УДК 004:338
Черніков М.М., Куваєва В.І., Болтьонков В.О., к.т.н.
Одеський національний політехнічний університет

СИСТЕМА РОЗПОДІЛЕНОЇ КОЛЕКТИВНОЇ ЕКСПЕРТИЗИ З ЗАХИСТОМ ВІД ФАЛЬСИФІКАЦІЙ

Черніков М.М., Куваєва В.І., Болтьонков В.О. Система розподіленої колективної експертизи з захистом від фальсифікацій. Розроблено систему колективного розподіленого експертного оцінювання, захищену від фальсифікацій. Побудова системи на основі блокчейн-технологій гарантує неможливість внесення цілеспрямованих змін як в індивідуальні експертні думки, так і в результати експертизи в цілому. Система ефективна в умовах участі в експертизі необ'єктивних експертів або корумпованих організаторів експертизи. Система практично реалізована на платформі Ethereum.
Ключові слова: розподілена колективна експертиза, захист від фальсифікацій, блокчейн-технології, Ethereum.

Черников М.Н., Куваева В.И., Болтенков В.А. Система распределения коллективной экспертизы с защитой от фальсификаций. Разработана система коллективного распределенного экспертного оценивания, защищенная от фальсификаций. Построение системы на основе блокчейн-технологий гарантирует невозможность внесения целенаправленных изменений как в индивидуальные экспертные мнения, так и в результаты экспертизы в целом. Система эффективна в условиях участия в экспертизе необъективных экспертов или коррумпированных организаторов экспертизы. Система практически реализована на платформе Ethereum.
Ключевые слова: распределенная коллективная экспертиза, защита от фальсификаций, блокчейн-технологии, Ethereum.

Chernikov M.M., Kuvaieva V.I., Boltenkov V.A. System of distributed collective expertise with protection from falsifications. A system for collective distributed expert estimation, protected from falsifications, has been developed. The construction of a system based on blockchain technologies guarantees the impossibility of making purposeful changes both in individual expert opinions and in the results of the expertise as a whole. The system is effective in terms of participation in the examination of biased experts or corrupt expertise organizers. The system is practically implemented on the Ethereum platform.
Keywords: distributed collective expertise, protection against falsifications, blockchain technology, Ethereum.

Постановка проблеми. У прийнятті складних рішень важливу роль відіграють процедури колективного експертного оцінювання – експертизи. З початку XXI століття поширилася розподілена мережева експертиза, що використовує сучасні інформаційні технології та мережі зв'язку і передачі даних. Перевагами такої експертизи є оперативність, точність, репрезентативність думок експертів, можливість безпосередніх комунікацій між експертами [1,2,3].

У той же час широке поширення дистанційних експертних систем створює більше можливостей для фальсифікації результатів експертного оцінювання. Зокрема, участь в процесі експертизи корумпованих експертів або модераторів експертизи дозволяє спотворити результати колективної оцінки в інтересах певних кіл, пов'язаних з організацією оцінювання варіантів рішення. С [4,5]. Тому розробка розподілених експертних систем, захищених від фальсифікацій є актуальною практичною проблемою.

Аналіз останніх досліджень та публікацій. Більшість публікацій, присвячених захисту систем колективної експертизи від фальсифікацій, спрямовані на розробку алгоритмів, що дозволяють виявити експертів, оцінки яких істотно відрізняються від середньї експертної думки шляхом застосування спеціальних алгоритмів обробки експертних оцінок [6,7]. Іншим варіантом є застосування для експертизи процедури Коупленда, фальсифікація в якій є NP-повною задачею і вимагає великих витрат обчислювального часу [8,9]. Незважаючи на складність пропонованих алгоритмів зазначені методи все одно не гарантують повного захисту системи від фальсифікацій.

Мета. Метою роботи є побудова системи розподіленої колективної експертизи, що максимально захищена від фальсифікацій, з застосуванням блокчейн-технологій.

Виклад основного матеріалу та обґрунтування результатів дослідження. Для захисту від фальсифікацій створено систему, колективної експертизи на основі технології блокчейну [10].

Блокчейн, тобто ланцюжок блоків транзакцій – є розподілена база даних, яка підтримує перелік записів, так званих блоків, що постійно зростає. База захищена від підробки та переробки. Кожен блок містить часову мітку та посилання на попередній блок хеш дерева.

