УДК 004.056.5:004.451.622
DOI: 10.31673/2409-7292.2025.032280

Побережник В.О., Опірський І.Р.

ДОСЛІДЖЕННЯ ШВИДКОДІЇ МЕТОДУ ПЕРЕДАЧІ ПОВІДОМЛЕННЬ ІЗ ЗАСТОСУВАННЯМ OFF-CHAIN ОБРОБКИ ДАНИХ ДЛЯ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ТА НЕЗМІННОСТІ ДАНИХ

Статтю присвячено емпіричному дослідженню метода передачі повідомлень, який застосовує IPFS як транзитну точку для передачі даних, блокчейн та смарт контракт як вмістилище доказів цілісності та незмінності і каналу передачі на основі протоколу WebSocket, який застосовується для підтримки каналу зв'язку між сторонами обміну повідомленнями в якому вони обмінюється не самими даними, а посиланнями на дані. Такий підхід зумовлений тим, що застосування лише технології блокчейн для обміну повідомленнями є недоцільним через ряд обмежень, тому такий контекст потребує пошуку шляхів нівелювання цих обмежень. Згадана системи розроблена на основі смарт контрактів, які виконуються в тестовій мережі Ethereum, застосування сервісу Pinata для доступу до IPFS та локального WebSocket сервера. Окрім розробки самої системи, в статті проведено розробку метода підтвердження цілісності та незмінності даних через застосування ідентифікаторів даних, які формує IPFS та їхню обробку в блокчейн шляхом застосування смарт контракту. На основі розробленого метода формується алгоритм передачі та підтвердження цілісності і незмінності даних. За допомогою розробленої та розгорнутих в локальній мережі вузла блокчейн та WebSocket сервера та зв'язку із IPFS через Pinata API проведено дослідження швидкодії системи через заміри затримок проходження повідомлення на вузлах системи та загального часу проходження, а також оптимізація алгоритму задля пришвидшення роботи системи. Одним із позитивних результатів є вдале емпіричне підтвердження можливості застосування такого підходу. Також проведено аналіз слабких та сильних сторін метода, сформульовано висновок про можливість застосування запропонованого методу в системах обміну повідомленнями, вказано напрямки подальших досліджень.

Ключові слова: blockchain, IPFS, WebSocket, децентралізація, цілісність, незмінність, смарт контракт, Ethereum.

Вступ

Наш час можна охарактеризувати швидким ростом об'ємів даних, які щодня обробляються різними сервісами, компаніями, користувачами тощо. Традиційним і панівним на цей момент є централізований підхід до зберігання, обробки чи передачі даних. Прикладами такого підходу є державні реєстри, медичні реєстри, фінансові установи, системи обміну повідомленнями тощо. Однак, попри свою розповсюдженість такі системи можна охарактеризувати низкою спільних недоліків. До них можна віднести: наявність центральної точки вразливості, яка ставить під загрозу доступність даних чи їхню цілісність у випадку успішних кібератак [1] чи апаратних або програмних збоїв [2], залежність від одного постачальника послуг [3], цензурування даних, складність доступу чи контроль над даними з боку однієї організації [4]. Проявлення цих проблем можна особливо гостро побачити у сферах, які чутливі до таких вразливостей, наприклад месенджерах [5], коли можливість обміну інформацією і приватність користувачів залежить від добропорядності сторони, яка надає такі послуги.

У відповідь на обмеження централізованої архітектури активно розвивається парадигма децентралізації, основною ідеєю якої є розподіл відповідальності за роботу системи чи мережі покладається на самих учасників такої системи чи мережі. Такий підхід дозволяє вилучити із процесу привілейованих учасників, які мають владу над всією системою, а натомість роблять всіх учасників системи рівноправними. Такий підхід дозволяє прибрати із системи центральну точку вразливості, можливість контролю над системою кимось одним чи групою привілейованих осіб.

Прикладом використання парадигми децентралізації є технологія розподіленого реєстру, і найбільш відомим представником технології є блокчейн. Він являє собою децентралізований, розподілений, хронологічно розподілений та криптографічно пов'язаний ланцюг із цифрових записів — блоків даних про транзакції. Його фундаментальними властивостями є незмінність, прозорість та авдитованість даних, що дозволяє забезпечити високий рівень довіри, оскільки будь-хто може перевірити збережені дані, і ніхто не може їх несанкціоновано змінити. Однак,

пропри свої переваги, технологія має недоліки, які можуть стати на шляху її широкого впровадження. До цих недоліків можна віднести складність масштабування, високу вартість запису даних в блокчейн, особливо великих, і низьку пропускну здатність [6].

Для вирішення цих проблем існує ряд способів, які дозволяють винести збереження та обробку даних поза межі блокчейн. Одним із них є застосування технології IPFS — InterPlanetary File System. Ця технологія є p2p протоколом, призначеним для створення розподіленої мережі розповсюдження та зберігання даних. Особливість цієї технології є вмістова адресація даних, яка на противагу локаційній адресації, яка застосовується, наприклад, в протоколі HTTP, опирається не на існування посилання на дані, а на існування самих даних, з яких формується ідентифікатор даних[7]. Такий підхід дозволяє вирішити проблему дуплікації даних, оскільки ідентифікатор даних формується із вмісту, а однаковий вміст, через застосування гешування, буде завжди давати один і той самий ідентифікатор. Однак ця технологія також має ряд недоліків, наприклад, система не гарантує довготривале збереження даних, оскільки вони зберігають доти, доки хоча б один вузол у них зацікавлений, не надає можливості відстеження змін даних чи фактичної їхньої зміни через генерацію нового ідентифікатора, що спричиняє до існування двох не пов’язаних між собою одиниць даних, які можуть мати мінімальні зміни, а також IPFS не надає можливостей контролю доступу чи підтвердження автентичності даних. Однак це не заважає розглядати саму технологію як проміжну ланку у передачі повідомлень між двома сторонами, оскільки така система не потребувала б відносно довгого зберігання даних, оскільки IPFS в цьому контексті відіграє лише транзитну точку.

Звідси виникає науково-технічна проблема комбінації властивостей блокчейну, які дозволяють зберігати дані у незмінному, цілісному та, основне, достовірному вигляді з властивостями IPFS, які дозволяють зберігати та поширювати дані із застосуванням парадигми децентралізації у гібридну архітектуру, яка дозволить обмін повідомленнями через систему, основану такій архітектурі. Один із способів вирішення цієї проблеми, орієнтований на дослідження швидкодії системи, представлений у цій статті.

Аналіз літературних джерел та формулювання проблеми

Поєднання технологій блокчейн і IPFS для розробки системи розповсюдження даних, яке розглядається у дослідженні [8], хоча й не спрямоване на дослідження такої передачі в контексті обміну повідомленнями, але дозволяє зробити висновок про можливість практичної реалізації системи, яка дозволяє не лише розповсюджувати дані через систему основану на парадигмі децентралізації, але й підтверджувати цілісність даних за допомогою технології блокчейн.

