Список використаних джерел

1. Історія штучного інтелекту. URL: https://uk.wikipedia.org/wiki/Історія_штучного_інтелекту

2. Перспективи використання штучного інтелекту у сфері медицини. URL: https://ena.lpnu.ua:8443/server/api/core/bitstreams/3e2bd0d2-cd07-4896-936d-771be7934e94/content

3. Sustainable Energy, Grids and Networks. URL: https://www.sciencedirect.com/journal/sustainable-energy-grids-and-networks

4. Вчитель + штучний інтелект – це майбутнє освіти. URL: https://intboard.ua/pres-sluzhba/blog/vchytel-shtuchnyy-intelekt-tse-maybutnye-osvity/

УДК 004.43:004.056.5

Мигун Р. С., здобувач 1 курсу спеціальності 122 Комп'ютерні науки,
Горяшин А. С., асистент кафедри інформаційних технологій

РЕАЛІЗАЦІЯ БЛОКЧЕЙН-ТЕХНОЛОГІЙ З ВИКОРИСТАННЯМ PYTHON ДЛЯ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ТА ЦІЛІСНОСТІ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

Блокчейн, або децентралізований цифровий реєстр – це особливий вид бази даних, який підтримується численними комп'ютерами, розміщеними в усьому світі. Дані блокчейну організовані в блоки, які розташовані в хронологічному порядку і захищені криптографією [1].

Ідея блокчейн-технології була описана ще в 1991 р., коли вчені-дослідники С. Хабер та В. Скотт Сторнетта запропонували обчислювально-практичне рішення для тимчасового маркування цифрових документів, щоб їх не можна було змінити або підробити. Система використовувала криптографічно захищений ланцюг блоків для зберігання документів із позначками часу, і у 1992 р. в конструкцію були додані дерева Меркла, що зробило її ефективнішою та дало змогу збирати кілька документів в один блок. Однак ця технологія не використовувалася, а патент втратив чинність у 2004 р., за чотири роки до появи Bitcoin [2].

Основні переваги блокчейну, які можна виділити:

Децентралізація – Блокчейн розподілений між багатьма вузлами, тому він більш стійкий до атак і витоку даних.

Прозорість – транзакції у блокчейні видні всім учасникам, що спрощує відстеження і перевірку транзакцій, а також забезпечує їх точність.

Незмінність – блокчейн-транзакції незмінні, тому їх можна перевірити та відстежити. Це відрізняється від традиційних систем, де транзакції можна змінити або видалити.

Ефективність – блокчейн-транзакції швидші, бо їх не потрібно перевіряти посередникам.

Система «без довіри» – блочейн-транзакції публічні та надійні, бо їх перевіряють і підтверджують усі учасники мережі, а не один посередник.

Алгоритм консенсусу – це принципи роботи блокчейну, які забезпечують безпеку мережі та узгодженість даних між вузлами. Це необхідно для підтримки цілісності та безпеки блокчейну. Завдяки цим принципам система не потребує адміністраторів та центральних сховищ. Алгоритми консенсусу підтверджують правильність інформації у кожному блоці системи.

Розробники пропонували безліч алгоритмів консенсусу, але найпопулярнішими стали ті, основу яких лежить:

Доказ роботи – Proof-of-Work або PoW;

Підтвердження частки – Proof-of-Stake або PoS [3].

Реальні приклади блокчейну

Відповідаючи на запитання «Які існують приклади блокчейну?», розглянемо реальні випадки впровадження технології блокчейн у різних галузях.

Фінансовий сектор: Ripple – це платіжна система, яка дає змогу здійснювати міжнародні перекази з низькими комісіями та миттєвими операціями.

Охорона здоров'я: MediChain – це платформа для зберігання медичних записів. Забезпечує безпеку та доступність даних для лікарів та пацієнтів.

Логістика: IBM Food Trust – це система, що пропонує прозорий та ефективний контроль відстеження продуктів харчування вздовж ланцюга постачання.

Що таке розподілена база даних? Найпростіший приклад пристрою блокчейн-сховища – величезна таблиця даних, яка мільйони разів дублюється на різних пристроях мережі. А ще в такій системі кожну хвилину оновлюються дані. Інформація, яка зберігається в ланцюжку блоків, – загальнодоступна. Вона постійно звіряється на основі тисяч дублів. У такого формату є безліч істотних переваг. Дані не містяться в якомусь одному місці, вони публічні й доступні. Немає централізованого сховища, яке можна було б зламати або пошкодити [4].

