МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
КАФЕДРА КОМП'ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри комп'ютеризованих
систем захисту інформації
_____ Михайло СТЕПАНОВ
«_____»_____ 2023 р.

На правах рукопису
УДК 004.424:003.26.09

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ВИЩОЇ ОСВІТИ
ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Програмний модуль захисту даних у хмарному сховищі на базі технології блокчейн

Виконавець: Олена ДЯДЮК
Керівник: к.т.н., доцент Іван ПАРХОМЕНКО
Консультант розділу «Охорона навколишнього середовища»: к.т.н., доцент Тетяна ДМИТРУХА
Нормоконтролер: к.т.н., доцент Іван ПАРХОМЕНКО

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет: Кібербезпеки та програмної інженерії

Кафедра: Комп'ютеризованих систем захисту інформації

Освітній ступінь: Магістр

Спеціальність: 125 «Кібербезпека»

Освітньо-професійна програма: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютеризованих систем захисту інформації

_____ Михайло СТЕПАНОВ

«__» _____ 20__ р.

ЗАВДАННЯ

на виконання кваліфікаційної роботи

здобувача вищої освіти Дядюк Олени Віталіївни

1. Тема: *Програмний модуль захисту даних у хмарному сховищі на базі технології блокчейн* затверджена наказом ректора від «15» вересня 2023 р. №1814/ст.
2. Термін виконання: з 16.10.2023 р. по 22.12.2023 р.
3. Вихідні дані: дослідження різних типів хмарних сховищ, загроз інформації, яка в них зберігається, методів захисту даних в хмарах; спосіб використання технології блокчейн з метою захисту інформації; технологія Ethereum; реалізація та дослідження програмного модуля хмарного сховища, яке захищене з використанням технології блокчейн.
4. Зміст пояснювальної записки: хмарні сховища, їх класифікація, загрози збереженим в них даним; технологія блокчейн, та її використання з метою захисту інформації; реалізація програмного модуля хмарного сховища на базі технології блокчейн; оточуюче середовище.

КАЛЕНДАРНИЙ ПЛАН

виконання кваліфікаційної роботи

№ п/п	Етапи виконання кваліфікаційної роботи	Термін виконання етапів	Примітка
1.	Уточнення постановки задачі	16.10.2023	*Виконано*
2.	Аналіз літературних джерел	22.10.2023	*Виконано*
3.	Обґрунтування вибору рішення	24.10.2023	*Виконано*
4.	Збір інформації	30.10.2023	*Виконано*
5.	Аналіз хмарних сховищ різних типів, видів загроз даним, які в них зберігаються	6.11.2023	*Виконано*
6.	Дослідження методів захисту інформації, яка зберігається у хмарних сховищах	14.11.2023	*Виконано*
7.	Аналіз технології блокчейн	16.11.2023	*Виконано*
8.	Реалізація та дослідження власного програмного модуля хмарного сховища з використанням технології блокчейн	25.11.2023	*Виконано*
9.	Оформлення презентації	27.11.2023	*Виконано*
10.	Перевірка на антиплагіат	13.12.2023	*Виконано*
11.	Оформлення і друк пояснювальної записки	15.12.2023	*Виконано*
12.	Отримання рецензій	18.12.2023	*Виконано*
13.	Підготовка до захисту	21.12.2023	*Виконано*

Консультанти з окремих розділів

Розділ	Консультант	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона навколишнього середовища	Дмитруха Т. І.		

Дата видачі завдання: «16» жовтня 2023 р.

Здобувач вищої освіти О. В. Дядюк

(підпис, дата)

Керівник кваліфікаційної роботи І. І. Пархоменко

(підпис, дата)

РЕФЕРАТ

Кваліфікаційна робота на тему «Програмний модуль захисту даних у хмарному сховищі на базі технології блокчейн» складається зі вступу, основної частини, яка містить 4 розділи із висновками до кожного, загального висновку та списку використаної літератури. Загальний обсяг роботи – 66 сторінок. Робота містить 26 рисунків та 3 таблиці. Список використаних джерел включає 25 джерел.

Метою кваліфікаційної роботи є реалізація програмного модуля хмарного сховища, яке захищене з використанням технології блокчейн.

У основній частині кваліфікаційної роботи розглянуті хмарні сховища, їх основні типи, загрози даним, які в них зберігаються, основні методи захисту інформації в хмарах, технологія блокчейн.

Проведені дослідження базуються на сучасних методах реалізації захищених хмарних сховищ, методах шифрування даних, використанні технології блокчейн для захисту інформації.

Реалізація програмного модуля хмарного сховища з смарт-контрактом на блокчейні Ethereum з використанням AES шифрування з 256-розрядним ключем є досить актуальною сьогодні, оскільки хмарні технології широко використовуються у різних галузях життя, і безпека інформації в хмарах є головним завданням.

Програмна реалізація дозволяє забезпечити цілісність та конфіденційність інформації у хмарних сховищах.

Ключові слова: хмарне сховище, центр обробки даних, масштабованість, надлишковість, симетричний алгоритм блочного шифрування, технологія блокчейн, смарт-контракт.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

IT	–	інформаційні технології;
ПК	–	персональний комп'ютер;
ОС	–	операційна система;
API	–	Application User Interface – програмний інтерфейс додатку;
AES	–	Advanced Encryption Standard – симетричний алгоритм блочного шифрування;
SHA	–	Secure Hash Algorithm – алгоритм криптографічного хешування;
U2F	–	Universal 2nd Factor – протокол для двофакторної аутентифікації;
OTP	–	One Time Password – одноразовий пароль;
VPN	–	Virtual Private Network – віртуальна приватна мережа;
VLAN	–	Virtual Local Area Network – віртуальна локальна комп'ютерна мережа;
SSH	–	Secure Shell – мережевий протокол рівня застосунків;
TLS	–	Transport Layer Security – криптографічний протокол, що надає можливості безпечної передачі даних в Інтернеті;
HTTPS	–	Hyper Text Transfer Protocol Secure – захищений протокол передачі гіпертексту;
npm	–	Node Package Manager – менеджер пакунків для мови програмування JavaScript;
SPA	–	Single Page Application – односторінковий веб-застосунок.

ВСТУП

Актуальність теми. У сучасному світі, зі зростанням розвитку інформаційних технологій, стає все більш актуальним завдання збереження великих обсягів даних. Хмарні сховища все частіше використовуються як альтернатива зовнішнім носіям інформації, і цей метод збереження даних стає вибором як для окремих користувачів, так і для цілих підприємств.

Використання хмарних технологій стає зручним методом організації роботи в різних компаніях, оскільки вони надають можливість отримувати цілодобовий доступ до даних не лише з робочого ПК, а й з домашнього або навіть мобільного телефона.

Зрозуміло, зі збільшенням популярності хмарних сховищ серед користувачів, набуває важливості питання безпеки збережених даних і захисту їх від несанкціонованого доступу.

Оскільки в хмарах може бути збережений великий обсяг важливої інформації, зловмисники все частіше спрямовують свої атаки на ці ресурси. Потрапивши до даних і здійснивши їх крадіжку або спотворення, може бути завдана серйозна шкода як користувачам, так і підприємствам.

З метою захисту конфіденційних даних, провайдери хмарних послуг впроваджують нові алгоритми, методи та засоби, такі як шифрування, захист даних при передачі, аутентифікація і інші. Проте користувачам також важливо не забувати про безпеку та дотримуватися певних правил.

Мета роботи: реалізація програмного модуля хмарного сховища, яке захищене завдяки використанню технології блокчейн.

Щоб досягнути цієї мети необхідним є розв’язання таких **задач:**

- аналіз характеристик хмарних сховищ, основних видів загроз, сучасних засобів захисту даних у хмарному середовищі, дослідження технології блокчейн та її застосування;
- реалізація програмного модуля хмарного сховища мовою JavaScript з

використанням фреймворків vue.js, bootstrap.js та смарт-контракту мовою Solidity на базі блокчейна Ethereum;

- тестування програмного модуля хмарного сховища, його функціоналу, перевірка забезпечення цілісності і конфіденційності збережених в хмарі даних.

Об'єкт дослідження: процес захисту даних, які знаходяться у хмарному сховищі.

Предмет дослідження: засоби та механізми для забезпечення захисту даних, що знаходяться у хмарі.

Галузь застосування: даний програмний модуль хмарного сховища на базі технології блокчейн може знайти застосування у сфері інформаційної безпеки, особливо для підвищення рівня захисту даних, зберігаючи їх у хмарових сервісах.

Новизна полягає у реалізації програми з використанням сучасної та найнадійнішої технології – блокчейн, за допомогою смарт-контракта, для створення безпечного хмарного сховища.

Практична цінність полягає у створенні програмного модуля хмарного сховища на базі блокчейн з написанням смарт-контракту на Ethereum та з використанням AES-шифрування з 256-розрядним ключем, який забезпечує цілісність та конфіденційність інформації.

Апробація: основні положення роботи доповідалися та обговорювалися на конференції X International Conference Information Technology and Implementation (Satellite), 21 листопада 2023 р. та опубліковано статтю та тези доповіді: Diadiuk Olena, Sudnitsyn Mykhailo, "Money laundering and fraud detection: the challenges of blockchain transaction analysis", X International Conference Information Technology and Implementation (Satellite), Nov. 2023.

РОЗДІЛ 1. ХМАРНІ СХОВИЩА, ЇХ КЛАСИФІКАЦІЯ, ЗАГРОЗИ ЗБЕРЕЖЕНИМ В НИХ ДАНИМ

1.1. Хмарне сховище

У сучасному світі, з ростом інформаційних технологій, використання зовнішніх носіїв для зберігання великих обсягів даних стає менш обов'язковим. Замість цього, набуває популярності використання хмарних сховищ. Цей метод зберігання даних зручний як для окремих користувачів, так і для підприємств.

Хмарне сховище представляє собою концепцію зберігання інформації, де цифрові дані, такі як документи, фотографії, відео та інші форми медіа, розміщуються на віртуальних або хмарних серверах. Ця модель дозволяє передавати дані на зовнішні системи зберігання та мати до них доступ за необхідності. Хмарне сховище дає можливість користувачам зберігати важливі дані або медіафайли на віддалених серверах, що належать стороннім розробникам. Користувачі можуть звертатися до цих серверів у будь-який момент через Інтернет. Це сховище, відоме також як допоміжне, обслуговується та управляється постачальником хмарних послуг.

Хмарне сховище імітує функціональність локальних пристроїв зберігання даних, дозволяючи зберігати або отримувати доступ до різних типів контенту в будь-який момент часу з віртуального середовища. Будь-яка інформація, завантажена в хмару, зберігається віддалено в надійних віддалених дата-центрах. Коли користувачі зберігають дані на хмарних серверах, ця інформація переходить під відповідальність стороннього постачальника послуг за певну плату.

Щоб отримати доступ до збережених даних, користувачі автентифікуються за допомогою захищених облікових даних. Після авторизації вони можуть отримувати, змінювати або видаляти свої дані. Хмарні сервіси

зберігання даних пропонують такі функції, як шифрування, резервне копіювання та відновлення даних.

Зберігання, управління та доступ до даних стають динамічнішими та ефективнішими завдяки використанню хмарних сховищ. Воно поєднує в собі заходи безпеки, надмірність і масштабованість для забезпечення доступності та цілісності даних, що зберігаються в різних моделях. Це змінило те, як приватні особи та організації керують інформацією, забезпечивши надійне та гнучке рішення для зберігання даних у цифрову епоху.

Застосування хмарних сховищ має свої позитивні та негативні сторони. Одним з ключових викликів є зменшення негативних аспектів для забезпечення безпечного зберігання важливих даних.

Перевагами хмарних сховищ є:

1) Гнучкість і простота доступу: використання хмарного сховища передбачає, що дані не обмежені одним конкретним місцем. Різні користувачі можуть звертатися до ресурсів, які знаходяться в хмарі, з будь-якого місця та пристрою за їхнім вибором без будь-яких проблем з завантаженням чи встановленням.
2) Підтримка віддаленого керування: хмарне сховище також відкриває можливість для віддаленого управління внутрішніми IT-командами або постачальниками керованих послуг. Вони можуть виправляти неполадки без необхідності фізично знаходитися на конкретному робочому місці, що сприяє швидкому вирішенню проблем.
3) Швидка масштабованість: основною перевагою хмарного сховища є можливість додавати нові ресурси за декілька клацань комп’ютерної миші. В умовах раптового збільшення обсягів даних ця можливість сприяє безперебійності бізнес-процесів.
4) Надлишковість для резервного копіювання: дублювання та розміщення тих самих даних в кількох місцях є важливими для ефективного механізму створення резервних копій. Хмарні технології забезпечують безпеку даних у віддалених місцях, щоб запобігти можливим стихійним

лихам, аваріям або кібератакам.

5) Довгострокова економія витрат: у перспективі на довгостроковий період хмарне сховище може значно зменшити витрати на апаратне забезпечення, приміщення для зберігання, енергопостачання та персонал, що неминуче збільшуються разом з розвитком організації.

Недоліками хмарних сховищ є:

1) Ризик блокування постачальника: якщо всі дані зберігаються на одній загальнодоступній хмарній платформі, існує ризик того, що постачальник може заблокувати доступ. Це можна вирішити, використовуючи гібридний або мультихмарний план, де встановлюється достатня взаємодія між різними середовищами для уникнення таких обмежень.
2) Проблеми безпеки, пов'язані з використанням хмари декількома користувачами: загальнодоступні хмарні середовища використовуються декількома клієнтами, що може підвищити ризик вразливості у сфері безпеки. Цьому можна запобігти, захищаючи хмарні дані та використовуючи приватну хмару для збереження конфіденційної інформації.
3) Ризик збою та простою: хмарні платформи, що керуються зовнішніми постачальниками, можуть стикнутися збоями через відмову, що може призвести до недоступності даних та програм, розміщених в цих середовищах. Угоди про рівень обслуговування повинні встановлювати показники надійності, і також важливо використовувати додаткову надлишковість для найважливіших даних.
4) Фрагментація IT-ландшафту: використання хмарного сховища без попереднього плану може призвести до поступового розпаду IT-ландшафту, тому необхідно розробити детальний стратегічний план, який включає короткострокові, середньострокові та довгострокові плани використання хмарних технологій.

Хмарне сховище виконує функції віртуального центру обробки даних,

надаючи кінцевим користувачам та додаткам інфраструктуру віртуального сховища, яку можна масштабувати відповідно до потреб програми. Зазвичай воно функціонує за допомогою веб-API, яке реалізовано віддалено через взаємодію з внутрішньою інфраструктурою хмарного сховища.

Рис.1.1. Принцип функціонування хмарного сховища

Хмарне сховище включає принаймні один сервер даних, доступний користувачеві через мережу Інтернет. Користувач завантажує файли на цей сервер даних, який подальше розподіляє повідомлення на кілька серверів, виконуючи це або вручну, або автоматизовано. Після цього дані можна отримати через веб-інтерфейс.

Хмарні сервери – це віртуальні (не фізичні) сервери, що працюють у хмарному обчислювальному середовищі, до яких може отримати доступ необмежена кількість користувачів за запитом. Хмарні сервери працюють так само, як і фізичні сервери, і виконують подібні функції, такі як зберігання даних і запуск додатків. Хмарні сервери, як правило, доступні через Інтернет і використовуються для широкого спектру додатків, таких як веб-сервери, бази даних, файлові сервери та додатки "програмне забезпечення як послуга" (SaaS). Оскільки хмарні сервери є віртуальними, їх можна легко створювати, масштабувати та видаляти за потреби, що робить їх дуже адаптивними та економічно ефективними.

Хмарні системи зберігання даних зазвичай базуються на сотнях серверів даних. З огляду на те, що комп'ютери періодично потребують технічного обслуговування або ремонту, важливо зберігати інформацію на кількох машинах, щоб забезпечити надмірність. Без такого резервування хмарна система зберігання даних не могла б гарантувати клієнтам можливість доступу до своєї інформації в будь-який момент. Більшість систем зберігають однакові дані на серверах, які живляться від різних джерел.

