для забезпечення цілісності та перевірюваності журналів аудиту. Завдяки детальному дослідженню концептуальної структури та аналізу потенційних викликів і рішень, цей підхід довів свою ефективність у підвищенні надійності та безпеки аудиту системних подій. Результати цього дослідження надають основу для майбутніх імплементацій, досліджень та рішень для організацій, що прагнуть покращити свої процеси аудиту системних подій.

Ключові слова: аудит подій системи, блокчейн, eBPF, імутабельність, моніторинг.

Глущенко Павло Костянтинович, аспірант кафедри захисту інформації, інститут ІКТА, Національний університет Львівська Політехніка.

Pavlo Hlushchenko, Ph.D. Candidate, Information Security department, Lviv Polytechnic National University.
E-mail: pavlo.k.hlushchenko@lpnu.ua.
Orcid ID: 0000-0002-1262-5484.

Дудикевич Валерій Богданович, доктор технічних наук, професор кафедри захисту інформації, ІКТА, Національний університет Львівська Політехніка.

Valerii Dudykevych, Doctor of technical sciences, professor of Information Security department, Lviv Polytechnic National University.
E-mail: valerii.b.dudykevych@lpnu.ua.
Orcid ID: 0000-0001-8827-9920.

DOI: 10.18372/2410-7840.26.18845
УДК 004.056.5

ЗАХИЩЕНЕ ЗБЕРІГАННЯ ДАНИХ ІЗ ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ БЛОКЧЕЙН ETHEREUM

Олег Гарасимчук, Юрій Наконечний, Тарас Луковський, Роман Андріїв, Тарас Наконечний

Постійне збільшення обсягів даних породжує проблеми, пов'язані з вибором ефективних методів та засобів зберігання, а також забезпеченням захисту цих даних від несанкціонованого доступу. У статті детально розглянуто критичну тему збереження та захисту важливої інформації в умовах зростання обсягів даних і численності кібератак. Необхідність надійного збереження і захисту даних наростає, особливо у контексті підвищеної загрози з боку зловмисників. Висвітлюється, як блокчейн-технології, особливо на базі платформи Ethereum, можуть вирішити проблеми надійного збереження і безпеки даних. Ethereum пропонує альтернативу традиційній клієнт-серверній моделі, децентралізуючи зберігання даних за допомогою розподіленої мережі вузлів. Ця технологія значно підвищує безпеку, ускладнюючи несанкціонований доступ до інформації, оскільки для злому приватного ключа потрібні значні обчислювальні ресурси. Смарт-контракти на Ethereum дозволяють створювати застосунки, які виконуються точно відповідно до заздалегідь визначених умов, без можливості втручання третіх осіб. Це особливо важливо для месенджерів, де конфіденційність і доступність даних мають принципове значення. Вартість транзакцій в блокчейні, хоча й висока, компенсується високою надійністю та безпекою зберігання даних. Застосована методологія підтверджує, що використання публічного (децентралізованого) сховища даних є безпечним, оскільки зламати приватний ключ Ethereum практично неможливо.

***Ключові слова**: кібербезпека, зберігання даних, блокчейн, Ethereum.*

ВСТУП

У сучасному світі спостерігається стрімке збільшення обсягів інформації, що містить важливі дані [1-2]. Ця велика кількість інформації потребує систематизації за різними критеріями, а також надійного зберігання та захисту від несанкціонованого доступу. Останнім часом спостерігається зростання кількості атак на інформаційні ресурси [3-5], що підкреслює важливість постійного вдосконалення технологій кібербезпеки та розробки нових методів протидії зловмисним діям. Поруч із цим, розвиток інфраструктури для забезпечення безпеки даних стає ключовим аспектом у забезпеченні інформаційної безпеки [6-10]. Значна частина цієї інформації зберігається на серверах та у хмарах, що належать таким відомим компаніям як Amazon, Google, Apple та Facebook [11-16]. Такому способу зберігання сприяють значні переваги, оскільки згадані компанії володіють штатом кваліфікованих фахівців, які забезпечують надійне обслуговування, а також ці компанії беруть на себе основні витрати, що пов’язані з безвідмовною роботою та хостингом.

Але, незважаючи на всі зручності, є великий мінус – вразливість. Зловмисники можуть отримати небажаний доступ до файлів користувача без його ж відома, атакуючи сторонній сервіс або впливаючи на нього, тобто вони можуть вкрасти, розкрити або змінити важливу інформацію [17-18].

Методи і технології віддалених мережевих атак постійно вдосконалюються, а існуючі алго-

ритми і системи шифрування не завжди повністю захищають конфіденційну інформацію. Проте розвиток блокчейн-технологій, які мають високу криптографічну стійкість, також не стоїть на місці.

