DOI: https://doi.org/10.36910/6775-2524-0560-2025-59-15
УДК: 004.056
Жикін Юрій Сергійович, аспірант
http://orcid.org/0009-0001-5930-1444
Онай Микола Володимирович, к.т.н., доцент
http://orcid.org/0000-0002-4938-8355
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», м. Київ, Україна

МЕТОД КОНСОЛІДАЦІЇ ВИХОДІВ БІТКОЇН-ТРАНЗАКЦІЙ ЗА ДОПОМОГОЮ СИМУЛЯЦІЇ ВИТРАЧАННЯ

Жикін Ю. С., Онай М. В. Метод консолідації виходів Біткоїн-транзакцій за допомогою симуляції витрачання. У Біткоїн-протоколі кожна транзакція споживає транзакційні входи – записи, що зв'язують певну кількість одиниць біткоїна з певним власником, за допомогою транзакційних входів – записів-посилань на входи, і створює нові входи з новими кількостями і власниками, відображаючи таким чином акт зміни власності. Користувачі, що часто отримують платежі, накопичують велику кількість дрібних невикористаних виходів, що призводить до підвищення вартості майбутніх транзакцій, зниження швидкості роботи програмного забезпечення та загроз приватності фінансових даних. З метою зменшення кількості виходів користувачі використовують операції консолідації, які об'єднують багато дрібних виходів в один більший. Проте такі транзакції або їх каскади легко ідентифікуються в графі транзакцій, що дозволяє встановити зв'язки між виходами, що беруть участь в одній операції консолідації, погіршуючи фінансову приватність користувача. У цій статті пропонується метод багатокрокової консолідації виходів шляхом симуляції природного витрачання виходів. Для цього використовуються симулюючі транзакції з кількома входами та двома виходами, які виглядають як звичайні платежі з цільовим виходом та виходом-рештою. Запропонований метод ітеративно зменшує множину невикористаних виходів користувача до встановленого розміру без утворення очевидних патернів у графі транзакцій. Аналіз вартості показує, що навіть у випадку, коли користувач жертвує ефективністю заради мінімізації розміру груп пов'язаних виходів, загальна вартість багатокрокової консолідації зростає лише у 2-4 рази порівняно з простою консолідацією. У статті також формулюються напрями подальших досліджень, зокрема пошук патернів у графі транзакцій, що можуть бути випадками застосування складних схем багатокрокової консолідації виходів, а також пошук патернів зворотних операцій розбиття виходів на частини, що може бути ознакою злочинної діяльності, тощо.
Ключові слова: Біткоїн, граф транзакцій, аналіз графа, вартість транзакцій, консолідація, обфускація, приватність, безпека, криптовалюти, криптографія, еліптичні криві.

Zhykin Yu., Onai M. Method of Bitcoin transaction output consolidation using spending simulation. In the Bitcoin protocol, each transaction consumes transaction outputs – records that associate a specific amount of bitcoin with a particular owner – via transaction inputs, which are references to existing outputs, and creates new outputs with new amounts and owners, thus reflecting an act of ownership transfer. Users who frequently receive payments accumulate a large number of small unspent outputs, which increases costs of future transactions, decreases performance of wallet software, and threatens financial privacy. In order to reduce the number of outputs, users perform consolidation operations that merge many small outputs into a larger one. However, such transactions or their cascades are easily identifiable in the transaction graph, which allows linking the outputs involved in a single consolidation operation and thus deteriorates the user's financial privacy. This paper proposes a method of multi-step output consolidation through simulated natural spending. It uses simulating transactions with multiple inputs and two outputs which resemble typical payments with a target output and a change output. The proposed method iteratively reduces the set of user's unspent outputs to a certain target size without creating obvious patterns in the transaction graph. Cost analysis demonstrates that even when the user sacrifices efficiency in favor of minimizing the size of linked output groups, the total cost of multi-step consolidation increases only by a factor of 2-4 compared to simple consolidation. The article also outlines future research directions, including the identification of transaction graph patterns that may indicate the use of complex multi-step consolidation schemes, as well as patterns of reverse operations of splitting outputs into smaller parts, which may indicate illegal activities.
Keywords: Bitcoin, transaction graph, graph analysis, transaction fees, consolidation, obfuscation, privacy, security, cryptocurrencies, cryptography, elliptic curves.

