Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Аналіз та створення NFT за допомогою біткоїн-інскрипцій

(тема)

Виконав:
здобувач 4 року навчання,
групи КІУКІ-21-2
Матвій ЖИГАЛКА
(власне ім'я, прізвище)

Спеціальність
123 «Комп'ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма
Комп'ютерна інженерія
(повна назва освітньої програми)

Керівник: ас. Віталій СІТНІКОВ
(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ _____ Андрій КОВАЛЕНКО
(підпис) (власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти перший (бакалаврський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Комп’ютерна інженерія
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві Жигалці Матвію Ігоровичу
(прізвище, ім’я, по батькові)

1. Тема роботи Аналіз та створення NFT за допомогою біткоїн-інскрипцій

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії 17 червня 2025 р.

3. Вхідні дані до роботи 1) блокчейн; 2) bitcoin; 3) інскрипції; 4) nft.

4. Перелік питань, що потрібно опрацювати у роботі

1) принцип побудови транзакцій та структура блокчейну;
2) розвиток протоколу Ordinals та концепція інскрипцій;
3) технологічні особливості створення NFT у мережі Bitcoin;
4) реалізація інскрипцій у тестовій мережі Signet;
5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій

Слайд-презентація – 17

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Аналіз літератури та огляд існуючих рішень	28.05.25-30.05.25	
2	Вибір технології реалізації інскрипцій	31.05.25-02.06.25	
3	Розгортання повної ноди Bitcoin Core	03.06.25-07.06.25	
4	Практична реалізація процесу інскрипції	07.06.25-10.06.25	
5	Тестування створених inscription-транзакцій	11.06.25-13.06.25	
6	Оформлення пояснювальної записки	13.06.25-14.06.25	
7	Подання кваліфікаційної роботи керівникові	14.06.25-15.06.25	
	та підготовка до попереднього захисту		
8	Подання кваліфікаційної роботи на	15.06.25-16.06.25	
	рецензування		

Дата видачі завдання " 26 " травня 2025 р.

Здобувач _____
(підпис)

Керівник роботи _____ ас. Віталій СІТНІКОВ
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 62 с., 20 рис., 1 дод., 16 джерел.

BITCOIN, БЛОКЧЕЙН, КРИПТОГРАФІЯ, HASH-ФУНКЦІЯ, ПРОТОКОЛ ORDINALS, NFT, TAPROOT, SEGWIT, SATOSHI, INSCRIPTION, SIGNET, BITCOIN CORE.

Метою кваліфікаційної роботи є дослідження інноваційного підходу до створення цифрових активів у мережі Bitcoin шляхом реалізації інскрипцій на основі протоколу Ordinals.

У ході виконання кваліфікаційної роботи буде проаналізовано принципи функціонування блокчейну Bitcoin, криптографічні основи безпеки, а також технічну реалізацію NFT у мережах Bitcoin, Ethereum і Solana. Особливу увагу буде приділено впровадженню оновлень SegWit та Taproot, які дозволили зберігати довільні дані безпосередньо в блокчейні. У тестовому середовищі Signet буде розгорнуто повноцінну ноду Bitcoin Core, а також буде проведено практичне створення інскрипції з урахуванням технічних обмежень мережі. Робота має на меті продемонструвати практичну цінність і перспективи ончейн-зберігання цифрових об’єктів у Bitcoin як нової форми цифрової власності.

ABSTRACT

Bachelor's thesis: 62 pages, 20 figures, 1 appendices, 16 sources.

BITCOIN, BLOCKCHAIN, CRYPTOGRAPHY, HASH FUNCTION, ORDINALS PROTOCOL, NFT, TAPROOT, SEGWIT, SATOSHI, INSCRIPTION, SIGNET, BITCOIN CORE.

The major goal of this thesis is research on an innovative approach to creating digital assets in the Bitcoin network through the implementation of inscriptions based on the Ordinals protocol.

In order to explore innovative approaches to creating digital assets within the Bitcoin network, this qualification thesis will analyze the fundamental principles of Bitcoin blockchain operation, the cryptographic foundations of its security, and the technical implementation of NFTs in Bitcoin, Ethereum, and Solana ecosystems. Special emphasis will be placed on the SegWit and Taproot upgrades, which will enable the embedding of arbitrary data directly into the blockchain. A full Bitcoin Core node will be deployed in a Signet test environment, and a practical inscription will be carried out with regard to the network's technical limitations. This research will demonstrate the practical applicability and promising future of on-chain digital object storage in Bitcoin as a novel form of digital ownership.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – інтерфейс прикладного програмування (англ., Application Programming Interface)

CLI – інтерфейс командного рядка (англ., Command Line Interface)

GUI – графічний інтерфейс користувача (англ., Graphical User Interface)

MAST – змерклюване дерево абстрактного синтаксису (англ., Merklized Abstract Syntax Tree)

NFT – невзаємозамінний токен (англ., Non-Fungible Token)

P2P – однорангова мережа (англ., Peer-to-Peer)

RPC – віддалений виклик процедур (англ., Remote Procedure Call)

SHA – алгоритм безпечного хешування (англ., Secure Hash Algorithm)

UTXO – Unspent Transaction Output (англ., невитрачений вихід транзакції)

ВСТУП

Цифрові технології дедалі глибше проникають у фінансову, правову та культурну сфери. Одним із найяскравіших проявів цієї трансформації є поява та поширення блокчейн-технологій. З-поміж усіх реалізацій блокчейну ключове місце посідає Bitcoin - перша у світі децентралізована цифрова валюта, яка, завдяки алгоритму Proof of Work, забезпечує безпечне й безпосереднє передавання вартості без участі посередників.

Спочатку мережа Bitcoin була орієнтована виключно на фінансові транзакції. Однак згодом, після впровадження технічних оновлень (зокрема SegWit у 2017 році та Taproot у 2021 році), з’явилася можливість додавати до транзакцій довільні дані у спеціально відведені поля, зберігаючи при цьому відповідність консенсусу мережі. Ці зміни стали основою для нового класу застосувань блокчейну Bitcoin, зокрема - для реалізації інскрипцій.

У 2023 році було представлено протокол Ordinals, який започаткував нову парадигму у розвитку блокчейну Bitcoin. Його суть полягає у нумерації кожного сатоші (найменшої одиниці біткоїна) за порядком створення та можливості прикріплення до нього унікального цифрового контенту. Завдяки цьому кожен сатоші може бути перетворений на невзаємозамінний цифровий об’єкт, що функціонує аналогічно до NFT, але без використання смарт-контрактів і створення додаткових токенів.

Цей підхід викликав жваву дискусію у спільноті: з одного боку, Ordinals відкривають нові можливості для цифрового мистецтва, збереження історичних артефактів, архівування даних у незмінному вигляді; з іншого - породжують питання щодо масштабованості, економічної доцільності та відповідності філософії Bitcoin.

1 ОСНОВИ БЛОКЧЕЙНУ БІТКОЇН

1.1 Історія створення Bitcoin

Історія появи Bitcoin починається з недовіри до уряду у поєднанні з фінансовим крахом 2007 - 2008 року, який можна порівняти хіба що лише з Великою депресією 1930-х років. Криза виявила слабкі сторони традиційної банківської системи та показала потребу в альтернативній фінансовій системі, непідконтрольній централізованим структурам - банкам та іншим комерційно-кредитно-фінансовим організаціям.

У жовтні 2009 року Сатоші Накамото опублікував статтю «Bitcoin: Peer-to-Peer Electronic Cash System». У січні 2009 року біткойн вперше став доступним для громадськості. Сатоші Накамото, побачив потенціал використання децентралізованої мережі для створення цифрової валюти, несприйнятливої до державного втручання та маніпуляцій. У документі стверджується, що метою біткойна є створення валюти, яка дозволяє надсилати платежі від однієї сторони до іншої без проходження через фінансову установу.

Крім того, він цікавився криптографією, що частково пояснює, чому саме ця технологія була ключовим елементом системи. Система пропонує користувачам повну прозорість - кожен бажаючий може переглянути інформацію про всі транзакції в мережі, а рівні права користувачів мережі позбавляють необхідності довіряти третім особам.

Натомість, Bitcoin був розроблений таким чином, що користувачі можуть обмінюватися засобами один з одним безпосередньо через однорангову мережу. Це мережа, де всі користувачі мають рівні повноваження і пов'язані один з одним безпосередньо, без центрального сервера або компанії-посередника.

Мережа є повністю публічною, тобто будь-яка людина у світі, яка має

підключення до Інтернету та пристрій, який може до нього підключитися, може брати участь у ній без обмежень.

Вона також має відкритий вихідний код, тобто будь-який може переглянути або поділитися вихідним кодом, на основі якого було створено Bitcoin.

Загалом систему можна уявити як Інтернет, але для грошей. Інтернет є повністю цифровим, ним не володіє та не контролює жодна людина. Мережа не має кордонів, будь-хто, хто має електрику та пристрій, може підключитися до нього, вона працює цілодобово, і люди, які її використовують, можуть легко обмінюватися даними між собою.

Сатоші Накамото - це псевдонім однієї людини, групи осіб, що стоять за першими кодами. Були докладені великі зусилля, щоб розкрити особу Накамото. Особа досі невідома громадськості.

1.2 Криптографія

Криптографія - це глибока академічна область досліджень, що використовує безліч передових математичних методів, які, як відомо, тонкі і складні. На щастя, біткоїн спирається лише кілька відносно простих і добре відомих криптографічних конструкцій.

Перший криптографічний примітив - криптографічна хеш-функція. Хеш-функція - це математична функція з такими трьома властивостями:

- Вхідними даними може бути будь-який рядок будь-якого розміру;
- Він робить вихід фіксованого розміру;
- Він ефективно обчислюваний.

Ці властивості визначають загальну хеш-функцію, яку можна використовувати для побудови структури даних, наприклад, хеш-таблиці. Щоб хеш-функція була криптографічно безпечною, ми вимагаємо, щоб вона мала наступні три додаткові властивості, а саме стійкість до колізій, скритність та дружність до головоломок.

1.2.1 Стійкість до колізій

Перша властивість, яка нам потрібна від криптографічної хеш-функції, це її стійкість до колізій. Колізія відбувається, коли два різні входи виробляють однаковий вихід. Хеш-функція стійка до колізій, якщо ніхто не може знайти колізію. Зображено на рисунку 1.1.

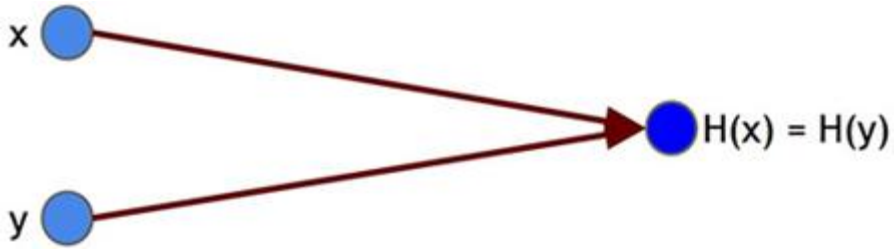


Рисунок 1.1 – Колізія хешів

Таким чином, хеш-функція стійка до колізії, коли ймовірність її виявлення настільки мала, що для цього знадобляться мільйони років обчислень. можуть вважатися стійкими (наприклад, SHA-256).

Серед різних алгоритмів SHA групи SHA-0 і SHA-1 більше не є безпечними, оскільки в них були виявлені колізії.

1.2.2 Скритність прообразу

Друга властивість, яку ми хочемо отримати від хеш-функцій, - це приховання. Приховування означає, що якщо ми знаємо результат роботи хеш-функції y = H(x), то ми не маємо реальної можливості визначити, що було на вході, тобто значення x.

Однак якщо можливі значення x заздалегідь відомі та обмежені, зловмисник може спробувати хешувати кожне з них по черзі і порівнювати результат з y, поки не знайде збіг. Таким чином він зможе визначити, яке значення x було на вході, і хеш-функція не захищатиме вихідні дані від розкриття.

Щоб гарантувати властивість приховування, важливо, щоб не було значення x, яке зустрічалося б з великою ймовірністю. Тобто, значення x має вибиратися з набору, в якому воно практично непередбачуване. У цьому випадку метод підбору стає неефективним, тому що ймовірність «відгадати» дуже мала. Для покращення приховування можна додатково використовувати метод, який приховає навіть передбачувані вхідні дані.