Деякі дослідники стверджують, що Інтернет завжди був децентралізованим, і фрагментарний рух з'явився навколо використання нових інструментів, включаючи технологію блокчейн, для досягнення цієї мети. Ethereum є однією з найновіших технологій, яка приєдналася до цього руху, та має на меті використовувати блокчейн для заміни третіх сторін в Інтернеті – тих, які зберігають дані, передають іпотечні кредити, відстежують складні фінансові інструменти тощо.

Ethereum має потенціал стати «світовим комп'ютером», який децентралізує або навіть демократизує існуючу клієнт-серверну модель. З Ethereum сервери та хмари замінюються тисячами так званих «вузлів», якими керують добровольці з усього світу (таким чином утворюючи «світовий комп'ютер»).

Використовуючи блокчейн-платформу Ethereum, месенджери можуть стати максимально безпечними в роботі. Злом приватного ключа Ethereum неможливий навіть з сучасними технологіями, оскільки на його підбір піде колосальна кількість часу.

Децентралізований підхід зберігання даних в блокчейні гарантує невтручання третіх осіб, що не може бути забезпечено централізованими підходами до резервного копіювання інформації – хмарними сховищами, серверами. При цьому нема необхідності впроваджувати два типи чатів – хмарні та з наскрізним шифруванням. Усі повідомлення зашифровані одним типом шифрування, і доступ до них можна отримати з будь-якого пристрою в режимі реального часу. Відповідно, можна відновити всі повідомлення, фотографії, документи та інші файли, отримані та надіслані раніше.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ ТА ПУБЛІКАЦІЙ

Технологія Ethereum [19-20] дозволить використовувати одну і ту ж функціональність людям у всьому світі, що дасть змогу їм конкурувати за надання послуг поверх цієї інфраструктури.

Усі сучасні програми, від банківських до фітнес-додатків і додатків для обміну повідомленнями, покладаються на компанію (або іншу сторонню службу) для зберігання інформації про кредитні картки, історію покупок та інших особистих даних розташовану десь, зазвичай на серверах, контрольованих третіми сторонами. Ідея Ethereum полягає в тому, що одна організація більше не контролюватиме та раптово блокуватиме програму, що працює на технології блокчейн [21-22]. Вносити зміни може тільки користувач, а не будь-який інший об'єкт.

Ethereum поєднує в собі контроль над тим, якою інформацією обмінювалися люди в минулому, з легкістю доступу до інформації, до якої ми звикли в цифрову епоху. Щоразу, коли змінюється або додається чи видаляється інформація, кожен вузол мережі вносить зміни.

Ethereum – це адаптивна та гнучка блокчейн-платформа з відкритим вихідним кодом, яка дозволяє створювати та використовувати децентралізовані програми, що працюють на технології блокчейн. Ethereum є, мабуть, найбільш розвиненою, але складною системою, коли-небудь створеною.

Незважаючи на складність протоколу та механізми безпеки, які були розроблені на сьогоднішній день, повний вузол Ethereum складається з трьох основних частин:

– блокчейн;

– однорангова мережа (peer-to-peer);

– віртуальна машина.

Блокчейн Ethereum – це, по суті, транзакційна машина станів, яка працює за допомогою транзакцій. Визначення машини станів має на увазі, що цей механізм зчитує вхідні дані і на їх основі переходить в новий стан (рис. 1).

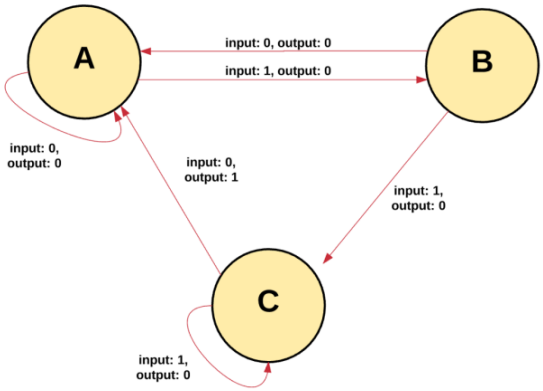

Рис. 1. Парадигма блокчейну Ethereum [23]

У випадку з машиною станів Ethereum відправною точкою є «стан генезису». Це як чистий аркуш або чистий бланк, поки в мережі не відбудуться будь-які транзакції. Після завершення транзакцій стан генезису переходить у кінцевий стан. У будь-який момент часу цей кінцевий стан є поточним станом Ethereum.

Стан Ethereum передбачає величезну кількість транзакцій. Ці транзакції групуються в «блоки». Блок містить групи транзакцій, і кожен блок

з'єднується з попереднім, тим самим утворюючи ланцюжок (рис. 2).

