DOI 10.31891/2307-5732-2024-341-5-3
УДК 004.65:519.17

ЖИКІН ЮРІЙ
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"
https://orcid.org/0009-0001-5930-1444
e-mail: yzhykin@protonmail.com
ОНАЙ МИКОЛА
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"
https://orcid.org/0000-0002-4938-8355
e-mail: onay@pzks.fpm.kpi.ua

RDF-МОДЕЛЬ ГРАФА БІТКОЇН-ТРАНЗАКЦІЙ

Ланцюг блоків Біткоїна містить історію всіх Біткоїн-транзакцій від початку роботи мережі. Для того, щоб кожен учасник мережі міг легко перевірити відповідність будь-якої транзакції набору характеристик, що описують правильну транзакцію, вся інформація в транзакції, а саме кількість біткоїна, що змінила власність, попередні власники та нові власники, є відкритою. Сукупно ця інформація утворює граф Біткоїн-транзакцій, що відображає історію зміни власності одиниць біткоїна від початку роботи мережі. Цей граф може бути розширений інформацією про зв'язки між ідентичностями користувачів Біткоїн-мережі (електронними поштами тощо) та псевдонімними ідентичностями Біткоїн-мережі (Біткоїн-адресами), і такий розширений граф може використовуватись для аналізу грошових потоків між користувачами Біткоїн-мережі. Вивчення можливостей такого аналізу є дуже важливим з точки зору захисту транзакційної приватності індивідуального користувача.

У даному дослідженні пропонується модель графа Біткоїн-транзакцій на основі технології Інфраструктури Опису Ресурсів, що дозволяє швидко знаходити попередні та наступні транзакцій на довільній відстані від транзакції, що розглядається, а також розширювати граф транзакцій анотаціями про можливі зв'язки між транзакціями та пов'язаними зовнішніми даними без необхідності змінювати схему бази даних. Також у дослідженні порівнюється швидкість пошуку попередніх та наступних транзакцій у RDF-представленні графа та альтернативних представленнях.

Описана модель може бути використана на практиці для побудови графа Біткоїн-транзакцій у вигляді множини тверджень у базі даних, що може зберігати RDF-трійки, та здійснення пошуку патернів у такому графі за допомогою будь-якої мови запитів, що підтримується такою базою даних.

Ключові слова: біткоїн, криптовалюта, блокчейн, криптографія, еліптична крива, граф транзакцій, граф знань, онтологія, RDF, SPARQL.

YURI ZHYKIN
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"
MYKOLA ONAI
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute"

RDF MODEL OF BITCOIN TRANSACTION GRAPH

The Bitcoin blockchain contains the history of all Bitcoin transactions since the start of Bitcoin network operation. In order to allow each network participant to easily verify the compliance of any transaction with the set of characteristics that describe a valid transaction, all the information in the transaction, namely the amount of Bitcoin transferred, previous owners, and new owners, is fully open. Aggregated, this information forms the Bitcoin transaction graph, which shows the history of ownership changes of Bitcoin units since the start of network operation. This graph can be extended with information about the connections between Bitcoin network user identities (such as email addresses) and pseudonymous identities within the Bitcoin network (Bitcoin addresses), and such an extended graph can be used to analyze money flows between Bitcoin network users. Studying the possibilities of such analysis is crucial for protecting the transactional privacy of individual users.

This study proposes a model of the Bitcoin transaction graph based on the Resource Description Framework (RDF) technology, which allows for the quick identification of previous and next transactions at any distance from the transaction under consideration, as well as the extension of the transaction graph with annotations about possible connections between transactions and related external data without the need to modify the database schema. The study also compares the performance of the search for previous and next transactions in the RDF representation of the graph and alternative representations.

The described model can be practically used to build a Bitcoin transaction graph in the form of a set of statements in a database capable of storing RDF triples as well as search for patterns in such a graph using any query language supported by such a database.