1.2.3 Дружність до головоломок

Хеш-функція називається доброзичливою до головоломок, якщо для будь-якого n-бітного значення y, коли k вибирається з розподілу з високою мінімальною ентропією, неможливо знайти x, таке що H(k || x) = y швидше, ніж за час, близьке до 2^n.

Правильна хеш-функція влаштована так, що, якщо потрібно знайти вхідні дані, які дадуть конкретний результат, то не можна придумати алгоритм, який був би кращим за випадковий перебір. Єдиний спосіб розв'язання такого завдання – це випадковий перебір вхідних даних. Чим більше можливих варіантів, тим складнішим і довшим буде пошук. Тож для якісної хеш-функції добір значень, які призводять до певного результату, має вимагати значних часових та обчислювальних ресурсів.

1.2.4 SHA-256

Одним із популярних алгоритмів, що використовуються в Bitcoin, є SHA-256 із сімейства SHA-2.[2] Він перетворює вхідні дані довільної довжини на хеш фіксованої довжини, стійкий до колізій, за допомогою перетворення Меркла-Дамгарда. Це перетворення приймає фіксовану функцію стиснення, стійку до колізій, і робить її здатною працювати з входами довільної довжини.

Перетворення Меркла-Дамгарда використовує функцію стиснення, яка

приймає вхідні дані довжиною m та видає вихід меншої довжини n. Вхідне повідомлення ділиться на блоки і кожен блок обробляється функцією стиснення. Для першого блоку використовується початкове значення (вектор ініціалізації), а результат останнього блоку це підсумковий хеш. Зображенно на рисунку 1.2.

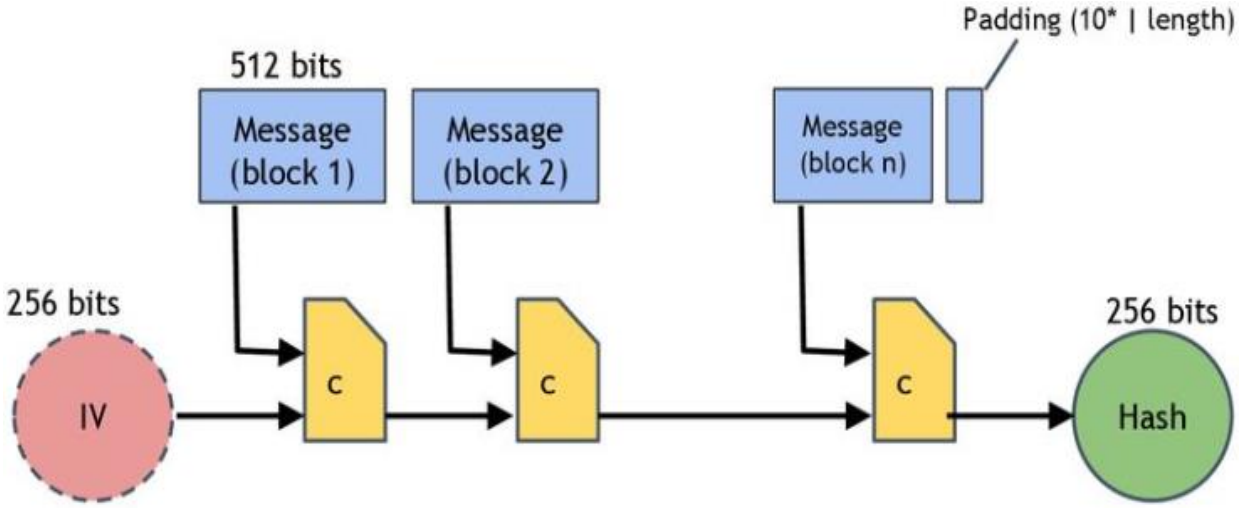


Рисунок 1.2 – Хеш-функція SHA-256

SHA-256 має:

- Функцію стиснення, односторонню функцію яка перетворює більші дані на менші та робить зворотне обчислення вихідних даних складним;

- Вектор ініціалізації (IV) фіксоване початкове значення першого блоку;

- Зміцнення Меркла-Дамгарда додатковий етап, який робить конструкцію безпечнішою, додаючи до кінця повідомлення його довжину та спеціальні символи.

Перетворення Меркла-Дамгарда та доповнення повідомлення гарантують, що довжина повідомлення завжди кратна розміру блоку, запобігаючи ймовірності простих колізій хешів.

1.3 Принцип роботи блокчейну Bitcoin

1.3.1 Транзакції

Сатоші Накамото, творець біткоїну, визначив електронну монету як ланцюжок цифрових підписів[1]. Кожен власник передає монету наступному, підписуючи цифровим підписом хеш попередньої транзакції та відкритий ключ наступного власника та додаючи їх у кінець монети. Одержувач платежу може перевірити підписи, щоб перевірити ланцюжок володіння. Зображенно на рисунку 1.3.

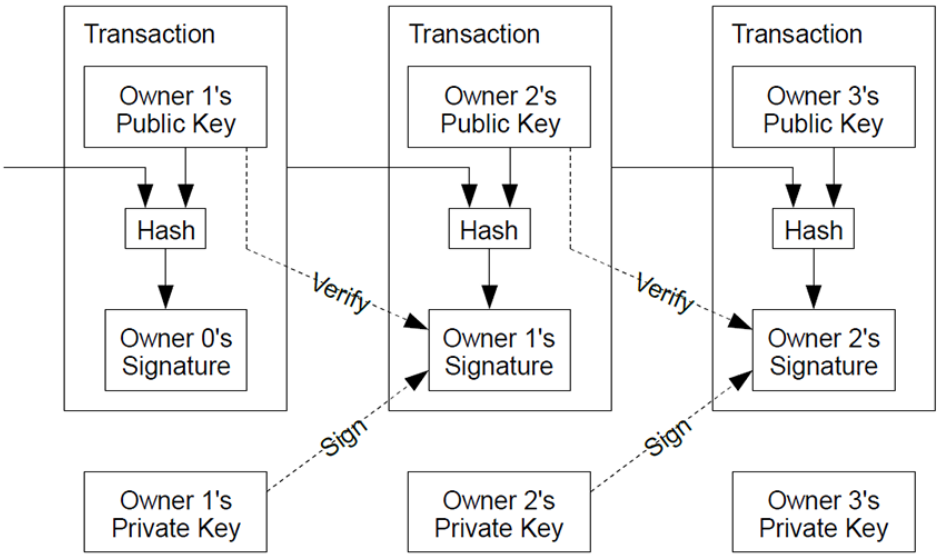


Рисунок 1.3 – Електронна монета як ланцюжок цифрових підписів

Проблема, звичайно, полягає в тому, що одержувач платежу не може перевірити, що один із власників не витратив монету двічі. Поширеним рішенням є запровадження довіреного центрального органу, або монетного двору, який перевіряє кожну транзакцію щодо подвійної витрати. Після кожної транзакції монета має бути повернута до монетного двору для випуску нової монети, і тільки монети, випущені безпосередньо монетним двором, вважаються невикористаними двічі. Проблема з цим рішенням

полягає в тому, що доля всієї грошової системи залежить від компанії, яка управляє монетним двором, і кожна транзакція має проходити через них, як у банку.

Нам потрібен спосіб, щоб одержувач платежу знав, що попередні власники не підписували жодних ранніх транзакцій. Для наших цілей враховується рання транзакція, тому нас не цікавлять подальші спроби подвійної витрати. Єдиний спосіб підтвердити відсутність транзакції - знати про всі транзакції. У моделі, заснованої на монетному дворі, монетний двір знав про всі транзакції та вирішував, яка з них надійшла першою. Щоб досягти цього без довіреної сторони, транзакції мають бути публічно оголошені, і нам потрібна система, щоб учасники погодили єдину історію порядку їх отримання. Одержувачу платежу необхідний доказ того, що на момент кожної транзакції більшість вузлів погодилися, що її було отримано першою.

Рішення, запропоноване Накамото, починається з сервера тимчасових міток. Сервер тимчасових міток працює, беручи хеш блоку елементів, яким потрібно присвоїти тимчасову мітку, і широко публікуючи хеш. Кожна тимчасова мітка включає в свій хеш попередню мітку, утворюючи ланцюжок, в якій кожна додаткова тимчасова мітка посилює попередні. Зображенно на рисунку 1.4.

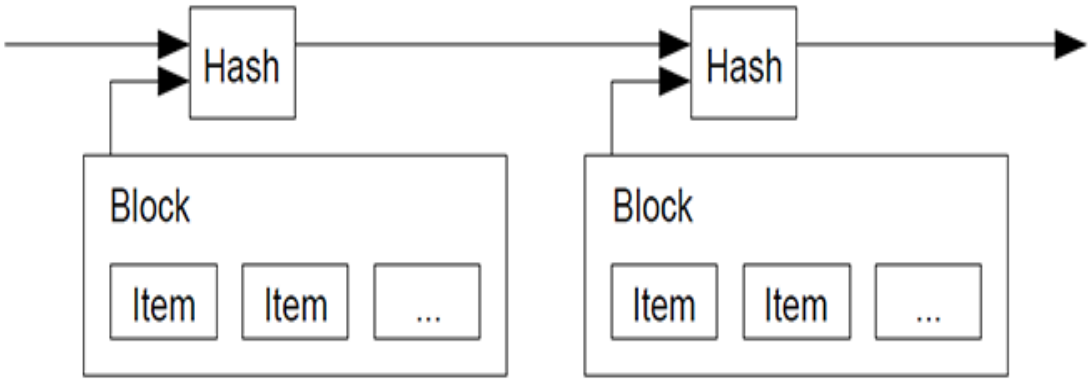


Рисунок 1.4 – Отримання хешу блоку елементів і хешу попередньої мітки для тимчасової мітки

1.3.2 Доказ роботи

Для реалізації розподіленого сервера тимчасових міток на основі однорангової мережі нам потрібно буде використовувати систему доказу роботи. Доказ роботи включає сканування значення, яке при хешуванні, наприклад, за допомогою SHA-256 починається з деякої кількості нульових біт[4]. Середня необхідна робота експонентно залежить від необхідної кількості нульових біт і може бути перевірена шляхом виконання одного хеша.

Для нашої мережі тимчасових міток ми реалізуємо доказ роботи шляхом збільшення одноразового номера в блоці доти, доки не буде знайдено значення, яке дасть хешу блоку потрібні нульові біти. Після того, як зусилля ЦП були витрачені на те, щоб він задовольняв доказ роботи, блок не можна змінити без повторного виконання роботи. Оскільки пізніші блоки вишиковуються в ланцюжок після нього, робота зі зміни блоку включатиме повторну обробку всіх блоків після нього. Зображенно на рисунку 1.5.

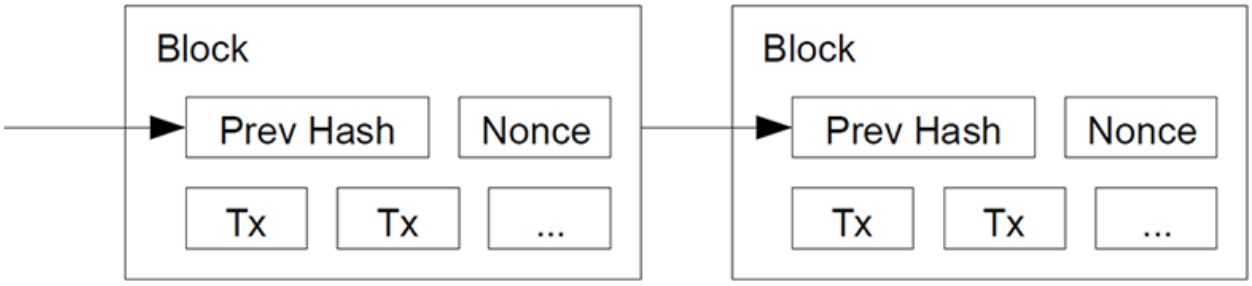


Рисунок 1.5 – Nonce як доказ роботи

За угодою перша транзакція в блоці - це особлива транзакція, яка запускає нову монету, що належить творцю блоку. Це додає стимул для вузлів підтримувати мережу і забезпечує спосіб початкового розподілу монет в обігу, оскільки немає центрального органу їхнього випуску. Постійне додавання постійної кількості нових монет аналогічне золотовидобувачам, які витрачають ресурси для додавання золота в обіг. У нашому випадку

витрачаються процесорний час та електрика.Стимул також може фінансуватись за рахунок комісій за транзакції. Якщо вихідне значення транзакції менше вхідного значення, різниця є комісією за транзакцію, яка додається до значення стимулу блоку, що містить транзакцію. Після того, як задана кількість монет надійде в обіг, стимул може повністю перейти на комісії через транзакцію та стати повністю вільним від інфляції.