1.2. Класифікація хмарних сховищ

Хмарні сховища дозволяють користувачам зберігати цифрові дані та отримувати до них доступ з будь-якого місця через мережу Інтернет. З іншого боку, хмарне сховище можна розглядати як віртуальний жорсткий диск. Ця система виявляється універсальною, оскільки її можна використовувати як для збереження важливих даних, таких як текстові документи і відеофайли, так і для обробки складних даних та запуску програм через доступ до хмари. Для визначення оптимального вибору хмарного сховища користувачеві слід спочатку визначити свої конкретні потреби та варіанти використання.

Рис.1.2. Типи хмарних сховищ

Приватне хмарне сховище, відоме також як корпоративне або внутрішнє хмарне сховище, є формою зберігання даних, де інформація розташована в інтранеті компанії чи організації і захищена власним брандмауером компанії. Воно є вигідним вибором для компаній із дорогими центрами обробки даних, оскільки може самостійно керувати конфіденційністю даних. Основною перевагою використання приватного хмарного сховища є повний контроль для користувача. З іншого боку, однією з головних недоліків такого підходу є витрати на обслуговування та регулярні оновлення.

Публічне хмарне сховище включає кілька адміністративних елементів управління і може бути доступне онлайн для користувача та будь-якої іншої особи, яку користувач авторизує. Користувачу або компанії, які використовують публічне хмарне сховище, не потрібно здійснювати обслуговування системи. Такі сховища надаються різними постачальниками послуг, такими як Amazon Web Services (AWS), IBM Cloud, Google Cloud і Microsoft Azure. Публічне хмарне сховище легко масштабується, доступне за доступною ціною, надійне і забезпечує неперервний моніторинг і мінімальне технічне обслуговування.

Гібридне хмарне сховище представляє собою поєднання приватного і публічного хмарних сховищ. Цей тип сховища, як вказує його назва, пропонує користувачеві оптимальні характеристики обох типів – безпеку приватної хмари та персоналізацію публічної хмари. В гібридному хмарному сховищі можна зберігати дані в приватній хмарі, при цьому завдання обробки інформації можна віддавати на обробку в публічну хмару за допомогою хмарних обчислень. Гібридне хмарне сховище є доступним за ціною, пропонує легкість налаштування та зручний контроль користувача.

Спільне хмарне сховище є варіантом приватного хмарного сховища, спрямованого на надання хмарних рішень для конкретних компаній або спільнот. У цій моделі постачальники хмарних сховищ предоставляють свою хмарну архітектуру, програмне забезпечення та інші інструменти розробки, щоб відповідати унікальним вимогам спільноти. Дані зберігаються у приватному хмарному сховищі спільноти для відповіді на конкретні потреби щодо безпеки

та відповідності. Спільне хмарне сховище є оптимальним рішенням для медичних, фінансових або юридичних компаній, які строго дотримуються внутрішніх політик.

Крім того, хмарні сховища можна класифікувати за способом зберігання даних на три основні категорії:

1) об’єктне сховище;
2) файлове сховище;
3) блочне сховище.

Об’єктне хмарне сховище надає можливість зберігання різноманітних даних протягом тривалого періоду, спрощує доступ до інформації і ідеально підходить для неструктурованих, двійкових або BLOB-об’єктів. Цей вид сховища легко адаптується до частих відмов компонентів системи завдяки постійному моніторингу, виявленню помилок стійкості та автоматичному відновленню. Об’єктне сховище може вміщувати об'ємні набори даних та файли.

У файловому хмарному сховищі дані зберігаються у вигляді файлів, які впорядковуються та розташовуються у папках, створюючи каталоги, підкаталоги та інші структури. Файли в сховищі файлів можна легко перейменовувати, видаляти або налаштовувати без необхідності використання додаткових інтерфейсів. Файлове хмарне сховище ідеально підходить для зберігання неструктурованих даних або файлів, які використовуються спільно. Однією зі значущих переваг цього типу сховища є його простота в управлінні. Крім того, файлове сховище полегшує обмін файлами і спільну роботу.

У блочному хмарному сховищі дані розбиваються на блоки та зберігаються в системі, яку можна фізично розподілити. Кожен блок має свій унікальний ідентифікатор, що дозволяє системі відстежувати та збирати їх за необхідності. Цей тип сховища характеризується низькою затримкою та високою продуктивністю, і він ідеально підходить для роботи з віртуальними машинами. Блочна система зберігання використовується в тих випадках, коли необхідне швидке отримання та маніпулювання даними.

Фізичне розташування хмарного сервера може вплинути на продуктивність резервного копіювання та відновлення. Тому це один із ключових пріоритетів, який слід враховувати при виборі правильного хмарного сервісу для резервного копіювання.

Якщо хмарний сервер розташований далеко від основного місця розташування, швидкість передачі даних до/з хмарного сервера може бути низькою. Якщо хмарний сервер розташований занадто близько до основного офісу, то будь-які стихійні лиха, такі як землетруси, повені або перебої в електропостачанні, можуть порушити роботу бізнесу, що призведе до втрати даних, часу та доходу. Тому рішення про місце розташування має ґрунтуватися на важливості даних, типі можливих стихійних лих і вартості.

Крім того, деякі компанії можуть мати регуляторні вимоги до місць зберігання даних. Такі організації повинні ретельно проаналізувати свої вимоги і вибрати хмарний сервіс, який дозволяє передавати і зберігати дані в авторизованих місцях, схвалених компанією.

1.3. Загрози даним, які зберігаються у хмарних сховищах

Хмарне сховище забезпечує зручний спосіб зберігання інформації та надає користувачам можливість отримувати доступ до своїх даних в будь-якому місці та у будь-який час, на практично будь-якому пристрої. Отже, забезпечення безпеки хмарного сховища стає однією з ключових задач.

Корпорації та бізнеси активно використовують хмарні послуги через їх економічну ефективність та гнучкі альтернативи в порівнянні з вартістю та локальним обладнанням. Однак перехід до бізнесу в хмарі призводить до виникнення нових ризиків для конфіденційних файлів та інформації, оскільки дані, збережені в хмарі, виходять за рамки багатьох традиційних заходів безпеки, які використовуються для захисту конфіденційних даних на локальних серверах.

Отже, для забезпечення безпеки хмарного сховища підприємства повинні вживати додаткових заходів безпеки, на доповнення до базових заходів, які надають постачальники. Незважаючи на це, переваги хмарного сховища, такі як масштабованість і доступність, разом із зменшенням витрат на IT, сприяють швидкому впровадженню цієї технології на підприємствах.

При виборі рішення для захисту хмарного сховища, підприємства повинні переконатися, що обрана система надає постійний моніторинг та видимість для всіх взаємодій даних з додатками хмарного сховища. Вона також повинна гарантувати детальний контроль над переміщенням файлів, використовуючи події браузера та операційної системи, включаючи спільний доступ до файлів і сайтів хмарного сховища. Рішення повинно інтегруватися з провідними постачальниками хмарних сховищ, щоб розширити заходи захисту даних, забезпечити автоматичне шифрування конфіденційних даних перед передачею, а також точно класифікувати дані, які завантажуються з веб-додатків.

Оскільки 96% компаній повідомляють про недостатній рівень захисту конфіденційних хмарних даних, питання безпеки хмарних обчислень не можуть залишатися другорядними.

Розвиток хмарних сховищ уможливив значну частину стрімкої цифрової трансформації, яку люди відчули на собі з початком пандемії. За прогнозами, до 2029 року світовий ринок хмарних сховищ, який сьогодні становить 83,41 мільярда доларів, зросте до 376,37 мільярда доларів.

Хоча хмарні сховища відкривають багато можливостей для оптимізації процесів і підвищення продуктивності, ми також повинні пам'ятати про серйозні ризики, які вони несуть.

Ці ризики пов'язані не стільки з технологією, скільки з пов'язаною з нею поведінкою. Зі збільшенням кількості сервісів, які інтегруються з цими хмарними накопичувачами, зростає ймовірність порушень безпеки.

Так само, оскільки віддалена робота стала звичним явищем, багато лідерів у сфері безпеки висловлюють занепокоєння щодо сплеску небезпечних звичок, таких як збереження файлів в особистих і корпоративних хмарах без дозволу.

Щоб з'ясувати, як забезпечити надійний захист для даних, які зберігаються у хмарних сховищах, розглянемо основні загрози та проведемо короткий аналіз.

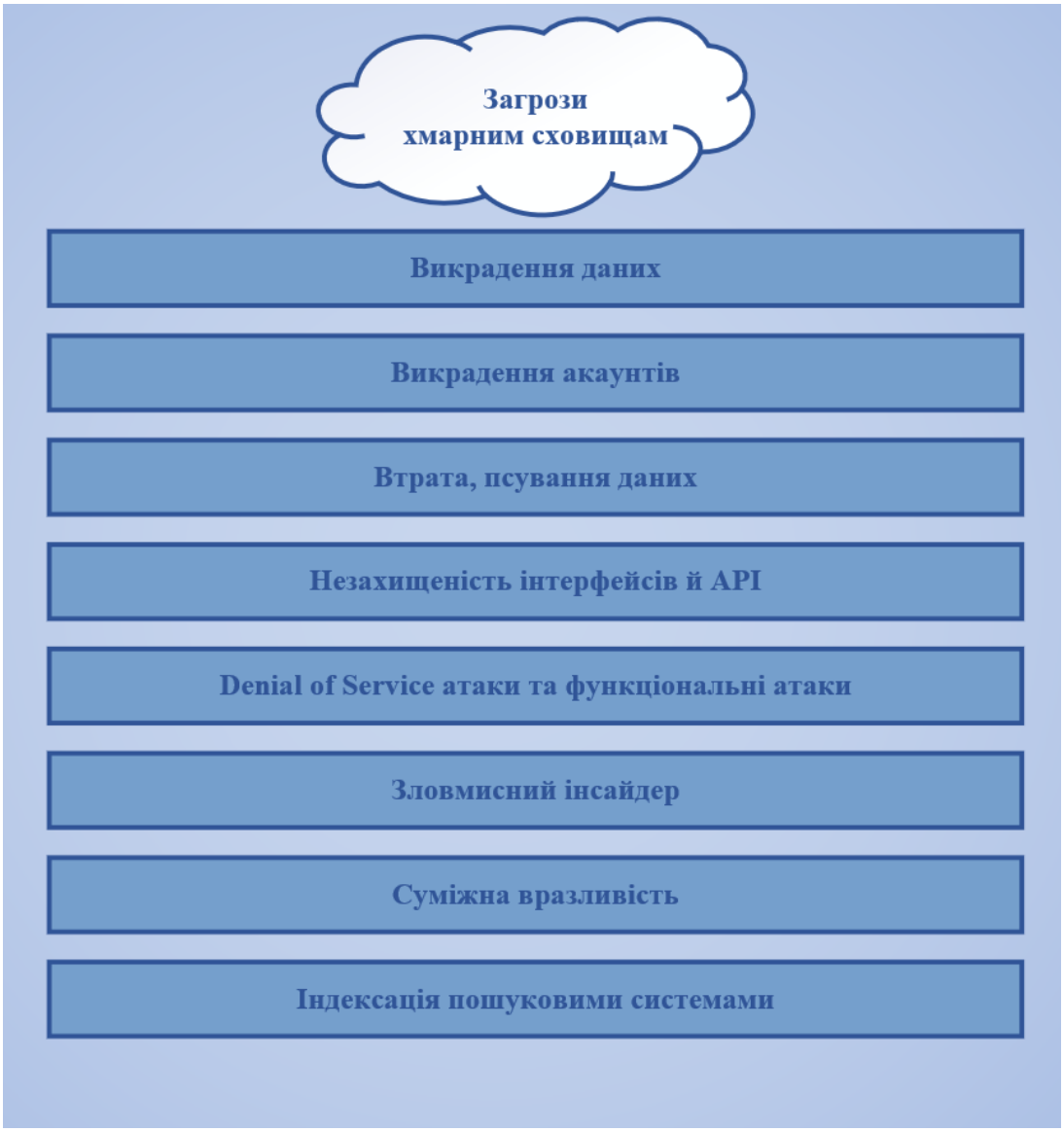

Рис.1.3. Загрози, які виникають при використанні хмарних сховищ

Детальніше розглянемо сутність кожної з позначених на рис.1.3 загроз, які виникають при використанні хмарних сховищ.

Викрадення даних: часто зустрічаються напади на сервер хмарного сховища з метою отримання електронних адрес користувачів, які потім використовуються для проведення фішингових атак на їхні поштові скриньки.

Викрадення акаунтів: часто стається вторгнення в обліковий запис, і до виявлення його власника зловмисники мають повний контроль над збереженими в хмарі даними, здатні перехоплювати, спотворювати або видаляти інформацію.

Втрата, псування даних: внаслідок системних помилок, недбалості користувачів, недосконалості програмного забезпечення хмарних сховищ, фізичного знищення серверів чи дата-центрів збережені дані можуть бути спотворені або повністю втрачені.

Незахищеність інтерфейсів й API: хмарні сховища стають уразливими до кібератаками у випадках, коли відсутні належні заходи захисту в системах управління контейнерами та API-інтерфейсах.

Denial of Service атаки та функціональні атаки: завдяки своїй багатошаровій структурі хмарне сховище може стати уразливим до атак "відмова в обслуговуванні", які викликають перевантаження інфраструктури та спричиняють використання великого обсягу системних ресурсів..

Зловмисний інсайдер: можлива атака на дані відбувається всередині хмарного сховища, що означає, що зловмисник, який має відповідні права доступу до системи, здійснює цей напад.

Суміжна вразливість: після того, як ресурс був скомпрометований, зловмисник може використовувати шкідливі сценарії для здійснення атак на всіх, хто має доступ до цього ресурсу. Це призводить до виникнення вторинної вразливості.

Індексація пошуковими системами: через широкий функціонал пошукових систем браузерів та передовим веб-технологіям, а також через спрямовані дії чи програмні помилки, інформація, збережена в хмарних сховищах, може стати доступною для загального використання.

За останні часи, коли підприємства все активніше використовують хмарні системи, безпека хмарного сховища стає важливою проблемою як в IT-архітектурі компаній, так і в стратегіях інформаційної безпеки. Організації розпізнають, наскільки критично захищати конфіденційні дані, при цьому надаючи співробітникам можливість використовувати продуктивність та гнучкість хмарних рішень.

Перехід на платформи хмарного зберігання даних призводить до підвищення продуктивності організації та надійності продукту. Проте для цього

також важлива потужна стратегія безпеки в хмарному сховищі, яка включає в себе вибір оптимальних інструментів, алгоритмів, політик, методів і стандартних рекомендацій. Обрання комплексного інструменту захисту даних у хмарі покращить рівень безпеки, зменшить ризики та гарантує захист від різноманітних порушень.

1.4. Аналіз методів захисту інформації в хмарних сховищах

Для вивчення аспектів захисту даних у хмарних сховищах розглянемо термін "безпека даних у хмарному сховищі". Це, як і в інших галузях кібербезпеки, охоплює різноманітні технології, практики та політики, спрямовані на захист даних від несанкціонованого доступу та атак, а також забезпечення їх цілісності. Безпека даних у хмарі вирішує питання збереження інформації, розташованої поза власним центром обробки даних або комп'ютером. Основними аспектами є захист від шкідливих програм, DDoS-атак, несанкціонованого доступу та інших загроз, а також запобігання витоку даних у віртуалізованих середовищах, забезпечення аварійного відновлення та безперервності бізнес-процесів. Хмарні сервіси відповідають вимогам щодо рівня обслуговування (SLA) та вимагають керівництва кваліфікованими експертами.