Постановка проблеми. Транзакція у Біткоїн-протоколі складається з структур, які називаються входами та виходами. Вихід моделює власність користувача над певною кількістю біткоїна, і в свою чергу складається з двох компонентів: цілого числа, що представляє кількість базових одиниць біткоїна, та програми блокування, що визначає умову використання цієї кількості біткоїна. Транзакція створює певну кількість нових виходів і для їх створення повинна знищити певну кількість виходів, загальна сума біткоїна у яких є не меншою за кількість біткоїна у створених виходах. Вхід є посиланням на існуючий вихід, який знищується певною транзакцією, і містить програму розблокування, що задовольняє програму блокування відповідного виходу [1]. Весь біткоїн у обігу в кожен момент часу представляється множиною невикористаних виходів, і Біткоїн-транзакція є транзакцією не у фінансовому сенсі, а у сенсі бази даних, що представляє цю множину. Транзакція видаляє з цієї бази даних певну підмножину існуючих виходів і додає нові виходи.

Кожен учасник Біткоїн-мережі підтримує власну копію цієї бази даних і основна функція всього Біткоїн-протоколу – забезпечити цілісність і ідентичність усіх копій цієї бази даних у всіх учасників мережі.

Користувач Біткоїн-системи, що отримує багато платежів, накопичує невикористані виходи, що призводить одночасно до декількох проблем. По-перше, комісія у Біткоїн-протоколі є динамічною, залежить від навантаження на мережу в даний момент, і сплачується за розмір транзакції, оскільки основним ресурсом, який споживає транзакція, є місце у ланцюгу блоків. Якщо у майбутньому користувачу доведеться здійснювати транзакцію на значно більшу суму, ніж типовий платіж, який він отримує, транзакція міститиме багато дрібних входів, і комісія за таку транзакцію може бути дуже високою. По-друге, оскільки програмне забезпечення, що працює з транзакціями (так званий Біткоїн-гаманець) зазвичай використовує алгоритми оптимізації розміру під час підбору входів для нової транзакції, чим більший простір пошуку оптимальної комбінації входів, тим повільніше працюватиме гаманець. Насамкінець, велика кількість невикористаних входів у гаманці ускладнює контроль за тим, щоб виходи з різних джерел не потрапляли в одну транзакцію, коли це може порушувати фінансову приватність користувача [2].

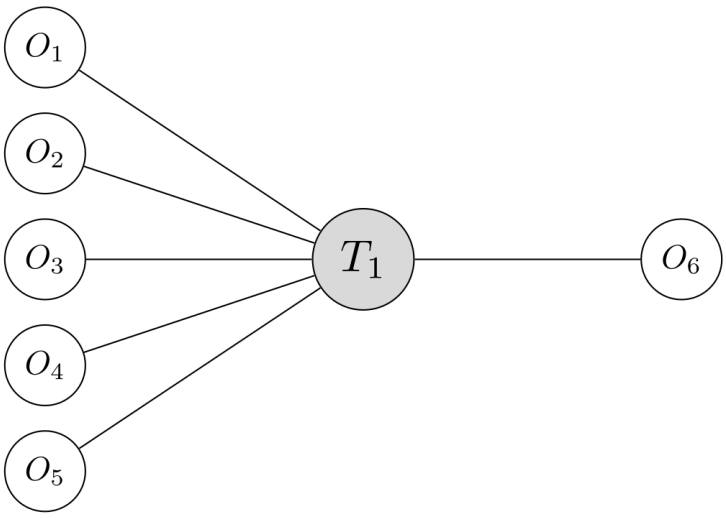

Рис. 1. Операція консолідації виходів

Як наслідок, однією з поширених типових операцій в Біткоїн-мережі є операція консолідації виходів: користувач час від часу надсилає самому собі транзакцію, яка знищує багато дрібних виходів та створює один великий. Такі операції в свою чергу породжують нову проблему: їх легко ідентифікувати під час аналізу графа транзакцій, знайшовши всі транзакції T такі, що кількість входів NIN(T) > 1, а кількість виходів NOUT(T) = 1, тобто транзакцій типу "багато входів, один вихід" (див. рис. 1). Оскільки зміст таких операцій є очевидним, всі виходи у кожній такій транзакції можуть бути позначені як такі, що належать одному й тому ж користувачу, що погіршує фінансову приватність у системі і спрощує подальший аналіз графа транзакцій. Наприклад, користувач, що консолідує кілька сотень виходів, накопичених протягом кількох років, розкриває всю історію отриманих платежів або значну її частину, а також загальну кількість біткоїна, яку він накопичив, і ця інформація може бути використана зловмисниками.