Стимул може допомогти спонукати вузли залишатися чесними. Якщо жадібний зловмисник зможе зібрати більше потужності процесора, ніж усі чесні вузли, йому доведеться вибирати між використанням її для обману людей шляхом крадіжки їхніх платежів або її використанням для генерації нових монет. Він мав би знайти більш вигідним грати за правилами, такими правилами, які сприяють йому з великою кількістю нових монет, ніж усі інші разом узяті, ніж підривати систему та обґрунтованість свого власного багатства.

1.3.3 Механізм консенсусу

Доказ роботи також вирішує проблему визначення представництва у прийнятті рішень більшістю. Якби більшість ґрунтувалася на принципі одна IP-адреса-один голос, її міг би підірвати будь-хто, хто може виділити багато IP-адрес.Доказ роботи насправді є одним ЦП-одним голосом. Рішення більшості представлено найдовшим ланцюжком, в який вкладено найбільшу кількість зусиль за доказом роботи. Якщо більшість потужності ЦП контролюється чесними вузлами, чесна ланцюжок зростатиме найшвидше і випереджати будь-які конкуруючі ланцюжка[3]. Щоб змінити минулий блок, зловмиснику доведеться переробити доказ роботи блоку та всіх блоків після нього, а потім наздогнати та перевершити роботу чесних вузлів. Імовірність того, що повільніший зловмисник наздожене, зменшується експоненційно в міру додавання наступних блоків.

Вузли завжди вважають найдовший ланцюжок правильним і

продовжують працювати над його розширенням. Якщо два вузли одночасно транслюють різні версії наступного блоку, деякі вузли можуть одержати одну чи іншу першими. У цьому випадку вони працюють над першим отриманим ланцюжком, але зберігають іншу гілку на випадок, якщо вона стане довшою. Зв'язок буде розірвано, коли буде знайдено наступний доказ роботи і одна гілка стане довшою; вузли, які працювали над іншою гілкою, потім переключаться на довшу.

Для запуску мережі необхідно виконати такі кроки:

- Нові транзакції транслюються на всі вузли;
- Кожен вузол збирає нові транзакції у блок;
- Кожен вузол працює над пошуком складного доказу роботи свого блоку;
- Коли вузол знаходить підтвердження роботи, він транслює блок усім вузлам;
- Вузли приймають блок тільки в тому випадку, якщо всі транзакції в ньому дійсні та ще не витрачені. Вузли виражають своє прийняття блоку, працюючи над створенням наступного блоку в ланцюжку, використовуючи хеш прийнятого блоку як попередній хеш.

Нові трансляції транзакцій не обов'язково мають досягати всіх вузлів. Поки вони досягають багатьох вузлів, вони незабаром потраплять у блок. Трансляції блоків також терпимі до відкинутих повідомлень. Якщо вузол не отримує блоку, він запитає його, коли отримає наступний блок і зрозуміє, що пропустив один.

1.3.4 Конфіденційність

Традиційна банківська модель досягає рівня конфіденційності, обмежуючи доступ до інформації для сторін, що беруть участь, і довіреної третьої сторони. Необхідність публічно оголошувати про всі транзакції виключає цей метод, але конфіденційність все ще може бути збережена

шляхом переривання потоку інформації в іншому місці: збереження анонімності відкритих ключів. Громадськість може бачити, що хтось надсилає суму комусь іншому, але без інформації, що пов'язує транзакцію з будь-ким. Це схоже на рівень інформації, що публікується фондовими біржами, де час і розмір окремих угод «стрічка» публікуються, але без зазначення сторін. Зображено на рисунку 1.6.

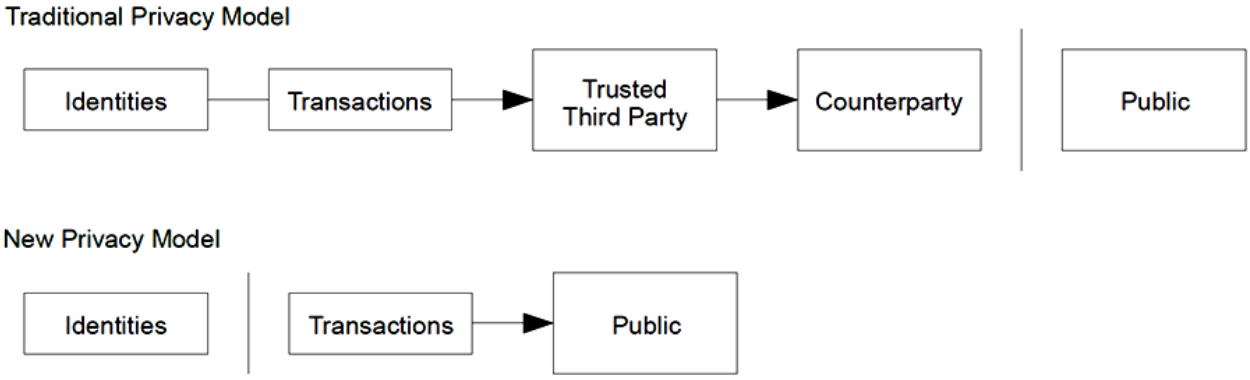


Рисунок 1.6 – Традиційна модель конфіденційності та нова модель конфіденційності

1.3.5 Структура даних блокчейну біткоїну

Для збереження у реєстрі Bitcoin кожна окрема транзакція має бути вбудована у структуру даних блоку Bitcoin. Блокчейн Bitcoin є послідовністю блоків, пов'язаних хеш-значеннями. Кожен блок складається із заголовка блоку та тіла блоку. Транзакції зберігаються в тілі блоку, а дайджест-інформація та інші ідентифікатори записуються в заголовок блоку. Блокчейн підтримується вузлами, що у мережі, і узгодженість даних між вузлами забезпечується відповідно до визначених правил Консенсусу. Огляд структури даних блокчейну Bitcoin представлений на рисунку 1.7.

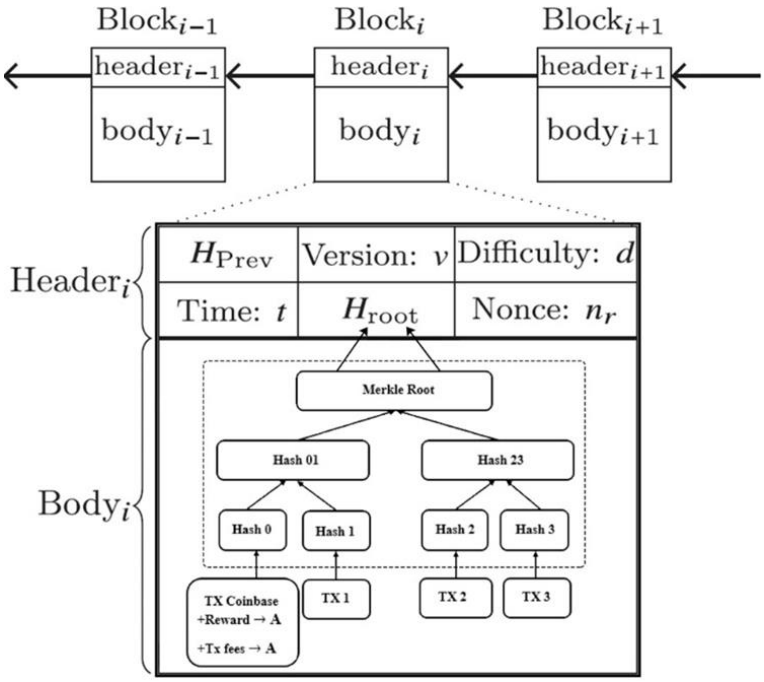
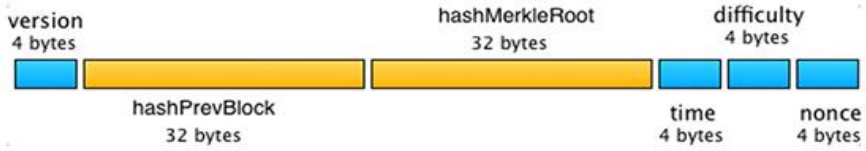


Рисунок 1.7 – структури даних блокчейна Біткойн.

Блок "розв'язаний" (опублікований і вважається дійсним для пирів), коли хеш заголовка блоку нижче поточної мети. Заголовок блоку складається з 640 біт (80 байт), як показано нижче. Більшість полів є константами, але майнери можуть грати з одним із них: nonce. Зображено на рисунку 1.8.



Field	Purpose	Updated When	Size (Bytes)
Version	A version number to track software/ protocol upgrades	When software is upgraded, a new version is specified	4
Previous Block Hash	256-bit hash of the previous block header	A new block comes in	32
Merkle Root	A hash of the root of the merkle tree of this block's transactions	A transaction is accepted	32
Timestamp	The approximate creation time of this block	Every few seconds	4
Bits(Target)	Current target in compact format	The difficulty is adjusted	4
Nonce	A counter used for the Proof-of-Work algorithm	A hash is tried (increments)	4

Рисунок 1.8 – Поля та структура заголовка блоку.

1.4 Майнінг біткоїнів

Біткоїн, на перший погляд, складається з трьох речей. По-перше, це протокол (або набір правил), який визначає, як має працювати мережа. По-друге, це програмний проект, який реалізує цей протокол. По-третє, це мережа комп'ютерів та пристроїв, що працюють під керуванням програмного забезпечення, яке використовує протокол для створення та управління валютою біткоїну. Для біткоїнів немає центрального органу, схожого на центральний банк, який контролює валюти.

Натомість програмісти вирішують складні головоломки, щоб схвалити транзакції біткоїну і отримати біткоїни як винагороду. Ця діяльність називається майнінгом біткоїну. Майнінг визначений у протоколі, реалізований у програмному забезпеченні та є важливою функцією в управлінні мережею біткоїну. Майнери перевіряють кожну транзакцію, вони створюють і зберігають усі блоки та досягають консенсусу про те, які блоки слід включити в ланцюжок блоків.

Щоб стати майнером Bitcoin, потрібно приєднатися до мережі Bitcoin і підключитися до інших вузлів. Після підключення вам необхідно виконати 5 завдань:

- Прослуховування транзакцій;
- Підтримуйте ланцюжок блоків та слухайте нові блоки;
- Зібрати блок-кандидат;
- Знайти nonce, який зробить ваш блок дійсним;
- Очікувати що ваш блок буде ухвалено.

1.4.1 Пошук допустимого блоку

Є ланцюжок блоків, де кожен заголовок блоку вказує на заголовок попереднього блоку в ланцюжку, а потім усередині кожного блоку є дерево Меркла всіх транзакцій, включених до цього блоку. Перше, що ви робите як

майнер - це компілюєте набір допустимих транзакцій, які у вас є з вашого пулу транзакцій, що очікують, в дерево Меркла. Звичайно, ви можете вибрати, скільки включити транзакцій, до межі загального розміру блоку. Потім ви створюєте блок із заголовком, який вказує на попередній блок. У заголовку блоку є 32-бітне поле nonce, і ви продовжуєте пробувати різні nonce, шукаючи той, який змушує хеш блоку бути нижчим за цільовий.

Для практичного простого прикладу, щоб почати з необхідної кількості нулів, майнер може почати з nonce 0 і послідовно збільшувати його на одиницю у пошуках nonce, що робить блок допустимим. Пошук допустимого блоку зображено на рисунку 1.9.