Інше дослідження [9] розглядає застосування технології блокчейн у контексті інтернету транспортних засобів, який можна віднести до множини інтернету речей. Воно досліджує зокрема проблеми проведення миттєвих платежів та комунікації, наприклад, затримки у проведенні таких транзакцій із застосування смарт контрактів у мережі Ethereum. Отримані результати дозволяють розглядати застосування блокчейну і смарт контрактів у контексті застосунків реального часу через невеликі затримки у розробленій системі (0 – 3 секунди), що дозволяє розглядати її як систему у реальному часі, що є важливим аргументом для застосування згаданих технологій у контексті месенджерів.

Ще одна робота досліджує застосування блокчейну для розробки методу обміну повідомленнями [10]. В роботі розглядається застосування лише технології блокчейн для обміну повідомленнями, однак згадуються можливі обмеження в такому підході.

В одному з попередніх досліджень ми також досліджуємо способи застосування технології блокчейн для побудови сервісів обміну повідомленнями, а також вказуємо на недоліки, які будуть притаманні застосуванню лише технології блокчейн та пропонуємо різні шляхи вирішення проблеми через застосування підходу поєднання вузлів різного типу, однак такому підходу теж характерні ті самі обмеження, які накладаються технологією блокчейн

© Побережник В.О., Опірський І.Р. (2025) Дослідження швидкодії методу передачі повідомлень із застосуванням OFF-CHAIN обробки даних для забезпечення цілісності та незмінності даних. Сучасний захист інформації, 3(63), 230–242. https://doi.org/10.31673/2409-7292.2025.032280

[11]. Інші наші дослідження досліджують використання смарт контрактів у системах основаних на технології блокчейн для розмежування доступу [12] та застосування поєднання IPFS та смарт контрактів для контролю над даними [13]. Попри те, що концепції запропоновані у цих дослідженнях не стосуються контексту обміну самими повідомленнями, вони розглядають способи утворення гібридних систем, які водночас дають можливість підтверджувати автентичність даних, обробляти та передавати їх поза межами блокчейн за допомогою технології IPFS.

Концепції, які розглянуті цих дослідженнях, дозволяють сформулювати гіпотезу про можливість розробки методу, який буде використовувати гібридну систему із IPFS та блокчейн для обміну інформацією.

Мета роботи та завдання дослідження

Метою роботи є розробка методу, який дозволить передавати повідомленням між користувачами через IPFS і підтверджувати їхню незмінність та цілісність, та дослідження швидкодії цього методу. Відповідно завдання, які стоять перед цією роботою, включають розробку методу підтвердження цілісності та незмінності даних, розробка компонентів системи, які будуть забезпечувати роботу розробленого методу та дослідження швидкодії розробленої системи.

Методологія дослідження

Методологія дослідження покладається на поетапний підхід оснований на поєднанні теоретичних та емпіричних аспектів задля створення методу, який підтверджує цілісність та незмінність даних. За допомогою формальної логіки виведено сам принцип підтвердження, який потім було формалізовано у математичну формулу. За допомогою моделювання було розроблено концепцію системи, яка потім була реалізована на практиці. Опісля реалізації системи було емпірично досліджено швидкодію системи шляхом емпіричних дослідів, які полягали у замірі затримок на кожному етапі проходження даних та загального часу проходження сигналу. Ці досліди проводилися на системі розгорнутій у локальній мережі.

На основі зібраних даних було проведено аналіз результатів та їхню інтерпретацію, яка дозволила сформулювати висновок про можливість застосування пропонованого методу в контексті систем обміну повідомленнями, проаналізовано сильні та слабкі сторони, а також сформовано напрямки для подальших досліджень.

Концепція системи

Основана ідея системи полягає в тому, щоб забезпечити передачу інформації через децентралізовану систему розповсюдження даних, а також зберегти відбиток даних, який дозволить підтвердити цілісність та незмінність даних у блокчейні. Такий підхід дозволить зберігати відбитки даних у захищеному від несанкціонованих змін чи втрати стані.

Головна проблема застосування лише блокчейну для передачі даних полягає у тому, що він не пристосований для такого застосування, оскільки великі об’єми даних негативно впливають на його продуктивність загалом. Звідси й з’являється потреба для винесення обробки даних поза межі блокчейну, оскільки такий підхід дозволяє знизити навантаження на блокчейн та винести збереження чи обробку даних поза його межі, а в самому блокчейні зберігати лише критичні дані. У випадку із підтвердженням достовірності даних, саме відбитки даних є критичними і підлягають для збереження у блокчейн. Отже, збережений відбиток даних у блокчейні можна вважати критерієм достовірності даних, а тоді обрахунок відбитку отриманих даних і перевірка його збіжності із збереженим у блокчейні можна вважати критерієм достовірності даних, тобто збіжність обрахованого і збереженого відбитків означають цілісність та незмінність даних, а розбіжність — порушення цілісності або незмінності і те, що даним, обрахований відбиток яких не збігається із збереженим, не можна довіряти. Це можна математично формалізувати у вигляді наступного виразу:

$$f(d) = \begin{cases} 1, & H(d) = B(d) \\ 0, & H(d) \neq B(d) \end{cases}, \quad (1)$$

© Побережник В.О., Опірський І.Р. (2025) Дослідження швидкодії методу передачі повідомлень із застосуванням OFF-CHAIN обробки даних для забезпечення цілісності та незмінності даних. Сучасний захист інформації, 3(63), 230–242. https://doi.org/10.31673/2409-7292.2025.032280

де $f(d)$ — функція достовірності даних, $H(d)$ — обрахований відбиток даних, $B(d)$ — збережений відбиток у блокчейні, d — дані.

Варто зазначити, що при застосуванні формули 1, при обчисленні відбитку даних потрібно застосовувати той самий алгоритм, що використовувався при створенні відбитку, який був збережений у блокчейні, оскільки в іншому випадку, такі відбитку не будуть збігатися і навіть достовірні дані не зможуть бути підтвердженими.

У загальному випадку для створення відбитків даних можна застосовувати алгоритми гешування, оскільки вони одночасно дозволяють створити відбиток даних і не дозволяють відновити початкові дані із такого відбитку, що є особливо важливим при збереження будь-яких даних, а особливо чутливих, у мережі блокчейн. Відповідно функція, яка застосовується для обчислення $H(d)$ і $B(d)$ може бути довільною і залежить лише від того, який алгоритм доцільно обрати в конкретній ситуації, однак слід уникати застосування тих геш-функції, які уже були визнані неефективними чи ненадійними, як, наприклад, MD5 [14] чи сімейство алгоритмів SHA-1 [15] тощо.

У контексті застосування технології IPFS для передачі даних, критерієм достовірності даних може стати CID. CID — це цифровий ідентифікатор даних, що зберігаються в мережі IPFS, та в основі якого лежить алгоритм гешування SHA-256 [16]. Попри те, що CID — не є геш-значенням даних, його можна застосувати як критерій достовірності даних, оскільки CID вказує на конкретні дані, а будь-які зміни у них призведуть до ефекту лавинних змін, внаслідок чого сам CID буде уже відрізнятися. Демонстрацію цього наведено на рисунку 1.