Функція хмарних служб безпеки полягає в забезпеченні неперервного захисту інформації. Постачальники хмарних послуг обмежують несанкціонований доступ, застосовуючи шифрування для захисту збережених в хмарі даних та надаючи різноманітні засоби контролю доступу. Крім того, вони пропонують можливості відновлення та резервного копіювання даних у випадку їх втрати.

Щоб гарантувати захист даних, збережених в хмарних системах, користувачам і підприємствам слід дотримуватися певних правил, окрім заходів, які приймають провайдери хмарних послуг. Розглянемо кілька таких правил:

1) Найбезпечніший варіант – це хмарний постачальник, який має центри обробки даних, розташовані географічно поруч із користувачем, і дає можливість чітко визначити, де будуть зберігатися дані. Це гарантує безперебійність бізнесу, ефективність та забезпечує переваги з точки зору правової безпеки.
2) Варто обмежити вибір країн для зберігання даних у тих випадках, коли вони не надають належного значення правилам щодо збереження даних або відомо, що в них відсутні ефективні методи безпеки.
3) Важливо перевірити тип підключення, шлюз та послуги брандмауера, які надає хмарний постачальник. Також варто оцінити наявність шифрування даних як у спокійному стані, так і під час передачі. Слід звернути увагу на ключові аспекти, такі як шифрування AES, SSL/TLS, методи автентифікації та авторизації, розподіл ролей та використання протоколу HTTPS.
4) Для захисту даних важливо враховувати концепцію наскрізного шифрування (E2EE). Завдяки E2EE можна бути впевненим у безпеці даних, навіть у випадку порушення безпеки у постачальника хмарних послуг. Кожен завантажений файл буде залишатися зашифрованим.
5) Зберігайте лише ті файли, до яких ви часто звертаєтеся, і не зберігайте документи, що містять паролі, номери соціального страхування, номери кредитних карток тощо.
6) Пароль – це перша лінія захисту доступу до інформації компанії. Однак найкращий у світі захист не допоможе, якщо зловмисник отримає доступ до ваших імен користувачів і паролів. Окрім вибору надійного та унікального пароля, переконайтеся, що ви часто змінюєте його і не повторюєте в інших онлайн-акаунтах. Надійні паролі ускладнюють хакерські атаки на ваші дані і знижують ймовірність того, що хмара

піддасться небажаному вторгненню.

Постачальники хмарних послуг також працюють над зменшенням ризиків втрат та несанкціонованого доступу до конфіденційної інформації користувачів за допомогою впровадження різноманітних алгоритмів та методів захисту даних.

Рис.1.4. Найпопулярніші методи захисту даних в хмарному сховищі

Шифрування на серверах хмарних сховищ означає, що дані зберігаються виключно у зашифрованому вигляді, і їхній ідентифікатор відомий лише за метаінформацією. Ключі знаходяться на окремих серверах управління ключами, і взаємодія між даними та ключами відбувається на стороні користувача. Для забезпечення захисту використовуються різні методи, такі як асиметричне шифрування, Advanced Encryption Standard (AES) від 128 до 2048 біт, RSA, а також хешування за допомогою Secure Hash Algorithm (SHA).

Аутентифікація – це перевірка того, чи особа є власником даних, і, зазвичай, вона включає в себе введення пароля або відповідь на контрольне запитання. Двофакторна аутентифікація, особливо з використанням бездротового протоколу Universal 2nd Factor (U2F) та одноразових паролів One Time Password (OTP), є ефективними методами забезпечення безпеки.

Ізоляція користувачів включає в себе використання окремих віртуальних машин та мереж для кожного користувача. Це може включати в себе такі засоби,

як VPN-тунелі та VLAN-мережі.

Для захисту даних під час їх передачі використовують приватні мережі та протоколи, щоб забезпечити, що несанкціоновано перехоплені дані не матимуть значення для особи без ключів шифрування. Наприклад, це може бути забезпечено за допомогою протоколів, таких як Secure Shell (SSH), Transport Layer Security (TLS) та HyperText Transfer Protocol Secure (HTTPS).

Порівнюючи сучасних постачальників хмарних сховищ, можна визначити, що їхні спектри послуг досить схожі. Наразі вибір хмарного сховища частіше здійснюється на основі цінового діапазону. Однак також важливо приділяти увагу рівню підтримки, яку надають ці постачальники. При виборі провайдера хмарного сховища важливо враховувати підтримку платформ для використання (Windows, Mac, iPhone, Android, телефони BlackBerry та інші). Для зменшення витрат можна розглядати SaaS, що дозволяє знизити вартість ліцензування. Якщо хмарне сховище не надає наскрізне шифрування, рекомендується перед перенесенням даних до хмари зашифрувати їх спочатку, щоб підвищити рівень безпеки.

В таблиці 1.1 коротко описані на сьогоднішній день одні з найпопулярніших методів захисту даних у хмарних сховищах.

Таблиця 1.1.

Порівняння методів захисту даних у найпопулярніших хмарних сховищах

Хмарне сховище	Методи захисту даних
Google Cloud	Криптозахист, захист при передачі даних
iCloud	Захист AES 128, захист даних при передачі TLS 1.2, аутентифікація U2F
Dropbox	Аутентифікація, захист при передачі даних, криптозахист AES 256 на сервері і для контрактних акаунтів
OneDrive	Криптозахист AES 256, двофакторна аутентифікація
SpiderOak	Криптозахист, аутентифікація U2F

1.5. Висновки до першого розділу

Підсумовуючи інформацію, представлену у першому розділі, важливо відзначити, що на сучасний момент хмарні сховища стали зручним рішенням для зберігання інформації як для окремих користувачів, так і для підприємств. Цей метод використання отримав широкий розповсюдження у всьому світі, і в даний час головним завданням є забезпечення належного захисту важливих даних, що зберігаються в хмарних сховищах.

Хмарне сховище представляє собою модель зберігання цифрової інформації, де дані розміщуються на віртуальних або хмарних серверах, що належать третім сторонам. Це дозволяє передавати дані на зовнішню інфраструктуру та отримувати до них доступ за необхідності.

Існує розподіл хмарних сховищ на приватні, публічні, гібридні та спільні, а також класифікація за типом зберіганих даних: об’єктне, файлове та блочне сховище.

Через широке поширення хмарних сховищ, вони стають об'єктом інтересу для зловмисників, які прагнуть отримати доступ до конфіденційної інформації, перехопити її або спотворити. Крім того, випадки викрадення інформації користувачів, зокрема їхніх електронних адрес, для подальших фішингових атак не є рідкістю.

На даному етапі важливо розглядати питання захисту інформації, яка зберігається у хмарних сховищах. Цю задачу слід розглядати як з точки зору користувачів (підприємств), так і постачальників хмарних послуг.

Користувачам важливо ретельно обирати постачальника хмарних послуг, звертаючи увагу не лише на його цінові пропозиції, але й на обсяг наданих послуг. Якщо в обраному хмарному сховищі відсутнє шифрування даних, можна вжити власних заходів щодо цього питання.

Для захищеного збереження даних постачальники зазвичай застосовують шифрування, аутентифікацію, захист даних при передачі, ізоляцію користувачів.

РОЗДІЛ 2. ТЕХНОЛОГІЯ БЛОКЧЕЙН, ТА ЇЇ ВИКОРИСТАННЯ З МЕТОЮ ЗАХИСТУ ІНФОРМАЦІЇ

2.1. Поняття блокчейн

Блокчейн – це розподілена база даних або реєстр, до якого мають доступ вузли комп'ютерної мережі. Вони найбільш відомі завдяки своїй вирішальній ролі в криптовалютних системах для підтримання безпечного та децентралізованого обліку транзакцій, але їхнє застосування не обмежується криптовалютами. Блокчейн можна використовувати для того, щоб зробити дані в будь-якій галузі незмінними – термін, який використовується для опису неможливості їх зміни.

Оскільки змінити блок неможливо, довіра потрібна лише в момент, коли користувач або програма вводить дані. Цей аспект зменшує потребу в довірених третіх сторонах, які зазвичай є аудиторами або іншими людьми, що збільшують витрати і припускаються помилок.

З моменту появи Bitcoin у 2009 році використання блокчейну стрімко зросло завдяки створенню різних криптовалют, додатків для децентралізованих фінансів (DeFi), невзаємозамінних токенів (NFT) та смарт-контрактів.

Ви можете бути знайомі з електронними таблицями або базами даних. Блокчейн дещо схожий на них, оскільки це база даних, в яку вводиться і зберігається інформація. Але ключова відмінність між традиційною базою даних або електронною таблицею і блокчейном полягає в тому, як структуровані дані та доступ до них.

Блокчейн складається з програм, які називаються скриптами, що виконують завдання, які зазвичай виконуються в базі даних: вводиться інформація та отримується до неї доступ, а також зберігається. Блокчейн є розподіленим, а це означає, що декілька копій зберігаються на багатьох машинах, і всі вони повинні збігатися для того, щоб він був дійсним.

Блокчейн збирає інформацію про транзакції і вносить її в блок, як комірку в електронній таблиці, що містить інформацію. Коли блок заповнений, інформація проходить через алгоритм шифрування, який створює шістнадцяткове число, що називається хеш.

Потім хеш вводиться в заголовок наступного блоку і шифрується з іншою інформацією в блоці. Таким чином створюється серія блоків, які з'єднуються ланцюжком.

Транзакції проходять певний процес, залежно від того, в якому блокчейні вони відбуваються. Наприклад, у блокчейні Bitcoin, якщо ініціюється транзакція за допомогою криптовалютного гаманця – додатку, який надає інтерфейс для блокчейну, це запускає послідовність подій.

У Bitcoin транзакція надсилається до пулу пам'яті, де вона зберігається в черзі, доки її не підхопить майнер або валідатор. Як тільки вона потрапляє в блок і блок заповнюється транзакціями, він закривається і шифрується за допомогою алгоритму шифрування. Після цього починається майнінг.

Вся мережа працює одночасно, намагаючись "розгадати" хеш. Кожен з них генерує випадковий хеш, за винятком "nonce", скорочення від числа, яке використовується один раз.

Кожен майнер починає з нульового nonce, який додається до випадково згенерованого хешу. Якщо це число не дорівнює або менше цільового хешу, до nonce додається значення одиниці, і генерується новий хеш блоку. Так триває до тих пір, поки майнер не згенерує правильний хеш.

Після закриття блоку транзакція вважається завершеною. Однак блок не вважається підтвердженим, поки не буде підтверджено п'ять інших блоків. Підтвердження займає у мережі близько години, оскільки в середньому на один блок витрачається трохи менше 10 хвилин (перший блок з вашою транзакцією і п'ять наступних блоків, помножені на 10, складають близько 60 хвилин).

Не всі блокчейни дотримуються цього процесу. Наприклад, мережа Ethereum випадковим чином вибирає одного валідатора з усіх користувачів зі

стейком ефіру для перевірки блоків, які потім підтверджуються мережею. Це набагато швидше і менш енергоємно, ніж процес у Bitcoin.

Блокчейн дозволяє розподіляти дані в базі даних між кількома мережевими вузлами – комп'ютерами або пристроями, на яких працює програмне забезпечення для блокчейну – в різних місцях. Це не лише створює надлишковість, але й зберігає достовірність даних. Наприклад, якщо хтось спробує змінити запис в одному екземплярі бази даних, інші вузли не дадуть цьому статися. Таким чином, жоден вузол в мережі не може змінити інформацію, що зберігається в ньому.

Завдяки такому розподілу і зашифрованому доказу того, що робота була виконана, інформація та історія (як і транзакції в криптовалюті) є незворотними. Такий запис може бути списком транзакцій (наприклад, з криптовалютою), але блокчейн також може зберігати різноманітну іншу інформацію, наприклад, юридичні контракти, державні ідентифікатори або інвентаризацію компанії.

Через децентралізовану природу блокчейну, всі транзакції можна прозоро переглядати, або маючи особистий вузол, або використовуючи дослідники блокчейну, які дозволяють будь-кому бачити транзакції, що відбуваються в реальному часі. Кожен вузол має власну копію ланцюжка, яка оновлюється при підтвердженні та додаванні нових блоків. Це означає, що при бажанні можна відстежити транзакцію, куди б вона не пішла.

Наприклад, у минулому відбувалися хакерські атаки на біржі, що призводило до втрати великих сум криптовалюти. Хоча хакери могли бути анонімними – за винятком адреси їхніх гаманців – криптовалюту, яку вони видобули, легко відстежити, оскільки адреси гаманців публікуються в блокчейні.

Блокчейн можна використовувати для незмінного запису будь-якої кількості даних. Це можуть бути транзакції, голоси на виборах, інвентаризація товарів, державні ідентифікатори, документи на житло та багато іншого.

В даний час десятки тисяч проектів прагнуть впровадити блокчейн у різні способи, щоб допомогти суспільству, окрім простого запису транзакцій - наприклад, як спосіб безпечного голосування на демократичних виборах.

Незмінність блокчейну означає, що фальсифікувати голосування стане набагато складніше. Наприклад, система голосування може працювати таким чином, що громадянам кожної країни буде видаватися одна криптовалюта або токен.

2.2. Використання технології блокчейн для забезпечення цілісності і конфіденційності інформації

Основною метою блокчейну є забезпечення можливості запису та поширення цифрової інформації, при цьому заборонено її редагування. Таким чином, ця технологія виступає основою для створення реєстрів, які неможливо змінити, видалити або знищити. Тому блокчейн часто відомий також як технологія розподіленого реєстру (DLT).

Щоб зрозуміти аспекти безпеки використання технології блокчейн, розглянемо приклад. Представимо, що хакер, який контролює вузол у мережі блокчейн, намагається змінити ланцюг та вкрасти інформацію у всіх інших користувачів. Якщо він змінює свою власну копію, вона перестане відповідати іншим копіям. Коли інші користувачі обмінюються своїми копіями, вони помітять, що ця копія вирізняється, і цю хакерську версію відкидають як неправомірну. Для успішної атаки такого роду хакерові необхідно одночасно контролювати та змінювати 51% або більше копій блокчейну, щоб його нова версія стала прийнятою як основна та узгоджена. Здійснення такої атаки вимагає значних фінансових і ресурсних витрат, оскільки хакер повинен буде переробити всі блоки з їхніми відмінними відмітками часу та хеш-кодами.

Але звичайно, як і кожна технологія, блокчейн має свої сильні та слабкі сторони.

У таблиці 2.1 більш конкретизовано представлені основні переваги та недоліки технології блокчейн.

Таблиця 2.1

Порівняння переваг та недоліків технології блокчейн

Переваги технології	Недоліки технології
Підвищена точність за рахунок виключення участі людини в перевірці	Значна вартість технології
Зменшення витрат завдяки виключенню сторонньої перевірки	Низький рівень операцій в секунду
Децентралізація підвищує рівень безпеки системи	Обмеження зберігання даних
Прозора технологія	Регулювання залежить від юрисдикції та залишається невизначеним
Спосіб захисту особистої інформації для громадян країн з слаборозвинутим урядом	Можливість використання в незаконній діяльності

Про переваги технології:

1) Точність ланцюга: транзакції в мережі блокчейн затверджуються тисячами комп'ютерів і пристроїв. Це усуває майже всіх людей від процесу перевірки, що призводить до зменшення людських помилок і точного запису інформації. Навіть якщо комп'ютер у мережі припуститься обчислювальної помилки, вона вплине лише на одну копію блокчейну і не буде прийнята рештою мережі.
2) Зменшення витрат: зазвичай споживачі платять банку за перевірку транзакції або нотаріусу за підписання документа. Блокчейн усуває

необхідність у перевірці третьою стороною, а разом з нею і пов'язані з цим витрати. Наприклад, власники бізнесу сплачують невелику комісію, коли приймають платежі кредитними картками, оскільки банки та платіжні системи повинні обробляти ці транзакції. З іншого боку, біткойн не має центрального органу і має обмежену комісію за транзакції.