Keywords: Bitcoin, cryptocurrency, blockchain, cryptography, elliptic curve, transaction graph, knowledge graph, ontology, RDF, SPARQL.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Біткоїн – це перша децентралізована система електронних грошей з можливістю програмування семантики транзакцій, що була запропонована і введена в експлуатацію особою під псевдонімом Сатоші Накамото [1] у 2008 році, і з тих пір набирає все більшої популярності по всьому світу. Зокрема у 2024 році Біткоїн є легальним платіжним інструментом у Сальвадорі та ЦАР, а офіційний акаунт України у соціальній мережі Twitter публікував Біткоїн-адресу для пожертвувань 26 лютого 2022 року, на початку широкомасштабного вторгнення Росії в Україну [2]. Популярність Біткоїна продовжує зростати, і на даний момент навколо Біткоїн-мережі сформувалась активна екосистема, що складається з спільноти розробників програмного забезпечення, своєрідного політичного та економічного світогляду, сервісів, освітніх проєктів.

Через зростання популярності багато нових користувачів Біткоїна не мають того рівня технічних знань,

який був поширеним серед перших користувачів, більшість з яких були криптографами та розробниками програмного забезпечення, тому все більш важливою стають різні аспекти безпеки повсякденного користування Біткоїном, зокрема безпека приватності персональних даних.

Біткоїн-протокол підтримує у кожного учасника мережі повну копію всієї історії транзакцій від початку роботи мережі, і безпека Біткоїн-системи базується на тому, що кожен учасник у будь-який момент може перевірити цілісність і правильність стану власності на всіма біткоїнами у обігу, який обчислюється безпосередньо з записів про транзакції. Для цього усі дані у транзакціях, а саме інформація про кількість біткоїна, яка була надіслана, попередній власник та наступний власник, є відкритими і утворюють граф грошових потоків в системі. Будь-який учасник мережі, що має власну копію історії транзакцій, може проглянути цю інформацію для будь-якої транзакції, що відбулась з моменту початку існування Біткоїн-мережі. Оскільки історичні дані не змінюються з часом, що гарантується протоколом, зацікавлена сторона може побудувати базу даних Біткоїн-транзакцій, доповнену інформацією про зв'язки між ідентичностями користувачів та псевдонімними ідентичностями, що існують в межах Біткоїн-системи (Біткоїн-адреси), і використовувати цю базу даних з метою деанонімізації користувачів Біткоїна.

Одним з найпростіших прикладів використання такої бази даних є встановлення кількості біткоїна, що належить тій чи іншій людині, що може бути використано в злочинних цілях. На даний момент існує кілька компаній, що займаються аналізом транзакційних даних, але методи, що використовуються, не є публічними. На думку авторів, публічні досліджень методів аналізу графа Біткоїн-транзакцій важливі в першу чергу для розуміння ризиків для індивідуального користувача, оскільки «скриньку Пандори» аналізу транзакційних даних у Біткоїн-системі вже неможливо закрити. Розуміння можливостей аналізу графа транзакцій дозволить краще розуміти, яким чином повинні конструюватись Біткоїн-транзакції та їх послідовності, щоб уникати витоків даних, які можуть призвести до порушення транзакційної приватності користувача.

Аналіз досліджень та публікацій

У попередньому дослідженні [3] розглядається загальний підхід до аналізу графа Біткоїн-транзакцій з використанням пошуку патернів типових операцій. Зокрема пропонується процес побудови графа знань про транзакції загальний метод патернового аналізу графа шляхом візуального пошуку патернів типових операцій, у даному дослідженні розглядається модель графа Біткоїн транзакцій у конкретному представленні, адаптованому до існуючих інструментів автоматизованого пошуку патернів у графі. Типові операції, подані у [3] у вигляді схематичних зображень, у даному дослідженні сформульовані у вигляді запитів мовою, що призначена для ефективного пошуку патернів у великих графах, які зберігаються у графових базах даних.