Аналіз існуючих досліджень та публікацій. У статті "Bitcoin Wiki: Privacy" [2] детально розглядаються проблеми приватності Біткоїн-транзакцій, зокрема поширені хибні уявлення, потенційні вразливості, пов'язані з повторним використання адрес, аналізом графа транзакцій і централізованими сервісами, які можуть накопичувати інформацію про своїх клієнтів. Також вміщено загальні рекомендації щодо транзакційної гігієни – набору практик і методів, які можуть підвищити приватність транзакційних даних, зокрема уникання повторного використання адрес та використання так званих протоколів другого рівня, таких як мережі платіжних каналів [3] та CoinJoin [4]. Хоча рекомендації, описані у цій статті, стосуються майже всіх Біткоїн-транзакцій, консолідація виходів у ній не розглядаються.

У роботі під назвою "Studying Bitcoin privacy attacks and their Impact on Bitcoin-based Identity Methods" [5] автори аналізують публікації, пов'язані з деанонімізацією користувачів Біткоїна і

підсумовують, що, що евристичне припущення щодо єдиного власника входів у транзакціях з багатьма входами є одним з найпоширеніших евристичних припущень у різних методах деанонімізації. Однією з перших публікацій, що формулюють це евристичне припущення, є "Quantitative Analysis of the Full Bitcoin Transaction Graph" від 2013 року [6]. Винятком для такого припущення є CoinJoin-транзакції, але їх легко відрізнити від інших транзакцій через особливість роботи протоколу CoinJoin [4]. Транзакція консолідації є особливим випадком транзакції з багатьма входами, для якої можна застосувати додаткове евристичне припущення про те, що і вихід з такої транзакції належить тому ж власнику.

Публікацій, що безпосередньо розглядають обфускацію консолідації транзакційних виходів, на даний момент не існує. При цьому приклади використання деяких примітивних методів обфускації можна знайти в історії транзакцій Біткоїн-мережі. Ймовірним пояснення відсутності публікацій на цю тему є ненадійний але популярний принцип безпеки шляхом приховування деталей (Security through Obscurity) [7]: схема, за допомогою якої досягається обфускація певних операцій в мережі Біткоїн, повинна залишатись в таємниці, щоб мати практичну цінність. Також не виключено, що метод, запропонований нижче, вже використовується у мережі, але не є описаним з тієї ж причини.

Постановка завдання. Оскільки, як було зазначено вище, на даний момент не існує публікацій, що розглядають обфускацію консолідації виходів, то передусім необхідно виділити і розглянути практичні схеми консолідації, що зустрічаються у реальному графі транзакцій. Наступною і основною задачею даного дослідження є розробка методу консолідації виходів, який не встановлює очевидних зв'язків між усіма виходами, які консолідуються, і при цьому є достатньо ефективним з точки зору сумарної комісії за транзакції. Насамкінець, необхідно оцінити вартість запропонованого методу з точки зору комісій за транзакції порівняно з простою консолідацію.