Блок транзакцій – спеціальна структура для запису групи транзакцій. Щоб транзакція вважалася достовірною («підтвердженою»), її формат і підписи повинні бути перевірені, лише після цього групу транзакцій записати в спеціальну структуру – блок. Інформацію в блоках можна швидко перевірити. Кожен блок завжди містить інформацію про попередній блок. Усі блоки

можна вибудувати в один ланцюжок, що містить інформацію про всі вчинені коли-небудь операції. Перший блок в ланцюжку – первинний блок – розглядається як окремий випадок, оскільки в нього відсутній материнський блок. Якщо використовувати систему блокчейн як місце зберігання даних, можливо забезпечити гарантований захист від зміни даних після запису у систему. Також, якщо використати якусь загальну систему та зробити колективну експертизу відкритою (але сховати справжні імена експертів за їх унікальними адресами), можливо надати доступ контролюючим особам для перевірки даних через стандартні методи блокчейну, код яких відкритий та перевірений. Для вирішення задачі обрано блокчейн-систему Ethereum [12-13]. Ethereum (укр. Етеріум) – платформа для створення практично будь-яких децентралізованих онлайн-сервісів на базі блокчейна, що працюють на базі розумних контрактів. Ethereum реалізована як єдина децентралізована віртуальна машина. Оскільки Ethereum сильно спрощує і здешевлює впровадження блокчейна, його широко використовують сьогодні багато великих корпорацій для самих різних цілей.Для того, щоб транзакція потрапила до блоку, необхідно провести процес майнінгу транзакції.Майнінг – це діяльність з підтримки розподіленої платформи і створення нових блоків з можливістю отримати винагороду в формі емітованої валюти і комісійних зборів у різних криптовалютах. Вироблені обчислення потрібні для забезпечення захисту від повторного використання одних і тих же одиниць валюти, а зв'язок майнінгу з емісією стимулює людей витрачати свої обчислювальні потужності і підтримувати роботу мереж. Але звичайний майнінг підходить для дуже великої системи і зовсім не підходить до локального блокчейну, тому що не потрібно видобувати нову валюту та використовувати великі потужності для підтримки усієї системи в цілому. Існує декілька різних способів додавання транзакцій до блоку. Звичайний майнінг має загальну назву PoW (Proof of Work)[12,13] .

Доказ виконання роботи (PoW) – це принцип захисту систем від зловживання послугами. Він заснований на необхідності виконання стороною, що робить запит (клієнтом) деякої досить складної тривалої роботи (PoW-завдання, одностороння функція), результат якої легко і швидко перевіряється стороною, що обробляє запит (сервером). Головна особливість цих схем полягає в асиметрії витрат часу – досить велика тривалість для ініціатора запиту і висока швидкість для відповіді. Подібні схеми також відомі як Client Puzzle (функція клієнтської головоломки), Computational Puzzle (обчислювальна головоломка), або CPU pricing function [12].

Метод підтвердження частки (PoS – Proof of Stake) – метод захисту в Ethereum, заснований на необхідності доказу зберігання певної кількості коштів на рахунку. При використанні цього методу алгоритм криптовалюти з більшою ймовірністю вибере для підтвердження чергового блоку в ланцюжку обліковий запис з великою кількістю коштів на рахунку. Метод використовують як альтернативу методу PoW (доказ виконання роботи), в якому більшу ймовірність підтвердження блоку має обліковий запис з великими обчислювальними потужностями.

Метод авторизованих користувачів (PoA – Proof of Activity) – метод захисту в розподілених блокчейн системах, де можливість додавання транзакцій до блоків мають певні авторизовані вузли.Для локальної системи блокчейну найкраще підходить метод PoA, тому що усі вірні транзакції будуть додані до блоків, список майнерів додається до першого блоку (genesis) та не може бути змінений без доступу до активних майнінг вузлів.

Для створення алгоритму колективної дистанційної експертизи та зберігання даних створюється смарт-контракт. Смарт-контракт – це комп'ютерний протокол, який спрощує, верифікує, або забезпечує дотримання переговорів, або виконання договору, перевіряє непотрібні пункти договору. Смарт-контракти, як правило, мають інтерфейс користувача і часто слідують логіці договірних положень. Прихильники розумних контрактів стверджують, що таким чином багато видів договірних положень може бути здійснено частково або повністю, самостійно або вдвох. Смарт-контракти спрямовані на забезпечення безпеки, яка перевершує традиційне договірне право, а також на зменшення операційних витрат. Смарт контракт додається до системи один єдиний раз та його код не може бути змінений після цього. Він зберігається на окремій адресі та усі транзакції, в яких виконуються методи смарт-контракту спрямовані до цієї адреси.