Формат, що пропонується у даному дослідженні для представлення графа транзакцій, називається Інфраструктурою Опису Ресурсів (англ. Resource Description Framework, RDF) [4]. RDF – це напрямлений граф, що складається з тверджень-трійок «суб'єкт», «предикат» та «об'єкт». Кожна така трійка є ребром у графі між двома вершинами-ресурсами, а кожен компонент трійки (суб'єкт, предикат та об'єкт) є уніфікованим ідентифікатором ресурсу (URI). Графові бази даних, що можуть безпосередньо зберігати RDF-трійки, називаються сховищами трійок.

Серед переваг RDF-представлення, що є важливими для задачі побудови графа знань про Біткоїн транзакції, є безсхемовість (граф може мати довільну кількість типів вершин і довільну кількість типів зв'язків між вершинами, а отже додавання нових анотацій до інформації про транзакції у графі не потребуватиме змін в існуючому графі) та можливість швидкого обходу графа вздовж ребер на довільну відстань (ключова властивість, необхідна для ефективного пошуку патернів у графі). Пошук патернів у RDF-графах можна здійснювати за допомогою однієї з спеціальних мов запитів, таких як SPARQL [5].

Формулювання цілей статті

Метою роботи є розробка RDF-моделі графа Біткоїн-транзакцій що дозволяє здійснювати швидкий пошук транзакційних патернів за допомогою стандартизованої мови запитів SPARQL а також розширювати граф транзакцій анотаціями з зовнішніх джерел без необхідності зміни схеми бази даних.

Для досягнення даної мети у даному дослідженні ставляться наступні задачі:

- розробити онтологію графа Біткоїн транзакцій, що описує ключові класи ресурсів та можливі зв'язки між ними;
- побудувати аналогічну реляційну модель для розробленої онтології та порівняти швидкість пошуку патернів транзакцій на певній відстані від транакції, що розглядається, у RDF-моделі та реляційній моделі;
- розглянути типові операції, описані у [3] як SPARQL-запити у RDF-моделі графа Біткоїн-транзакцій.

У даній роботі пропонується RDF-модель графа Біткоїн-транзакцій, представлена у вигляді множини RDF-трійок у графовій базі даних AllegroGraph [6], що підтримує SPARQL як основну мову запитів для пошуку патернів у графах.

Також під час проведення дослідження було розроблено компонент програмного забезпечення [7] для трансформації об'єктної моделі Біткоїн-транзакцій і їх входів та виходів у RDF-трійки для подальшого зберігання їх у базі даних AllegroGraph та набір запитів мовою SPARQL для пошуку патернів типових операцій. Ці запити наведені нижче.

Виклад основного матеріалу

Розглянемо онтологію Біткоїн-транзакцій, а саме класи ресурсів у RDF-графі транзакцій та можливі

типи зв'язків між ними. З метою забезпечення сумісності з існуючими інструментами для роботи з RDF-даними використовуватимемо стандартні відношення, де це можливо. RDF-трійки у цьому розділі подані у синтаксисі Turtle [8]. Також для зручності як у ідентифікаторах ресурсів класів та відношень онтології, так і у ідентифікаторах ресурсів самого графа транзакцій використовується префікс «bp».

Найбільш важливим класом ресурсів у RDF-графі Біткоїн-транзакцій є клас Output або клас транзакційних виходів. Ресурси даного класу представляють певні кількості біткоїна, які належали конкретному власнику в певний момент часу. Транзакційний вихід однозначно ідентифікується парою значень «ідентифікатор транзакції» - «індекс виходу», тому використаємо цю інформацію для ідентифікації відповідних ресурсів. Основною властивістю транзакційного виходу є кількість біткоїна, яку він містить, для цього використаємо RDF-властивість bp:amount. Окрім того, більшість транзакційних виходів є так званими стандартними виходами, які мають чітко визначену структуру і визначений формат адреси. Оскільки адреса часто є ідентифікатором, який може зустрітись в зовнішніх джерелах інформації, нам потрібно включити адресу в граф таким чином, щоб інструменти візуалізації показували її в першу чергу, тому використаємо стандартну властивість rdfs:label. Приклад опису транзакційного виходу у вигляді RDF-трійок:

```
bp:cfa1924e4b962ab1aec214ecc8ac13df67cc429ab016fb9c11bab3d18465f819:1
   a bp:Output;
   bp:amount 10000;
   rdfs:label bc1prj2jre963376ly0x6w3ajxddrg236etskn2ljy8ef3ncf7lzgkzs0r4yrj;
```