3) Децентралізація: блокчейн не зберігає жодної інформації в центральному місці. Замість цього блокчейн копіюється і поширюється по мережі комп'ютерів. Щоразу, коли до блокчейну додається новий блок, кожен комп'ютер у мережі оновлює свій блокчейн, щоб відобразити цю зміну.

   Завдяки тому, що інформація поширюється мережею, а не зберігається в одній центральній базі даних, блокчейн стає складніше підробити.

4) Прозорі транзакції: розрахунки за транзакціями, здійсненими через центральний орган, можуть зайняти до кількох днів. Наприклад, якщо ви спробуєте внести чек у п'ятницю ввечері, ви можете не побачити кошти на своєму рахунку до ранку понеділка. Фінансові установи працюють у робочий час, зазвичай п'ять днів на тиждень, але блокчейн працює 24 години на добу, сім днів на тиждень і 365 днів на рік.

   У деяких блокчейнах транзакції можуть бути завершені за лічені хвилини і вважатися безпечними вже через кілька хвилин. Це особливо корисно для транскордонних торгів, які зазвичай займають набагато більше часу через різницю в часових поясах і той факт, що всі сторони повинні підтвердити обробку платежу.

5) Безпечні транзакції: після того, як транзакція записана, її автентичність має бути перевірена мережею блокчейн. Після того, як транзакція підтверджена, вона додається до блоку блокчейну. Кожен блок у блокчейні містить свій унікальний хеш і унікальний хеш блоку, що стоїть перед ним. Тому блоки не можуть бути змінені після того, як мережа підтвердить їх.

Мабуть, найголовніша особливість блокчейну та криптовалюти – це можливість користуватися ними будь-кому, незалежно від етнічної приналежності, статі, місцезнаходження чи культурного походження. За даними Світового банку, приблизно 1,3 мільярда дорослих не мають банківських рахунків або будь-яких засобів для зберігання своїх грошей чи багатства.

Більше того, майже всі ці люди живуть у країнах, що розвиваються, де економіка перебуває в зародковому стані і повністю залежить від готівки.

Цим людям часто платять фізичною готівкою. Потім їм доводиться зберігати цю готівку в схованках у себе вдома або в інших місцях, що спонукає грабіжників до насильства чи крадіжок. Хоча крадіжка не є неможливою, криптовалюта робить її більш складною для потенційних злодіїв.

Про недоліки технології:

1) Вартість технології: хоча блокчейн може заощадити користувачам гроші на комісіях за транзакції, ця технологія далеко не безкоштовна. Наприклад, система підтвердження роботи мережі Bitcoin для перевірки транзакцій споживає величезні обсяги обчислювальної потужності. У реальному світі енергія, яку споживають мільйони пристроїв у мережі Bitcoin, є більшою, ніж щорічно споживає Пакистан.

   Починають з'являтися деякі рішення цих проблем. Наприклад, були створені ферми з видобутку токенів, які використовують сонячну енергію, надлишковий природний газ з місць гідророзриву пластів або енергію вітрових електростанцій.

2) Швидкість і неефективність даних: Bitcoin є ідеальним прикладом можливої неефективності блокчейну. Система PoW Bitcoin займає близько 10 хвилин, щоб додати новий блок до блокчейну. За такої швидкості, за оцінками, мережа блокчейн може обробляти лише близько трьох транзакцій в секунду (TPS).

Хоча інші криптовалюти, такі як Ethereum, працюють краще, ніж Bitcoin, блокчейн все ще обмежує їх можливості. Застарілий бренд Visa, для порівняння, може обробляти 65 000 TPS.

Вирішення цієї проблеми розроблялося роками. Наразі існують блокчейни, які можуть похвалитися більш ніж 30 000 TPS.

Прогнозується, що злиття Ethereum між основною мережею і ланцюгом-маяком дозволить обробляти до 100 000 TPS після того, як він розгорне серію оновлень, які включають шардінг – розбиття бази даних, щоб більше пристроїв (телефонів, планшетів і ноутбуків) могли працювати з Ethereum. Очікується, що це збільшить участь в мережі, зменшить перевантаження і підвищить швидкість транзакцій.

3) Обмеження зберігання даних: інша проблема полягає в тому, що кожен блок може зберігати лише певну кількість даних. Дискусія про розмір блоку була і залишається однією з найактуальніших проблем масштабування блокчейну в майбутньому.
4) Регулювання: багато хто в криптопросторі висловлює занепокоєння щодо державного регулювання криптовалют. Хоча стає все важче і майже неможливо знищити щось на кшталт Bitcoin, оскільки його децентралізована мережа зростає, уряди теоретично можуть зробити незаконним володіння криптовалютами або участь в їхніх мережах.

   Це занепокоєння з часом зменшилося, оскільки такі великі компанії, як PayPal, почали дозволяти клієнтам використовувати криптовалюти на своїх платформах електронної комерції.
5) Незаконна діяльність: хоча конфіденційність в мережі блокчейн захищає користувачів від зломів і зберігає конфіденційність, вона також дозволяє здійснювати незаконну торгівлю і діяльність в мережі блокчейн. Найвідомішим прикладом використання блокчейну для незаконних транзакцій, ймовірно, є "Шовковий шлях" – онлайн-платформа для торгівлі наркотиками та відмивання грошей, що діяла з лютого 2011 року до жовтня 2013 року.

Темна мережа дозволяє користувачам купувати і продавати нелегальні товари без відстеження за допомогою браузера Tor, а також здійснювати незаконні покупки в біткоїнах та інших криптовалютах. Це різко контрастує з американським законодавством, яке вимагає від постачальників фінансових послуг отримувати інформацію про своїх клієнтів при відкритті рахунків. Вони повинні перевірити особу кожного клієнта і підтвердити, що він не фігурує в жодному списку відомих або підозрюваних терористичних організацій.

Цю систему можна розглядати як "за", так і "проти". Вона дає будь-кому доступ до фінансових рахунків, але дозволяє злочинцям легше здійснювати транзакції. Багато хто стверджує, що позитивні сторони криптовалют, такі як банківське обслуговування небанківського світу, переважають негативні, особливо коли більшість незаконних операцій все ще здійснюються за допомогою готівки, яку неможливо відстежити.

Простіше кажучи, блокчейн – це спільна база даних або реєстр. Фрагменти даних зберігаються в структурах даних, відомих як блоки, і кожен вузол мережі має копію всієї бази даних. Безпека забезпечується тим, що більшість не прийме цю зміну, якщо хтось спробує відредагувати або видалити запис в одній копії реєстру.

2.3. Блокчейн Ethereum

Ethereum – це блокчейн-мережа, яка представила вбудовану мову програмування, що завершується за Тьюрінгом, яка може бути використана для створення різних децентралізованих додатків (також званих dApps).

Ethereum – це платформа з відкритим вихідним кодом на основі блокчейну для розгортання комп'ютерних програм, які називаються смарт-контрактами, без участі центрального органу влади.

Смарт-контракти Ethereum, які часто називають "світовим комп'ютером", лежать в основі відкритого ринку цифрових додатків (dApps). Від фінансів та управління до ігор та соціальних мереж, dApps руйнують дозволені моделі Amazon, Google та Facebook.

Ethereum також використовує власну криптовалюту під назвою Ether (ETH), щоб стягувати з dApps плату за оренду обчислювальних потужностей. Децентралізована мережа комп'ютерів фіксує баланс рахунків ETH і стан смарт-контрактів, зберігаючи цю інформацію в блокчейні Ethereum.

Ethereum був продуманий у 2013 році канадським програмістом Віталієм Бутеріним. Ідея Бутеріна розвинулася з альтернативної децентралізованої платіжної мережі, коли він зрозумів, що блокчейн можна використовувати не лише для обміну новими формами інтернет-грошей.

Ethereum був запущений у липні 2015 року, і його успіх надихнув зростаючу екосистему, яка підтримує мільярди доларів вартості. Конкуруючі блокчейни, такі як Solana та Cardano, намагаються імітувати та вдосконалити концепцію децентралізованих обчислень Ethereum.

Як і будь-яка комп'ютерна система, Ethereum має власну мову Solidity, якою пишуться смарт-контракти. Існує також засіб для обробки закодованих інструкцій в смарт-контрактах, який називається віртуальна машина Ethereum (EVM).

EVM є спільною для всіх комп'ютерів, що входять до мережі Ethereum. Їхнє завдання – підтримувати канонічну версію EVM; незаперечне і незмінне джерело істини про стан додатків і облікових записів, які підтримує Ethereum.

Щоразу, коли відбувається зміна в смарт-контракті або балансу облікового запису, що містить ETH, транзакція надсилається в EVM, а її стан оновлюється і записується в блокчейн Ethereum, при цьому вся мережа погоджується з її точністю.

За виконання смарт-контрактів Ethereum стягує плату, виражену в одиницях Ефіру, відомих як GAS.

Щоб підтримувати точність і незалежність свого блокчейну і забезпечити відкритий доступ для всіх користувачів, Ethereum використовує відкриту мережу з трьома різними учасниками, відомими як ноди.

Нода – це комп'ютер, на якому запущено програмне забезпечення (клієнт), що дозволяє Ethereum функціонувати, яке називається клієнтом. Клієнти доступні на різних мовах програмування і з різних незалежних джерел, що створює різноманітність клієнтів, що сприяє загальній безпеці Ethereum.

Повні вузли запускають дві частини клієнтського програмного забезпечення: одну для виконання інструкцій смарт-контракту або нових транзакцій в EVM; інша – програмне забезпечення для консенсусу, яке гарантує, що зміни є дійсними.

Запустивши клієнт-валідатор, повні вузли також можуть виконувати функцію валідатора. Валідатор може пропонувати нові блоки, що містять пакет змін останнього стану і балансу, для додавання в блокчейн. За свої послуги валідатори отримують винагороду в ETH, а також частину комісії, що стягується за транзакції.

Процес узгодження дійсності інформації, записаної в нових блоках, в Ethereum називається Proof-of-Stake (PoS).

В рамках PoS повні вузли, які виступають в ролі валідаторів (також відомі як стейкери), обираються для створення нових блоків і перевірки транзакцій на основі суми ETH, яку вони вкладають в блок.

Валідатори повинні внести щонайменше 32 ETH, але чим більше стейків перевищує цей поріг, тим більша ймовірність того, що валідатор буде обраний для пропозиції нового блоку. Альтернатива – внести менше 32 ETH в стейкінг-пул.

Пули стейкінгу управляються одним валідатором, який має право запропонувати новий блок. Винагорода, яку ви отримаєте, буде пропорційна вашому внеску в пул.

У системі консенсусу PoS валідатори випадковим чином обираються, щоб пропонувати нові блоки і підтверджувати транзакції в слотах кожні 12 секунд.

Задекларований ETH виступає в якості застави – фінансового зобов'язання, яке перешкоджає валідації недійсних транзакцій або спробам подвійних витрат, оскільки ваш стейк може бути зменшений або конфіскований в якості штрафу.

Токеноміка описує, як механізми розподілу та винагороди впливають на пропозицію криптовалют та її зміну з часом. Збільшення пропозиції створює інфляцію, знижуючи купівельну спроможність валюти та її ефективність як засобу збереження вартості.

Під час запуску була випущена певна кількість ETH, відома як попередній майнінг (pre-mine). Частина була доступна для публічної купівлі через первинну пропозицію монет (ICO), а частина була розподілена між засновниками та учасниками.

Коли Ethereum запускався, він використовував той самий механізм консенсусу, що і біткоїн (Proof-of-Work). Але без обмеження пропозиції винагороди за блок призводили до щорічної інфляції пропозиції ETH на рівні 4,5%.

На додаток до винагороди за блок, валідатори Ethereum заохочуються додавати транзакції та зміни стану до нових блоків. Це залишається справедливим і в умовах PoS, але запроваджений у серпні 2021 року механізм спалювання частини комісій за транзакції мав вирішальний вплив на токеноміку, знизивши річну інфляцію Ethereum майже до нуля.

Злиття заблокувало стейкінг ETH, а це означає, що стейкхолдери не могли вивести ні свої початкові частки, ні накопичені винагороди до Шанхайського оновлення, яке відбулося 12 квітня 2023 року.

Тепер, коли стейкінг розблоковано і інфляція нейтральна, є надія, що винагорода валідатора Ethereum буде конкурувати з прибутковістю ринку облігацій з фіксованим доходом, який оцінюється в 133 трильйони доларів, що більш ніж в 100 разів перевищує ринкову капіталізацію всієї криптоекосистеми.

Після злиття Ethereum все ще обмежений у кількості транзакцій, що обробляються в секунду, що підвищує вартість блокчейну і призводить до поширення рішень другого рівня. Однак, злиття заклало основу для покращення

масштабованості Ethereum за допомогою майбутньої функції під назвою Danksharding.

Данкхардінг дозволить використовувати нову тимчасову форму зберігання блоків, відому як блоби, які з часом можна буде видалити. Таким чином, звільнивши сховище блоків, яке є найбільшим джерелом комісійних.

Таблиця 2.2

Переваги та недоліки Ethereum

Переваги Ethereum	Недоліки Ethereum
Децентралізація	Використання складної мови програмування
Швидке розгортання	Проблеми з масштабуванням
Дозволена мережа	Інвестування в Ethereum може бути ризикованим
Розмір мережі	
Приватні транзакції	
Масштабованість і продуктивність	
Завершеність	
Токенізація	
Інтероперабельність та відкритий код	
Стандарти	

Про переваги:

1) Децентралізація: децентралізована структура Ethereum ефективно розподіляє знання і довіру між учасниками мережі, усуваючи необхідність в центральному органі для управління системою і посередництва в транзакціях.
2) Швидке розгортання: замість того, щоб будувати реалізацію блокчейну з нуля, організації можуть швидко створювати і адмініструвати приватні мережі блокчейну, використовуючи універсальну SaaS-платформу, таку як Hyperledger Besu.
3) Дозволена мережа: існує багато рівнів протоколів з відкритим вихідним кодом, які дозволяють підприємствам створювати публічні або приватні мережі Ethereum, гарантуючи, що їх рішення відповідає всім регуляторним нормам і стандартам безпеки.
4) Розмір мережі: основна мережа Ethereum демонструє, що мережа з сотнями вузлів і мільйонами користувачів може функціонувати. Більшість бізнес-конкурентів блокчейну мають мережі з менш ніж десятьма вузлами і не мають прецедентів великої та успішної мережі. Для корпоративної співпраці, яка неодмінно переросте кілька вузлів, масштаб мережі є важливим.
5) Приватні транзакції: в Ethereum підприємства можуть отримати деталізацію конфіденційності, приєднавшись до приватних партнерств з рівнями приватних транзакцій. Приватна інформація шифрується і передається лише тим, кому вона потрібна.
6) Масштабованість і продуктивність: Консорціумні мережі, створені на Ethereum, можуть перевершити загальнодоступну мейннет-мережу і зрости до сотень транзакцій в секунду і більше, в залежності від налаштування мережі, завдяки консенсусу Proof of Authority та індивідуальним обмеженням на час блокування і газ.

   У найближчому майбутньому Ethereum зможе збільшити свою пропускну здатність завдяки рішенням на рівні протоколу, таким як

шардінг і офф-чейн, а також рішенням для масштабування на рівні 2, таким як Plasma і державні канали.