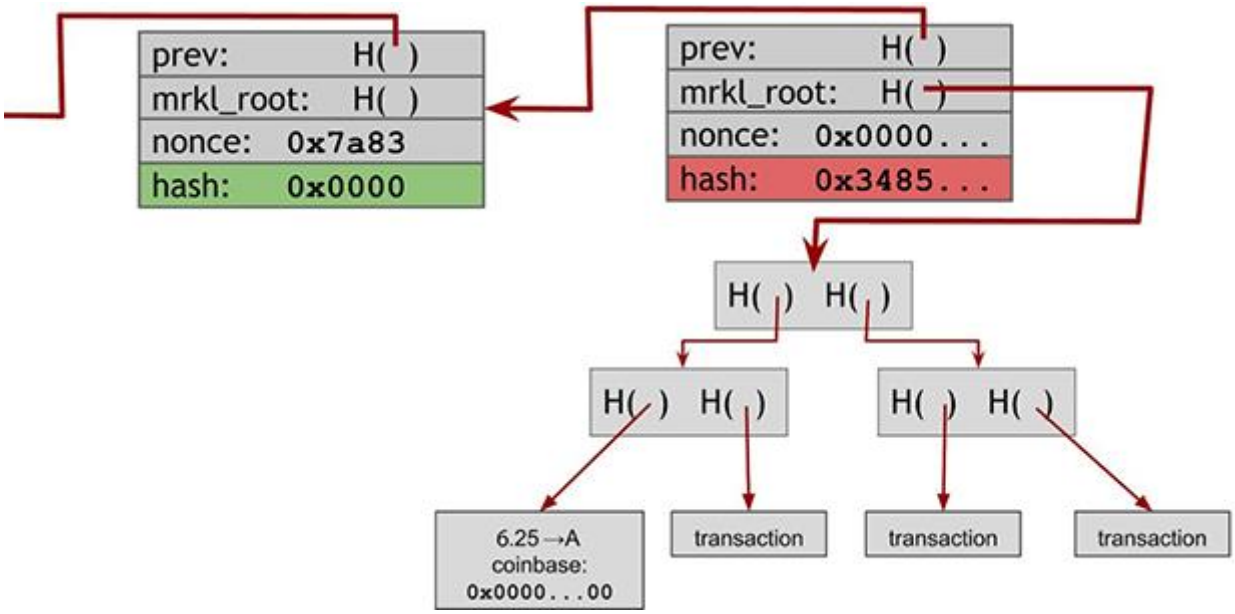


Рисунок 1.9 – Пошук допустимого блоку

Зазвичай доводиться перебирати всі можливі значення 32-бітного nonce і цього, як правило, недостатньо, щоб отримати потрібний хеш. Якщо всі значення в заголовку блоку вже використані, майнери змінюють "додатковий nonce" у транзакції Coinbase - це спеціальна транзакція, яка включає винагороду за майнінг.

Кожна зміна додаткового nonce у транзакції Coinbase змінює все дерево транзакцій (дерево Меркла), що робить процес складнішим та витратнішим

за обчисленнями. Тому майнери намагаються змінювати додатковий nonce тільки у крайньому випадку, коли всі значення для звичайного nonce у заголовку блоку вже вичерпані.

Оскільки більшість комбінацій не вийде потрібний хеш, майнери змушені довго перебирати значення. Але якщо вони будуть наполегливими, то, зрештою, знайдуть комбінацію nonce, яка підходить, і зможуть створити блок із потрібним хеш-значенням. Після цього вони повідомляють про знахідку, сподіваючись отримати винагороду.

1.4.2 Складність майнінгу Bitcoin

Складність майнінгу Bitcoin – це показник, який визначає, наскільки складно здобути новий блок. Щоб блок вважався дійсним, його хеш повинен бути нижчим за певне цільове значення. Чим нижче це значення, тим складніше знайти хеш, тому складність майнінгу тісно пов'язана з цією метою.

У мережі Bitcoin є два важливі значення:

- Мета (Біти), 256-бітове число, яке представляє максимальний хеш, при якому блок може вважатися допустимим. Чим вища мета, тим простіше добути блок (менша складність), і навпаки. Ця мета записується у спрощеній формі для зберігання та розрахунків. Зображено на рисунку 1.9;
- Складність, числове вираз того, наскільки складно знайти хеш блоку в порівнянні з найпростішою складністю, що дорівнює 1 (такою була складність при видобутку першого блоку, генезису блоку). Складність показує, наскільки складніше почало майнути блоки порівняно з початковим значенням. Зображено на рисунку 1.10.

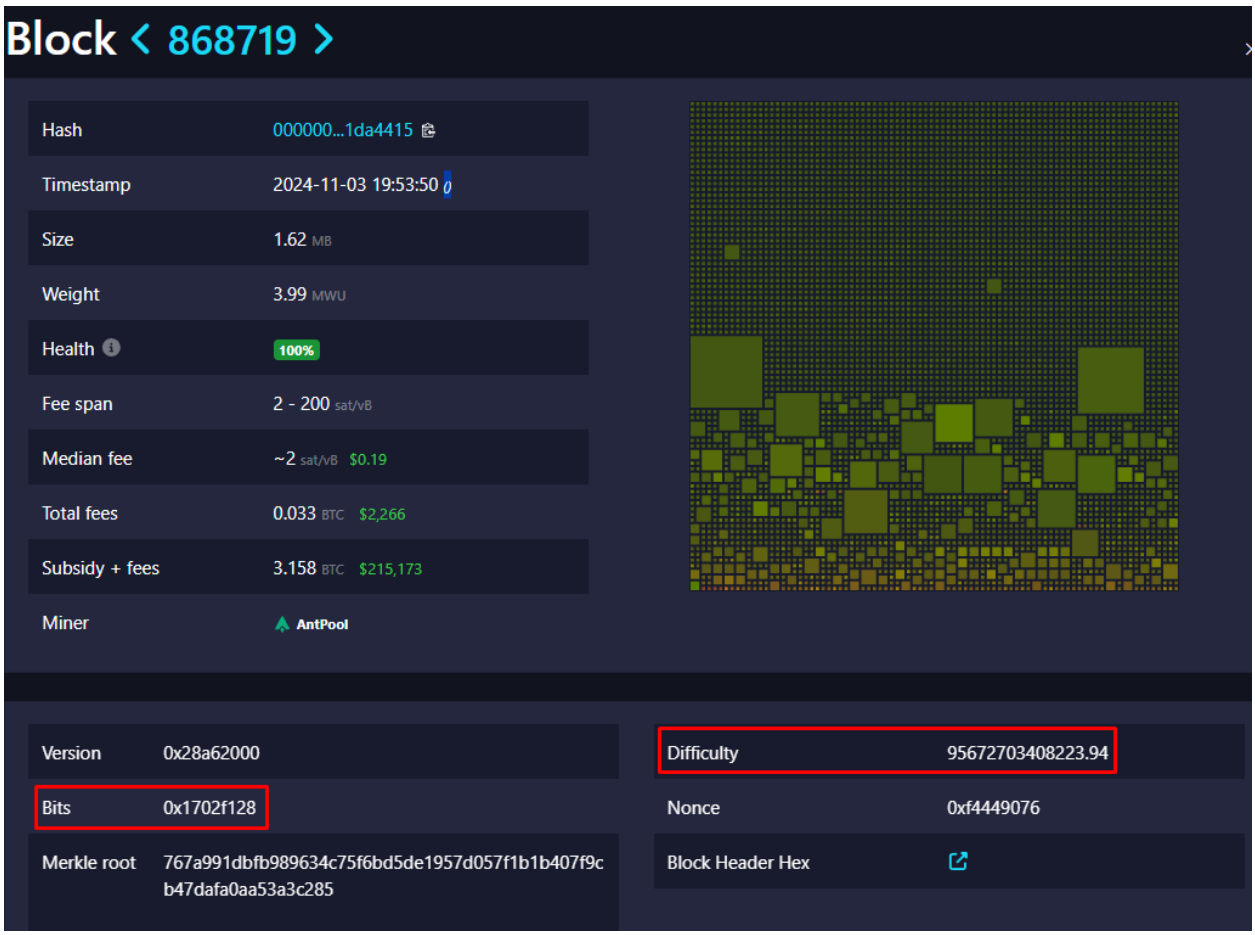


Рисунок 1.10 – Інформація про блок

Кожні 2016 блоків (приблизно кожні два тижні) складність коригується, щоб забезпечити стабільний час знаходження блоку – близько 10 хвилин.

Якщо за останні два тижні блоки знаходилися швидше, складність збільшиться і навпаки. Ця система підтримує баланс між обчислювальною потужністю мережі та стабільністю часу знаходження блоків.

Складність майнінгу зазвичай зростає, оскільки до мережі підключаються нові майнери та вдосконалюється обладнання. Чим більше майнерів беруть участь і чим потужніше їх пристрої, тим швидше знаходяться блоки, що викликає автоматичне збільшення складності. Складність, що зростає, відображає тенденцію до постійного збільшення хешрейту мережі, хоча це не завжди лінійний або експоненційний процес - складність може коливатися в залежності від ринкових умов та інтересу до

майнінгу.

Кожен майнер обчислює складність незалежно та приймає лише ті блоки, які відповідають його власним розрахункам. Якщо майнери знаходяться на різних гілках блокчейну, вони можуть мати різні значення складності. Але якщо два майнери працюють над продовженням одного і того ж блоку, вони використовуватимуть однакове значення складності, що забезпечує консенсус у мережі.

2 ІНСКРИПЦІЇ ТА NFT У БЛОКЧЕЙНІ БІТКОЇНА

2.1 Мультиагентні підходи у створенні цифрових активів Bitcoin

У сучасному цифровому середовищі спостерігається стрімке зростання обсягів даних, що генеруються у фінансових системах, блокчейн-мережах та цифровій економіці. Традиційні інструменти обробки інформації поступово поступаються місцем високорівневим автоматизованим підходам, які дозволяють забезпечити масштабованість, швидкість та достовірність результатів. У таких умовах мультиагентні системи, алгоритми штучного інтелекту та розподілені обчислення набувають особливого значення для побудови інфраструктури цифрових активів, зокрема - у мережі Bitcoin.

Одним із прикладів сучасного застосування таких підходів є використання протоколу Ordinals, що дає змогу створювати інскрипції - цифрові об’єкти, вбудовані безпосередньо в блокчейн Bitcoin. На відміну від смарт-контрактних NFT в Ethereum або Solana, інскрипції є повністю ончейн-структурами, де контент закріплюється за конкретним сатоші, без необхідності у зовнішніх базах даних чи токен-стандартах.

У свою чергу, автоматизація процесу формування інскрипцій вимагає надійної інфраструктури: повноцінної вузлової системи (ноди), RPC-інтерфейсів, синхронізації з мережею, роботи з CLI-інструментами (ord, bitcoin-cli) та постійного моніторингу стану блокчейну. Для оптимізації цих задач можуть бути задіяні мультиагентні системи, здатні координувати генерацію, обробку та перевірку інскрипцій у реальному часі. Застосування розподілених платформ типу Apache Spark, а також ML-моделей для класифікації типів інскрипцій (графічні, текстові, мультимедійні) може значно прискорити обробку, валідацію та візуалізацію цифрових артефактів у Bitcoin.

Таким чином, інтеграція інструментів штучного інтелекту та агентних

підходів в екосистему Bitcoin відкриває нові перспективи не лише для розвитку цифрової власності, але й для побудови самокерованих систем у фінансових, логістичних та наукових доменах. Мережа Bitcoin із її надійністю, незмінністю та глобальним охопленням може стати основою для нових інтелектуальних рішень у сфері децентралізованих активів та знань.[16]

2.2 Виникнення та розвиток інскрипцій у мережі Bitcoin

Блокчейн біткоїна, створений Сатоші Накамото у 2009 році, спочатку був задуманий як децентралізована платіжна система, яка дозволяє здійснювати грошові перекази без посередників. Архітектура мережі була навмисно обмеженою: у ній відсутні смарт-контракти, а можливості для взаємодії з даними за межами простих транзакцій були мінімальними. На цьому фоні Bitcoin здобув репутацію цифрового золота - активу, що слугує збереженню вартості, але не обов’язково виступає універсальним середовищем для децентралізованих додатків або цифрового контенту.

Значні зрушення почалися з прийняттям двох важливих оновлень протоколу. Segregated Witness (SegWit) у 2017 році та Taproot у 2021 році. Вони заклали фундамент для нового покоління інновацій на базі Bitcoin, зокрема - для створення інскрипцій і NFT[5], [8].

SegWit був запроваджений як рішення для масштабування мережі. Його головною метою було усунення проблеми гнучкості транзакцій (transaction malleability) та оптимізація структури блоку. Завдяки відокремленню підписів транзакцій в окрему частину блоку - так званий witness - з'явилася можливість ефективніше використовувати обмежений обсяг блоку (1 МБ), що опосередковано відкрив шлях до запису довільних даних у блокчейн. Хоча це не було основною метою оновлення, воно дало нові інструменти розробникам, які прагнули виходити за межі класичних фінансових транзакцій.

Подальший прорив відбувся з активуванням Taproot. Це оновлення об’єднало кілька важливих технологічних рішень - серед них особливо вирізняється використання MAST, що дозволяє записувати складні скрипти з високою приватністю. Крім того, Taproot надає можливість створення більш оптимізованих виходів транзакцій (script paths), зменшуючи їхній розмір та покращуючи ефективність. І саме завдяки цьому з’явилась можливість зберігати довільні дані (наприклад, зображення, аудіо, HTML-код) безпосередньо в блокчейні Bitcoin - чого раніше практично не робили через технічні обмеження і високу вартість.