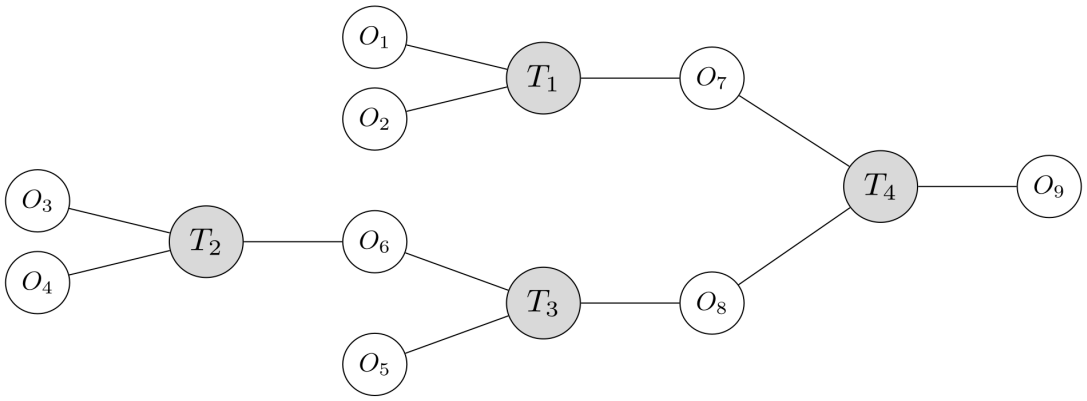

Рис. 2. Дерево багатокрокової консолідації

Виклад основного матеріалу дослідження. Одним з примітивних методів обфускації консолідації виходів є багатокрокова консолідація, під час якої користувач здійснює об'єднання невеликої кількості виходів (2–4) у один. Якщо багатокрокова консолідація збігається у єдиний кінцевий вихід, то такі операції можна ідентифікувати, шукаючи замість єдиної транзакції типу "багато входів, один вихід" дерева таких транзакцій (див. рис. 2). Зміст такого патерну в графі транзакцій є настільки ж очевидним, як і зміст патерну, що відповідає простій операції консолідації. При цьому сумарна комісія за всі транзакції такої багатокрокової консолідації є значно більшою, тому такий метод практично є навіть гіршим за просту консолідацію.

Дещо кращим підходом є часткова консолідація, що не збігається у єдиний кінцевий вихід, а лише зменшує множину невикористаних виходів користувача до певного прийнятного розміру. Наприклад, користувач може зменшити кількість своїх виходів втричі, розбивши їх на групи по 3 виходи і здійснивши одну операцію консолідації для кожної такої групи. Недоліком такої часткової консолідації є те, що в загальному випадку має сенс здійснювати лише одну її ітерацію, оскільки об'єднання виходів попередньої ітерації утворює дерева (рис. 2). Фактично, часткова консолідація з багатьма ітераціями є еквівалентною багатокроковій консолідації.

Якщо користувач отримує і здійснює платежі, і відношення кількості надісланих платежів до кількості отриманих платежів за певний проміжок часу є меншою за певне значення N, то такий

користувач може регулювати кількість накопичених виходів, здійснюючи платежі транзакціями, що мають в середньому N входів, і таким чином консолідуючи виходи шляхом їх природного витрачання. Втім, такий підхід теж має декілька недоліків. По-перше, він не працює у випадках, коли користувач є продавцем товарів чи послуг, і отримує кошти значно частіше, ніж витрачає їх. По-друге, кожна така транзакція розкриває власність над всіма входами у ній контрагенту користувача, який до того ж знає, який саме вихід такої транзакції є цільовим, а який – рештою, і при цьому може сам бути задіяним у зборі інформації про виходи, що належать його контрагентам.

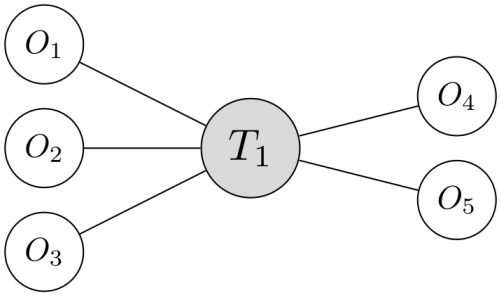

Рис. 3. Типова транзакцій з цільовим виходом та виходом-рештою

Типова транзакція між двома користувачами зображена на рис. 3. Така транзакція зазвичай має декілька входів та 2 виходи: цільовий вихід та вихід-решту. Вихід-решта є необхідним, оскільки Біткоїн-транзакція не може знищити існуючий вихід частково і повинна поглинути його повністю, а різниця між кількістю біткоїна на вході і на виході транзакції автоматично стає комісією за неї [1]. В загальному випадку неможливо відрізнити, який з виходів є цільовим, а який – рештою, але під час аналізу графа транзакцій можуть використовуватись різні евристичні припущення [8], наприклад:

1. Гаманці намагаються мінімізувати розмір кожної створеної транзакції з метою мінімізації комісії, тому, якщо у одному з виходів кількість біткоїна є меншою за кількість у якомусь з входів, можна зробити припущення, що саме цей вихід і є рештою, інакше розмір транзакції не є оптимальним.
2. Оскільки кількість у цільовому виході часто формується людиною, вона з більшою ймовірністю буде невипадковим числом, наприклад 12'300'000 базових одиниць (0.123 біткоїна), тоді як кількість у виході-решті виглядатиме як випадкове число, наприклад 42491457 базових одиниць (0.42491457 біткоїна), якщо сумарна кількість на входів виглядає як випадкове число.