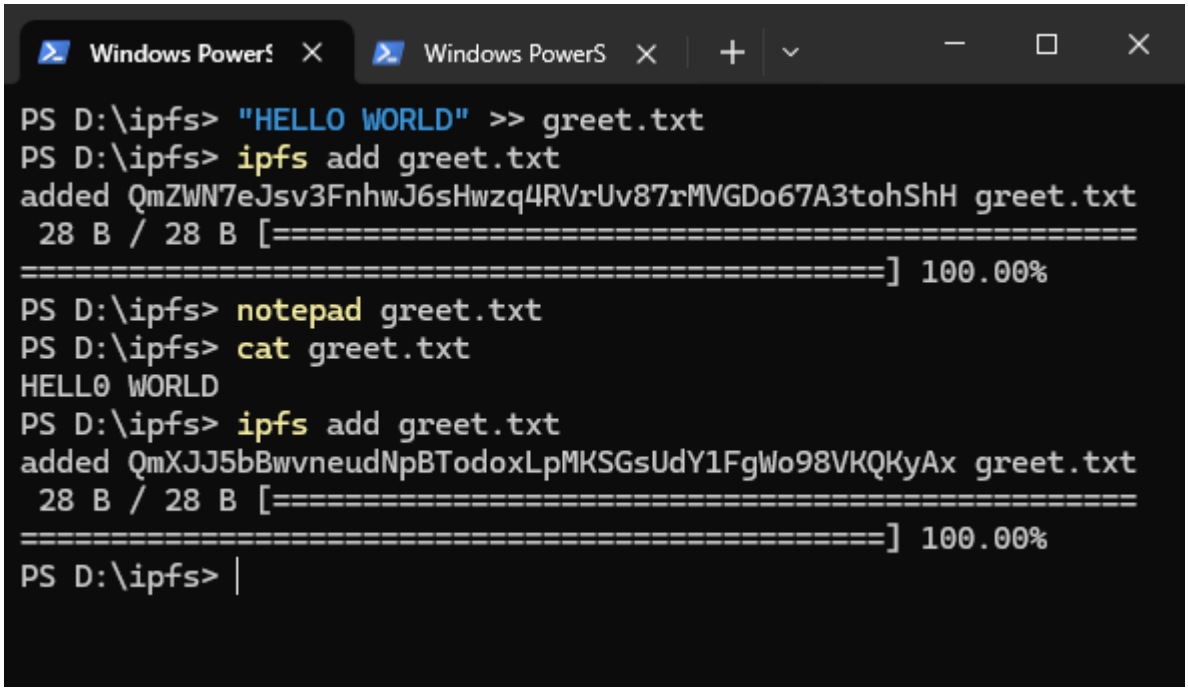

Рис. 1. Демонстрація ефекту лавинних змін у CID.

Для демонстрації цього було встановлено вузол IPFS на локальну машину та виконано наступні кроки:

1. Створено текстовий файл greet.txt із вмістом «HELLO WORLD».
2. Потім за допомогою команди ipfs add greet.txt додано файл до IPFS та отримано його ідентифікатор: QmZWN7eJsv3FnhwJ6sHwzq4RVrUv87rMVGDo67A3tohShH.
3. Потім було змінено його вміст із «HELLO WORLD» на «HELL0 WORLD», тобто літера «O» в першому слові змінена на цифру «0» задля візуальної схожості, та збережено зміни.

4. Потім greet.txt було знову додано в IPFS і отримано такий ідентифікатор: QmXJJ5bBwvneudNpBTodoxLpMKSGsUdY1FgWo98VKQKyAx.

5. При порівнянні двох ідентифікаторів можна зробити висновок, що ідентифікатори файлів значно відрізняються і дозволяють відслідкувати присутність змін, попри візуальну схожість тексту, однакову назву файлу та однаковий розмір (28 байтів).

Зважаючи на цей ефект, можна припустити, що такий ідентифікатор можна використовувати як доказ цілісності та незмінності даних. Тому, можна модифікувати формулу 1 до наступного вигляду:

$$f(d) = \begin{cases} 1, & I(d) = B(I(d)) \\ 0, & I(d) \neq B(I(d)) \end{cases}, \quad (2)$$

де $I(d)$ — переданий ідентифікатор даних збережених в IPFS, $B(I(d))$ — ідентифікатор даних, який збережений в блокчейні, d — дані. Тобто тепер перевірка зводиться не до обчислення геш-значень даних, а до перевірки ідентифікаторів, а отже збіжність переданого і збереженого в блокчейні ідентифікатора можна вважати доказом цілісності.

Оскільки умови підтвердження цілісності даних було формалізовано, можна перейти до розробки системи, яка буде засновуватися на застосуванні доказу цілісності описаного формулою 2.

Систему, яка буде використовувати запропонований критерій можна розділити на три частини: блокчейн, IPFS та канал передачі даних між користувачами. Задля простоти обміну даними було застосовано протокол WebSocket, оскільки він дозволяє обмінювати даними між сторонами в режимі реального часу, постійно підтримуючи з'єднання між ними, що спрощує роботу із таким каналом передачі, оскільки при використанні цього протоколу існує двосторонні зв'язок між клієнтом та сервером, який не потребує послідовного обміну повідомленнями між ними [17]. Тому цю властивість можна використати для обміну повідомленнями між самими користувачами. Приклад такої комунікації схематично зображено на рисунку 2, який ілюструє незалежність проходження повідомлень від сценарію запит-відповідь.

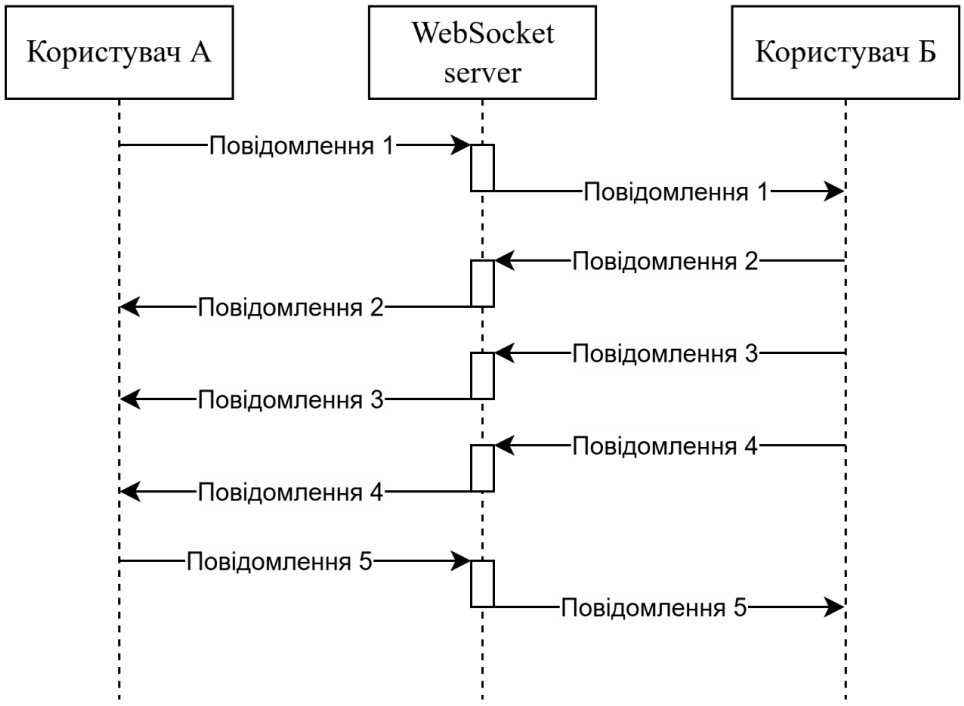

Рис. 2. Схематичний приклад комунікації між двома користувачами із застосуванням протоколу WebSocket.