Послідовність всього процесу отримання оцінки експерта зображена на рис.1.

Спочатку дані надходять від користувача, що заповнив їх у веб-інтерфейсі, до серверу (Java backend). Дані перевіряються на правильність та можливість експерта поставити оцінку. Якщо усі дані вірні, сервер відправляє їх на зберігання до блокчейну, використовуючи методи смарт-контракту. Сам

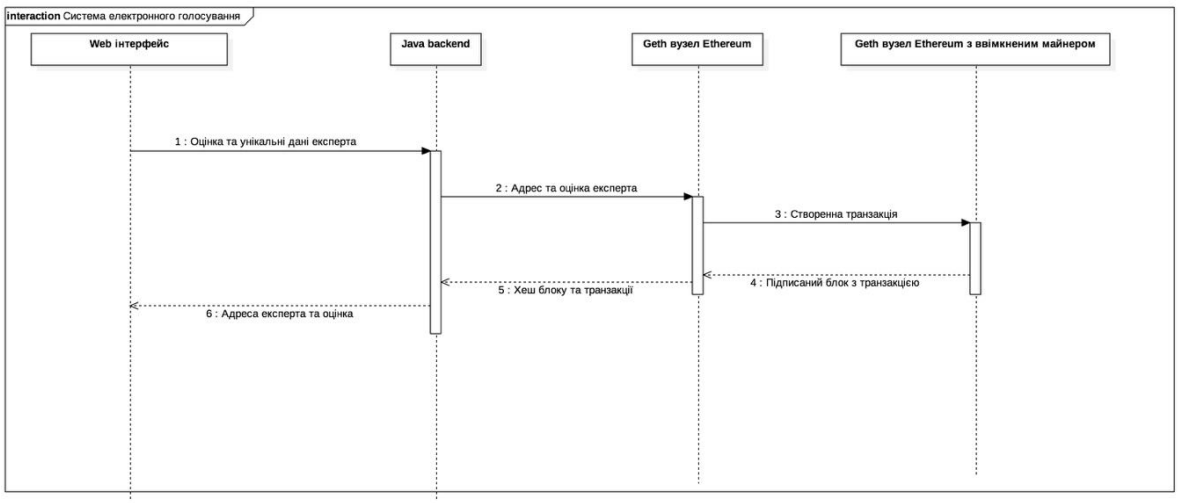

Рисунок 1 – Послідовність процесу отримання оцінки експерта

вузол geth (так називається вузол блокчейну Ethereum) перевіряє дані та створює транзакцію виклика метода смарт-контракту. Для підписки транзакції та додавання до блоку необхідно передати її до майнеру (якщо блокчейн працює на PoW), або до авторизованого вузла, що має можливість підписувати блоки (PoA). Майнером може виступати також вузол до якого надходять дані, але у більшості випадків для цього є окремі вузли. Для функціонування системи колективної експертизи необхідно розгорнути систему блокчейн. Для того, щоб запустити локальну систему блокчейн Ethereum з використанням методу PoA, необхідно створити декілька акаунтів, та додати їх адреси до genesis блоку, щоб вони мали можливість додавати транзакції до наступних блоків. Також необхідно додати початкову кількість монет, щоб мати можливість виконувати транзакції для даних адресів. Це здійснюється наступним кодом.

```
{
   "config": {
      "chainId": 1515,
      "homesteadBlock": 1,
      "eip150Block": 2,
      "eip150Hash": "0x0000000000000000000000000000000000000000000000000000000000000000",
      "eip155Block": 3,
      "eip158Block": 3,
      "byzantiumBlock": 4,
   "clique": {
         "period": 1,
         "epoch": 30000
      }
   },
   "nonce": "0x0",
   "extraData":
"0x0000000000000000000000000000000000000000000000000000000000000000008a58f09194e403d02
a1928a7bf78646cfc260b087366ef81db496edd0ea2055ca605e8686eec1e600000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000",
   "gasLimit": "0x59A5380",
   "difficulty": "0x1",
   "mixHash": "0x0000000000000000000000000000000000000000000000000000000000000000",
   "coinbase": "0x0000000000000000000000000000000000000000",
   "alloc": {
      "08a58f09194e403d02a1928a7bf78646cfc260b0": {
         "balance": "0x200000000000000000000000000000000000000000000000000000000000000000"
      },
      "87366ef81db496edd0ea2055ca605e8686eec1e6": {
```

```
        "balance": "0x200000000000000000000000000000000000000000000000000000000000000"
      },
      "F464A67CA59606f0fFE159092FF2F474d69FD675": {
        "balance": "0x200000000000000000000000000000000000000000000000000000000000000"
      }
   },
   "number": "0x0",
   "gasUsed": "0x0",
   "parentHash": "0x0000000000000000000000000000000000000000000000000000000000000000"
}
```