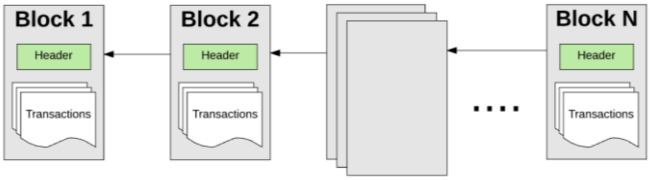

Рис. 2. Блоки транзакцій [23]

Для того, щоб змусити мережу перейти з одного стану в інший, транзакція повинна бути визнана дійсною, для чого вона повинна пройти процес валідації (перевірки та затвердження), відомий як майнінг. Майнінг – це процес, при якому група вузлів мережі (комп'ютерів) витрачає свої обчислювальні ресурси на генерацію блоку дійсних транзакцій.

Будь-який обчислювальний вузол мережі (їх ще називають «нодами»), який оголошує себе майнером, може претендувати на створення та перевірку блоку транзакцій.

Багато вузлів з усього світу намагаються створювати та перевіряти блоки одночасно. Кожен майнер при записі блоку в блокчейн надає математичний «доказ», який виступає в якості гарантії: якщо доказ існує, то блок повинен бути дійсним.

Для того, щоб додати блок до основного блокчейну, майнер повинен підтвердити його перед усіма своїми конкурентами. Процес верифікації кожного блоку шляхом надання вузлами математичного доведення називається proof-of-work [24].

Оскільки блокчейн є однотонним механізмом для запису транзакцій із загальним станом, правильний поточний стан є єдиною глобальною істиною, яку повинні прийняти всі. Наявність декількох станів (або ланцюжків) зруйнує всю систему, оскільки неможливо домовитися про те, який стан мережі слід вважати істинним. Кожен раз, коли формується кілька варіантів ланцюжків, формується розгалуження. Зазвичай розгалужень уникають, оскільки вони руйнують систему та змушують користувачів вибирати, якому ланцюжку їм довіряти.

ПОСТАНОВКА ЗАВДАННЯ

Основна мета даної роботи полягає у дослідженні сучасних методів зберігання та захисту інформації, з особливим акцентом на використанні технології блокчейн, зокрема платформи Ethereum. Потрібно дослідити можливості децентралізованих додатків, смарт-контрактів, а також ефективність і безпеку зберігання даних на блокчейні у порівнянні з традиційними методами, з метою виявлення найбільш надійних способів захисту від несанкціонованого доступу.

Для досягнення цієї мети необхідно виконати наступні задачі: провести аналіз платформи Ethereum, зокрема її архітектури, можливості смарт-контрактів та принципи роботи децентралізованих додатків; дослідити ефективність та безпеку платформи Ethereum порівнюючи її з традиційними методами зберігання даних на базі певних критеріїв включаючи вартість транзакцій і потенціал масштабування.

РЕАЛІЗАЦІЯ СИСТЕМИ

Створення ключів та їх використання для транзакцій

Створення Ethereum ID та використання його для надсилання в блокчейн. Ідентифікатор – це об'єкт із закритим ключем і відповідним відкритим ключем та його адресою. Для створення нового ідентифікатора викликається функція createIdentity, яка повертає його (рис. 3).

```
const EthCrypto = require('eth-crypto');

const identity = EthCrypto.createIdentity();

console.dir(identity);
/* > {
    address: '0x3f243FdacE01Cfd9719f7359c948A11361f32471',
    privateKey:
'0x107be946709e41b7895eea9f2dacf998a0a9124acbb786f0fd1a826101581a07',
    publicKey: 'bf1cc3154424dc22191941d9f4f50b063a2b663a2337e5548abea633c1d06ece...'
} */
```

Рис. 3. Створення нового ідентифікатора

Складові ідентифікатора:

Приватний ключ (privateKey), який ніколи нікому не повинен розголошуватися. Його можна використовувати для підпису та розшифрування повідомлень, а також для створення відкритого ключа (publicKey).

Відкритий ключ відкривається кожного разу, коли щось підписується за допомогою privateKey. Також поширеним є надсилання відкритого ключа іншим людям, щоб вони могли зашифрувати дані з його допомогою, які потім можна розшифрувати лише за допомогою правильного privateKey.

Існує два способи представлення publicKey: стисненим і нестисненим способом. EthCrypto завжди створює нестиснений ключ, який починається з 0x04.

Стислі клавіші починаються з 0x03 або 0x02. Для представлення ключа відокремимо від нього початок 04 і внутрішньо додаймо його при здійсненні криптографічних викликів.

Адреса обчислюється з останніх 20 байт хешу keccak-256 publicKey. Він використовується для представлення ідентифікатора. Немає можливості обчислити publicKey з адреси.

Це означає, що кожного разу, коли нам потрібно зашифрувати дані для когось, ми повинні спочатку отримати publicKey.

Існує два способи представлення адреси. Звичайна адреса пишеться в нижньому регістрі і представляє лише 20 байт хешу. Формат контрольної суми містить великі літери, які призначені для виявлення помилок при введенні адреси вручну.

