УДК 004.77+004.9 DOI: 10.31673/2412-4338.2025.014506

Гашко Андрій Олександрович
Державний університет інформаційно-комунікаційних технологій, м. Київ
ORCID 0000-0001-5124-5102

Бондарчук Андрій Петрович
Державний університет інформаційно-комунікаційних технологій, м. Київ
ORCID 0000-0001-5124-5102

Трембовецький Максим Петрович
Київський національний університет ім. Т. Шевченка, м. Київ
ORCID: 0000-0002-5240-7131

Чумак Олександр Ілліч
Воєнна академія, м. Київ
ORCID: 0000-0003-3876-8149

АВТОМАТИЗОВАНИЙ МЕТОД ПЕРЕВІРКИ ПРАВИЛЬНОСТІ ВИКОНАННЯ СМАРТ-КОНТРАКТІВ В БЛОКЧЕЙН МЕРЕЖІ

Анотація: *У статті розглядається автоматизований метод перевірки правильності виконання смарт-контрактів у блокчейн-мережі Solana. Актуальність дослідження обумовлена зростанням популярності Web3 додатків та необхідністю забезпечення їх безпеки, оскільки навіть незначні помилки в коді смарт-контрактів можуть призводити до серйозних фінансових втрат. Головною метою є розробка методики автоматизованої перевірки смарт-контрактів, яка дозволяє виявляти вразливості, такі як відсутність перевірки прав засновника, помилки в арифметичних операціях та відсутність підписів чеків транзакцій. Використовуючи техніку статичного аналізу на мові програмування Rust, автори пропонують підхід, що забезпечує швидкий аналіз, менше 3 хвилин на контракт та автоматичне генерування звітів про виявлені вразливості. Методика базується на аналізі зовнішніх потоків даних через смарт-контракти, що дозволяє виявляти потенційні загрози на ранніх етапах. Для автоматизації процесу використовуються сценарії на Python та Bash, які інтегруються з хмарними сервісами, такими як Amazon Web Services, для масштабування аналізу. Результати тестування на реальних Web3 додатках демонструють ефективність методики, зокрема зменшення часу аналізу та покращення точності виявлення помилок. Важливим аспектом дослідження є постійне оновлення баз знань та інструментів аналізу, що дозволяє враховувати нові типи атак та вразливостей. Стаття також підкреслює важливість інтероперабельності між різними блокчейн-мережами, що залишається складною задачею, але є ключовим елементом для майбутнього розвитку Web3. Результати дослідження демонструють, що запропонована методика є перспективною для масштабування та адаптації до нових викликів у блокчейн-екосистемах, таких як Solana. Таким чином, розроблений підхід до автоматизованої перевірки смарт-контрактів не лише підвищує рівень безпеки Web3 додатків, але й сприяє їхньому подальшому розвитку, забезпечуючи стабільність та надійність у умовах динамічного розвитку блокчейн-технологій.*

Ключові слова: *блокчейн, смарт-контракт, Solana, інформаційна система, Rust, автоматизована перевірка, безпека, децентралізація, оптимізація.*

Hashko Andrii
State university of information and communication technologies, Kyiv
ORCID 0000-0001-5124-5102

Bondarchuk Andrii
State university of information and communication technologies, Kyiv
ORCID 0000-0001-5124-5102

© Гашко А.О., Бондарчук А.П., Трембовецький М.П., Чумак О.І. 2025

Trembovetskyi Maksym
Taras Shevchenko National University of Kyiv
ORCID: 0000-0002-5240-7131