Наступною важливою структурою є власне сама транзакція, якій відповідає клас Tx. Ресурси даного класу групують ресурси класу Output виходи у множини входів та виходів даної транзакції. Обчислення ідентифікатора транзакції визначене Біткоїн-протоколом, тому ми безпосередньо використовуємо ідентифікатор транзакції як ідентифікатор відповідного ресурсу. Приклад опису транзакції у вигляді RDF-трійок:

```
bp:cfa1924e4b962ab1aec214ecc8ac13df67cc429ab016fb9c11bab3d18465f819
   a bp:Transaction;
   bp:output bp:cfa1924e4b962ab1aec214ecc8ac13df67cc429ab016fb9c11bab3d18465f819:1;
   bp:output bp:cfa1924e4b962ab1aec214ecc8ac13df67cc429ab016fb9c11bab3d18465f819:2.
```

Транзакційні входи не містять важливої інформації, вони є допоміжними структурами даних, тому відсутні у запропонованій RDF-моделі з метою зменшення розміру графа. Натомість, у графі безпосередньо представлені зв'язки між транзакційними виходами та транзакціями, для яких вони є входами, за допомогою відношення bp:input, наприклад:

```
bp:91431e82c2452e083602f4252f8a42e95501bd6cf897a74a7ff97e9ff6e6c923:2
   a bp:Output;
   bp:amount 538700;
   bp:input bp: cfa1924e4b962ab1aec214ecc8ac13df67cc429ab016fb9c11bab3d18465f819.
```

Оскільки патерни типових операцій, зокрема ті, що описані у [3], часто використовують інформацію про кількість виходів транзакцій для визначення можливого типу операції (наприклад транзакція консолідації виходів має велику кількість входів, але лише один вихід), в поточній моделі доведеться здійснювати агрегацію проміжних результатів для підрахунку кількості виходів та входів, що може бути доволі складним з точки зору часу обчисленням для великого підграфа. Оскільки ця інформація присутня під час трансформування об'єктної моделі транзакції у RDF-представлення, пропонується ввести дві додаткові властивості транзакції, bp:inputCount та bp:outputCount, об'єктами яких є цілочисельні значення кількості входів та виходів транзакції відповідно.

Користувачі можуть бути представлені довільними ресурсами і пов'язані з транзакційними виходами довільними RDF-зв'язками. На рис. 1 подано наведено модель типової Біткоїн-транзакції (T3), що має 2 входи (O1 та O2) та 2 виходи (O4 та O5). Також у графі для прикладу зображений ресурс користувача (E1), пов'язаний відношеннями bp:outputOwner з транзакційними виходами O1, O2 та O5.

Побудувавши граф Біткоїн-транзакцій у RDF-представленні, ми можемо подати патерни типових операцій, описані у [13], у вигляді SPARQL-запитів у сховище RDF-трійок.