Тепер, коли усі важливі компоненти системи розглянуті, можна навести концепцію системи, яка буде забезпечувати передачу стійких до несанкціонованої зміни повідомлення і яка наведена на рисунку 3.

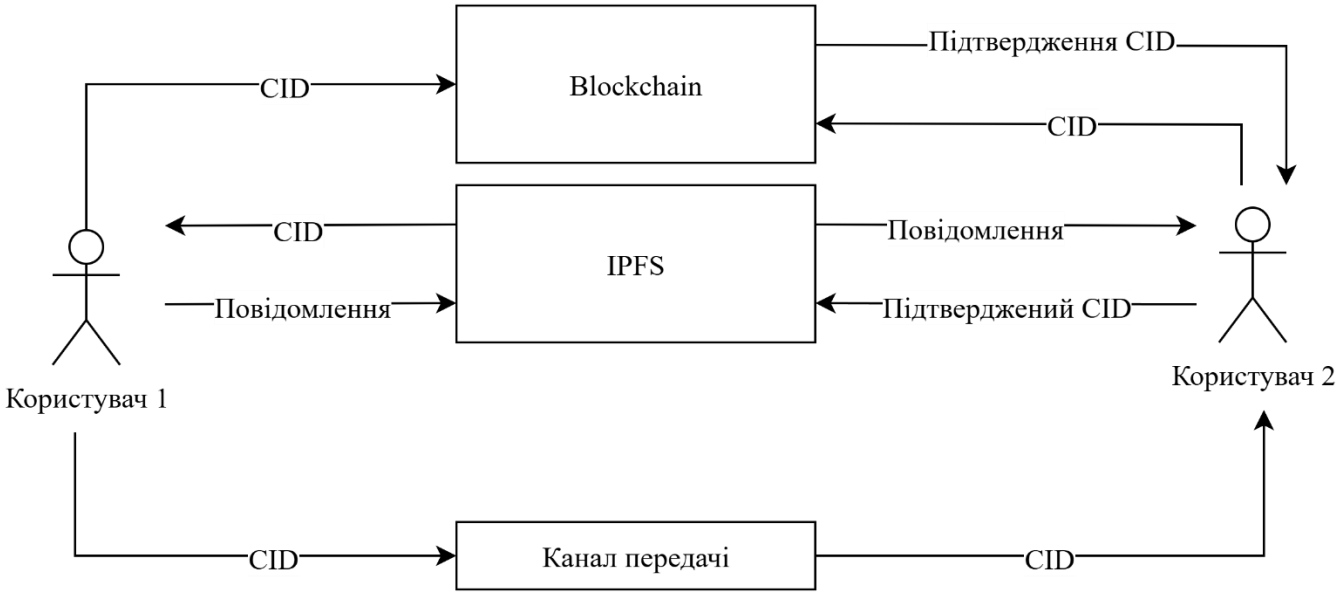

Рис. 3. Концептуальна схема системи обміну повідомленнями із застосуванням Blockchain, IPFS та каналу передачі.

Ця система об’єднує раніше згадані три компоненти, кожен з яких відіграє важливу роль в забезпеченні роботи системи. Блокчейн, а саме смарт-контракт в ньому, відповідає за збереження доказів цілісності та незмінності і їхнє підтвердження, IPFS — за передачу самих даних, канал передачі — за передачу повідомлення із ідентифікатором даних.

Також, для візуального відображення проходження повідомлення через систему на рисунку 4 зображено діаграму проходження повідомлення.

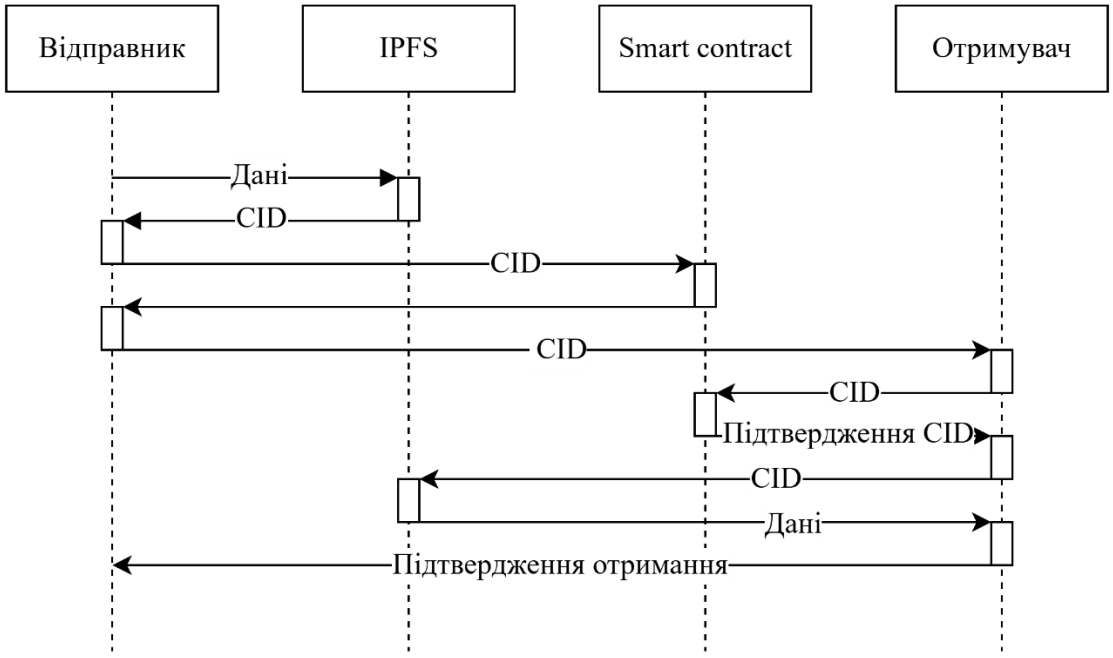

Рис. 4. Діаграма проходження повідомлення крізь запропоновану систему від відправника до отримувача.

© Побережник В.О., Опірський І.Р. (2025) Дослідження швидкодії методу передачі повідомлень із застосуванням OFF-CHAIN обробки даних для забезпечення цілісності та незмінності даних. Сучасний захист інформації, 3(63), 230–242. https://doi.org/10.31673/2409-7292.2025.032280

На діаграмі зображеній на рисунку 4 можна помітити, що самі дані в системі передаються лише двічі — при передачі до IPFS і при їхньому отриманні з IPFS, а у всіх інших випадках в системі циркулює лише ідентифікатор даних — CID.