Chumak Oleksandr
Military Academy, Kyiv
ORCID: 0000-0003-3876-8149

AUTOMATED METHOD FOR VERIFYING THE CORRECTNESS OF THE EXECUTION OF SMART CONTRACTS IN THE BLOCKCHAIN NETWORK

Abstract: *The article examines an automated method for verifying the correctness of smart contracts in the Solana blockchain network. The relevance of the research is driven by the growing popularity of Web3 applications and the need to ensure their security, as even minor errors in smart contract code can lead to significant financial losses. The primary goal is to develop an automated verification methodology for smart contracts that can detect vulnerabilities such as the absence of founder rights verification, arithmetic operation errors, and missing transaction check signatures. Using static analysis techniques in the Rust programming language, the authors propose an approach that enables rapid analysis—taking less than three minutes per contract—and automatic generation of reports on identified vulnerabilities. The methodology is based on analyzing external data flows through smart contracts, allowing for the early detection of potential threats. To automate the process, Python and Bash scripts are employed, integrating with cloud services such as Amazon Web Services to scale the analysis. Testing results on real Web3 applications demonstrate the effectiveness of the methodology, particularly in reducing analysis time and improving the accuracy of error detection. An important aspect of the research is the continuous updating of knowledge bases and analysis tools, enabling the consideration of new types of attacks and vulnerabilities. The article also highlights the importance of interoperability between different blockchain networks, which remains a challenging task but is a key element for the future development of Web3. The research results show that the proposed methodology is promising for scaling and adapting to new challenges in blockchain ecosystems such as Solana. Thus, the developed approach to automated smart contract verification not only enhances the security of Web3 applications but also contributes to their further development, ensuring stability and reliability in the dynamic evolution of blockchain technologies.*

Keywords: *blockchain, smart contract, Solana, information system, Rust, automated verification, security, decentralization, optimization.*

1. Вступ

Згідно результатів нещодавніх досліджень в галузі інформаційних технологій та комп'ютерних наук, вчені почали розкривати можливості потенційного застосування технології блокчейн у сучасному суспільстві – де головним завданням є створення та впровадження діючих методик децентралізованих, розподілених та публічних цифрових каталогів в яких всі об'єкти з'єднані у вигляді послідовних блоків які називаються транзакціями або нодами. Однією з найбільш відомих блокчейн платформ є Bitcoin, проте платформи типу Ethereum, Solan, Cardano, є блокчейн платформами нового покоління, з'явились пізніше та пропонують можливості які відсутні на блокчейн платформі Bitcoin.

Головною відмінністю Ethereum та Solana від Bitcoin є використання алгоритму досягнення консенсусу (Proof of Stack), який є основою «Смарт контрактів» у блокчейн мережі «Ethereum» та є невід'ємною складовою блокчен мережі «Solana». Даний алгоритм за своїм призначенням має фіксувати умови угод між всіма зацікавленими сторонами. Блокчейн мережа Bitcoin використовує старіший та набагато повільніший метод доказу роботи (Proof of Work).

Головним принципом «смарт контракту» є дотримання точної логіки та термінів угоди між сторонами. Методи автоматизованого виконання смарт контрактів дозволяють зменшити

або зовсім усунути посередників при заключенні угод між сторонами, на приклад таких як, класичні та централізовані фінансові установи, нотаріуси та інше.

В цій статті ми розглянемо блокчейн-платформу третього покоління «Solana» та web3 додатки що на ній створені. Блокчейн мережа “Solana” так само як і блокчейн мережа «Ethereum» використовує алгоритм доказу угоди «Proof of Stack» але “Solana” значно пришвидшила обробку транзакцій блокчейн мережі за рахунок імплементування алгоритму доказу угоди «Proof of Time». Блокчейн мережа «Solana» демонструє ріст ринкової капіталізації та популярності у користувачів в першу чергу за рахунок зростання кількісті web3 додатків які використовують мережу Solana для вирішення власних завдань. Головною відмінністю Solana від Ethereum є набагато більша швидкість проведення транзакцій та кратне зменшення їх вартості. Мовою програмування смарт-контрактів блокчейн мережі Solana є Rust, це багатоцільова мова програмування розроблена для вирішення широкого кола завдань пов’язаних з продуктивністю та безпекою.