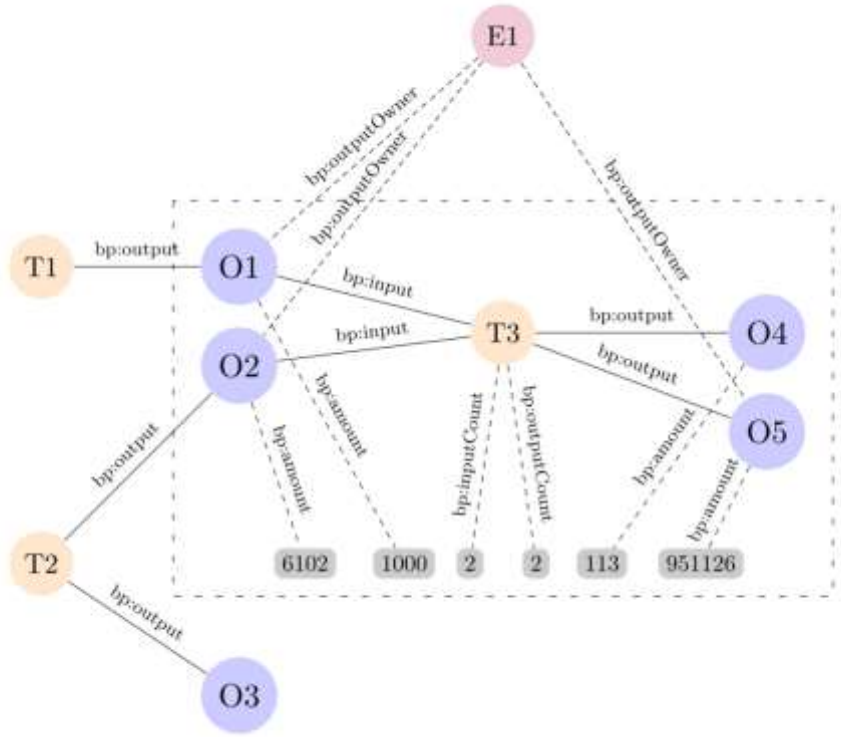

Рис. 1. RDF-модель типової Біткоїн-транзакції

Транзакції консолідації виходів – це транзакції, що містять багато входів і лише один вихід. Усі такі транзакції можна знайти у RDF-графі за допомогою наступного запиту:

```
SELECT ?t
{
 ?t bp:inputCount ?icount.
 ?t bp:outputCount ?ocount.
 FILTER (?icount > 10 && ?ocount = 1)
}
```

Типові CoinJoin-транзакції мають велику кількість входів та таку ж кількість виходів з однаковою сумою біткоїна. Усі такі транзакції можна знайти за допомогою наступного запиту (перевірка на однакові кількості пропущена):

```
SELECT ?t
{
 ?t bp:inputCount ?icount.
 ?t bp:outputCount ?ocount.
 FILTER (?icount > 10 && ?ocount = ?icount)
}
```

Ланцюжки тимчасових адрес – це багатокрокової операції, що складаються з послідовностей транзакцій з одним входом та одним виходом, і часто можуть вказувати на спробу власника біткоїна приховати свій кінцевий транзакційний вихід. Усі такі операції з довжиною ланцюжка 4 можна знайти у графі за допомогою наступного запиту:

```
SELECT ?t1 ?t4
{
 ?t1 bp:inputCount ?icount1.
 ?t1 bp:outputCount ?ocount1.
 FILTER (?icount1 = 1 && ?ocount1 = 1)
 ?t1 bp:output/bp:input ?t2.
 ?t2 bp:inputCount ?icount2.
 ?t2 bp:outputCount ?ocount2.
```

```
  FILTER (?icount2 = 1 && ?ocount2 = 1)
  ?t2 bp:output/bp:input ?t3.
  ?t3 bp:inputCount ?icount3.
  ?t3 bp:outputCount ?ocount3.
  FILTER (?icount3 = 1 && ?ocount3 = 1)
  ?t3 bp:output/bp:input ?t4.
  ?t4 bp:inputCount ?icount4.
  ?t4 bp:outputCount ?ocount4.
  FILTER (?icount4 = 1 && ?ocount4 = 1)
}
```