Транзакція Ethereum – це, по суті, об'єкт json із певними значеннями (рис. 4)

```
const rawTransaction = {
        from: identity.address,
        to: '0x86Fa049857E0209aa7D9e616F7eb3b3B78ECfdb0',
        value: 1000000000000000000,

        nonce: 0,
        gasPrice: 5000000000,
        gasLimit: 21000
    };
```

Рис. 4. Створення транзакції

Перш ніж транзакція може бути відправлена на вузол, вона повинна бути підписана privateKey і переведена в шістнадцятковий рядок (рис. 5).

```
const serializedTx = EthCrypto.signTransaction(
        rawTransaction,
        identity.privateKey
    );
    console.log(serializedTx);
```

Рис. 5. Підписання за допомогою privateKey

Тепер рядок транзакції можна відправити в блокчейн. З метою тестування там створюється і перевіряється локальний тестовий ланцюжок.

Для створення локальної тестової мережі використовується ganache-cli та підключається до екземпляра web3 для взаємодії з ним (рис. 6).

```
const Web3 = require('web3');
const ganache = require('ganache-cli');


const web3 = new Web3();


const ganacheProvider = ganache.provider({

accounts: [{
secretKey: identity.privateKey,
balance: web3.utils.toWei('10', 'ether')
}]
});


web3.setProvider(ganacheProvider);
```

Рис. 6. Створення тестової локальної мережі

Функція sendSignedTransaction викликається для відправки підписаної транзакції в тестовий ланцюжок. Ganache негайно виконає транзакцію, і квитанція повернеться назад.

Щоб переконатися, що транзакція спрацювала, перевірю баланс адрес одержувачів (рис. 7).

```
const receipt = await web3.eth.sendSignedTransaction(serializedTx);
const balance = await
web3.eth.getBalance('0x86Fa049857E0209aa7D9e616F7eb3b3B78ECfdb0');
console.log(balance);
```

Рис. 7. Надсилання транзакції

Підпис та перевірка даних за допомогою Solidity

Підпис даних проводиться за допомогою JavaScript, а перевірка підпису в смарт-контракті Solidity. Спочатку створюємо два ідентифікатори: автора та одержувача. Після цього запускаємо локальну тестову мережу. У тестнеті надається creatorIdentity баланс в 10 Ефірів.

Також надано один Ефір одержувачу, тому є достатньо газу для відправки транзакцій. Перед тим як укласти контракт з локальним блокчейном, потрібно скомпілювати код Solidity в байт-код за допомогою JavaScript-версії компілятора solc (рис. 8).

```
const path = require('path');
const SolidityCli = require('solidity-cli');
const contractPath = path.join(__dirname, '../../contracts/DonationBag.sol');
const compiled = await SolidityCli.compileFile(contractPath);
const compiledDonationBag = compiled[':DonationBag'];

const createCode = EthCrypto.txDataByCompiled(
    JSON.parse(compiledDonationBag.interface),
    compiledDonationBag.bytecode,
    [creatorIdentity.address]
);

console.dir(compiledDonationBag);
```

Рис. 8. Компіляція в байт-код

Після того, коли отримано байт-код контракту, можна відправити транзакцію для створення нового екземпляра в локальному тестовому ланцюжку (рис. 9).

```
const receipt = await web3.eth.sendSignedTransaction(serializedTx);
const contractAddress = receipt.contractAddress;

console.log(contractAddress);
```

Рис. 9. Надсилання транзакції

Тепер контракт знаходиться в блокчейні. Щоб переконатися, що він розгорнутий правильно, можна викликати функцію.

Перш ніж можна буде підписати пожертви, потрібно надіслати певну цінність контракту.

Підписуючи повідомлення, не підписується адреса одержувача безпосередньо, а лише хеш, який складав деякі агреговані дані:

1. Prefix: щоб гарантувати, що автора не можна обдурити, випадково підписавши дійсну транзакцію Ethereum, замінюємо підписані дані чимось унікальним для нашої системи;

2. contractAddress: цілком можливо, що у творця є більше одного екземпляра контракту, завантаженого в блокчейн. При цьому підписи можуть бути відтворені в інших копіях. Щоб за-

побігти цій атаці, також додаю адреси контрактів до підписаного хешу;

3. receiverAddress: підписуючи цю адресу, творець доводить, що ця адреса має отримати пожертву.