На цьому тлі у 2023 році розробник Кейсі Родармор представив протокол Ordinals[5], який дозволяє прив’язати унікальні дані до окремих satoshi (1 bitcoin = 100 000 000 satoshi). Основна ідея полягає у нумерації кожного сатоші в історії транзакцій Bitcoin згідно з порядком їх створення (наприклад, в якому блоці він був видобутий). Завдяки цьому можна приписати метадані або цифровий контент конкретному сатоші, що перетворює його на об'єкт цифрової власності - аналогічний до NFT.

Важливо зазначити, що цей підхід докорінно відрізняється від реалізації NFT в мережі Ethereum, де кожен токен є окремим смарт-контрактом або частиною токен-стандарту ERC-721 чи ERC-1155[6]. У випадку з Ordinals, жодних нових токенів не створюється - інскрипції просто записуються у стандартні транзакції Bitcoin у спеціальних полях, які не порушують правил консенсусу.

Іншими словами, інскрипція є простою транзакцією з довільним контентом, яка зберігається у полі witness у Taproot-виході[5], [8]. Це дає змогу безпосередньо зберігати весь об’єкт, наприклад, PNG-зображення або текстовий файл в самому блокчейні, а не лише посилання на зовнішнє джерело, як це часто буває в Ethereum NFT. Завдяки цьому, Ordinals пропонують вищий ступінь незмінності й децентралізації, адже інскрипція не залежить від зовнішнього хостингу чи серверів.

Поява Ordinals викликала широкий резонанс у спільноті. З одного боку,

багато користувачів вітали нову функціональність, яка відкривала шлях до NFT, цифрового мистецтва та колекціонування в екосистемі Bitcoin. З іншого боку, критики вбачали в інскрипціях потенційну загрозу: через зростання обсягу блоків, використання місця для нефінансових даних, підвищення вартості комісій та відхилення від оригінального бачення Сатоші Накамото.

Втім, попри суперечки, інтерес до Ordinals продовжував зростати. За перші місяці після запуску було створено мільйони інскрипцій, з'явилися маркетплейси, гаманці та інфраструктура, що підтримує NFT у мережі Bitcoin. Це свідчить про великий попит на подібні рішення, а також про еволюцію ролі Bitcoin від простої платіжної системи до універсального протоколу для збереження цифрових активів.

Таким чином, поява протоколу Ordinals є результатом технічної еволюції Bitcoin, що стала можливою завдяки впровадженню SegWit і Taproot. Вона також є проявом загальної тенденції до розширення функціоналу блокчейнів і демократизації створення цифрової власності навіть у найбільш консервативних мережах.

2.3 Принцип роботи Ordinals

Протокол Ordinals, запропонований у 2023 році Кейсі Родармором, став технічною інновацією, що дозволяє зберігати довільні цифрові дані безпосередньо у блокчейні Bitcoin. Основна ідея полягає в нумерації окремих сатоші і можливості прикріплення до них унікального цифрового вмісту - інскрипцій. У результаті кожен такий сатоші перетворюється на об'єкт, аналогічний NFT, без необхідності створення нових токенів або використання смарт-контрактів.

2.3.1 Нумерація сатоші

У межах протоколу Ordinals передбачається детермінована нумерація

всіх сатоші в історії блокчейну Bitcoin відповідно до їх порядку створення. Для цього використовується концепція послідовної індексації, яка охоплює:

- Порядковий номер блоку, в якому сатоші був згенерований;
- Порядок сатоші всередині блоку;
- Історію передачі сатоші через транзакції.

Цей механізм дозволяє однозначно ідентифікувати кожен сатоші та створити своєрідну карту всіх існуючих одиниць біткоїна. Таким чином, з’являється можливість "приписати" довільні метадані або файли конкретному сатоші.

2.3.2 Інскрипції у структурі транзакцій

Сам процес інскрипції передбачає запис довільних даних у спеціальне поле Bitcoin-транзакції - witness. Це поле було запроваджене під час оновлення Segregated Witness (SegWit) і надалі доповнене оновленням Taproot, яке дозволяє ефективніше і приватніше реалізовувати складні скрипти. Важливо зазначити, що інскрипція не змінює сам сатоші - зміни відбуваються лише в транзакції, яка передає володіння ним.

Алгоритм створення інскрипції включає такі етапи:

- Користувач визначає конкретний сатоші, до якого буде прив’язано інскрипцію;
- Створюється Bitcoin-транзакція з Taproot-виходом, у полі witness якої вміщується цифровий вміст (зображення, текст, аудіо тощо);
- Транзакція включається до блоку та стає частиною незмінного реєстру;
- За допомогою індексації Ordinals можна визначити, який сатоші містить конкретну інскрипцію.

З технічної точки зору, інскрипція - це не що інше, як довільна послідовність байтів, записана у частину Taproot-скрипта. Важливо підкреслити, що ці дані є повністю ончейн, тобто зберігаються в самому

блокчейні, що відрізняє їх від більшості NFT в Ethereum, які зазвичай містять лише посилання на зовнішнє сховище.

2.3.3 Особливості та переваги

Підхід, запропонований у протоколі Ordinals, має низку принципових відмінностей від традиційної моделі NFT:

- Відсутня потреба у створенні нових токенів або застосуванні сторонніх стандартів;
- Кожен інскриптований сатоші має стабільний, математично виведений ідентифікатор;
- Усі дані записуються безпосередньо в блокчейн Bitcoin, що забезпечує максимальну децентралізацію і незмінність;
- Інскрипції не порушують жодних правил протоколу Bitcoin, що дозволяє їм бути повністю легітимними транзакціями.

Таким чином, Ordinals використовує вже існуючі можливості протоколу Bitcoin для реалізації концепції цифрової унікальності та власності, не вдаючись до принципів, властивих іншим мережам із розширеними можливостями, як-от Ethereum. Завдяки цьому інскрипції відкривають новий вектор розвитку для екосистеми Bitcoin, зберігаючи при цьому фундаментальні принципи децентралізації, простоти та незмінності.

2.4 NFT в інших блокчейнах

Незважаючи на те, що протокол Ordinals у мережі Bitcoin виконує схожі функції з невзаємозамінними токенами (NFT) в інших блокчейнах, його концепція, технічна реалізація та ідеологічна основа суттєво відрізняються. Для повноцінного розуміння ролі інскрипцій у Bitcoin доцільно провести порівняльний аналіз із NFT в екосистемах Ethereum та Solana, які є найбільш розвиненими з погляду ринку NFT.

2.4.1 NFT в Ethereum

Ethereum був першим блокчейном, де з’явилися повноцінні NFT. Завдяки підтримці Turing-повноцінних смарт-контрактів, платформа дозволяє створювати токени з унікальними властивостями, використовуючи стандарти ERC, зокрема:

- ERC-721 для створення невзаємозамінних токенів;
- ERC-1155 мультистандарт для комбінованих активів.

Ці стандарти дозволяють:

- Визначати права власності;
- Додавати метадані;
- Використовувати динамічну логіку (роялті, оренда тощо);
- Створювати інтерактивні цифрові об’єкти.

Однак, більшість NFT в Ethereum не зберігають повний контент ончейн. Зазвичай лише хеш або посилання (URI) на зовнішнє сховище, наприклад, IPFS або centralized CDN.

2.4.2 NFT в Solana

Solana також підтримує NFT, хоча з дещо іншою архітектурою. Завдяки високій пропускній здатності та низькій вартості транзакцій, ця платформа стала популярною серед масового NFT-маркету. NFT у Solana використовують Metaplex Token Metadata Standard[7], який дозволяє керувати токенами через окремі об’єкти-метадані, а не тільки самі токени.

Основні переваги:

- Швидке створення і передача NFT;
- Низька вартість mint-операцій;
- Зручна інтеграція з централізованими сервісами.

Проте, подібно до Ethereum, більшість NFT у Solana також залежать від зовнішніх джерел для зберігання файлів.

2.4.3 Порівняння з NFT в інших блокчейнах

Попри спільну мету, надання унікальності цифровим об’єктам, підходи до реалізації NFT у Bitcoin (Ordinals), Ethereum та Solana суттєво відрізняються:

- Bitcoin (Ordinals) найрадикальніше рішення з ідеологічного погляду. Воно базується на максимальній децентралізації, відсутності смарт-контрактів і повному зберіганні даних ончейн. Завдяки цьому інскрипції є прозорими, незмінними й не потребують сторонніх сервісів для існування. Проте обмежена масштабованість і слабка інфраструктура поки стримують широке впровадження;
- Ethereum найбільш функціонально багата платформа для NFT. Завдяки смарт-контрактам, стандартам ERC і великій екосистемі Ethereum забезпечує широкі можливості кастомізації, логіки роялті, гейміфікації та взаємодії з іншими dApp. Основний недолік - залежність від зовнішніх сховищ і висока вартість транзакцій у моменти навантаження;
- Solana прагне до балансу між доступністю і функціональністю. Вона забезпечує швидке та дешеве створення NFT з базовою логікою, що робить її зручною для масового ринку. Проте технічна складність, залежність від централізованих сервісів та ризики відключень мережі залишаються суттєвими факторами.

Загалом, Ordinals - це більш фундаменталістський підхід до цифрової унікальності, тоді як Ethereum і Solana пропонують більш гнучкі, але технічно складні й частково централізовані рішення.

2.5 Переваги та обмеження NFT у Bitcoin

Використання протоколу Ordinals для реалізації невзаємозамінних токенів (NFT) у мережі Bitcoin є важливою інновацією, що позначає принципово новий підхід до цифрової унікальності, зберігання та власності в

межах першого і найнадійнішого блокчейну. Проте така реалізація супроводжується як очевидними перевагами, так і суттєвими обмеженнями, обумовленими архітектурою мережі, її функціональними характеристиками та філософією дизайну.

2.5.1 Надійність і незмінність даних

Однією з головних переваг NFT у Bitcoin є повна ончейн-природа інскрипцій, яка радикально відрізняється від підходів, домінуючих в екосистемах Ethereum і Solana. Всі інскрипції зберігаються безпосередньо в полі witness транзакцій - частині блокчейну, що була введена з оновленням Segregated Witness (SegWit), а її гнучкість посилена активацією Taproot. Це забезпечує:

- Незалежність від зовнішніх джерел зберігання, на відміну від URI-посилань чи IPFS, які є вразливими до зникнення чи модифікації;
- Криптографічну незмінність, оскільки інскрипція є інтегральною частиною ланцюга блоків і не може бути видалена або змінена без радикального порушення консенсусу;
- Архівну стійкість, що забезпечує довготривале збереження цифрових артефактів незалежно від комерційних або технологічних змін у майбутньому.

Таким чином, NFT у Bitcoin мають унікальний потенціал для використання в контекстах, де незмінність і довговічність цифрових об’єктів є критично важливими - наприклад, в архівуванні культурної спадщини, наукових даних або прав власності.

2.5.2 Вартість і масштабованість

Попри описані переваги, протокол Ordinals суттєво обмежений в аспектах економічної ефективності та масштабованості:

- Розмір блоку Bitcoin обмежений, 1-4 МБ у перерахунку на weight units, що створює дефіцит простору, особливо в умовах зростання попиту на запис великих інскрипцій[10];

- Вартість транзакцій при записі великих файлів у форматі inscribe значно вища, ніж звичайні платежі часто у десятки або сотні доларів США;

- Відсутність підтримки Layer 2-рішень для інскрипцій, аналогічних Ethereum L2 або Solana-компресії, робить масове застосування NFT у Bitcoin технічно неефективним.

Це створює структурну конкуренцію між NFT і фінансовими транзакціями за обмежений ресурс блоку, що суперечить первинній платіжній меті мережі.

2.5.3 Критика та технічні виклики

Інскрипції також стали об’єктом контроверсійної дискусії в межах Bitcoin-спільноти та серед технічних дослідників:

- Критики вважають, що великі inscribe-транзакції створюють навантаження на повні вузли та порушують оптимізацію зберігання[10];

- Ordinals не підтримують смарт-контракти або динамічну логіку (роялті, оренда, автоматичне оновлення даних), що обмежує варіативність і функціонал NFT;

- Існує ймовірність того, що деякі майнери або провайдери повних нодів у майбутньому будуть фільтрувати інскрипції, наприклад, через правові або етичні міркування;

- Відсутність нативної підтримки в основному клієнті Bitcoin Core, що робить роботу з інскрипціями потенційно вразливою до змін у клієнтах або мережевій політиці.

