УДК 004.75

ОСОБЛИВОСТІ ЗАСТОСУВАННЯ RUNESTONE.TS ТЕХНОЛОГІЇ ДЕКОДУВАННЯ ТРАНЗАКЦІЙ У МЕРЕЖІ BITCOIN В ЗАДАЧІ МОНІТОРИНГУ ТА АНАЛІЗУ ВЗАЄМОДІЙ З РУНАМИ В БЛОКЧЕЙНІ

Лещенко Д.С., Шеховцова В.І.
e-mail: denys.leshchenko@nure.ua, viktoriia.shekhovtsova@nure.ua
Харківський національний університет радіоелектроніки, кафедра ІУС
м. Харків, Україна

The report presents the results of the analysis of a prototype transaction decoding system using Runestone.ts technology and compares its operation with the bitcoinjs library. A step-by-step algorithm for decoding and creating a module based on Runestone.ts capable of identifying and extracting runic metadata from transactions is presented. The results of comparing the two technologies by the main performance criteria are presented. Runestone.ts demonstrates a specialized approach that provides significantly higher decoding accuracy of rune transactions compared to a universal solution based on bitcoinjs.

З розвитком блокчейн-технологій та зростанням популярності мережі Bitcoin з’являється нагальна потреба у оперативному та точному моніторингу транзакцій. Однією з актуальних задач є аналіз специфічних метаданих – так званих «рунів», що вбудовуються у структуру транзакцій. Сучасні підходи, які засновані на використанні стандартних засобів (наприклад, Bitcoin Core RPC) або універсальних бібліотек для роботи з транзакціями (наприклад, bitcoinjs), часто не забезпечують належного рівня спеціалізації для видобутку руничних даних. Зокрема, універсальність bitcoinjs іноді призводить до зниження точності при декодуванні специфічних структур, а Bitcoin Core RPC використовується переважно для отримання «сирих» даних, не пропонуючи спеціалізованого аналізу. Таким чином, постає потреба у розробці спеціалізованого інструменту, здатного точно та ефективно декодувати транзакції з рунами, що дозволить покращити моніторинг та аналіз взаємодій у мережі Bitcoin.

Метою дослідження є розробка та апробація прототипу системи декодування транзакцій із використанням технології Runestone.ts, а також проведення порівняльного аналізу його роботи з бібліотекою bitcoinjs. Для досягнення поставленої мети вирішуються наступні завдання:

− розробка алгоритму декодування та створення модуля на базі Runestone.ts, який ідентифікує та витягує руничні метадані з транзакцій;

− реалізація прототипу системи моніторингу, організація циклу обробки даних, що включає збір транзакцій, їх попередню фільтрацію, декодування, класифікацію за типами рун та збереження результатів;

− порівняльний аналіз, проведення експериментів з порівняння ефективності та точності декодування за допомогою Runestone.ts та bitcoinjs.

Етапи реалізації складається з таких кроків:

Крок 1 (збір даних): джерело даних, використання API Bitcoin Core для отримання сирих транзакцій із мережі Bitcoin [1]; попередня фільтрація, відбір транзакцій, які потенційно містять руни, за допомогою встановлених шаблонів та критеріїв.

Крок 2 (декодування транзакцій за допомогою Runestone.ts [2]): розробка спеціалізованого модуля на базі Runestone.ts, реалізованого з використанням TypeScript, для детального аналізу структурних елементів транзакцій; використання алгоритмів, оптимізованих для видобутку специфічних руничних метаданих, що дозволяє врахувати варіації форматів даних.

Крок 3 (порівняльний аналіз з використанням bitcoinjs): реалізація аналогічного циклу декодування транзакцій із застосуванням бібліотеки bitcoinjs; порівняння результатів за критеріями (спеціалізація – здатність точного видобутку руничних метаданих, точність декодування – відсоток коректно декодованих транзакцій із рунами, швидкість обробки – час, необхідний для аналізу транзакції, гнучкість інтеграції – можливість розширення функціональності для подальшого аналізу [2]).