Перехід до Web3 як нової ери інтернету забезпечить децентралізацію, контроль над даними та нові економічні можливості. Web3 базується на блокчейн-технологіях, смарт-контрактах та криптографії, що робить її перспективною. Якісні смарт-контракти у web3 застосунках матимуть вирішальний вплив на реформацію людських суспільних взаємовідносинах, чим і забезпечується великий потенціал для зростання популярності блокчен мереж у майбутньому та технології в цілому. Централізовані фінансові установи вже зараз активно впроваджують та використовують смарт-контракти у сферах запозичень, кредитування, страхування, торгівлі та звітності.

2. Аналіз літературних даних і постановка проблеми

Питання функціонування та проблематику блокчейн мереж досліджується багатьма іноземними вченими [1-4]. Виявлено, що критичним є масштабованість блокчейн мереж [5], оскільки зростання кількості транзакцій призводить до збільшення навантаження на мережу, що ускладнює швидку обробку смарт-контрактів. Іншою важливою проблемою є енергоефективність, особливо для мереж, які використовують алгоритм Proof of Work, що робить їх менш придатними для масштабного використання смарт-контрактів [6-7]. Безпека смарт-контрактів залишається ключовим аспектом [9, 13], оскільки навіть незначні помилки в коді можуть призводити до серйозних фінансових втрат [8]. Автоматизована перевірка правильності виконання смарт-контрактів стає необхідним інструментом для виявлення вразливостей, таких як недостатня перевірка прав засновника чи помилки в арифметичних операціях. Нарешті, інтероперабельність між різними блокчейн-мережами залишається складною задачею, що обмежує можливості інтеграції та взаємодії смарт-контрактів у різних екосистемах [11-12].

3. Мета і задачі дослідження

Метою дослідження є розробка методики автоматизованої перевірки смарт-контрактів на блокчейн платформі «Solana».

Для досягнення поставленої мети необхідно вирішити такі завдання:

- дослідити техніку статичного аналізу на мові програмування Rust та методики для автоматичного виявлення вразливостей коду;
- дослідити шляхи застосування методів автоматичної перевірки смарт-контрактів на різних етапах оновлення тестуємих web3 програм, для створення стабільної методики перевірки смарт-контрактів на основі інструменту статичного аналізу запущеного на наборі даних з проекту паспортизації Web3 діджитал продуктів;
- розробити нові підходи до автоматичного генерування звіту про будь-яку виявлену вразливість у процесах виконання смарт контрактів та дотримання часових вимог;
- дослідити можливість зменшення часу аналізу кожного смарт-контракту до трьох хвилин.

4.1. Контроль вразливостей смарт-контрактів за принципом техніки статичного аналізу.

В екосистемі Solana популярними є такі вразливості як – відсутність перевірки засновника, відсутність підписаних чеків транзакцій та недостатність або переповнення арифметичних операцій. Ці вразливості є добре відомі, тому є доцільним застосування методики автоматичного аналізу таких програм (смарт-контрактів) на дані вразливості використовуючи техніку статичного аналізу. Мінімальне описання цієї архітектури продемонстровано на Рис. 1

