КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.31.1033
УДК 004.056

Побережник Василь Олегович
Аспірант кафедри захисту інформації
Національний Університет «Львівська політехніка», Львів, Україна
ORCID: 0000-0002-7523-2557
vasyl.poberezhnyk@gmail.com

Наконечний Тарас Ігорович
Аспірант кафедри захисту інформації
Національний університет «Львівська політехніка», Львів, Україна
ORCID: 0009-0003-4487-9424
taras.i.nakonechnyi@lpnu.ua

ДОСЛІДЖЕННЯ ГІБРИДНОЇ СИСТЕМИ КЕРУВАННЯ ОБЛІКОВИМИ ДАНИМИ НА ОСНОВІ ПОЄДНАННЯ ТЕХНОЛОГІЙ DID, IPFS, BLOCKCHAIN

Анотація. У цій роботі розглядаються концепції децентралізованих ідентифікаторів та пов'язаних з ними облікових даних, система керування такими даними та відмінності між традиційними та децентралізованими ідентифікаторами. Окрім цього розглянуто переваги та недоліки децентралізованих ідентифікаторів, які уже існують та проведено пошук вирішення викликів, які стоять перед такими ідентифікаторами. До викликів, які стоять перед децентралізованими ідентичностями віднесено проблему керування ключами, складність впровадження, регуляторну невизначеність, вартість операційної підтримки і зручність користування. Для вирішення цих проблем при запропонували концепцію системи із використанням гібридного підходу, який характеризується використанням поєднанням клієнт-серверної архітектури із збереження облікових даних у IPFS та відбитків у блокчейн, незмінність якого має убезпечити від використання підроблених облікових даних. Також порівняли способи прив'язування даних IPFS до сталого ідентифікатора, який може вказувати на різні дані: IPNS, DNSLink, ENS. На основі цього розробили ENS-IPFS модель децентралізованого ідентифікатора, та продемонстрували шлях інтеграції такої децентралізованої моделі в гібридну систему керування обліковими даними. Під час розробки системи ми також розробили математично-логічний апарат, який дозволяє визначити приналежність облікових даних до множини надійних, та описали функцію автентифікації користувача через застосування відбитку його облікових даних та вирішення криптографічного завдання для доказу того, що він володіє приватним ключем, на основі якого було сформовано публічний ключ, який у поєднанні із ідентифікатором даних формує відбиток, який перевіряється на відповідність тому, що є збереженим у блокчейн. Результати проведеного дослідження дозволили підтвердити гіпотезу про можливість застосування гібридних технологій у контексті використання децентралізованих технологій та керування обліковими даними. Окрім цього отримані дані дозволили сформувати шляхи для подальших досліджень і сформувати висновок про можливість застосування підходу як такого, що має перспективи для застосування паралельні із централізованими та децентралізованими системами, а також може відігравати проміжну ланку в переході від централізованої до децентралізованої моделі.

Ключові слова: DID; IPFS; blockchain; приватність; децентралізація; смарт контракти; IPNS; DNSLink.

ВСТУП

Однією із характеристики сьогодення є обробка великої кількості інформації пов'язаної із ледь не всіма сферами діяльності людини: урядування, бізнес, охорона

здоров'я, наука, відпочинок, розваги, мистецтво тощо — всі вони пов'язані тим, що збираються і обробляють інформацію про користувачів, яка необхідна для роботи систем, особливо тих, які залежать від облікових даних користувачів. Ці дані дозволяють сформувати профіль користувача, керувати його дозволами в системи, надавати необхідні послуги чи персоніфікувати дані, які користувач отримає. Очевидно що такі системи оперують неймовірною кількістю інформації про конкретного користувача, що може створювати занепокоєння щодо безпеки даних, оскільки такі об'єми інформації можуть цікавити не лише авторизованого обробника такої інформації, але й зловмисників. Це пов'язано із тим, що персональна інформація користувача дозволяє ідентифікувати його, визначити його вподобання, створити певний профіль користувача який може використовуватися, наприклад, для атаки на користувача через соціальну інженерію, шантаж тощо, тому захист облікових даних користувача і його приватності є одним із найважливіших завдань будь-якого обробника даних.

Для керування обробки таких даних використовують системи керування обліковими даними, наприклад ті, які розробляються корпораціями Microsoft чи Oracle. Застосування цих систем надає ряд переваг, наприклад спрощений облік користувачів і надання доступу до ресурсів підприємства [1]. Іншим прикладом є соціальні мережі, які обслуговують мільйони користувачів і мають доступ до не меншої кількості облікових записів, кожен з яких потребує методів і засобів обробки. Ба більше, концептуальної різниці між корпоративною системою керування обліковими даними і такою системою в соціальній мережі чи месенджере або інтернет-магазині практично немає, оскільки вони виконують одне і те саме завдання, якщо абстрагуватися від специфіки напрямку,— вони дозволяють створювати і керувати обліковими записами. Найрозповсюдженішими і, напевне, традиційним підходом є застосування централізованих систем.

Постановка проблеми. Традиційні системи керування обліковими даними, тобто централізовані системи, і самі централізовані облікові дані давно здобули популярність серед користувачів і надавачів послуг, урядових організацій тощо. Ця популярність зумовлена тим, що такі системи є відносно простими для реалізації і підтримки, спрощує доступ до даних і процес керування. Але через свою централізовану природу вони мають ряд недоліків:

1. Центральна точка відмови: у випадку компрометації бази даних, користувача з привілеями чи системи керування обліковими даними, вся інформації про користувачів опиняється під загрозою, що призводило до масштабних витоків персональних даних [2].
2. Пастка постачальника: користувачі, а також системи чи сервіси, можуть залежати від єдиного провайдера ідентифікації, наприклад від його політик, надійності систем чи банальної працездатності.
3. Відсутність контролю з боку користувача: користувачі не мають повного контролю над власними цифровими ідентифікаторами, оскільки вони здебільшого обробляються провайдером ідентифікаторів, а не самим користувачем.

Останній пункт, наприклад, намагаються вирішити законодавчо, наприклад в Європейському Союзі є запроваджені GDPR [3], які повинні регулювати те, як обробляється персональна інформація громадян ЄС. Однак такий підхід є лише частковим вирішенням проблеми, оскільки він з одного боку працює лише з організаціями які надають послуги громадянам ЄС, а з іншого — виявлення порушень вимог GDPR є нетривіальним завданням. Тож проблему застосування централізованих систем доступу можна охарактеризувати як

Аналіз останніх досліджень і публікацій. Способом, який дозволяє вирішувати згадані недоліки є застосування децентралізованих ідентифікаторів [4], які дозволяють забезпечувати захист приватності користувачів і контроль над власними даними як властивість підходу за замовчуванням. Суть цього підходу полягає в тому, що після отримання будь-яких облікових даних, вони переходять у розпорядження самому користувачу і зберігаються у децентралізованій системі, що можна охарактеризувати як самосуверенну ідентичність [5], а децентралізовані ідентифікатори є основою такого підходу. Наприклад, ЄС має пілотний проект, який досліджує застосування таких ідентифікаторів та заснований на технології блокчейн, — European Blockchain Service Initiative [6]. Окрім цього існує ряд інших провайдерів децентралізованих ідентифікаторів, які надають можливість створити власний децентралізований ідентифікатор на основі блокчейн чи смарт контрактів [7].