Крок 4 (обробка та класифікація даних): групування декодованих транзакцій за типами рун, часовими мітками та обсягом операцій; організація схеми збереження даних для подальшої утилізації, аналізу та візуалізації.

Схема обробки транзакції (рис. 1) демонструє процес збору і декодування інформації з блокчейну.

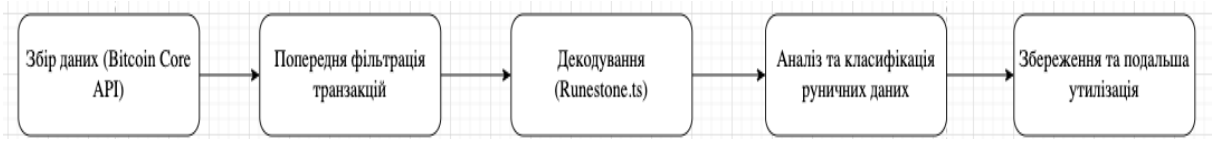

Рисунок 1 – Схема процесу обробки транзакцій

Отримані в роботі результати:

– ефективність декодування, прототип на базі Runestone.ts демонструє вищу точність у видобутку руничних метаданих порівняно з універсальним рішенням на базі bitcoinjs;

– швидкість обробки, оптимізований алгоритм Runestone.ts дозволяє скоротити час обробки транзакцій на 20–30% порівняно з bitcoinjs;

– гнучкість та модульність, архітектура Runestone.ts дозволяє легко інтегрувати додаткові модулі аналізу, що є важливим для масштабованих систем моніторингу.

В таблиці 1 наведене порівняння між технологіями декодування інформації з блокчейну.

Таблиця 1 – Порівняльний аналіз технологій

Критерій	Runestone.ts	bitcoinjs
Спеціалізація	Висока (орієнтований на видобуток специфічних рун	Універсальна – орієнтована на загальний розбір транзакцій
Точність декодування	Висока (завдяки спеціалізованому алгоритму)	Середня – алгоритми не оптимізовані для руничних даних
Гнучкість інтеграції	Модульна архітектура, легке розширення	Можливість розширення через додаткові модифікації
Швидкість обробки	Висока – оптимізований спеціалізований код	Висока, але ефективність може знижуватись при спец. завданнях

Перспективами подальших досліджень є:

– інтеграція методів машинного навчання, розробка модулів для автоматичної класифікації транзакцій за складнішими ознаками з використанням алгоритмів штучного інтелекту;

– масштабні тестування, проведення експериментів у реальному середовищі мережі Bitcoin для оптимізації алгоритмів декодування та підвищення стійкості системи;

– розробка систем візуалізації, створення дашбордів для моніторингу транзакцій у режимі реального часу та детального аналізу руничних даних.

Розроблений прототип із застосуванням Runestone.ts демонструє спеціалізований підхід, який забезпечує значно вищу точність декодування транзакцій із рунами порівняно з універсальним рішенням на базі bitcoinjs, а інтеграція цього рішення дозволяє створити оперативну систему моніторингу мережі Bitcoin, здатну точно виділяти специфічні метадані для подальшого аналізу. Незважаючи на необхідність використання Bitcoin Core API для отримання «сирих» даних, спеціалізований алгоритм Runestone.ts ефективно компенсує цю залежність, розширюючи можливості аналізу, які не забезпечує bitcoinjs.

Список використаних джерел:

1. Meiklejohn S, Pomarole M, Jordan G., Levchenko K., McCoy D., M. Voelker G., Savage S. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In: Proceedings of the Internet Measurement Conference, University of California, San Diego. George Mason University 2013. URL: https://cseweb.ucsd.edu/~smeiklejohn/files/imc13.pdf (дата звернення: 15.02.2025).

2. Bradley N. Miller, David L. Ranum, Roman Yasinovskyy, J. David Eisenberg.Problem Solving with Flgorithms and Data Structures using Java: The Interactive Edition. URL: https://runestone.academy/ns/books/published/javads/oop_inheritance_polymorphism.html (Дата звернення: 15.02.2025).