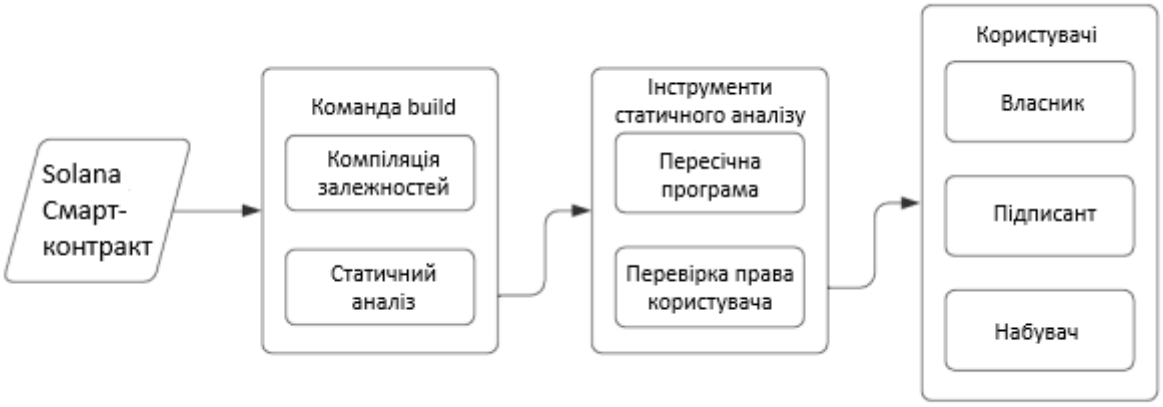

Рис. 1. Опис запуску техніки статичного аналізу.

За допомогою техніки статичного аналізу вдалось об’єднати декілька методів для автоматичного виявлення описаних вразливостей аналізуючи зовнішні потоки даних через програму (смарт-контракт) на рівень достовірності, прав засновника та підпис чеку транзакції. Техніка перевірка достовірності та прав засновника є подібними, де потік умовних операторів виконуваних інструкцій смарт-контракту можна вважати безпечним лише за умови наявності перевірки прав засновника у кожній окремій функції.

```
35        let ix = anchor_lang::solana_program::system_instruction::transfer(
36            &ctx.accounts.user.key(),
37            &ctx.accounts.campaign.key(),
38            amount
39        );
40        anchor_lang::solana_program::invoke(
41            &is_signer,
42            &[
43                ctx.accounts.user.to_account_info(),
44                ctx.accounts.campaign.to_account_info()
45            ]
46        );
```

Рис 2. Запит на перевірку прав засновника

На рис. 2 представлені умовні оператори виконуваних інструкцій смарт-контракту. Перевірку підпису чеків транзакцій виконаємо за подібною технікою, що і перевірка прав засновника, шляхом попередньо визначених тверджень в інструкціях смарт-контракту та

послідуючої перевірки всіх змінних смарт-контракту типу **AccountInfo** що використовуються разом з полем **is_signer**. Проведемо дослідження де метою буде викликати потенційну вразливість шляхом використання довільного списку назв змінних для **AccountInfo**.

```
19      pub fn withdraw(ctx: Context<Withdraw>, amount: u64) -> ProgramResult {
20          let campaign = &mut ctx.accounts.campaign;
21          let user = &mut ctx.accounts.user;
22          if campaign.admin != *user.key {
23              return Err(ProgramError::IncorrectProgramId);
24          }
25          let rent_balance = Rent::get()?.minimum_balance(campaign.to_account_info().data_len());
26          if **campaign.to_account_info().lamports.borrow() - rent_balance < amount {
27              return Err(ProgramError::InsufficientFunds);
28          }
29          **campaign.to_account_info().try_borrow_mut_lamports()? -= amount;
30          **user.to_account_info().try_borrow_mut_lamports()? += amount;
31          Ok(())
32      }
```

Рис. 3. Перевірка підпису чеків транзакцій за допомогою узгоджених тверджень

Як і у випадку перевірки прав засновника ми визначили **switchInt** на рис. 3, як одну з інструкцій для перевірки підпису чеку транзакції разом з функцією перевірки прав засновника.

Для визначення проблеми недостатності або переповнення арифметичних операцій ми аналізуємо той самий потік зовнішніх даних в рамках програми (смарт-контракту) на використання зовнішнього аргументу у будь-якій арифметичній операції. Якщо дана арифметична операція є неперевіреною то про цю проблему автоматично формується звіт.

```
89  fn ddca::add_funds (_1: anchor_lang::Context<AddFundsInputAccounts>, _2: u64) -> std::result::Result<(), anchor_lang::prelude::ProgramError> {
90
91      debug ctx => _1;
92      debug deposit_amont => _2;
93      _118 = _2;
94      _119 = <anchor_lang::Account<DdcaAccount> as DerefMut>::deref_mut(move _120) -> bb50;
95
96      bb50:{
97          ((*_119).9: u64) = Add(((*_119).9: u64), move _118);
98      }
99  }
```