Як бачимо, сама технологія децентралізованих ідентифікаторів є доволі перспективною, зважаючи на те, що її розвитком займаються різні організації, а також ЄС. Окрім цього, переваги методу, дозволяють його розглядати, як можливу альтернативу або паралельний метод керування обліковими даними паралельно із централізованими системами. До цих переваг можна віднести такі пункти:

1. Автентифікація на основі асиметричної криптографії: для авторизації та автентифікації використовуються приватний ключ, який перебуває під контролем лише користувача і більше нікому не відомий.
2. Відсутність центральної точки вразливості: облікові дані поширюються у децентралізованій мережі, що усуває єдину точку відмови та атак.
3. Конфіденційність та мінімальне розкриття даних: децентралізовані ідентифікатори дозволяють застосовувати принцип вибіркового розкриття, що дозволяє користувачу видавати тільки ту інформацію про себе, яка мінімально необхідна, а не більш широкі об’єми, як це відбувається у централізованих системах.
4. Незалежність користувача: окрім того, що користувач сам контролює власні ключі, він також не є прив’язаним до якогось одного розпорядника облікових даних, оскільки після отриманні ідентифікатора він зберігається незалежно від провайдера.

Однак такий спосіб має свої недоліки, які можуть стати на заваді застосування чи реалізації. До основних недоліків методу можна віднести такі пункти:

1. Керування ключами доступу: оскільки доступ до даних опирається на приватний ключ користувача, втрата такого ключа може призвести до повної втрати доступу до своїх даних.
2. Складність впровадження: одні
3. Регуляторна невизначеність: на відміну від централізованих системи, такий підхід досі перебуває у сірій зоні правового поля і його застосування в юридичному контексті перебуває у стані невизначеності, навіть згаданий приклад із EBSI перебуває у стані дослідження і при виявленні несумісностей проект обіцяють згорнути.
4. Вартість зміни даних: кожна транзакція у блокчейн чи смарт контракті вимагає використання обчислювальних ресурсів, ціна яких залежить від завантаження мережі, розміру мережі тощо і може становити відносно великі суми.
5. Зручність користування: зручність користування такими ідентифікаторами може відлякувати користувачів, оскільки такі ідентифікатори можуть мати складний для розуміння і запам’ятовування формат.

Аналізуючи ці переваги та недоліки, а також зіставлення їх із недоліками централізованих систем, можна зрозуміти, що хоча такий метод має своє переваги, які є достатньо вагомим, щоб розглядати його, як можливий для застосування, однак його недоліки спонукають до пошуку рішень згаданих проблем.

Мета статті. Метою статті є дослідження застосування децентралізованих ідентифікаторів в системах керуванням обліковими даними, які застосовують гібридний підхід у своїй побудові, тобто поєднують у собі традиційну централізовану архітектуру і децентралізовані системи на кшталт блокчейн та IPFS. Розробка моделі ідентифікатора, який буде зрозумілий і зручний для користувача, і одночасно забезпечуватиме переваги методу і нівелюватиме його основні недоліки.

МЕТОДИКА ДОСЛІДЖЕННЯ

У цьому дослідженні ми використовували якісне порівняння характеристик пропонованого методу із централізованим.

Для побудови пропонованої системи було розроблено гібридний підхід із застосуванням традиційної клієнт-серверної архітектури, інтеграцію смарт контрактів локальну версію блокчейну Ehtereum та його тестову мережу Sepolia, а також IPFS. Для цього було використано такі компоненти:

- Hardhat3.0 — середовище для розробки та тестування смарт контрактів Ethereum.
- Ethers — JS-бібліотека, яка дає можливість інтегрувати блокчекйн і смарт контракти.
- Express — використовується для забезпечення роботи сервера і маршрутизації запитів.
- Solidity — мова програмування смарт контрактів Ethereum.
- NodeJS — відіграє роль основи для запуску сервера та hardhat.
- PinataSDK — застосовується для інтеграції IPFS.
- JavaScript — мова програмування клієнт-серверної частини.
- Pug — HTML-шаблонізатор.

Окрім цього було розроблено математичний логічний апарат, який описує критерії визначення децентралізованого ідентифікатора як надійного, та дозволяє формалізувати визначення принадежності ідентифікатора до множини верифікованих і прийняти рішення про надання прав доступу.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Для дослідження нашої гіпотези ми розробили гібридну систему, яке скеладається із децентралізованої ідентичності, блокчейн та смарт контрактів, IPFS, та традиційної клієнт-серверної архітекту. Компоненти і процес розробки і дослідження системи описано в наступних підрозділах.

Розробка DID. В основі пропонованої децентралізованої ідентичності лежить поєднання пари приватний-публічний ключ, які чітко дозволяють верифікувати власника ключа за допомогою відкритого ключа і не розкрити його закритий ключ. Це досягається за допомогою застосування криптографії на еліптичних кривих.

При такому підході, публічний ключ є точкою на еліптичній кривій – результат операції множення на еліптичній кривій закритого ключа і певної точки генератора. Особливістю такої операції є те, що не існує операції оберненої до множення на

еліптичних кривих, тобто навіть знання точки G не дозволить отримати приватний ключ. На сьогодні єдиним можливим способом дізнатися приватний ключ з публічного є застосування перебору, і використання такого підходу до атаки на приватний ключ є фактично неможливим, при правильній реалізації алгоритмів. Також метод ECDSA, який засновується на використанні криптографії на еліптичних кривих є перевіреним часом способом для підтвердження є, так би мовити, стандартом при використанні цифрових підписів [8]. Зважаючи на це, вибір такої пари ключів дозволяє одночасно верифікувати власника ключа і захистити його ключ від компрометації через атаки підбору ключів, а також дати користувачу ідентифікатор, який є практично випадковим і ніяк не ідентифікує самого користувача, оскільки сам ключ є випадковим і ніяк не пов'язується із будь-якими персональними даними при самому створенні.

З такою основою, можна перейти до розробки самої децентралізованої ідентичності. Тут ми пропонуємо підхід оснований на застосуванні поєднання різних децентралізованих технологій: блокчейн, смарт контракти та IPFS.

В такому підході IPFS відповідає за збереження самих даних децентралізованої особистості, смарт-контракт дозволяє пов'язати їх між публічним ключем та IPFS-даними, а блокчейн потрібен для роботи самих контрактів і забезпечення прозорості та надійності збережених даних у смарт контракті.

Відношення між згаданими засобами продемонстровано на рисунку N.

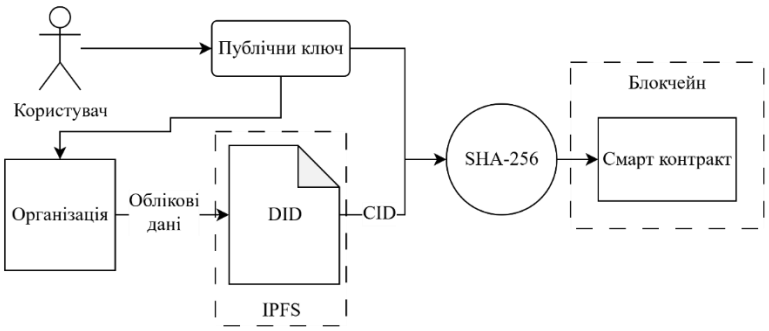

Рис. 1. Зв'язок між компонентами пропонованої DID

Як видно з рисунку 1, IPFS-дані (DID) — це дані, які представляють собою децентралізовану ідентичність. Її особливість, як було згадано раніше, полягає в тому, що контролю над нею має сам користувач, на відміну від традиційних систем, де контроль мають централізовані установи, тому користувач може сам вирішувати як користуватися цими даними. При застосуванні такого підходу, установа, яка раніше створювала облікові дані, тепер переходить до способу, коли вона може лише надавати або блокувати доступ до свої сервісів користувачу, замість створення нового облікового запису користувача із обробкою персональних даних тощо. Іншими словами, користувач має можливість керувати власними даними та вирішувати те, хто буде мати до них доступ, а інша сторона може лише вирішувати допускати користувача до власних сервісів чи ні.