Зважаючи на запропоновану систему (рис. 3) і діаграму проходження повідомлення у системі (рис. 4), алгоритм надсилання даних можна описати такими кроками:

1. Користувач А формує повідомлення та зберігає його в IPFS.
2. IPFS після збереження повідомлення (даних) надсилає у відповідь користувачу А ідентифікатор даних (CID).
3. Користувач А зберігає у блокчейн ідентифікатор даних шляхом збереження CID у смарт контракті.
4. Смарт контракт підтверджує збереження ідентифікатора.
5. Після підтвердження збереження ідентифікатора, користувач А формує надсилає через канал передачі ідентифікатор даних користувачу Б.
6. Користувач Б надсилає отриманий ідентифікатор у смарт контракт для перевірки.
7. Смарт контракт надсилає результат перевірки користувачу Б.
8. Якщо перевірка успішна, користувач Б завантажує дані з IPFS користуючись підтвердженим ідентифікатором даних.
9. Якщо ідентифікатор не підтверджується смарт контрактом, то ідентифікатор даних вважається зловмисним і дані з IPFS, на які він покликається, не завантажуються.
10. При необхідності через канал передачі користувач Б надсилає користувачу А підтвердження отримання даних.

Такий алгоритм застосовується в експериментальній системі, які побудовано на основі програмного пакету Hardhat, який дозволяє розгорнути локальний вузол блокчейн Etherum і виконання смарт контрактів в ньому, Pinata --- сервісу, який дозволяє взаємодіяти і мережею IPFS через API-запити та WebSocket сервера на основі Nodejs, який відповідає за утворення каналу зв’язку між сторонами обміну повідомленням, окрім цього було розроблено клієнтський додаток із застосуванням JavaScript бібліотеки ethers.js та JS-коду для зв’язку через канал обміну і взаємодії із смарт контрактом. Сам смарт-контракт був розроблений мовою програмування Solidity та розгорнутий в локальній тестовій мережі Hardhat. На рисунку 5 зображено UML-діаграму смарт контракту IntegrityStorage.sol, який відповідає за збереження ідентифікаторів.

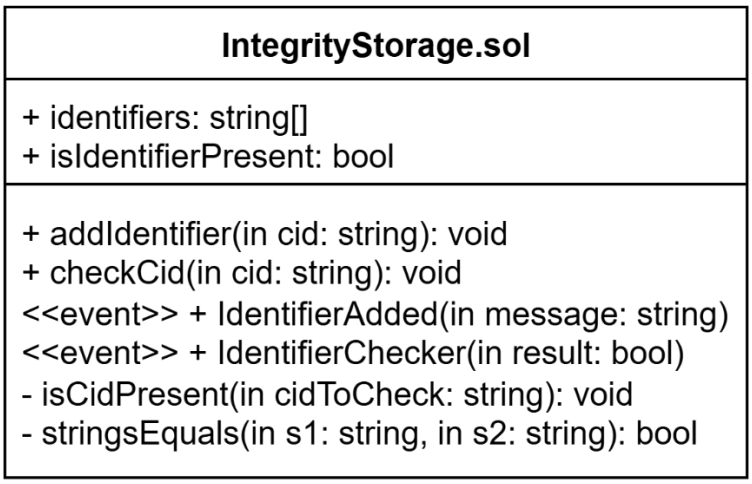

Рис. 5. UML-діаграма смарт контракту IntergrityStorage.sol

Як видно з діаграми (рис. 5) смарт контракт відповідає також за валідацію наданого CID. Перевірка на стороні смарт контракту є сильною стороною, оскільки результат виконання смарт контракту є незалежним від інших сторін, що дозволяє незалежно підтвердити CID як

© Побережник В.О., Опірський І.Р. (2025) Дослідження швидкодії методу передачі повідомлень із застосуванням OFF-CHAIN обробки даних для забезпечення цілісності та незмінності даних. Сучасний захист інформації, 3(63), 230–242. https://doi.org/10.31673/2409-7292.2025.032280

достовірні. Загалом, функціонал самого дещо ширший, що видно із рис. 5, однак основними найважливішими тут є саме функції addIdentifier та checkCid, а також події IdentifierAdded та IdentifierChecker. Функції дозволяють додати та перевірити ідентифікатор даних, а події дозволяють повідомити клієнтську частину про виконання функцій да досягання певного стану у смарт контракті. Пізніше, саме обробники цих подій на клієнтській стороні дозволять просуватися вперед за алгоритмом, згаданим раніше, наприклад, клієнтський застосунок надсилає ідентифікатор у смарт контракт, смарт контракт зберігає ідентифікатор та транслює подію IdentifierAdded та повідомлення яке в неї передав смарт контракт, а після цього обробник події IdentifierAdded на стороні клієнта обробляє її і дозволяє надіслати повідомлення із ідентифікатором адресату.

Після формування самого смарт контракту можна перейти до формування структури повідомлення, яке буде передаватися. Структуру повідомлення, яке буде зберігатися в IPFS та яке буде передаватися адресату через канал передачі зображено на рисунку 6.

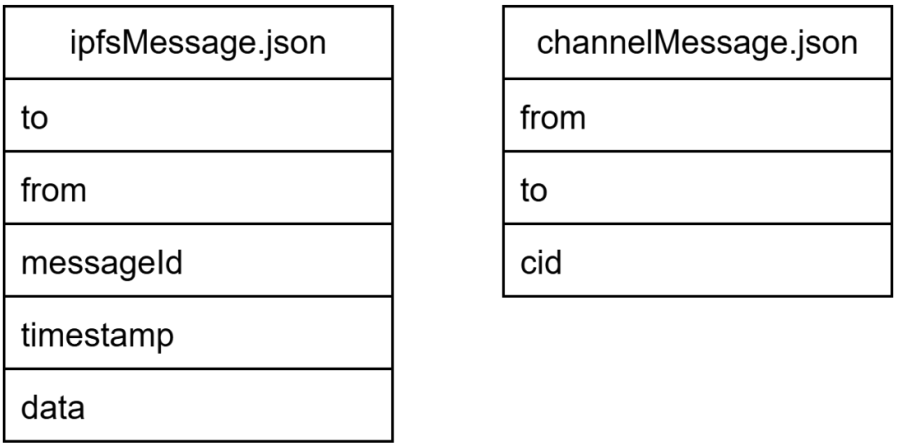

Рис. 6. Приклади структури повідомлень, які зберігаються в IPFS — ipfsMessage.json і повідомлень, які передаються через канал зв'язку — channelMessage.json

Тобто поле cid в channelMessage.json пістить ідентифікатор, який вказує на конкретний ipfsMessage.json збережений в IPFS, що дозволяє пов'язати повідомлення-обгортку із фактичним повідомленням. Для тестової системи сервіс Pinata було обрано для роботи із IPFS через можливість завантаження даних в IPFS через API, що спрощує роботу і дозволяє передавати дані одразу із клієнтського застосунку за допомогою JS-коду.

Результати

Тестову систему було розроблено та запущено на різних пристроях, вузли системи та характеристики техніки описано в таблиці 1.