3 ПІДГОТОВКА ДО РЕАЛІЗАЦІЇ ИНСКРИПЦІЙ В МЕРЕЖІ BITCOIN

3.1 Вибір середовища розробки та підготовка конфігурації

Для успішної реалізації механізму інскрипцій у мережі Bitcoin необхідно спершу створити відповідне тестове середовище, яке дозволить безпечно й ефективно випробувати всі функціональні елементи проєкту до їх запуску в основній мережі. На даному етапі критично важливо обрати правильну мережу для тестування, налаштувати основне програмне забезпечення та забезпечити всі необхідні інструменти для подальшої роботи з інскрипціями.

Зважаючи на специфіку проєкту, пов’язану із взаємодією з блокчейном, обробкою транзакцій та використанням протоколу Ordinals, ключовим компонентом є локальна нода Bitcoin. Саме вона дозволяє повністю контролювати процес, здійснювати операції без посередників і працювати з реальними механізмами протоколу Bitcoin.

У зв’язку з цим, перш за все, обґрунтовується вибір програмного забезпечення Bitcoin Core - офіційного клієнта мережі Bitcoin з відкритим кодом, який забезпечує повноцінну синхронізацію з мережею, генерацію адрес, обробку транзакцій та взаємодію з іншими інструментами через RPC.

Крім вибору клієнта, критичне значення має і вибір мережі, в якій буде здійснюватися тестування. Хоча мережею за замовчуванням вважається testnet, для даного проєкту доцільно використовувати signet - альтернативну тестову мережу, що має переваги у вигляді швидшої синхронізації, більшої стабільності та низьких вимог до системних ресурсів. Ці дії є підготовчим етапом, який забезпечує базу для подальших кроків, пов’язаних зі створенням, передачею та обробкою інскрипцій у рамках експериментальної реалізації.

3.2 Bitcoin Core

Bitcoin Core є офіційною та найбільш авторитетною реалізацією клієнта Bitcoin - одночасно програмною інфраструктурою і ядром вузлової мережі, яке виконує повноцінну валідацію блоків, зберігання повної історії транзакцій і надає API для взаємодії з блокчейном. Його роль у процесі створення інскрипцій, зокрема згідно з протоколом Ordinals, є фундаментальною, оскільки саме на ньому базується повна технічна інфраструктура, що забезпечує роботу з низькорівневими транзакційними структурами.

3.2.1 Функціональна сутність та архітектура

Bitcoin Core побудований за принципами високої модульності й забезпечує набір ключових можливостей: зберігання і перевірку всіх блоків з моменту створення мережі, верифікацію транзакцій відповідно до консенсусу, розподілене зберігання копії блокчейну, а також підтримку P2P-протоколу для обміну даними між нодами. Його архітектура базується на C++ та бібліотеці Boost, а основна логіка поділена на мережевий стек, обробку блоків, валідацію транзакцій і JSON-RPC інтерфейс.

Історично Bitcoin Core був створений як продовження оригінального клієнта Сатоші Накамото та наразі підтримується широкою міжнародною спільнотою розробників. Він служить своєрідним золотим стандартом для вузлів мережі Bitcoin, і його оновлення формують еволюцію протоколу Bitcoin загалом.

3.2.2 Повноцінна нода та важливість індексації

У контексті створення інскрипцій критично важливим є використання Bitcoin Core саме в режимі повної ноди (full node). Така нода забезпечує

локальну перевірку всіх блоків та транзакцій без покладання на сторонні сервери, що дозволяє досягти максимальної незалежності, достовірності даних і безпеки. Це особливо актуально в умовах роботи з Taproot-транзакціями, які використовуються протоколом Ordinals для збереження inscription-даних у сегменті witness.

Однією з основних вимог до Bitcoin Core для використання з Ordinals є активація індекса транзакцій (параметр txindex=1 у конфігурації). За замовчуванням Bitcoin Core не зберігає повний індекс транзакцій для економії місця на диску, однак саме ця опція уможливлює доступ до історичних даних про будь-яку транзакцію - без неї обробка inscription втрачає функціональність.

3.2.3 Інтерфейс JSON-RPC і можливості інтеграції

Bitcoin Core включає вбудований RPC-сервер, який реалізує повноцінний API для взаємодії з блокчейном. Через цей інтерфейс можна виконувати сотні команд - від перевірки балансу адреси до створення необроблених транзакцій (raw transactions), що є важливим для програмної генерації inscription. RPC-інтерфейс також використовується в інтеграціях із клієнтами типу ord та іншими системами, що автоматизують створення, управління й індексацію інскрипцій.[13]

Для налаштування RPC доступу необхідно задати rpcuser, rpcpassword, порт (8332 для mainnet, 18332 для testnet, 38332 для signet) та дозволені IP (rpcallowip=127.0.0.1), після чого програмне забезпечення на локальній машині може звертатися до API Bitcoin Core для отримання актуальної інформації з блокчейну.

3.2.4 Підтримка тестових середовищ і роль у розробці

Bitcoin Core підтримує кілька ізольованих середовищ для розробників:

Testnet, Signet і Regtest. Кожна з них має своє призначення:

- Testnet публічна тестова мережа з нестабільною генерацією блоків;
- Signet стабільна мережа з блоками, що створюються централізовано під контролем Bitcoin Foundation, ідеальна для розробки;
- Regtest повністю локальна мережа з ручною генерацією блоків.

У межах цього проєкту обрано Signet, оскільки вона дозволяє ефективно симулювати реальні умови Bitcoin-мережі при мінімальних вимогах до ресурсів. Розмір блокчейну Signet становить лише 1-5 ГБ, а синхронізація триває від 15 до 30 хвилин, що робить її оптимальною платформою для інтеграції з протоколом Ordinals на ранніх етапах.

3.3 Тестова мережа Signet

У процесі розробки, тестування та перевірки функціональності систем, що базуються на протоколі Bitcoin, надзвичайно важливим є використання ізольованих середовищ. Одним із найбільш зручних і надійних таких середовищ є Signet - офіційна тестова мережа Bitcoin, яка була представлена у версії Bitcoin Core 0.21.0 як альтернатива Testnet з поліпшеними характеристиками стабільності та передбачуваності. У рамках цього дослідження саме Signet використовується для практичної реалізації інскрипцій з огляду на її технічні переваги.[11]

3.3.1 Призначення та переваги Signet

Signet є централізовано керованою тестовою мережею, блоки в якій створюються за допомогою контрольованого механізму підписів. Це відрізняє її від традиційної Testnet, де блоки генеруються в децентралізований спосіб, що часто призводить до нестабільності в частоті створення блоків та непередбачуваності мережевих умов.

Signet має низку суттєвих переваг, які роблять його зручним

інструментом для тестування та розробки в середовищі Bitcoin. Однією з ключових переваг є швидка та стабільна синхронізація, що забезпечує ефективну роботу з мережею без затримок. Усі блоки в Signet мають спеціальні підписи, завдяки чому виключаються спам-атаки та нестабільна генерація блоків, яка характерна для традиційного Testnet. Інтерфейси, транзакції та консенсусні правила в Signet повністю ідентичні основній мережі Bitcoin, що дозволяє безпечно тестувати функціонал у максимально наближених умовах. Також Signet підтримується faucet-ресурсами, які спрощують отримання монет для тестування, і має вбудовану підтримку протоколу Ordinals, що є важливою функцією для роботи з цифровими активами у біткоїн-мережі.

3.3.2 Сценарії застосування у контексті інскрипцій

Signet забезпечує контрольовану та безпечну платформу для створення і тестування інскрипцій без ризику втрати коштів або порушення консенсусу в основній мережі. У рамках реалізації проєкту було обрано саме цю мережу, оскільки вона дозволяє відтворити повноцінну логіку inscription-процесу - від генерації транзакцій до перевірки їх коректного запису в блокчейн - без надмірних обчислювальних і часових витрат. Крім того, Signet підтримується основними інструментами з екосистеми Ordinals, такими як ord, unisat, sparrow, тощо, що забезпечує повну сумісність зі стандартними сценаріями створення inscription.

4 РЕАЛІЗАЦІЯ ПРОЦЕСУ СТВОРЕННЯ NFT ЗА ДОПОМОГОЮ ІНСКРИПЦІЙ

4.1 Налаштування середовища виконання

Першим етапом реалізації процесу створення NFT за допомогою інскрипцій є розгортання повноцінного вузла Bitcoin Core, налаштованого для роботи в тестовій мережі Signet. Для цього було обрано середовище WSL (Windows Subsystem for Linux) з інстальованим дистрибутивом Ubuntu 22.04 LTS, що дозволяє зручно працювати з нативними Linux-утилітами безпосередньо в середовищі Windows.

Для завантаження використовувалась офіційна збірка Bitcoin Core Після розпакування архіву до директорії /bitcoin-28.0/, виконувані файли bitcoind та bitcoin-cli стали доступними для локального запуску.

Далі було створено окрему конфігураційну директорію для роботи в мережі Signet за шляхом /root/.bitcoin/signet/bitcoin.conf. Вміст цього конфігураційного файлу зображено на рисунку 4.1.

*bitcoin.conf

```
# bitcoin.conf для Signet – швидка тестова мережа для розробників, зручна для роботи з інскрипціями

# Вмикає роботу в мережі Signet (альтернатива Testnet, але з більш стабільними блоками)
signet=1

# Виділення 4000 МБ оперативної пам'яті для кешу бази даних – прискорює обробку блоків
dbcache=4000

# Обмеження кількості одночасних з'єднань з іншими вузлами (50 – оптимально для швидкої синхронізації)
maxconnections=50

# Увімкнення індексу транзакцій – необхідно для роботи з Ordinals і Inscription
txindex=1

# RPC-конфігурація для взаємодії з Bitcoin Core через зовнішні інструменти (наприклад, ord)
[signet]

# Дозволяє використовувати RPC-сервер
server=1

# Ім'я користувача для RPC-доступу (використовується в клієнтах, як-от ord)
rpcuser=bitcoinuser

# Пароль для RPC-доступу (забезпечує базову авторизацію)
rpcpassword=bitcoinpass123

# Дозволити доступ до RPC лише з локального комп'ютера
rpcallowip=127.0.0.1

# Порт для RPC у мережі Signet (за замовчуванням 38332)
rpcport=38332

```

Рисунок 4.1 – Конфігураційний файл bitcoin.conf для роботи у мережі Signet

Після збереження файлу Bitcoin Core запускається з цими параметрами, і система починає процес синхронізації з Signet-мережею. Залежно від швидкості інтернету та обсягу кешу, процес повної синхронізації може займати від 15 до 30 хвилин.

Після запуску вузла Bitcoin Core у режимі мережі Signet необхідно перевірити поточний стан синхронізації, наявність індексу транзакцій та готовність ноди до взаємодії з клієнтами, зокрема ord. Це здійснюється за допомогою команди getblockchaininfo, яка повертає детальну інформацію про поточний стан блокчейну, рівень складності, кількість завантажених блоків, обсяг даних на диску тощо. Зображено на рисунку 4.2.

```
root@DESKTOP-BR8GLU4:~# bitcoin-cli -signet getblockchaininfo
{
  "chain": "signet",
  "blocks": 255710,
  "headers": 256302,
  "bestblockhash": "0000000c4e73839619e22194c15b67e01a173eddbfd197bf55b71cd96da1e093",
  "difficulty": 0.04756651218830634,
  "time": 1749539815,
  "mediantime": 1749539690,
  "verificationprogress": 0.9999457106132729,
  "initialblockdownload": true,
  "chainwork": "00000000000000000000000000000000000000000000000043fccbd9a6d",
  "size_on_disk": 11298309901,
  "pruned": false,
  "warnings": ""
}
root@DESKTOP-BR8GLU4:~# bitcoin-cli -signet getblockchaininfo
{
  "chain": "signet",
  "blocks": 256302,
  "headers": 256302,
  "bestblockhash": "0000000455933f6fb489cd5489ca41f8dccf5dd5b2dbb46eb8c296aaf2bf7d62",
  "difficulty": 0.05017417601347472,
  "time": 1749854771,
  "mediantime": 1749852953,
  "verificationprogress": 0.9999997386636378,
  "initialblockdownload": false,
  "chainwork": "00000000000000000000000000000000000000000000000045caa900e7b",
  "size_on_disk": 11629431145,
  "pruned": false,
  "warnings": ""
}
```

Рисунок 4.2 – Стан ноди після синхронізації у мережі Signet

У результаті виводиться детальна інформація про поточний стан ланцюга блоків, включаючи:

- Активний ланцюг;
- Кількість синхронізованих блоків;

- Рівень завершеності верифікації, що вказує на повну синхронізацію;
- Статус, що підтверджує завершення початкового завантаження;
- Обсяг даних на диску, що відповідає стандартному розміру повної Signet-ноди;
- Параметр pruned вказує, що індексація повна, без обрізки блоків, що є обов'язковим для роботи з Ordinals.