За основу децентралізованої ідентичності, ми обрали стандарт WC3-DID [9], який описує DID-документ та способи запису інформації в ньому через JSON, та спростили задля спрощення реалізації прототипу системи.

Структуру тестової децентралізованої ідентичності зображено на рисунку 2.

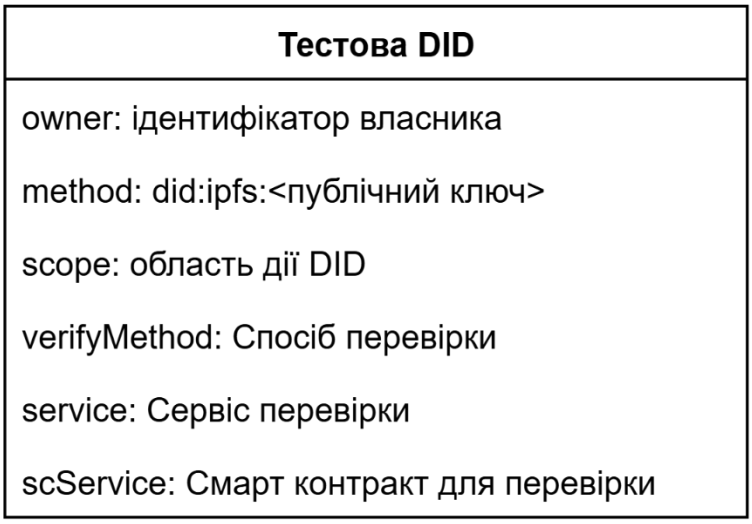

Рис. 2. Структура тестової DID

Як бачимо, сама структура дещо спрощена, якщо порівнювати її із тим, що описано у стандарті. Це зроблено заради спрощення процесу розробки системи, оскільки вона слугувала експериментальним майданчиком, для дослідження можливості використання нашого підходу. Після цього ми обрали спосіб прив'язки даних між публічним ідентифікатором та даними DID. Для цього ми провели аналіз трьох популярних способів пов'язування даних із різними ідентифікаторами, що описано у наступному підрозділі.

Порівняння методів прив'язування IPFS-контенту до ідентифікатора. Основна проблема в оновленні даних, які зберігаються в IPFS, полягає в тому, що при будь-якій зміні самих даних, буде змінюватися і CID, тобто їхній ідентифікатор в системі, що робить оновлення будь-яких даних, які збережені в цій системі, і їхнє відслідковування нетривіальним завданням, оскільки такі дані, буду вважатися фактично новими і ніяк не пов'язаними із попередньою версією [10]. Зокрема ця проблема часто зустрічається при розміщенні, наприклад, веб-сайтів у Web3.0, де місцем збереження даних цих веб сайтів часто є саме IPFS.

Для вирішення цієї проблеми часто застосовують такі підходи: IPNS – InterPlanetary Name Spaces, DNSLink – пов`язування традиційного DNS імені із контентом у мережі IPFS та BNS – Blockchain Name Spaces, реалізацією якого є ENS (Ethereum Name Spaces).

Ці підходи покликані вирішувати проблему пов'язування даних IPFS та їхніх ідентифікаторів, які постійно змінюються, із сталим вказівником на необхідні дані. Попри те, що ці методи виконуються одне й те саме завдання, спосіб виконання цього завдання кардинально відрізняється одне від одного.

Підхід заснований на IPNS покладається на два підходи: DHT та PubSub. Його ідея полягає в тому, щоб посилатися на дані не через ідентифікатор контенту, як при роботі із файлами в IPFS, а використовувати вказівник, який може змінюватися і після оновлення вказувати уже на інший ідентифікатор. Відмінність зображено на рисунку 3.

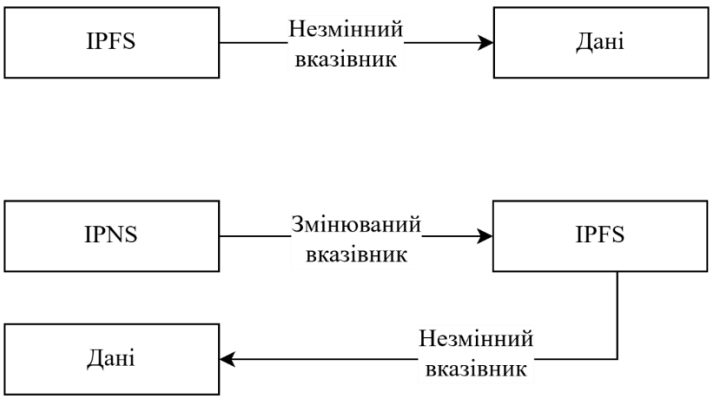

Рис. 3. Відмінність між адресацією за контентом та IPNS

За замовчуванням використовується механізм DHT — розподілені хеш таблиці, які оновлюються за замовчуванням кожні 48 годин для забезпечення існування в них актуальної інформації. Хоча цей й зроблено заради актуальності інформації, такий підхід має недолік, який полягає в тому, що уже ідентифікатор, який існує, необхідно постійно републікувати в мережу, тобто необхідно, щоб існував хоча б один вузол, який знає про IPNS-ідентифікатор і регулярно його публікує в мережу. Наприклад Kubo, одна з реалізацій вузлів IPFS, виконує цю операцію кожних чотири години. На практиці користувачі також часто послуговуються сервісами, які дозволяють виконувати це завдання замість них.

На противагу цьому підходу, метод PubSub дозволяє підписатися на оновлення конкретного IPNS, через застосування peer-to-peer повідомлень. Цей спосіб є швидшим, для отримання нових змін, однак він має свій недолік. PubSub-записи є ефемерними і існують впродовж процесу поширення, тому запровадження такого підходу також потребує впровадження додаткового шару в систему – шару стійкості. Також відмінність цих підходів полягає у швидкості отримання даних: при застосуванні DHT процес отримання даних може займати довший час, ніж PubSub, оскільки потрібно знайти актуальний запис в мережі.

Ще одним підходом є застосування DNSLink. Тут, для отримання стійкого посилання на дані, використовується поєднання традиційних централізованих систем із децентралізованою IPFS. Сам підхід до застосування DNSLink не є чимось новим і використовується для утворення стійкого посилання між традиційними іменем у DNS та децентралізованими даними і опирається на RFC 1464 [11]. Для цього використовується стандартний TXT-запис формату dnslink=<value> у налаштуваннях доменного імені. На рисунку 4 зображено приклад такого файлу.