Тепер одержувач має підпис від творця, який він може надіслати до контракту, щоб запросити пожертву. Якщо все пройшло правильно, у одержувача має бути більше Ефіру (рис. 10).

```
const receiverBalance = await web3.eth.getBalance(recieverIdentity.address);
console.dir(receiverBalance);
// '1999802840000000000'
```

Рис. 10. Перевірка балансу

Шифрування та підписання повідомлень

За допомогою ключів Ethereum можна не тільки взаємодіяти з блокчейном, але й використовувати їх для безпечного обміну повідомленнями через взаємні ненадійні канали. Для надсилання повідомлень будуть використовуватися ідентифікатори Ethereum. Спочатку створюємо два ідентифікатори, користувача A і користувача B. Користувач A відправляє повідомлення «Hello, B» (рис. 11).

```
const signature = EthCrypto.sign(
    A.privateKey,
    EthCrypto.hash.keccak256(secretMessage)
);
const payload = {
    message: secretMessage,
    signature
};
const encrypted = await EthCrypto.encryptWithPublicKey(
    B.publicKey, privateKey
    JSON.stringify(payload)
);

const encryptedString = EthCrypto.cipher.stringify(encrypted);
```

Рис. 11. Шифрування та підписання повідомлення

Коли одержувач приймає повідомлення, він починає з його розшифрування за допомогою свого приватного ключа, а потім перевіряє підпис (рис. 12).

```
const encryptedObject = EthCrypto.cipher.parse(encryptedString);

const decrypted = await EthCrypto.decryptWithPrivateKey(
    B.privateKey,
    encryptedObject
);
const decryptedPayload = JSON.parse(decrypted);

const senderAddress = EthCrypto.recover(
    decryptedPayload.signature,
    EthCrypto.hash.keccak256(payload.message)
);

console.log(

    senderAddress +
    ': ' +
    decryptedPayload.message
);
```

Рис. 12. Розшифрування

Тепер, коли користувач B отримав повідомлення, він також може відповісти. Для цього він повинен відновити відкритий ключ користувача A.

РОЗРАХУНОК ВАРТОСТІ ЗБЕРІГАННЯ

Gas – це внутрішня валюта мережі Ethereum, яка використовується для укладання транзакцій та контрактів [25]. Gas – це одиниця, яка вимірює кількість обчислювальних зусиль, які будуть потрібні для виконання певних операцій. Оплата за обчислення відбувається завжди, незалежно від того, відбулася транзакція чи ні. Навіть у випадках, коли транзакція відхилена, вузли повинні підтверджувати та виконувати обчислення. Тому оплата роботи вузлів відбувається незалежно від успішності угоди.

Ціна на Gas за обчислення або контракт налаштована для роботи з повним походженням Ethereum Тюрінга та його EVM (Ethereum Virtual Machine – віртуальної обчислювальної машини, розподіленого комп'ютера, що відповідає за обчислення в мережі). Ідея полягає в тому, щоб обмежити нескінченні цикли. Так, наприклад, 1 Gas може виконати рядок коду або якусь команду. Якщо на рахунку недостатньо Ефіру для здійснення транзакції або відправки повідомлення, то він вважається недійсним. Ідея полягає в тому, щоб зупинити атаки типу «відмова в обслуговуванні» з нескінченних циклів, підвищити ефективність коду та змусити зловмисника платити за ресурси, які він використовує, від пропускної здатності до обчислень процесора та зберігання даних.

Чим складніші команди, які потрібно виконати, тим більше Gas доведеться заплатити. Наприклад, якщо Користувач A хоче відправити Користувачеві B 1 Ефір – загальна сума в 1,00001 Ефір повинна бути сплачена Користувачем A. Однак, якщо A захоче украсти контракт з B в залежності від майбутньої ціни Ефір, буде більше рядків виконуваного коду і більше споживання енергії, розміщеного в розподіленій мережі Ефір, і, отже, Користувачеві A доведеться заплатити більше, ніж 1 Gas, для виконання транзакції.

Деякі обчислювальні кроки дорожчі за інші, або тому, що вони є дорогими обчислювальними кроками, або тому, що вони збільшують обсяг даних, які повинні зберігатися в блокчейні.

Апроксимація

Розрахувати ліміт Gas можна за такою формулою:

$$gasLimit = G_{transaction} + G_{txdatanonzero} * dataByteLenght, \quad (1)$$

де $G_{transaction}$ – це стандартна комісія за кожну транзакцію, що дорівнює 21000 Gas; $G_{txdatanonzero}$–

оплата за кожен ненульовий байт даних або код для транзакції. Ціна 68 Gas; $dataByteLenght$ – розмір даних у байтах. У даному випадку, щоб визначити прибутковість зберігання даних на блокчейні Ethereum, потрібно вирішити наступну задачу: яку ціну Gas необхідно встановити, щоб повідомлення вважалося доставленим за певний час. Тобто потрібно визначити ціну Gas, при якій швидкість підтвердження транзакції буде задовольняти умові «не більше N секунд». Взято графік часу підтвердження транзакції як функцію вартості Gas (табл. 1).