Таблиця 1

Опис технічних характеристик вузлів розробленої системи

Вузол	Розміщення	ОС	Процесор	ОЗП	Примітки
WebSocket сервер	Локальна мережа	Raspbian	Broadcom BCM2711 1.50GHz	8GB	Працює на основі Nodejs
Клієнт	Локальна мережа	Windows 10	Intel i7-4770 3.40 GHz	24GB	Для розробки використано бібліотеку etherjs 6.7.15 та JavaScrip та HTML.
Вузол Etherum	Локальна мережа	Windows 10	Intel i7-4770 3.40 GHz	24GB	Вулоз розгорнуто за допомогою програмного пакету Hardhat3
IPFS	Мережа інтернет	—	—	—	—

Для перевірки роботи системи було створено наступні сценарії:

1. Передача повідомлення розміром 1 Kb.
2. Передача повідомлення розміром 10 Kb.
3. Передача повідомлення розміром 100 Kb.
4. Передача повідомлення розміром 1 Mb.

Кожен сценарій виконувався 10 разів під час чого проводилися заміри затримки в доставці повідомлення на кожному етапі та загальної. Час доставки повідомлення можна виразити такою формулою:

$$T = t_{ui} + t_{bs} + t_{ws} + t_{bc} + t_{di}, \quad (3)$$

де T — час доставки, t_{ui} — час завантаження в IPFS та отримання CID, t_{bs} — час збереження в смарт контракті, t_{ws} — час передачі через WebSocket, t_{bc} — час підтвердження CID смарт контрактом, t_{di} — час завантаження повідомлення з IPFS.

Результати вимірювань наведено в таблиці 2.

Таблиця 2

Результати сценаріїв

Сценарій	**T avg, c**	**T min, c**	**T max, c**
1 kb	9.95	7,67	11,38
10 kb	9,42	8,23	10,65
100 kb	10,05	7,72	12,27
1 Mb	10,76	8,91	13,04

Зважаючи на значні, як для сервісу обміну повідомленнями, затримки було оптимізовано алгоритм роботи системи. Оскільки смарт контракт транслює всі слухачам подію, а не лише тому, хто спричинив її появу, то процес отримання підтверджених ідентифікаторів можна оптимізувати для отримувача, якщо він буде одразу запам'ятовувати ідентифікатори, які смарт контракт транслював в ролі доданих, що дозволить вилучити із загального часу передачі повідомлення (2) затримку t_{bc}, що дозволить оптимізувати роботу систему і прискорити отримання повідомлення адресатом.

Нові результати за тими самими сценаріями з оптимізованим алгоритмом наведено у таблиці 3.

Таблиця 3

Результати сценаріїв із оптимізованим алгоритмом

Сценарій	**T сер, c**	**T min, c**	**T max, c**
1 kb	6,20	4,64	8,40
10 kb	5,61	4,14	7,92
100 kb	6.40	4.40	8,33
1 Mb	6,22	3,45	8,32

Оптимізація алгоритму доставки повідомлення дозволила отримати приріст в швидкодії системи у кожному зі сценаріїв.

Аналіз кожного етапу затримки показав значні затримки на всіх етапах окрім передачі через WebSocket, які коливалися на цьому етапі в межах 0,02–0,1с. Також слід наголосити на тому, що розмір самого повідомлення фактично не впливав на затримки t_{bs}, t_{ws} та t_{bc}, оскільки розмір даних, які передавалися завжди був однаковим, це зумовлено тим, що CID має сталий розмір, відповідно розмір даних, які передавалися у смарт контракт був завжди однаковим. З цієї самої причини і немає впливу не час проходження через WebSocket, оскільки при проходженні через нього повідомлення завжди мало однаковий розмір, що зображено на рисунку 7.

© Побережник В.О., Опірський І.Р. (2025) Дослідження швидкодії методу передачі повідомлень із застосуванням OFF-CHAIN обробки даних для забезпечення цілісності та незмінності даних. Сучасний захист інформації, 3(63), 230–242. https://doi.org/10.31673/2409-7292.2025.032280

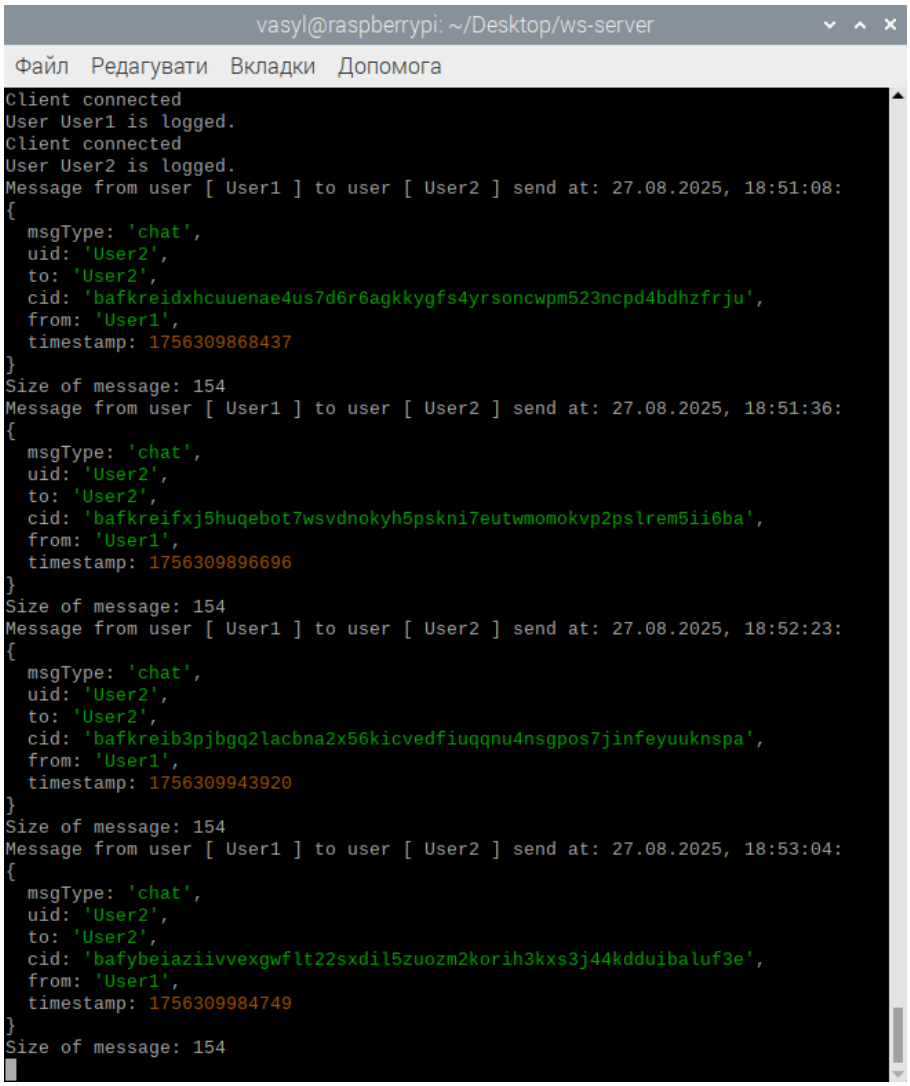

Рис. 7. Логи передачі повідомлень через WebSocket сервер із однаковим розміром для повідомлень 1kb, 10kb, 100kb та 1Mb

Також на рисунку 7 можна побачити поля, яких не було на рисунку 6. Ці поля не є обов'язковими для доставки повідомлення і додані для роботи з API розробленого WebSocket сервера і логування повідомлень з метою отримання числових значень затримки. Стандартний вигляд повідомлень може стати корисним інструментом для захисту приватності користувачів, при присутності зловмисника, який може підслуховувати канал передачі, оскільки однакова довжина повідомлення не дозволяє встановити ні тип даних, якими обмінюються, ні їхній вміст.