7) Завершеність: метод консенсусу в блокчейні гарантує, що запис транзакції є захищеним від підробки та канонічним. Для різних екземплярів корпоративної мережі Ethereum пропонує механізми консенсусу, що налаштовуються, такі як RAFT і IBFT, які забезпечують негайне завершення транзакції і зменшують необхідну інфраструктуру, яку вимагає алгоритм Proof of Work.
8) Токенізація: будь-який об'єкт, зареєстрований в цифровому форматі, може бути токенізований на Ethereum. Організації можуть фракціонувати раніше монолітні активи (нерухомість), розширювати свою продуктову лінійку (раритетні предмети мистецтва) і відкривати нові моделі мотивації, токенізуючи активи (управління даними на основі краудсорсингу).
9) Інтероперабельність та відкритий вихідний код: на Ethereum консорціуми не прив'язані до IT-середовища одного постачальника. Наприклад, клієнти Amazon Web Services можуть використовувати Blockchain Business Cloud від Kaleido для запуску приватних мереж. Екосистема Ethereum, як і спільнота Java, заохочує внески в кодову базу за допомогою пропозицій щодо покращення Ethereum (EIP).
10) Стандарти: Ethereum – це місце, де ви хочете встановити високу планку. Екосистема не піддається фрагментації завдяки протоколам дизайну токенів (ERC20), людським іменам (ENS), децентралізованому сховищу (Swarm) та децентралізованим комунікаціям (Whisper). Клієнтська специфікація 1.0 Корпоративного альянсу Ефіріуму описує архітектурні компоненти для сумісних корпоративних блокчейн-реалізацій. Найближчим часом EEA має намір випустити версію 2.0 специфікації.

Про недоліки:

1) Використання складної мови програмування: хоча Ethereum є повною мовою програмування Тьюринга і використовує мову програмування, подібну до C++, Python та Java, вивчення Solidity, рідної мови Ethereum, може бути складним завданням. Однією з найбільш важливих проблем є дефіцит класів, дружніх до початківців.
2) Проблеми з масштабуванням: на відміну від Bitcoin, який має єдину мету, Ethereum має реєстр, платформу для смарт-контрактів і так далі, що може призвести до помилок, збоїв і зломів.
3) Інвестування в Ethereum може бути ризикованим: інвестування, як і в будь-яку іншу криптовалюту, може бути ризикованим. Криптовалюти дуже волатильні, що призводить як до значних прибутків, так і до значних збитків. Ціна ETH значно змінювалася в минулому, що може бути значним недоліком для деяких інвесторів, особливо для новачків. Крім того, комісія Ethereum змінюється, що є незручним.

Ethereum є універсальною платформою "все в одному", що базується на блокчейні. Найпопулярнішим питанням зараз є те, чи є ефір кращим за інші криптовалюти. Ethereum пропонує значну перевагу в тому, що, крім інвестицій в ETH, можна також інвестувати у фірми, які розробляють додатки, що використовують мережу Ethereum.

2.4. Висновки до другого розділу

Щоб підсумувати інформацію, яка була розглянута у другому розділі, варто зазначити, що блокчейн є системою, яка фіксує інформацію так, що зміни, втручання чи обман системи стають важкими або неможливими. Ця технологія забезпечує децентралізовану безпеку даних, тому вона може бути ефективно використана для захисту інформації, що зберігається у хмарному сховищі.

З багатьма практичними застосуваннями цієї технології, які вже впроваджуються та досліджуються, блокчейн нарешті робить собі ім'я, не в останню чергу завдяки криптовалюті. Як модне слово на язиці кожного інвестора в країні, блокчейн покликаний зробити бізнес і державні операції більш точними, ефективними, безпечними і дешевими, з меншою кількістю посередників.

На початку третього десятиліття існування блокчейну вже не стоїть питання, чи перейдуть традиційні компанії на цю технологію – питання лише в тому, коли це станеться. Сьогодні ми бачимо поширення NFT і токенізацію активів. Як наслідок, наступні десятиліття стануть значним періодом зростання для блокчейну.

Ethereum – це платформа з відкритим вихідним кодом на базі блокчейну для розгортання комп'ютерних програм, які називаються смарт-контрактами, без проходження через центральний орган влади.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЯ ХМАРНОГО СХОВИЩА НА БАЗІ ТЕХНОЛОГІЇ БЛОКЧЕЙН

3.1. Використані інструменти та технології

Для впровадження хмарного сховища було використано технологію блокчейн, а саме Ethereum. Ethereum представляє собою децентралізовану глобальну платформу, засновану на технології блокчейн. Відомий своєю криптовалютою, ефіром (ETH), Ethereum дозволяє будь-кому створювати захищені цифрові технології. Платформа підтримує розробку смарт-контрактів, що спрощує впровадження логіки хмарного сховища. Крім того, Ethereum визнаний як один з найнадійніших блокчейнів із смарт-контрактами на сьогоднішній день.

В програмі використовано емулятор мережі Ethereum – Goerli, який призначений для випробування розробки блокчейну перед його розгортанням у Mainnet, основній мережі Ethereum.

Goerli, також відомий як Görli, був вперше представлений у вересні 2018 року і є одним з найвідоміших тестових мереж Ethereum. Більше того, це один з двох первинних тестових мереж разом з Sepolia, які не були застарілими. Тестова мережа Goerli розпочалася як проект хакатону ETHBerlin, але офіційний запуск відбувся лише на початку 2019 року на GörliCon в Берліні.

На той час Goerli була першою тестовою мережею Ethereum з підтвердженням повноважень (PoA), створеною спільнотою, з крос-клієнтською функціональністю. Механізм консенсусу PoA Goerli змушував користувачів розкривати свої реальні ідентифікаційні дані, щоб отримати право на валідацію блоків. Крім того, як тільки більшість валідаторів погоджуються з формуванням нового блоку, він підписується і додається до найдовшого ланцюга Goerli.

Таким чином, Ethereum перейшов від традиційного механізму консенсусу

з доказом роботи (PoW) до PoS. Більш того, важливим кроком в процесі злиття став перехід Goerli від PoA до PoS, який відбувся в два етапи в серпні на початку цього року. Початковим етапом було попередньо необхідне оновлення під назвою Bellatrix; тим часом, на другому етапі Goerli об'єднався з тестовою мережею Prater Beacon Chain, PoS-мережею, що працює паралельно з основною мережею Ethereum.

Злиття Goerli і перехід на PoS стало останньою можливістю для розробників переконатися, що їхні валідатори PoS були налаштовані правильно перед переходом на мейннет. Це також був останній тестовий запуск перед офіційним злиттям мережі Ethereum з Beacon Chain.

Мейннет, в контексті технології блокчейн, – це повністю розроблена і функціонуюча мережа блокчейн, в якій криптовалютні транзакції транслюються, перевіряються і записуються в розподіленому реєстрі. Це протилежність тестової мережі, і вона означає, що блокчейн готовий до публічного використання, дозволяючи користувачам відправляти і отримувати криптовалютні транзакції.

Мейннет – це основа мережі блокчейн, яка відповідає за перевірку, запис і трансляцію криптовалютних транзакцій. Це означає, що блокчейн завершив свою фазу розвитку і тепер відкритий для громадськості. Цей етап є антиподом тестової мережі, яка використовується розробниками для усунення несправностей і тонкого налаштування функціональності блокчейну до того, як він буде визнаний безпечним і готовим до запуску в основну мережу.

Мережеві транзакції та майнери: коли користувач ініціює транзакцію в мейннеті, інформація поширюється по всіх вузлах мережі. Тут в гру вступають майнери. Ці спеціальні вузли змагаються у вирішенні складних математичних головоломок, щоб додати транзакцію до наступного блоку в ланцюжку блоків. Перший майнер, який досягне успіху, отримує заздалегідь визначену винагороду в криптовалюті, що стимулює його участь.

Основні компоненти майнерської мережі: майнери – це основа майнерської мережі, які керують вузлами і використовують обчислювальну потужність для перевірки транзакцій і створення блоків. Цей кропіткий процес

забезпечує незмінність і безпеку мережі. Але запуск мейннет передбачає ряд кроків, включаючи продаж токенів, забезпечення фінансування та ретельне тестування продукту.

Постійний розвиток мережі: навіть після офіційного запуску мейннет проходить постійне тестування в паралельній тестовій мережі. Це окреме середовище дозволяє розробникам експериментувати з новими протоколами та оновленнями, зберігаючи при цьому основну мережу недоторканою. Залежно від дорожньої карти проекту, розробники можуть впроваджувати хардфорки або софтфорки для покращення функціональності.

Смарт-контракти – це програми, що зберігаються в блокчейні, які запускаються при дотриманні заздалегідь визначених умов. Зазвичай вони використовуються для автоматизації виконання угоди, щоб усі учасники могли одразу отримати результат, без участі посередника чи втрати часу. Смарт-контракти працюють, дотримуючись простих операторів «якщо/коли … то …», які записані в коді на блокчейні. Мережа комп'ютерів виконує дії, коли були виконані та перевірені заздалегідь визначені умови. Ці дії можуть включати видачу коштів відповідним особам, реєстрацію транспортного засобу, надсилання сповіщень або видачу квитка. Після завершення транзакції блокчейн оновлюється. Це означає, що транзакцію не можна змінити, і лише сторони, яким надано дозвіл, можуть бачити результати.

Виконаний контракт в термінах смарт-контракту означає успішне завершення угоди, запрограмованої в смарт-контракті. Як тільки всі умови, зазначені в коді смарт-контракту, виконані і необхідні дії здійснені, контракт вважається виконаним. Популяризовані блокчейном Ethereum, смарт-контракти призвели до появи в мережі безлічі децентралізованих додатків (DApps) та інших варіантів використання.

Однією з ключових переваг блокчейн-мереж є автоматизація завдань, які традиційно потребують стороннього посередника. Наприклад, замість того, щоб банк затверджував переказ коштів від клієнта до фрілансера, цей процес може відбуватися автоматично завдяки смарт-контракту. Це скорочує час і витрати,

пов'язані з традиційним виконанням контрактів.

В реалізованому хмарному сховищі усі дані ефективно зберігаються на блокчейні. Розробка смарт-контракту виконана мовою Solidity за допомогою середовища Remix IDE. Це середовище сприяє швидкому циклу розробки та має розширений набір плагінів із зрозумілим графічним інтерфейсом, який використовується на всіх етапах розробки смарт-контракту.

```
// SPDX-License-Identifier: MIT

pragma solidity >=0.7.0 <0.9.0;

/**
 * @title Cloud storage
 */
contract Storage {
    // Saved files.
    mapping(address => bytes[]) files;

    // Upload file to storage.
    function uploadFile(bytes calldata _file) external {
        files[msg.sender].push(_file);
    }

    // Get your files.
    function loadFiles() external view returns(bytes[] memory) {
        return files[msg.sender];
    }

    // Remove file.
    function removeFile(uint index) external returns(bool) {
        bytes[] storage _files = files[msg.sender];
        if (index > _files.length) {
            return false;
        }
        for(uint i = index; i < _files.length - 1; i++) {
            _files[i] = _files[i + 1];
        }
        _files.pop();
        return true;
    }
}
```

Рис.3.1. Смарт-контракт

Для ефективної роботи з хмарним сховищем було вирішено впровадити односторінкову програму (SPA). Односторінкова програма (SPA) представляє собою веб-сайт або веб-додаток, який динамічно оновлює поточну веб-сторінку, отримуючи нові дані з веб-сервера (в даному випадку, з блокчейну). Це відмінно від стандартного методу веб-браузера, який завантажує цілі нові сторінки. Такий

підхід дозволяє користувачам використовувати хмарне сховище на будь-якій платформі, яка підтримує браузер. Зокрема, можливо також взаємодіяти з хмарним сховищем за допомогою прямих мережевих запитів до блокчейну. На рисунку 3.2 представлено реалізовану односторінкову програму.

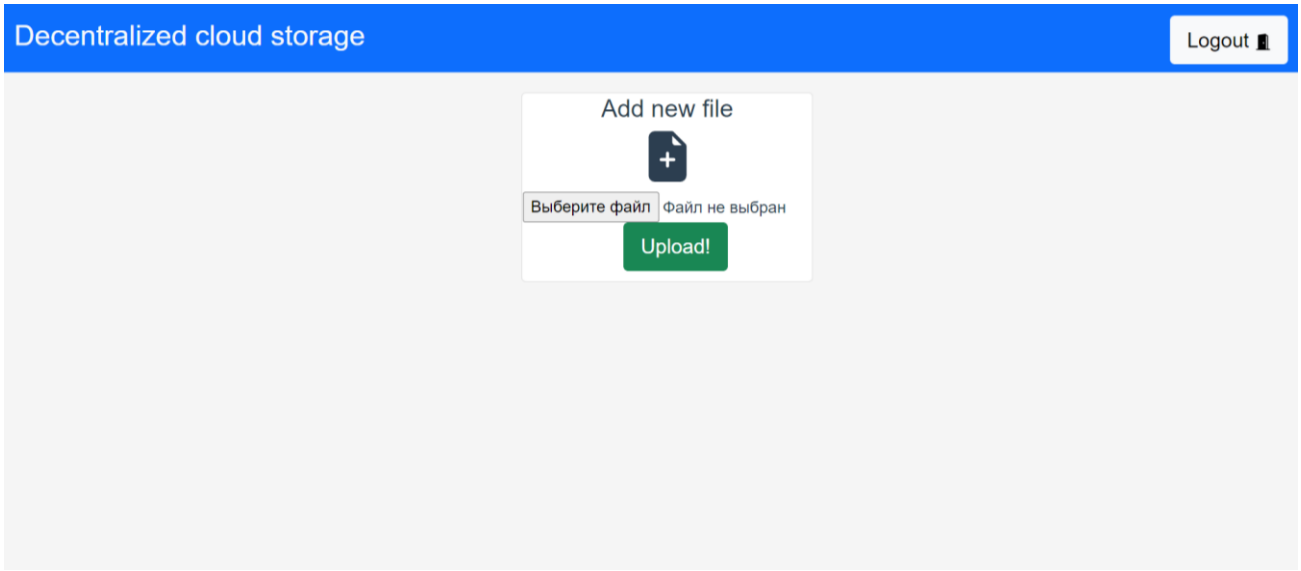

Рис.3.2. Односторінкова програма

Програмний модуль був розроблений мовою JavaScript і використовував фреймворки vue.js та bootstrap.js. Для взаємодії з блокчейном використовувалася бібліотека ethers.js, а для криптографії використовувався CryptoJS. В ролі платформи використовувався фреймворк node.js, а для керування пакетами використовувався npm як пакетний менеджер.

JavaScript – це кросплатформенна об'єктно-орієнтована мова сценаріїв, яка використовується для створення інтерактивних веб-сторінок (наприклад, зі складною анімацією, кнопками, що клікаються, спливаючими меню тощо). Існують також більш просунуті серверні версії JavaScript, такі як Node.js, які дозволяють додавати на веб-сайт більше функціональності, ніж завантаження файлів (наприклад, спільна робота в режимі реального часу між декількома комп'ютерами). Усередині хост-середовища (наприклад, веб-браузера) JavaScript може бути підключений до об'єктів цього середовища, щоб забезпечити програмний контроль над ними.

npm – це менеджер пакетів за замовчуванням для JavaScript Node.js. Він

складається з двох основних частин:

1) інструмент CLI (інтерфейс командного рядка) для публікації та завантаження пакетів;
2) онлайн-сховище, що містить пакети JavaScript.

Вся розробка виконувалася у середовищі Visual Studio Code. Це середовище комбінує в собі простоту редактора коду з усім необхідним для основного циклу розробки. Visual Studio Code забезпечує повну підтримку редагування коду, навігацію, зручне налагодження, має багату модель розширюваності і легко інтегрується з існуючими інструментами. Це середовище також зручне тим, що його можна використовувати для більшості існуючих мов програмування.

Visual Studio Code – це безкоштовний, легкий, але потужний редактор вихідного коду, який працює на робочому столі та в Інтернеті і доступний для Windows, macOS, Linux та Raspberry Pi OS. Він має вбудовану підтримку JavaScript, TypeScript та Node.js, а також багату екосистему розширень для інших мов програмування (таких як C++, C#, Java, Python, PHP та Go), середовищ виконання (таких як .NET та Unity), середовищ (таких як Docker та Kubernetes) та хмар (таких як Amazon Web Services, Microsoft Azure та Google Cloud Platform).