Метод обфускації консолідації виходів, що пропонується у даному дослідження, симулює природну консолідацію шляхом витрачання за допомогою транзакцій, які виглядають як типові транзакції з 2-ма виходами. Симулюючі транзакції повинні мати в середньому більше 2 входів для того, щоб кількість виходів зменшувалась з часом. Кількість входів у симулюючій транзакції визначає швидкість консолідації і обирається залежно від того, що є більш важливим для користувача: мінімізація комісії, чи мінімізації розміру груп виходів, зв'язаних між собою окремими транзакціями. Кількість входів також може обиратись випадково для кожної транзакції для підвищення асиметричності результуючого патерну у графі транзакцій.

Алгоритм консолідації ітеративно зводить кількість наявних виходів до цільової максимальної кількості N за допомогою симулюючих транзакцій з K входами (параметри N та K встановлюються користувачем):

1. Якщо кількість накопичених виходів перевищує значення N, вибираємо K довільних підтверджених невикористаних виходи з різних транзакцій, які будуть входами симулюючої транзакції.
2. Обчислюємо мінімальну кількість біткоїна у вході симулюючої транзакції A_{min} і створюємо 2 нових виходи такі, що кількість біткоїна є випадковим числом, більшим за A_{min}. Таким чином жоден вихід не є більше схожим на вихід-решту, ніж інший, відповідно до зазначених вище евристичних припущень. Список подібних припущень і, відповідно, даний крок алгоритму можуть бути розширені.
3. Конструюємо і публікуємо транзакцію з обраними входами та створеними виходами.

4. Повторюємо кроки 1-3, доки можливо задовольнити умови у кроці 1.

Патерн у графі транзакцій, що породжується цим алгоритмом, є значно складнішим за дерево консолідації (рис. 2) і тому значно гірше піддається автоматизованому розпізнаванню, але оскільки алгоритм можна продовжувати навіть до отримання єдиного виходу, має сенс накладати певні обмеження на використання виходів з попередніх ітерацій.