Також при використанні сценарію із розміром 1Mb та оптимізованим алгоритмом, було виявлено приріст в швидкодії за рахунок зменшення затримки на завантаженні повідомлень в IPFS та збереженні ідентифікатора в смарт контракті. Це слід пов'язувати із можливим зменшеним навантаженням на мережу IPFS або збільшенням кількості активних вузлів в мережі. Цей результат дозволяє зробити висновок про вплив завантаження IPFS мережі на швидкодію досліджуваного методу. Щодо затримок на збереженні у смарт контрактах, то його варто вважати випадковим, оскільки дані записуються в блокчейн після підтвердження блоку, а для цього потрібно додати новий блок в мережу, що є випадковим процесом. Можливий вплив міг би мати розмір блокчейну на цей процес, але для висновок про його вплив у нас замало даних, оскільки цей аспект не досліджувався в цій роботі, бо під час замірів

розміри блокчейну не перевищували 100 блоків в обох випадках, однак теоретично такий вплив міг би з'явитися при досягання достатньо великому розміру.

Інтерпретація результатів

Основним позитивним результатом цього дослідження стало те, що вдалося експериментально підтвердити гіпотезу застосування технології блокчейн для передачі даних через застосування комбінації технології блокчейн та IPFS. Окрім цього було емпірично визначено слабкі та сильні сторони методу, які збігаються із визначеними в попередніх дослідженнях [11], [12], [13]. Переваги та недоліки пропонованої системи наведено у таблиці 4.

Таблиця 4

Переваги та недоліки методу

Критерій	Переваги	Недоліки
Швидкодія	Розмір даних не впливає на затримки у каналі передачі та взаємодії із смарт контрактом.	Відносно довгі затримки при взаємодії із блокчейном і смарт контрактом, IPFS.
Цілісність та незмінність даних	Подвійне забезпечення підтвердження цілісності та незмінності даних через застосування IPFS ідентифікатора даних та його збереження у смарт контракті.	Оптимізований алгоритм потребує перебування адресата в мережі. Взаємодія із смарт контрактами в мережі Ethereum є платною [18].
Загальна безпека даних	Можливість розширювати функції смарт контракту для керування доступом. Можливість впровадження наскрізного шифрування.	IPFS за замовчування не використовує шифрування даних.
Інтеграція у наявні системи	Можливість інтегрувати принцип у існуючі канали передачі інформації через незалежність способу підтвердження даних від каналу передачі.	Можливі складнощі у розробці необхідних компонентів у існуючі системи. Відносна новизна запропонованих технологій, що може впливати на присутність необхідних експертів на ринку. Можливі складнощі у розробці смарт контрактів.
Незалежність компонентів	Можливість додавати в існуючі системи лише модуля пов'язаного із IPFS або лише смарт контракту.	Попри свою незалежність одне від одного, додавання лише одного компоненту негативно впливатиме на надійність методу, оскільки IPFS відповідає за децентралізовану передачу даних, хоча має умовний відбиток даних (ідентифікатор), а смарт контракт відповідає за саме підтвердження відбитку як вартого довіри.

Зважаючи на переваги методу і його відносно стабільні результати у швидкодії дозволяє розглядати його як можливий спосіб передачі даних між користувачами. Однак слід наголосити на тому, що емпірично отримані значення затримок спонукаю до пошуку шляхів оптимізації такого методу, оскільки такі затримки є відносно великими для спілкування в режимі реального часу.

З іншого боку такий метод дозволяє його застосування у системах, які не потребують відносно швидкої відповіді, наприклад, системи для обміну файлами, електронна пошта, розповсюдження відкритих даних тощо. Водночас недоліки та слабкі сторони цього методу дають можливість сформулювати можливі напрямки для подальших досліджень. В цьому контексті можна сформулювати два основних напрямки.

По-перше, це практичне дослідження в пливу IPFS-кластерів на швидкодію такого способу. Основна ідея цього напрямку — це розробка окремого кластеру IPFS для

© Побережник В.О., Опірський І.Р. (2025) Дослідження швидкодії методу передачі повідомлень із застосуванням OFF-CHAIN обробки даних для забезпечення цілісності та незмінності даних. Сучасний захист інформації, 3(63), 230–242. https://doi.org/10.31673/2409-7292.2025.032280

обслуговування передачі даних. Перевага окремого кластеру над застосуванням публічної мережі IPFS буде в тому, що цей кластер буде спрямований лише на обслуговування системи обміну повідомленнями, що має знизити загальне навантаження на IPFS модуль системи. Завданнями такого дослідження можуть стати дослідження впливу кількості вузлів на швидкодію мережі, швидкість поширення даних в такій мереж, вплив фактору реплікації даних на швидкодію, а також розробка шлюзів контролю доступу до мережі та автентифікації користувачів, для підвищення безпеки мережі та розмежування доступу.

По-друге, дослідження впливу інших типів блокчейну для передачі та збереження доказів цілісності та незмінності. Цей напрямок випливає є можливим через те, що існую інші типи блокчейну, які більш пристосовані для швидкої обробки даних. Наприклад позитивний вплив може мати впровадження Hyperladger Fabric замість Ethereum. Цей блокчейн є блокчейном приватного типу, швидкодія яких, як відомо, переважає швидкодію блокчейнів відкритого типу, як Ethereum. Втім цей ефект досягається за рахунок зниження рівня децентралізації, оскільки кількість вузлів стає обмеженою впливом самої організації, яка контролює такий блокчейн. Іншими претендентами можуть стати блокчейни ICP, BNB Chain, Solana, Taraxa тощо, оскільки вони забезпечують обробку великої кількості транзакцій на секунду, особливо якщо порівнювати з Ethereum [19].

Окремим напрямком пов’язаним із Ethereum може стати емпіричне дослідження впливу рішень Layer 2 на запропонований метод, які дозволяють оптимізувати використання блоків Ethereum через ущільнення блоку, шляхом об’єднання кількох транзакцій в одну (Rollup), створення каналів стану чи утворення дочірніх блокчейнів. Варто зазначити, що такі рішення стосуються не тільки мережі Ethereum.

Також перспективним напрямком може стати розробка методу утворення децентралізованого каналу передачі повідомлень-обгорток, які доставляють ідентифікатори даних. Цей напрямок зумовлений тим, що застосування каналу передачі таких повідомлення опирається на існування центрального WebSocket сервера, що вводить у, на перший погляд, децентралізовану систему центральну точку вразливості, що негативно впливає на стійкість всієї системи і, фактично, є найслабшою точкою.

Також варто наголосити на тому, що в цьому дослідженні не досліджується криптографічний захист даних від несанкціонованого доступу та методи керування доступом, ключами шифрування тощо, оскільки метою статті було дослідити можливість реалізації запропонованих раніше концепцій емпірично, однак нехтування згаданими властивостями не може бути застосоване до розробки реальної системи, на відміну від тестового прототипу.