Давайте спробуємо розгорнути свій блокчейн на мові програмування python. Цей код зроблений для ознайомлення, на більш відповідальних проєктах розробка блокчейну робиться не за 1 місяць [5].

```
# -*- coding: utf-8 -*-

from hashlib import sha256
import json
```

```
from time import time

class Block:

    def __init__(self, timestamp=None, data=None):
        self.timestamp = timestamp or time()
        # У this.data повинна зберігатися інформація, на кшталт відомостей про транзакції.
        self.data = [] if data is None else data
        self.prevHash = None # Хеш минулого блока
        self.nonce = 0
        self.hash = self.getHash()

    def getHash(self):

        hash = sha256()
        hash.update(str(self.prevHash).encode('utf-8'))
        hash.update(str(self.timestamp).encode('utf-8'))
        hash.update(str(self.data).encode('utf-8'))
        hash.update(str(self.nonce).encode('utf-8'))
        return hash.hexdigest()

    def mine(self, difficulty):
        # Тут запускається цикл, що працює до тих пір, поки хеш не буде починатися з рядка
        # 0...000 довжини <difficulty>.
        while self.hash[:difficulty] != '0' * difficulty:
            print("process")
            # Інкрементуємо nonce, що дає змогу отримати абсолютно новий хеш.
            self.nonce += 1
            # Перераховуємо хеш блоку з урахуванням нового значення nonce.
            self.hash = self.getHash()

class Blockchain:

    def __init__(self):
        # У цій властивості будуть утримуватися всі блоки.
        self.chain = [Block(str(int(time())))]
        self.difficulty = 1
        self.blockTime = 30000

    def getLastBlock(self):
        return self.chain[len(self.chain) - 1]

    def addBlock(self, block):
            # Оскільки ми додаємо новий блок, prevHash буде хешем попереднього останнього блоку.
        block.prevHash = self.getLastBlock().hash
        # Оскільки тепер в prevHash є значення, ми повинні перерахувати хеш блоку.
```

```
            block.hash = block.getHash()
            block.mine(self.difficulty)
            self.chain.append(block)

                self.difficulty += (-1, 1)[int(time()) - int(self.getLastBlock().timestamp) < self.blockTime]

        def isValid(self):
            # Перед перебором ланцюжка блоків необхідно встановити i в 1, оскільки до первинного блоку жодних блоків немає. В результаті ми починаємо з другого блоку.
            for i in range(1, len(self.chain)):
                currentBlock = self.chain[i]
                prevBlock = self.chain[i - 1]

                # Перевірка
                    if (currentBlock.hash != currentBlock.getHash() or prevBlock.hash != currentBlock.prevHash):
                        return False

            return True

        def __repr__(self):
                return json.dumps([{'data': item.data, 'timestamp': item.timestamp, 'nonce': item.nonce, 'hash': item.hash, 'prevHash': item.prevHash} for item in self.chain], indent=4)
```

Тепер після того, як у нас є каркас блокчейну, ми можемо в ньому зберігати різну інформацію, наприклад:

```
JeChain = Blockchain()

# Додамо новий блок
JeChain.addBlock(Block(str(int(time())), ({"from": "John", "to": "Bob", "amount": 1001})))

# Вивід оновленого блокчейна
print(JeChain)
```

Після запуску коду ми зможемо побачити приблизно ось таку інформацію:

```
[
    {
        "data": [],
        "timestamp": "1700350466",
        "nonce": 0,
        "hash":
"60f3cef058fab68497d70e80df4c22970d5919e2787a1d53f6619929c6a60e13",
        "prevHash": null
    },
```

```
        {
          "data": {
            "from": "John",
            "to": "Bob",
            "amount": 1001
          },
          "timestamp": "1700350466",
          "nonce": 16,
          "hash":
"0c5ddae74fbdb8203add0b95cb784006776550fed7acef8022d445f96d5f50f7",
          "prevHash":
"60f3cef058fab68497d70e80df4c22970d5919e2787a1d53f6619929c6a60e13"
        }
      ]
```


Висновки. Отже, блокчейн – це децентралізована база даних, яка використовує криптографію для забезпечення безпеки та цілісності даних. Це робить його привабливим для широкого спектра застосувань, як-от фінансові послуги, охорона здоров'я та логістика. Блокчейн-транзакції незмінні, що робить їх прозорими та надійними. Це може допомогти зменшити ризик шахрайства та зловживань. Алгоритми консенсусу забезпечують безпеку мережі блокчейну та узгодженість даних між вузлами. Це робить блокчейн більш стійким до атак і витоку даних. Загалом блокчейн є потужною технологією, яка має потенціал змінити спосіб, яким ми зберігаємо та обмінюємося даними.

Список використаних джерел

1. Що таке блокчейн і як він працює? URL: https://academy.binance.com/uk/articles/what-is-blockchain-and-how-does-it-work
2. Історія блокчейну. URL: https://academy.binance.com/uk/articles/history-of-blockchain
3. Що таке блокчейн? Пояснюємо простими словами. URL: https://blog.whitebit.com/ uk/what-is-blockchain-technology/
4. Blockchain. URL: https://astwellsoft.com/uk/blog/blockchain.html