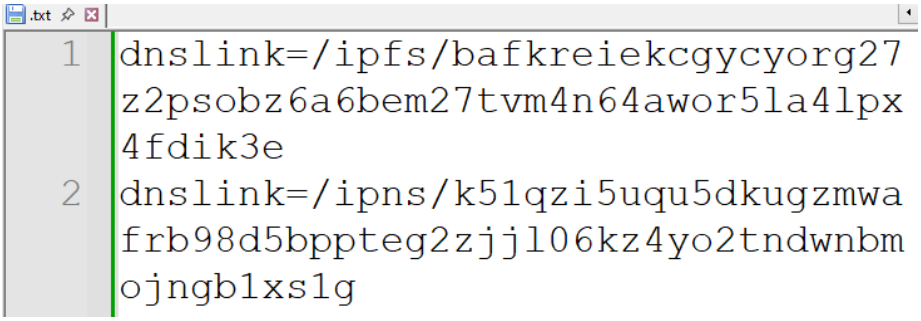

Рис. 4. Приклад запису TXT-файлу для DNSLink

Специфікація DNSLink вимагає, що це виконувалося на окремому субдомені «_dnslink», наприклад, _dnslink.example.com, який належить до простору основного доменного імені example.com. Застосування цього підходу також можна розділити на умовні два сценарії: посилання на CID та посилання на IPNS. При першому підході завжди буде відбуватися посилання на вказані дані у CID. Такий підхід хоча і є простішим, однак при необхідності змін даних, потрібно буде завжди оновлювати TXT-запис, щоб він вказував актуальні дані. При другому підході сформується статичне посилання на IPNS, яке не потрібно буде змінювати, при зміні самих даних, однак будуть вводитися ті самі обмеження, що і при застосуванні IPNS. Також варто зазначити, що при такому підході втрачається перевага децентралізації в деякій мірі, оскільки тут застосовується уже інфраструктура, яка є централізованою і може залежати від впливу власників інфраструктури, цензури тощо. Втім, слід наголосити, що такий вплив буде зберігатися саме на частину, яка залежить від провайдера послуг, який обслуговує доменне ім'я, а не на самі дані, що зберігаються в IPFS.

Наступним методом є застосування ENS — Ethereum Name Space, або Простір Імен Ефіру. Цей підхід є найскладнішим із цих трьох, але й водночас є найбільш гнучким, а також децентралізованим, за своєю природою, оскільки опирається на застосування смарт контрактів та блокчейн. На рисунку 5 зображено створений запис в тестовій мережі Sepolia.

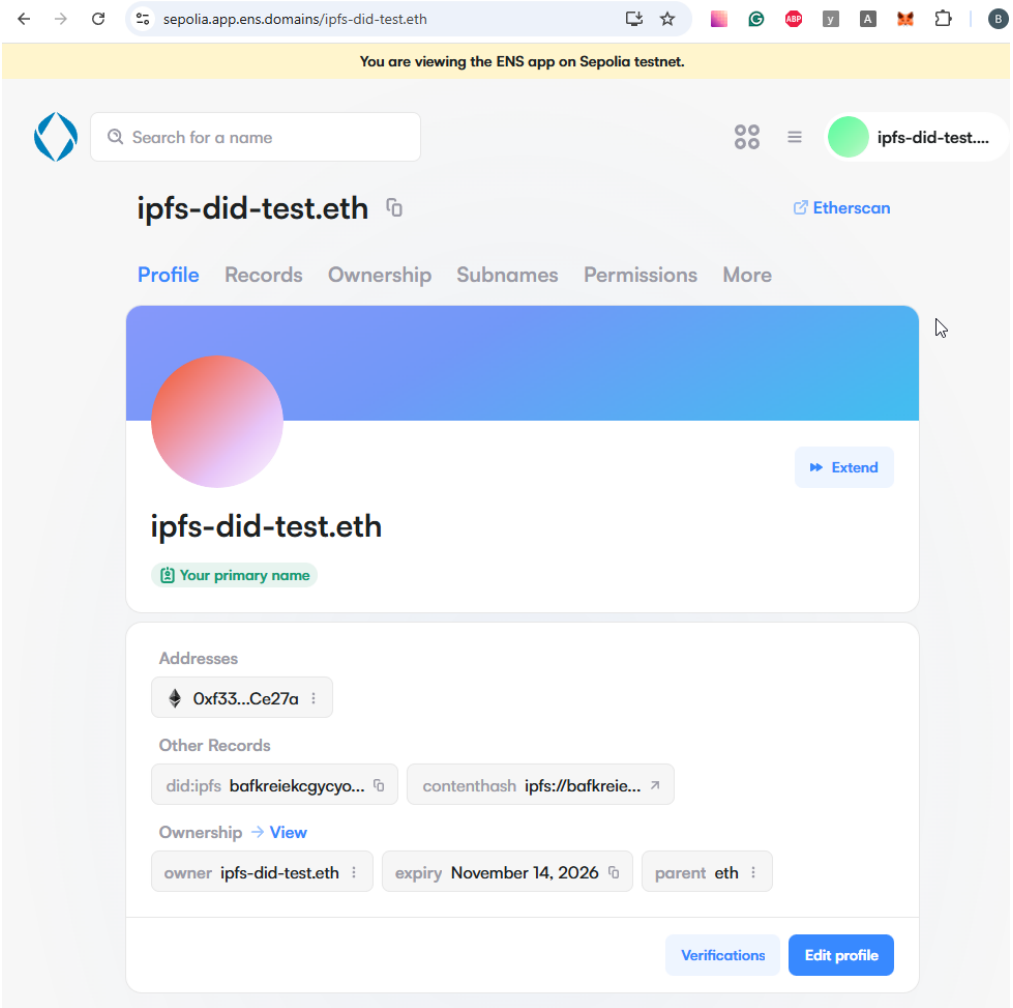

Рис. 5. ENS-запис в тестовій мережі Sepolia

Як видно на рисунку X, цей запис має різні прив'язки до контенту, наприклад contenthash, який вказує на той самий файл, який був прикладом для IPNS, а також текстовий запис did:ipfs, який також вказує на той самий файл. Назва did:ipfs вказує на те, що він посилається на децентралізовану ідентичність — DID, та метод її обробки — IPFS. Фактично ENS можна класифікувати як NFT [12], а точніше ERC-721, які зберігаються у блокчейн.

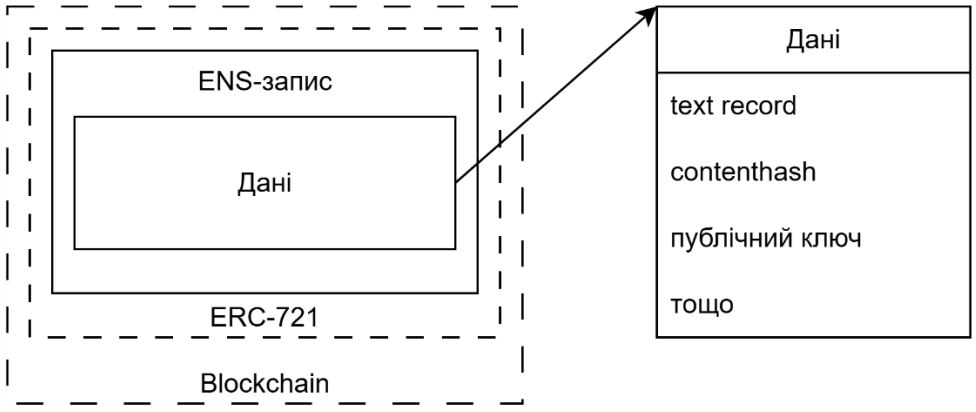

Рис. 6. Зображення приналежності ENS до множини ERC-721

Тобто ENS є смарт контрактом, що випливає із стандарту NFT, тому застосування ENS не лише дозволяє встановити зв'язок між даними в IPFS та ідентифікатором, але й додати будь-яку логіку, яку можна запрограмувати через застосування смарт контрактів, що значно розширює можливості таких ідентифікаторів.

У таблиці один наведено порівняльну характеристику розглянутих способів пов'язування даних між ідентифікатором та контентом в IPFS.