Висновки

У цій статті реалізовано практичний метод реалізації концепцій запропонованих у попередніх дослідження, сформовано критерій цілісності та незмінності даних, а також вказано переваги та недоліки реалізованого методу. Окрім цього проведено дослідження та отримано емпіричні дані про затримки доставки повідомлення таким способом. На основі отриманих даних сформульовано можливі напрямки додаткових досліджень, які б могли дозволити застосування методу у системах, які потребують більш швидкого обміну даними, ніж швидкість, яка була отримана при реалізації і тестуванні запропонованого методу.

Результати дослідження дозволяють розглядати метод, як можливий для застосування у системах, які не залежать від швидкої доставки повідомлень адресату, однак потребують високого рівня довіри до інформації, яка надсилається.

Запропонований метод дозволяє підтвердити цілісність та незмінність даних, шляхом подвійної перевірки ідентифікатора даних, яка забезпечується одночасним використанням IPFS та збереженням ідентифікаторів у смарт контрактах, які існують в блокчейні. Окрім цього, застосування IPFS дозволяє передавати між сторонами комунікації не лише текстові дані, але й файли, що дає забезпечити обмін медіа файлами різного роду, що може значно розширити сферу застосування методу.

© Побережник В.О., Опірський І.Р. (2025) Дослідження швидкодії методу передачі повідомлень із застосуванням OFF-CHAIN обробки даних для забезпечення цілісності та незмінності даних. Сучасний захист інформації, 3(63), 230–242. https://doi.org/10.31673/2409-7292.2025.032280

Перелік посилань

1. У Мін'юсті заявили про масштабний збій у роботі держреєстрів. Суспільне Новини. URL: https:// suspilne.media/906711-u-minusti-zaavili-pro-masstabnij-zbij-u-roboti-derzreestriv/ (дата звернення: 31.08.2025).

2. У Приватбанку стався збій з каналами зв'язку: відбуваються затримки в роботі. Новини України - останні новини України сьогодні - УНІАН. URL: https: // www.unian.ua/economics/finance/zbiy-v-privatbanku-stavsya - zbiy - z – kanalami - zv-yazku-banku-vidbuvayutsya-zatrimki-v-roboti-novini-ukrajina-11797827.html (дата звернення: 31.08.2025).

3. Збій в роботі «Дії», сайту «Нової пошти» і терміналів. De Novo завершила розслідування аварії в дата-центрі. Спільнота програмістів | DOU. URL: https://dou.ua/lenta/news/de-novo-named-cause-of-the-accident/ (дата звернення: 31.08.2025).

4. Gebhart G., Kohno T. Internet censorship in thailand: user practices and potential threats. 2017 IEEE european symposium on security and privacy (euros&p), м. Paris, 26–28 квіт. 2017 р. 2017. URL: https://doi.org/10.1109 /eurosp.2017.50 (дата звернення: 31.08.2025).

5. Zhang L., Ji Q., Yu F. The security analysis of popular instant messaging applications. 2017 international conference on computer systems, electronics and control (ICCSEC), м. Dalian, 25–27 груд. 2017 р. 2017. URL: https://doi.org/10.1109/iccsec.2017.8446863 (дата звернення: 31.08.2025).

6. Blockchain challenges and opportunities: a survey / Z. Zheng та ін. International journal of web and grid services. 2018. Т. 14, № 4. С. 352. URL: https://doi.org/10.1504/ijwgs.2018.095647 (дата звернення: 31.08.2025).

7. Benet J. IPFS - content addressed, versioned, P2P file system. 2014. (Препринт). URL: https://arxiv.org/abs/ 1407.3561 (дата звернення: 31.08.2025).

8. An improved P2P file system scheme based on IPFS and Blockchain / Y. Chen та ін. 2017 IEEE International Conference on Big Data (Big Data), м. Boston, MA, 11–14 груд. 2017 р. 2017. URL: https://doi.org/10.1109/ bigdata.2017.8258226 (дата звернення: 31.08.2025).

9. Rateb J. Blockchain for the internet of vehicles: a decentralized iot solution for vehicles communication and payment using ethereum: Дисертація на здобуття ступеня доктора філософії. Париж, Франція, 2021. URL: https://hal.science/tel-03563633 (дата звернення: 31.08.2025).

10. Secure messaging platform based on blockchain / U. P. Ellewala та ін. 2020 2nd international conference on advancements in computing (ICAC), м. Malabe, Sri Lanka, 10–11 груд. 2020 р. 2020. URL: https://doi.org/ 10.1109/ icac51239.2020.9357306 (дата звернення: 31.08.2025).

11. Побережник В., Опірський І. Розробка концепції методу використання технології блокчейн для побудови системи обміну повідомленнями. Захист інформації. 2023. Т. 25, № 2. С. 62–70. URL: https://doi.org/10. 18372/2410-7840.25.17673 (дата звернення: 31.08.2025).

12. Balatska V., Poberezhnyk V., Opirskyy I. Use of non-fungible tokens and blockchain to demarcate access to public registries. Cybersecurity: education, science, technique. 2024. Т. 4, № 24. С. 99–114. URL: https://doi.org/10. 28925/2663-4023.2024.24.99114 (дата звернення: 31.08.2025).

13. Balatska V., Poberezhnyk V. The concept of applying blockchain technologies to increase the security of personal data of the "diya" platform: compliance with the requirements of the gdpr and ukrainian legislation. Cybersecurity: education, science, technique. 2024. Т. 2, № 26. С. 268–290. URL: https://doi.org/10.28925/2663-4023. 2024.26.681 (дата звернення: 31.08.2025).

14. Turner S., Chen L. Updated security considerations for the MD5 message-digest and the HMAC-MD5 algorithms. RFC Editor, 2011. URL: https://doi.org/10.17487/rfc6151 (дата звернення: 31.08.2025).

15. Wang X., Yin Y. L., Yu H. Finding collisions in the full SHA-1. Advances in cryptology – CRYPTO 2005. Berlin, Heidelberg, 2005. С. 17–36. URL: https://doi.org/10.1007/11535218_2 (дата звернення: 31.08.2025).

16. Content Identifiers (CIDs) | IPFS Docs. IPFS Documentation | IPFS Docs. URL: https://docs.ipfs.tech/ concepts/content-addressing/#what-is-a-cid (дата звернення: 31.08.2025).

17. Comparative review of selected internet communication protocols / L. Kamiński та ін. Foundations of computing and decision sciences. 2023. Т. 48, № 1. С. 39–56. URL: https://doi.org/10.2478/fcds-2023-0003 (дата звернення: 31.08.2025).

18. Ethereum gas and fees: technical overview | ethereum.org. ethereum.org. URL: https://ethereum.org/ en/ developers /docs/gas/ (дата звернення: 31.08.2025).

19. Fastest blockchains by TPS [2025] | chainspect. Chainspect. URL: https://chainspect.app/dashboard?gainers= false&order=desc&sort=tps (дата звернення: 31.08.2025).

Надійшла 31.08.2025