Окрім легкого та швидкого запуску, Visual Studio Code має функцію завершення коду IntelliSense для змінних, методів та імпортованих модулів; графічну налагодження; лінтування, багатокурсорне редагування, підказки параметрів та інші потужні функції редагування; зручну навігацію та рефакторинг коду; вбудований контроль вихідного коду, включаючи підтримку Git. Багато з цього було адаптовано з технології Visual Studio.

Власне код Visual Studio створено за допомогою оболонки Electron, Node.js, TypeScript і протоколу Language Server Protocol, і оновлюється щомісяця. Численні розширення оновлюються так само часто, як і код. Підтримка різних мов програмування та їх розширень варіюється від простого підсвічування синтаксису та узгодження дужок до налагодження та

рефакторингу. Ви можете додати базову підтримку для вашої улюбленої мови за допомогою колоризаторів TextMate, якщо немає мовного сервера.

На рис.3.3 зображено інтерфейс Visual Studio Code.

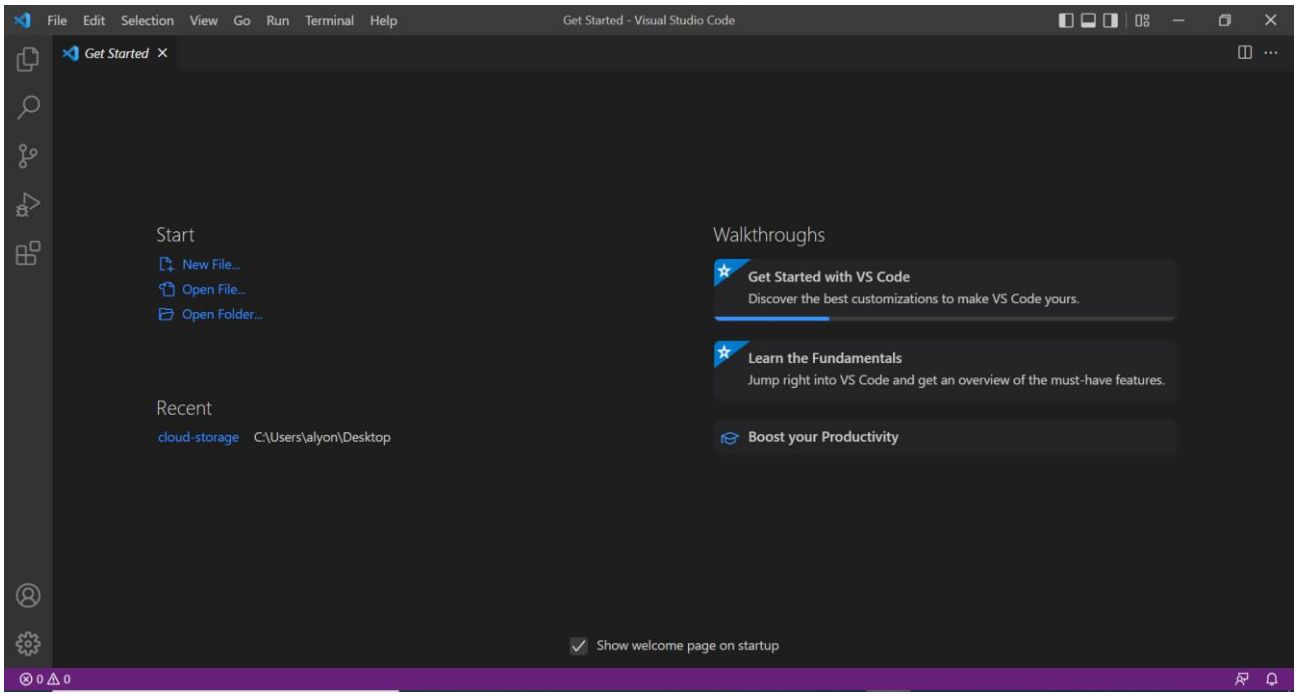

Рис.3.3. Середовище програмування Visual Studio Code

Package.json – файл для npm, де можна побачити використані бібліотеки та фреймворки, а також скрипти для запуску/збирання проекту.

```
{} package.json > ...
1   {
2     "name": "cloud-storage",
3     "version": "0.0.0",
      ▷ Debug
4     "scripts": {
5       "dev": "vite",
6       "build": "vite build",
7       "preview": "vite preview --port 4173"
8     },
9     "dependencies": {
10      "bootstrap": "^5.1.3",
11      "bootstrap-icons": "^1.8.3",
12      "crypto-js": "^4.1.1",
13      "ethers": "^5.6.8",
14      "vue": "^3.2.36",
15      "vue-horizontal": "^1.0.2"
16    },
17    "devDependencies": {
18      "@vitejs/plugin-vue": "^2.3.3",
19      "vite": "^2.9.9"
20    }
21  }
```

Рис.3.4. Файл package.json

3.2. Програмна реалізація хмарного сховища

В реалізованому хмарному сховищі усі дані файлів знаходяться в хеш-мапі files, що реалізована на смарт-контракті. Вся інформація про кожен файл зберігається у зашифрованому вигляді. Завантажити всі файли користувача можна використовуючи метод loadFiles(). Для внесення змін до файлів користувача доступні методи: uploadFile(bytes calldata _file), removeFile(uint index).

Для входу в хмарне сховище вам знадобиться скористатися приватним ключем, який був згенерований в основній мережі Ethereum. У випадку введення некоректного ключа вхід буде невиконаним, і прийдеться повторно ввести правильний ключ.

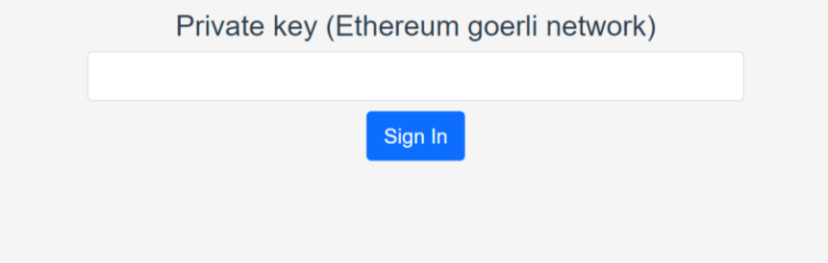

Рис.3.5. Початковий екран

```
<template>
    <div class="vue-tempalte">
        <form>
            <div class="form-group">
                <h3 class = "top-buffer">Private key (Ethereum goerli network)</h3>
                <input class="form-control form-control-lg w-50 mx-auto" id="key"/>
            </div>
            <button @click="login()" class="btn btn-primary btn-lg btn-block top-buffer">Sign In</button>
        </form>
    </div>
</template>
<script>
    export default {
        data() {
            return {}
        },
        methods: {
            login: function() {
                console.log(document.getElementById('key').value)
                localStorage.setItem('key', document.getElementById('key').value)
            }
        }
    }
</script>
```

Рис.3.6. Програмний код компоненти форми входу

При вході в хмарне середовище, вікно прийме наступний вигляд:

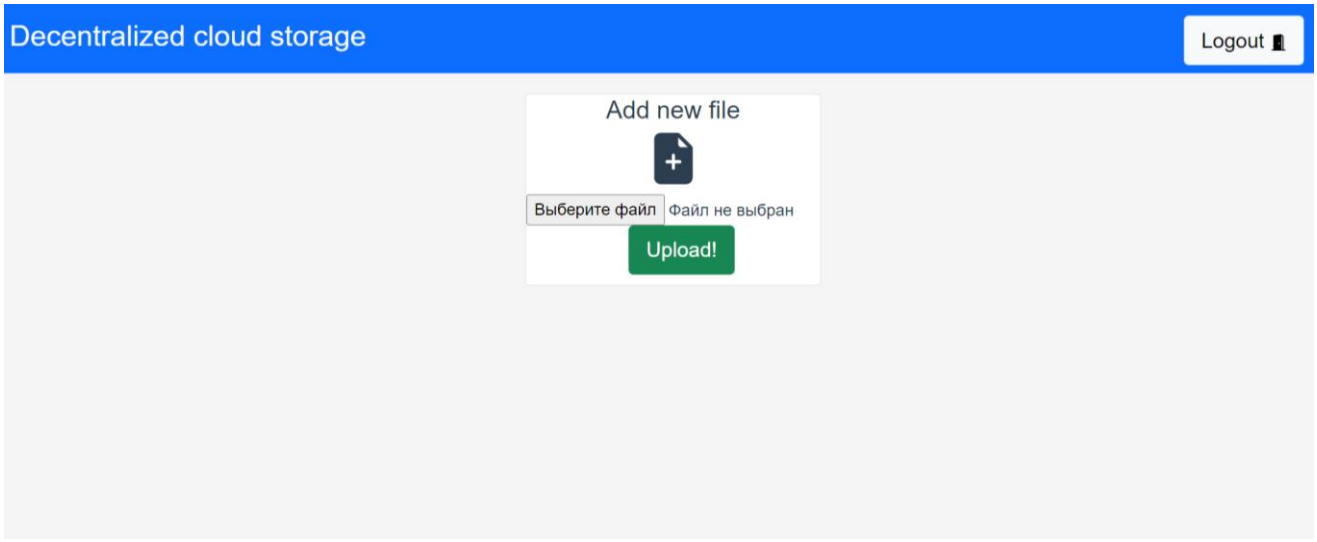

Рис.3.7. Вікно виконаного входу в хмарне сховище

Як видно на цьому етапі, ви маєте змогу завантажувати файли у хмарне сховище, і вгорі розташована кнопка для виходу.

Наступним кроком буде спроба завантаження файлу у наше сховище. Для цього натискайте кнопку "Виберіть файл", оберіть необхідний файл та натискайте " Upload!".

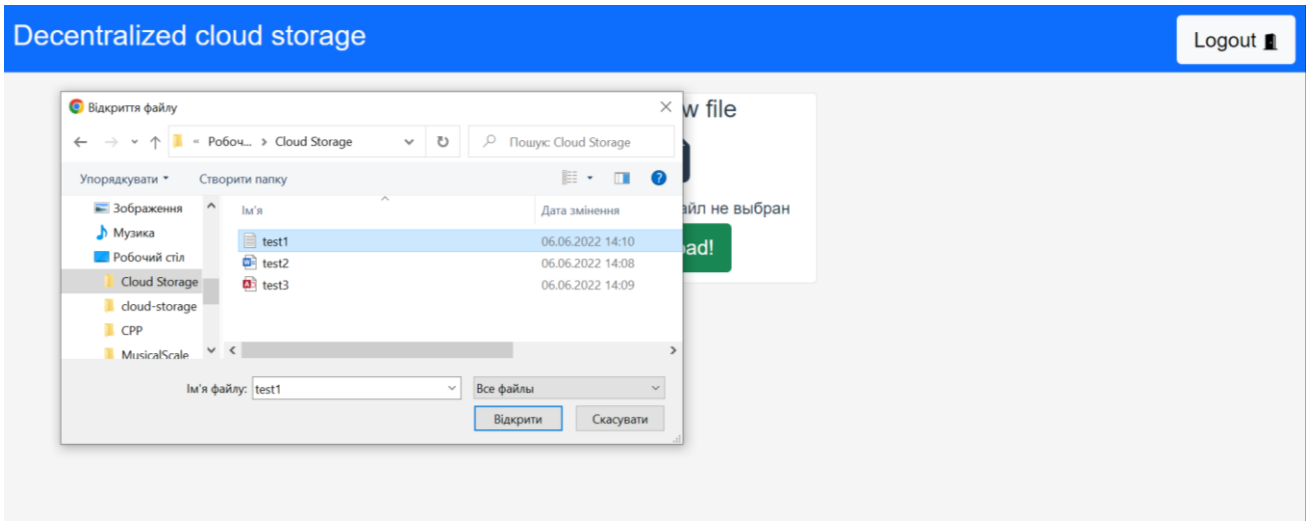

Рис.3.8. Завантаження файлу

Процес завантаження файлу може займати приблизно одну хвилину, оскільки протягом цього часу відбудеться транзакція, і файл буде збережено у блокчейні.

Як виглядає завантажений файл зображено на рис.3.9.

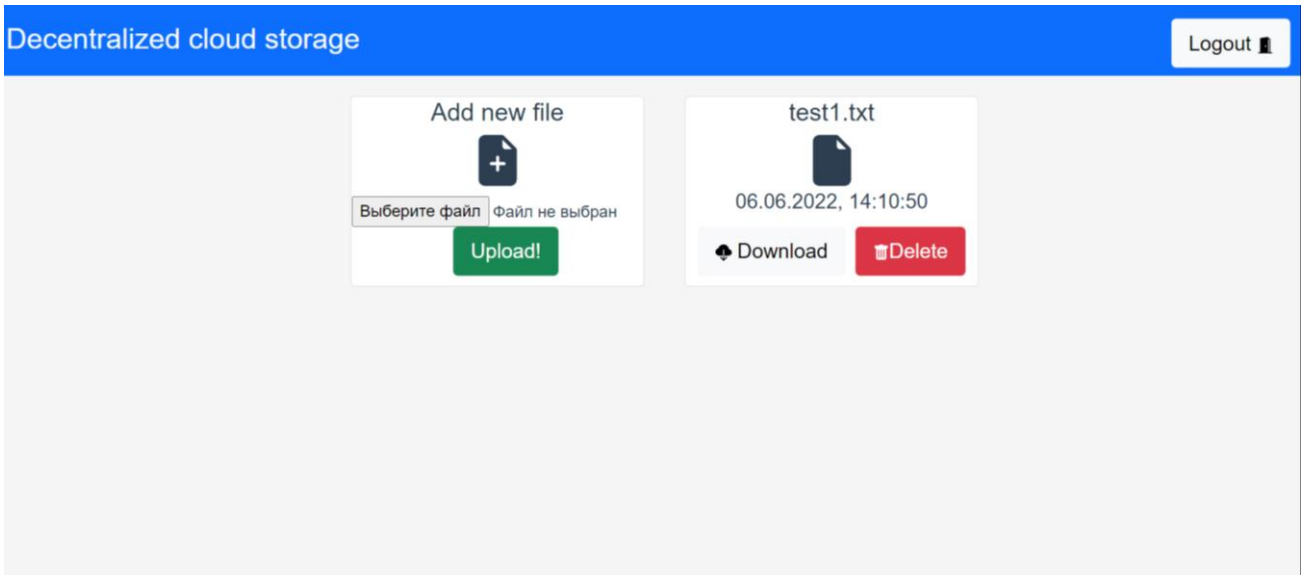

Рис.3.9. Завантажений файл в хмарне сховище

Щоб впевнитися в правильності роботи процесу завантаження, давайте спробуємо файли різних форматів.

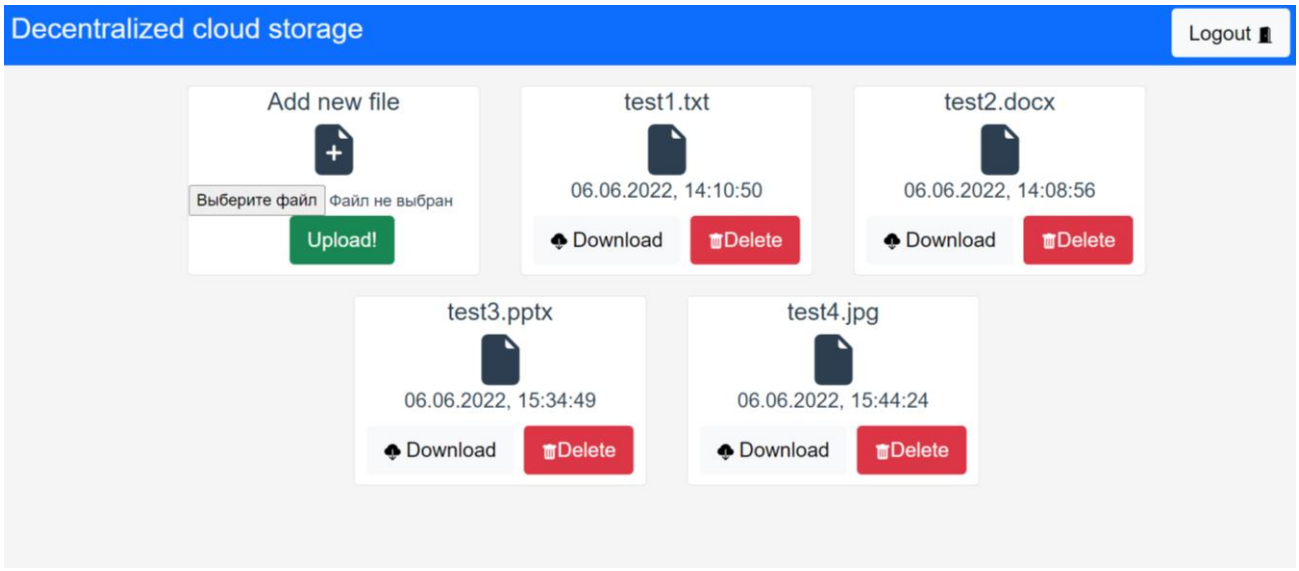

Рис.3.10. Завантажені файли різних типів

Як видно, у створене хмарне сховище можна зберегти файли різних форматів.