Таблиця 1

Порівняння методів прив'язуванні IPFS даних

Критерій	IPNS	DNSLink	ENS
Базова технологія	Асиметрична криптографія	Система доменних імен (DNS)	Блокчейн та смарт контракти
Метод прив'язки	Створення змінюваного вказівника на CID	Місток між DNS та CID/IPNS	Місток між смарт контрактом та даними
Ідентифікатор	Хеш значення публічного ключа вузла	Традиційне доменне ім'я	Доменне ім'я у домені верхнього рівня .eth
Механізм оновлення	Публікація новго запису в DHT/PubSub	Зміна TXT-запису в DNS налаштуваннях	Транзакція в блокчейн з викликом смарт контракту
Вартість оновлення	Майже безкоштовна, витрати лише на обчислення	Відносно низька, уже враховано в послуги з надання доменного імені	Висока
Затримка оновлення	При DHT — відносно висока, PubSub — низька.	Залежить від TTL TXT-запису. (60 секунд – 24 години)	Відносно швидка (Залежить від швидкості блоку Ethereum ~12 сек.)

Стійкість ідентифікатора	Потребує постійної републікації	Теоретично стійкий. (Залежить від провайдера)	Стійкий. (Зберігається в мережі допоки сплачена оренда імені)
Можливі загрози	Компрометація приватного ключа	Компрометація облікового запису користувача, DNS-спуфіг, BGP-хайджекінг, цензура тощо	Компрометація приватного ключа, втрата криптогаманця, вразливості смарт контрактів
Тип ідентифікаторів	IPNS	IPNS, CID	Адреси, CID, тестові метадані, смарт контракти тощо. Майже не обмежені через застосування смарт контрактів.
Децентралізація	Теоретично високий рівень децентралізації, практично — залежить від кількості вузлів в мережі, місця розташування вузлів тощо.	Низька, оскільки залежить від надійності та доброчесності провайдерів.	Високий рівень децентралізації. Присутній парадокс децентралізації, коли децентралізовані ідентифікатори посилаються на дані, що зберігаються централізовано [13].

З порівняння видно, що кожен з методів має свої переваги та недоліки, наприклад IPFS підхід залежить від «живучості» ідентифікатора, та потреби його регулярного оновлення за для збереження в мережі, DNSLink залежить від постачальників послуг і може піддаватися тим самим загрозам, що й централізовані системи, що значно знижує рівень децентралізації, а метод ENS хоча і є найбільш гнучким і практично повністю децентралізованим, об’єкти, на які він посилається, можуть залежати від централізованих систем, що знову знижує децентралізацію підходу.

Якщо розглядати ці підходи в контексті децентралізованих особистостей, то тут також варто опиратися на можливість забезпечення анонімності користувача. В такому випадку DNSLink буде програвати, оскільки для реєстрації доменного імені, необхідно вказувати особисті дані, що уже є загрозою найвищого рівня для користувача, оскільки такі дані дозволяють чітко встановити особистість, що уже компрометує весь наступний ланцюжок. Однак такий недолік буде найбільш вагомим у системах, де є важливою приватність та анонімність користувачів.

Також недолік може мати застосування IPNS, оскільки невчасно оновлення DHT, може призвести до втрати можливості отримання даних за ідентифікатором. Втім ця проблема не є незворотною, оскільки IPNS формується із приватного ключа вузла, що дає змогу відновити можливість отримання даних, оскільки сам ідентифікатор є сталим. Відповідно така проблема може стати причиною тимчасової втрати доступу до даних за посиланням, але самі дані будуть залишатися доступними, оскільки вони дублюються на кількох вузлах.

Розробка системи керування. Систему керування обліковими записами можна умовно розділити на дві підсистеми: керування обліковими записами та система керування доступом. Відповідно і для системи основаної на поєднанні пропонованих технології в цьому контексті вона не буде відрізнятися. На рисунку 7 зображено схему пропонованої системи.

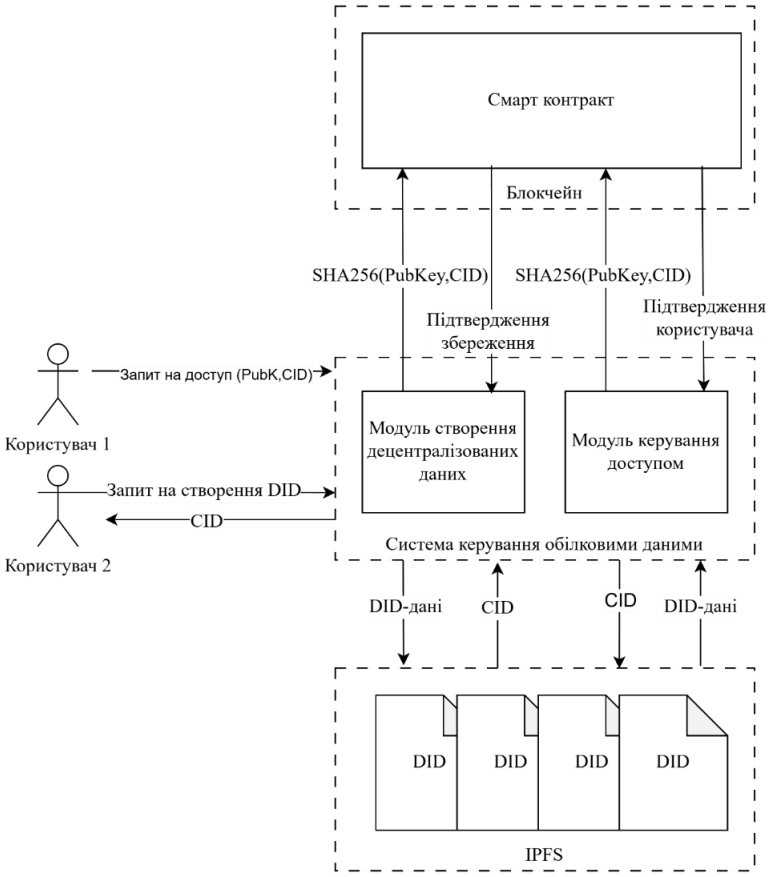

Рис. 7. Зображення структури тестової системи

В цій системі існує два алгоритми: створення нових облікових записів, авторизація існуючих. Для створення нового децентралізованого облікового запису виконаються такі кроки:

1. Користувач надсилає запит на створення нового DID та надає публічний ключ.
2. Система створює новий запис, встановлює публічний ключ як ідентифікатор у записі та завантажує документ у IPFS.
3. Система генерує Hash(PublicKey:CID) та передає його у смарт контракт.
4. Смарт контракт підтверджує збереження даних.
5. Після підтвердження система передає користувачу CID.

Для отримання доступу до захищеної частини алгоритм буде мати такий вигляд:

1. Користувач передає пару PublicKey:CID в систему.
2. Система генерує геш значення пари перевіряє наявність цього значення у смарт контракті.
3. Якщо значення знайдено, то система завантажує DID з IPFS та перевіряє надані права, якщо не знайдено — відмовляє у доступі.
4. Якщо необхідні права доступу є у DID, то користувачу надається доступ, якщо немає, то користувач отримує відмову.

Для дослідження пропонованого підходу було обрано модель, в якій смарт контракт забезпечує існування сховища підтверджений даних, яке складається із гешованих значень пар публічного ключа і ідентифікатора DID в IPFS, тож якщо DID належить до множити значень збережених у смарт контракті, його можна вважати підтвердженим користувачем. Виразити це можна цим виразом:

$$VcS = \{h_1, h_2 \dots h_n\} \quad (1)$$

де VcS – множина підтверджених геш значень пар ключів, h_i – геш значення пари публічний ключ і ідентифікатор IPFS, а h_i визначається таким виразом:

$$h$$

де h_i – геш значення, H – функція гешування sha256, $PubKey_i$ – публічний ключ користувача, CID_i – ідентифікатор DID в IPFS. Опираючи на це, можна зробити критерій належності DID до автентифікованих користувачів виражений таким виразом: $=$

$$\equiv$$

де VC – верифікований користувач. Тож на основі цього виразу можна розробити критерій прийняття рішення про допуск користувача до системи:

$$=$$

де A – допуск користувача, L – функція перевірки користувача, $nonce$ – випадкове число для перевірки підпису користувача.