До цього genesis-блоку було додано інформацію про декілька адрес, що мають можливість додавати транзакції до блоків, та підписувати їх. Також цим адресам було додано початкову кількість умовних монет (одиниць ресурсу) Ether. Наприклад, адреса 08a58f09194e403d02a1928a7bf78646cfc260b0 додана до поля "extra data" для підключення в список авторизованих адресів та в список "alloc" для надання початкових монет. Поле clique вказує, що створений блокчейн працює на основі методу PoA та буде збирати блоки кожну секунду (за це відповідає параметр period). Усі інші параметри є автогенерованими для того, щоб перший блок не відрізнявся по структурі від наступних. Далі для запуску вузлів geth на основі цього genesis блоку необхідно додати цей блок до кожного з вузлів за допомогою команди geth init. Виклики методів смарт-контракту також є транзакціями, які повинні потрапляти до блоку.

Методи смарт-контракту перевіряють та захищають від повторної оцінки та несанкціонованого доступу до процедури оцінювання. Під час запуску системи потрібно мати список з усіх експертів, що беруть участь у колективній оцінці. Наприклад, якщо система використовується для оцінки діяльності парламенту, і експертами є усі громадяни України, в такому випадку їх дані потрібно занести до серверу (Java backend), тоді сервер при розгортанні контракту створить усім експертам унікальні адреси та додасть їх до списку контракту. Ось так виглядає конструктор контракту (тут і далі фрагменти коду наведені на Solidity – об'єктно-орієнтованій мові програмування контрактів для платформи Ethereum):

```
address[] public votersAddresses;
address[] public votedAddresses;
mapping(address => uint) public voterMarks;
function Voting(address[] _voters) public {
   for (uint i = 0; i < _voters.length; i++) {
      votersAddresses.push(_voters[i]);
   }
}
```

Змінна votersAddresses зберігає адреси усіх учасників експертизи. Після прийняття оцінки від якогось учасника його адреса додається в масив votedAddresses. Карта votedMarks служить для зберігання оцінки експерта напроти його адреси. Ось так виглядає метод оцінки:

```
function vote(uint mark) public ableToVote {
   voterMarks[msg.sender] = mark;
   votedAddresses.push(msg.sender);
}
```

Метод оцінки є захищеним модифікатором ableToVote. Цей модифікатор перевіряє, чи дійсно є експерт у загальному списку, та чи не була його оцінка вже прийнята. Тільки якщо обидві умови виконані, транзакція буде виконана. Це здійснюється так.

```
modifier ableToVote {
   bool flag = false;
   // Check if address is in voters list;
   for (uint i = 0; i < votersAddresses.length; i++) {
      if (msg.sender == votersAddresses[i]) flag = true;
   }
   if (flag == false) revert();
   // Check if not voted yet;
   if (votedAddresses.length != 0) {
      for (uint k = 0; k < votedAddresses.length; k++) {
```

```
        if (msg.sender == votedAddresses[k]) revert();
      }
    }
    _;
}
```

Для з'єднання серверу з вузлом geth служить RPC інтерфейс Ethereum, що має назву web3 та строгий список методів. На сервері використовується бібліотека web3j, яка є імплементацією web3 на мові програмування Java. Сервер та geth спілкуються через текстові повідомлення з використанням протоколу HTTP. Основними повідомленнями серверу, які відправляються до вузла, є створення контракту, створення користувачів з адресами, та виклики методів смарт-контракту (Наприклад, метод vote). Для прийняття рішення можливо використати будь-який з відомих методів, або навіть декілька, оскільки у контракті зберігається масив усіх оцінок. Метод обробки результатів колективної експертизи можна додати до коду контракту, або до коду серверної сторони проекту. Це може бути метод медіанних оцінок [14], або один з методів соціального вибору для отримання консенсусного колективного рішення [15]. Щоб перевірити вірність винесеного рішення можна підключитись до будь-якого вузла блокчейну в системі через стандартний клієнт Ethereum, який має назву Mist, отримати масив оцінок експертів та підрахувати результат незалежним методом, або викликати метод підрахування, якщо його реалізувано на стороні коду контракту. Результати вірного функціонування розробленої системи колективної дистанційної експертизи перевірено та підтверджено на низці тестових та практичних прикладів експертного оцінювання.