Рис. 4. Представлення неперевіреної арифметичної операції додавання

4.2 Методи автоматизованої перевірки смарт контрактів

Для використання методу у продакшині ми провели дослідження методів автоматизованої перевірки смарт-контрактів. Для цього, в рамках дослідження, було розроблено сценарій запуску статичного аналізу завданням якого є автоматичний пошуку вразливостей у реальних web3 застосунках створених на блокчейн платформі Solana. Даний сценарій ми застосовували до реальних програм. Подібним сценарій ми використовували кожного дня застосовуючи його на тих самих програмах web3 застосунках для того щоб покращити його надійність у цих самих web3 застосунках.

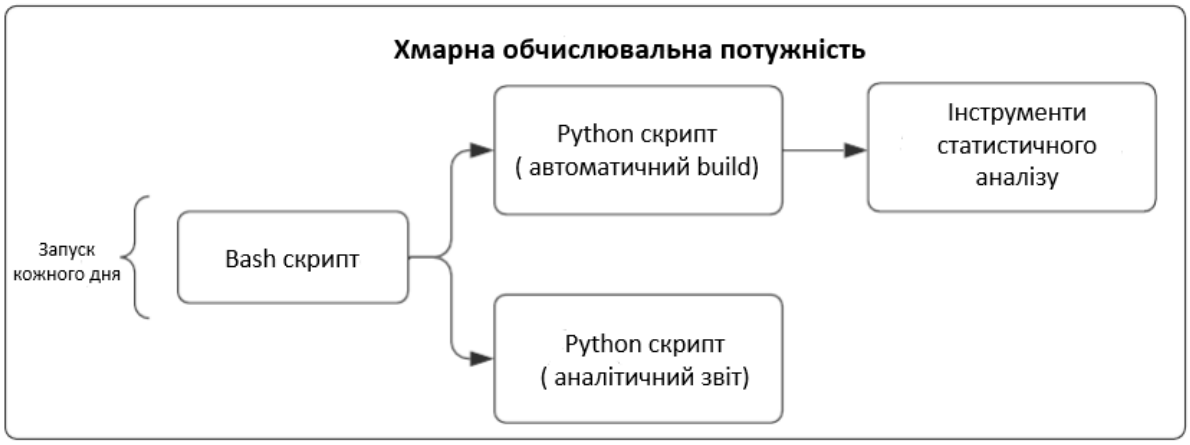

Рис. 5. Структурне зображення схеми проведення досліду

Як показано на Рис.5 сценарій що запускався кожного дня о 23:00 є комбінацією Bash та Python скриптів для автоматичного створення GitHub комітів, білдингу та компіляції необхідних інструментів що засовуються до web3 додатків. Для запуску процесу автоматизованого тестування було використано Amazon Web Services та операційну систему Ubuntu 22 LTS. Сценарій написаний на мові програмування Python та створений для автоматизації процесу обробки вихідного каталогу даних наших web3 застосунків для подальшої її передачі на обробку інструментами статичного аналізу. У фінальній частині обробки, аналітична інформація про знайдені вразливості зберігається в електронні звіти, які в подальшому використовуються алгоритмом як база знань. Намагаючись розробити найоптимальніший алгоритм застосування методики автоматизованої перевірки смарт-контрактів було написано ще один сценарій на Python. Завданням цього сценарію є порівнювати конкретно знайдені вразливості поточного дня з цими ж вразливостями але виявленими іншими конкретними днями та часом раніше. Головним чином для аналізу використовуються звіти що створюють базу знань для конкретного сценарію. В результаті доповнивши постійну та автоматизовану роботу інструментів статичного аналізу іншими аналітичними інструментами та додатковими методами перевірки що базуються на постійно оновлюваних електронних звітів із бази знань вдалось покращити показник виклику помилки та виявленню вразливостей які не були зафіксовані раніше.