Цей результат підтверджує, що вузол повністю синхронізований та готовий до створення інскрипцій у тестовій мережі. Важливо також зазначити, що середовище Signet дозволяє швидко тестувати механізми створення NFT без необхідності чекати підтвердження мережі або витрачати реальні біткоїни, що робить його ідеальним вибором на етапі розробки та верифікації логіки NFT-механізму у межах протоколу Ordinals.

4.2 Встановлення та компіляція утиліти ord

Після завершення налаштування та успішної синхронізації локального вузла Bitcoin Core у середовищі тестової мережі Signet наступним логічним етапом є інтеграція інструменту, який безпосередньо реалізує протокол Ordinals. У межах цієї роботи було обрано CLI-утиліту з відкритим кодом під назвою ord, що забезпечує створення, управління та інспекцію інскрипцій - метаданих або файлів, закріплених за окремими сатоші в блокчейні Bitcoin.[15]

Першим кроком є отримання вихідного коду утиліти ord з офіційного репозиторію GitHub. Після клонування коду необхідно переконатися, що в системі встановлено актуальне середовище розробки для мови програмування Rust, оскільки проєкт ord базується саме на цій мові. Якщо Rust не встановлено, його можна інсталювати через офіційний інсталятор. Rust є сучасною системною мовою з високою продуктивністю та безпекою, що робить її оптимальним вибором для розробки криптовалютного програмного забезпечення. У результаті виконання компіляції в каталозі

./target/release/ буде згенеровано виконуваний файл ord, готовий до запуску.

4.2.1 Запуск локального сервера ord

Функціонування утиліти ord передбачає наявність локального HTTP-сервера, який здійснює індексацію блокчейну, взаємодіє з RPC-інтерфейсом Bitcoin Core та обслуговує запити користувача. Таким чином, для повноцінної роботи рекомендується запускати утиліту ord у двох окремих термінальних вікнах або сесіях. [11]

У першому вікні виконується запуск індексуючого сервера з прив'язкою до RPC-інтерфейсу локального вузла Bitcoin Core. Зображено на рисунку 4.3.

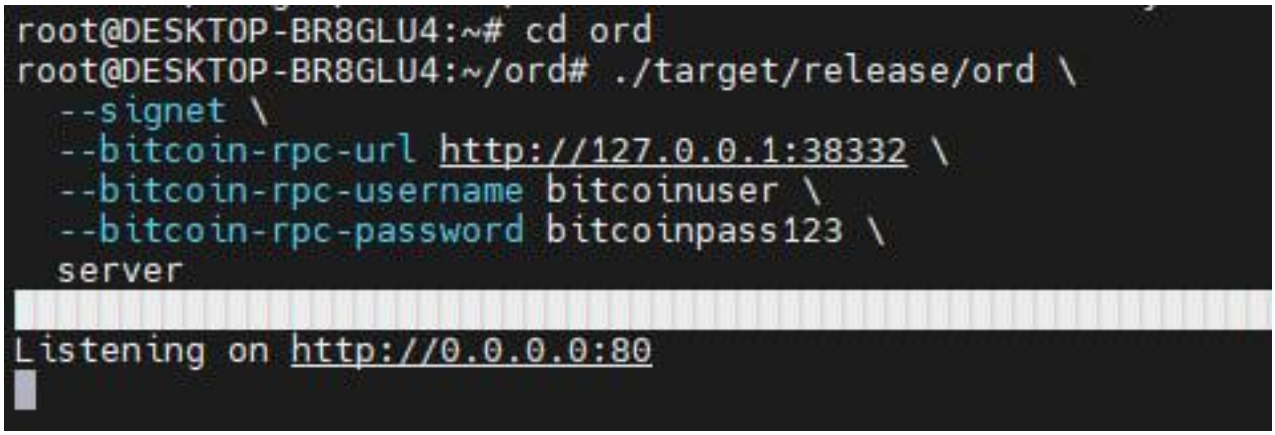


Рисунок 4.3 – Процес запуску локального HTTP-сервера

Цей процес є блокуючим: сервер продовжує роботу у фоновому режимі, обслуговуючи вхідні запити, доки вручну не буде завершено процес. Під час першого запуску виконується синхронізація з локальним вузлом, що включає індексацію транзакцій, блоків та адрес за стандартом Taproot.

У другому вікні відбувається клієнтська взаємодія з сервером через HTTP-протокол. Зокрема, всі запити здійснюються через порт localhost:80, який прослуховується утилітою ord за замовчуванням. Наприклад, для створення нового гаманця. У відповідь утиліта виводить mnemonic-фразу, що

є сидом HD-гаманця і має бути збережена користувачем у надійному місці. Вона використовується для відновлення доступу до гаманця в майбутньому та є критично важливою для безпеки. Зображено на рисунку 4.4.

```
root@DESKTOP-BR8GLU4:~# cd ord
root@DESKTOP-BR8GLU4:~/ord# ./target/release/ord --signet \
  --bitcoin-rpc-url http://127.0.0.1:38332 \
  --bitcoin-rpc-username bitcoinuser \
  --bitcoin-rpc-password bitcoinpass123 \
  wallet create
{
  "mnemonic": "blue fine remember caution into crush clerk stove giraffe tent diary wife",
  "passphrase": ""
}
root@DESKTOP-BR8GLU4:~/ord#
```

Рисунок 4.4 – Генерація mnemonic-фрази

Наступним кроком є отримання першої Taproot-адреси для прийому коштів, необхідних для подальшого створення інскрипції. Це здійснюється через окреме вікно термінала, де виконується запит до локального сервера за допомогою команди, яка повертає список згенерованих адрес, придатних для надсилання тестових монет у мережі Signet. Зображено на рисунку 4.5.

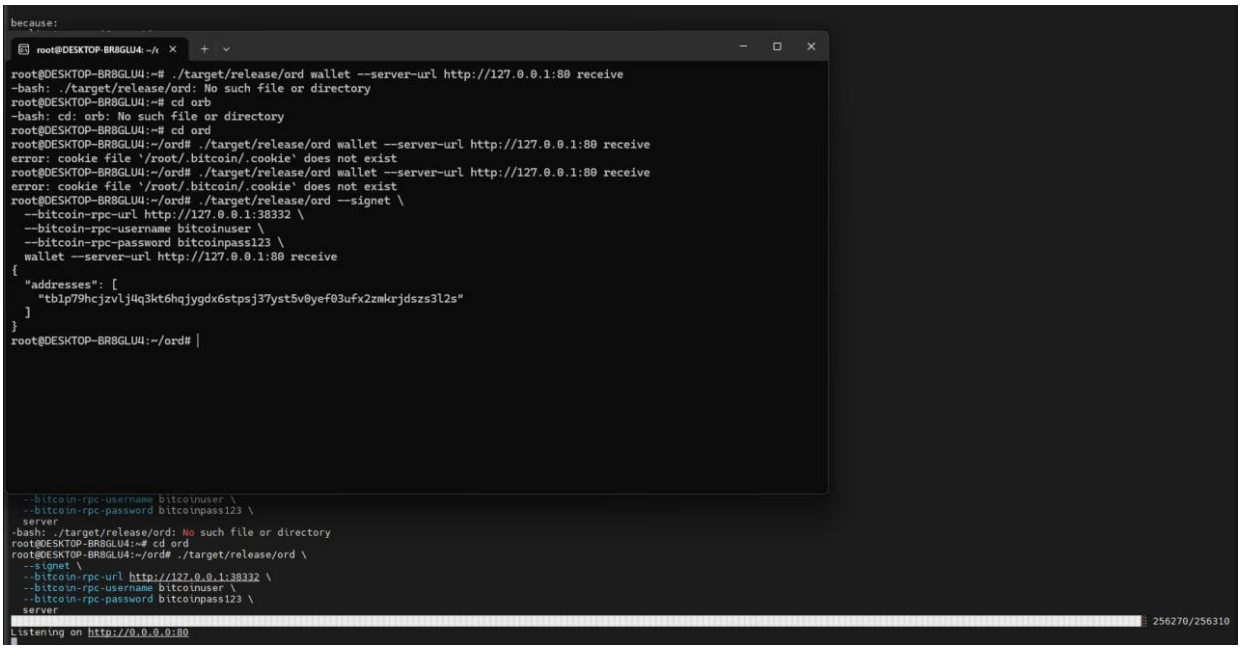


Рисунок 4.5 – Сторення Taproot-адреси

4.2.2 Отримання тестових токенів

Коли Taproot-адреса була згенерована, наступним кроком є отримання тестових біткоїнів у мережі Signet для покриття комісійних витрат під час створення інскрипцій. Оскільки Signet є тестовою мережею, монети в ній не мають реальної вартості й можуть бути отримані через спеціальні фаусети.

На сторінці фаусету необхідно вставити згенеровану Taproot-адресу у відповідне поле й надіслати запит. Через декілька секунд кошти з'являться як непідтверджена транзакція в блокчейні, що можна перевірити за допомогою блок-експлорера. Зображено на рисунку 4.6.

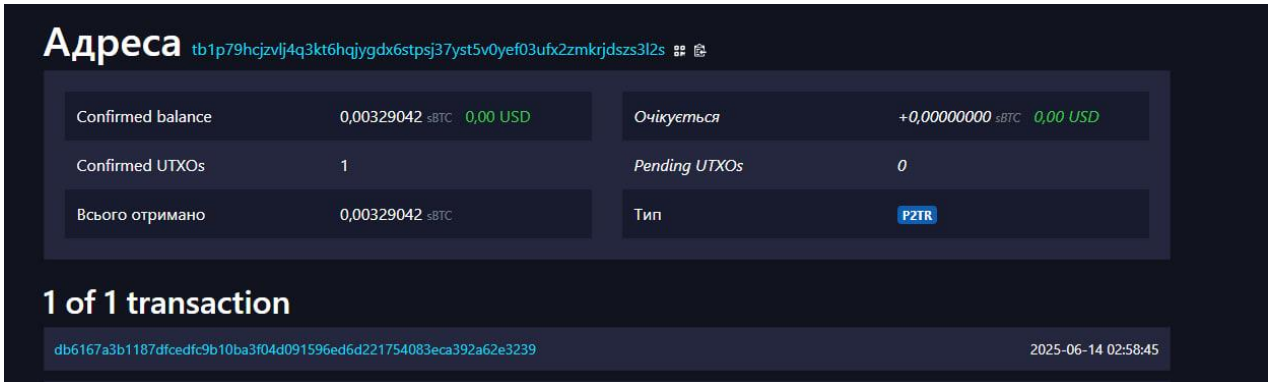


Рисунок 4.6 – Інформація про отриманні токени з mempool

Наявність хоча б одного UTXO (непотрачений вихід транзакції) на гаманці є обов'язковою умовою для продовження інскрипції. Тому цей крок завершує підготовку середовища для безпосереднього створення NFT через протокол Ordinals. Для перевірки балансу гаманця або наявних UTXO застосовуються відповідно команди, які зображені на рисунку 4.7.

```
root@DESKTOP-BR8GLU4:~/ord# ./target/release/ord --signet \
  --bitcoin-rpc-url http://127.0.0.1:38332 \
  --bitcoin-rpc-username bitcoinuser \
  --bitcoin-rpc-password bitcoinpass123 \
  wallet cardinals
[
  {
    "output": "db6167a3b1187dfcedfc9b10ba3f04d091596ed6d221754083eca392a62e3239:1526",
    "amount": 329042
  }
]
```

Рисунок 4.7 – Вивід команди про стан UTXO

4.3 Створення інскрипції

4.3.1 Стиснення файлу для інскрипції

На етапі створення інскрипції доцільно завчасно підготувати вхідний медіафайл - зображення, яке буде закодоване безпосередньо у транзакції мережі Bitcoin. Важливо враховувати, що мережа Signet, як і інші мережі Bitcoin, накладає обмеження на максимальний розмір даних, які можуть бути інтегровані в один вихід транзакції. Для тестової мережі Signet це обмеження становить 1024 байти.