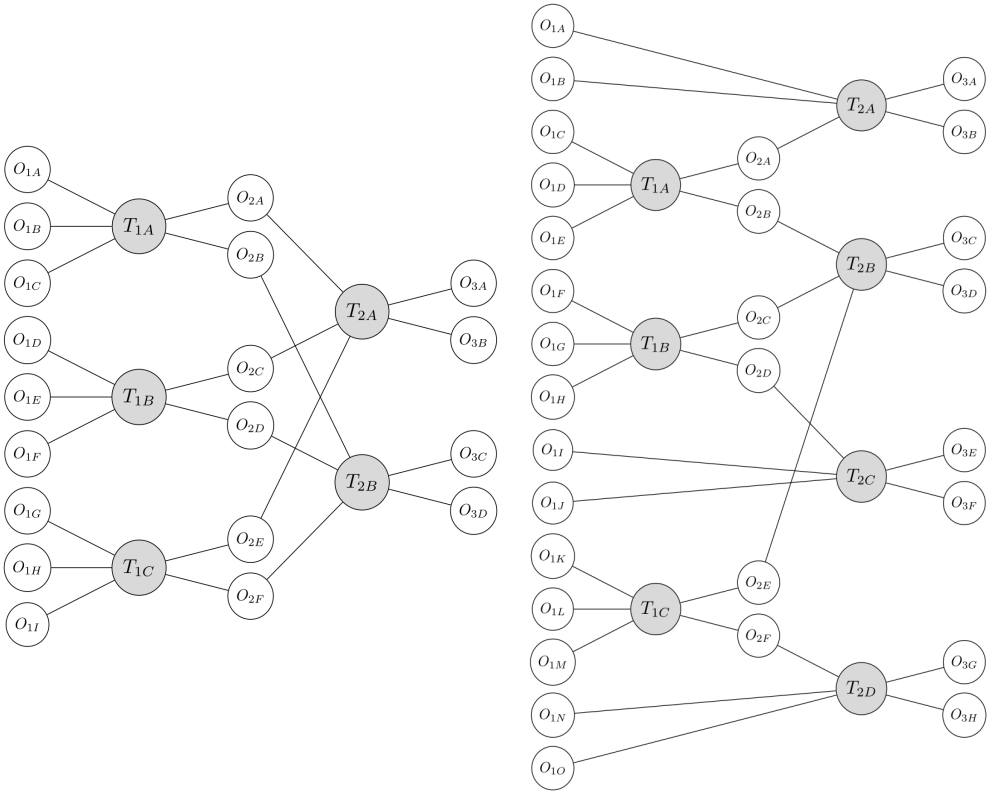

Рис. 4. Симетричний (зліва) та асиметричний (справа) графи консолідації

Розглянемо випадок, коли загальна кількість виходів невелика і всі виходи з попередньої ітерації стають входами для наступної, утворюючи осьово-симетричний граф. Приклад такого графа подано на рис. 4 зліва. На першій ітерації 9 виходів об'єднуються в групи по 3 виходи 3-ма транзакціями (T_{1A}, T_{1B} та T_{1C}). Використовуючи евристичне припущення про єдиного власника входів транзакції, графовий аналіз припускає, що ці 3 транзакції відображають 3-х власників, і їх виходи – 6-х власників (3-х нових отримувачів та 3-х попередніх). При цьому завдяки кроку 2 алгоритму, наведеного вище, неможливо відрізнити, який з виходів є цільовим, а який – рештою. Але на наступній ітерації транзакції T_{2A} та T_{2B} об'єднують групи по 3 виходи з попередньої ітерації, зменшуючи кількість потенційних власників з 6 до 2.

Для того, щоб унеможливити подібні припущення під час аналізу графа транзакцій, можна розширити крок 1 алгоритму, наведеного вище, правилом, що якщо 3 виходи попередньої ітерації стали входами однієї транзакції на даній ітерації, то інші виходи з тих же транзакцій попередньої ітерації повинні стати входами різних транзакцій наступної ітерації. Приклад такого графа зображений на рис. 4 справа. Тут друга ітерація консолідації (транзакції T_{2A}, T_{2B}, T_{2C} та T_{2D}) не зменшує кількості потенційних власників виходів попередньої ітерації (транзакцій T_{1A}, T_{1B} та T_{1C}). Окрім цього, утворений патерн графа не має осьової симетрії і є значно складнішим для розпізнавання навіть візуально.

Порівняємо вартість здійснення простої консолідації N виходів однією транзакцією та багатокрокової консолідації N виходів симулюючими транзакціями, кожна з яких має K виходів. Оскільки комісія сплачується за розмір транзакції у віртуальних байтах, то достатньо порівняти сумарний розмір всіх транзакцій для обох випадків. Середній розмір входу становить 108 байтів, а середній розмір виходу – 33 байти (оцінки зроблені за допомогою калькулятора розміру транзакції

[9]). Для зручності порівняння припустимо, що багатокрокова консолідація триває до отримання єдиного виходу. Вартість простої консолідації – це сумарна вартість N входів та одного результуючого виходу. Вартість багатокрокової консолідаці – це сума вартостей всіх ітерацій, де вартість однієї ітерації – це сума вартостей всіх входів для виходів, що були створені попередньою ітерацією, та всіх виходів, що будуть створені даною ітерацією. Кількість ітерацій можна обчислити з параметрів N та K. Відношення вартостей багатокрокової консолідації та простої консолідації можна обчислити за допомогою наступної процедури, поданої мовою Python:

```
def CostIncrease(n, k):
    simple_cost = n * SIN + SOUT
    complex_cost = 0
    n_steps = math.floor(math.log(n, k / 2))
    n_left = n
    for i in range(n_steps):
        step_cost = n_left * SIN + 2 * (n_left / k) * SOUT
        complex_cost += step_cost
        n_left = 2 * (n_left / k)
    return complex_cost / simple_cost
```