4.3. Оновлення даних при автоматизовану виявленні загроз

Оновлення даних є критично важливим для ефективного виявлення загроз у Web3 додатках, оскільки блокчейн-екосистеми постійно розвиваються. База знань, яка зберігає звіти про виявлені вразливості, регулярно поповнюється новими даними для врахування змін у типах атак. Інструменти статичного аналізу, такі як Rust-аналізатори, потребують постійного оновлення для підтримки нових вразливостей, таких як недостатність арифметичних операцій чи проблеми з перевіркою прав засновника. Автоматизація процесу оновлення за допомогою сценаріїв на Python та Bash забезпечує безперервний моніторинг та аналіз, що дозволяє швидко реагувати на нові загрози. Наприклад, щоденний запуск сценарію о 23:00 гарантує актуальність інструментів та баз даних. Використання хмарних сервісів, таких як Amazon Web Services, дозволяє масштабувати процес аналізу для великих проектів. Постійне оновлення методів аналізу дозволяє виявляти нові типи атак, які раніше не були відомі. Це особливо важливо для Web3 додатків, де вразливості можуть призводити до серйозних фінансових

втрат. Таким чином, оновлення даних є ключовим елементом для підтримки високого рівня безпеки в блокчейн-екосистемах.

5. Обговорення результатів дослідження з початкового тестування на Web3 додатках

Результати початкового тестування демонструють ефективність методики автоматизованої перевірки смарт-контрактів на платформі Solana. Вона успішно виявляє відомі вразливості, такі як відсутність перевірки прав засновника, підписів чеків транзакцій чи проблеми з арифметичними операціями. Автоматизація процесу тестування за допомогою сценаріїв на Python та Bash забезпечує швидкий аналіз, що дозволяє перевіряти кожен контракт менше ніж за 3 хвилини. База знань, яка формується на основі звітів, дозволяє порівнювати результати тестування за різні дні та виявляти нові загрози. Наприклад, порівняння результатів за тиждень допомагає виявити вразливості, які не були зафіксовані раніше. Використання інструментів статичного аналізу дозволяє аналізувати зовнішні потоки даних та їх вплив на виконання смарт-контрактів. Це особливо важливо для Web3 додатків, де вразливості можуть призводити до серйозних фінансових втрат. Методика також дозволяє покращувати надійність додатків шляхом регулярного тестування та виправлення помилок. Таким чином, вона є перспективною для масштабування та адаптації до нових викликів у блокчейн-екосистемах, таких як Solana.

6. Висновки

Результати дослідження підтвердили ефективність запропонованої методики автоматизованої перевірки смарт-контрактів у блокчейн-мережі Solana. Використання техніки статичного аналізу дозволило виявити основні вразливості, такі як недостатня перевірка прав засновника, відсутність підписаних чеків транзакцій та помилки в арифметичних операціях. Проведені експерименти показали, що застосування автоматизованих скриптів на Python та Bash у поєднанні з хмарними сервісами дозволяє зменшити час перевірки смарт-контракту до трьох хвилин, що значно підвищує швидкість аналізу без втрати точності. Важливим аспектом дослідження є постійне оновлення баз знань та інструментів аналізу, що дозволяє враховувати нові види атак і вразливостей, які з’являються в блокчейн-екосистемах. Використання хмарних сервісів, таких як Amazon Web Services, забезпечує масштабованість процесу аналізу, що дозволяє застосовувати методику до великої кількості Web3 додатків у реальному часі. Запропонована методика є перспективною для подальшого масштабування та адаптації до нових викликів у сфері блокчейн-безпеки. Вона не тільки підвищує рівень безпеки Web3 додатків, але й сприяє їхньому стабільному розвитку, забезпечуючи надійність та захист від можливих атак. Таким чином, автоматизована перевірка смарт-контрактів може стати ключовим елементом для подальшого розвитку екосистеми Solana та інших блокчейн-платформ нового покоління.