Таблиця 1

Ціна Gas і час підтвердження

Ціна Gas	2	3	4	5	6	7	8	9	10
Час підтвердження (в секундах)	961,8	869,1	473,1	465,1	132,4	8,5	8	2	1,1

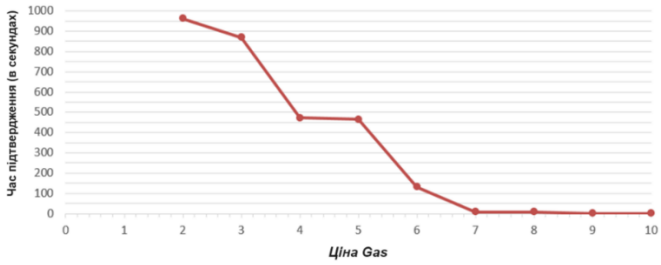

Рис. 13. Час підтвердження ціни Gas

Згідно з графіком, для того, щоб повідомлення було відправлено протягом 400 секунд, необхідно поставити вартість Gas на рівні 5,2. Метою є зробити розсилку якомога дешевшою. Взято кубічну регресію, яка дозволить визначити оптимальні витрати (рис. 13).

Використовуючи метод найменших квадратів, знайду кубічну функцію, за допомогою якої можна провести апроксимацію.

Рівняння регресії:

$$y = ax^3 + bx^2 + cx+d. \quad (2)$$

Коефіцієнти *a*, *b*, *c* і *d* можна знайти з розв'язку системи:

$$\begin{matrix} a & x_i^3 + b & x_i^2 + c & x_i + nd = & y_i, \\ a & x_i^4 + b & x_i^3 + c & x_i^2 + d & x_i = x_i y_i, \\ a & x_i^5 + b & x_i^4 + c & x_i^3 + d & x_i^2 = x_i^2 y_i, \\ a & x_i^6 + b & x_i^5 + c & x_i^4 + d & x_i^3 = x_i^3 y_i. \end{matrix} \quad (3)$$

Якщо опустити обчислення, то шукана функція виглядатиме наступним чином:

$$y = 2{,}3364x^3 + 22.021x^2 - 146{,}16x + 1356{,}1. \quad (4)$$

Намалюємо функцію на графіку і відстежимо вартість Gas протягом 400 секунд (рис. 14).

Як видно з рисунку, на відправку повідомлення можна витратити менше Gas – 5,1 замість 5,2. Звідси і економія. Далі буде проведена спроба здешевити транзакції (табл. 2). Можна помітити, що при вартості Gas 4,6 час підтвердження перевищує вказаний ліміт в 400 секунд.

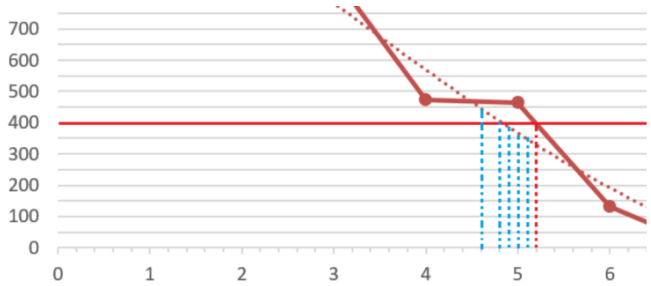

Рис. 14. Апроксимація

Таблиця 2

Ціна Gas після апроксимації

Ціна Gas	5,1	5	4,9	4,6
Час підтвердження (в секундах)	350	360	380	450

Зрештою, при всіх перевагах впровадженої системи, звичайний користувач не буде витрачати кошти на відправку того ж повідомлення в інших системах обміну повідомленнями. У розглянутому випадку за повідомлення доведеться заплатити, але при цьому дані будуть відправлятися і зберігатися максимально безпечно, а доступ до них можна буде отримати з будь-якого пристрою за допомогою єдиного облікового запису.

ВИСНОВКИ

Сучасний світ надзвичайно високо цінує інформацію як глобальний ресурс. Зберігання цієї інформації вимагає надійності та безпеки, а також можливості оперативного доступу до даних. Постійне зростання обсягів даних створює виклики, пов'язані з вибором надійних методів та способів зберігання, а також захисту цих даних від несанкціонованого доступу.

В роботі була досліджена платформа Ethereum для створення децентралізованих додатків, її смарт-контракти та принцип їх роботи. Ця технологія блокчейн є найбезпечнішою для зберігання даних, оскільки для злому приватного ключа Ethereum знадобляться століття. Завдяки цьо-

му месенджер на основі цієї технології позбавить від необхідності реалізовувати два типи чатів: секретні та хмарні. Крім того, гарантується невтручання третіх осіб і зловмисників в таємне листування. Користувачі можуть отримати доступ до раніше надісланих і отриманих файлів і повідомлень з будь-якого пристрою в режимі реального часу. Ця функція вкрай корисна в разі втрати або зміни смартфона, або при використанні одного і того ж облікового запису на різних пристроях.

В ході тестування цієї технології з'ясувалося, що зберігати дані в блокчейні хоча й дороге через досить високу вартість транзакцій, але є максимально надійним і безпечним.