Алгоритм роботи функції авторизації користувача має такі кроки:

1. Система надсилає користувачу випадкові дані для підпису. Користувач підписує дані та надсилає у відповідь підпис, CID, публічний ключ.
2. Система перевіряє підпис, якщо автентичність підтверджується, то далі відбувається перевірка автентичності пари за виразом (3).
3. Якщо автентичність підтверджено, то система завантажує DID і перевіряє права доступу, інакше у доступі буде відмовлено.
4. Користувач отримує доступ згідно з його правами, або отримує відмову, якщо таких прав немає.

При такому підході можна вилучити з процесу автентифікації передачу паролів чи особистих даних, що може підвищити рівень безпеки користувачів, а збереження даних верифікації користувачів у блокчейн – від несанкціонованої зміни. Також захист цілісності та достовірності DID може бути гарантований тим, що будь-які зміни будуть одразу відображені, оскільки сам CID зміниться і згенерований з його використанням геш буде відрізнятися від того, що уже є збереженим у смарт контракті.

Однак під час дослідження запропонованої моделі було суттєвий недолік, який може впливати на зручність і правильність користування системою, — застосування публічного ключа як децентралізований ідентифікатор. $)$

Хоча сам публічний ключ може слугувати ідентифікатором, однак його застосування є недоцільним хоча б тому, що це незручно з точки зору користувача і може призводити до помилок, а також така необхідність щоразу вводити такий ключ може відлякувати користувачів.

Для нівелювання цього недоліку ми застосували згаданий у попередньому розділі підхід — використання ENS для зв’язування ідентифікатора і даних. Такий підхід дозволив утворити зручний для запам’ятовування та користування ідентифікатор, посилання на який дозволяє отримати як і саму зв’язку публічного ключа та ідентифікатора даних (CID), а також автоматизувати отримання необхідних даних із ENS. Окрім цього, такий підхід є найбільш децентралізованим, оскільки ENS існує в $\in$ просторі блокчейн Ethereum, що дозволяє отримати справжній децентралізований ідентифікатор із пов’язаними даними в децентралізованій мережі. На рисунку 8 зображено отриманий результат розв’язку ENS-імені в DID.

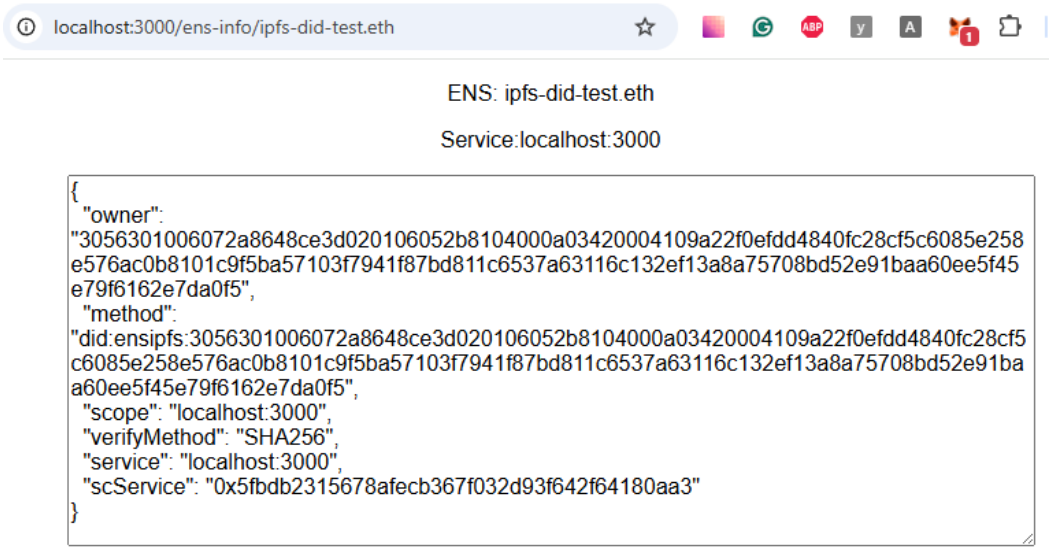

Рис. 8. Приклад розв'язку ENS-імені в DID

З метою перевірки гіпотези про застосування гібридних технологій, було вирішено, що саме система контролю буде приймати рішення про видачу чи відмову опираючись на дані, які збережені у IPFS та блокчейн. На рисунку 9 зображено діаграму проходження запиту. На рисунку 9 зображено діаграму проходження запиту.

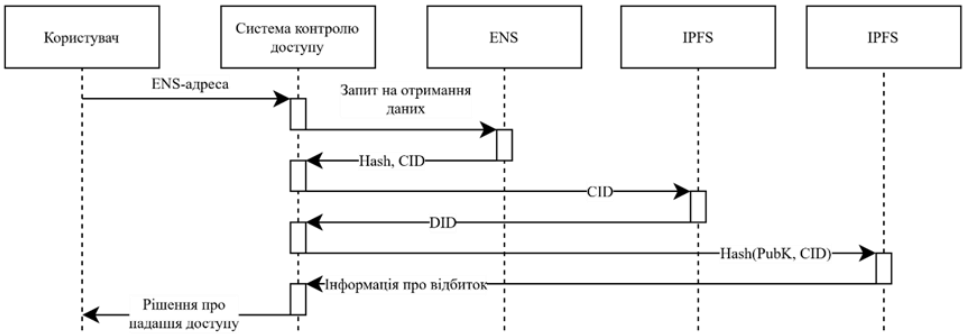

Рис. 9. Діаграма зображено діаграму проходження запиту авторизації

Як бачимо, застосування ENS-запису позбавляє користувача необхідності вводити дані, які раніше потрібно було ввести вручну. Таким чином виходить спростити сам процес для користувача, дати йому можливість користуватися ідентифікатором, який є не лише децентралізованим, але є легким для запам'ятовування, що також дозволить знизити кількість хибних спроб входу.

Після проведення згаданих дослідів, ми змогли отримати прототип гібридної системи, який дозволяє досліджувати запропоновано гіпотезу. Основними перевагами згаданого методу є те, що він може застосовувати уже наявну інфраструктуру публічних мереж IPFS та блокчейн Ethereum, що частково вирішує проблему розгортання додаткової інфраструктури, однак застосування Ethereum може нести додаткові фінансові витрати, особливо при розробці алгоритмів, які можуть виявитися неоптимізованим для роботи із смарт контрактами. Наприклад, на відміну від традиційних систем, смарт контракти оперують змінними, які знаходяться в різних областях пам'яті: storage, memory, calldata. Ці типи пам'яті мають різну ціну опрацювання (gasfee). Наприклад при зміні даних в області storage, варто прямувати до

мінімальної кількості операцій з ними, оскільки такі зміни є найдорожчими. Така потреба веде до потреби «керування пам'яттю» в смарт контрактах, що може стати викликом для розробки системи, оскільки такі операції в традиційних мовах беруть на себе здебільшого засоби мов програмування, а не спеціалісти. Такий недолік може призвести до відмови у обслуговуванні смарт контракту, коли ціна на проведення операції буде безмежно великою.

