Gruzdo, I., Kyrychenko, I., Tereshchenko, G., Shanidze, N., "Metrics applicable for evaluating software at the design stage," 2021 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), 2021. – CEUR-WS, 2021, ISSN 16130073. – Volume 2870, PP. 916–936.