ТЕХНОЛОГІЇ БЛОКЧЕЙНУ ТА СМАРТ-КОНТРАКТИ – РОЗРОБЛЕННЯ АЛГОРИТМІВ СИСТЕМНОГО ЗАХИСТУ

Кундос Максим
кандидат технічних наук, старший викладач кафедри
інформаційних систем та обчислювальних методів,
Приватного вищого навчального закладу
«Міжнародний економіко-гуманітарний університет
імені академіка Степана Дем’янчука»

Колупаєв Борис
доктор фізико-математичних наук, професор кафедри
інформаційних систем та обчислювальних методів
Приватного вищого навчального закладу
«Міжнародний економіко-гуманітарний університет
імені академіка Степана Дем’янчука»

Надозірний Святослав
викладач кафедри інформаційних систем
та обчислювальних методів
Приватного вищого навчального закладу
«Міжнародний економіко-гуманітарний університет
імені академіка Степана Дем’янчука»
м. Рівне, Україна

У міру трансформації цифрового простору в умовах децентралізованої економіки, технології блокчейну відіграють дедалі вагомішу роль у побудові надійних та прозорих інформаційних систем. Особливе місце в цій екосистемі займають смарт-контракти, які забезпечують автоматичне виконання умов угод без необхідності у довірених третіх сторонах. Проте зростання складності децентралізованих застосунків (dApps) та атак на них викликає потребу в науковому обґрунтуванні алгоритмів їхнього системного захисту.

Розглянемо підходи до побудови безпечних смарт-контрактів із використанням алгоритмічних і системних засобів захисту [1].

Виокремимо основні уразливостей смарт-контрактів:

Порушення контролю доступу (Access Control Vulnerabilities). Найпоширеніша причина втрат у 2024–2025 роках — понад 953 млн доларів

встановлено унаслідок атак на недосконало реалізовані механізми ролей і дозволів.

Атаки типу Reentrancy (повторний виклик). Одні з найпотужніших атак, що дозволяють зловмисникам викликати контракти до завершення первинного виконання та вкрасти кошти. Відомий випадок — DAO-атака, що спричинила втрату $50 млн.

Цілісність логіки (Logic Errors / Inconsistent State Handling). Поведінкові помилки в логіці або некоректна обробка стану можуть призводити до втрат або відмов у виконанні. Виявлено випадки, коли контракти входили в неконсистентний стан і переставали функціонувати.

Переповнення або недоповнення цілочисельних типів (Integer Overflow/Underflow). До Solidity 0.8.x арифметичні помилки могли невірно обчислювати значення, що давало змогу зловмисникам змінювати баланси або логіку роботи.

Маніпуляції оракулами (Price Oracle Manipulation / Flash Loan Attacks). Атаки на контракти, що використовують дані оракулів (наприклад, курс токенів), особливо з використанням flash loans, призвели до значних фінансових втрат.

Перевірка зовнішніх викликів (Unchecked External Calls). Виклики до зовнішніх контрактів без перевірки успішності можуть залишати небезпечні вразливості [2].

Залежність від блок-таймштемпу (Timestamp Dependence). Використання block.timestamp для критичних функцій у смарт-контрактах відкрито шлях маніпуляціям з боку добувачів (~miner).

Небезпечна випадковість (Insecure Randomness). Використання передбачуваних джерел для генерації випадковості (наприклад, blockhash) можна легко експлуатувати.

Атаки на бізнес-логіку (Business Logic Errors). Помилки в реалізації бізнес-процесів (наприклад, розрахунок винагород або мінімальних порогів) можуть призводити до несправедливих переваг чи втрат ресурсів.

Соціальна інженерія у смарт-контрактах (словесні пастки / хонеурот-и). Контракти можуть приховано включати ретримерні пастки, використовувані для заманювання користувачів [3]. Проведене дослідження вказує на нові форми атак: маніпулювання адресами, homograph-атаки.

Проаналізуємо існуючі інструменти перевірки безпеки смарт-контрактів [4,5]: Slither – статичний аналізатор для Solidity використовується для виявлення вразливостей, таких як переповнення, повторне використання змінних, помилки доступу тощо; MythX - хмарна платформа для статичного та динамічного аналіз; Certora призначена для перевірки коректності логіки смарт-контракту за допомогою специфікацій; Oyente призначена для статичного аналіз байткоду, пошуку потенційних проблем, таких як DAO-атаки, нескінченні петлі.

Архітектура захищеного смарт-контракту має реалізовувати механізм: контроля доступу (Ownable, AccessControl), emergency stop (Circuit Breaker), аудиту дій користувача (event logging).

Захищений смарт-контракт — це повноцінна програмна архітектура, що враховує потенційні ризики, вразливості та вимоги до масштабованості. Використання сучасних інструментів аналізу й захисту є обов'язковим для запобігання критичним вразливостям, які можуть призвести до значних фінансових або репутаційних втрат.

Алгоритм верифікації безпеки, можна інтегрувати у CI/CD-пайплайн децентралізованого застосунку, а інструменти формальної верифікації (наприклад, Manticore) здатні виявити логічні пастки ще до публікації контракту в мережі.

Смарт-контракти є ядром децентралізованих економічних систем, але їхня безпека є критичною для довіри користувачів. Представлений системний підхід дозволяє не лише знижувати ризики кібератак, а й інтегрувати захисні алгоритми в архітектуру рішень з перших етапів проєктування [2,5]. У перспективі дослідження можуть бути спрямовані на:

− створення інтелектуальних агентів для автоматичного моніторингу поведінки контрактів;

− впровадження Zero-Knowledge Proofs (ZKP) для валідації умов без розкриття конфіденційної інформації;

− розробку фреймворків з підтримкою шаблонів безпечних контрактів;

− формалізацію ISO-стандартів для безпеки Web3-застосунків.

ЛІТЕРАТУРА

1. Kiani, R., & Sheng, V. S. (2024). Automated repair of smart contract vulnerabilities: A systematic literature review. *Electronics*, *13*(19), 3942. https://doi.org/10.3390/electronics13193942
2. Sendner, C., Petzi, L., Stang, J., & Dmitrienko, A. (2023). Vulnerability scanners for Ethereum smart contracts: A large-scale study. *arXiv preprint*.
3. HajiHosseinKhani, S., et al. (2025). Unveiling smart contract vulnerabilities: Toward profiling detection methods. *Journal of Systems Architecture*.
4. Li, Lantian; Chen, Yuyu; Wu, Jingwen; Pan, Yue; & Yu, Zhongxing. (2025). Understanding inconsistent state update vulnerabilities in smart contracts. *arXiv preprint.*
5. Khodadadi, M., & Tahmoresnezhad, J. (2023). HyMo: Vulnerability detection in smart contracts using a novel multi-modal hybrid model. *arXiv preprint.*