У зв’язку з цим, великі файли наприклад, зображення у форматі PNG або JPG мають бути попередньо стиснуті до допустимого розміру. У ході реалізації був обраний формат WebP, який забезпечує ефективне стиснення зображень. Для цього використовувались засоби графічного конвертування, зокрема утиліти cwebp (CLI). Зображено на рисунку 4.8.

```
root@DESKTOP-BR8GLU4:~/ord# cwebp cats.png -resize 50 50 -q 20 -o cats.webp
Saving file 'cats.webp'
File:      cats.png
Dimension: 50 x 50 (with alpha)
Output:    884 bytes Y-U-V-All-PSNR 31.14 36.32 36.71   32.31 dB
           (2.83 bpp)
block count:  intra4:          9  (56.25%)
              intra16:         7  (43.75%)
              skipped:         2  (12.50%)
bytes used:  header:            33  (3.7%)
             mode-partition:    47  (5.3%)
             transparency:      416 (99.0 dB)
 Residuals bytes  |segment 1|segment 2|segment 3|segment 4|  total
    macroblocks:  |      19%|      19%|      31%|      31%|      16
      quantizer:  |      78 |      66 |      56 |      48 |
  filter level:   |      30 |      15 |       9 |       7 |
Lossless-alpha compressed size: 415 bytes
  * Header size: 39 bytes, image data size: 376
  * Precision Bits: histogram=5 transform=5 cache=0
  * Palette size:   151
```

Рисунок 4.8 – Результат виконання команди стиснення зображення

Після цього зображення було перекодовано в оптимізований формат WebP та повторно перевірене щодо відповідності обмеженням. Правильно підготовлений файл дозволив успішно створити інскрипцію, про що

йтиметься у наступному підрозділі.

Цей етап є критично важливим, оскільки недотримання ліміту призводить до неможливості формування транзакції інскрипції, що зупиняє весь процес запису NFT у блокчейн. Таким чином, попередня обробка і стиснення є обов'язковою частиною інструментарію розробника у контексті роботи з протоколом Ordinals.

4.3.2 Проведення інскрипції

Після успішного запуску локального сервера ord та отримання Taproot-адреси з активами в мережі Signet, можна безпосередньо перейти до ключового етапу - створення інскрипції (inscription), що є аналогом NFT у протоколі Ordinals. [12], [15]

Інскрипція передбачає прикріплення певного контенту, стисненого зображення у форматі WebP, до конкретного сатоші у вихідному UTXO. Це досягається за допомогою wallet inscribe, яка створює та підписує транзакцію, що додає inscription до вихідного набору даних блокчейну. На рисунку 4.9 зображено результат успішного проведення інскрипції.

```
root@DESKTOP-BR8GLU4:~/ord# ./target/release/ord --signet \
  --bitcoin-rpc-url http://127.0.0.1:38332 \
  --bitcoin-rpc-username bitcoinuser \
  --bitcoin-rpc-password bitcoinpass123 \
  wallet inscribe \
  --fee-rate 10 \
  --file cats.webp
{
  "commit": "70f7dd00036731e59ee8eb7aaedbdda0451dde1196803e15018bfb7a329d6f0b",
  "commit_psbt": null,
  "inscriptions": [
    {
      "destination": "tb1pvkpy70tyznl9t0exrfdgqlh436cypz5sja2tfgafxsmu44h6x6tsnfjz2m",
      "id": "bff0642c5ccd1ba1cd863cd1e6c06491e4adc0ec3accc22b48858ab988879c9ci0",
      "location": "bff0642c5ccd1ba1cd863cd1e6c06491e4adc0ec3accc22b48858ab988879c9c:0:0"
    }
  ],
  "parents": [],
  "reveal": "bff0642c5ccd1ba1cd863cd1e6c06491e4adc0ec3accc22b48858ab988879c9c",
  "reveal_broadcast": true,
  "reveal_psbt": null,
  "rune": null,
  "total_fees": 5110
}
```

Рисунок 4.9 – Результат виконання команди інскрипції

Процес інскрипції передбачає два основних етапи:

- Створення чорнової транзакції з маркуванням виходу;
- Публікація готової транзакції з даними, що містять сам цифровий об’єкт.

Після підтвердження такої транзакції у блокчейні Bitcoin, у Signet це відбувається майже миттєво, інскрипція стає частиною ланцюга блоків. Та через деякий час блок з транзакцією підтверджуєтеся майнером. Зображено на рисунку 4.10.

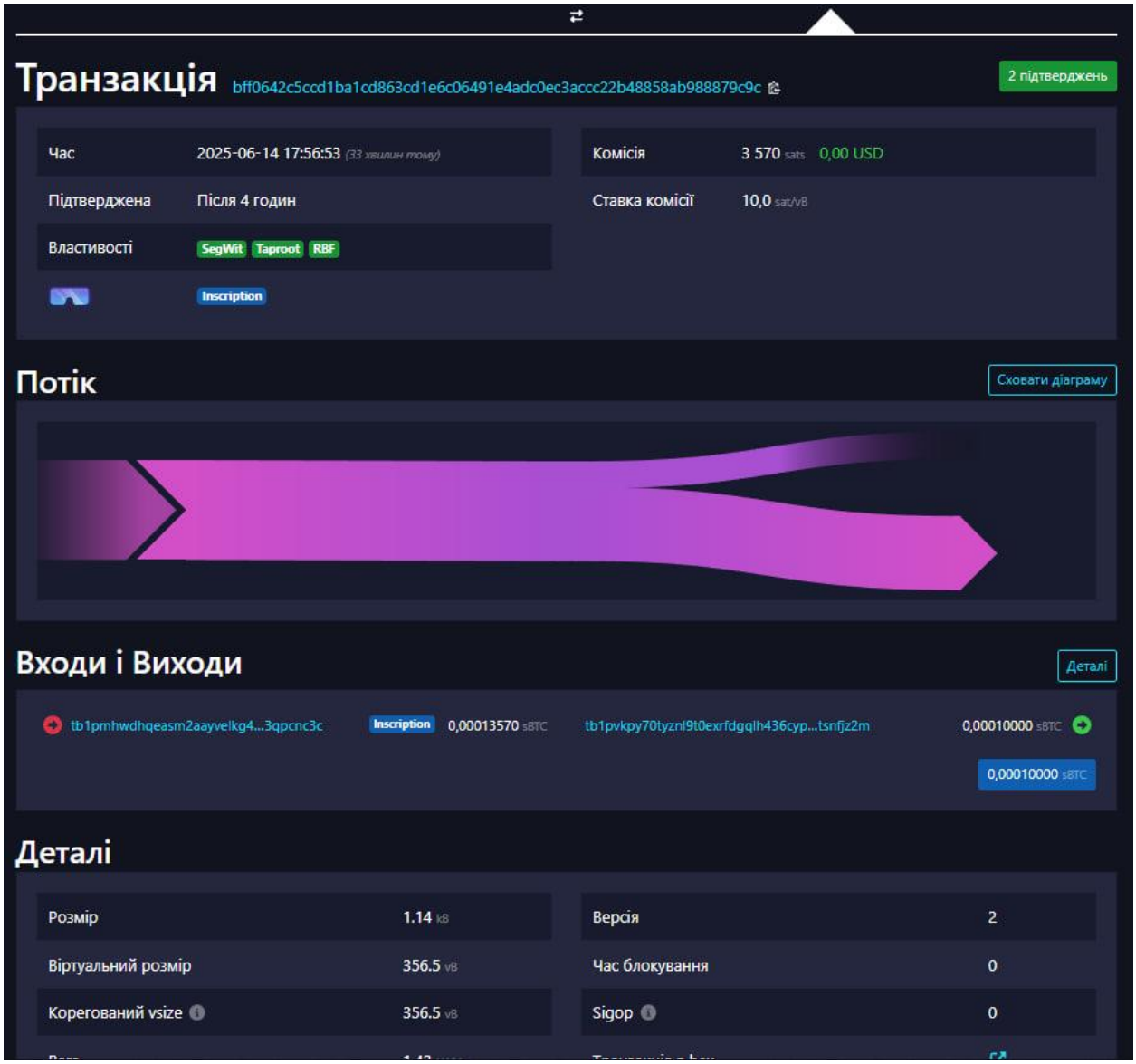


Рисунок 4.10 – Підтвердження транзакції в mempool

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було здійснено комплексне дослідження та практична реалізація процесу створення унікальних цифрових об’єктів у мережі Bitcoin за допомогою інскрипцій протоколу Ordinals. У рамках дослідження було розгорнуто повноцінне середовище для тестування, яке включає запуск локального вузла Bitcoin Core у режимі Signet, налаштування параметрів конфігурації для роботи з RPC та повною індексацією транзакцій, а також інтеграцію з утилітою ord, що забезпечує функціонал для створення Taproot-гаманця, генерації адрес, управління UTXO та безпосереднього створення інскрипцій.

У ході реалізації було здійснено отримання тестових біткоїнів через фаусет, створено та оптимізовано зображення відповідно до обмежень мережі Signet, після чого проведено інскрипцію даного файлу в блокчейн та перевірено її наявність за допомогою block explorer. Отримані результати підтвердили працездатність та ефективність описаного підходу до розміщення цифрового контенту безпосередньо в ланцюгу блоків Bitcoin, без потреби у зовнішньому сховищі. Таким чином, кваліфікаційна робота підтвердила перспективність використання Ordinals як інноваційного методу створення NFT у межах біткоїн-екосистеми, який забезпечує високу ступінь децентралізації, довговічності зберігання та прозорості для кінцевого користувача.

Результати дослідження можуть бути використані як основа для подальшої розробки користувацьких інтерфейсів, інструментів автоматизації створення інскрипцій, а також для вивчення економічної доцільності та правових аспектів збереження цифрових активів у блокчейні.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System [Електронний ресурс] / Satoshi Nakamoto. – 2008. – Режим доступу: https://bitcoin.org/bitcoin.pdf

2. Antonopoulos, A. M. Mastering Bitcoin: Unlocking Digital Cryptocurrencies / Andreas M. Antonopoulos. – 2nd ed. – Sebastopol, CA: O'Reilly Media, 2017. – 416 с.

3. Narayanan, A. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction / Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder. – Princeton: Princeton University Press, 2016. – 336 с.

4. Hasu; Zhu, S. Anatomy of Proof-of-Work [Електронний ресурс] / Hasu, Su Zhu // Unchained Capital, 2018. – Режим доступу: https://unchained.com/blog/anatomy-of-proof-of-work

5. Rodarmor, C. The Ordinals Protocol [Електронний ресурс] / Casey Rodarmor. – 2023. – Режим доступу: https://docs.ordinals.com

6. Ethereum Foundation. EIP-721: Non-Fungible Token Standard [Електронний ресурс]. – Режим доступу: https://eips.ethereum.org/EIPS/eip-721

7. Metaplex Foundation. Metaplex Token Metadata Standard [Електронний ресурс]. – Режим доступу: https://docs.metaplex.com/

8. Taproot Wizards. Taproot and Bitcoin Upgrades Explained [Електронний ресурс]. – 2022. – Режим доступу: https://taproot.wtf

9. Protocol Labs. InterPlanetary File System (IPFS) Documentation [Електронний ресурс]. – Режим доступу: https://docs.ipfs.tech

10. Binance Academy. What Is Blockchain Bloat? [Електронний ресурс]. – Режим доступу: https://academy.binance.com/en/articles/what-is-blockchain-bloat

11. The Bitcoin Signet Network [Електронний ресурс]. – Режим доступу: https://bitcoin.sipa.be/signet/

12. Rodarmor, C. Sending Ordinals on Signet [Електронний ресурс]. – Режим доступу: https://rodarmor.com/blog/sending-ordinals-on-signet/

13. OrdinalsBot. 2023 Year in Review [Електронний ресурс]. – Режим доступу: https://blog.ordinalsbot.com/2023-year-in-review

14. Bitcoin Core documentation – JSON-RPC, txindex and network configs [Електронний ресурс]. – Режим доступу: https://bitcoincore.org/en/doc/

15. Ordinals. Ord CLI GitHub repository – Installation and usage [Електронний ресурс]. – Режим доступу: https://github.com/ordinals/ord

16. Жигалка М.І., Сітніков В.І Мультиагентні системи та їх застосування в логістиці та економіці: тези доповідей п’ятнадцятої міжнародної науково-технічної конференції. Т.3: секції 3,4. Баку: ІСУ АР; Харків: НТУ «ХПІ»;. Харків: ХНУРЕ; Харків: НАУ «ХАІ»; Жиліна: УмЖ. 2025. С. 49.