Загалом варто зазначити, що блокчейн-технології, зокрема Ethereum, мають потенціал стати ключовим інструментом у захисті та збереженні інформації у майбутньому. Їх застосування може радикально змінити підходи до безпеки даних, пропонуючи рішення, які будуть важко зламати та легко масштабувати у глобальному масштабі

ЛІТЕРАТУРА

[1]. Олег Дейнека, Олег Гарасимчук. Дослідження проблем класифікації та безпечного зберігання даних // Безпека інформації. 2023. Т. 29, № 2. С. 147-153.

[2]. Олег Дейнека, Олег Гарасимчук. Виклики та стратегії зберігання великих обсягів даних у сучасному світі //Захист інформації, Том 25 № 4 (2023) С. 197-206.

[3]. B. Matturdi, X. Zhou, S. Li and F. Lin, "Big Data security and privacy: A review," in China Communications, vol. 11, no. 14, pp. 135-145, 2014, DOI: 10.1109/CC.2014.7085614.

[4]. Susukailo, V., Opirskyy, I., Vasylyshyn, S. Analysis of the attack vectors used by threat actors during the pandemic // 2020 IEEE 15th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2020 Proceedings, 2020, 2, pp. 261-264, 9321897. DOI: 10.1109/CSIT49958.2020.9321897.

[5]. Islam MN, Zaki T, Uddin MS, Hasan MM. Security threats for big data: An empirical study. Int J Inf Commun Technol Human Dev (IJICTHD). 2018; 10(4): pp. 1-18. DOI: 10.4018 / IJICTHD. 2018-100101.

[6]. A. Singh, A. Kumar, S. Namasudra: DNACDS: Cloud IoE big data security and accessing scheme based on DNA cryptography. Frontiers Comput. Sci. 18(1): 181801 (2024) DOI: 10.1007/s11704-022-2193-3.

[7]. O.I. Harasymchuk, Yu.M. Kostiv, V.M. Maksymovych, M.M. Mandrona. Generator of pseudorandom bit sequence with increased cryptographic security. Metallurgical and Mining Industry: scientific and technical jornal, Dnipropetrovsk. 2014. No. 5. pp. 25-29.

[8]. Available at: https: //www.metaljournal.com.ua/assets/ Journal / 6-KostivY.pdf (Accessed: 15 March 2024).

[9]. Lakhno V., Kozlovskii V., Boiko Y., Mishchenko A., Opirskyy I. "Management of information protection based on the integrated implementation of decision support systems" // Eastern-european journal of enterprise technologies. Information and controlling system. Vol 5, No 9(89), 2017. pp. 36-41. DOI: 10.15587/1729-4061.2017.111081.

[10]. Hulak, H., Kriuchkova, L., Skladannyi, P., & Opirskyy, I. (2021). Formation of requirements for the electronic record-book in guaranteed information systems of distance learning. Paper presented at the CEUR Workshop Proceedings, 2923 137-142. Available at: https: // ceur-ws.org/Vol-2923/paper-15.pdf (Accessed: 15 March 2024).

[11]. Maksymovych, V.; Shabatura, M.; Harasymchuk, O.; Shevchuk, R.; Sawicki, P.; Zajac, T. Combined Pseudo-Random Sequence Generator for Cybersecurity. Sensors 2022, 22, 9700. pp. 1-17. https:// doi.org /10.3390/s22249700.

[12]. Aujla GS, Chaudhary R, Kumar N, Das AK, Rodrigues JJ. SecSVA: secure storage, verification, and auditing of big data in the cloud environment. IEEE Commun Mag. 2018; 56(1): pp. 78-85.

[13]. Vyas J, Modi P. Providing confidentiality and integrity on data stored in cloud storage by hash and meta–data approach. Int J Adv Res Eng Sci Tech. 2017; 4: pp. 38-50.

[14]. Priyadarshini P., Veeramanju K. T., A Systematic Review of Cloud Storage Services – A Case Study on Amazon Web Services International Journal of Case Studies in Business, IT, and Education (IJCSBE), ISSN: 2581-6942, Vol. 6, No. 2, August 2022.

[15]. Pronika and S. S. Tyagi, "Secure Data Storage in Cloud using Encryption Algorithm," 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), Tirunelveli, India, 2021, pp. 136-141, DOI: 10.1109/ICICV50876.2021.9388388.

[16]. https://learn.microsoft.com/en-us/azure/compliance/offerings/offering-soc-2#azure-and-soc-2-type-2.

[17]. https://aws.amazon.com/compliance/soc-faqs/.

[18]. Reena, M. and Nargunam, A.S., 2019. Secured Storage of Big Data in Cloud. International Journal of Recent Technology and Engineering (IJRTE), 8 (2S3), pp.6-10.

[19]. Li, Yibin & Gai, Keke & Qiu, Longfei & Qiu, Meikang & Zhao, Hui. (2016). Intelligent Cryptography Approach for Secure Distributed Big Data Storage in Cloud Computing. Information Sciences. 387. 10.1016/j.ins.2016.09.005.