Ще одним аргументом на застосування такого методу є те, що дані зберігаються у відкритих децентралізованих мережах, що значно підвищує їхню стійкість до втрати чи несанкціонованої зміни, що забезпечується самими технологіями, однак вони також мають вразливості пов'язані із застосуванням смарт контрактів. Наприклад вразливості залишені випадково чи навмисно при написанні смарт контрактів.

У таблиці 2 наведено порівняння пропонованого методу із традиційними і уже наявними децентралізованими.

Таблиця 2

Порівняння пропонованого гібридного методу із централізованим та децентралізованим

Критерій	**Традиційні**	**Децентралізовані**	**Гібридні (пропонований метод)**
Розпорядник даних	Установа	Користувач	Користувач
Складність зміни даних	Просто	Висока	Висока
Складність відновлення доступу	Відносно просто	Складно	Складно
Приватність	Низька	Висока	Висока
Модель довіри	Інституціональна, публічні ключі	Криптографічні докази, розподілені реєстри	Публічні ключі, криптографічні докази, розподілені реєстри
Рівень децентралізації	Низький: присутня центральна точки вразливості/відмови	Високий: повністю децентралізовані	Середній: рішення про надання доступу приймається централізовано, збереження даних і доказів є децентралізованим.
Складність впровадження	Низька	Висока: повної інтеграції із децентралізованими технологіями та блокчейн.	Середня: потребує часткової інтеграції
Вартість змін даних	Низька: зміни в облікових даних входять у вартість роботи системи	Висока: кожна зміна в блокчейні чи смарт контракт потребує плати за проведення транзакції.	Середня: зміни даних у IPFS входять витрати на операційну діяльність, зміни у ENS — зменшені, оскільки обробляється відносно невеликий об'єм інформації, як і зміни відбитків у блокчейн.

З таблиці видно, що пропонований метод усуває деякі з недоліків централізованих систем, зокрема наявність централізованої точки вразливості, хоча й не повністю, оскільки самі дані особистості і ідентифікатор зберігаються незалежно від центральної

системи, ця система є необхідним компонентом, оскільки вона відповідає, наприклад, за авторизацію користувача тощо.

Важливим компонентом пропонованого методу є використання ENS, оскільки дані, з якими можна його пов'язати, дозволяють автоматизувати процес передачі даних між користувачем і системою, забезпечуючи при цьому децентралізацію підходу і анонімність користувача. Що вирішує одразу кілька проблем, зокрема те, що при такому підході мінімізується кількість посилок, які можуть бути викликані людським фактором, а також вирішує одну з проблем, яка притаманна багатьом технологіям, які пов'язані із блокчейн — відсутність зручного для користування ідентифікатора, оскільки ENS дозволяє одразу зберегти і пов'язати дані з ідентифікатором, а при запиті до смарт контракту надати необхідні дані. На додачу у цьому підході користувачу не потрібно оперувати складними для запам'ятовування публічними ключами чи адреса, а достатньо лише запам'ятати свій ENS-ідентифікатор, наприклад *ipfs-did-test.eth*. Ше одним недоліком методу, як і більшості децентралізованих, є проблема відновлення доступу до ідентифікатора, оскільки відновлення втраченого приватного ключа є практично неможливим. Однак, пропонований метод може мати в собі шляхи часткового вирішення проблеми: застосування приватних ключів гаманця. Наша модель засновувалась на тому, що ENS відігравав роль вказівника на облікові дані та ідентифікатор, однак модифікація системи і самих даних облікового запису, може дозволити підтвердити володіння обліковими даними через застосування підпису смарт контракту, а не лише генерацію підпису на основі закритого ключа ідентифікатора. Більше того, такий метод може дозволити створювати ланцюжки облікових даних, які через застосування підпису дозволятимуть одночасно відслідковувати зміни в облікових даних, а також підтверджувати право володіння ними. Однак тоді вразливість втрати ключа перейде із пари публічний-приватний ключ, приватний ключ власник смарт контракту і коло замкнеться. Тому вирішення такої проблеми є нетривіальним завдання і потребує детального пошуку рішення.

Як бачимо, запропонований варіант є справді гібридним рішенням і дозволяє нівелювати обмеження не гібридних методів в тій чи іншій мірі.

Проведене дослідження дозволяє підтвердити гіпотезу про можливість застосувати гібридних систем для керування обліковими даними, а також сформувати напрямки подальших досліджень, які розглядаються у висновках.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Проведена робота дозволила дослідити можливість інтеграції децентралізованих ідентифікаторів, зокрема запропонованої моделі на основі поєднання ENS та IPFS, із традиційними система, які засновуються на централізованому підході. В ході проведених експериментів ми отримали прототип гібридної системи керування обліковими записами із застосуванням принципів децентралізації. Основною перевагою запропонованого методу впровадження децентралізованих систем у керування обліковими даними є те, що він використовує для роботи уже готові публічну мережу IPFS та блокчейн Ethereum для зберігання децентралізованих ідентифікаторів, утворення зв'язку між ідентифікатором та даними, а також перевірці автентичності даних через застосування смарт контрактів.

Вагомим позитивним результатом роботи є те, що нам вдалося отримати емпіричне підтвердження працездатності нашої концепції та розглянути застосування гібридних

системи, що поєднують у собі централізовану та децентралізовану парадигми, як підхід, що може використовуватися як альтернативний до централізованих систем, а також, як проміжну ланку переходу від централізованих до децентралізованих систем.

Також ця робота дозволяє сформувати напрямки подальших дослідження децентралізації в контексті управління обліковими даними і застосуванням методу використання гібридних систем. Перш за все це напрямок пов'язаний із розробкою надійних способів відновлення даних, оскільки вся безпека ґрунтується на застосування приватного ключа, то його втрата чи викрадення може призвести до втрати доступу чи імперсоніфікації зловмисником, тому необхідний пошук дієвих та надійних шляхів відновлення доступу і блокування дій із втраченими записами.

Наступним важливим напрямком досліджень є пошук шляхів інтеграції децентралізованих методів у законодавство. У цьому напрямку запропонована гібридна система може стати основою для пошуку вирішення проблеми, оскільки принципи розроблені у роботі, дають змогу інтегрувати блокчейн та IPFS у існуючі системи, які уже є об'єктом юридичного регулювання і можуть дозволити вивчати нові технології через призму всім відомих і перевірених технологій, що, на нашу думку, мало б спростити процес дослідження цієї проблеми.

Іншим напрямком досліджень є можливість інтеграції таких систем із уже наявними, наприклад із згаданим спочатку EBSI, а також пошук можливих шляхів оптимізації роботи такої системи, зокрема здешевлення модифікацій стану смарт контракту, оскільки такі заходи є недешевими в блокчейні і погано спроектовані алгоритми можуть негативно впливати на вартість застосування такого методу.