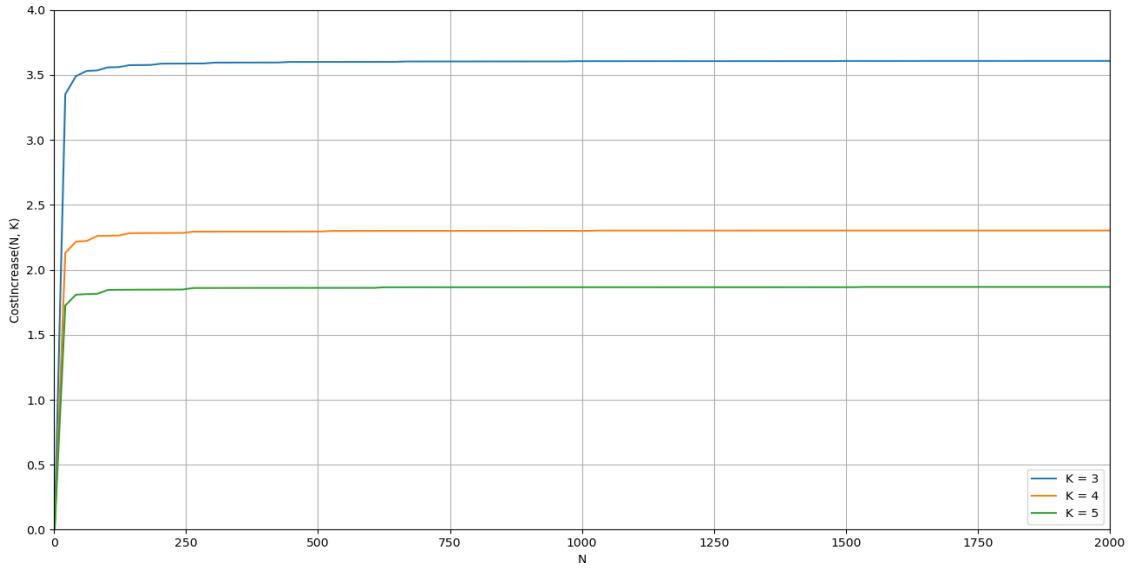

Рис. 6. Збільшення вартості операції консолідації

Як видно з графіків на рис. 6, коефіцієнт збільшення вартості залежить лише від кількості виходів K у симулюючих транзакціях і не залежить від кількості накопичених виходів N для достатньо великих N. При цьому для випадку K = 3, що мінімізує розмір груп виходів, між якими встановлюються зв'язки, а тому є найгіршим з точку зору ефективності, збільшення комісії за операцію консолідації становить приблизно 3.6 рази, що є допустимим, оскільки консолідація зазвичай здійснюється в періоди малого навантаження на мережу, коли спостерігається мінімальна допустима комісія (1 базова одиниця за байт). Наприклад, станом на 9 квітня 2025 року в умовах мінімальної комісії у мережі приблизна вартість простої консолідації 100 виходів становить 8.31 USD, тоді як вартість багатокрокової консолідації цих же виходів симулюючими транзакціями становить 30.50 USD.

Висновки та перспективи подальших досліджень. Запропонований метод багатокрокової консолідації транзакційних виходів за допомогою симуляції витрачання дозволяє зменшити кількість дрібних виходів, що належать користувачу, об'єднуючи їх у більші виходи. Окремі транзакції у такій консолідації не зв'язують між собою великі групи виходів, на відміну від простої консолідації однією транзакцією, а також не утворюють дерев консолідації у графі транзакцій, і тому не зв'язують між собою виходи, що не є зв'язаними в межах однієї транзакції, на відміну від примітивної багатокрокової консолідації. Збереження приватності досягається за рахунок

зменшення ефективності: багатокрокова консолідація за визначенням є дорожчою, ніж проста консолідація однією транзакцією, з точки зору кількості записів у блоках Біткоїна (а отже і сумарної комісії за виконання операції). При цьому збільшення вартості консолідації практично не залежить від кількості накопичених виходів і є прийнятним за умови відсутності навантаження на мережу під час здійснення консолідації (приблизно 3.6 рази у найгіршому випадку).

Важливим напрямком подальших досліджень є вивчення можливостей розпізнавання складних методів консолідації виходів на основі певних особливостей результуючого патерну у графі Біткоїн-транзакцій (симетричності, тощо). Пов'язаною з проблемою розпізнавання консолідації виходів є проблема розпізнавання зворотної операції: розбиття великого виходу, що є, наприклад, результатом крадіжки біткоїна, на менші виходи з метою їх легалізації.