Список використаної літератури

1. Zhang, Y., Chen, Z., Sun, Y., Liu, Y., & Zhang, L. (2023, July). Blockchain network analysis: A comparative study of decentralized banks. In: Science and Information Conference. Cham: Springer Nature Switzerland, 2023. p. 1022-1042. https://arxiv.org/pdf/2212.05632

2. Tao, Bishenghui, Ivan Wang-Hei Ho, and Hong-Ning Dai. Complex network analysis of the bitcoin blockchain network. In: 2021 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2021. p. 1-5.

3. Cho, Seong-Hwan. (2018). A study on analysis of the trend of blockchain by key words network analysis. The Journal of Korea Institute of Information, Electronics, and Communication Technology, 11(5), 550-555. https://arxiv.org/pdf/2011.09318

4. Wu, J., Liu, J., Zhao, Y., & Zheng, Z. (2021). Analysis of cryptocurrency transactions from a network perspective: An overview. Journal of Network and Computer Applications, 190, 103139.

5. Scherer, Mattias. "Performance and scalability of blockchain networks and smart contracts." (2017). https://www.diva-portal.org/smash/get/diva2:1111497/fulltext01.pdf

6. Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf, H., & Capkun, S. (2016, October). On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC conference on computer and communications security (pp. 3-16). https://eprint.iacr.org/2016/555.pdf

7. Sriman, B., Ganesh Kumar, S., Shamili, P. (2021). Blockchain technology: Consensus protocol proof of work and proof of stake. In Intelligent Computing and Applications: Proceedings of ICICA 2019 (pp. 395-406). Springer Singapore. https://doi.org/10.1007/978-981-15-5566-4_34

8. R. P. George, B. L. Peterson, O. Yaros, D. L. Beam, J. M. Dibbell, and R. C. Moore, “Blockchain for business,” Journal of Investment Compliance, vol. 20, no. 1, pp. 17–21, 2019, doi: 10.1108/joic-01-2019-0001.

9. S. Hemang, “Security tokens: architecture, smart contract applications and illustrations using SAFE,” Managerial Finance, vol. ahead-of-p, no. ahead-of-print. Jan. 01, 2019, doi: 10.1108/MF-09-2018-0467.

10. D. R. E. and G. Paul, “Smart contracts: will Fintech be the catalyst for the next global financial crisis?,” Journal of Financial Regulation and Compliance, vol. ahead-of-p, no. ahead-of-print. Jan. 01, 2019, doi: 10.1108/JFRC-09-2018-0122.

11. T. Feng, X. Yu, Y. Chai, and Y. Liu, “Smart contract model for complex reality transaction,” International Journal of Crowd Science, vol. 3, no. 2, pp. 184–197, 2019, doi: 10.1108/ijcs-03-2019-0010.

12. B. Willi and M. A. I., “From digital currencies to digital finance: the case for a smart financial contract standard,” The Journal of Risk Finance, vol. 19, no. 1, pp. 76–92, Jan. 2018, doi: 10.1108/JRF-02-2017-0025.

13. Zhurakovskyi, B., Otrokh, S., Poliakov, M., Poliakov, O., Skladannyi, P. (2024). Enhancing information transmission security with stochastic codes. Classic, Quantum, and Post-Quantum Cryptography 2024, 3829, 62-69.

14. Маяраш Д. Г., Жебка, В. В., Корецька В. О., Гордієнко К. О. (2022). Цифрова трансформація діяльності оператора телекомунікацій на основі систем операційної та бізнеспідтримки. Зв’язок, (1), 10-15.

15. Yakovenko, A. (2018). Solana: A new architecture for a high performance blockchain v0. 8.13. https://coincode-live.github.io/static/whitepaper/source001/10608577.pdf