Ще одним можливим і не менш важливим напрямком є дослідження реалізації принципів обмеженого розголошення, наприклад застосування доказів із нульовим розголошенням для часткового розголошення даних і створення основі цього перевірених облікових даних, які, наприклад, дозволили б застосування таких даних в площині урядування, банківських послуг тощо.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Baklykov, M. I. (2024). Efficiency of using IAM system at the enterprise. Connectivity, 167(1). https://doi.org/10.31673/2412-9070.2024.014851
2. Makhdoom, I., Zhou, I., Abolhasan, M., Lipman, J., & Ni, W. (2020). PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities. Computers & Security, 88, 101653. https://doi.org/10.1016/j.cose.2019.101653
3. Zaeem, R. N., & Barber, K. S. (2020). The effect of the GDPR on privacy policies. ACM Transactions on Management Information Systems, 12(1), 1–20. https://doi.org/10.1145/3389685
4. Cucko, S., & Turkanovic, M. (2021). Decentralized and self-sovereign identity: Systematic mapping study. IEEE Access, 9, 139009–139027. https://doi.org/10.1109/access.2021.3117588
5. Побережник, В., Балацька, В., & Опірський, І. (2024). Концепція самосуверенної ідентичності як альтернатива традиційним методам автентифікації. У Інформаційна безпека та інформаційні технології ІБІТ 2024: Збірник наукових праць V міжнародної науково-практичної конференції.
6. What is EBSI - EBSI -. (б. д.). European Commission. https://ec.europa.eu/digital-building-blocks/sites/spaces/EBSI/pages/590447955/What+is+EBSI
7. Fdhila, W., Stifter, N., Kostal, K., Saglam, C., & Sabadello, M. (2021). Methods for decentralized identities: Evaluation and insights. У Lecture notes in business information processing (с. 119–135). Springer International Publishing. https://doi.org/10.1007/978-3-030-85867-4_9
8. Mohamed*, M. H., Sobky, W. I. E., & Hamdy, S. (2021). Elliptic curve digital signature algorithm challenges and development stages. International Journal of Innovative Technology and Exploring Engineering, 10(10), 121–128. https://doi.org/10.35940/ijitee.j9433.08101021

9. Sporny, M., Longley, D., Sabadello, M., Reed, D., Steele, O., & Allen, C. (2022, 19 липня). Decentralized identifiers (dids) v1.0. W3C. https://www.w3.org/TR/did-1.0/
10. How IPFS works | IPFS Docs. (б. д.). IPFS Documentation | IPFS Docs. https://docs.ipfs.tech/concepts/how-ipfs-works/#subsystems-overview
11. Rosenbaum, R. (1993, травень). RFC 1464: Using the domain name system to store arbitrary string attributes. IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc1464
12. Balatska, V., Poberezhnyk, V., & Opirskyy, I. (2024). Use of non-fungible tokens and blockchain to demarcate access to public registries. Cybersecurity: Education, Science, Technique, 4(24), 99–114. https://doi.org/10.28925/2663-4023.2024.24.99114
13. Balduf, L., Korczyński, M., Ascigil, O., Keizer, N. V., Pavlou, G., Scheuermann, B., & Król, M. (2023). The cloud strikes back: Investigating the decentralization of IPFS. У IMC '23: ACM internet measurement conference. ACM. https://doi.org/10.1145/3618257.3624797

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023

Vasyl O. Poberezhnyk
PhD student of Information Security Department
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0002-7523-2557
vasyl.poberezhnyk@gmail.com

Taras I. Nakonechnyi
PhD student of Information Security Department
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0009-0003-4487-9424
taras.i.nakonechnyi@lpnu.ua

RESEARCH OF A HYBRID ACCOUNTING DATA MANAGEMENT SYSTEM BASED ON THE COMBINATION OF DID, IPFS, BLOCKCHAIN TECHNOLOGIES

Abstract. This paper examines the concepts of decentralized identifiers and associated credentials, the system for managing such data, and the differences between traditional and decentralized identifiers. In addition, the advantages and disadvantages of decentralized identifiers that already exist are considered and a solution to the challenges facing such identifiers is sought. The challenges facing decentralized identities include the problem of key management, implementation complexity, regulatory uncertainty, cost of operational support, and ease of use. To address these problems, a system concept using a hybrid approach is proposed, which is characterized by using a combination of client-server architecture for storing credentials in IPFS and fingerprints in the blockchain, the immutability of which should protect against the use of fake credentials. Methods for binding IPFS data to a persistent identifier that can point to different data are also compared: IPNS, DNSLink, ENS. Based on this, we developed the ENS-IPFS model of a decentralized identifier and demonstrated a way to integrate such a decentralized model into a hybrid credential management system. During the development of the system, we also developed a mathematical-logical apparatus that allows us to determine whether a credential belongs to a set of reliable ones, and described the algorithm of user authentication through the use of a fingerprint of his credential and solving a cryptographic problem to prove that he owns a private key, on the basis of which a public key was formed, which, in combination with the data identifier, forms a fingerprint that is checked for compliance with what is stored in the blockchain. The results of the study allowed us to confirm the hypothesis about the possibility of using hybrid technologies in the context of using decentralized technologies and credential management. In addition, the data obtained allowed us to form paths for further research and to form a conclusion about the possibility of applying the approach as one that has prospects for parallel application with centralized and decentralized systems and can also play an intermediate link in the transition from a centralized to a decentralized model.

Keywords: DID; IPFS; blockchain; privacy; decentralization; smart contracts; IPNS; DNSLink.

REFERENCES

1. Baklykov, M. I. (2024). Efficiency of using IAM system at the enterprise. Connectivity, 167(1). https://doi.org/10.31673/2412-9070.2024.014851
2. Makhdoom, I., Zhou, I., Abolhasan, M., Lipman, J., & Ni, W. (2020). PrivySharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities. Computers & Security, 88, 101653. https://doi.org/10.1016/j.cose.2019.101653
3. Zaeem, R. N., & Barber, K. S. (2020). The effect of the GDPR on privacy policies. ACM Transactions on Management Information Systems, 12(1), 1–20. https://doi.org/10.1145/3389685
4. Cucko, S., & Turkanovic, M. (2021). Decentralized and self-sovereign identity: Systematic mapping study. IEEE Access, 9, 139009–139027. https://doi.org/10.1109/access.2021.3117588

5. Poberezhnyk, V., Balatska, V., & Opirskyi, I. (2024). The concept of self-sovereign identity as an alternative to traditional authentication methods. In Cybersecurity and information technology. CIT 2024: Collection of scientific papers of V International Scientific and Practical Conference.
6. What is EBSI - EBSI -. (б. д.). European Commission. https://ec.europa.eu/digital-building-blocks/sites/spaces/EBSI/pages/590447955/What+is+EBSI
7. Fdhila, W., Stifter, N., Kostal, K., Saglam, C., & Sabadello, M. (2021). Methods for decentralized identities: Evaluation and insights. У Lecture notes in business information processing (с. 119–135). Springer International Publishing. https://doi.org/10.1007/978-3-030-85867-4_9
8. Mohamed*, M. H., Sobky, W. I. E., & Hamdy, S. (2021). Elliptic curve digital signature algorithm challenges and development stages. International Journal of Innovative Technology and Exploring Engineering, 10(10), 121–128. https://doi.org/10.35940/ijitee.j9433.08101021
9. Sporny, M., Longley, D., Sabadello, M., Reed, D., Steele, O., & Allen, C. (2022, 19 липня). Decentralized identifiers (dids) v1.0. W3C. https://www.w3.org/TR/did-1.0/
10. How IPFS works | IPFS Docs. (б. д.). IPFS Documentation | IPFS Docs. https://docs.ipfs.tech/concepts/how-ipfs-works/#subsystems-overview
11. Rosenbaum, R. (1993, травень). RFC 1464: Using the domain name system to store arbitrary string attributes. IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc1464
12. Balatska, V., Poberezhnyk, V., & Opirskyy, I. (2024). Use of non-fungible tokens and blockchain to demarcate access to public registries. Cybersecurity: Education, Science, Technique, 4(24), 99–114. https://doi.org/10.28925/2663-4023.2024.24.99114
13. Balduf, L., Korczyński, M., Ascigil, O., Keizer, N. V., Pavlou, G., Scheuermann, B., & Król, M. (2023). The cloud strikes back: Investigating the decentralization of IPFS. У IMC '23: ACM internet measurement conference. ACM. https://doi.org/10.1145/3618257.3624797