У додатку до можливості обробки різних типів файлів, реалізоване хмарне сховище є універсальним і сумісним з усіма операційними системами, які підтримують різні веб-браузери, включаючи Windows, Linux, MacOS, iOS, Android та інші.

Компонента UploadFile відповідає за вибір файлу, та його завантаження. Можемо побачити цю компоненту на рис.3.11.

```
<script>
import { ethers } from "ethers";
import AES from "crypto-js/aes";
import CryptoJS from "crypto-js";
const address = "0xB9F7cd5508493b8eD05d081F246C73716Dde2BC3";
const fileEncodingPattern = ['string', 'uint256', 'bytes']
const abi = [{"inputs":[{"internalType":"uint256","name":"index","type":"uint256"}],"name":"removeFile",
export default {

  name: '',
  data () {
    return {
      msg: ''
    }
  },
  methods: {
    onFileChanged (event) {
      this.selectedFile = event.target.files[0]
      const file = event.target.files[0]
      console.log(file)
    },
    onUpload () {
      const reader = new FileReader();
      const fileByteArray = [];
      reader.readAsArrayBuffer(this.selectedFile)
      reader.onloadend = (evt) => {
      if (evt.target.readyState === FileReader.DONE) {
        const arrayBuffer = evt.target.result,
          array = new Uint8Array(arrayBuffer);
        for (const a of array) {
          fileByteArray.push(a);
        }
        console.log(fileByteArray)

        let name = this.selectedFile.name;
        let date = this.selectedFile.lastModified;
        let file_bytes = ethers.utils.defaultAbiCoder.encode(fileEncodingPattern, [name, date, fileByteAr
        console.log(file_bytes);
        let provider = ethers.providers.getDefaultProvider("goerli");
        let private_key = localStorage.getItem('key').toString()
        let wallet = new ethers.Wallet(private_key, provider);

        let contract = new ethers.Contract(address, abi, wallet);
        let data = CryptoJS.enc.Hex.parse(file_bytes);
        let file_encrypted = AES.encrypt(data, private_key).toString();
        console.log(file_encrypted);
        let bytes = ethers.utils.defaultAbiCoder.encode(["string"], [file_encrypted])
        console.log(bytes)
        contract.uploadFile(
          bytes
        ).then(result => {
          console.log('upload file result: ' + result)
        })
        }
      }


    }
  }
```

Рис.3.11. Код компоненти завантаження файлу

Давайте розглянемо функції, які доступні для вже завантажених файлів у хмарному сховищі. За допомогою наявних кнопок, ви можете зручно завантажити файл на свій пристрій або видалити його з хмарного сховища.

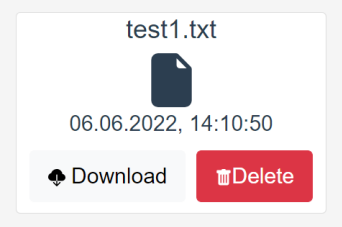

Рис. 3.12. Дії із завантаженим файлом

Далі виконаємо завантаження файлу test1.txt з хмарного сховища, щоб перевірити правильність роботи програми та коректність шифрування та розшифрування даних. Для цього слід натиснути кнопку «Download» на відповідному файлі.

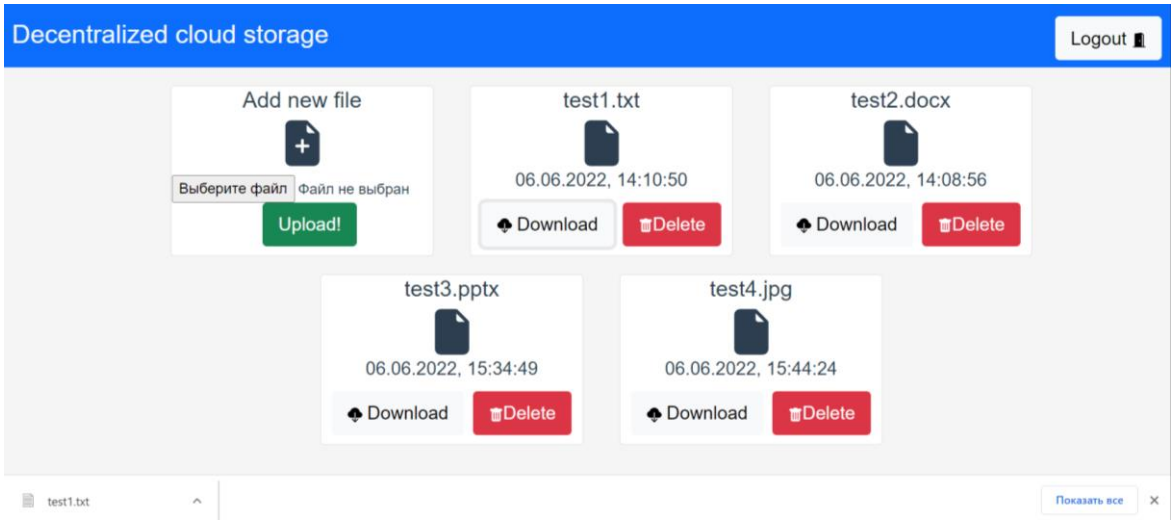

Рис.3.13. Завантаження файла із хмарного сховища

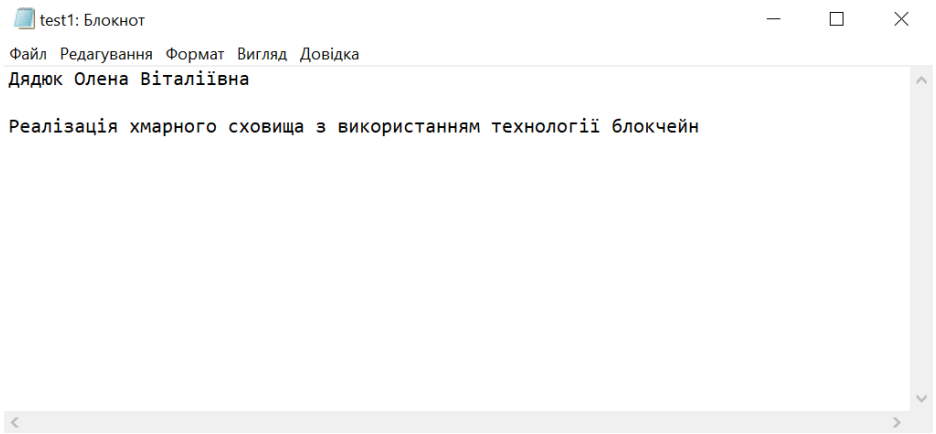

Рис.3.14. Завантажений файл

Як можна побачити на рис.3.14, файл test1.txt був успішно завантажений, його вміст відтворюється правильно. Отже, можна стверджувати, що шифрування та розшифрування було виконано правильно.

Тепер видалимо завантажений раніше у хмарне сховище файл test4.jpg. Для цього на файлі слід натиснути кнопку «Delete».

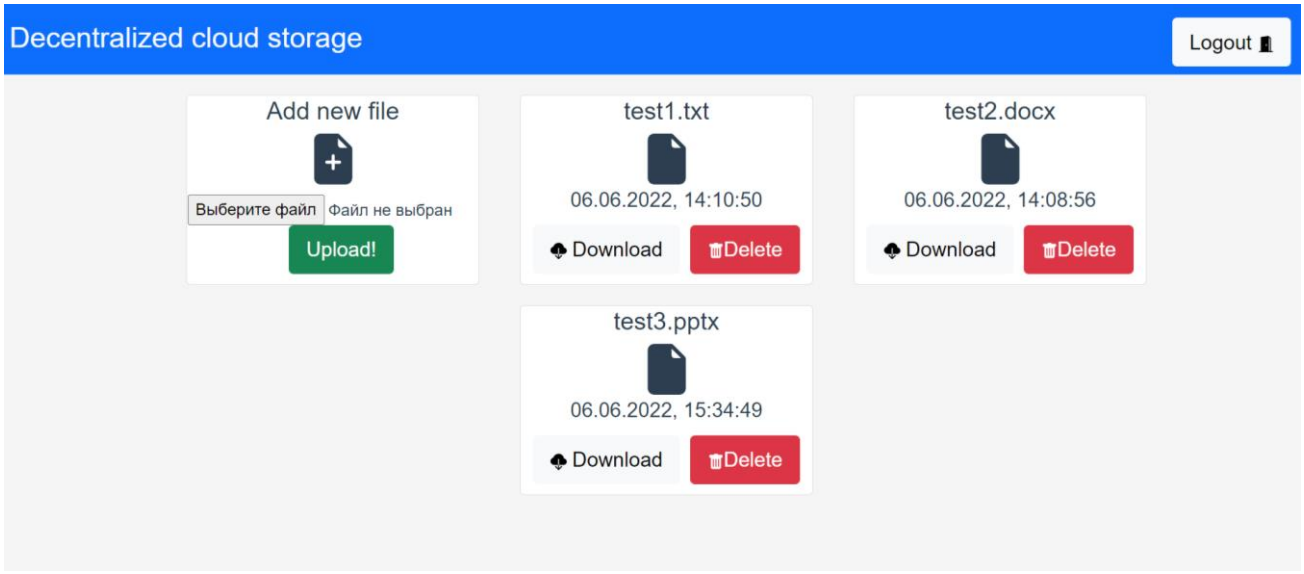

Рис.3.15. Видалення файлу

Файл test4.jpg було видалено із хмарного сховища.

Час, необхідний для виконання операцій завантаження та видалення файлу з хмарного сховища, становить приблизно стільки ж, як і час для завантаження файлу в хмару, середньо до однієї хвилини.

Операції завантаження та видалення файлу реалізовані у компоненті File. На зображенні – фрагменти коду із даними операціями.

Спочатку операція завантаження файла.

```
52  <script>
53  import { ethers } from "ethers";
54  const address = "0xB9F7cd5508493b8eD05d081F246C73716Dde2BC3";
55  const abi = [{"inputs":[{"internalType":"uint256","name":"index","type":"uint256"}],"name":"removeFile"
56  export default {
57      props: ['file'],
58    methods: {
59      onDownload () {
60          var link=document.createElement('a');
61          link.href=window.URL.createObjectURL(this.file.blob);
62          link.download=this.file.name;
63          link.click();
64      },
```

Рис.3.16. Код завантаження файлу із хмарного сховища

Далі фрагмент коду із операцією видалення файлу.

```
    onDelete () {
        let provider = ethers.providers.getDefaultProvider("goerli");
        let private_key = localStorage.getItem('key').toString()
        let wallet = new ethers.Wallet(private_key, provider);

        let contract = new ethers.Contract(address, abi, wallet);
        console.log('remove index' + this.file.index)
        contract.removeFile(
          this.file.index
        ).then(result => {
          console.log('remove file result: ' + result)
        })
    }

  },
};
</script>
```

Рис.3.17. Код видалення файлу із хмарного сховища

Компонента App визначає основне вікно програми. У верхньому меню розміщена кнопка вийти з облікового запису. Нижче наведений список файлів, які збережені у блокчейні, а також вбудований компонент для завантаження нових файлів. В компоненті App вбудована логіка для проведення операцій завантаження та розшифрування файлів з блокчейну.

```
export default {
  name: "app",
  components: {File},
  data() {
    return {
      rows: []
    }
  },
  created() {
    if (localStorage.getItem("key") !== null) {
      this.load()
    }
  },
  methods: {
    check: function () {
      console.log(localStorage.getItem("key"));
      return localStorage.getItem("key") === null;
    },
    logout: function () {
      localStorage.removeItem("key");
      this.$forceUpdate();
    },
    load() {
      let provider = ethers.providers.getDefaultProvider("goerli");
      let private_key = localStorage.getItem('key').toString()
      let wallet;
      try {
        wallet = new ethers.Wallet(private_key, provider)
      } catch {
```

Рис.3.18. Код компоненти App

Всі операції із файлами фіксуються у блокчейні як транзакції. Історію цих транзакцій можна переглянути за допомогою Etherscan.

Etherscan – це онлайн-платформа для дослідження блоків і аналізу, яка надає можливість отримати інформацію про будь-які транзакції в блокчейні Ethereum, які перебувають на розгляді або вже підтверджені.

Etherscan може допомогти визначити, як ви взаємодієте з блокчейном, іншими гаманцями та децентралізованими додатками. Крім того, ця інформація може допомогти вам залишатися в безпеці і виявляти підозрілу поведінку.

Щоб скористатися Etherscan, потрібно ввести адресу гаманця, ідентифікатор транзакції (TXID), адресу контракту або інший унікальний ідентифікатор в поле пошуку. Інформація, що відображається, залежить від критеріїв пошуку, але здебільшого вона містить транзакції, адреси, часові мітки та пов'язані з ними значення.

Через Etherscan також можна безпосередньо взаємодіяти зі смарт-контрактами для здійснення транзакцій, моніторингу.

Etherscan не надає гаманець Ethereum і не несе відповідальності за зберігання приватних ключів. Крім того, торгівля на платформі Etherscan заборонена. Вона слугує лише сховищем даних блокчейну та базою даних для смарт-контрактів.

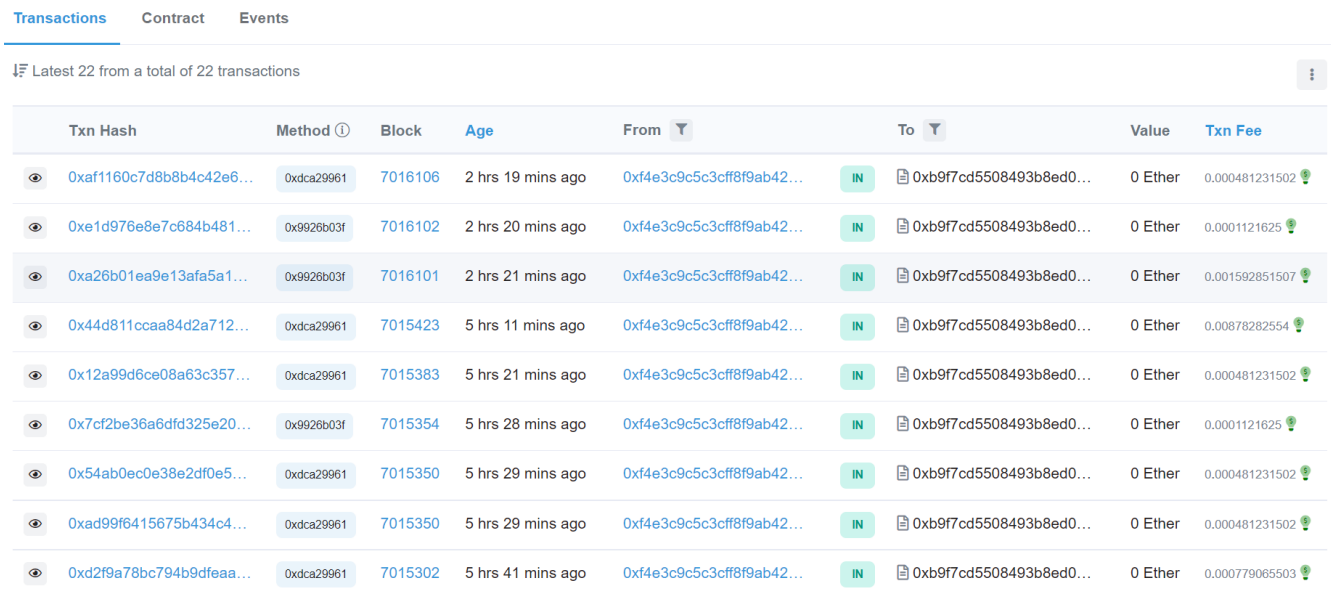

Рис.3.19. Скриншот із Etherscan

3.3. Використаний метод шифрування даних

AES представляє собою симетричний блочний метод шифрування, оскільки використовує один і той же ключ як для шифрування, так і для дешифрування інформації. Цей алгоритм використовує кілька раундів для шифрування даних, і саме ці раунди забезпечують високий рівень безпеки у випадку AES.