Висновки. Фальсифікація результатів колективної дистанційної експертизи, що можлива при наявності в складі експертної групи необ'єктивних або корумпованих експертів або модераторів експертизи, є серйозною практичною проблемою. Для вирішення цієї проблеми запропонована побудова системи експертизи на основі блокчейн-технології. В якості блокчейн-інструменту використано платформу Ethereum. Практична реалізація виконана з використанням мови програмування Solidity. Колективна експертиза створена як розумний контракт. Практична реалізація системи підтвердила її працездатність та захищеність від можливих фальсифікацій.

1. Gubanov D., Korgin N., Novikov D., Raikov A. E-Expertise: Modern Collective Intelligence. — Springer International Publishing, Switzerland, 2014 . — 112 p.
2. Масыч М.А., Целых А.А. Методы и технологии проведения сетевой экспертизы инновационных проектов // Современные проблемы науки и образования. — 2013. — № 6. [Електронний ресурс] / Режим доступу: http://www.science-education.ru/ru/article/view?id=11642 — Назва з екрану.
3. Newman M.E.J. The Structure of Scientific Collaboration Networks // Proc. Natl. Acad. Sci. USA. — 2001. — Vol. 98. —Pp. 404—409.
4. Zhu Y., Truszczynski M. Manipulation and Bribery When Aggregating Ranked Preferences. // In: Walsh T. Algorithmic Decision Theory. Lecture Notes in Computer Science, Vol/ 9346. — Springer: Cham, 2015. — Pp. 86-102.
5. Beliakov G., James S., Smith L., Wilkin T. Biased experts and similarity based weights in preferences aggregation.//In: EUSFLAT 2015: Proceedings of the 16th World Congress of the International-Fuzzy-Systems-Association (IFSA). — Amsterdam : Atlantis Press. — 2015. — Pp. 363-370.
6. Rezvani M., Allahbakhsh M., Vigentini L., Ignjatovic A., Jha S. An Iterative Algorithm for Reputation Aggregation in Multi-dimensional and Multinomial Rating Systems. // In: Federrath H., Gollmann D. (Eds.) ICT Systems Security and Privacy Protection. SEC 2015. IFIP Advances in Information and Communication Technology, vol 455. Springer, Cham, 2015. — Pp. 189-203.
7. Hoffman K., Zage D., Nita-Rotaru C. A Survey of Attack and Defense Techniques for Reputation Systems // ACM Comput. Surv. — 2009. — №42(1). — Pp.1-31.
8. Davies J., Katsirelos G., Narodytska N. ,Toby Walsh T., Xia L. Complexity of Algorithms for the Manipulation of Borda, Nanson's and Baldwin's Voting Rules // Artificial Intelligence. — 2014. — Vol. 217. —— P. 20-42.
9. Faliszewski P., Hemaspaandra T., Schnoor H. Manipulation of Copeland Elections // In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1. — Torronto, 2010. — Pp.367-374.
10. Bambara J. J., Allen P. R. Blockchain. A practical guide to developing business, law, and technology solutions. — McGraw-Hill Education, 2018 — 302 p.
11. Prusty N. Building Blockchain Projects. Develop real-time practical DApps using Ethereum and JavaScript. — Birmingham — Mumbai: Packt Publ., 2017. — 245 p.
12. Bashir I. Mastering Blockchain. — Birmingham: Packt Publ., 2017 — 540 p.
13. Болтенков В.А., Куваева В.И., Позняк А.В. Анализ медианных методов консенсусного агрегирования ранговых предпочтений // Інформатика та математичні методи в моделюванні. — 2017. — Том 7, №4. — С. 307-317.
14. Болтенков В.А., Куваева В.И., Червоненко П.П. Применение методов социального выбора в задачах агрегирования оценок в ранговых шкалах // Системные технологии. — 2018. — Вып. 2 (115). — С.93 — 102.