Ще одним цікавим напрямком подальших досліджень є методи як здійснення, так і розпізнавання консолідації виходів у графі транзакцій, що використовують розширення Біткоїн-протоколу під назвою Taproot [10]. Це розширення впроваджує альтернативний метод криптографічних підписів з підтримкою агрегації і тому дозволяє створювати кооперативні виходи з багатьма власниками, які при цьому не відрізняються від звичайних виходів з єдиним власником, а отже евристичні припущення щодо єдиного власника всіх входів таких транзакцій може бути неактуальним. Зокрема підграф транзакції консолідації та підграф транзакції, що створює кооперативний вихід, виглядатиме однаково.

Список бібліографічного опису

1. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. 9 p. URL: https://bitcoin.org/bitcoin.pdf (date of access: 15.04.2025).
2. Privacy. 2025. URL: https://en.bitcoin.it/wiki/Privacy (date of access: 15.04.2025).
3. Poon J., Dryja T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. 2016. 59 p. URL: https://lightning.network/lightning-network-paper.pdf (date of access: 15.04.2025).
4. Maxwell G. CoinJoin: Bitcoin privacy for the real world. URL: https://bitcointalk.org/?topic=279249 (date of access: 15.04.2025).
5. Ghesmati S., Fdhila W., Weippl E. Studying Bitcoin privacy attacks and their Impact on Bitcoin-based Identity Methods. International Conference on Business Process Management. 2021. URL: https://eprint.iacr.org/2021/1088.pdf (date of access: 15.04.2025).
6. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In Sadeghi, A.R., ed.: Financial Cryptography and Data Security. Volume 7859 of Lecture Notes in Computer Science. Springer Berlin Heidelberg. 2013. P. 6-24. URL: https://eprint.iacr.org/2012/584.pdf (date of access: 15.04.2025).
7. Security through obscurity. 2025. URL: https://en.wikipedia.org/wiki/Security_through_obscurity (date of access: 15.04.2025).
8. Nick J. D. Data-Driven De-Anonymization in Bitcoin : Master Thesis. Zürich, 2015. 34 p. URL: https://doi.org/10.3929/ethz-a-010541254 (date of access: 15.04.2025).
9. Transaction size calculator. Bitcoin Optech. URL: https://bitcoinops.org/en/tools/calc-size/ (date of access: 15.04.2025).
10. Wuille P., Nick J., Towns A. BIP 0341: Taproot: SegWit version 1 spending rules. 2020. URL: https://en.bitcoin.it/wiki/BIP_0341 (date of access: 15.04.2025).

References

1. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. 9 p. URL: https://bitcoin.org/bitcoin.pdf (date of access: 15.04.2025).
2. Privacy. 2025. URL: https://en.bitcoin.it/wiki/Privacy (date of access: 15.04.2025).
3. Poon J., Dryja T. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. 2016. 59 p. URL: https://lightning.network/lightning-network-paper.pdf (date of access: 15.04.2025).
4. Maxwell G. CoinJoin: Bitcoin privacy for the real world. URL: https://bitcointalk.org/?topic=279249 (date of access: 15.04.2025).
5. Ghesmati S., Fdhila W., Weippl E. Studying Bitcoin privacy attacks and their Impact on Bitcoin-based Identity Methods. International Conference on Business Process Management. 2021. URL: https://eprint.iacr.org/2021/1088.pdf (date of access: 15.04.2025).
6. Ron, D., Shamir, A.: Quantitative analysis of the full bitcoin transaction graph. In Sadeghi, A.R., ed.: Financial Cryptography and Data Security. Volume 7859 of Lecture Notes in Computer Science. Springer Berlin Heidelberg. 2013. P. 6-24. URL: https://eprint.iacr.org/2012/584.pdf (date of access: 15.04.2025).
7. Security through obscurity. 2025. URL: https://en.wikipedia.org/wiki/Security_through_obscurity (date of access: 15.04.2025).
8. Nick J. D. Data-Driven De-Anonymization in Bitcoin : Master Thesis. Zürich, 2015. 34 p. URL: https://doi.org/10.3929/ethz-a-010541254 (date of access: 15.04.2025).
9. Transaction size calculator. Bitcoin Optech. URL: https://bitcoinops.org/en/tools/calc-size/ (date of access: 15.04.2025).
10. Wuille P., Nick J., Towns A. BIP 0341: Taproot: SegWit version 1 spending rules. 2020. URL: https://en.bitcoin.it/wiki/BIP_0341 (date of access: 15.04.2025).