[20]. Onwubiko, Austine & Singh, Raman & Awan, Shahid & Pervez, Zeeshan & Ramzan, Naeem. (2023). Enabling Trust and Security in Digital Twin Management: A Blockchain-Based Approach with Ethereum and IPFS. Sensors (Basel, Switzerland). 23. DOI: 10.3390/s23146641.
[21]. Huang, Qichen. (2023). Ethereum: Introduction, Expectation, and Implementation. Highlights in Science, Engineering and Technology. 41. pp. 175-182. DOI: 10.54097/hset.v41i.6804.
[22]. Ren, Y.; Huang, D.; Wang, W.; Yu, X. BSMD: A blockchain-based secure storage mechanism for big spatio-temporal data. Future Gener. Comput. Syst. 2023, 138, pp. 328-338.
[23]. Ali, S., Wang, G., White, B., & Cottrell, R. L. (2018). A Blockchain-Based Decentralized Data Storage and Access Framework for PingER. 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/ 12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), New York, NY, USA, pp. 1303-1308. https: // doi.org/ 10.1109 / TrustCom/BigDataSE.2018.00179.
[24]. https://www.preethikasireddy.com/post/how-does-ethereum-work-anyway.
[25]. M. M. Arer, P. M. Dhulavvagol and S. G. Totad, "Efficient Big Data Storage and Retrieval in Distributed Architecture using Blockchain and IPFS," 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 2022, pp. 1-6, doi: 10.1109/I2CT54291.2022.9824566.
[26]. Koutmos D. Network Activity and Ethereum Gas Prices. Journal of Risk and Financial Management. 2023; 16(10):431. https: //doi.org/10.3390/jrfm16100431.

SECURE DATA STORAGE USING THE ETHEREUM BLOCKCHAIN TECHNOLOGY

The constant increase in the amount of data creates problems related to the choice of effective methods and means of storage and ensuring the protection of this data from unauthorized access. The article details the critical topic of preserving and protecting the importance of information in the context of growing data volumes and the number of cyberattacks. The impossibility of reliable data storage and protection is increasing, especially in the context of the increased threat from attackers. It highlights how blockchain technologies, especially based on the Ethereum platform, can solve the problems of reliable storage and data security. Ethereum offers an alternative to the traditional client-server model by decentralizing data storage through a distributed network of nodes. This technology significantly improves security, making unauthorized access to information difficult, and preventing the hacking of the private key, which requires significant computing resources. Smart contracts on Ethereum allow the creation of applications that execute precisely according to predetermined conditions, without the possibility of third-party intervention. This is especially important for messengers, where data privacy and availability are of fundamental importance. Although high, the cost of transactions in the blockchain is compensated by the high reliability and security of data storage. The applied methodology confirms that public (decentralized) data storage is safe since it is practically impossible to break the private key of Ethereum.

Keywords: cybersecurity, data storage, blockchain, Ethereum.

Гарасимчук Олег Ігорович, к.т.н., доцент, доцент кафедри захисту інформації Національного університету «Львівська політехніка».
Oleh Harasymchuk, Ph.D., Associate Professor of the Department of Information Security, National University "Lviv Polytechnic".
E-mail: oleh.i.harasymchuk@lpnu.ua.
Orcid ID: 0000-0002-8742-8872.

Наконечний Юрій Маркіянович, к.т.н., доцент, доцент кафедри захисту інформації Національного університету «Львівська політехніка».
Yurii Nakonechnyi, Ph.D., Associate Professor of the Department of Information Security, National University "Lviv Polytechnic".
E-mail: yurii.m.nakonechnyi@lpnu.ua.
Orcid ID: 0000-0002-6046-6190

Луковський Тарас Ігорович, к.т.н., старший викладач ЗВО кафедри захисту інформації Національного університету «Львівська політехніка».
Taras Lukovskyy, Ph.D., Senior Lecturer of the Department of Information Security, National University "Lviv Polytechnic".
E-mail: taras.i.lukovskyi@lpnu.ua.
Orcid ID: 0009-0008-1652-8121.

Андріїв Роман Ростиславович, асистент кафедри, викладач кафедри управління інформаційною безпекою Львівського державного університету безпеки життєдіяльності.
Roman Andriiv, lecturer of the Department of Information Security Management of Lviv State University of Life Safety.
E-mail: r.andriiv@ldubgd.edu.ua.
Orcid ID: 0009-0001-2450-5439.

Наконечний Тарас Ігорович, аспірант, спеціальності «Кібербезпека та захист інформації» Національного університету «Львівська політехніка».
Taras Nakonechnyi, Postgraduate the Department of Information Security, National University "Lviv Polytechnic".
E-mail: aras.i.nakonechnyi@lpnu.ua.
Orcid ID: 0009-0003-4487-9424.