Розглянемо випадок, коли досліджується конкретна Біткоїн-адреса, яка пов'язана з певною ідентичністю (наприклад, публічна адреса для пожертвувань якогось сервісу). Один з найпростіших і найбільш важливих початкових пошуків у графі транзакцій – це пошук множини транзакційних виходів, які були використані для надсилання коштів на адресу, що розглядається, як безпосередньо, так і через певну кількість проміжних адрес. SPARQL-запит для пошуку такого підграфа транзакцій виглядає наступним чином (параметр N – максимальна глибина пошуку або максимальна відстань від заданої адреси в графі транзакцій):

```
SELECT ?source
{
  BIND ("bc1ptxh27...lfm8yj" AS ?address)
  ?target rdfs:label ?address.
  ?source (bp:input/bp:output){1, N} ?target.
}
```

Швидкість виконання такого запиту для різних Біткоїн-адрес наведено в таблиці 1. Для вимірювання використовувався граф Біткоїн-транзакцій для 380-ти блоків, що містить 1745281 транзакцій. Для порівняння у таблиці також наведено швидкість виконання аналогічного запиту у реляційній базі даних, що містить ті ж транзакції.

Таблиця 1

Швидкість виконання запитів для пошуку всіх попередніх транзакційних виходів

Максимальна глибина пошуку N	К-сть знайдених виходів	Час SPARQL-запиту, с	Час SQL-запиту, с
1	67	0.017	0.164
2	98	0.052	0.343
4	116	0.066	2.829
8	118	0.070	15.239

Причиною такої різниці в швидкості виконання подібного запиту є те, що запит у реляційній базі обчислює з'єднання (вибірку над декартовим добутком) проміжних результатів для кожного кроку вглиб графа, тоді як безпосереднє представлення зв'язків у RDF-графі дозволяє уникнути побудови повної множини проміжних результатів.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

Запропонована RDF-модель графа Біткоїн-транзакцій дозволяє формулювати ефективні SPARQL-запити для пошуку патернів типових операцій у графі, і при цьому дозволяє розширювати граф додатковими вершинами, що представляють зовнішні ідентичності користувачів, та зв'язками між транзакціями та їх виходами, що не є частиною самого графа транзакцій. Таким чином в процесі накопичення інформації утворюється RDF-граф знань про Біткоїн-транзакцій, який може бути використаний для пошуку типових інформацій зв'язків транзакцій з зовнішніми ідентичностями користувачів на пов'язані транзакції. RDF-представлення графа дозволяє значно ефективніше, порівняно з реляційним представленням, здійснювати пошук пов'язаних транзакцій вглиб графа, що є одним з найбільш важливих типів пошуку для аналізу графа транзакцій.

Подальші дослідження, як було зазначено і у [3], полягають у виділенні типових багатокрокових Біткоїн-операцій як патернів у графі транзакцій та формулювання цих патернів у вигляді SPARQL-запитів для автоматизованого пошуку. Слід також зазначити, що дана RDF-модель може з певними змінами бути використана для будь-яких транзакційних систем, у яких транзакція містить інформацію про надсилачів та отримувачів.

References

1. Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf.
2. Stand with the people of Ukraine. Now accepting cryptocurrency donations. Bitcoin, Ethereum and USDT. https://x.com/Ukraine/status/1497594592438497282.
3. Zhykin, Y., Onai, M. (2024). Pattern-based Bitcoin Transaction Graph Analysis. Herald of Khmelnytskyi National University. Technical sciences. 333, 2 (Apr. 2024), 322–328. DOI: https://doi.org/10.31891/2307-5732-2024-333-2-51.
4. Resouce Description Framework (RDF). https://www.w3.org/RDF/.
5. SPARQL 1.1 Query Language. https://www.w3.org/TR/sparql11-query/
6. AllegroGraph: Knowledge Graph + LLM Solutions. https://allegrograph.com/.
7. BRDF: Bitcoin chain data represented as RDF. https://github.com/rodentrabies/brdf.
8. RDF 1.1 Turtle. https://www.w3.org/TR/turtle/.