Існує три варіанти довжин ключа для шифрування AES: 128 біт, 192 біти та 256 бітів. Незважаючи на те, що довжина ключа відрізняється у цих варіантах, розмір блоку залишається постійним і складає 128 біт (або 16 байтів).

Процес шифрування AES є відносно простим для розуміння, що сприяє легкій реалізації, а також високій швидкості як у шифруванні, так і у дешифруванні. Крім того, AES вимагає менше обсягу пам'яті, порівняно з багатьма іншими методами шифрування. У разі потреби в додатковому рівні безпеки можна поєднувати AES з різними протоколами безпеки або навіть іншими методами шифрування.

Завдяки своїй надійності, шифрування AES тримається як стандарт шифрування протягом двадцяти років.

В програмній реалізації для зручності був використаний 256-бітний ключ, який отримано з приватного ключа для блокчейна. Для шифрування використовувалася реалізація AES з бібліотеки CryptoJS.

```
let data = CryptoJS.enc.Hex.parse(file_bytes);
let file_encrypted = AES.encrypt(data, private_key).toString();
```

Рис.3.20. Шифрування

```
let file_bytes_decrypted = '0x' + AES.decrypt(str, private_key).toString().substring(2);
```

Рис.3.21. Дешифрування

Модифікація даних до/після шифрування здійснюється через специфічне представлення у програмі.

Окрім того, шифрується вся інформація про сам файл, включаючи метадані (ім’я, дата редагування).

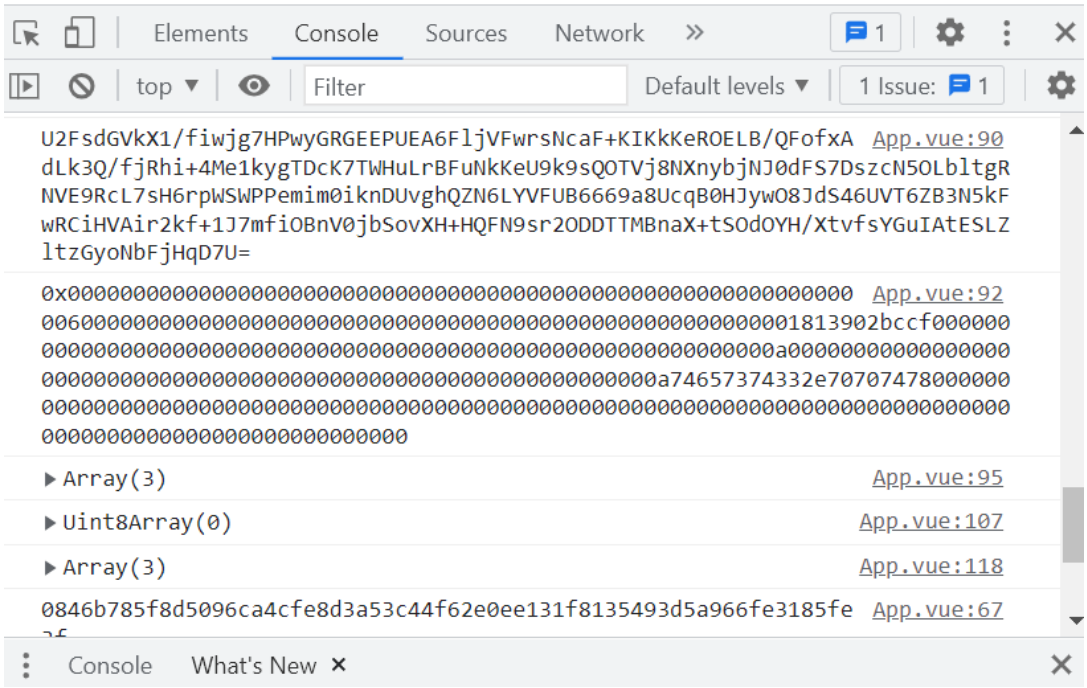

Рис.3.22. Зашифрований файл

3.4. Висновки до третього розділу

Отже, у третьому розділі описаний програмний модуль хмарного сховища на базі технології блокчейн.

Розроблена програма сумісна з найпопулярнішими операційними системами (Windows, Linux, MacOS, IOS, Android та інші). Крім того, створено зручний інтерфейс для взаємодії з смарт-контрактом та інтуїтивно зрозумілий графічний інтерфейс для користувача.

Використано блокчейн Ethereum, який є найнадійнішим на сьогодні, і його можливості спрощують реалізацію логіки хмарного сховища. Написання смарт-контракту виконувалося у середовищі Remix IDE за допомогою мови програмування Solidity.

Усі етапи розробки відбувалися у середовищі Visual Studio Code, використовуючи мову програмування JavaScript. Для підтримки функціональності були використані фреймворки vue.js та bootstrap.js, а також бібліотека ethers.js для взаємодії з блокчейном. Для криптографічних операцій використовувалася бібліотека CryptoJS. Для ефективного управління залежностями використовувався пакетний менеджер npm.

Додатковий захист інформації у хмарному сховищі здійснюється за допомогою симетричного блочного шифру AES із 256-бітним ключем.

Система хмарного сховища, яку було реалізовано, гарантує стійкість цілісності та забезпечує високий рівень конфіденційності інформації.

РОЗДІЛ 4. ОТОЧУЮЧЕ СЕРЕДОВИЩЕ

Взаємодія людини з навколишнім середовищем включає в себе різноманітні аспекти, оскільки цей процес впливає на екологічну, соціокультурну, економічну та глобальну динаміку. Давайте розглянемо детально кілька ключових аспектів цієї взаємодії.

Спочатку аспекти екологічного впливу:

1) Атмосфера: викиди транспорту, промисловість та інші джерела сприяють забрудненню повітря.
2) Вода: викиди токсичних речовин, виливи нафти та хімічних речовин у водні ресурси.
3) Ґрунт: використання пестицидів, сміттєзвалища та інші джерела забруднення ґрунту.
4) Ліси: широкомасштабне вирубування лісів для деревини та землевикористання.
5) Водні ресурси: надмірне використання води для промисловості та сільського господарства.
6) Мінерали та руди: видобуток, що може впливати на розміщення та ландшафт.
7) Деградація екосистем, втрата природних місць проживання та зменшення чисельності видів.

Соціокультурні аспекти:

1) Вплив культурних, релігійних та історичних факторів на відношення до природи.
2) Формування екологічної свідомості та відповідального співпорозуміння.

Економічні аспекти:

1) Ефективність та сталість використання ресурсів в економіці.
2) Раціональне управління природними ресурсами.

3) Зв'язок між розвитком економіки та збереженням екосистем.

Глобальний вплив на зміни клімату:

1) Роль людської діяльності у зміні клімату.
2) Пошук альтернативних технологій для зменшення викидів.
3) Зусилля у сфері адаптації та стратегії для вирішення проблем, пов'язаних із змінами клімату.

Ці аспекти взаємодії з навколишнім середовищем вимагають інтегрованого підходу та спільних зусиль для забезпечення сталого розвитку та збереження природи для майбутніх поколінь.

Захист навколишнього середовища – це невід'ємна частина сучасного світу, що передбачає комплекс заходів для збереження та забезпечення стійкості природних екосистем. Цей процес включає в себе різноманітні аспекти, охоплюючи екологічний, соціальний та економічний виміри.

У сучасному світі виникають проблеми, пов'язані із забрудненням повітря, води та грунту, використанням невідновлюваних ресурсів та змінами клімату. Заходи для захисту навколишнього середовища орієнтовані на вирішення цих проблем та створення умов для сталого розвитку.

Важливим аспектом є збереження біорізноманіття, що включає в себе створення заповідників та природних резерватів, а також відновлення пошкоджених екосистем. Зниження рівня забруднення та відходів є ще однією ключовою складовою захисту природи, включаючи впровадження стандартів для обмеження викидів та підтримку вторинної переробки.

Необхідно також працювати над сталим використанням природних ресурсів, зокрема через раціональне лісовпорядкування та ефективне використання водних ресурсів. Заходи щодо захисту клімату включають в себе пошук альтернативних джерел енергії та обмеження викидів парникових газів.

Екологічна освіта та інформування громадськості грають важливу роль у формуванні екологічної свідомості. Розвиток зеленої економіки та сприяння екологічно відповідальним підприємствам є важливим кроком у напрямку сталого розвитку.

Сумарно, захист навколишнього середовища вимагає інтегрованого підходу, що поєднує зусилля громадськості, урядових інституцій, науковців та бізнесу.

Люди вживають різноманітних заходів для збереження навколишнього середовища, таких як використання відновлюваних джерел енергії, посадка дерев, вторсировинна переробка відходів, підвищення енергоефективності та інші практики для зменшення викидів та впливу на природу.

Понадто, споживачі вибирають екологічно чисті товари та послуги, розвивають екологічні звички, такі як використання повторної посуди чи обмін товарів, а також ведуть освітню роботу щодо екологічної усвідомленості в громадах. Групи активістів і громадських організацій працюють над підтримкою та впровадженням стандартів та політик з охорони довкілля, сприяючи сталому використанню ресурсів та зменшенню негативного впливу на природу.

ВИСНОВКИ

Підсумовуючи представлену у роботі інформацію, слід відзначити, що зараз використання хмарних сховищ є поширеним явищем, тому особливу увагу слід приділяти питанням безпеки даних.

Застосування технології блокчейн та симетричного шифрування AES у створенні хмарного сховища гарантує надійність цілісності та забезпечує конфіденційність збережених у ньому даних.

В ході виконання даної роботи було виконано наступні задачі:

- проаналізовано характеристики хмарних сховищ, основні види загроз, сучасні засоби захисту даних у хмарному середовищі, опрацьовано інформацію про технології блокчейн та її застосування;
- реалізовано програмний модуль хмарного сховища мовою JavaScript з використанням фреймворків vue.js, bootstrap.js та смарт-контракту мовою Solidity на базі блокчейна Ethereum;
- протестовано програмний модуль хмарного сховища, його функціонал, перевірено забезпечення цілісності і конфіденційності збережених в хмарі даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What Is Cloud Storage? Definition, Types, Benefits, and Best Practices [Електронний ресурс]. – Режим доступу до ресурсу: https://www.toolbox.com/tech/cloud/articles/what-is-cloud-storage/.
2. Що таке хмарні сховища та як вони працюють [Електронний ресурс]. – Режим доступу до ресурсу: https://info.nic.ua/uk/blog-uk/cloudstorage-2/.
3. What is Cloud Storage Security? Securing Cloud Storage Usage [Електронний ресурс]. – Режим доступу до ресурсу: https://digitalguardian.com/blog/what-cloud-storage-security-securing-cloud-storage-usage.
4. Cloud Storage Security: How to Secure Your Data in the Cloud? [Електронний ресурс]. – Режим доступу до ресурсу: https://tresorit.com/blog/cloud-storage-security/.
5. How Cloud Storage Works [Електронний ресурс]. – Режим доступу до ресурсу: https://computer.howstuffworks.com/cloud-computing/cloud-storage.htm.
6. 3 Types of Cloud Storage – An Infographic Guide [Електронний ресурс]. – Режим доступу до ресурсу: https://vexxhost.com/blog/3-types-of-cloud-storage-an-infographic-guide/.
7. Безпека і ризики хмарного збереження даних [Електронний ресурс]. – Режим доступу до ресурсу: https://ojs.ukrlogos.in.ua/index.php/grail-of-science/article/view/10691/10048.
8. Зберігання даних компаній у хмарному сховищі – наскільки це наразі безпечно [Електронний ресурс]. – Режим доступу до ресурсу: https://www.unian.ua/techno/zberigannya-danih-kompaniy-u-hmarnomu-shovishchi-naskilki-ce-narazi-bezpechno-11812803.html.
9. Apple Platform Security [Електронний ресурс]. – Режим доступу до ресурсу: https://support.apple.com/uk-ua/guide/security/welcome/web.
10. Cloud VPN overview [Електронний ресурс]. – Режим доступу до ресурсу:

https://cloud.google.com/network-connectivity/docs/vpn/concepts/overview#comparison_table.

11. 10 BEST Free Cloud Storage Providers (Online Storage 2022) [Електронний ресурс]. – Режим доступу до ресурсу: https://www.softwaretestinghelp.com/cloud-storage-providers/.
12. Advanced Encryption Standard [Електронний ресурс]. – Режим доступу до ресурсу: https://uk.wikipedia.org/wiki/Advanced_Encryption_Standard.
13. Що таке блокчейн технологія: це не лише про криптовалюту [Електронний ресурс]. – Режим доступу до ресурсу: https://futurenow.com.ua/shho-take-blokchejn-tehnologiya-tse-ne-lyshe-pro-kryptovalyutu/.
14. Blockchain Definition: What You Need to Know [Електронний ресурс]. – Режим доступу до ресурсу: https://www.investopedia.com/terms/b/blockchain.asp#toc-pros-and-cons-of-blockchain.
15. What are smart contracts on blockchain? [Електронний ресурс]. – Режим доступу до ресурсу: https://www.ibm.com/topics/smart-contracts.
16. Goerli [Електронний ресурс]. – Режим доступу до ресурсу: https://www.anyblockanalytics.com/networks/ethereum/goerli/.
17. Ethereum Definition [Електронний ресурс]. – Режим доступу до ресурсу: https://www.investopedia.com/terms/e/ethereum.asp.
18. What is Etherscan and how to use it [Електронний ресурс]. – Режим доступу до ресурсу: https://inevitable.education/uncategorized/what-is-etherscan-and-how-to-use-it/.
19. Welcome to Remix’s documentation! [Електронний ресурс]. – Режим доступу до ресурсу: https://remix-ide.readthedocs.io/en/latest/.
20. What is AES encryption and how does it work? [Електронний ресурс]. – Режим доступу до ресурсу: https://cybernews.com/resources/what-is-aes-encryption/.
21. How To Secure Data in the Cloud [Електронний ресурс]. – Режим доступу

до ресурсу: https://www.frameworkit.com/blog/5-tips-to-keep-your-data-secure-in-the-cloud.

22. What is Ethereum? [Електронний ресурс]. – Режим доступу до ресурсу: https://www.geeksforgeeks.org/what-is-ethereum/.
23. Advantages and Disadvantages of Ethereum? [Електронний ресурс]. – Режим доступу до ресурсу: https://www.analyticssteps.com/blogs/advantages-and-disadvantages-ethereum.
24. Goerli Testnet [Електронний ресурс]. – Режим доступу до ресурсу: https://moralis.io/goerli-eth-what-is-the-goerli-testnet/.
25. Everything about JavaScript [Електронний ресурс]. – Режим доступу до ресурсу: https://faun.pub/everything-about-javascript-e4e5f7ad7fdb.